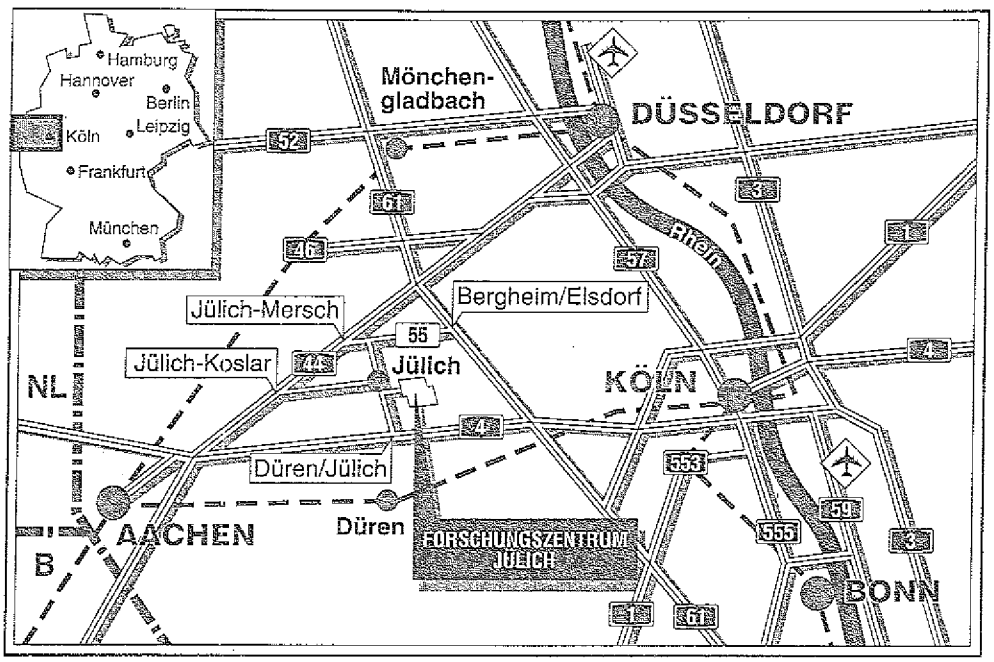
FORSCHUNGSZENTRUM JÜLICH GmbH

Institut für Kernphysik

Eine Influenzdriftkammer mit Flash ADC Auslese

Olaf Steinkamp



Berichte des Forschungszentrums Jülich ; 2420
ISSN 0366-0885
Institut für Kernphysik Jül-2420

Zu beziehen durch: Forschungszentrum Jülich GmbH · Zentralbibliothek
Postfach 1913 · D-5170 Jülich · Bundesrepublik Deutschland
Telefon: 02461/61-6102 · Telefax: 02461/61-6103 · Telex: 833556-70 kfa d

Eine Influenzdriftkammer mit Flash ADC Auslese

Olaf Steinkamp

Inhalt

Kapitel 1

Motivation

Das Institut für Kernphysik 1 (IKP1) des Forschungszentrums Jülich (KFA) ist im Rahmen einer internationalen Kollaboration an dem Experiment PS185 [KIL81] am 'Low Energy Antiproton Ring' (LEAR) am CERN beteiligt. Ein weiterer Schwerpunkt seiner Aktivitäten liegt in der Planung und Vorbereitung von Experimenten für den Protonenspeicherring COSY (*Cooler Synchrotron*), der zur Zeit an der KFA aufgebaut wird und 1992 in Betrieb gehen soll.

Sowohl für das Experiment PS185 als auch für verschiedene COSY–Experimente ist der Einsatz einer Influenzdriftkammer (IDC[1]) als Vertex–Detektor geplant.

Die Anforderungen, die sich für einen solchen Detektor ergeben

- gute Ortsauflösung,
- hohe Ratenverträglichkeit und
- geringe Massenbelegung,

sollen am Beispiel des Experiments PS185 näher erläutert werden.

PS185

In dem Experiment PS185 wird die Produktion von Strangeness tragenden Hyperon–Antihyperon–Paaren im Proton–Antiproton–Stoß nahe der Schwelle untersucht. Im Detail sind das die Reaktionen:

[1] Die Abkürzung IDC folgt aus der englischen Bezeichnung *Induction Drift Chamber*.

$$\begin{aligned}
\overline{p}p &\rightarrow \overline{\Lambda}\Lambda \\
\overline{p}p &\rightarrow \overline{\Sigma^0}\Sigma^0 \\
\overline{p}p &\rightarrow \Lambda\overline{\Sigma^0} + c.c. \\
\overline{p}p &\rightarrow \overline{\Sigma^\pm}\Sigma^\pm
\end{aligned}$$

Bisher konzentrierten sich die Messungen auf die Produktion von neutralen $\Lambda\overline{\Lambda}$– und $\Lambda\overline{\Sigma^0}$–Hyperon–Paaren, sie sollen aber auf die geladenen $\Sigma^\pm\overline{\Sigma^\pm}$–Kanäle erweitert werden. Die mittleren Lebensdauern τ, die Reichweiten $c \cdot \tau$ und die wichtigsten Zerfallskanäle dieser Teilchen sind in der folgenden Tabelle zusammengefaßt:

Teilchen	τ $[10^{-10}s]$	$c \cdot \tau$ [cm]	Zerfallskanäle
Σ^+	0,799	2,40	$p\pi^0$ $n\pi^+$
Σ^-	1,479	4,44	$n\pi^-$

Im Gegensatz zu den bisher gemessenen neutralen Teilchen zerfallen die $\Sigma^\pm\overline{\Sigma^\pm}$ jeweils in ein geladenes und ein neutrales Teilchen. Da das neutrale Teilchen im Detektor nicht gesehen wird, ist es zur kinematischen Rekonstruktion eines Ereignisses notwendig, die Spuren und die Zerfallsorte der $\Sigma^\pm$ und $\overline{\Sigma^\pm}$ selbst zu vermessen. Aus der Reichweite dieser Teilchen folgt die Forderung nach einem Detektor mit einer Ortsauflösung $\leq 50\mu m$ und der Fähigkeit, eine genügende Anzahl von Meßpunkten auf dieser Länge zu erhalten. Die $\Sigma^\pm\overline{\Sigma^\pm}$ werden bei den am LEAR verfügbaren Strahlimpulsen unter einem Winkel $\leq 15^o$ zur Strahlrichtung erzeugt. Der Detektor muß also auch im Bereich des Strahls sensitiv sein. Aus der $\overline{p}$–Rate folgt daraus die Forderung nach einer Ratenverträglichkeit von $\geq 10^6/(s \cdot \mathrm{mm}^2)$. Die Massenbelegung des Detektors sollte aus zwei Gründen gering sein: zum Einen, um zusätzlichen Untergrund durch Reaktionen des Strahls im Detektor zu vermeiden, zum anderen, um Coulomb–Vielfach–Streuung der Σ s zu verringern.

Hinsichtlich der Ortsauflösung und der Ratenverträglichkeit stellen Halbleiterzähler sicherlich eine günstige Lösung dar, doch ist ihre Massenbelegung etwa einen Faktor zehn größer als die von Vieldrahtkammern. Bei herkömmlichen Driftkammern, wo die Ortsauflösung aus einer Messung der Driftzeit resultiert, schließen sich jedoch eine gute Ortsauflösung und hohe Ratenverträglichkeit gegenseitig aus. Die Influenzdriftkammer umgeht dieses Problem.

Influenzdriftkammer

Die Influenzdriftkammer wurde von Walenta in Siegen vorgeschlagen und entwickelt [ROD86, WAL88]. Sie ist ein neuer Typ von Vieldrahtkammer, der alle gestellten Anforderungen erfüllen kann. Der Teilchennachweis beruht auf der Ionisation in einem Gas, weshalb eine geringe Massenbelegung möglich ist. Die Ortsauflösung wird aber nicht durch eine Driftzeitmessung, sondern durch die Messung influenzierter Signale erreicht. Die bei einer Primärionisation erzeugten Elektronen driften auf einen Signaldraht, wo sich eine Ladungslawine ausbildet. Die Asymmetrie der von dieser Ladungslawine auf die benachbarten Potentialdrähte influenzierten Signale ist ein Maß für den Ort der Primärionisation. Aus der hohen Granularität des Detektors (Drahtabstände $\approx 300\,\mu m$, Zelldicken $\approx 3\,mm$), folgt die hohe Ratenverträglichkeit und gute Ortsauflösung. Eine vorsichtige Abschätzung für die Ratenverträglichkeit ist $10^8/(\mathrm{cm}^2 \cdot s)$, eine Ortsauflösung von $\sigma = 25\mu m$ für senkrechten Teilcheneinfall ist gemessen worden ([WAL88]). Die Messungen haben aber auch gezeigt, daß die Ortsauflösung sich in der bislang getesteten Version einer symmetrischen Kammer mit ADC–Auslese erheblich verschlechtert, wenn die Spur unter einem Winkel Θ zur Senkrechten durch die Zelle läuft (in erster Näherung mit $\approx 6\mu m/1°$, [KRÄ86]). Wie in Simulationsrechnungen der IDC gezeigt wurde [WAL89, WAL88], resultiert diese Verschlechterung aus statistischen Schwankungen der Primärionisation entlang der Spur des Teilchens.

Deshalb besteht die Erwartung, daß die Ortsauflösung, insbesondere bei geneigten Spuren, sich durch die Auslese des Signals mit 100MHz–Flash–ADCs (FADCs) erheblich verbessern läßt, da es mit diesen möglich sein sollte, eine zeitliche Struktur der Signale aufzulösen und den Einfluß der statistischen Schwankungen zu reduzieren. Anschaulich erreicht man mit der FADC–Auslese über die Driftzeit der Elektronen eine begrenzte Ortsauflösung in der Richtung senkrecht zur Drahtebene. Zur Vermeidung von Ambiguitäten zwischen Elektronen, die von oben, und solchen, die von unten auf den Signaldraht driften, muß eine solche Kammer asymmetrisch aufgebaut sein, das heißt eine Kathode muß möglichst nah an die Drahtebene herangeführt werden. Labormessungen haben gezeigt, daß eine Ebene einer solchen asymmetrischen IDC bis zu einem Kathodenabstand von $200\,\mu m$ stabil betrieben werden kann ([GEI88]).

Im Rahmen der vorliegenden Arbeit wurde zum ersten Mal eine asymmetrische Testkammer mit mehreren Ebenen aufgebaut. Die Elektronik wurde zusammengestellt und getestet. Eine Testmessung mit 3 GeV Elektronen fand am DESY statt. Technische Probleme mit den Hochspannungs–Zuführungen ergaben erhebliche Zeitverzögerungen während der Testmessung, so daß in

der zur Verfügung stehenden Zeit nur mit einer geringen Gasverstärkung und nur einer Gassorte (Argon/Methan) gemessen werden konnte. Aussagen über die Ortsauflösung bleiben daher weiteren Messungen mit höherer Gasverstärkung und dichteren Gasen (zum Beispiel Isobutan) vorbehalten, die im Sommer 1990 stattfinden werden. Anhand der genommenen Daten konnten jedoch Auswerte–Algorithmen entwickelt und getestet werden, erste Aussagen über die Auslese mit FADCs sind möglich.

Kapitel 2

Grundlagen

2.1 Prinzip der Influenzdriftkammer (IDC)

Eine Kammerebene einer Influenzdriftkammer, wie sie von Walenta et. al. [ROD86] [WAL88] entwickelt wurde, besteht aus einer Drahtebene und zwei symmetrischen Drifträumen der Dicke L, die durch Kathodenfolien begrenzt sind (siehe Abbildung 2.1). Die Drahtebene besteht aus sich abwechselnden dünnen Anodendrähten und dickeren Potentialdrähten mit dem Drahtabstand d. Eine Driftzelle ist definiert durch einen Anodendraht, die beiden benachbarten Potentialdrähte und die Kathodenfolien (im Bild gestrichelt eingezeichnet). Der Drahtabstand beträgt 300 μm, der Abstand zwischen der Drahtebene und den Kathodenfolien 3 mm. Damit ergibt sich eine Zellgröße von $600\,\mu m \cdot 6\,mm$.

Die Potentiale auf den Drähten und den Kathoden werden so gewählt, daß das Feld um den Anodendraht zylindersymmetrisch, außerhalb der Drahtebenen homogen ist (siehe Abbildungen 2.2 und 2.3)[1]. Es besteht also ein linearer Zusammenhang zwischen der x–Koordinate einer Feldlinie im homogenen Bereich und dem Winkel α, unter dem diese Feldlinie auf den Anodendraht trifft.

Beim Durchgang eines geladenen Teilchens werden durch Ionisation des Kammergases Cluster von Elektron–Ion–Paaren erzeugt. Die Elektronen driften entlang einer Feldlinie zum Anodendraht, den sie unter dem Winkel α_0 erreichen. Für eine Primärionisation an der Stelle (x_0, z_0) im Bereich des homogenen Feldes ist $\alpha_0 \propto x_0$ und unabhängig von z_0. Damit kann man umgekehrt

[1] Die Feldlinienberechnung erfolgte mit einem von Weihs und Zech in Siegen entwickelten Programm ([WEI89]).

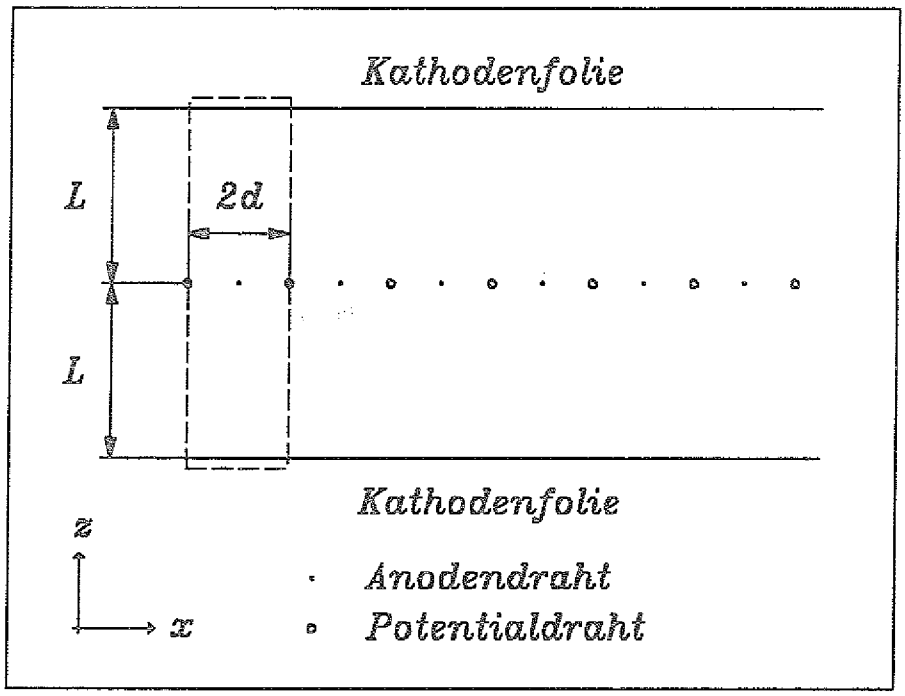


Abbildung 2.1: *Aufbau einer symmetrischen Influenzdriftkammer. Gestrichelt sind die Grenzen einer Driftzelle eingezeichnet.*

aus einer Messung des Winkels α_0 die x–Koordinate der Primärionisation bestimmen:

$$x_0 = k \cdot \alpha_0.$$

Der Proportionalitätsfaktor k ergibt sich aus dem Drahtabstand ($x_{max} = d = 300\,\mu m$) und dem Winkelbereich, unter dem die Feldlinien aus dem homogenen Feldbereich auf den Anodendraht treffen ($\alpha_{max} \approx 50°$):

$$k = \frac{x_{max}}{\alpha_{max}} \approx \frac{300\,\mu m}{50°}.$$

Abbildung 2.4 zeigt die elektrische Feldstärke längs eines Schnittes durch einen Anodendrahtes in der Richtung senkrecht zur Drahtebene (berechnet mit dem Siegener Feldprogramm ([WEI89])). Im Bereich hoher Feldstärken in der Nähe des Anodendrahtes wird die kinetische Energie der Elektronen so groß, daß sie durch Stoßionisation sekundäre Elektron–Ion–Paare erzeugen; es entsteht eine Ladungslawine. Bei nicht zu hoher Gasverstärkung ($\leq 5 \cdot 10^4$) bleibt diese Lawine auf dem Anodendraht lokalisiert [WAL78, MAT88], so daß die Information über den Winkel α_0 erhalten bleibt.

Die erzeugten Sekundärelektronen driften innerhalb von weniger als einer Nanosekunde auf den Anodendraht. Zurück bleibt eine Ionenwolke mit der

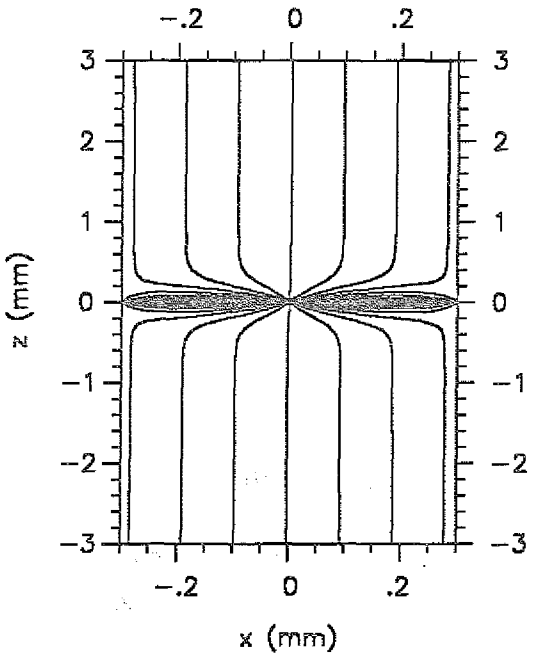


Abbildung 2.2: *Feldlinien in einer Zelle einer symmetrischen Influenzdriftkammer. Die Darstellung ist in der z–Achse um einen Faktor sieben gestaucht.*

positiven Ladung q^+, die langsamer (Millisekunden) zu den Kathoden abfließt. Diese Ladungswolke influenziert Ladungen auf allen Leitern in ihrer Umgebung, insbesondere auch auf den benachbarten Potentialdrähten. Da die Ladungswolke auf einer Seite des Anodendrahts lokalisiert ist, sind die auf den beiden Potentialdrähten influenzierten Signale Q_l und Q_r unterschiedlich groß.

Die auf einen Draht influenzierte Ladung ist gegeben durch

$$Q_{inf} = f_{Draht}(\vec{r}) \cdot q^+.$$

Für jeden Draht kann man die Wichtungsfunktion $f_{Draht}(\vec{r})$ nach dem Gaußschen Reziprozitätstheorem[2] aus dem Potential gewinnen, das an der Stelle $\vec{r}$ herrscht, wenn der betrachtete Draht auf Einheitspotential und alle anderen Elektroden auf Masse liegen (dazu siehe [WAL78]). Für eine qualitative Abschätzung des Effekts ist aber eine Kenntnis der Wichtungsfunktion nicht notwendig.

Mit den in Abbildung 2.5 eingeführten Bezeichnungen ist

[2]Gaußsches Reziprozitätstheorem: Seien (V_i, q_i) und $(\tilde{V}_i, \tilde{q}_i)$ zwei Sätze von Potentialen und Ladungen in einem gegebenen System von Leitern, so gilt: $\sum_i V_i \tilde{q}_i = \sum_i \tilde{V}_i q_i$.

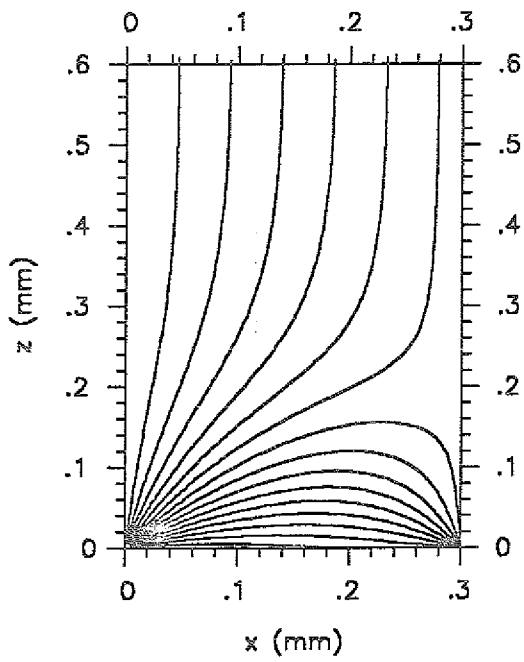


Abbildung 2.3: *Feldlinien in der Nähe der Drahtebene.*

$$
\begin{aligned}
Q_l &= f_p(r_l) \cdot q^+ \\
Q_r &= f_p(r_r) \cdot q^+ \\
\Delta P &= Q_l - Q_r \\
&= (f_p(r_l) - f_p(r_r)) \cdot q^+ \\
&\approx (f_p(d + s\sin\alpha) - f_p(d - s\sin\alpha)) \cdot q^+ \\
&\approx 2s \cdot \sin\alpha \cdot \left.\frac{df_p}{dr}\right|_{r=d} \cdot q^+
\end{aligned}
$$

Die Differenz ΔP der Potentialdrahtsignale ist also proportional der Ladung q^+ und dem Sinus des Winkels α. Normiert man das Differenzsignal mit dem Anodensignal $A = Q_a$, so ist das Ergebnis unabhängig von dem absoluten Wert der Ladung q^+:

$$\frac{\Delta P}{A} = \frac{Q_l - Q_r}{Q_a} \propto 2s \cdot \sin\alpha.$$

Diese Abhängigkeit der Differenz der influenzierten Ladungen vom Sinus des Einfallswinkels ist von Walenta experimentell bestätigt worden [WAL78].

Man erhält somit aus der Messung der auf die Potentialdrähte influenzierten

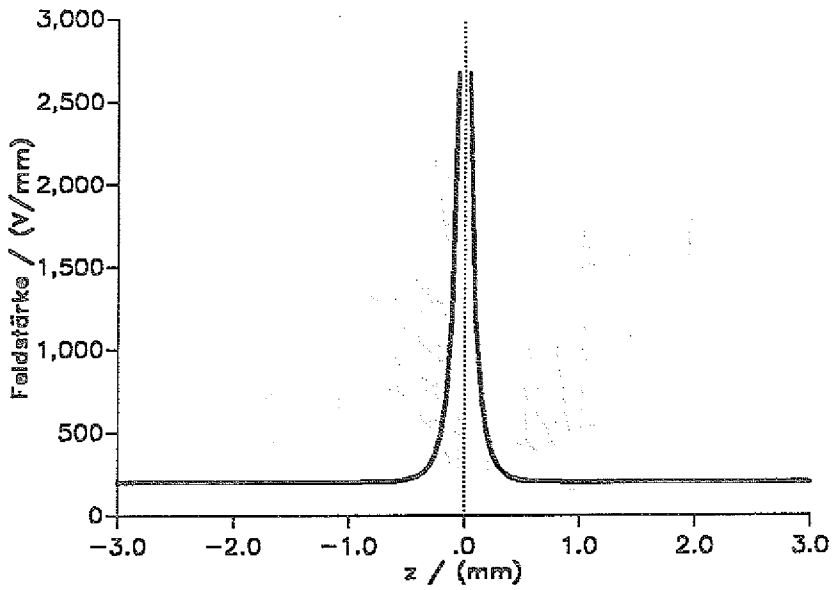


Abbildung 2.4: *Elektrische Feldstärke längs eines Schnittes durch einen Anodendraht, senkrecht zur Drahtebene.*

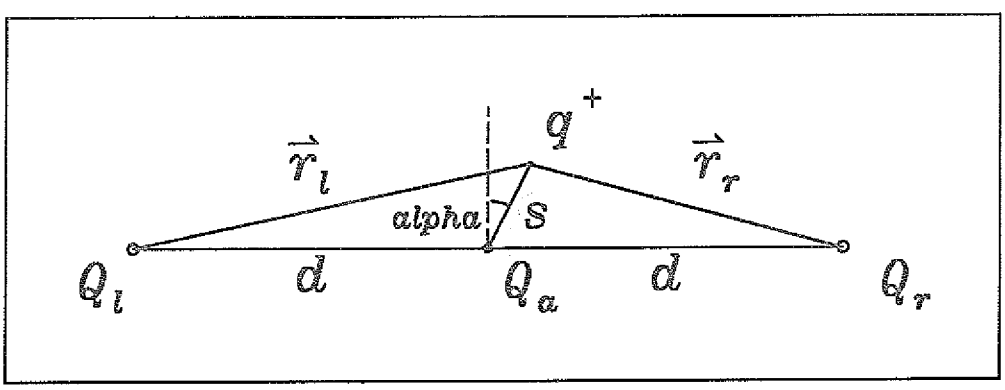


Abbildung 2.5: *Zur Berechnung der von einer Ladung q^+ auf die Potentialdrähte influenzierten Ladungen Q_l und Q_r.*

Ladungen eine Ortsauflösung in der x-Koordinate:

$$\begin{aligned} x_0 &\propto \alpha_0 \\ &\propto \arcsin\left(\frac{\Delta P}{A}\right) \end{aligned}$$

Testmessungen mit symmetrischen Influenzdriftkammern ergeben eine Orts-

auflösung von $\sigma_x = 25\,\mu m$ für Spuren, die die Kammer senkrecht zur Drahtebene durchqueren [ROD86, KRÄ86]. Diese Messungen zeigen aber auch, daß sich die Ortsauflösung bei geneigten Spuren verschlechtert. Auf die die Ortsauflösung begrenzenden Effekte wird im folgenden Abschnitt eingegangen.

2.2 Ortsauflösung der IDC

Abgesehen von 'technischen' Problemen, wie dem Rauschen der Ausleseelektronik und Ungenauigkeiten bei der mechanischen Fertigung der Kammer, ist die Ortsauflösung der IDC begrenzt durch die folgenden physikalischen Effekte:

- die transversale Diffusion der Primärelektronen,
- die räumliche Ausdehnung der Ladungslawine am Draht,
- δ–Elektronen,
- statistische Schwankungen in der Primärionisation bei geneigten Spuren und
- statistische Schwankungen der Ionisationen im inhomogenen Feldbereich.

Im Folgenden soll ein Überblick über die Beiträge dieser Effekte zur Ortsauflösung gegeben werden. Sie hängen von der Geometrie der Kammer, der Hochspannung und den Eigenschaften des Kammergases ab. Zahlenwerte werden für eine Kammer mit einem Drahtabstand $d = 300\,\mu m$ und einem Kathodenabstand von $L = 3\,mm$ und für zwei Gase angegeben: Ar/CH_4 (90/10) und reines i–C_4H_{10}. Eine Zusammenfassung der für die Rechnungen relevanten Parameter dieser Gase findet sich in der folgenden Tabelle (alle Werte aus [FEH83]):

	Ar/CH_4 (90/10)	i–C_4H_{10}
Z/A (gemittelt)	0,5	0,86
ρ (g/cm^3)	$1,6 \cdot 10^{-3}$	$2,42 \cdot 10^{-3}$
dE/dx (keV/cm)	2,34	4,50
n_p (1/cm bei 1 atm)	28	(46)
n_e (1/cm bei 1 atm)	90	195
$\sigma_T(\mu m/\sqrt{\text{cm}})$ (bei 2kV/cm und 1 atm)	600	100

wobei

$$
\begin{aligned}
Z &= \text{Kernladungszahl,} \\
A &= \text{Atomgewicht,} \\
\rho &= \text{Dichte,} \\
dE/dx &= \text{Wahrscheinlichste Energieabgabe pro Längeneinheit,} \\
n_p &= \text{Anzahl der pro Längeneinheit erzeugten Ladungscluster,} \\
n_e &= \text{Anzahl der pro Längeneinheit erzeugten Elektron–Ion–Paare und} \\
\sigma_T &= \text{Transversale Diffusion der Elektronen.}
\end{aligned}
$$

Ferner wird angenommen, daß die Signale mit ADCs ausgelesen werden, das heißt, daß die Ladungen auf den Drähten über die gesamte Signallänge integriert werden.

Diffusion der Primärelektronen

Der Driftbewegung der Elektronen entlang der Feldlinien überlagert ist eine statistische Bewegung. Sie führt dazu, daß sich aus annähernd punktförmigen Ladungsclustern mit zunehmender Driftzeit räumlich ausgedehnte Ladungsverteilungen entwickeln. Die Ortsauflösung ist begrenzt durch die statistischen Schwankungen der Schwerpunkte dieser Ladungsverteilungen, also

$$\Delta x_{Diff} = \frac{1}{\sqrt{n_T}} \cdot \sigma_{trans},$$

wobei n_T die Anzahl der in der Zelle erzeugten Elektron–Ion–Paare und σ_{trans} die transversale Diffusion in der Zelle sind. Bei einer Spurlänge von 6 mm und einer maximalen Driftstrecke der Elektronen von 3 mm ergeben sich für die beiden Gase (bei 2 kV/cm und Normaldruck) folgende Werte:

	Ar/CH_4 (90/10)	i–C_4H_{10}
n_T	54	117
σ_{trans}	330 μm	55 μm
Δx_{Diff}	45 μm	5 μm

Räumliche Ausdehnung der Ladungslawine

Die durch die Gasverstärkung am Anodendraht erzeugte Ladungslawine hat eine räumliche Ausdehnung. Die Bestimmung des Winkels α_0, unter dem die Ladungslawine den Draht erreicht, ist begrenzt durch die statistische Schwankung des Schwerpunkts der Ladungslawine. Sei $\Delta\alpha$ der Fehler bei der Bestimmung des Winkels α_0 für die von einem Primärelektron erzeugte Ladungslawine, so ist für n_T Primärelektronen

$$\Delta x_{av} = \frac{x_{max}}{\alpha_{max}} \cdot \frac{1}{\sqrt{n_T}} \cdot \Delta\alpha.$$

Bei der gewählten Kammergeometrie und Feldkonfiguration ist $x_{max} = 300\mu m$ und $\alpha_{max} \approx 50°$. Übernehmen wir einen in [HUT85] für $\Delta\alpha$ angegebenen Wert von $(8 \pm 1)°$ (gemessen in Dimethyläther), so ergibt sich für die beiden betrachteten Gase die folgende Abschätzung:

	Ar/CH_4 (90/10)	i-C_4H_{10}
n_T	54	117
Δx_{av}	$6,5\,\mu m$	$4,5\,\mu m$

δ–Elektronen

Bei dem Prozeß der Primärionisation überträgt ein geladenes Teilchen durch Stoß an einem Atom des Kammergases diesem eine Energie, die mindestens gleich seiner Ionisationsenergie ist. Überschüssige Energie wird in kinetische Energie des dabei freigesetzten Elektrons umgesetzt. Man nennt dieses Elektron 'δ'–Elektron, wenn seine kinetische Energie so groß ist, daß es sich eine Strecke von der Teilchenspur entfernt und durch Ionisationen längs seiner Spur weitere Ladungscluster erzeugt. Die Ionisationen längs der Spur dieses 'δ'–Elektrons verschlechtern die Ortsauflösung.

Die Wahrscheinlichkeit, δ–Elektronen größerer Reichweite zu erzeugen, ist stark unterdrückt (siehe Abbildung 2.6). Nimmt man also an, daß in einer Zelle nur ein δ–Elektron erzeugt wird, so ist

$$\Delta x_\delta = \int_{-L}^{+L} \int_0^{E_\delta^{max}} \frac{E_\delta}{E_p + E_\delta} \cdot R_\delta \cdot \Phi(E_\delta)\, dE_\delta\, dx$$

wobei

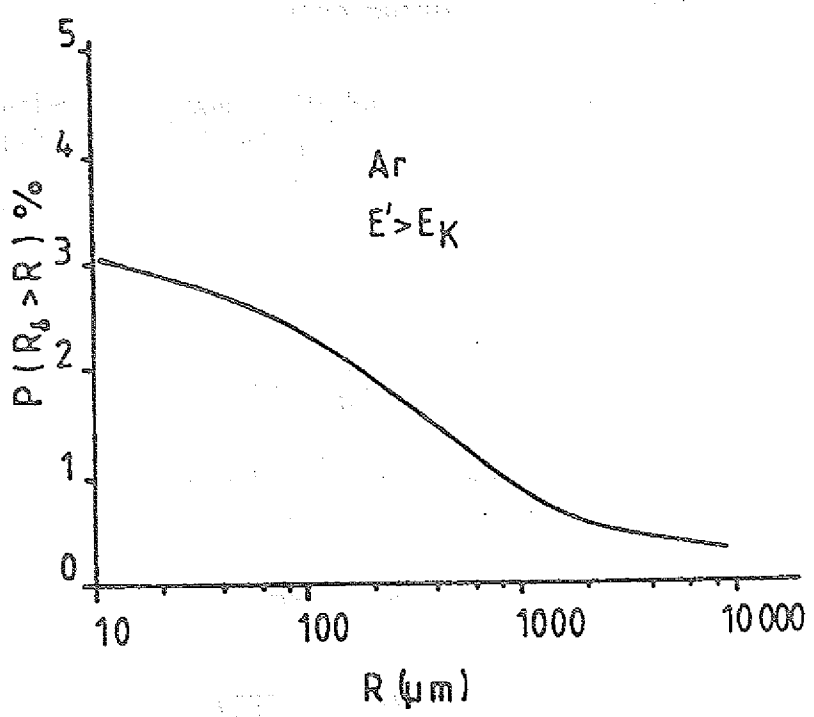


Abbildung 2.6: *Wahrscheinlichkeit, daß ein δ-Elektron mit einer Reichweite $R_\delta \geq R$ erzeugt wird (für Argon bei Normaldruck, aus [WAL83]).*

E_δ	=	Energie des δ–Elektrons
E_δ^{max}	=	obere Grenze für die Energie der δ–Elektronen
E_p	=	gesamte in der Zelle deponierte Energie
R_δ	=	Reichweite des δ–Elektrons
$\Phi(E_\delta)\, dE_\delta\, dx$	=	Wahrscheinlichkeit, ein δ–Elektron der Energie E_δ auf der Länge dx zu erzeugen

Für $\Phi(E_\delta)$ wird in [WAL83] die folgende, auf einer klassischen Rechnung mit freien Elektronen beruhende, Abschätzung angegeben:

$$\Phi(E)\, dE\, dx = \frac{\tilde{A} \cdot \rho}{E^2}\, dE\, dx$$

mit $\tilde{A} = 153,6 \cdot Z/A\, keV cm^2/g$.

Die Reichweite eines δ–Elektrons hängt von seiner Energie ab (siehe Abbildung 2.7), und ist für kleine Werte in erster Näherung dieser proportional. Aus Abbildung 2.7 entnimmt man einen Proportionalitätsfaktor $\approx 40\,\mu$ m/keV.

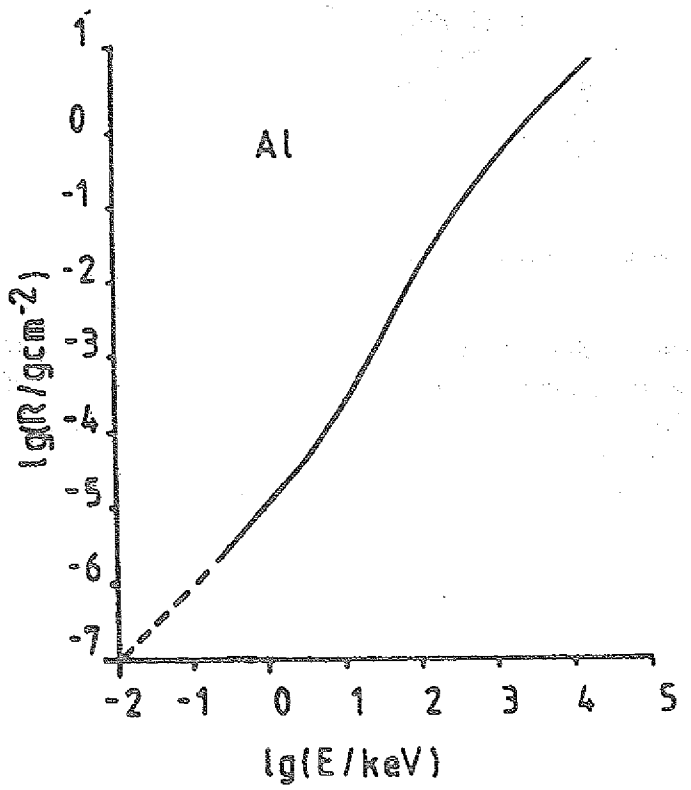


Abbildung 2.7: *Praktische Reichweite eines Elektrons in Abhängigkeit von seiner kinetischen Energie, gemessen für dünne Aluminium-Schichten (aus [WAL83]).*

Ist die Reichweite eines δ–Elektrons groß gegen die Ortsauflösung der Kammer, so kann es aufgrund seiner Spur erkannt werden und trägt somit nicht zur Verschlechterung der Ortsauflösung bei. In [KRÄ86] wird daraus eine maximale Energie der beitragenden δ–Elektronen von 2,8 keV abgeleitet.

Mit diesen Überlegungen folgt für den Beitrag zur Ortsauflösung:

$$\begin{aligned}
\Delta x_\delta &= \int_{-L}^{+L} \int_0^{2,8keV} 40 \frac{\mu m}{keV} \cdot \frac{E_\delta^2}{E_p + E_\delta} \cdot \frac{\tilde{A} \cdot \rho}{E_\delta^2} \, dE_\delta \, dx \\
&= 2L \cdot 40 \frac{\mu m}{keV} \cdot \int_0^{2,8keV} \frac{1}{E_p + E_\delta} \, dE_\delta \\
&= 2L \cdot 40 \frac{\mu m}{keV} \cdot \ln \left(\frac{E_p + 2,8\, keV}{E_p} \right).
\end{aligned}$$

Mit $2L = 6\, mm$ ergeben sich für die beiden Gase die folgenden Werte:

	Ar/CH_4 (90/10)	i-C_4H_{10}
$A \cdot \rho$	0,3 keV/cm	0,1 keV/cm
E_p	2,7 keV	1,4 keV
Δx_δ	5 μm	3 μm

Statistische Schwankungen in der Primärionisation ...

Sowohl die Dichte der von einem geladenen Teilchen entlang seiner Spur erzeugten Ionisationszentren als auch die Zahl der bei jeder Primärionisation erzeugten Elektron–Ion–Paare sind statistisch verteilt. Damit unterliegt auch der Schwerpunkt der Ladungsverteilung statistischen Schwankungen. Sei λ eine Koordinate in Richtung der Teilchenspur, λ_i der Ort des i-ten Ionisationszentrums und n_i die Zahl der erzeugten Elektron–Ion–Paare, dann ist der Ladungsschwerpunkt

$$\overline{\lambda} = \frac{\sum n_i \cdot \lambda_i}{\sum n_i}$$

Für die Schwankung $\Delta\overline{\lambda}$ des Ladungsschwerpunkts wird in [KRÄ86] die folgende obere Abschätzung gegeben:

$$\Delta\overline{\lambda} = \frac{1}{\sqrt{2}} \cdot \Lambda \cdot \frac{\Delta n_T}{n_T}$$

wobei Λ die Länge der Spur, $n_T = \sum n_i$ die Zahl der auf dieser Länge erzeugten Elektron–Ion–Paare und Δn_T deren Schwankung ist. Messungen für verschiedene Gase zeigen

$$\Delta n_T / n_T \approx 1/\sqrt{n_T}.$$

...bei geneigten Spuren

Bei der Auslese werden die Ladungen auf den Drähten mit einer endlichen Zeitkonstante integriert und damit die Signale über eine Anzahl N von Primärionisationen gemittelt. Bei der IDC bedeutet dies, daß man einen über diese Primärionisationen gemittelten Ort

$$\overline{x} = \frac{\sum n_i \cdot x_i}{\sum n_i}$$

bestimmt.

Für Spuren, die die Kammer senkrecht zu den Drahtebenen kreuzen, ist die x–Koordinate für alle Primärionisationen gleich. Dagegen ist für geneigte Spuren mit dem in Abbildung 2.8 eingeführten Winkel Θ

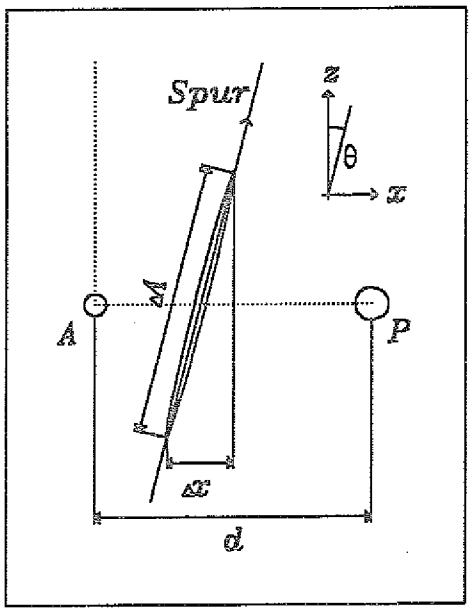


Abbildung 2.8: *Zur Definition des Winkels* Θ.

$$x_i = \lambda_i \cdot \sin\Theta.$$

Aus der statistischen Schwankung $\Delta\overline{\lambda}$ des Ladungsschwerpunkts folgt damit ein Fehler

$$\begin{aligned}\Delta x_\Theta &= \Delta\overline{\lambda} \cdot \sin\Theta \\ &= \frac{1}{\sqrt{2}} \cdot \Lambda \cdot \frac{\Delta n_T}{n_T} \cdot \sin\Theta \\ &= \frac{1}{\sqrt{2}} \cdot 2L \cdot \frac{1}{\sqrt{n_T}} \cdot \tan\Theta\end{aligned}$$

in der Ortsmessung, der der Projektion dieser Schwankung auf die x–Achse entspricht.

Für kleine Winkel Θ ist $\tan\Theta \approx \Theta$ und damit der Fehler proportional dem Winkel. Für die Verschlechterung der Ortsauflösung pro Grad ergeben sich folgende Zahlenwerte:

	Ar/CH_4 (90/10)	i–C_4H_{10}
n_T	54	117
$\Delta x_\Theta/1^o$	$10\,\mu m$	$7\,\mu m$

Die Anzahl der erzeugten Elektron–Ion–Paare ist proportional der Länge der Spur und damit der Zelldicke L. Damit folgt auch:

$$\Delta x_\Theta \propto \sqrt{L}.$$

Der Fehler kann also durch den Übergang zu dünneren Ebenen verringert werden.

... im inhomogenen Feldbereich

Statistische Schwankungen der Primärionisation im inhomogenen Feldbereich führen auch für senkrechte Spuren zu einer Verschlechterung der Ortsauflösung, weil hier der Winkel α, auch von der z–Koordinate der Primärionisation, genauer vom Abstand $|z|$ der Primärionisation von der Drahtebene, abhängt.

Betrachten wir eine senkrechte Spur im Abstand x_0 vom Anodendraht (siehe Abbildung 2.9). Die im homogenen Feldbereich erzeugten Ladungen errei-

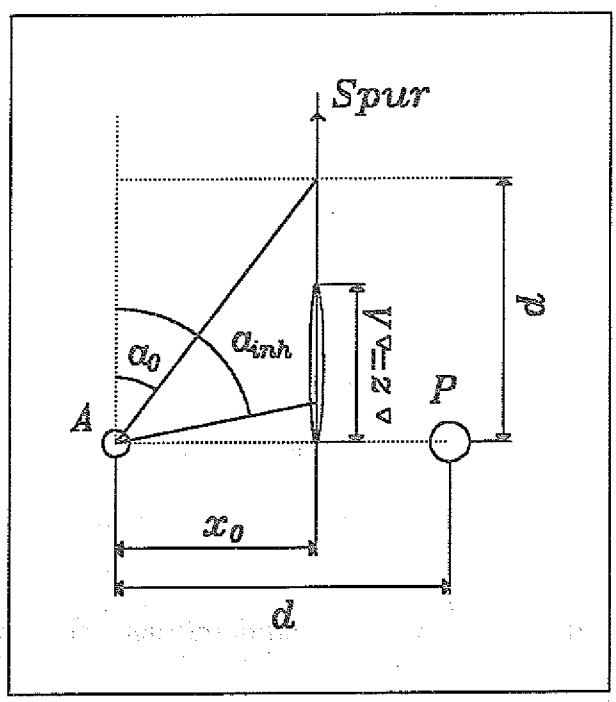


Abbildung 2.9: *Zur Abschätzung des Einflusses statistischer Schwankungen im inhomogenen Feldbereich.*

chen den Draht unter einem Winkel

$$\alpha_0 \approx \arctan \frac{x_0}{d},$$

wobei benutzt wird, daß die Dicke des inhomogenen Feldbereichs in etwa dem Drahtabstand d entspricht.

Die im inhomogenen Feldbereich erzeugten Ladungen treffen im Mittel unter einem Winkel

$$\overline{\alpha_{inh}} = \arctan \frac{x_0}{\overline{z_{inh}}}$$

auf den Draht, wobei

$$\overline{z_{inh}} = \frac{\sum n_i \cdot |z_i|}{\sum n_i}$$

der Schwerpunkt des Abstands der Ladungsverteilung von der Drahtebene ist.

Aus statistischen Schwankungen

$$\begin{aligned} \Delta\overline{z_{inh}} &= \Delta\overline{\lambda_{inh}} \\ &= \frac{1}{\sqrt{2}} \cdot 2d \cdot \frac{\Delta n_{inh}}{n_{inh}} \\ &= \frac{1}{\sqrt{2}} \cdot 2d \cdot \frac{1}{\sqrt{n_{inh}}} \end{aligned}$$

des Ladungsschwerpunkts (n_{inh} sei die Anzahl der im inhomogenen Feldbereich erzeugten Elektron–Ion–Paare) folgen statistische Schwankungen

$$\begin{aligned} \Delta\overline{\alpha_{inh}} &= \frac{\partial\overline{\alpha_{inh}}}{\partial\overline{z_{inh}}} \cdot \Delta\overline{z_{inh}} \\ &= \frac{1}{1+(x_0/\overline{z_{inh}})^2} \cdot \frac{x_0}{\overline{z_{inh}}^2} \cdot \Delta\overline{z_{inh}} \\ &= \frac{1}{1+(x_0/\overline{z_{inh}})^2} \cdot \frac{x_0}{\overline{z_{inh}}^2} \cdot \frac{1}{\sqrt{2}} \cdot 2d \cdot \frac{1}{\sqrt{n_{inh}}} \end{aligned}$$

dieses mittleren Winkels. Sowohl x_0 als auch $\overline{z_{inh}}$ können Werte zwischen 0 und dem Drahtabstand d annehmen. Setzt man für eine Abschätzung des Fehlers $x_0 = \overline{z_{inh}} = d/2$, so ist:

$$\begin{aligned} \Delta\overline{\alpha_{inh}} &= \frac{1}{2} \cdot \frac{2}{d} \cdot \frac{1}{\sqrt{2}} \cdot 2d \cdot \frac{1}{\sqrt{n_{inh}}} \\ &= \frac{\sqrt{2}}{\sqrt{n_{inh}}}. \end{aligned}$$

Das über die gesamte Zelle integrierte Signal setzt sich zusammen aus dem Signal des homogenen und dem des inhomogenen Feldbereichs. Man bestimmt einen 'mittleren' Ort

$$x = \frac{x_{max}}{\alpha_{max}} \cdot \left(\frac{d}{L} \cdot \overline{\alpha_{inh}} + \frac{L-d}{L} \cdot \alpha_0\right).$$

Für den Beitrag Δx_{inh} des inhomogenen Feldbereichs zur Ortsauflösung gilt damit

$$\begin{aligned}\Delta x_{inh} &= \frac{x_{max}}{\alpha_{max}} \cdot \frac{d}{L} \cdot \Delta\overline{\alpha_{inh}} \\ &= \frac{x_{max}}{\alpha_{max}} \cdot \frac{d}{L} \cdot \frac{\sqrt{2}}{\sqrt{n_{inh}}}.\end{aligned}$$

Mit $x_{max} = d = 300\,\mu m$, $L = 3\,mm$ und $\alpha_{max} \approx 50°$ ergeben sich die folgenden Zahlenwerte für die betrachteten Gase:

	Ar/CH_4 (90/10)	i-C_4H_{10}
n_T	5	12
Δx_{inh}	$22\,\mu m$	$14\,\mu m$

Dieser Beitrag hängt ab vom Anteil des inhomogenen Feldbereichs am gesamten Kammervolumen. Dieser ist bei der gewählten Geometrie (d/L = 1/10) klein, wächst aber mit dünner werdenden Ebenen. Damit steht dieser Effekt in Konkurrenz zur oben aus dem Fehler für geneigte Spuren abgeleiteten Forderung nach dünneren Ebenen.

2.3 Asymmetrische IDC mit FADC–Auslese

Bei der Auslese mit herkömmlichen ADCs werden die auf die Drähte influenzierten Ladungen über die gesamte Signallänge integriert. In Abschnitt 2.2 wurde gezeigt, daß dies zu einer Verschlechterung der Ortsauflösung für geneigte Spuren führt, die durch die statistische Verteilung der Primärionisationen entlang der Teilchenspur bedingt ist. Dieser Fehler kann erheblich verringert werden, wenn es gelingt, die zeitliche Struktur des Signals aufzulösen, und damit über die Driftzeit der Elektronen die Verteilung der Primärionisationen längs der Spur zu messen:

Gelingt es, das Signal mit einer Zeitkonstanten Δt aufzulösen, so entspricht dies einer Ortsauflösung $\Delta z = v_{Drift} \cdot \Delta t$ in der Richtung senkrecht zur Drahtebene. Anschaulich erreicht man damit eine Unterteilung des Driftvolumens in $N = L/\Delta z$ Intervalle der Dicke Δz. Für den Fehler $\Delta\tilde{x}_\Theta$ in einem solchen Intervall gilt mit der in Abschnitt 2.2 hergeleiteten Gleichung:

$$\Delta\tilde{x}_\Theta = \frac{1}{\sqrt{2}} \cdot \Delta z \cdot \frac{1}{\sqrt{\bar{n}}} \cdot \tan\Theta$$

wobei $\tilde{n} = 1/N \cdot n_T$ die Zahl der in diesem Intervall und n_T die Zahl der in der Kammerebene erzeugten Elektron–Ion–Paare ist. Das gesamte Driftvolumen entspricht N Meßintervallen, so daß für den Fehler Δx_Θ in einer Ebene gilt:

$$\begin{aligned}
\Delta x_\Theta &= \frac{1}{\sqrt{N}} \cdot \Delta\tilde{x}_\Theta \\
&= \frac{1}{\sqrt{N}} \cdot \frac{1}{\sqrt{2}} \cdot \Delta z \cdot \frac{1}{\sqrt{\tilde{n}}} \cdot \tan\Theta \\
&= \frac{1}{\sqrt{N}} \cdot \frac{1}{\sqrt{2}} \cdot \frac{2L}{2N} \cdot \frac{1}{\sqrt{n_T/N}} \cdot \tan\Theta \\
&= \frac{1}{2N} \cdot \{\frac{1}{\sqrt{2}} \cdot 2L \cdot \frac{1}{\sqrt{n_T}} \cdot \tan\Theta\}
\end{aligned}$$

Der Term in geschweiften Klammern entspricht dem in Abschnitt 2.2 für die Messung des integrierten Signals hergeleiteten Fehler. Dieser kann also um einen Faktor

$$\frac{1}{2N} = \frac{v_{Drift} \cdot \Delta t}{2L}$$

verringert werden.

Auslese mit FADCs

Bei der Auslese mit Flash–ADCs ([FAR83, WAL85]) kann das Signal mit einer Frequenz von 100 MHz abgetastet werden. Die Bandbreite des Signals wird in den FADCs vorgeschalteten Verstärkern auf 30 MHz begrenzt[3], dies entspricht einer Integration der Signale mit einer Zeitkonstanten von etwa 30 ns.

Die Driftgeschwindigkeit der Elektronen hängt von der Gassorte, vom Druck und von der elektrischen Feldstärke ab (siehe Abbildung 2.10). Die Feldstärke im homogenen Feldbereich der IDC beträgt etwa 2kV/cm, damit ergeben sich für Ar/CH_4 (90/10) und für i–C_4H_{10} die folgenden Werte:

	Ar/CH_4 (90/10)	i–C_4H_{10}
v_{Drift}(cm/μs) (bei 2kV/cm und 1 atm)	2,5	4,5

Die Driftgeschwindigkeit kann aber durch geringe Beimengungen eines Quencher–Gases (wie Äthanol) erheblich verringert werden. Werte um 1 $cm/\mu s$ scheinen deshalb realistisch.

[3]Die Bandbreite eines Signals sollte mindestens einen Faktor drei kleiner sein als die Frequenz, mit der es abgetastet wird.

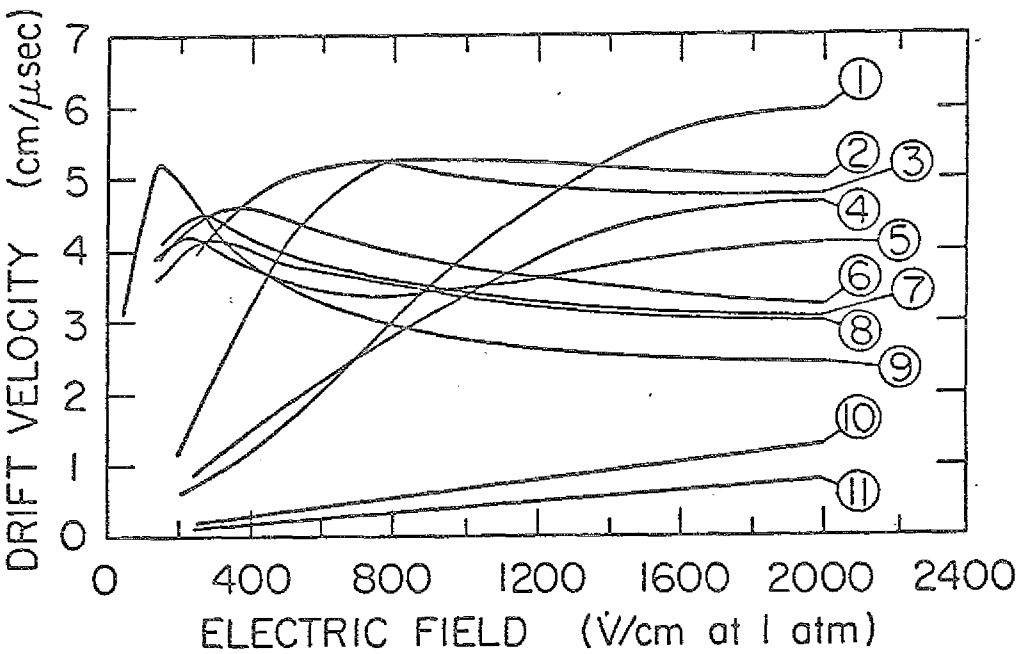


Abbildung 2.10: *Driftgeschwindigkeit für Elektronen in verschiedenen Gasen in Abhängigkeit von der Feldstärke: (4)=i-C_4H_{10}, (9)=Ar/CH_4 (90/10) (aus [FEH83]).*

Damit ergeben sich für $\Delta x_{\odot}/1°$ die folgenden Zahlenwerte:

	Ar/CH_4 (90/10)	i-C_4H_{10}
$v_{Drift} = 4,5\,cm/\mu s$	—	$1,6\,\mu m$
$v_{Drift} = 2,5\,cm/\mu s$	$1,3\,\mu m$	—
$v_{Drift} = 1,0\,cm/\mu s$	$0,5\,\mu m$	$0,35\,\mu m$

Die bei Primärionisationen im inhomogenen Feldbereich um die Drähte erzeugten Elektronen haben den kürzesten Driftweg, erreichen also als erste den Anodendraht. Damit kann die Ortsauflösung bei der Auslese mit FADCs zusätzlich verbessert werden, indem man den ersten Teil des Signals und damit den Beitrag der statistischen Schwankungen der Primärionisation im inhomogenen Feldbereich ausblendet.

Asymmetrische Geometrie

Aufgrund der Messung der Driftzeit kann nicht zwischen Primärionisationen aus der oberen und der unteren Halbebene des Driftvolumens unterschie-

den werden. Um Ambiguitäten zu vermeiden, muß also ein Driftraum ausgeschaltet werden. Daraus folgt die Forderung nach einer asymmetrischen Geometrie der IDC, wie sie Abbildung 2.11 zeigt. Den Feldverlauf in einer

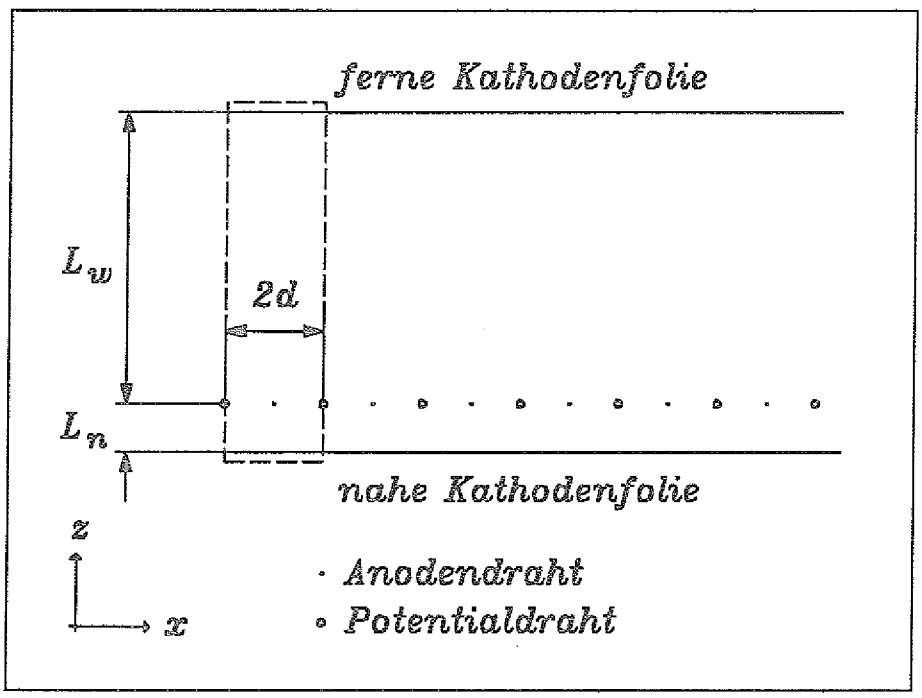


Abbildung 2.11: *Aufbau einer asymmetrischen Influenzdriftkammer.*

Zelle einer asymmetrischen Kammer zeigt Abbildung 2.12. Um zu hohe Feldstärken im schmalen Driftraum zu vermeiden, wird das Potential auf der nahen Kathodenfolie reduziert. Laborversuche mit einer Ebene einer asymmetrischen Kammer haben gezeigt, daß diese bis zu einem Kathodenabstand von $L_n = 200\mu m$ stabil betrieben werden konnte [GEI88].

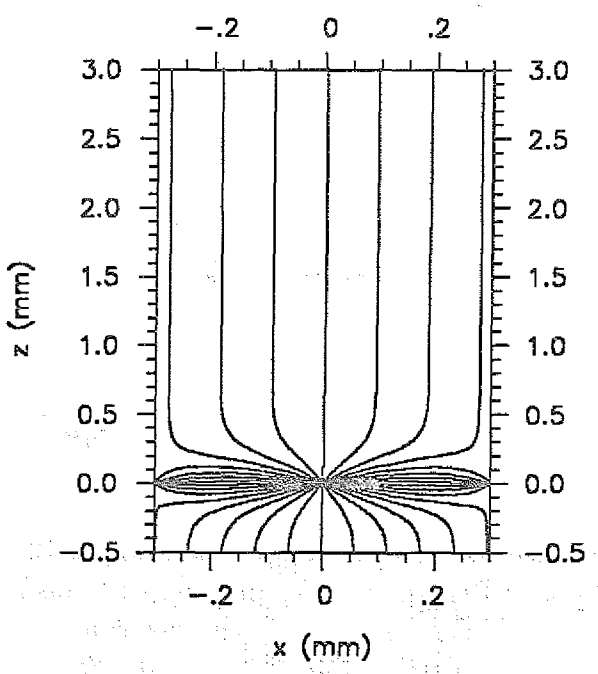


Abbildung 2.12: *Feldverlauf in einer asymmetrischen Driftzelle.*

Kapitel 3

Meßaufbau

3.1 Mechanischer Aufbau der Testkammer

Die Geometrie einer Ebene der Testkammer zeigt Abbildung 3.1.

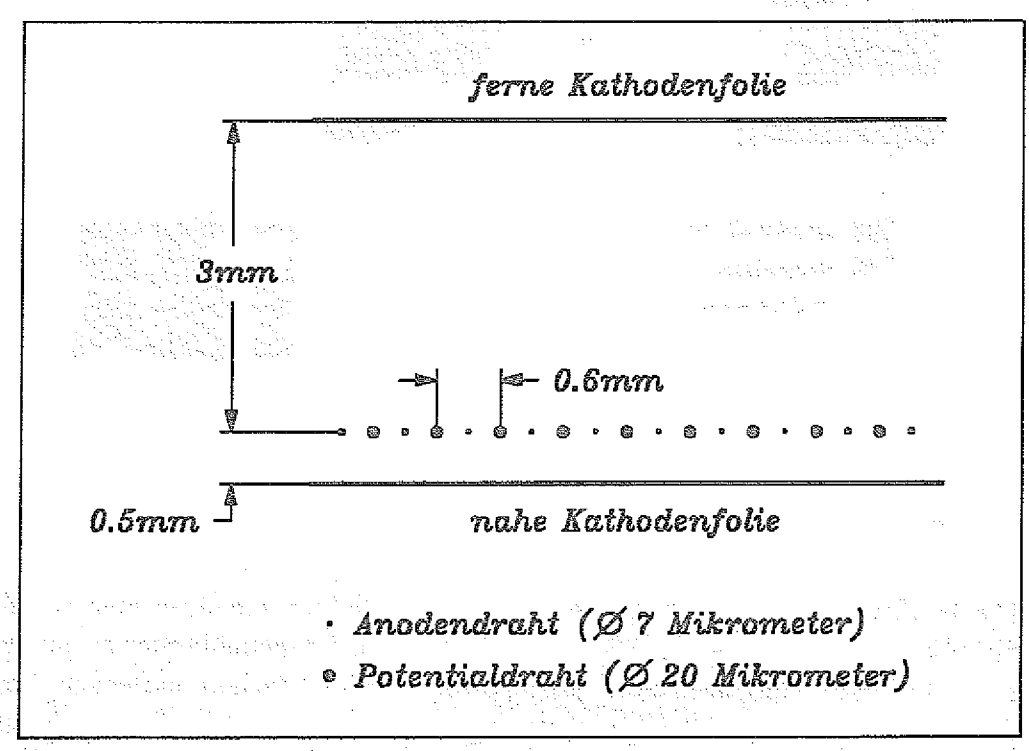


Abbildung 3.1: *Die geometrischen Parameter einer Ebene der Testkammer.*

Sie wird mechanisch wie folgt realisiert: Die Drahtebenen werden von 2,5 mm

dicken Rahmen aus Dur–Aluminium getragen, die so gefräst sind, daß der Abstand von der Oberkante des Rahmens zur Drahtebene 0,5 mm beträgt. Für die Kathoden werden beidseitig mit Aluminium beschichtete Mylarfolien verwendet, die auf einen 1 mm dicken Rahmen aus *Gema G10* aufgeklebt werden. Dieser Kathodenrahmen wird mit der Folie zur Drahtebene auf einen Drahtrahmen geschraubt. Damit ergibt sich der Abstand zwischen Drahtebene und naher Kathodenfolie zu 0,5 mm. Die rückseitige Beschichtung der Folie dient als ferne Kathode für die nächste Drahtebene. Der nächste Drahtrahmen liegt bündig auf dem Kathodenrahmen auf, so daß sich der Abstand zwischen der fernen Kathodenfolie und der Drahtebene aus der Dicke des Kathdodenrahmens (1 mm) plus dem Abstand von der Unterkante des Drahtrahmens zur Drahtebene (2 mm) zu 3 mm ergibt. Einen Schnitt durch einen Stack von vier Kammerebenen zeigt Abbildung 3.2.

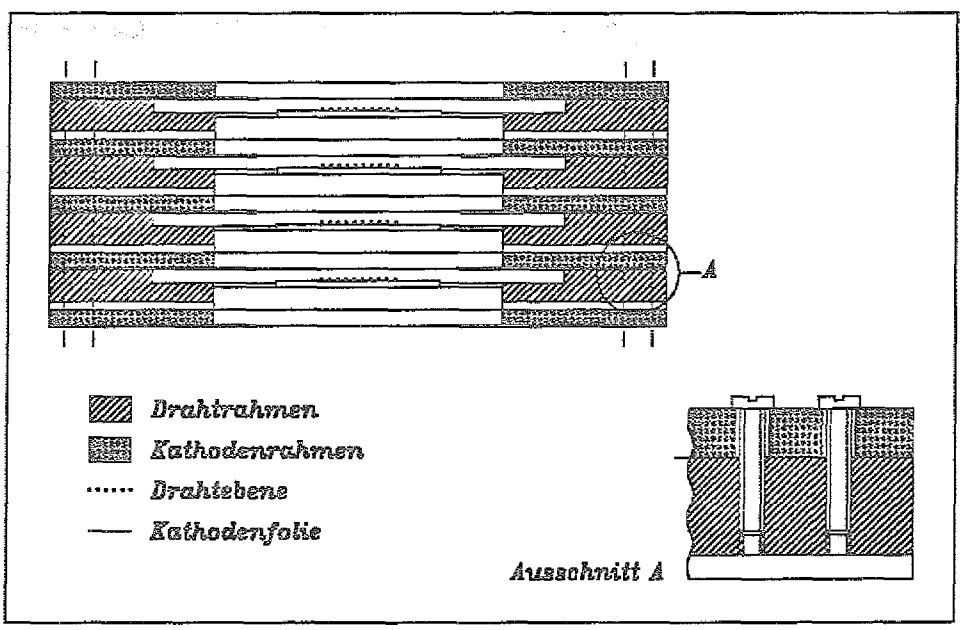


Abbildung 3.2: *Schnitt durch einen Stack aus vier Kammerebenen.*

Die Drahtrahmen werden einzeln in einem gasdichten Gehäuse aus Aluminium befestigt. So ist es möglich, die Drahtebenen gegeneinander zu justieren, und bei Störungen in einer Ebene diese relativ einfach zu entfernen oder zu ersetzen. Zur Signaldurchführung aus dem Gehäuse werden Kaptonfolien mit aufgedruckten Leiterbahnen verwendet, die gasdicht gepresst werden.

In den anschließenden Abschnitten werden die einzelnen Komponenten der Kammer im Detail beschrieben.

3.1.1 Drahtebenen

Wir verwenden goldplattierte Wolfram–Drähte der Firma [LumaLampan]. Die Drahtrahmen sind aus Dur–Aluminium gefräst. Darauf aufgeklebt werden Keramikplättchen, auf denen die Drähte aufgeklebt werden, sowie mit Leiterbahnen bedruckte Kaptonfolien zur Auslese der Drähte (siehe Abbildung 3.3). Dabei werden zur einen Seite die Anoden– zur anderen Seite die Potentialdrähte ausgelesen. Jede Drahtebene besteht aus insgesamt

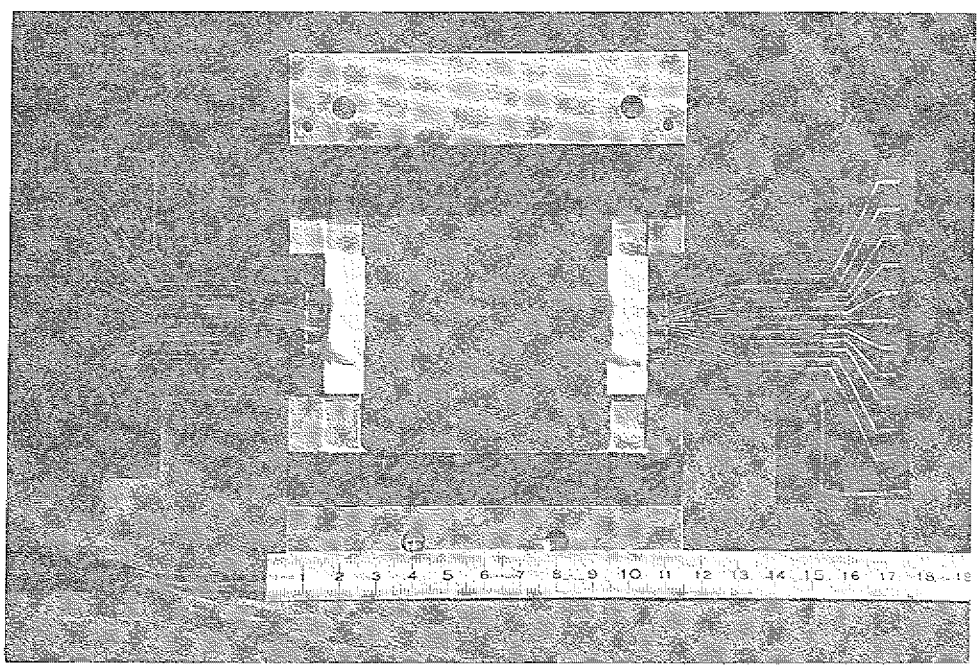


Abbildung 3.3: *Foto eines Drahtrahmens, aufgeklebt sind die Keramikplättchen und die bedruckten Kaptonfolien zur Auslese der Drähte. Die freie Länge der Drähte beträgt 68 mm.*

61 Drähten, von denen dreizehn (sechs Anodendrähte und sieben Potentialdrähte) sechs 'aktive' Zellen aufspannen, während die restlichen Drähte der Feldformung dienen. Die folgende Tabelle enthält eine komplette Auflistung der Drahtkonfiguration:

äußere Randgruppe	3 Anodendrähte 12 μm	3 Potentialdrähte 20 μm
innere Randgruppe	9 Anodendrähte 10 μm	9 Potentialdrähte 20 μm
'aktive' Detektorfläche	6 Anodendrähte 7 μm	7 Potentialdrähte 20 μm
innere Randgruppe	9 Anodendrähte 10 μm	9 Potentialdrähte 20 μm
äußere Randgruppe	3 Anodendrähte 12 μm	3 Potentialdrähte 20 μm

Zur Herstellung der Drahtebenen dient die in Abbildung 3.4 gezeigte Draht-

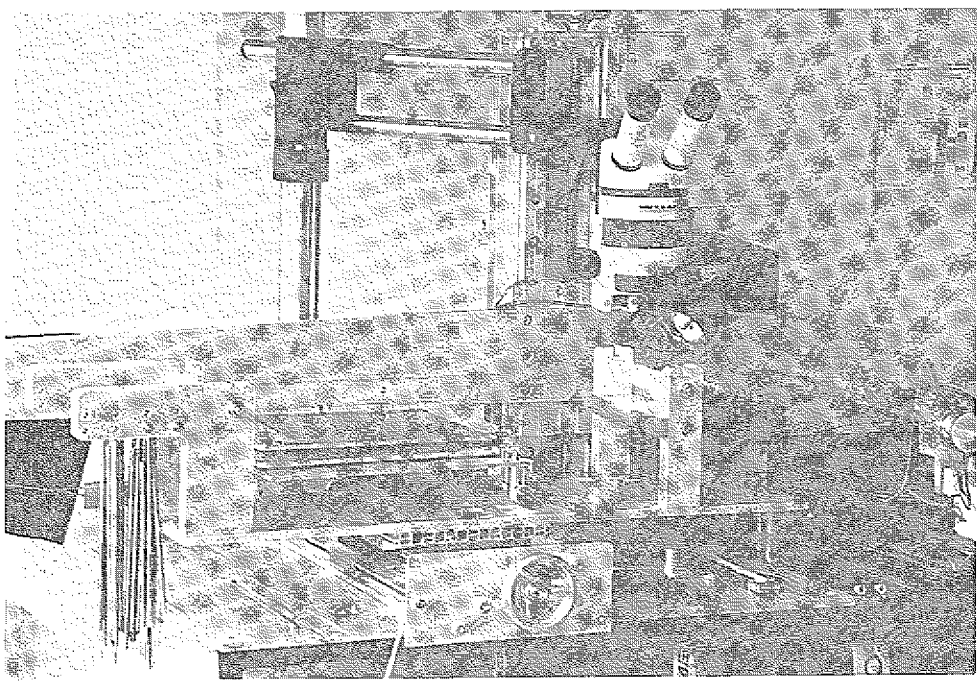


Abbildung 3.4: *Foto der Drahtlegemaschine. Die beiden Plexiglas-Gewindestangen definieren die Drahtabstände. Rechts sind die Drähte festgelötet, die links angelöteten Messing-Gewichte bestimmen die Drahtspannung.*

legemaschine. Die Drähte werden über zwei parallele Gewindestangen gelegt. Damit ist der Drahtabstand durch den Nutabstand (die Steigung der Gewindestangen) bestimmt. Wir haben diese Steigung zu $(298.0 \pm 1.9)\,\mu m$ gemessen. Die Drahtspannung wird durch Gewichte definiert, die an die Drahtenden gelötet werden. Wir haben diese Gewichte für jede Drahtdicke

so gewählt, daß die Drahtspannung etwa 50–60 % der vom Hersteller angegebenen Zugfestigkeit beträgt.

Drahtdicke $[\mu m]$	Querschnitt $[10^{-5}\,mm^2]$	Zugfestigkeit $[kp/mm^2]$	$[g]$	Gewicht $[g]$
7	3.85	350	13	7.5
10	7.85	350	27	15.
12	11.31	350	40	22.5
20	31.41	290	90	48.

Zunächst werden alle Drähte für eine Ebene gelegt. Dann wird der Rahmen mit einem Lineartisch von unten gegen die Drähte gefahren, die Drähte auf den Keramikplättchen festgeklebt (als Kleber verwenden wir hier *TorrSeal* der Firma [Varian]), an einem Ende direkt hinter dem Kleber abgeschnitten und an dem Ende, an dem sie ausgelesen werden, auf die Leiterbahnen gelötet (siehe Abbildung 3.5).

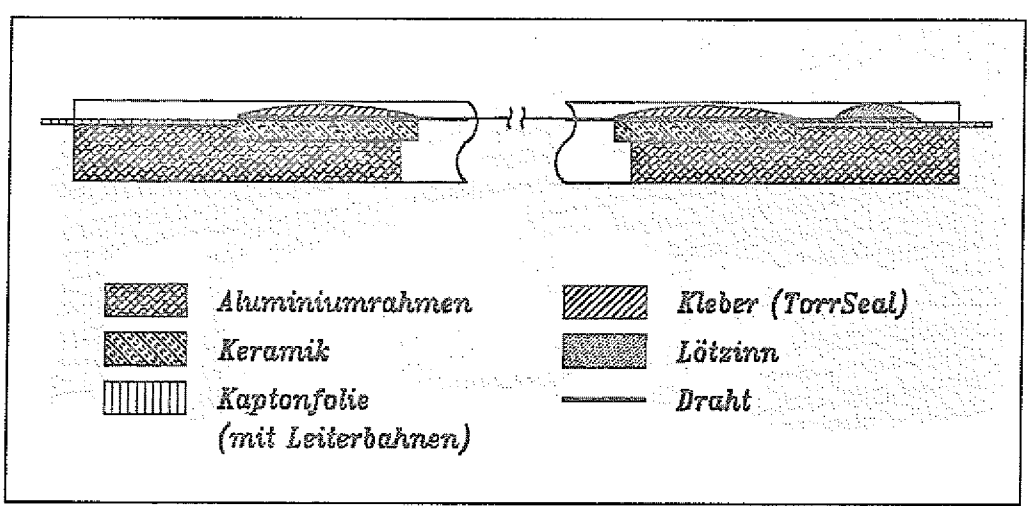


Abbildung 3.5: *Schnitt durch einen Drahtrahmen, senkrecht zur Drahtebene. Dargestellt ist die Befestigung der Drähte auf den Keramikplättchen und auf den Leiterbahnen.*

3.1.2 Kathodenfolien

Für die Kathodenfolien verwenden wir beidseitig mit 2 μm Aluminium beschichtete 25 μm dicke Mylarfolien.

Um zu hohe Feldstärken im schmalen Teilraum der Kammerebene zu vermeiden, müssen auf den fernen und nahen Kathodenfolien unterschiedliche Potentiale eingestellt werden. Die Potentialdifferenzen zwischen beiden Seiten der Folie erreichen etwa 500 V, die Hochspannungsfestigkeit der Folien wurde jedoch bis 8 kV getestet. Durch Abätzten mit NaOH wird die Aluminiumbeschichtung an störenden Stellen entfernt. Die Folie wird gespannt, auf den Rahmen aus *GEMA G10* geklebt und dieser auf einen Drahtrahmen geschraubt. Die Geometrie der Kathodenrahmen wurde so gewählt, daß sie die Drahtebenen nicht vollständig abdecken und die Drähte von oben sichtbar sind (siehe Abbildung 3.6). Damit ist es möglich, die Drahtebenen nach

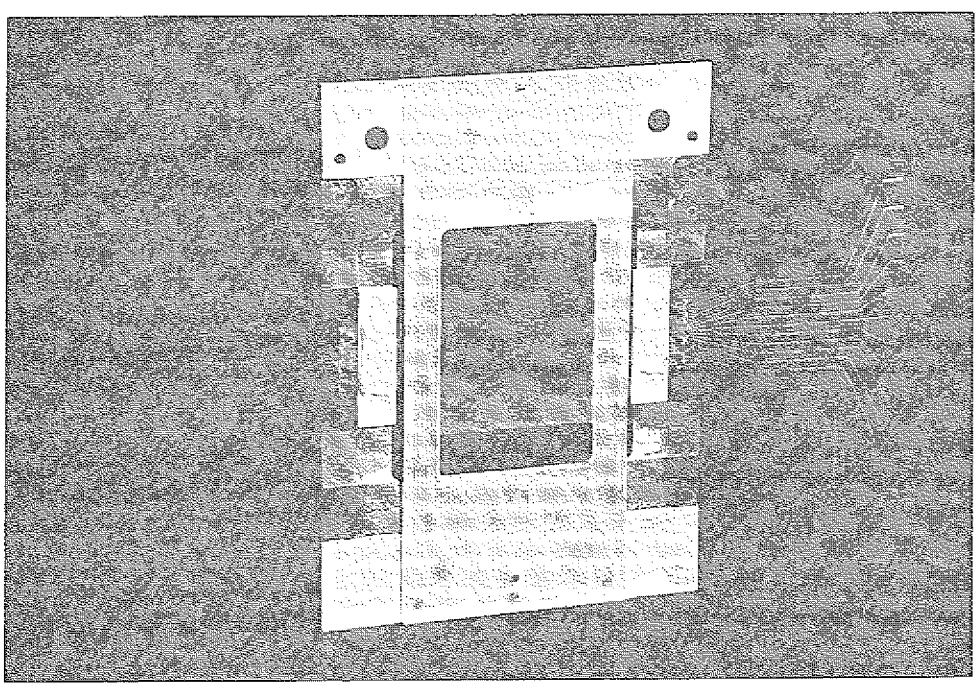


Abbildung 3.6: *Foto eines auf einen Drahtrahmen aufgeschraubten Kathodenrahmens.*

dem Einbau ins Kammergehäuse gegeneinander zu justieren.

Die Hochspannungsanschlüsse werden auf dem aluminiumbeschichteten Mylar geklemmt. Probleme ergeben sich, wenn die Folien über scharfe Kanten gebogen werden, da die Aluminiumbeschichtung an diesen Stellen abgerieben und damit der Kontakt unterbrochen wird.

3.1.3 Kammergehäuse

Das Gehäuse für die Testkammer wurde in der Konstruktionsabteilung des Instituts für Kernphysik entworfen, die auch die in Abbildung 3.7 gezeigte Ansicht anfertigte. In der Mitte des Gehäuses sind vier Kammerebenen angedeutet, links vorn und rechts hinten sieht man die runden Strahlfenster, die Anschlüsse links vorn sind für die Hochspannungsversorgung der Kathodenfolien sowie für das Gassystem. Die jeweils vier Schlitze links hinten und rechts vorn dienen der Durchführung der Kaptonfolien für die Signalauslese und Hochspannungsversorgung der Drähte. Sie sind mit einer Beschichtung aus Silikon ausgekleidet und werden durch Einführen von Keilen gasdicht gepresst. Auf den abstehenden Platinen werden die Vorverstärker befestigt. Der Deckel wird mit einem O–Ring abgedichtet.

Maximal vier Drahtrahmen mit aufgeschraubten Kathodenrahmen werden einzeln befestigt, sodaß sie nach dem Einbau gegeneinander justiert werden können. Dazu blickt man mit einem Mikroskop durch eines der Strahlfenster senkrecht von oben auf die Drähte und verschiebt durch Drehen zweier Schrauben die Rahmen, bis die Drahtebenen zur Deckung kommen.

3.2 Hochspannung

Ein Blockschaltbild der Hochspannungsversorgung für die Testkammer zeigt Abbildung 3.8.

Die Anodendrähte liegen auf Masse, damit keine Hochspannungsentkopplung zu den Vorverstärkern notwendig ist. Entkoppelkondensatoren mit einer Kapazität von 1 nF speichern bei einer anliegenden Spannung von 1 kV eine Ladung von

$$Q = C \cdot U = 1\,\mathrm{nF} \cdot 1\,\mathrm{kV} = 1\,\mu\mathrm{C},$$

die bei einer Entladung über den Draht abfließt, was ausreicht, die 7 μm dünnen Anodendrähte zu zerstören.

Für die Potentialdrähte und die nahen und fernen Kathodenfolien werden drei Hochspannungen benötigt. Wir benutzen zwei Netzgeräte vom Typ N-126 der Firma [CAEN] deren Ausgangsspannung im sogenannten REMOTE–Modus über eine am Eingang V_{SET} anliegende Steuerspannung (bei einer Übersetzung von 1kV/1V) geregelt wird. Eine Schaltung zur Regelung dieser Steuerspannungen wurde in Zusammenarbeit mit der Elektronikwerkstatt des IKP entwickelt. Über zwei 10–Gang–Potentiometer wird zunächst das Verhältnis der Steuerspannungen und dann deren absoluter Wert eingestellt

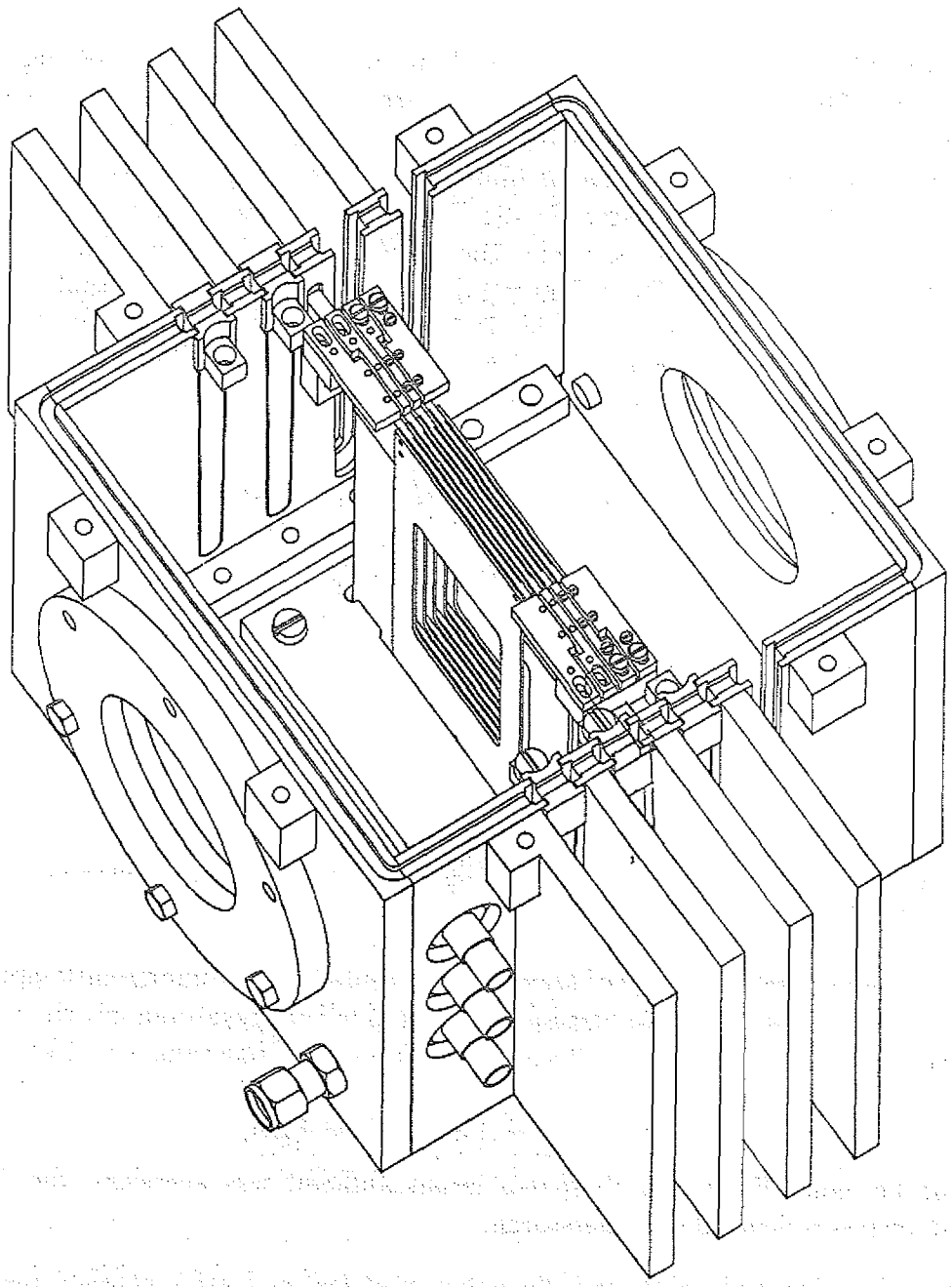


Abbildung 3.7: *Ansicht des offenen Kammergehäuses. Die Kantenlänge der Kammer beträgt 20 cm. Erklärungen zu dieser Zeichnung folgen im Text.*

(siehe Abbildung 3.9), so daß die Hochspannungen in konstantem Verhältnis

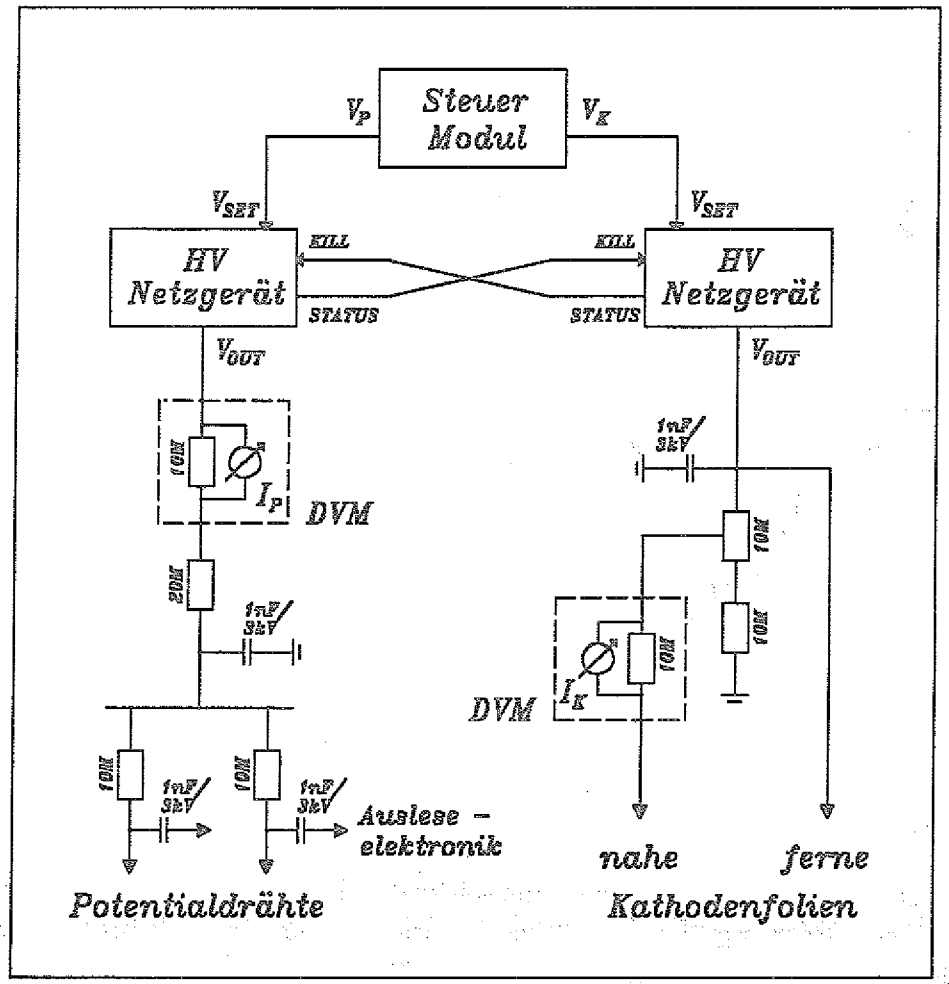


Abbildung 3.8: *Blockschaltbild der Hochspannungsversorgung.*

hochgefahren werden. Beide Größen hängen linear von den Potentiometereinstellungen ab, können also direkt an den Skalentrieben der Potentiometer abgelesen werden.

Die Spannungsteilung zwischen den nahen und den fernen Kathodenfolien wird über eine Potentiometerschaltung realisiert.

Zwei Digitalvoltmeter dienen der Überwachung der gezogenen Ströme. Sie werden als Spannungsmesser benutzt, aber in Reihe geschaltet. Bei einer Spannungsempfindlichkeit von 2mV und einem Innenwiderstand von 10MΩ beträgt dann die Stromempfindlichkeit

$$I = \frac{U}{R} = \frac{2\,\mathrm{mV}}{10\,\mathrm{M\Omega}} = 0{,}2\,\mathrm{nA}.$$

Damit können einzelne Überschläge durch die Schwankungen der Leckströme

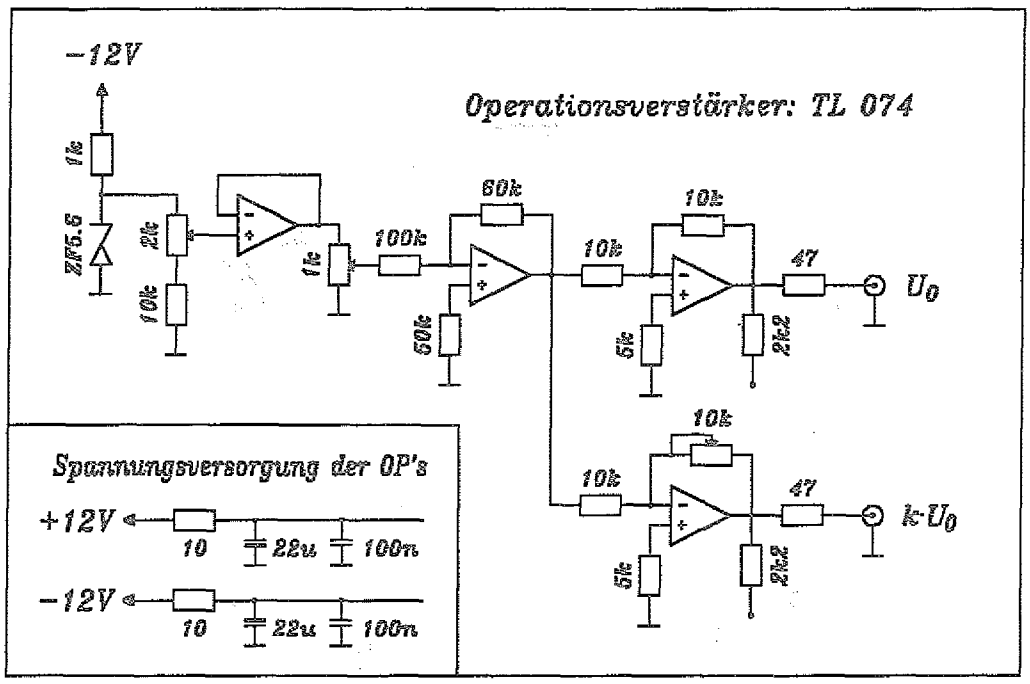


Abbildung 3.9: *Schaltung zur Steuerung zweier Hochspannungsnetzgeräte.*

erkannt werden.

Bei größeren Überschlägen soll eine sofortige Abschaltung der Hochspannung erfolgen. Dazu werden die Netzgeräte im sogenannten *Kill*–Modus betrieben: Übersteigt der gezogene Strom einen eingestellten Maximalwert, so wird die Hochspannung abgeschaltet und ein *Status*–Ausgang gesetzt. Außerdem verfügen die Netzgerte über einen sogenannten *Kill*–Eingang, der, wenn er gesetzt wird, ebenfalls zu einer Abschaltung führt. Die *Status*– und *Kill*–Anschlüsse der beiden Netzgeräte werden über Kreuz verschaltet, so daß bei einem Überschlag beide Netzgeräte abgeschaltet werden.

Die Hochspannungsanschlüsse für die Potentialdrähte sind außerhalb des Kammergehäuses auf die flexiblen Leiterbahnen gelötet, die Zuleitungen für die Kathodenfolien werden mit gasdichten SHV–Steckern in das Gehäuse geführt und direkt auf dem Kathodenrahmen geklemmt.

3.3 Gassystem

Uns steht ein geschlossenes Gassystem zur Verfügung, das auch für aggressive Gase (wie Ammoniak) geeignet ist. Als Besonderheit verfügt es über ein

eingebautes Proportionalzählrohr zur Online–Überwachung der Gasqualität. Da wir die Kammer bei dieser Testmessung ausschließlich mit Argon/Methan betrieben haben, hat das Gassystem nur eine untergeordnete Rolle gespielt und soll deshalb hier nicht näher beschrieben werden[1].

Zur Beimischung eines definierten Anteils Alkohol wird das Gas über zwei mit Äthanol gefüllte Blubberflaschen geleitet. Diese befinden sich in einem Temperierbad, mit dem die Temperatur auf etwa ±1° C genau geregelt werden kann. Über die Temperatur und damit über den Dampfdruck wird die Beimengung des Alkohols gesteuert. Sinnvoll erscheint eine Beimengung von etwa 5% Äthanol, dies entspricht bei Normaldruck einem Dampfdruck des Äthanols von etwa 50 mbar und einer Temperatur von etwa 13° C. Um eine Kondensation des Alkohols in der Kammer zu vermeiden, wird das Gehäuse mit einem elektrischen Heizelement auf einer Temperatur von etwa 25° C gehalten.

3.4 Ausleseelektronik

Abbildung 3.10 zeigt ein Blockschaltbild der Ausleseelektronik. Im Ge-

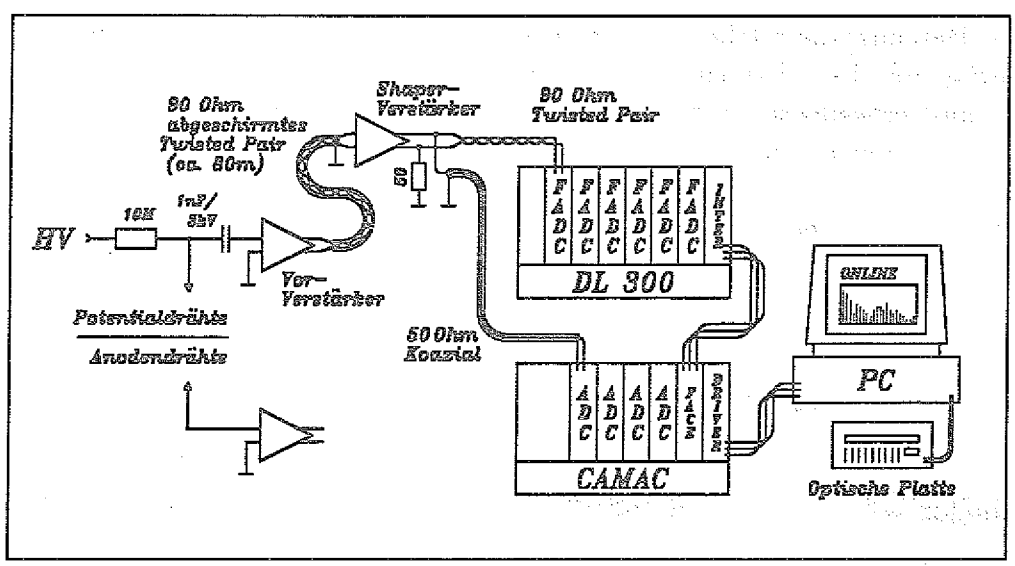


Abbildung 3.10: *Blockschaltbild der Ausleseelektronik.*

gensatz zu früheren Messungen, bei denen die Potentialdrahtsignale paarweise auf Differenzverstärker gegeben wurden, werden die Potentialdrähte

[1] Eine ausführliche Beschreibung dieses Gassystems findet sich in der Diplomarbeit von J.Geisbüsch [GEI88].

einzeln ausgelesen und die Bildung der Differenzsignale digital in der Offline–Auswertung vorgenommen.

Die Analogelektronik (Vorverstärker und Shaper–Verstärker) wurde von Walenta, Gebauer und Junge an der Gesamthochschule Siegen entwickelt und gefertigt. Wir haben sie in unserem Labor getestet, die Ergebnisse dieser Messungen werden in den anschließenden Abschnitten vorgestellt. Die Vorverstärker–Platinen sind direkt am Kammergehäuse befestigt, die Shaper–Verstärker haben wir in NIM–Module eingebaut, sie befinden sich in der Meßhütte. Zur Überbrückung der Distanz zwischen Meßplatz und Meßhütte und zur Verzögerung der Analogsignale[2] werden etwa 80 m lange Twisted–Pair–Kabel verwendet. Ein Kabel besteht aus 17 Paaren, die gemeinsam nach außen abgeschirmt sind. Wir haben auf der Länge von 80 m ein Übersprechen von etwa 10 % zwischen den Signalleitungen beobachtet, deshalb werden für spätere Messungen diese Kabel durch einzeln abgeschirmte Twisted–Pair–Kabel ersetzt werden.

Die gleichzeitige Auslese der Signale mit ADCs und FADCs ermöglicht in der Offline–Auswertung einen direkten Vergleich der Daten. Wir verwenden ADCs vom Typ 4300B FERA der Firma [LeCroy] und FADCs vom Typ DL 311 der Firma [Struck].

Die Datennahme erfolgt mit einem PC über CAMAC. Als Massenspeicher verfügt der PC über ein Laufwerk für Optische Platten der Firma Ricoh. Die auswechselbaren Optischen Platten haben eine Kapazität von 800 Mbyte, sind aber nur einmal beschreibbar.

Das Datennahmeprogramm ONLINE wurde, aufbauend auf Entwicklungen an der Gesamthochschule Siegen, in Jülich geschrieben.

3.4.1 Vorverstärker

Abbildung 3.11 zeigt die Schaltpläne der Vorverstärker für die Anodendrähte (negative Polarität) und die Potentialdrähte (positive Polarität).

Die Vorverstärker sind aus SMD–Bauteilen in Dickfilm–Technik gefertigt. Ein $1 \cdot 2\frac{2}{3}\text{inch}^2$ großes Keramik–Substrat trägt jeweils sechs Kanäle.

[2] Eine Verzögerung der Analogsignale ist wegen der Auslese mit ADCs notwendig. Für die FADC–Auslese erübrigt sie sich, wenn man die FADCs im *Common Stop*–Mode betreibt.

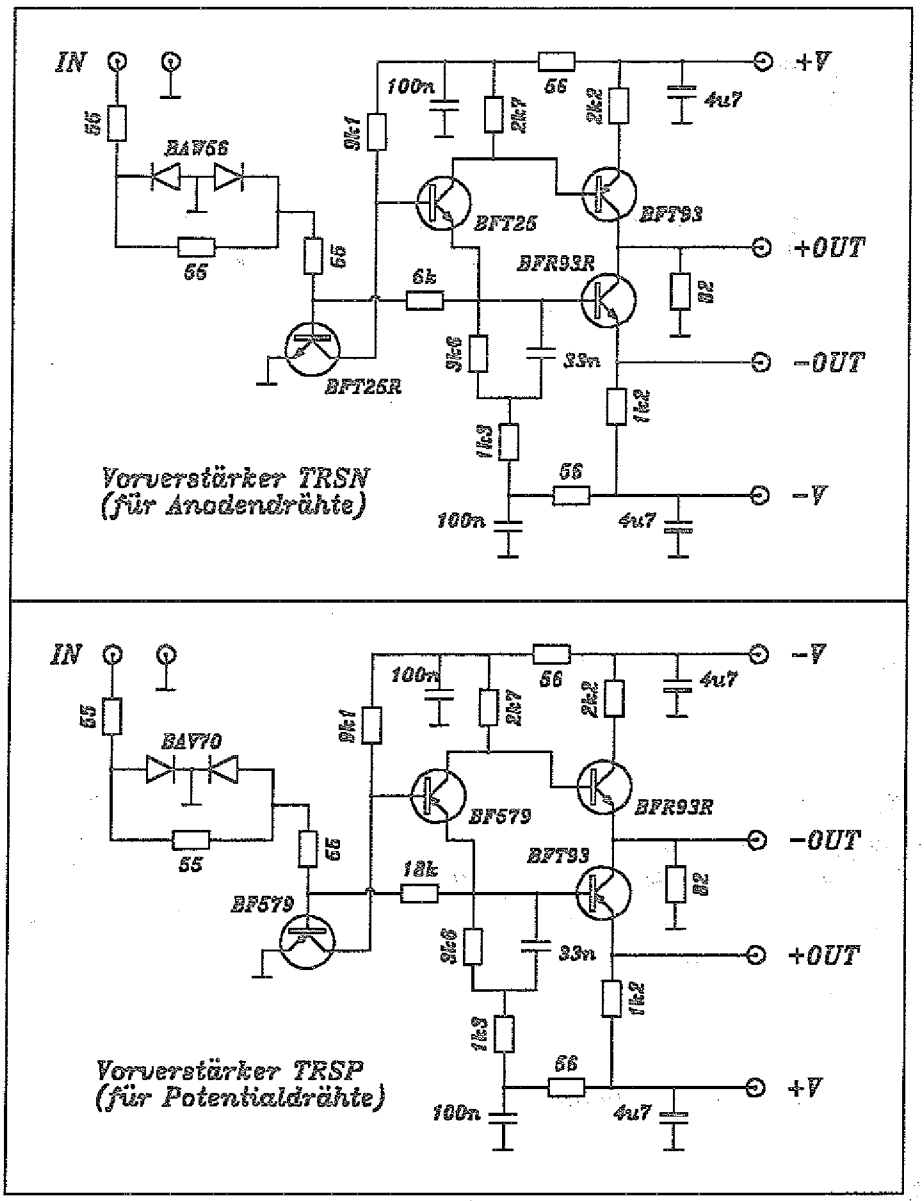


Abbildung 3.11: *Schaltpläne der Vorverstärker für die Anoden- und für die Potentialdrähte*

Testaufbau

Den Meßaufbau für den Test der Vorverstärker zeigt Abbildung 3.12. Jede Platine verfügt über einen Testeingang zur Überprüfung der Funktion und zur Eichung der Verstärker. Testpulse mit möglichst kurzer Anstiegszeit und einer Amplitude U_{test} werden über eine Kapazität C_k eingekoppelt. Die Koppelkapazität ist so klein (Größenordnung 1 pF), daß sie vollständig umgeladen

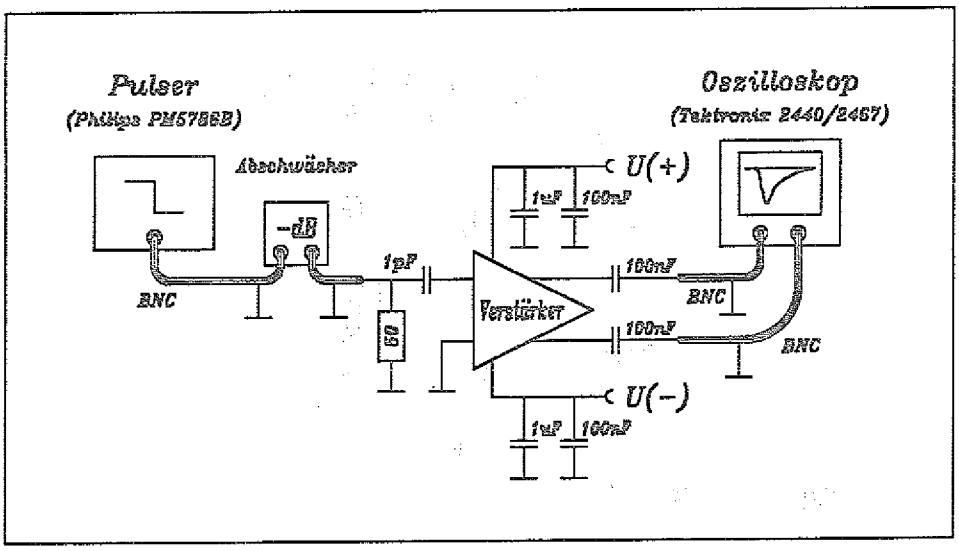


Abbildung 3.12: *Aufbau für den Test der Vorverstärker.*

und damit eine definierte Ladung $Q = C_k \cdot U_{test}$ auf die Eingänge der Vorverstärker gegeben wird. Realisiert ist diese Koppelkapazität durch zwei sich kreuzende Leiterbahnen. Der Testeingang ist intern mit 56 Ω abgeschlossen.

Die Anstiegszeit der Testpulse beträgt 2 ns (minimale Anstiegszeit für den verwendeten Pulser (Philips PM5786B)), ihre Amplitude wird durch vorgeschaltete passive Abschwächer variiert, um eventuelle Korrelationen zwischen Amplitude und Pulsform des Pulsers auszuschließen. Als Meßinstrumente stehen uns ein 350 MHz Analogoszilloskop (Tektronix 2467) und ein 500 MSample/s Digitaloszilloskop (Tektronix 2440) zur Verfügung.

Anstiegs- und Abklingzeit

Abbildung 3.13 zeigt das Ausgangssignal eines Vorverstärkers vom Typ TRSN (negative Polarität). Die Anstiegs- und Abklingzeiten (10% bis 90% der Amplitude) sind:

$$\tau_{rise} = 2{,}5\,\text{ns}$$
$$\tau_{fall} = 6\,\text{ns}.$$

Versorgungsspannung

Die Untersuchung der Verstärkung in Abhängigkeit von der Versorgungsspannung ergibt optimale Ergebnisse bei asymmetrischen Versorgungsspan-

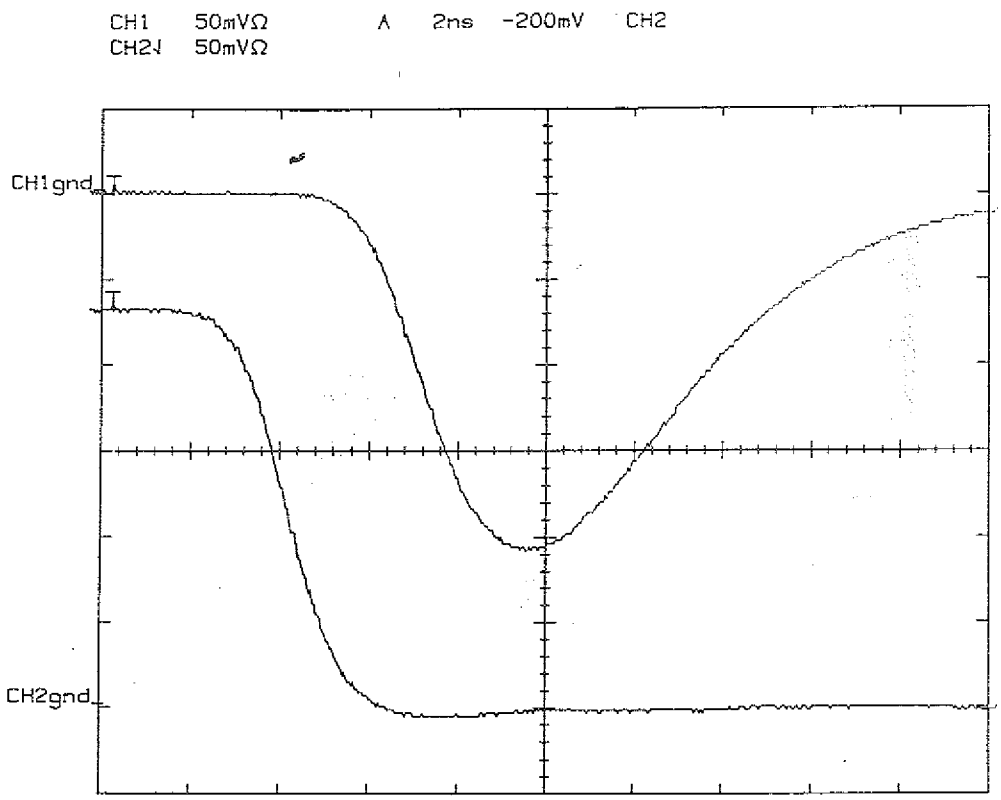


Abbildung 3.13: *Ausgangssignal des Vorverstärkers (obere Kurve) bei einer Anstiegszeit des Pulsersignals von 2 ns (untere Kurve).*

nungen von (+8V/-4V) für die Verstärker Typ TRSN und (+4V/-8V) für den Typ TRSP.

Eichung der Einkoppelkapazität

Zur Messung der Ladungsverstärkung der Vorverstärker muß zunächst die Koppelkapazität C_k des Testeingangs bestimmt werden. Dazu wird mit geeichten externen Kondensatoren eine Eichgerade (Amplitude des Vorverstärkersignals gegen Kapazität bei konstanter Amplitude und Anstiegszeit des Pulsersignals) aufgenommen und dann aus der über den Testeingang gemessenen Amplitude die Kapazität C_k bestimmt. Abbildung 3.14 zeigt eine solche Eichgerade. Für diesen Kanal ist $C_k = 0{,}9$ pF. Für die verschiedenen Kanäle schwanken die Werte zwischen 0,8 pF und 1,2 pF.

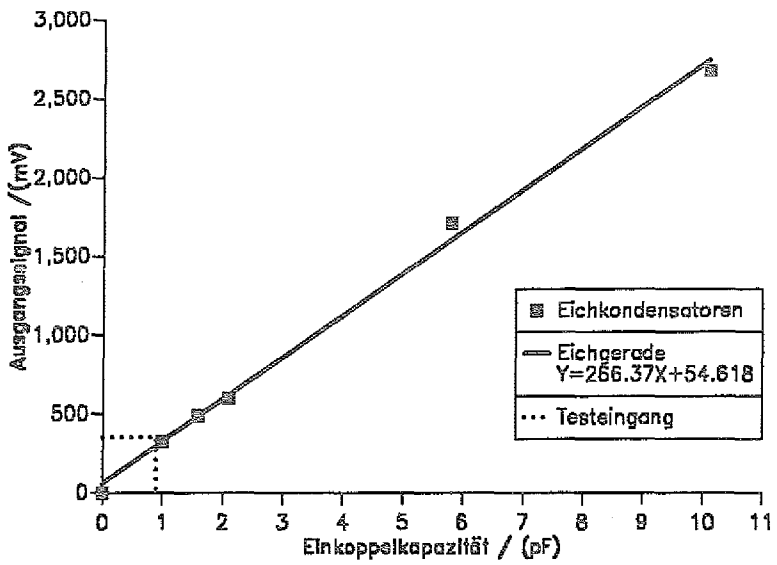


Abbildung 3.14: *Eichung der Koppelkapazität des Testeingangs.*

Bestimmung der Ladungsverstärkung

Ist die Einkoppelkapazität bekannt, so kann die Ladungsverstärkung der Vorverstärker bestimmt werden. Dazu trägt man das Ausgangssignal des Vorverstärkers gegen die eingekoppelte Ladung auf. Die Steigung dieser Eichgeraden ergibt die Ladungsverstärkung. In Abbildung 3.15 ist eine solche Eichkurve für einen Vorverstärker vom Typ TRSN dargestellt. Die Ausgangssignale des invertierenden und des nicht invertierenden Ausgangs wurden einzeln gemessen und dann aufsummiert.

Für die Ladungsverstärkung ergeben sich Werte um $750\,\mathrm{mV}/10^6 e^-$.

Rauschen

Das Rauschen der Vorverstärker wird bei offenem, nur mit der Kapazität C_k des Testeingangs belastetem, Eingang gemessen. Dazu wird das Rauschsignal 10-fach verstärkt (Ortec AN302) und auf das Analogoszilloskop gegeben. Durch Einstellen der minimalen und der maximalen Triggerschwelle wird die Fußbreite des Rauschens bestimmt. Das Eigenrauschen des Ortec–Verstärkers wird quadratisch abgezogen. Die für den invertierenden und den

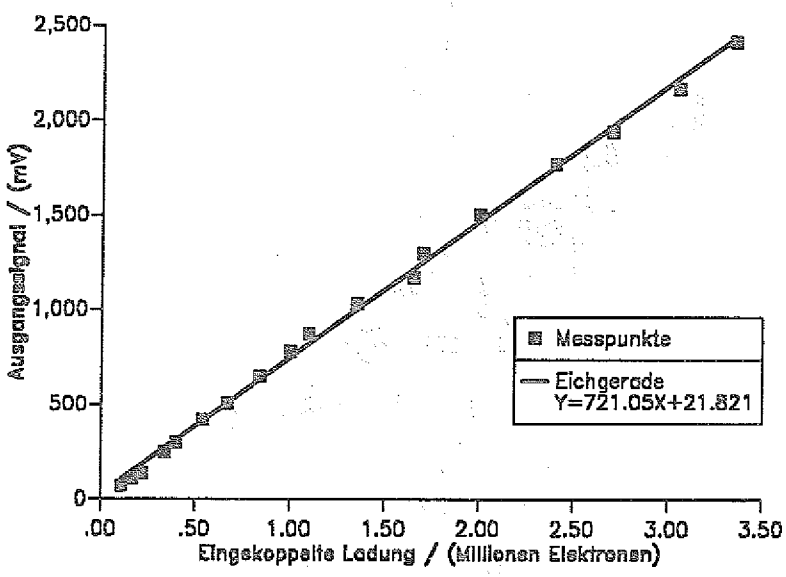


Abbildung 3.15: *Eichkurve zur Bestimmung der Ladungsverstärkung für einen Vorverstärker vom Typ* TRSN.

nicht invertierenden Ausgang bestimmten Rauschamplituden werden quadratisch aufsummiert und mit der gemessenen Ladungsverstärkung eine equivalente Rauschladung am Eingang des Vorverstärkers berechnet. Nimmt man an, daß die Fußbreite vier Standardabweichungen entspricht, so erhält man $\sigma_{Rausch} \approx 1500e^-$.

3.4.2 Signalformende Verstärker

Das Schaltbild der signalformenden Verstärker zeigt Abbildung 3.16. Es sind Differenzverstärker mit symmetrischen Eingängen, somit werden in die zuführenden Twisted–Pair–Kabel eingestreute Störsignale eliminiert. Die Signalformung (Verkürzung des Signals) erfolgt durch 'Pole–Zero'–Kompensation. Die Zeitkonstante der 'Pole–Zero'–Stufe wird über das Potentiometer P_1 eingestellt, über das Potentiometer P_2 kann die Verstärkung etwa um einen Faktor zehn variiert werden.

Die Shaper sind ebenfalls in Dickfilm–Technik aus SMD–Bauteilen gefertigt, ein $2 \cdot 2\,\text{inch}^2$ großes Keramik–Substrat trägt zwei Kanäle. Jeweils vier Shaper–Platinen (acht Kanäle) haben wir in ein NIM–Modul eingebaut. Die

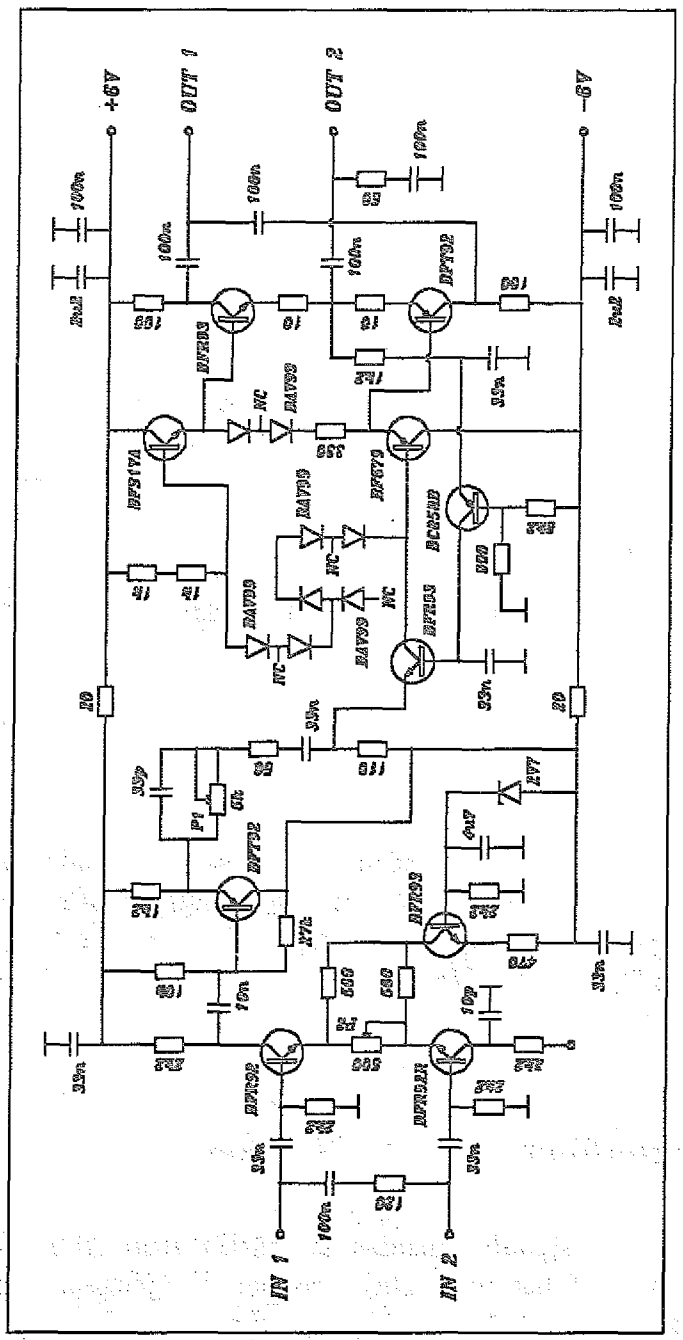


Abbildung 3.16: *Schaltplan der signalformenden Verstärker*

Versorgungsspannung von +/-6 V wird aus dem NIM–Crate entnommen und zusätzlich geglättet.

Anschluß der FADCs und der ADCs

Das Signal wird am Ausgang der Verstärker geteilt. Über Twisted–Pair–Kabel werden die FADCs angeschlossen. Die ADCs verfügen über asymmetrische Eingänge (Signal gegen Masse mit 50 Ω–Abschluß). Sie werden mit dem negativen Ausgang der Verstärker verbunden, der positive Ausgang wird direkt am Verstärker–Ausgang mit 50 Ω gegen Masse abgeschlossen. Die Verstärkung der Shaper ist auch bei minimaler Einstellung so groß, daß trotz der Signalteilung die Amplitude an den Eingängen der FADCs zu groß ist. Deshalb haben wir in den FADC–Zweig steckbare Abschwächer angebracht, die durch X–Glieder realisiert sind.

Optimierung der Signalformung

Zur Optimierung der Signalformung werden in den Testeingang der Vorverstärker eingekoppelte Pulsersignalen benutzt. Für die Verstärkung werden die kleinsten möglichen Werte gewählt, wobei versucht wurde, für die einzelnen Kanäle Unterschiede in der Verstärkung der Vorverstärker auszugleichen.

Eichung der Auslese–Elektronik

Ebenfalls mit Pulsersignalen wird eine Eichung der aus Vorverstärker, Twisted–Pair–Kabel, signalformendem Verstärker und ADC bestehenden Auslesekette vorgenommen. Abbildung 3.17 zeigt eine Eichgerade für einen Kanal. Aufgetragen ist die im ADC gemessene Kanalzahl gegen die in den Vorverstärkereingang eingekoppelte Ladung. Damit kann aus den mit der Kammer gemessenen Amplituden eine Abschätzung der erreichten Gasverstärkung vorgenommen werden.

Pole–Zero–Kompensation

Das Prinzip der Signalformung durch 'Pole–Zero'–Kompensation soll anhand von Abbildung 3.18 erklärt werden. Für die Ausgangsspannung V_a gilt die Differentialgleichung

$$\frac{1}{k\tau_1}U_a + \frac{dU_a}{dt} = \frac{1}{\tau_1}U_i + \frac{dU_i}{dt}$$

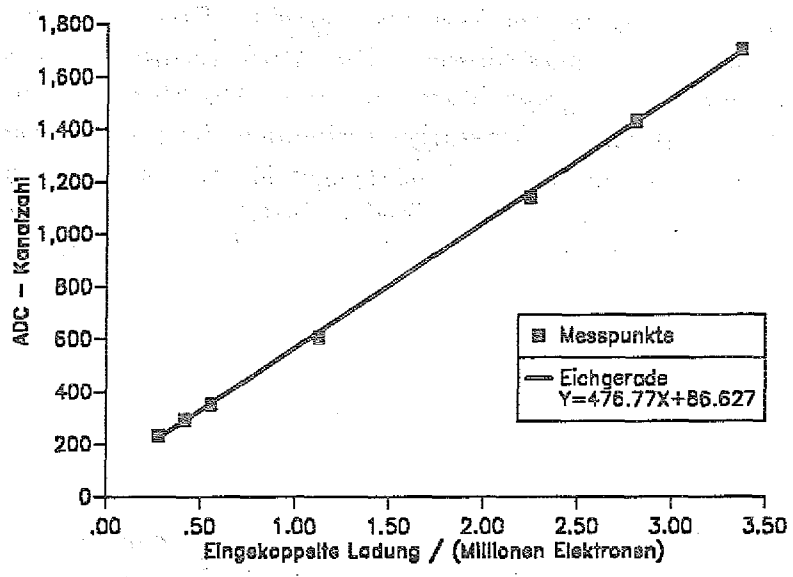


Abbildung 3.17: *Eichung der Ausleseelektronik mit Pulsersignalen.*

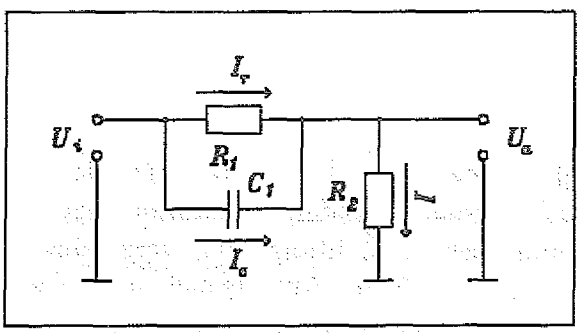


Abbildung 3.18: *Schaltung für 'Pole-Zero'-Kompensation.*

mit $\tau_1 = R_1 C_1$ und $k = R_2/(R_1 + R_2) < 1$. Die Eingangsspannung U_i ist ein exponentiell abfallendes Signal $U_0 \exp(-t/\tau_{in})$. Eingesetzt in die Differentialgleichung ergibt dies

$$\frac{dU_a}{dt} + \frac{1}{k \cdot \tau_1} = \frac{U_0}{\tau_1} \exp\left(\frac{-t}{\tau_{in}}\right) - \frac{U_0}{\tau_{in}} \exp\left(\frac{-t}{\tau_{in}}\right).$$

Man wählt R_1 so, daß $\tau_1 = \tau_{in}$. Damit kompensieren sich die Terme auf der rechten Seite der Gleichung, übrig bleibt eine homogene Differentialgleichung

mit der Lösung

$$U_a = U_0 \exp\left(-\frac{t}{\tau_s}\right) \qquad \text{mit} \quad \tau_s = k \cdot \tau_{in}.$$

Die Signalformung mit 'Pole-Zero'-Kompensation ergibt also ein mit der Zeitkonstante $k\tau_{in}$ exponentiell abfallendes Signal.

3.5 Trigger

Ein Blockschaltbild der Triggerlogik zeigt Abbildung 3.19. Die Signale der

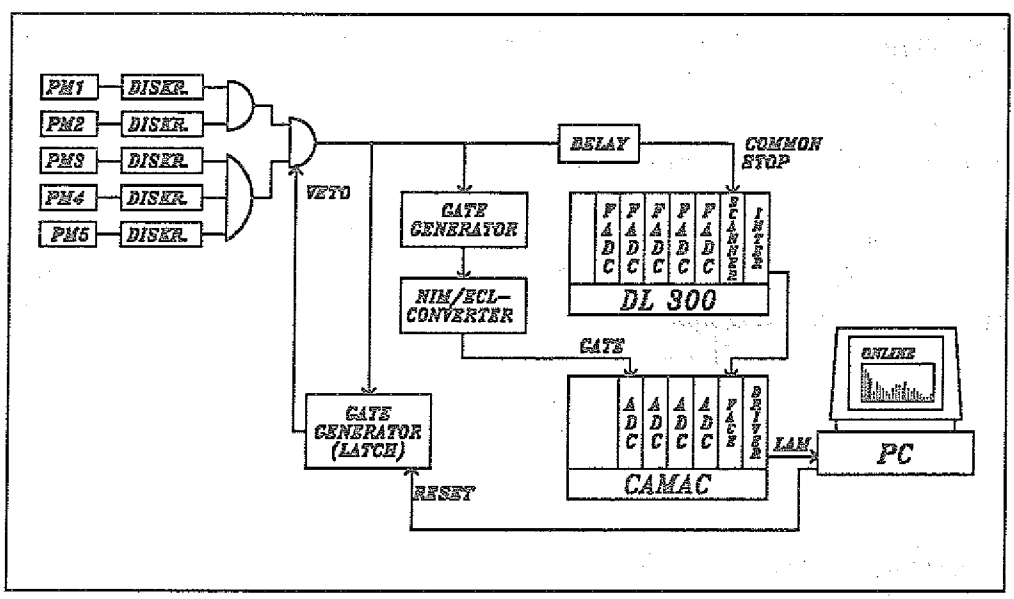


Abbildung 3.19: *Blockschaltbild der Triggerlogik.*

Triggerszintillatoren werden auf Diskriminatoren gegeben und in Koinzidenz geschaltet. Aus dem Trigger wird das *Gate* für die ADCs erzeugt, er ergibt das *Common Stop*-Signal für die FADCs und wird als *Veto* auf die Koinzidenz-Stufe zurückgegeben. Dieses *Veto* wird vom PC zurückgesetzt, wenn das Auslesen der Daten beendet ist. Somit wird ein neuer Trigger erst dann zugelassen, wenn die Auslese-Elektronik wieder frei ist.

Das Länge des ADC-Gates wird so gewählt, daß die Ladungen über das gesamte Signal integriert werden. Die Signallänge für Argon/Methan ist etwa 120 ns, entsprechend haben wir eine Gate-Länge von 150 ns gewählt.

Triggerszintillatoren

Zum Triggern der Kammer wurden zwei $5 \cdot 5 \cdot 50\,mm^3$ große Szintillatoren[3], aus *Bicron BC-404* konstruiert, die mit einer 1/2–Zoll Photomultiplier–Röhre (Hamamatsu R1450) ausgelesen werden. Um falsche Trigger durch in Plexiglaslichtleitern erzeugte Čerenkov–Strahlung zu vermeiden, werden die Szintillatoren über Luftlichtleiter an die Multiplier angekoppelt. Zur Verbesserung der Lichtsammlung sind der Szintillator und der Luftlichtleiter mit einer Hülle aus Aluminiumfolie umgeben, die zur Photomultiplier–Röhre adiabatisch aufgeht.

Effizienz für minimalionisierende Teilchen

Die Effizienz für den Nachweis minimalionisierender Teilchen erhält man durch Vergleich der mittleren Signalamplitude mit der Amplitude des 'single electron'–Peaks. Wir haben diese Werte mit einem Oszilloskop bestimmt und erhalten eine Amplitude von 5 mV für den 'single electron'–Peak und von etwa 250 mV für minimalionisierende Teilchen. Dies entspricht einem Mittelwert von etwa 50 ersten Dynoden–Elektronen für minimalionisierende Teilchen. Die Diskriminatorschwellen wurden auf 50 mV (entsprechend zehn ersten Dynoden–Elektronen) eingestellt. Nimmt man für die Anzahl der ersten Dynoden–Elektronen eine Poisson–Verteilung an, so ergibt sich die Ineffizienz eines Szintillators zu

$$\begin{aligned} P_\mu(n \leq 10) &= \sum_{n=1}^{10} \frac{e^{-\mu} \cdot \mu^n}{n!} \quad ; \quad \mu = 50 \\ &= 6,5 \cdot 10^{-12}. \end{aligned}$$

Die Nachweiswahrscheinlichkeit für minimalionisierende Teilchen ist folglich mit 100% anzusetzen.

3.6 Meßaufbau am Teststrahl

Die Testmessung der Kammer fand im November 1989 mit 3GeV–Elektronen am Teststrahl 21 des DESY in Hamburg statt.

Der Strahl ist begrenzt durch einen 30 cm dicken Kollimator mit einer $7 \cdot 7\,mm^2$ großen Strahlöffnung. Der Teststand befindet sich in einer Entfernung

[3]Beim Testrun standen zusätzlich drei fest installierte Triggerszintillatoren zur Verfügung, die in der Abbildung mit eingezeichnet sind (siehe Kapitel 3.6).

von etwa 7 m vom Kollimator und kann über zwei Lineartische in der Ebene senkrecht zum Strahl verfahren werden.

Am Meßplatz sind Triggerszintillatoren fest installiert, die wir zusätzlich zu unseren Szintillatoren benutzt haben (siehe Abbildung 3.20). Für den

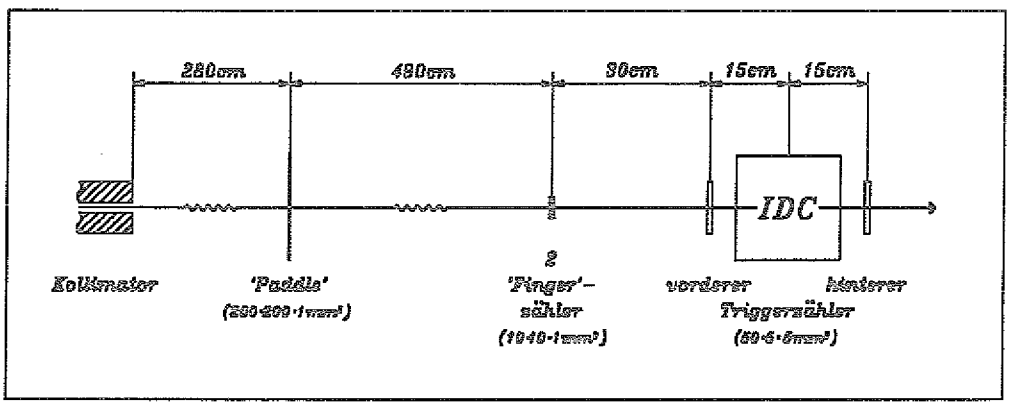


Abbildung 3.20: *Schematische Darstellung der für den Trigger verwendeten Szintillatoren.*

Trigger ergibt sich damit eine fünffach–Koinzidenz.

Justierung des Meßaufbaus

Der Meßaufbau aus Kammer und Triggerszintillatoren muß in Position und Winkel bezüglich des Strahls justiert werden. Zur Ausrichtung parallel zum Strahl wurde ein Laser benutzt, der über der Kollimatoröffnung befestigt und in Strahlrichtung ausgerichtet wurde. Auf den Szintillator–Gehäusen sind Meßspitzen befestigt, die im aufgeweiteten Laserstrahl ein Interferenzmuster erzeugen. Durch Beobachten dieses Interferenzmusters läßt sich der Aufbau auf besser als $0,1°$ genau justieren. Mit den beiden Triggerszintillatoren wird dann ein Profil des Strahlquerschnitts aufgenommen und der Meßaufbau auf die Strahlmitte eingestellt.

Strahldivergenz

Aus dem Querschnitt der Kollimatoröffnung (7 mm) und dem Abstand zur Kammer (7 m) ergibt sich eine geometrische Strahldivergenz von 1 mrad. Die Vielfachstreuung der Elektronen in 7 m Luft und insgesamt 7 mm Szintillator–Material zwischen Kollimatoröffnung und Kammer ergibt

0.6 mrad und 0.5 mrad [PDG88]. Insgesamt beträgt die Strahldivergenz damit etwa 2 mrad. Dies entspricht einer Verschmierung des Ortes von 7 μm auf 3,5 mm (der Dicke einer Kammerebene).

Kapitel 4

Auswertung der Daten

Zur Auswertung der Daten wurde auf dem IBM 3090–Rechner der KFA Jülich das Auswerteprogramm IDEFIX (für IDC Data Evaluation in Fortran under IBM–VM/XA) geschrieben.

In den Abschnitten dieses Kapitels erfolgt eine Abschätzung der von der IDC zu erwartenden Datenmenge, werden die Auswertung der ADC– und der FADC–Daten beschrieben, die Ergebnisse vorgestellt und miteinander verglichen. Diese Ergebnisse beruhen auf einer Messung von 11 000 Ereignissen mit folgenden Parametern:

Strahl	3 GeV Elektronen	
Einfallswinkel	senkrecht zu den Drahtebenen	
Hochspannung	Anodendrähte:	0 V
	Potentialdrähte:	760 V
	nahe Kathodenfolie:	900 V
	ferne Kathodenfolie:	1125 V
Gas	Ar/Me (90/10) mit 5 % Äthanol	
Druck	1100 mbar	

4.1 Abschätzung der Datenmenge

Die Datenmenge ist wesentlichen bestimmt durch die Auslese mit FADCs. Eine Abschätzung der FADC–Wortlänge für ein Ereignis ergibt sich aus den folgenden Parametern:

$$\text{Wortlänge pro Ereignis [byte]} = N_S \cdot N_E \cdot N_D \cdot N_M \cdot 1\,\text{byte}$$

wobei

$$\begin{aligned} N_S &= \text{Anzahl der geladenen Spuren pro Ereignis} \\ N_E &= \text{Anzahl der Kammerebenen} \\ N_D &= \text{Anzahl der pro Spur pro Ebene ausgelesenen Drähte} \\ N_M &= \text{Anzahl der Meßpunkte pro getroffenem Draht} \\ 1\,\text{byte} &= \text{Auflösung des FADC} \end{aligned}$$

Für die Testmessung nehmen diese Parameter die folgenden Werte an: Jedes Ereignis besteht aus genau einer geladenen Spur, also $N_S = 1$. Die Anzahl der Ebenen betrug zwei ($N_E = 2$). Nimmt man an, daß eine Spur im Mittel zwei Zellen pro Ebene berührt, so müssen fünf Drähte ausgelesen werden. Da aber immer eine gerade Anzahl von FADC–Kanälen ausgelesen wird kann[1] folgt $N_D = 6$. Bei einer Driftgeschwindigkeit der Elektronen von $2,5\,cm/\mu s$ beträgt die maximale Driftzeit etwa 120 ns, bei einer Abtastrate der FADCs von 100 MHz ergeben sich damit 12 Meßpunkte für einen FADC–Hit. Zur Untergrund–Bestimmung werden zusätzlich 15 Kanäle vor und nach jedem Hit ausgelesen, zwei Werte kommen für die Markierung der Start–Adresse und der Länge des Hits hinzu, insgesamt ist also $N_M \approx 44$. Damit ist für die Auslese mit FADCs die

$$\text{Wortlänge pro Ereignis} \approx 1 \cdot 2 \cdot 6 \cdot 45 \cdot 1\,\text{byte} = 540\,\text{byte}.$$

Die Auslese eines Ereignisses mit ADCs ergibt für jeden Draht[2] zwei Informationen (die Adresse und den Inhalt des ADC), die in einem 2 byte–Wort zusammengefaßt sind. Damit erhält man bei 24 Drähten zusätzlich etwa 50 byte für jedes Ereignis.

Insgesamt erwarten wir also eine Datenmenge von etwa 590 byte pro Ereignis. In guter Übereinstimmung damit befindet sich die gemessene Zahl von 3.7 Mbyte für 5000 Ereignisse, entsprechend 740 byte pro Ereignis[3].

Die für das Experiment PS 185 geplante Kammer soll aus 10 Ebenen bestehen. Entsprechend der zu messenden Reaktion wird die Anzahl der geladenen Spuren pro Ereignis mindestens vier betragen. Damit erhält man eine

[1] Im FADC–Crate werden die Daten von jeweils zwei Kanälen zusammenkodiert.

[2] Die von uns eingesetzten ADCs ermöglichen einen Untergrund–Abzug und "Zero Suppression". Für die Datenmenge ist eine Online–Reduktion der ADC–Daten jedoch unerheblich, deshalb machen wir von dieser Möglichkeit keinen Gebrauch, lesen bei jedem Ereignis alle Kanäle aus und führen die Datenreduktion in der Offline–Auswertung durch.

[3] Diese Zahl bezieht sich auf die vom PC unformatiert abgespeicherten Datenfiles. Diese können jedoch auf dem Großrechner nicht eingelesen werden, müssen also vor dem Transfer auf einen formatierten File umkopiert werden. Dabei erhöht sich die Datenmenge etwa um einen Faktor drei.

Abschätzung von etwa 15 kbyte für die Wortlänge pro Ereignis. Zum Vergleich: zur Zeit beträgt bei PS 185 die Wortlänge für ein Ereignis 0.3 kbyte.

4.2 Auswertung der ADC–Daten

Die ADC–Auslese ergibt für jedes Ereignis und für jeden Draht einen Wert, der der am Eingang des ADC anliegenden Ladung, integriert über die Dauer des Gates, proportional ist. Abb.4.1 zeigt ein typisches Spektrum der für einen Anoden–Draht gemessenen Werte. Weil wir bei jedem Ereignis alle

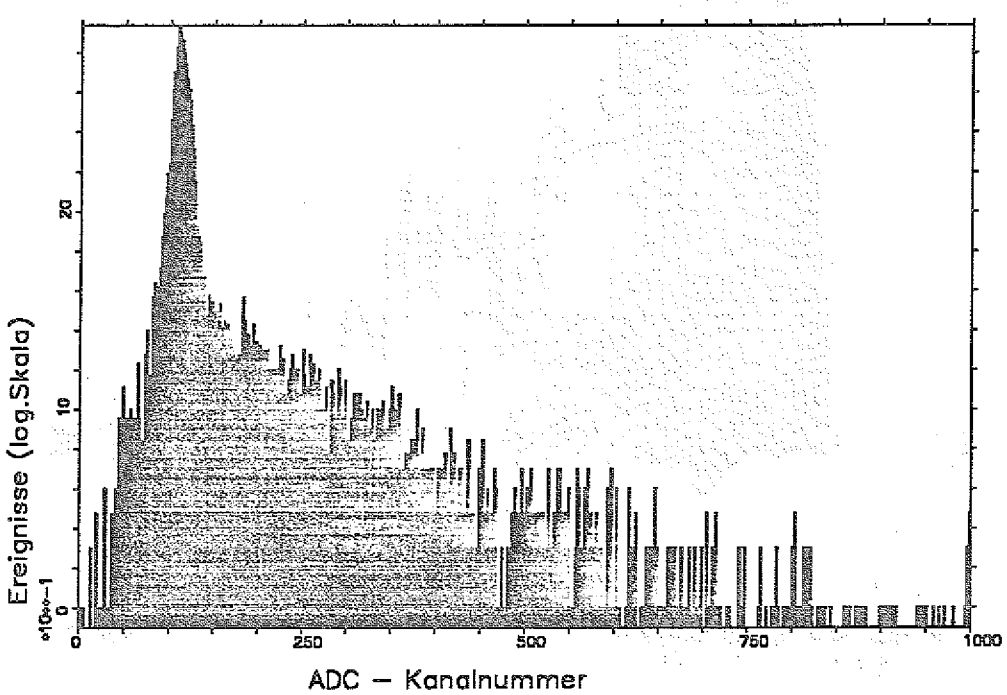


Abbildung 4.1: *Spektrum der* ADC*–Rohdaten für einen Anodendraht. Alle Ereignisse wurden eingetragen.*

ADCs auslesen, die meisten Drähte aber nicht getroffen werden, enthalten die Spektren zum größten Teil Rausch–Signale.

4.2.1 Reduktion der ADC–Daten

Die Auswertung der ADC–Daten beginnt mit der Reduktion der Rauschsignale. Dies geschieht durch das Setzen von Schwellen in der Signalhöhe. Die

Bestimmung dieser Schwellen aus den Rohdaten ist aber schwierig, da die Signale sich nicht deutlich vom Rauschen abheben.

In erster Näherung kann man annehmen, daß jede senkrechte Spur pro Ebene genau eine Zelle trifft, und daß die getroffene Zelle die mit dem höchsten Anoden-Signal in der Ebene ist. Trägt man also für jeden Draht nur die Ereignisse ein, bei denen er das höchste Signal der Ebene gesehen hat, so ergibt sich das in Abbildung 4.2 dargestellte Spektrum. Das Rauschen erscheint

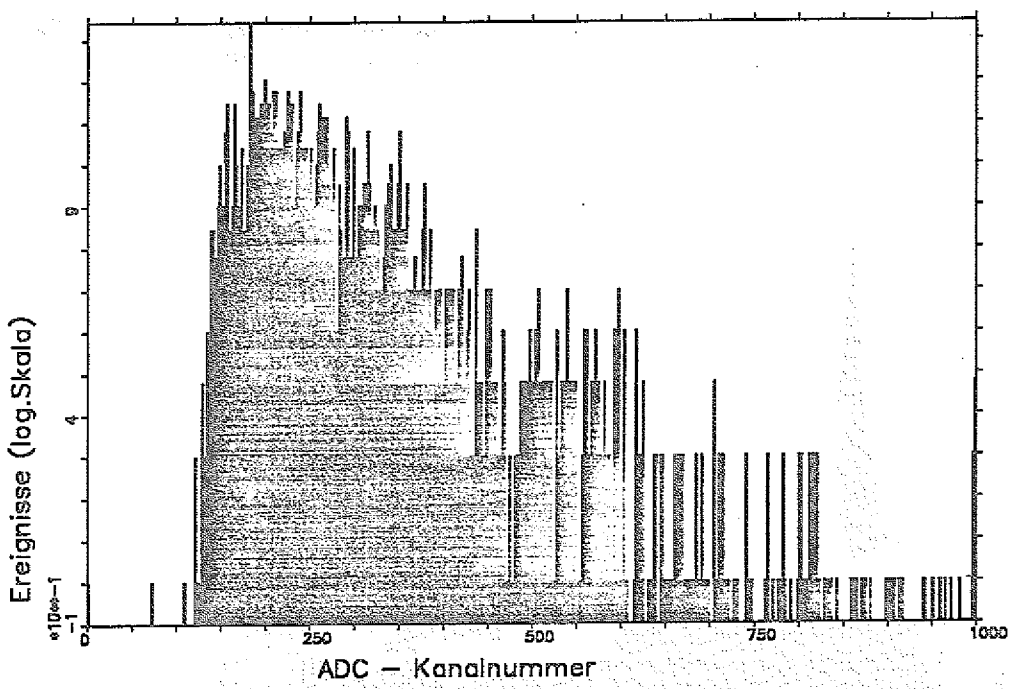


Abbildung 4.2: *Spektrum der* ADC*-Daten für einen Anodendraht. Ein Eintrag erfolgte nur, wenn das Signal dieses Drahtes das höchste in der Ebene war.*

stark unterdrückt, eine Landau-Verteilung ist zu erkennen. Betrachtet man jedoch die nach dieser Methode berechneten Trefferhäufigkeiten für die Zellen einer Ebene (s.Abb. 4.3), so stellt man eine deutliche Abweichung von der erwarteten Gleichverteilung[4] fest. Für diese Abweichungen gibt es mehrere Gründe:

- Unterschiedliche Verstärkungsfaktoren der Auslese-Elektronik für die einzelnen Kanäle. Diese können durch geeignete Normierungen korrigiert werden.

[4]Unter den Annahmen, daß alle Zellen gleiche Flächen belegen, und daß der Primär-Strahl über die gesamte aktive Detektorfläche homogen ist, sollten alle Zellen gleich häufig getroffen werden.

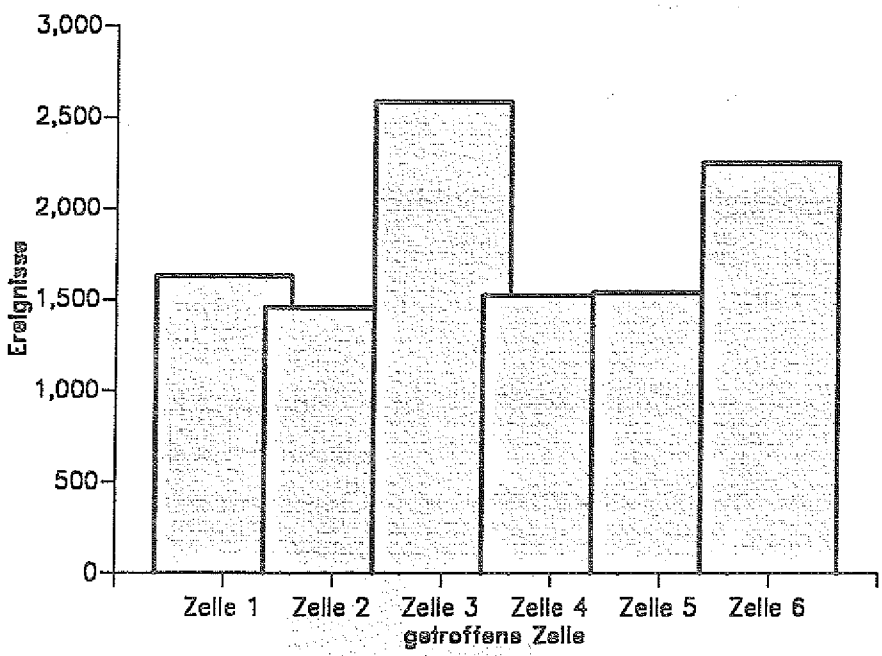


Abbildung 4.3: *Verteilung der berechneten Trefferhäufigkeiten für die Zellen einer Ebene. Ein Treffer wurde in der Zelle angenommen, deren Anoden–Signal das höchste in der Ebene war.*

- Wegen der zu geringen Gasverstärkung liegen die Signale zum Teil noch im Rauschen und können deshalb nicht sauber von diesem getrennt werden.

- Auch senkrechte Spuren können nicht immer einer eindeutig einer Zelle zugeordnet werden.

Diese Methode ist also nicht geeignet, die echten Signale aus den Daten herauszufiltern. Durch den Vergleich der so gewonnenen Spektren mit den Spektren der Rohdaten lassen sich aber sehr gut geeignete Schwellen für die Signal–Rausch–Trennung gewinnen. Diese Schwellen werden in der weiteren Auswertung der ADC–Daten benutzt.

4.2.2 $\frac{\Delta P}{A}$–Signale

Aus den ADC–Daten eines Anoden–Drahtes und der beiden benachbarten Potentialdrähte kann für jedes Event und für jede Zelle das $\frac{\Delta P}{A}$–Signal nach

$$\frac{\Delta P}{A} = \frac{\text{Signal rechter Potentialdraht} - \text{Signal linker Potentialdraht}}{\text{Signal Anodendraht}}$$

berechnet werden. Spektren der $\frac{\Delta P}{A}$–Daten für eine Zelle zeigen die Bilder 4.4 und 4.5. Man kann eine statistische Aussage über die Ortsauflösung

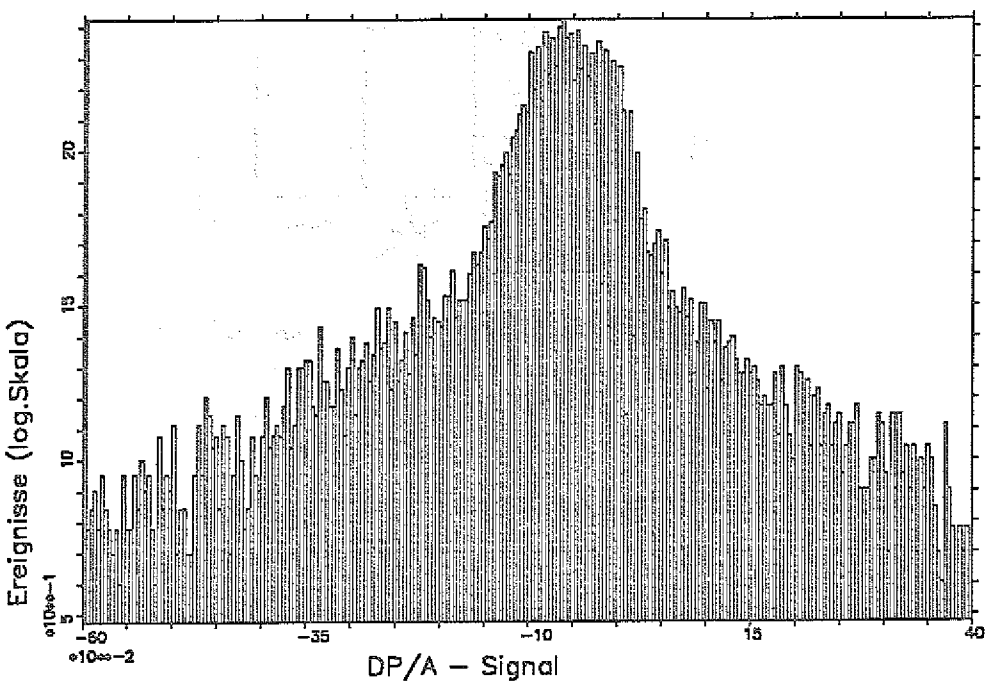


Abbildung 4.4: *Spektrum der aus den* ADC*–Rohdaten berechneten* $\frac{\Delta P}{A}$*–Signale für eine Zelle. Alle Ereignisse wurden eingetragen.*

gewinnen, indem man die für zwei hintereinanderliegende Zellen berechneten $\frac{\Delta P}{A}$–Signale in einem Scatter–Plot gegeneinander aufträgt. Für zwei genau hintereinander liegende Zellen (linkes Bild in Abb.4.7) erwartet man bei senkrechten Spuren einen linearen Zusammenhang der $\frac{\Delta P}{A}$–Signale, also eine (bei endlicher Ortsauflösung verbreiterte) Gerade im Scatter–Plot. Die Breite dieser Verteilung ist ein Maß für die erreichte Ortsauflösung. Einen Scatter–Plot der gemessenen Verteilung für ein Zellen–Paar zeigt Bild 4.6. Eingetragen wurden dabei nur solche Ereignisse, bei denen die Signale aller Drähte der beiden Zellen über den gewählten Schwellen liegen. Eine Verteilung dieser Form ergibt sich dann, wenn die beiden Ebenen nicht genau

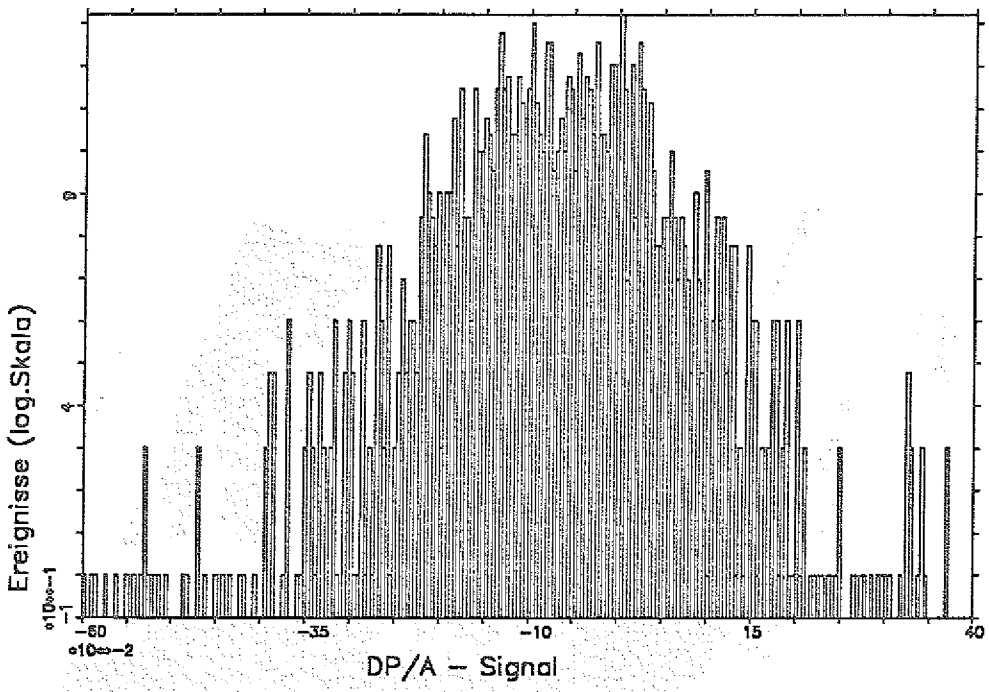


Abbildung 4.5: *Spektrum der aus den* ADC*-Daten berechneten* $\frac{\Delta P}{A}$*-Signale für eine Zelle. Eingetragen wurden nur Ereignisse, bei denen sowohl das Anoden- als auch die Potentialdraht-Signale über den gewählten Schwellen liegen.*

hintereinander liegen, sondern, wie im rechten Bild in Abb.4.7 dargestellt, um eine halbe Zelle gegeneinander verschoben sind. Dies wird deutlich, wenn man die Ereignisse nach den in der Abbildung markierten Bereichen getrennt untersucht. Für den mit 'C' markierten Bereich gilt zum Beispiel:

> In Ebene *'I'* läuft die Spur durch die Zelle ('5'-'6'-'7'), man erhält Gasverstärkung an Draht '6', ein großes influenziertes Signal auf Draht '5' und ein kleineres influenziertes Signal auf Draht '7', nur Rauschen auf den Drähten '4' und '8'. Für das $\frac{\Delta P}{A}$-Signal $(\frac{'5'-'3'}{'4'})$ erwartet man also einen großen positiven Wert. In Ebene *'II'* läuft die Spur durch die Zelle ('3'-'4'-'5'), Gasverstärkung erhält man an Draht '4', ein großes influenziertes Signal auf Draht '5' und ein kleineres influenziertes Signal auf Draht '3', nur Rauschen auf den Drähten '2' und '6'. Für das $\frac{\Delta P}{A}$-Signal $(\frac{'5'-'3'}{'4'})$ erwartet man also einen kleinen positiven Wert.

Entsprechende Überlegungen gelten für die anderen Bereiche. Die erwarteten Ergebnisse sind der folgenden Tabelle zusammengefaßt.

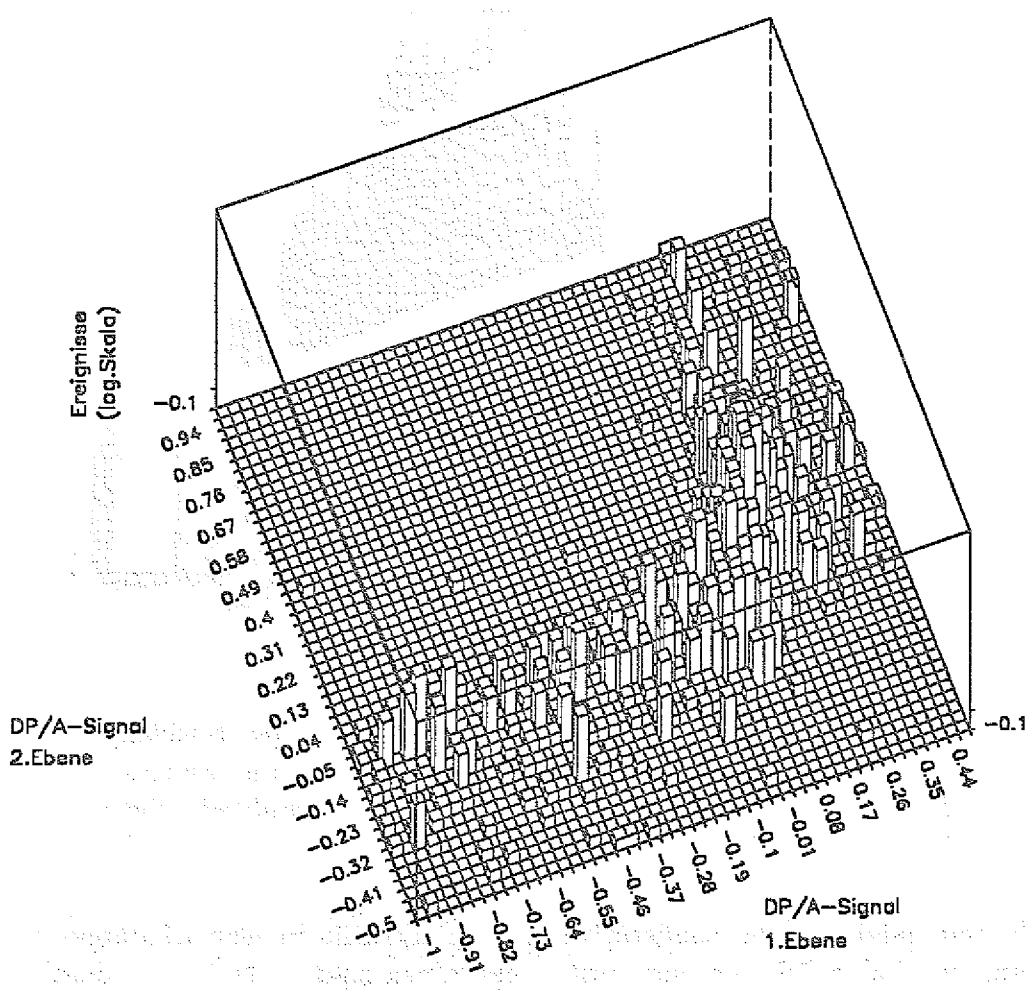


Abbildung 4.6: *Scatter-Plot der für zwei hintereinanderliegende Zellen aus den* ADC*-Daten berechneten* $\frac{\Delta P}{A}$*-Signale.*

Bereich	Ebene	Signal auf Drähten	kein Signal auf Drähten	$\frac{\Delta P}{A}$-Signal für Zelle (3-4-5)
A	I	3,4,5	2,6	klein, negativ
	II	1,2,3	0,4	groß, negativ
B	I	3,4,5	2,6	klein, positiv
	II	3,4,5	2,6	klein, negativ
C	I	5,6,7	4,8	groß, positiv
	II	3,4,5	2,6	klein, positiv

Markiert man die so gefundenen Bereiche in dem Scatter-Plot, so ergibt eine gute Übereinstimmung mit der gemessenen Verteilung.

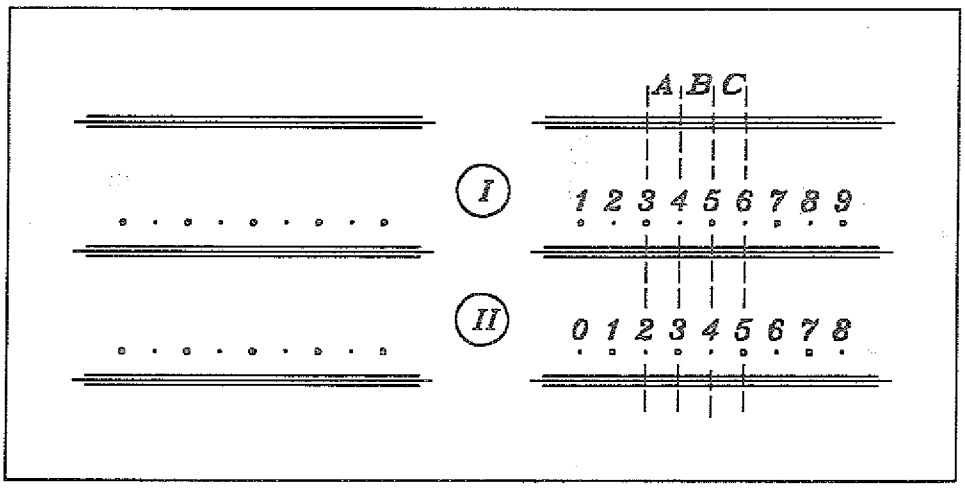


Abbildung 4.7: *Darstellung zweier genau hintereinander liegender Ebenen (links) und zweier um eine halbe Zelle gegeneinader verschobener Ebenen (rechts). Mit Ziffern markiert sind die Drähte (gerade Zahlen entsprechen Anoden-, ungerade Zahlen entsprechen Potentialdrähten). Untersucht wird das* $\frac{\Delta P}{A}$*-Signal der von den Drähten (3-4-5) gebildeten Zelle. Mit Buchstaben markiert sind die verschiedenen Bereiche, auf die im Text Bezug genommen wird.*

4.3 Auswertung der FADC–Daten

Die FADC–Auslese ergibt für jedes Ereignis und für jeden getroffenen Draht eine Reihe von Werten, die die zeitliche Entwicklung des Signals auf diesem Draht beschreiben. In Abbildung 4.8 sind für eine Spur die FADC–Signale der insgesamt sechs in zwei hintereinandergelegenen Zellen getroffenen Drähte aufgetragen. Deutlich zu erkennen ist eine zeitliche Struktur der Peaks, die auf die statistische Verteilung der Ionisationscluster entlang der Spur zurückzuführen ist. In diesen Bildern sind alle für dieses Ereignis ausgelesenen FADC–Daten, einschließlich der zur Baseline–Bestimmung vor und nach dem Hit aufgenommenen Kanäle, eingetragen. Zusätzlich sind die vom Software–Peakfinder berechnete Schwelle und die damit bestimmten Anfangs– und Endpunkte des Hits eingezeichnet (dazu Kapitel 4.3.1).

Probleme gab es mit dem zu kleinen dynamischen Bereich der eingesetzten FADCs. Abb.4.9 zeigt ein Ereignis, bei dem der FADC in Sättigung ist. Beim nächsten Einsatz der Kammer werden wir deshalb *nichtlineare* FADCs (Typ DL311) verwenden, die bei einer Auflösung von 8 bit einen effektiven dynamischen Bereich von 10 bit haben.

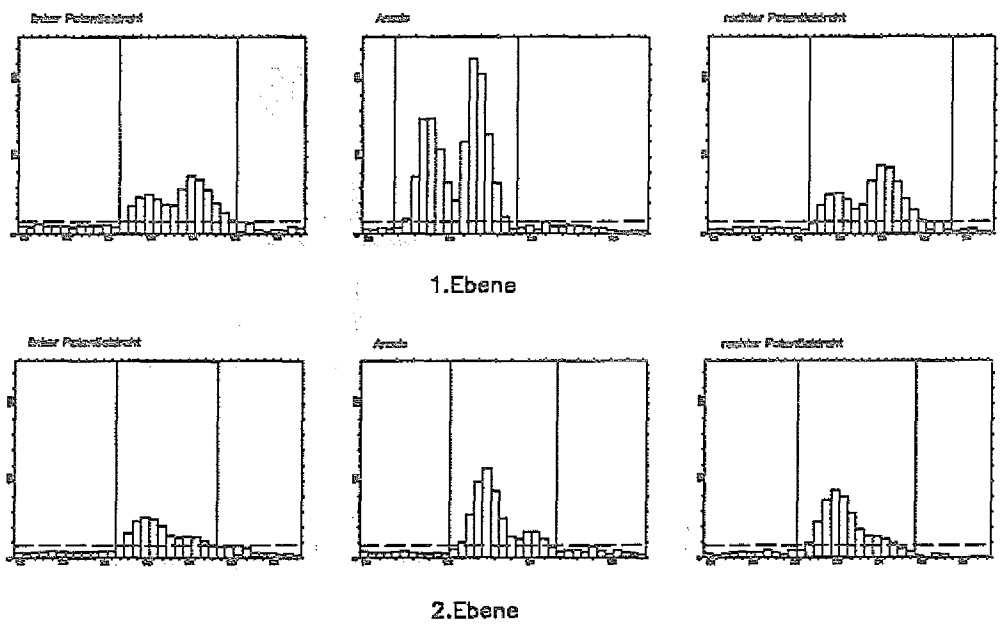


Abbildung 4.8: FADC-*Rohdaten der Signale auf den Drähten zweier hintereinanderliegender Zellen. Ein bin entspricht 10 ns.*

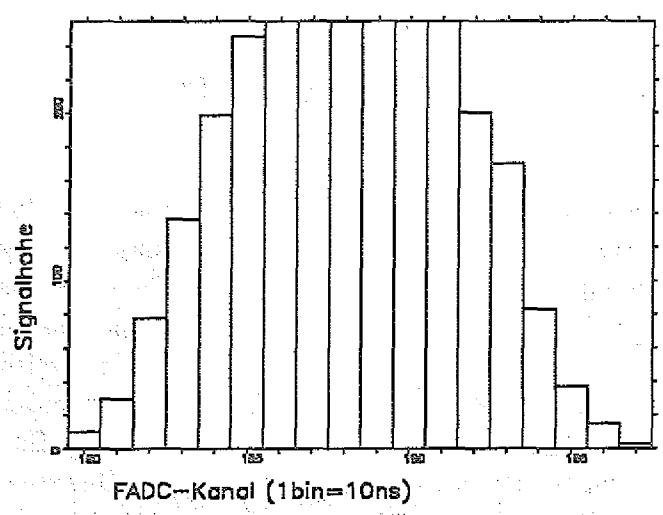


Abbildung 4.9: *Ein Ereignis, bei dem der* FADC *in Sättigung ist.*

4.3.1 Reduktion der FADC-Daten

Die Offline-Auswertung beginnt mit der Reduktion der Daten auf die eigentlichen Hits und dem Untergrundabzug. Als Untergrundsignal nehmen

wir den in dem Ereignis am häufigsten auftretenden Kanalinhalt an. Diese einfache Methode hat sich gut bewährt. Das wirkliche Ereignis wird dann aus den Rohdaten mit Hilfe eines Software-Hitfinders herausgefiltert. Dieser verlangt für einen *Start of Hit*, daß das Signal in fünf aufeinanderfolgenden Kanälen über einer gesetzten Schwelle[5] liegt, für ein *End of Hit* muß das Signal in zwei aufeinanderfolgenden Kanälen unterhalb dieser Schwelle liegen. Schließlich wird das Untergrundsignal abgezogen. Das Ergebnis dieser Datenreduktion ist in Abb. 4.10 für das gleiche Ereignis wie in Abb. 4.8 dargestellt. Berechnet man die Längen der reduzierten FADC-Signale, so

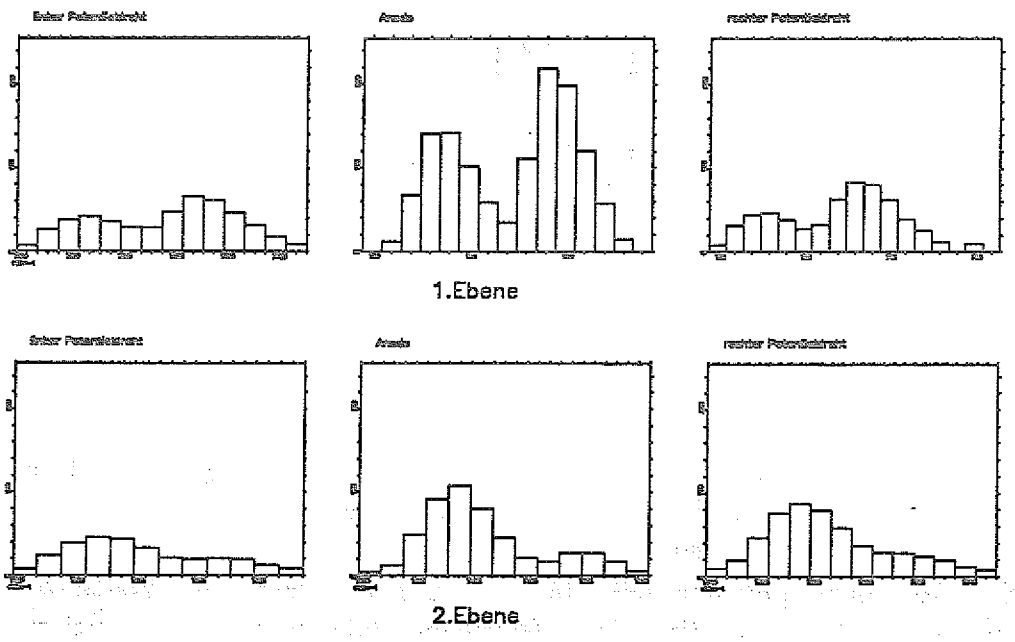


Abbildung 4.10: FADC-*Signale nach Anwenden des Software-Peakfinders und nach Untergrundabzug.*

erhält man die in Abb. 4.11 gezeigte Verteilung. Der Schwerpunkt dieser Verteilung liegt bei *130 ns*, was mit der erwarteten Signallänge recht gut übereinstimmt.

4.4 Vergleich der FADC-Daten und der ADC-Daten

Die Auslese mit ADCs basiert auf einer Messung der über die Dauer des Gates integrierten Ladung, die Auslese mit FADCs dagegen auf einer Spannungs-

[5]Berechnet als *Schwelle = Untergrund + 10*

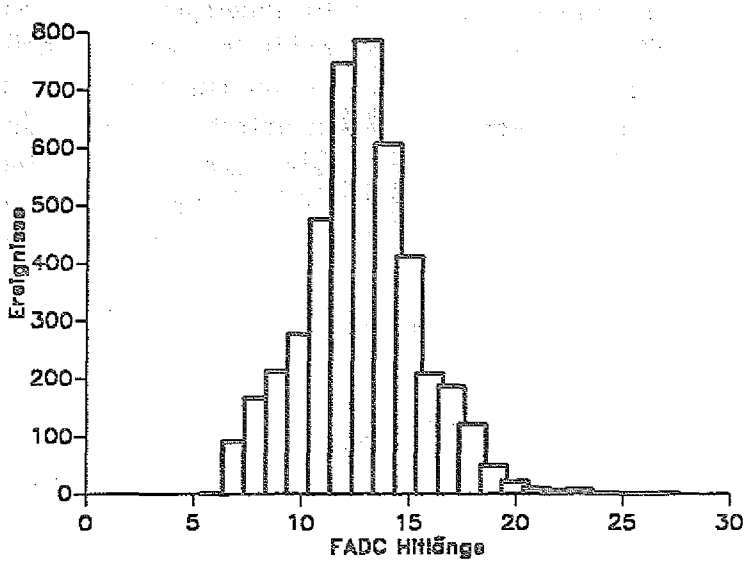


Abbildung 4.11: *Verteilung der Signallängen, wie sie aus den reduzierten* FADC*-Daten für einen Draht bestimmt werden.*

messung. Die Messwerte sind deshalb direkt nicht zu vergleichen. Bildet man aber für alle Ereignisse auf einem Draht jeweils die Summen über die FADC–Werte (siehe Abbildung 4.12), so sollten diese den entsprechenden ADC–Werten proportional sein. In Abbildung 4.13 sind diese Werte in einem Scatter–Plot gegeneinander aufgetragen. Dabei wurden die reduzierten FADC–Daten verwendet und nur solche Ereignisse eingetragen, bei denen das ADC–Signal über der gewählten Schwelle lag. Über einen weiten Bereich zeigt die Verteilung einen linearen Zusammenhang. Nur für große Signale weicht sie davon ab, weil die FADCs in Sättigung gehen (siehe Kapitel 4.3).

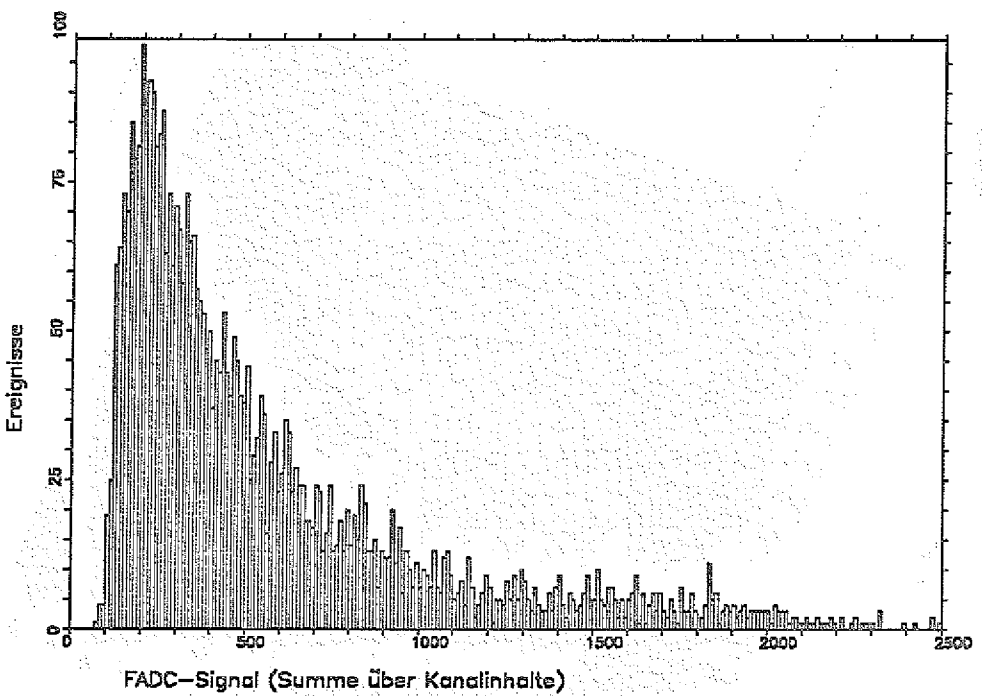


Abbildung 4.12: *Spektrum des über die Hitlänge aufsummierten* FADC-*Signals für einen Anodendraht.*

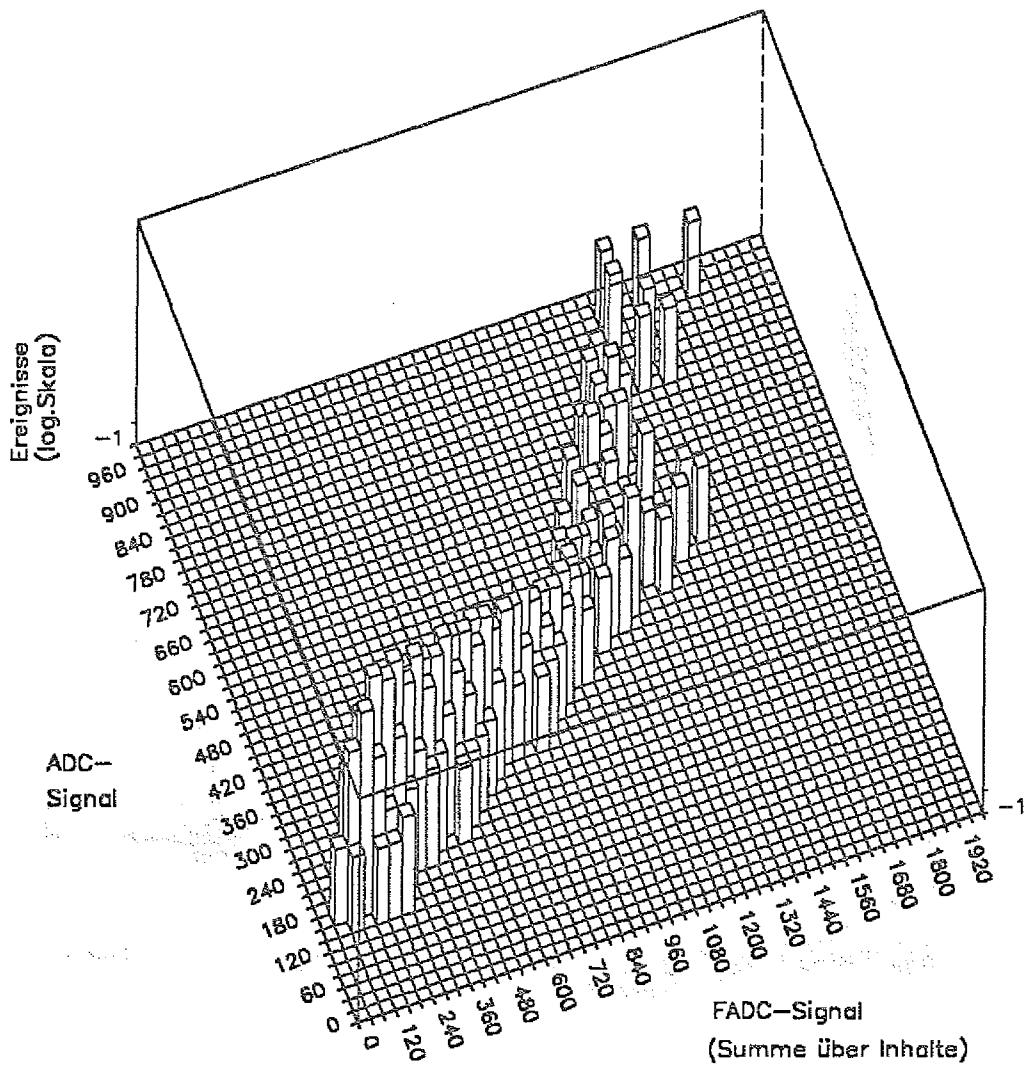


Abbildung 4.13: *Vergleich der* ADC- *und der* FADC-*Daten für einen Anodendraht. Für jedes Ereignis wurde die Summe über die* FADC-*Werte gebildet und das Ergebnis gegen den* ADC-*Wert aufgetragen.*

Kapitel 5

Zusammenfassung und Ausblick

Die Influenzdriftkammer verbindet eine hohe Ortsauflösung ($\approx 25\,\mu m$ für senkrechte Spuren) und Ratenverträglichkeit ($\approx 10^8 e^-/(cm^2 s)$) mit einer geringen Massenbelegung, und stellt deshalb einen einmaligen Detektortyp dar. In ihrer herkömmlichen Form mit ADC–Auslese verbietet sich allerdings ihr Einsatz in Fixed–Target–Experimenten durch die schlechte Ortsauflösung bei geneigten Spuren (Verschlechterung $\approx 6\mu m/1^o$). Ein Ausweg bietet sich in der Form einer asymmetrischen IDC mit FADC–Auslese an. Eine Abschätzung der verschiedenen Beiträge zur Ortsauflösung zeigt, daß die Winkelabhängigkeit bei geeignetem Gas auf einen Wert von $\leq 0,5\,\mu m/1^\circ$ begrenzt werden kann.

Im Rahmen dieser Arbeit wurde eine Testkammer mit mehreren Ebenen aufgebaut und mit minimalionisierenden Elektronen getestet. Es wurde gezeigt, daß ein stabiler Betrieb einer asymmetrischen IDC mit Kathodenabständen von 0,5 mm und 3,0 mm möglich ist und daß die Auslese mit FADCs eine Auflösung der zeitlichen Struktur der Signale zuläßt und konsistente Resultate liefert.

Wegen Problemen mit den Hochspannungszuführungen wurde die Kammer bei der Testmessung ausschließlich mit Argon/Methan betrieben. Mit diesem Gas erreicht man Gasverstärkung bei relativ geringen Feldstärken, es hat aber den Nachteil einer geringen Ionisationsdichte und großer transversaler Diffusion. Aus diesem Grunde ist eine Aussage über die absolute Ortsauflösung dieser Kammer noch nicht möglich. Zur Zeit wird an einer in diesen technischen Punkten verbesserten Version gearbeitet. Ein zweiter Testrun mit geeigneteren Gasen (Isobutan, Dimethyläther) ist für den Sommer 1990

in Vorbereitung. In dieser Messung soll dann die theoretisch vorhergesagte Verbesserung der Ortsauflösung bei geneigten Spuren experimentell bestätigt werden. Sollte dies gelingen, so steht mit der IDC mit FADC–Auslese ein idealer Vertexdetektor vor allem für Mittelenergiephysik–Experimente, wo es auf eine hohe Ortsauflösung und gleichzeitig geringe Massenbelegung ankommt, zur Verfügung.

Der erste Einsatz einer IDC im Experiment ist für Frühjahr/Sommer 1991 im Experiment PS185 vorgesehen, wo sie dazu dienen soll, die Spuren und Zerfallsvertices geladener Hyperonen ($\Sigma^{\pm}$ und $\overline{\Sigma^{\pm}}$) zu vermessen. Damit wird das Programm dieses Experiments komplettiert, und die Aussagen zur Reaktionsdynamik wesentlich erweitert.

Bibliographie

[FEH83] J. FEHLMANN,G. VIERTEL, Compilation of Data for Drift Chamber Operation, ETH Zürich, 1983

[FAR83] W. FARR,R. D. HEUER,A. WAGNER, IEEE Trans. Nucl. Sci. NS-30 (1983) 95.

[FIS78] J. FISCHER et al. , IEEE Trans. Nucl. Sci. NS-25 (1978) 794.

[GEI88] JOHANNES GEISBÜSCH, Diplomarbeit, Bonn 1989

[HUT85] J. HUTH, D. NYGREN, Nucl. Instr. and Meth. A241 (1985) 375.

[KIL81] K. KILIAN et al. , CERN/PSCC/81-69, PSCC/P49

[KRÄ86] MONIKA KRÄMER, Diplomarbeit, Siegen 1986

[MAT88] M. MATOBA et al. , IEEE Trans. Nucl. Sci. NS-35 (1988) 794.

[PDG88] PARTICLE DATA GROUP, Phys. Lett. B204, 1988

[ROD86] E. RODERBURG et al. , Nucl. Instr. and Meth. A252 (1986) 285.

[SEH89] G. SEHL, Jül–Spez–535, Jülich 1989

[WAL78] A. H. WALENTA, Nucl. Instr. and Meth. A151 (1978) 461.

[WAL83] A. H. WALENTA, Review of the Physics and Technology of Charged Particle Detectors, Proc. of 1983 SLC Summer Inst. on Part. Phys. , SLAC, Cal. , U.S.A.

[WAL85] P. v. WALTER, G. MILDNER, IEEE Trans. Nucl. Sci. NS-32 (1985) 626.

[WAL88] A. H. WALENTA ET AL. , Nucl. Instr. and Meth. A265 (1988) 69.

[WAL89] A. H. WALENTA, Limits of Position Resolution in an Induction Drift Chamber Due to Fluctuations in Ionisation Loss, Siegen 1989

[WEI89] W. WEIHS, G. ZECH, Numerical Computation of Electrostatic Fields in Multiwirechambers, Preprint SI-89-05, Siegen 1989

Verzeichnis der Firmenadressen

[CAEN] CAEN, Via Vetraia - 11, 55049 Viareggio, Italia

[LumaLampan] LUMALAMPAN AB, Box 701, S-39127 Kalmar, Sverige

[LeCroy] LE CROY GMBH, Werderstraße 48, Postfach 103767, D-6900 Heidelberg

[Struck] DR. B. STRUCK, Bäckerbarg 6, D-2000 Tangstedt

[Varian] VARIAN GMBH, Handwerksstraße 5-7, Postfach 801148, D-7000 Stuttgart 80

Anhang: Das Auswerte-Programm Idefix

Das in der Prozedursprache REXX geschriebene Programm IDEFIX EXEC dient zum Ausführen der Auswerteprogramme. Die Input– und Output–files werden definiert und die Fortran–Routinen aufgerufen.

```
/******************************
*  EXEC to start  I D E F I X  *
******************************/

SAY "-------------------"
SAY " Welcome to IDEFIX"
SAY "-------------------"
'CP SLEEP 3 SEC'
/*
**  select a runfile
*/
ARG RUNFILE
'SET EMSG OFF'
'STATE' RUNFILE
DO WHILE RC ^= 0
  'VMFCLEAR'
  'DROPBUF'
  SAY "-------------------------------------"
  SAY "List of Run-Files"
  SAY "-------------------------------------"
  'LISTFILE RUN_* DAT *'
  SAY "-------------------------------------"
  SAY "Enter the Run-Number (0=none):"
  PULL RUNNUMB
  IF RUNNUMB = 0 THEN CALL LEAVE
  RUNFILE = 'RUN_'RUNNUMB' DAT *'
  'STATE' RUNFILE
END
'SET EMSG ON'

/*
**  link CERNLIB
*/
'MATHDEF LIB CERNLIB'
/*
**  definition of standard input- and output-files:
**    lun 51 is the run data file
**    lun 21 is the H-Book output file
```

```
**    lun 22 is the output file for histogram listings
*/
'FILEDEF 61 DISK 'RUNFILE' ( PERM'
'FILEDEF 21 DISK HBOOK OUTPUT A2 ( PERM DISP MOD'
'FILEDEF 22 DISK HIST OUTPUT A2 ( PERM'
/*
**  display main menu
*/
CALL MENU
DO WHILE SELECT ^= 0
  /*
  **  call 'sub'-routines
  */
  IF SELECT = 1 THEN SAY " Not Yet Available !!!"
  IF SELECT = 2 THEN CALL FERA      /* FERA Evaluation */
  IF SELECT = 3 THEN CALL FADC      /* FADC Evaluation */
  IF SELECT = 4 THEN CALL FEFA      /* FERA/FADC Evaluation */
  /*
  **  display main menu
  */
   CALL MENU
END
/*
**  clear file definitions
*/
'FILEDEF 61 CLEAR'
'FILEDEF 21 CLEAR'
'FILEDEF 22 CLEAR'
/*
**  stop execution
*/
CALL LEAVE

/**********
* M E N U *
**********/

MENU:
  /*
  ** display the main menu
  */
  'VMFCLEAR'
  'DROPBUF'
  SAY "-------------------------------------"
  SAY "    M A I N  M E N U E"
  SAY "-------------------------------------"
  SAY " 1: Display Run Information"
  SAY " 2: Evaluate FERA Data"
  SAY " 3: Evaluate FADC Data"
  SAY " 4: Compare FERA Data with FADC Data"
  SAY " 0: Leave IDEFIX"
  SAY "-------------------------------------"
  SAY "Make Your Choice and Enter the Number:"
  /*
  **  select an item
  */
  PULL SELECT
  /*
  **  return to IDEFIX
  */
RETURN

/************
*  F E R A  *
```

```
************/

FERA:
  /*
  **  display FERA menu
  */
  CALL FERAMENU
  DO WHILE FERASEL ^= 0
    /*
    **  select and define parameter-file for FERA data evaluation
    */
    RC = 99
    DO WHILE RC ^= 0
      'VMFCLEAR'
      'DROPBUF'
      SAY "-------------------------------------"
      SAY "List of Parameter Files:"
      SAY "-------------------------------------"
      'LISTFILE FERA_* IFINPUT *'
      SAY "-------------------------------------"
      SAY "Make Your Choice and Enter the Number:"
      PULL PRMNUMB
      IF PRMNUMB = -1 THEN CALL LEAVE
      PRMFILE = 'FERA_'PRMNUMB' IFINPUT *'
      'STATE' PRMFILE
    END
    'FILEDEF 52 DISK 'PRMFILE' (PERM'
    /*
    **  select parameter file for H-BOOK spectra
    */
    IF FERASEL = 1 THEN 'FILEDEF 54 DISK HBOOFESP IFINPUT A2 (PERM'
    IF FERASEL = 2 THEN 'FILEDEF 54 DISK HBOOFESP IFINPUT A2 (PERM'
    IF FERASEL = 3 THEN 'FILEDEF 54 DISK HBOOFEHP IFINPUT A2 (PERM'
    IF FERASEL = 4 THEN 'FILEDEF 54 DISK HBOOFEDP IFINPUT A2 (PERM'
    IF FERASEL = 5 THEN 'FILEDEF 54 DISK HBOOFEDP IFINPUT A2 (PERM'
    IF FERASEL = 6 THEN 'FILEDEF 54 DISK HBOOFEDP IFINPUT A2 (PERM'
    /*
    **  forward selected menu-item to fortran routines
    */
    QUEUE FERASEL
    /*
    **  start Fortran routines
    */
    'CALLGR IFFERA IFLIBRY'
    /*
    ** clear file definitions
    */
    'FILEDEF 52 CLEAR'
    'FILEDEF 54 CLEAR'
    /*
    **  display FERA Menu
    */
    CALL FERAMENU
  END
  /*
  **  return to IDEFIX
  */
RETURN

/********************
*  F E R A M E N U  *
********************/

FERAMENU:
```

```
  /*
  **  display FERA Menu
  */
  'VMFCLEAR'
  'DROPBUF'
  SAY "-------------------------------------"
  SAY "    F E R A  M E N U E"
  SAY "-------------------------------------"
  SAY " 1: FERA Spectra (all cells)"
  SAY " 2: FERA Spectra (max. cell)"
  SAY " 3: Hit Plots"
  SAY " 4: dP/A Spectra (all cells)"
  SAY " 5: dP/A Spectra (max. cell)"
  SAY " 6: dP/A Spectra of Cells 8 & 10"
  SAY " 0: Return to MAIN Menu"
  SAY "-1: Leave IDEFIX"
  SAY "-------------------------------------"
  SAY "Make Your Choice and Enter the Number:"
  /*
  **  select an item
  */
  PULL FERASEL
  IF FERASEL = -1 THEN CALL LEAVE    /* stop execution */
  /*
  **  return to FERA
  */
RETURN


/************
*  F A D C  *
************/

FADC:
  /*
  **  display FADC menu
  */
  CALL FADCMENU
  DO WHILE FADCSEL ^= 0
    /*
    **  select and define parameter-file for FADC data evaluation
    */
    RC = 99
    DO WHILE RC ^= 0
      'VMFCLEAR'
      'DROPBUF'
      SAY "-------------------------------------"
      SAY "List of Parameter Files:"
      SAY "-------------------------------------"
      'LISTFILE FADC_* IFINPUT *'
      SAY "-------------------------------------"
      SAY "Make Your Choice and Enter the Number:"
      PULL PRMNUMB
      IF PRMNUMB = -1 THEN CALL LEAVE
      PRMFILE = 'FADC_'PRMNUMB' IFINPUT *'
      'STATE' PRMFILE
    END
    'FILEDEF 53 DISK 'PRMFILE' (PERM'
    /*
    **  define parameter file for H-BOOK spectra
    */
    IF FADCSEL = 4 THEN 'FILEDEF 54 DISK HBOOFASP IFINPUT A2 (PERM'
    IF FADCSEL = 5 THEN 'FILEDEF 54 DISK HBOOFALN IFINPUT A2 (PERM'
    IF FADCSEL = 6 THEN 'FILEDEF 54 DISK HBOOFADP IFINPUT A2 (PERM'
```

```
    /*
    **  forward selected item to Fortran routines
    */
    QUEUE FADCSEL
    /*
    **  start Fortran routines
    */
    'CALLGR IFFADC IFLIBRY'
    /*
    **  clear file-definitions
    */
    'FILEDEF 53 CLEAR'
    IF FADCSEL = 4 THEN 'FILEDEF 54 CLEAR'
    IF FADCSEL = 5 THEN 'FILEDEF 54 CLEAR'
    IF FADCSEL = 6 THEN 'FILEDEF 54 CLEAR'
    /*
    **  display FADC menu
    */
    CALL FADCMENU
  END
  /*
  **  Return to IDEFIX
  */
RETURN

/********************
*  F A D C M E N U  *
********************/

FADCMENU:
  /*
  **  display FADC menu
  */
  'VMFCLEAR'
  'DROPBUF'
  SAY "-------------------------------------"
  SAY "    F A D C  M E N U E"
  SAY "-------------------------------------"
  SAY " 1: Write Listing of FADC Data to Disk"
  SAY " 2: FADC Event-Plots (ONLINE data)"
  SAY " 3: FADC Event-Plots (reduced data)"
  SAY " 4: FADC spectra (integral of hit)"
  SAY " 5: Spectra of Length of FADC hits"
  SAY " 6: dP/A spectra (integral of hit)"
  SAY " 0: Return to MAIN Menu"
  SAY "-1: Leave IDEFIX"
  SAY "-------------------------------------"
  SAY "Make Your Choice and Enter the Number:"
  /*
  **  select an item
  */
  PULL FADCSEL
  IF FADCSEL = -1 THEN CALL LEAVE    /* stop execution */
  /*
  **  return to FADC
  */
RETURN


/**********************
*  F E R A / F A D C  *
**********************/

FEFA:
```

```
/*
**  display FERA/FADC menu
*/
CALL FEFAMENU
DO WHILE FEFASEL ^= 0
  /*
  **  select parameter-file for FERA data
  */
  RC = 99
  DO WHILE RC ^= 0
    'VMFCLEAR'
    'DROPBUF'
    SAY "-------------------------------------"
    SAY "List of FERA Parameter Files:"
    SAY "-------------------------------------"
    'LISTFILE FERA_* IFINPUT *'
    SAY "-------------------------------------"
    SAY "Make Your Choice and Enter the Number:"
    PULL PRMNUMB
    IF PRMNUMB = -1 THEN CALL LEAVE
    PRMFILE = 'FERA_'PRMNUMB' IFINPUT *'
    'STATE' PRMFILE
  END
  'FILEDEF 52 DISK 'PRMFILE' (PERM'
  /*
  **  select parameter-file for FADC data
  */
  RC = 99
  DO WHILE RC ^= 0
    'VMFCLEAR'
    'DROPBUF'
    SAY "-------------------------------------"
    SAY "List of FADC Parameter Files:"
    SAY "-------------------------------------"
    'LISTFILE FADC_* IFINPUT *'
    SAY "-------------------------------------"
    SAY "Make Your Choice and Enter the Number:"
    PULL PRMNUMB
    IF PRMNUMB = -1 THEN CALL LEAVE
    PRMFILE = 'FADC_'PRMNUMB' IFINPUT *'
    'STATE' PRMFILE
  END
  'FILEDEF 53 DISK 'PRMFILE' (PERM'
  /*
  **  select parameterfile for H-BOOK spectra
  */
  IF FEFASEL = 1 THEN 'FILEDEF 54 DISK HBOOFFSP IFINPUT A2 (PERM'
  IF FEFASEL = 2 THEN 'FILEDEF 54 DISK HBOOFFDP IFINPUT A2 (PERM'
  /*
  **  forward the selected item to Fortran routines
  */
  QUEUE FEFASEL
  /*
  **  start Fortran routines
  */
  'CALLGR IFFEFA IFLIBRY'
  /*
  **  clear file-definitions
  */
  'FILEDEF 52 CLEAR'
  'FILEDEF 53 CLEAR'
  'FILEDEF 54 CLEAR'
  /*
  **  display FERA/FADC menu
```

```
    */
    CALL FEFAMENU
  END
/*
**  Return to IDEFIX
*/
RETURN

/************************************
*  F E R A  /  F A D C  -  M E N U  *
************************************/

FEFAMENU:
  /*
  **  display FERA/FADC menu
  */
  'VMFCLEAR'
  'DROPBUF'
  SAY "----------------------------------------"
  SAY "    F E R A  /  F A D C  -  M E N U E"
  SAY "----------------------------------------"
  SAY " 1: Spectra of FERA data vs. FADC data"
  SAY " 2: Spectra of dP/A data (FERA vs. FADC)"
  SAY " 0: Return to MAIN Menu"
  SAY "-1: Leave IDEFIX"
  SAY "-------------------------------------"
  SAY "Make Your Choice and Enter the Number:"
  /*
  **  select an item
  */
  PULL FEFASEL
  IF FEFASEL = -1 THEN CALL LEAVE    /* stop execution */
  /*
  **  Return to FADC
  */
RETURN

/**************
*  L E A V E  *
**************/

LEAVE:
  /*
  **  stop execution of IDEFIX
  */
  'vmfclear'
  say "-----------------------------"
  say " Thank You For Using IDEFIX."
  say " Hope to See You Again Soon."
  say "      Have A Nice Day !"
  say "-----------------------------"
EXIT
```

Es gibt drei in Fortran geschriebene Hauptprogramme zur Auswertung der ADC–Daten (IFFERA FORTRAN), der FADC–Daten (IFFADC FORTRAN und zum Vergleich der ADC– und der FADC–Daten (IFFEFA FORTRAN). Sie sind in ihrer Struktur ähnlich, exemplarisch ist hier das Programm IFADC, das zur Auswertung der FADC–Daten dient, abgedruckt.

```
c     ==============================================================
```

```
c     Evaluate Online FADC Data
c     ==================================================================

      program iffadc

c     ------------------------------------------------------------------
c --> Declarations
c     ------------------------------------------------------------------

c --> Online Run-Header:
      common /conlhd/
     +   time, date,
     +   run, nevent,
     +   nfera, ferath, ntdc, tdcth,
     +   nadc, radc, adcth, nfadc, fadcba,
     +   bsamp, bscan, esamp, escan, msamp, mscan,
     +   anoth, potth, lchan, rchan
      character*10
     +   time, date
      integer*2
     +   run, nevent,
     +   nfera, ferath(48), ntdc, tdcth(48),
     +   nadc, radc, adcth(48), nfadc, fadcba(48),
     +   bsamp, bscan, esamp, escan, msamp, mscan,
     +   anoth, potth, lchan, rchan
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> H-BOOK
      common /pawc/
     +   hmemor(150000)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local :
      character*1
     +   decrun, decev, decide
      integer*4
     +   menu, i

c     ------------------------------------------------------------------
c --> calculations
c     ------------------------------------------------------------------

c --> read IDEFIX menu item from terminal buffer
      rewind 5
      read (5,*) menu
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> Initiate H-Book, if necessary
      if (
     +   (menu.eq.4)
     +   .or.
     +   (menu.eq.5)
     +   .or.
     +   (menu.eq.6)
     +   ) then
         call HLIMIT (150000)
         call HOUTPU (21)
      end if
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> start GR-Software if necessary
      if (
     +   (menu.eq.2)
     +   .or.
     +   (menu.eq.3)
     +   ) then
         call GRSTRT (35,8)
         call GRSCLC (2.,1.,39.5,28.7)
```

```
      endif
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> read online run header
      call onlhdrd
c --> read FADC data evaluation parameters
      call faprmrd
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> read H-Book spectra parameters, initiate H-BOOK spectra
      if (
     +   (menu.eq.4)
     +   .or.
     +   (menu.eq.5)
     +   ) then
         call hbprmrd
         call ihwiid
      else if (menu.eq.6) then
         call hbprmrd
         call ihan1d
         call ihan2d
      end if
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      decrun = 'y'
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      do while ((decrun.eq.'y').or.(decrun.eq.'Y'))
         call obey ('VMFCLEAR')
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c -----> evaluation
         if (menu.eq.1) then
c --------> generate listing of sorted FADC data
            decev = 'y'
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            do while ((decev.eq.'y').or.(decev.eq.'Y'))
c -----------> read event-vector
               call onlevrd
c -----------> sort fadc-data
               call fadcsrt
               call fadclst
               call obey ('VMFCLEAR')
               write (6,*) 'Listing of One Event Completed.'
               write (6,*) 'Do You Want More FADC-Listings (y/n) ?'
               rewind 5
               read (5,'(a1)') decev
            end do
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         else if (menu.eq.2) then
c --------> generate hit-plots of 'raw' FADC data
            decev = 'y'
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            do while ((decev.eq.'y').or.(decev.eq.'Y'))
               call onlevrd
c -----------> sort fadc-data
               call fadcsrt
c -----------> hitfinder, reduce FADC data
               call fahitfi
c -----------> fill data in spectra
               call faevpla
               write (6,*) 'Evaluation of One Event Completed.'
               write (6,*) 'Do You Want More FADC-Eventplots (y/n) ?'
               rewind 5
               read (5,'(a1)') decev
            end do
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         else if (menu.eq.3) then
c --------> generate hit-plots of reduced FADC data
```

```
            decev = 'y'
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            do while ((decev.eq.'y').or.(decev.eq.'Y'))
               call onlevrd
c -----------> sort fadc-data
               call fadcsrt
c -----------> hitfinder, reduce FADC data
               call fahitfi
c -----------> fill data in spectra
               call faevplr
               write (6,*) 'Evaluation of One Event Completed.'
               write (6,*) 'Do You Want More FADC-Eventplots (y/n) ?'
               rewind 5
               read (5,'(a1)') decev
            end do
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         else if (menu.eq.4) then
c --------> fill H-BOOK spectra with FADC data (integral of hit)
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            do i = 1, nevent, 1
c -----------> display message on screen each 128th event
               if ((ishft(ishft(i,-7),7)).eq.i)
     +            write (6,9000) run, i, nevent
c -----------> read event-vector
               call onlevrd
c -----------> sort fadc-data
               call fadcsrt
c -----------> hitfinder, reduce FADC data
               call fahitfi
c -----------> fill data in spectra
               call fhfadat
            end do
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         else if (menu.eq.5) then
c --------> fill H-BOOK spectra with length of FADC hits
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            do i = 1, nevent, 1
c -----------> display message on screen each 128th event
               if ((ishft(ishft(i,-7),7)).eq.i)
     +            write (6,9000) run, i, nevent
c -----------> read event-vector
               call onlevrd
c -----------> sort fadc-data
               call fadcsrt
c -----------> hitfinder, reduce FADC data
               call fahitfi
c -----------> fill data in spectra
               call fhfalen
            end do
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         else if (menu.eq.6) then
c --------> fill H-BOOK spectra with dP/A data
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            do i = 1, nevent, 1
c -----------> display message on screen each 128th event
               if ((ishft(ishft(i,-7),7)).eq.i)
     +            write (6,9000) run, i, nevent
c -----------> read event-vector
               call onlevrd
c -----------> sort fadc-data
               call fadcsrt
c -----------> hitfinder, reduce FADC data
               call fahitfi
c -----------> fill data in spectra
```

```
               call fhfadpa
            end do
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         end if
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c -----> join data of several runs (optional)
         write (6,*) '-------------------------------------------'
         write (6,*) ' Evaluation of one run completed.'
         write (6,*) ' Do You want to add data of another run (y/n)?'
         rewind 5
         read (5,'(a1)') decrun
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         if ((decrun.eq.'y').or.(decrun.eq.'Y')) then
            call obey ('IFADDRUN')
            call onlhdrd
         end if
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> close GR-software if necessary
      if (
     +   (menu.eq.2)
     +   .or.
     +   (menu.eq.3)
     +   ) then
         call GRENDE
      end if
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> output of H-BOOK results
      if (
     +   (menu.eq.4)
     +   .or.
     +   (menu.eq.5)
     +   .or.
     +   (menu.eq.6)
     +   ) then
c -----> listing of contents of 1-dim histograms (optional)
         call obey ('VMFCLEAR')
         decide = 'n'
         write (6,*) 'Do you want listings of histogram data (1-dim) ?'
         rewind 5
         read (5,'(a1)') decide
         if ((decide.eq.'Y').or.(decide.eq.'y')) call listhb1
c -----> listing of contents of 2-dim histograms (optional)
         call obey ('VMFCLEAR')
         decide = 'n'
         write (6,*) 'Do you want listings of histogram data (2-dim) ?'
         rewind 5
         read (5,'(a1)') decide
         if ((decide.eq.'Y').or.(decide.eq.'y')) call listhb2
c ----->
         call HISTDO
      end if

c     ------------------------------------------------------------------
c     format statements
c     ------------------------------------------------------------------

9000  format ('   Evaluating Run ',i4,'  -  Event ',i6,' of ',i6)

c     ------------------------------------------------------------------
      end
```

Alle Unterprogramme sind in dem file IFLIBRY FORTRAN zusammengefaßt.

```
c     ------------------------------------------------------------------
c     Library of IDEFIX Subroutines
c     ------------------------------------------------------------------
c     Table of Contents:
c      - ONLHDRD (Read Online Run Header)
c      - ONLEVRD (Read Online Event Vector)
c      - FEPRMRD (Read Parameters for FERA Data Evaluation)
c      - FAPRMRD (Read Parameters for FADC Data Evaluation)
c      - HBPRMRD (Read Parameters for Initiation of H-Book Spectra)
c      - IHWI1D  (Initiate 1-Dim H-Book-Spectra for Data of All Wires)
c      - IHWI2D  (Initiate 2-Dim H-Book-Spectra for Data of All Wires)
c      - IHAN1D  (Initiate 1-Dim H-Book-Spectra for Data of Anode Wires)
c      - IHAN2D  (Initiate 2-Dim H-Book-Spectra for Data of Anode Wires)
c      - IHHITPL (Initiate H-Book for Hit-Plots)
c      - FERASRT (Sort FERA Data)
c      - FHFEDAT (Fill Spectra with FERA Data - all cells)
c      - FHFEMAX (Fill Spectra with FERA Data - max. cell)
c      - FHFEDPA (Fill Spectra with FERA dP/A Data - all cells)
c      - FHFEDPM (Fill Spectra with FERA dP/A Data - max. cell)
c      - FHFEHPL (Fill Spectra with FERA Hit-Plots)
c      - FADCSRT (Sort FADC Data)
c      - FAHITFI (FADC Hitfinder)
c      - FADCLST (Write Listing of FADC Data to Disk)
c      - FAEVPLA (Fill Sorted FADC Data in GR-Histograms)
c      - FAEVPLR (Fill Reduced FADC Data in GR-Histograms)
c      - FHFADAT (Fill Spectra with FADC Data - all cells)
c      - FHFALEN (Fill Spectra with Length of FADC Hits)
c      - FHFADPA (Fill Spectra With FADC dP/A Data - Integral of Hit)
c      - FHFFDAT (Fill Spectra with FERA-Data vs. FADC-Data)
c      - FHFFDPA (Fill Spectra with FERA- vs. FADC- dP/A-Data)
c      - LISTHB1 (List Data of 1-Dim H-BOOK Histograms in file)
c      - LISTHB2 (List Data of 2-Dim H-BOOK Histograms in file)
c     ------------------------------------------------------------------

*********************************************************************

c     ==================================================================
c     read run header from ONLINE data file
c     ==================================================================

      subroutine onlhdrd

c     ------------------------------------------------------------------
c     declarations
c     ------------------------------------------------------------------

c --> online run-header
      common /conlhd/
     +   time, date,
     +   run, nevent,
     +   nfera, ferath, ntdc, tdcth,
     +   nadc, radc, adcth, nfadc, fadcba,
     +   bsamp, bscan, esamp, escan, msamp, mscan,
     +   anoth, potth, lchan, rchan
      character*10
     +   time, date
      integer*2
     +   run, nevent,
     +   nfera, ferath(48), ntdc, tdcth(48),
     +   nadc, radc, adcth(48), nfadc, fadcba(48),
     +   bsamp, bscan, esamp, escan, msamp, mscan,
```

```
     +   anoth, potth, lchan, rchan

c     ------------------------------------------------------------
c     calculations
c     ------------------------------------------------------------

c --> rewind to top of file, read run header
      rewind 51
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      read (51,9000) time, date
      read (51,9010) run, nevent
      read (51,9020) nfera
      read (51,9030) (ferath(i), i=1,nfera)
      read (51,9020) ntdc
      read (51,9030) (tdcth(i), i=1,ntdc)
      read (51,9010) nadc
      read (51,9020) radc
      read (51,9030) (adcth(i), i=1,nadc)
      read (51,9020) nfadc
      read (51,9030) (fadcba(i), i=1,nfadc)
      read (51,9010)
     +   bsamp, bscan, esamp, escan, msamp, mscan, anoth,
     +   potth, lchan, rchan

c     ------------------------------------------------------------
c     format statements
c     ------------------------------------------------------------

9000  format (6x,a10)
9010  format (30x,i6)
9020  format (30x,i6/)
9030  format (13i6)

c     ------------------------------------------------------------
      end

*********************************************************************

c     ============================================================
c     read event-vector from ONLINE data file
c     ============================================================

      subroutine onlevrd

c     ------------------------------------------------------------
c     declarations
c     ------------------------------------------------------------

c --> online event-vector
      common /conlev/
     +   ievent, nentrs, ev
      integer*2
     +   ievent, nentrs, ev(10000)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local
      integer*2
     +   i

c     ------------------------------------------------------------
c     calculations
c     ------------------------------------------------------------

c --> clear event-vector
      do i = 1, 10000, 1
```

```
         ev(i) = 0
      end do
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> read event-data
      read (51,9000,end=10) ievent
      read (51,9010,end=10) nentrs
      read (51,9020,end=10) (ev(i), i=1,nentrs)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      goto 99999

c     ------------------------------------------------------------------
c     error messages
c     ------------------------------------------------------------------

10    write (21,*) 'Error ocurred reading data of event ',ievent

c     ------------------------------------------------------------------
c     format statements
c     ------------------------------------------------------------------

9000  format (/30x,i6)
9010  format (30x,i6)
9020  format (13i6)

c     ------------------------------------------------------------------
99999 end

*********************************************************************

c     ==================================================================
c     Read parameters for FERA data evaluation
c     ==================================================================

      subroutine feprmrd

c     ------------------------------------------------------------------
c     declarations
c     ------------------------------------------------------------------

c --> online run-header
      common /conlhd/
     +   time, date,
     +   run, nevent,
     +   nfera, ferath, ntdc, tdcth,
     +   nadc, radc, adcth, nfadc, fadcba,
     +   bsamp, bscan, esamp, escan, msamp, mscan,
     +   anoth, potth, lchan, rchan
      character*10
     +   time, date
      integer*2
     +   run, nevent,
     +   nfera, ferath(48), ntdc, tdcth(48),
     +   nadc, radc, adcth(48), nfadc, fadcba(48),
     +   bsamp, bscan, esamp, escan, msamp, mscan,
     +   anoth, potth, lchan, rchan
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> channel to plane/wire relation
      common /cch2pw/
     +   plane, wire
      integer*2
     +   plane(60), wire(60)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> fera data evaluation parameters
      common /cfeprm/
```

```
     +   fepl, fewi,
     +   felcut, feucut, fenorm, feback
      integer*2
     +   fepl(60), fewi(60),
     +   felcut(5,12), feucut(5,12), feback(2)
      real*4
     +   fenorm(5,12)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local:
      character*1
     +   c
      integer*2
     +   chn

c     ---------------------------------------------------------------
c     calculations
c     ---------------------------------------------------------------

c --> clear plane, wire
      do chn = 1, 60, 1
         fepl(chn) = 0
         fewi(chn) = 0
         plane(chn) = 0
         wire(chn) = 0
      end do
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> rewind to top of file, skip comment lines
      rewind 52
      c = '*'
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      do while (c.eq.'*')
         read (52,'(a1)') c
      end do
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> read plane and wire of FERA channel for background subtraction
      read (52,9010) feback(1), feback(2)
c --> skip comment lines
      c = '*'
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      do while (c.eq.'*')
         read (52,'(a1)') c
      end do
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> read parameters for evaluation of FERA data
      do chn = 1, nfera, 1
         read (52,9020,end=99999,err=10)
     +      fepl(chn), fewi(chn),
     +      felcut(fepl(chn),fewi(chn)),
     +      feucut(fepl(chn),fewi(chn)),
     +      fenorm(fepl(chn),fewi(chn))
         plane(chn) = fepl(chn)
         wire(chn) = fewi(chn)
      end do
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      goto 99999

c     ---------------------------------------------------------------
c     error messages
c     ---------------------------------------------------------------

10    write (21,*) 'Error occured reading fera parameters.'

c     ---------------------------------------------------------------
c     format statements
```

```
c     ------------------------------------------------------------

9000  format (30x,i6/30x,a40,2(/30x,i6,2(/30x,f6.0)))
9010  format (9x,2(1x,i6))
9020  format (9x,4(1x,i6),1x,f6.3)

c     ------------------------------------------------------------
99999 end

*********************************************************************

c     ============================================================
c     Read Parameters for FADC Data Evaluation
c     ============================================================

      subroutine faprmrd

c     ------------------------------------------------------------
c     declarations
c     ------------------------------------------------------------

c --> online run-header
      common /conlhd/
     +   time, date,
     +   run, nevent,
     +   nfera, ferath, ntdc, tdcth,
     +   nadc, radc, adcth, nfadc, fadcba,
     +   bsamp, bscan, esamp, escan, msamp, mscan,
     +   anoth, potth, lchan, rchan
      character*10
     +   time, date
      integer*2
     +   run, nevent,
     +   nfera, ferath(48), ntdc, tdcth(48),
     +   nadc, radc, adcth(48), nfadc, fadcba(48),
     +   bsamp, bscan, esamp, escan, msamp, mscan,
     +   anoth, potth, lchan, rchan
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> channel to plane/wire relation
      common /cch2pw/
     +   plane, wire
      integer*2
     +   plane(60), wire(60)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> fadc data evaluation parameters
      common /cfaprm/
     +   fapl, fawi
      integer*2
     +   fapl(60), fawi(60)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local:
      character*1
     +   c
      integer*2
     +   chn

c     ------------------------------------------------------------
c     calculations
c     ------------------------------------------------------------

c --> clear plane, wire
      do chn = 1, 60, 1
         fapl(chn) = 0
         fawi(chn) = 0
```

```
         plane(chn) = 0
         wire(chn) = 0
      end do
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> rewind to top of file, skip comment lines
      rewind 53
      c = '*'
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      do while (c.eq.'*')
         read (53,'(a1)') c
      end do
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> read plane and wire
      do chn = 1, nfadc, 1
         read (53,9000,end=99999,err=10) fapl(chn), fawi(chn)
         plane(chn) = fapl(chn)
         wire(chn) = fawi(chn)
      end do
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      goto 99999

c     ------------------------------------------------------------------
c     error messages
c     ------------------------------------------------------------------

10    write (21,*) 'Error occured reading fadc-to-wire relation.'

c     ------------------------------------------------------------------
c     format statements
c     ------------------------------------------------------------------

9000  format (9x,2(1x,i6))

c     ------------------------------------------------------------------
99999 end

*********************************************************************

c     ==================================================================
c     Read Parameters for H-Book Histograms
c     ==================================================================

      subroutine hbprmrd

c     ------------------------------------------------------------------
c     declarations
c     ------------------------------------------------------------------

c --> H-Book Histogram Parameters
      common /chbprm/
     +   hindex, chtitl, nx, xmin, xmax, ny, ymin, ymax
      character*40
     +   chtitl
      integer*4
     +   hindex, nx, ny
      real*4
     +   xmin, xmax, ymin, ymax
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local:
      character*1
     +   c

c     ------------------------------------------------------------------
c     calculations
```

```
c     ---------------------------------------------------------------

c --> rewind to top of file, skip comment lines
      rewind 54
      c = '*'
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      do while (c.eq.'*')
         read (54,'(a1)') c
      end do
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> read parameters for initiation of H-BOOK histograms
      read (54,9000,err=10)
     +   hindex, chtitl, nx, xmin, xmax, ny, ymin, ymax
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      goto 99999

c     ---------------------------------------------------------------
c     error messages
c     ---------------------------------------------------------------

10    write (21,*) 'Error occured reading H-Book Parameters.'

c     ---------------------------------------------------------------
c     format statements
c     ---------------------------------------------------------------

9000  format (30x,i6/30x,a40,2(/30x,i6,2(/30x,f6.0)))

c     ---------------------------------------------------------------
99999 end

*********************************************************************

c     ===============================================================
c     sort FERA data
c     ===============================================================

      subroutine ferasrt

c     ---------------------------------------------------------------
c     declarations
c     ---------------------------------------------------------------

c --> online event-vector
      common /conlev/
     +   ievent, nentrs, ev
      integer*2
     +   ievent, nentrs, ev(10000)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> fera data evaluation parameters
      common /cfeprm/
     +   fepl, fewi,
     +   felcut, feucut, fenorm, feback
      integer*2
     +   fepl(60), fewi(60),
     +   felcut(5,12), feucut(5,12), feback(2)
      real*4
     +   fenorm(5,12)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> sorted fera-data
      common /cfesrt/
     +   fera
      integer*2
     +   fera(5,12)
```

```
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local
      integer*2
     +   pl, wi, pt

c     ---------------------------------------------------------------
c     calculations
c     ---------------------------------------------------------------

c --> nentrs = 0: return to main program
      if (nentrs.gt.0) then
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c -----> initiate matrix for sorted data
         do pl = 1, 5, 1
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            do wi = 1, 12, 1
               fera(pl,wi) = 0
            end do
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         end do
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c -----> search for start of fera-data (ev(pt) = -1000)
         pt = 1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         do while (
     +      (ev(pt) .ne. -1000)
     +      .and.
     +      (pt.lt.nentrs)
     +      )
            pt = pt+1
         end do
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         if (pt.eq.nentrs) then
            write (21,*) 'No FERA Data in Event ',ievent
         else
            pt=pt+1
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --------> sort FERA data (end of fera-data: ev(pt) = -1111)
            do while (ev(pt).ne.-1111)
c -----------> ev(pt) = - number of channel
c -----------> ev(pt+1) = content of channel ev(pt)
               fera(fepl(-ev(pt)),fewi(-ev(pt))) = ev(pt+1)
               pt = pt+2
            end do
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         end if
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end if

c     ---------------------------------------------------------------
      end

*********************************************************************

c     ===============================================================
c     Initiate 1-dim H-BOOK-spectra for all wires
c     ===============================================================

      subroutine ihwiid

c     ---------------------------------------------------------------
c     declarations
c     ---------------------------------------------------------------
```

```
c --> channel to wire/plane relation
      common /cch2pw/
     +   plane, wire
      integer*2
     +   plane(60), wire(60)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> H-Book histogram parameters
      common /chbprm/
     +   hindex, chtitl, nx, xmin, xmax, ny, ymin, ymax
      character*40
     +   chtitl
      integer*4
     +   hindex, nx, ny
      real*4
     +   xmin, xmax, ymin, ymax
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local:
      integer*4
     +   pl, wi, fl(5,12), chn, n

c     ------------------------------------------------------------------
c     calculations
c     ------------------------------------------------------------------

c --> clear flags
      do pl = 1, 5, 1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         do wi = 1, 12, 1
            fl(pl,wi) = 0
         end do
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> set flag, if wire exists
      do chn = 1, 60, 1
         pl = plane(chn)
         wi = wire(chn)
         if (
     +      (pl.gt.0)
     +      .and.
     +      (wi.gt.0)
     +      )
     +      fl(pl,wi) = 1
      end do
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> initiate histograms
      do pl = 1, 4, 1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         do wi = 1, 12, 1
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            if (fl(pl,wi).eq.1) then
               n = hindex+100*pl+wi
               call HBOOK1 (n, chtitl, nx, xmin, xmax, 0.)
            end if
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         end do
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do

c     ------------------------------------------------------------------
      end

*********************************************************************
```

```
c     ==================================================================
c     Initiate 2-dim H-BOOK-spectra for all wires
c     ==================================================================

      subroutine ihwi2d

c     ------------------------------------------------------------------
c     declarations
c     ------------------------------------------------------------------

c --> channel to wire/plane relation
      common /cch2pw/
     +   plane, wire
      integer*2
     +   plane(60), wire(60)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> H-Book histogram parameters
      common /chbprm/
     +   hindex, chtitl, nx, xmin, xmax, ny, ymin, ymax
      character*40
     +   chtitl
      integer*4
     +   hindex, nx, ny
      real*4
     +   xmin, xmax, ymin, ymax
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local:
      integer*4
     +   pl, wi, fl(5,12), chn, n

c     ------------------------------------------------------------------
c     calculations
c     ------------------------------------------------------------------

c --> clear flags
      do pl = 1, 5, 1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         do wi = 1, 12, 1
            fl(pl,wi) = 0
         end do
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> set flag, if wire exists
      do chn = 1, 60, 1
         pl = plane(chn)
         wi = wire(chn)
         if (
     +      (pl.gt.0)
     +      .and.
     +      (wi.gt.0)
     +      )
     +      fl(pl,wi) = 1
      end do
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> initiate histograms
      do pl = 1, 4, 1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         do wi = 1, 12, 1
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            if (fl(pl,wi).eq.1) then
               n = 600+hindex+100*pl+wi
               call HBOOK2 (
     +            n, chtitl, nx, xmin, xmax,
```

```
     +            ny, ymin, ymax, 0.
     +            )
            end if
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         end do
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do

c     ------------------------------------------------------------------
      end

*********************************************************************

c     ==================================================================
c     Initiate 1-dim H-BOOK spectra for anode wires (dP/A - data)
c     ==================================================================

      subroutine ihan1d

c     ------------------------------------------------------------------
c     declarations
c     ------------------------------------------------------------------

c --> channel to wire/plane relation
      common /cch2pw/
     +   plane, wire
      integer*2
     +   plane(60), wire(60)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> H-Book Histogram Parameters
      common /chbprm/
     +   hindex, chtitl, nx, xmin, xmax, ny, ymin, ymax
      character*40
     +   chtitl
      integer*4
     +   hindex, nx, ny
      real*4
     +   xmin, xmax, ymin, ymax
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local:
      integer*4
     +   pl, wi, fl(5,12), chn, n

c     ------------------------------------------------------------------
c     calculations
c     ------------------------------------------------------------------

c --> clear flags
      do pl = 1, 5, 1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         do wi = 1, 12, 1
            fl(pl,wi) = 0
         end do
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> set flag, if anode wire exists
      do chn = 1, 60, 1
         pl = plane(chn)
         wi = wire(chn)
         if (
     +      (pl.gt.0)
     +      .and.
     +      (wi.gt.0)
```

```
     +      )
     +      fl(pl,wi) = 1
      end do
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> initiate histograms
      do pl = 1, 4, 1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         do wi = 2, 10, 2
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --------> one-dimensional histograms
            if (fl(pl,wi).eq.1) then
               n = hindex*100*pl+wi
               call HBOOK1 (n, chtitl, nx, xmin, xmax, 0.)
            end if
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         end do
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do

c     ------------------------------------------------------------------
      end

*********************************************************************

c     ==================================================================
c     Initiate 2-dim H-BOOK spectra for anode wires (dP/A - data)
c     ==================================================================

      subroutine ihan2d

c     ------------------------------------------------------------------
c     declarations
c     ------------------------------------------------------------------

c --> channel to wire/plane relation
      common /cch2pw/
     +   plane, wire
      integer*2
     +   plane(60), wire(60)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> H-Book Histogram Parameters
      common /chbprm/
     +   hindex, chtitl, nx, xmin, xmax, ny, ymin, ymax
      character*40
     +   chtitl
      integer*4
     +   hindex, nx, ny
      real*4
     +   xmin, xmax, ymin, ymax
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local:
      integer*4
     +   pl, wi, fl(5,12), chn, n

c     ------------------------------------------------------------------
c     calculations
c     ------------------------------------------------------------------

c --> clear flags
      do pl = 1, 5, 1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         do wi = 1, 12, 1
            fl(pl,wi) = 0
         end do
```

```
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> set flag, if anode wire exists
      do chn = 1, 60, 1
         pl = plane(chn)
         wi = wire(chn)
         if (
     +      (pl.gt.0)
     +      .and.
     +      (wi.gt.0)
     +      )
     +      fl(pl,wi) = 1
      end do
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> initiate histograms
      do pl = 1, 4, 1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         do wi = 2, 10, 2
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            if (fl(pl,wi).eq.1) then
               n = hindex*600*100*pl+wi
               call HBOOK2 (
     +            n, chtitl, nx, xmin, xmax,
     +            ny, ymin, ymax, 0.
     +               )
            end if
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         end do
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do

c     ------------------------------------------------------------------
      end

*********************************************************************

c     ==================================================================
c     Initiate H-BOOK spectra for hit plots
c     ==================================================================

      subroutine ihhitpl

c     ------------------------------------------------------------------
c     declarations
c     ------------------------------------------------------------------

c --> channel to wire/plane relation
      common /cch2pw/
     +   plane, wire
      integer*2
     +   plane(60), wire(60)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> H-Book histogram parameters
      common /chbprm/
     +   hindex, chtitl, nx, xmin, xmax, ny, ymin, ymax
      character*40
     +   chtitl
      integer*4
     +   hindex, nx, ny
      real*4
     +   xmin, xmax, ymin, ymax
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local :
```

```
      integer*4
     +   pl, fl(5), chn, n

c     ------------------------------------------------------------------
c     calculations
c     ------------------------------------------------------------------

c --> clear flags
      do pl = 1, 5, 1
         fl(pl) = 0
      end do
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> set flag, if plane exists
      do chn = 1, 60, 1
         pl = plane(chn)
         if (pl.gt.0) fl(pl) = 1
      end do
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      do pl = 1, 4, 1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         if (fl(pl).eq.1) then
c --------> initiate histogram
            n = hindex*100*pl
            call HBOOK1 (n, chtitl, nx, xmin, xmax, 0.)
         end if
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do

c     ------------------------------------------------------------------
      end

*********************************************************************



c     ==================================================================
c     Fill H-BOOK spectra with FERA data of all cells
c     ==================================================================

      subroutine fhfedat

c     ------------------------------------------------------------------
c     declarations
c     ------------------------------------------------------------------

c --> H-Book Histogram Parameters
      common /chbprm/
     +   hindex, chtitl, nx, xmin, xmax, ny, ymin, ymax
      character*40
     +   chtitl
      integer*4
     +   hindex, nx, ny
      real*4
     +   xmin, xmax, ymin, ymax
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> fera data evaluation parameters
      common /cfeprm/
     +   fepl, fewi,
     +   felcut, feucut, fenorm, feback
      integer*2
     +   fepl(60), fewi(60),
     +   felcut(5,12), feucut(5,12), feback(2)
      real*4
```

```
     +   fenorm(5,12)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> sorted fera data
      common /cfesrt/
     +   fera
      integer*2
     +   fera(5,12)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local:
      integer*4
     +   pl, wi, n
      real*4
     +   bg, x

c     ----------------------------------------------------------------
c     calculations
c     ----------------------------------------------------------------

c --> determine background
      pl = feback(1)
      wi = feback(2)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      if (
     +   (pl.gt.0)
     +   .and.
     +   (wi.gt.0)
     +   ) then
         bg = fera(pl,wi)*fenorm(pl,wi)
      else
         bg = 0.
      end if
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      do pl = 1, 5, 1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         do wi = 1, 12, 1
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            if (
     +         (fera(pl,wi).ge.felcut(pl,wi))
     +         .and.
     +         (fera(pl,wi).le.feucut(pl,wi))
     +         ) then
               n = hindex*100*pl*wi
               x = (fera(pl,wi)*fenorm(pl,wi))-bg
               call HF1 (n, x, 1.)
            end if
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         end do
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do

c     ----------------------------------------------------------------
      end

*********************************************************************

c     ================================================================
c     Determine maximum FERA anode signal in each plane and
c     fill H-BOOK histograms
c     ================================================================

      subroutine fhfemax

c     ----------------------------------------------------------------
c     declarations
```

```
c     ------------------------------------------------------------

c --> H-BOOK histogram parameters
      common /chbprm/
     +   hindex, chtitl, nx, xmin, xmax, ny, ymin, ymax
      character*40
     +   chtitl
      integer*4
     +   hindex, nx, ny
      real*4
     +   xmin, xmax, ymin, ymax
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> fera data evaluation parameters
      common /cfeprm/
     +   fepl, fewi,
     +   felcut, feucut, fenorm, feback
      integer*2
     +   fepl(60), fewi(60),
     +   felcut(5,12), feucut(5,12), feback(2)
      real*4
     +   fenorm(5,12)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> sorted fera data
      common /cfesrt/
     +   fera
      integer*2
     +   fera(5,12)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local:
      integer*4
     +   mxwi, pl, wi, cnt, n
      real*4
     +   bg, mxco, nrm, x

c     ------------------------------------------------------------
c     calculations
c     ------------------------------------------------------------

c --> determine background
      pl = feback(1)
      wi = feback(2)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      if (
     +   (pl.gt.0)
     +   .and.
     +   (wi.gt.0)
     +   ) then
         bg = fera(pl,wi)*fenorm(pl,wi)
      else
         bg = 0.
      end if
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      do pl = 1, 5, 1
c -----> clear wire and value of maximum signal in plane
         mxwi = -1
         mxco = 0.
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c -----> determine maximum ANODE signal in each plane
         do wi = 2, 12, 2
            cnt = fera(pl,wi)
            nrm = cnt*fenorm(pl,wi)
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            if (
     +         (cnt.ge.felcut(pl,wi))
```

```
     +          .and.
     +          (cnt.le.feucut(pl,wi))
     +          .and.
     +          (nrm.gt.mxco)
     +          ) then
                mxco = nrm
                mxwi = wi
             end if
c            - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
          end do
c         - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c -----> fill histogram with evaluated signal
          if (.not.(pl.eq.1.and.mxwi.eq.4)) then
          do i = -1, 1, 1
             wi = mxwi+i
             if (wi.le.12) then
                n = hindex+100*pl+wi
                x = (fera(pl,wi)*fenorm(pl,wi))-bg
                call HF1 (n,x,1.)
             end if
          end do
          end if
c         - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do

c     ------------------------------------------------------------------
      end


*********************************************************************

c     ==================================================================
c     Determine FERA dP/A signal for all cells
c     and fill H-BOOK histograms
c     ==================================================================

      subroutine fhfedpa

c     ------------------------------------------------------------------
c     declarations
c     ------------------------------------------------------------------

c --> h-book histogram parameters
      common /chbprm/
     +   hindex, chtitl, nx, xmin, xmax, ny, ymin, ymax
      character*40
     +   chtitl
      integer*4
     +   hindex, nx, ny
      real*4
     +   xmin, xmax, ymin, ymax
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> fera data evaluation parameters
      common /cfeprm/
     +   fepl, fewi,
     +   felcut, feucut, fenorm, feback
      integer*2
     +   fepl(60), fewi(60),
     +   felcut(5,12), feucut(5,12), feback(2)
      real*4
     +   fenorm(5,12)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> sorted fera data
      common /cfesrt/
```

```
     +   fera
      integer*2
     +   fera(5,12)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local:
      integer*4
     +   pl, wi, n
      real*4
     +   bg, x, y

c     ---------------------------------------------------------------
c     calculations
c     ---------------------------------------------------------------

c --> determine background
      pl = feback(1)
      wi = feback(2)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      if (
     +   (pl.gt.0)
     +   .and.
     +   (wi.gt.0)
     +   ) then
         bg = fera(pl,wi)*fenorm(pl,wi)
      else
         bg = 0.
      end if
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      do pl = 1, 5, 1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         do wi = 2, 10, 2
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --------> fill histogram with dP/A-signal
            if (
     +         (fera(pl,wi-2).le.feucut(pl,wi-2))
     +         .and.
     +         (fera(pl,wi-1).ge.felcut(pl,wi-1))
     +         .and.
     +         (fera(pl,wi).ge.felcut(pl,wi))
     +         .and.
     +         (fera(pl,wi+1).ge.felcut(pl,wi+1))
     +         .and.
     +         (fera(pl,wi+2).le.feucut(pl,wi+2))
     +         .and.
     +         (((fera(pl,wi)*fenorm(pl,wi))-bg).gt.0.)
     +         ) then
               n = hindex+100*pl+wi
               x = ((fera(pl,wi+1)*fenorm(pl,wi+1))
     +            - (fera(pl,wi-1)*fenorm(pl,wi-1)))
     +            / ((fera(pl,wi)*fenorm(pl,wi))-bg)
               call HF1 (n,x,1.)
c              - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c -----------> fill 2-dim. histogram with dP/A-signal
c              (coincidence in two consecutive planes)
               if (
     +            (fera(pl+1,wi-2).le.feucut(pl+1,wi-2))
     +            .and.
     +            (fera(pl+1,wi-1).ge.felcut(pl+1,wi-1))
     +            .and.
     +            (fera(pl+1,wi).ge.felcut(pl+1,wi))
     +            .and.
     +            (fera(pl+1,wi+1).ge.felcut(pl+1,wi+1))
     +            .and.
     +            (fera(pl+1,wi+2).le.feucut(pl+1,wi+2))
```

```
     +            .and.
     +            (((fera(pl+1,wi)*fenorm(pl+1,wi))-bg).gt.0.)
     +            ) then
                  n = 600+hindex+100*pl+wi
                  y = ((fera(pl+1,wi+1)*fenorm(pl+1,wi+1))
     +               - (fera(pl+1,wi-1)*fenorm(pl+1,wi-1)))
     +               / ((fera(pl+1,wi)*fenorm(pl+1,wi))-bg)
                  call HF2 (n,x,y,1.)
               end if
c              - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            end if
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         end do
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do

c     ------------------------------------------------------------------
      end

*********************************************************************

c     ==================================================================
c     Determine FERA dP/A signal for cell with maximum ANODE content
c     in plane and fill H-BOOK histograms
c     ==================================================================

      subroutine fhfedpm

c     ------------------------------------------------------------------
c     declarations
c     ------------------------------------------------------------------

c --> h-book histogram parameters
      common /chbprm/
     +   hindex, chtitl, nx, xmin, xmax, ny, ymin, ymax
      character*40
     +   chtitl
      integer*4
     +   hindex, nx, ny
      real*4
     +   xmin, xmax, ymin, ymax
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> fera data evaluation parameters
      common /cfeprm/
     +   fepl, fewi,
     +   felcut, feucut, fenorm, feback
      integer*2
     +   fepl(60), fewi(60),
     +   felcut(5,12), feucut(5,12), feback(2)
      real*4
     +   fenorm(5,12)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> sorted fera data
      common /cfesrt/
     +   fera
      integer*2
     +   fera(5,12)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local:
      integer*4
     +   mxwi, pl, wi, cnt, n
      real*4
     +   bg, mxco, nrm, x
```

```
c     ------------------------------------------------------------
c     calculations
c     ------------------------------------------------------------

c --> determine background
      pl = feback(1)
      wi = feback(2)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      if (
     +   (pl.gt.0)
     +   .and.
     +   (wi.gt.0)
     +   ) then
         bg = fera(pl,wi)*fenorm(pl,wi)
      else
         bg = 0.
      end if
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      do pl = 1, 5, 1
c -----> clear wire and value of maximum signal in plane
         mxwi = -1
         mxco = 0.
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c -----> determine maximum ANODE signal in each plane
         do wi = 2, 10, 2
            cnt = fera(pl,wi)
            nrm = cnt*fenorm(pl,wi)
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            if (
     +         (cnt.ge.felcut(pl,wi))
     +         .and.
     +         (cnt.le.feucut(pl,wi))
     +         .and.
     +         (nrm.gt.mxco)
     +         ) then
               mxco = nrm
               mxwi = wi
            end if
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         end do
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c -----> fill histogram with dP/A signal of cell with max. anode signal
         if (((fera(pl,mxwi)*fenorm(pl,mxwi))-bg).gt.0.) then
            n = hindex*100*pl+wi
            x = ((fera(pl,mxwi+1)*fenorm(pl,mxwi+1))
     +         - (fera(pl,mxwi-1)*fenorm(pl,mxwi-1)))
     +         / (fera(pl,mxwi)*fenorm(pl,mxwi)-bg)
            call HF1 (n,x,1.)
         end if
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do

c     ------------------------------------------------------------
      end

*********************************************************************

c     ============================================================
c     Determine maximum FERA anode signal in each plane and
c     fill H-BOOK hit-plots
c     ============================================================

      subroutine fhfehpl
```

```
c     ------------------------------------------------------------
c     declarations
c     ------------------------------------------------------------

c --> h-book histogram parameters
      common /chbprm/
     +   hindex, chtitl, nx, xmin, xmax, ny, ymin, ymax
      character*40
     +   chtitl
      integer*4
     +   hindex, nx, ny
      real*4
     +   xmin, xmax, ymin, ymax
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> fera data evaluation parameters
      common /cfeprm/
     +   fepl, fewi,
     +   felcut, feucut, fenorm, feback
      integer*2
     +   fepl(60), fewi(60),
     +   felcut(5,12), feucut(5,12), feback(2)
      real*4
     +   fenorm(5,12)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> sorted fera data
      common /cfesrt/
     +   fera
      integer*2
     +   fera(5,12)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local:
      integer*4
     +   mxwi, pl, wi, cnt, n
      real*4
     +   mxco, nrm, x

c     ------------------------------------------------------------
c     calculations
c     ------------------------------------------------------------

c --> clear wire and value of maximum signal in plane
      mxwi = -1
      mxco = 0.
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      do pl = 1, 5, 1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c -----> determine maximum ANODE signal in each plane
         do wi = 2, 12, 2
            cnt = fera(pl,wi)
            nrm = cnt*fenorm(pl,wi)
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            if (
     +         (cnt.ge.felcut(pl,wi))
     +         .and.
     +         (cnt.le.feucut(pl,wi))
     +         .and.
     +         (nrm.gt.mxco)
     +         ) then
               mxco = nrm
               mxwi = wi
            end if
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         end do
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
c -----> fill hit-plots
         n = hindex+100*pl
         x = float(mxwi)
         call HF1 (n, x, 1.)
      end do

c     ------------------------------------------------------------------
      end

*********************************************************************

c     ==================================================================
c     sort FADC data
c     ==================================================================

      subroutine fadcsrt

c     ------------------------------------------------------------------
c     declarations
c     ------------------------------------------------------------------

c --> online run-header
      common /conlhd/
     +   time, date,
     +   run, nevent,
     +   nfera, ferath, ntdc, tdcth,
     +   nadc, radc, adcth, nfadc, fadcba,
     +   bsamp, bscan, esamp, escan, msamp, mscan,
     +   anoth, potth, lchan, rchan
      character*10
     +   time, date
      integer*2
     +   run, nevent,
     +   nfera, ferath(48), ntdc, tdcth(48),
     +   nadc, radc, adcth(48), nfadc, fadcba(48),
     +   bsamp, bscan, esamp, escan, msamp, mscan,
     +   anoth, potth, lchan, rchan
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> online event-vector
      common /conlev/
     +   ievent, nentrs, ev
      integer*2
     +   ievent, nentrs, ev(10000)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> fadc data evaluation parameters
      common /cfaprm/
     +   fapl, fawi
      integer*2
     +   fapl(60), fawi(60)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> sorted fadc-data
      common /cfasrt/
     +   fadc
      integer*2
     +   fadc(5,12,512)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local
      integer*4
     +   addr, cnt, hbyte, lbyte
      integer*2
     +   wi, pl, fpt, pt, fold, fnew, fskip, i

c     ------------------------------------------------------------------
c     calculations
```

```
c     ------------------------------------------------------------

c --> Clear Matrix for Sorted Data
      do pl = 1, 5, 1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         do wi = 1, 12, 1
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            do fpt = 1, 512, 1
               fadc(pl,wi,fpt) = 0
            end do
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         end do
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      pt = 1
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> search for start of fadc-data (ev(pt) = -4000)
      do while (
     +   (ev(pt) .ne. -4000)
     +   .and.
     +   (pt.le.nentrs)
     +   )
         pt = pt+1
      end do
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      if (pt.eq.nentrs) then
         write (21,*) 'No FADC data in event ',ievent
      else
c -----> initiate values
         fold = -1
         pt = pt+1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c -----> ev(pt) = -4444: end of fadc-data, return to main program
         do while (.not.((ev(pt).eq.-4444).and.(ev(pt+1).eq.0)))
c --------> ev(pt) = - start-address of hit
            fnew = 2*int((-ev(pt)+lchan)/1024)+1
            addr = -ev(pt)-((fnew-1)*512)
c --------> writing data of first hit in this module: fskip = 0
            if (fnew.ne.fold) fskip = 0
            fpt = fskip+1
c --------> first entry in fadc(*,*,fpt) = startaddress of hit
            fadc(fapl(fnew),fawi(fnew),fpt) = addr
            fadc(fapl(fnew+1),fawi(fnew+1),fpt) = addr
            pt = pt+1
            fpt = fpt+1
c --------> second entry in fadc(*,*,fpt) = no. of entries in hit
            fadc(fapl(fnew),fawi(fnew),fpt) = ev(pt)
            fadc(fapl(fnew+1),fawi(fnew+1),fpt) = ev(pt)
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --------> decode and fill fadc-content in matrix
            do i = 1, ev(pt), 1
               cnt = ev(pt+i)
               hbyte = ishft(ishft(cnt,16),-24)
               lbyte = ishft(ishft(cnt,24),-24)
               fpt = fpt+1
c -----------> fadc(*,*,fpt) = content of fadc
               fadc(fapl(fnew),fawi(fnew),fpt) = hbyte
               fadc(fapl(fnew+1),fawi(fnew+1),fpt) = lbyte
            end do
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            fpt = fpt+1
c --------> fadc(*,*,fpoint) = -9999: signal end of fadc-event-vector
            fadc(fapl(fnew),fawi(fnew),fpt) = -9999
```

```
            fadc(fapl(fnew+1),fawi(fnew+1),fpt) = -9999
c --------> initiate values for next fadc-hit
            fold = fnew
            fskip = fskip+2+ev(pt)
            pt = pt+ev(pt)+1
         end do
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end if


c     ------------------------------------------------------------------
      end

*********************************************************************

c     ==================================================================
c      List Sorted FADC Data
c     ==================================================================

      subroutine fadclst

c     ------------------------------------------------------------------
c     declarations
c     ------------------------------------------------------------------

c --> online event-vector
      common /conlev/
     +   ievent, nentrs, ev
      integer*2
     +   ievent, nentrs, ev(10000)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> sorted fadc data
      common /cfasrt/
     +   fadc
      integer*2
     +   fadc(5,12,512)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local
      integer*4
     +   pl, wi, lev, pt

c     ------------------------------------------------------------------
c     calculations
c     ------------------------------------------------------------------

      write (21,9000) ievent
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      do pl = 1, 5, 1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         do wi = 1, 12, 1
            lev = fadc(pl,wi,2)
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            if (lev.gt.0) then
               write (21,9001) pl, wi
               write (21,9002) fadc(pl,wi,1)
               write (21,9003) lev
               write (21,9004)
     +            (fadc(pl,wi,pt), pt=3,2+lev,1)
            end if
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         end do
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do
```

```
c     ------------------------------------------------------------------
c     format statements
c     ------------------------------------------------------------------

9000  format (' '/77('=')/22x,'E V E N T - N U M B E R  ',i6/77('='))
9001  format (' '/20('-'),'  P L A N E  ',i1,'  ',5('-'),
     +        '  W I R E  ',i2,'  ',20('-')/)
9002  format (' STARTADDRESS --------> ',i5)
9003  format (' NUMBER OF ENTRIES ---> ',i5)
9004  format (15(2x,i3))

c     ------------------------------------------------------------------
      end

*********************************************************************

c     ==================================================================
c     FADC Hit-Finder
c      - for each FADC-hit the baseline is determined as that FADC
c        content with the highest number of occurences
c      - a threshold for the peakfinder is calculated
c      -
c     ==================================================================

      subroutine fahitfi

c     ------------------------------------------------------------------
c     declarations
c     ------------------------------------------------------------------

c --> sorted fadc data
      common /cfasrt/
     +   fadc
      integer*2
     +   fadc(5,12,512)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> reduced fadc data
      common /cfared/
     +   baseln, thrshld, fared
      integer*4
     +   baseln, thrshld
      integer*2
     +   fared(5,12,512)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local:
      integer*4
     +   pl, wi, rpt, lev, cnt, i, pt,
     +   noc(256), mxo, sum

c     ------------------------------------------------------------------
c     calculations
c     ------------------------------------------------------------------

      do pl = 1, 4, 1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         do wi = 1, 12, 1
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c   ------> clear matrix of reduced FADC data
            do rpt = 1, 512, 1
               fared(pl,wi,rpt) = 0
            end do
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            lev = fadc(pl,wi,2)
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
            if (lev.gt.0) then
c              - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c ----------> clear vector of number of occurences of fadc-content
               do cnt = 1, 256, 1
                 noc(cnt) = 0
               end do
c              - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
               mxo = 0
c              - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c ----------> determine number of occurences for each FADC-content
               do pt = 3, 2+lev, 1
                 cnt = fadc(pl,wi,pt)
                 noc(cnt+1) = noc(cnt+1)+1
               end do
c              - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c ----------> determine maximum number of occurences and baseline
               do cnt = 1, 256, 1
c                 - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  if (noc(cnt).gt.mxo) then
                     mxo = noc(cnt)
                     baseln = cnt-1
                  end if
c                 - - - - - - - - - - - - - - - - - - - - - - - - - - -
               end do
c              - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c ----------> determine threshold
               thrshld = baseln+10
c ----------> reduce FADC data and fill vector
               pt = 3
c              - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c ----------> detect start of hit (five channels above threshold)
               do while (
     +            (pt.lt.2+lev)
     +            .and.(
     +               .not.(
     +                  (fadc(pl,wi,pt+1).gt.thrshld)
     +                  .and.
     +                  (fadc(pl,wi,pt+2).gt.thrshld)
     +                  .and.
     +                  (fadc(pl,wi,pt+3).gt.thrshld)
     +                  .and.
     +                  (fadc(pl,wi,pt+4).gt.thrshld)
     +                  .and.
     +                  (fadc(pl,wi,pt+5).gt.thrshld)
     +                  )
     +               )
     +            )
                  pt = pt+1
               end do
c              - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
               if (pt.lt.lev+2) then
                  fared(pl,wi,2) = fadc(pl,wi,1)+pt-3
                  rpt = 4
                  sum = 0
c                 - - - - - - - - - - - - - - - - - - - - - - - - - - -
c -------------> evaluate data till end of hit
c                 (two channels below threshold)
                  do while (
     +               (pt.lt.2+lev)
     +               .and.(
     +                  .not.(
c    +                     (fadc(pl,wi,pt-1).lt.thrshld)
c    +                     .and.
     +                     (fadc(pl,wi,pt).lt.thrshld)
```

```
     +                    .and.
     +                    (fadc(pl,wi,pt+1).lt.thrshld)
     +                    )
     +                 )
     +              )
                    fared(pl,wi,rpt) = fadc(pl,wi,pt)-baseln
c ----------------> calculate integral content of hit
                    sum = sum+fared(pl,wi,rpt)
                    pt = pt+1
                    rpt = rpt+1
                 end do
c                - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                 fared(pl,wi,rpt) = fadc(pl,wi,pt)-baseln
                 sum = sum+fared(pl,wi,rpt)
                 fared(pl,wi,1) = sum
c -------------> no. of entries in hit
                 fared(pl,wi,3) = rpt-3
              end if
c             - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
           end if
c          - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
        end do
c       - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do

c     ---------------------------------------------------------------
      end

*********************************************************************



c     ================================================================
c      Generate Hitplots of 'Raw' FADC Data Using GR-Software
c     ================================================================

      subroutine faevpla

c     ----------------------------------------------------------------
c     declarations
c     ----------------------------------------------------------------

c --> sorted fadc data
      common /cfasrt/
     +   fadc
      integer*2
     +   fadc(5,12,512)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> reduced fadc data
      common /cfared/
     +   baseln, thrshld, fared
      integer*4
     +   baseln, thrshld
      integer*2
     +   fared(5,12,512)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local
      character*48
     +   tx(14)
      integer*4
     +   pl, wi,
     +   bev, lev, eev,
     +   i, isk, jsk, nkurv
```

```
      real*4
     +   xfadc(1527), yfadc(1527),
     +   xthr(2), ythr(2), xhit(4), yhit(4),
     +   xcm, ycm, xmin, xmax, ymin, ymax

c     ------------------------------------------------------------
c     calculations
c     ------------------------------------------------------------

      do pl = 1, 5, 1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         do wi = 1, 12, 1
            bev = fadc(pl,wi,1)
            lev = fadc(pl,wi,2)
            eev = bev+lev-1
c           - - - - - - - - - - - - - - - - - - - - - - - - - - -
            if (lev.gt.0) then
c              - - - - - - - - - - - - - - - - - - - - - - - - - -
c ----------> draw histogram of FADC contents
               do i = 0, lev-1, 1
                  xfadc(3*i+1) = float(bev+i)-0.5
                  yfadc(3*i+1) = 0.
                  xfadc(3*i+2) = float(bev+i)-0.5
                  yfadc(3*i+2) = 1.* fadc(pl,wi,3+i)
                  xfadc(3*i+3) = float(bev+i)+0.5
                  yfadc(3*i+3) = 1.* fadc(pl,wi,3+i)
               end do
c              - - - - - - - - - - - - - - - - - - - - - - - - - -
               xfadc(3*lev+1) = float(eev)+0.5
               yfadc(3*lev+1) = 0.
               call KURVEF (xfadc,yfadc,3*lev+1,14)
c              - - - - - - - - - - - - - - - - - - - - - - - - - -
c ----------> draw threshold for software peakfinder
               xthr(1) = float(bev)-0.5
               xthr(2) = float(eev)+0.5
               ythr(1) = thrshld
               ythr(2) = thrshld
               call KURVEF (xthr,ythr,2,114)
c              - - - - - - - - - - - - - - - - - - - - - - - - - -
c ----------> draw start/stop of software hitfinder
               xhit(1) = 1.*fared(pl,wi,2)-0.5
               yhit(1) = 0.
               xhit(2) = xhit(1)
               yhit(2) = 255.
               xhit(3) = 1.*(fared(pl,wi,2)+fared(pl,wi,3))-0.5
               yhit(3) = 255.
               xhit(4) = xhit(3)
               yhit(4) = 0.
               call KURVEF (xhit,yhit,4,14)
c              - - - - - - - - - - - - - - - - - - - - - - - - - -
               xcm = -36.
               ycm = -25.
               isk = 5
               jsk = 5
               xmin = float(bev)-0.5
               xmax = float(eev)+0.5
               ymin = 0.
               ymax = 255.
               nkurv = -1
               call GRBLD (xcm,ycm,isk,jsk,xmin,xmax,ymin,ymax,nkurv)
c              - - - - - - - - - - - - - - - - - - - - - - - - - -
               do i = 1, 14, 1
                  tx(i) = ' '
               end do
```

```
c               - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                tx(2) = 'FADC-Channel (Time/10ns)'
                tx(7) = '$Threshold for Software Peakfinder'
                tx(9) = '$FADC-Content'
                write (tx(14),9000) pl, wi
c               call GRLGND (tx)
                call GRNXTF
             endif
c            - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
          end do
c         - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do

c     ----------------------------------------------------------------
c     format statements
c     ----------------------------------------------------------------

9000  format (' FADC - Plane ',i1,', Wire ',i2)

c     ----------------------------------------------------------------
      end

*********************************************************************

c     ================================================================
c      Generate Hitplots of reduced FADC Data Using GR-Software
c     ================================================================

      subroutine faevplr

c     ----------------------------------------------------------------
c     declarations
c     ----------------------------------------------------------------

c --> reduced fadc data
      common /cfared/
     +   baseln, thrshld, fared
      integer*4
     +   baseln, thrshld
      integer*2
     +   fared(5,12,512)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local
      character*48
     +   tx(14)
      integer*4
     +   pl, wi,
     +   bev, lev, eev,
     +   i, isk, jsk, nkurv
      real*4
     +   xfadc(1527), yfadc(1527),
     +   xcm, ycm, xmin, xmax, ymin, ymax

c     ----------------------------------------------------------------
c     calculations
c     ----------------------------------------------------------------

      do pl = 1, 5, 1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         do wi = 1, 12, 1
            bev = fared(pl,wi,2)
            lev = fared(pl,wi,3)
            eev = bev+lev-1
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
            if (lev.gt.0) then
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c ----------> draw histogram of FADC contents
               do i = 0, lev-1, 1
                  xfadc(3*i+1) = float(bev+i)-0.5
                  yfadc(3*i+1) = 0.
                  xfadc(3*i+2) = float(bev+i)-0.5
                  yfadc(3*i+2) = 1.*fared(pl,wi,4+i)
                  xfadc(3*i+3) = float(bev+i)+0.5
                  yfadc(3*i+3) = 1.*fared(pl,wi,4+i)
               end do
c              - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
               xfadc(3*lev+1) = float(eev)+0.5
               yfadc(3*lev+1) = 0.
               call KURVEF (xfadc,yfadc,3*lev+1,14)
c              - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
               xcm = 36.
               ycm = 25.
               isk = 5
               jsk = 5
               xmin = float(bev)-0.5
               xmax = float(eev)+0.5
               ymin = 0.
               ymax = 255.
               nkurv = -1
               call GRBLD (xcm,ycm,isk,jsk,xmin,xmax,ymin,ymax,nkurv)
c              - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
               do i = 1, 14, 1
                  tx(i) = ' '
               end do
c              - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
               tx(2) = 'FADC-Channel (Time/10ns)'
               tx(9) = '$FADC-Content'
               write (tx(13),9000) pl, wi
               write (tx(14),9010) fared(pl,wi,1)
c              call GRLGND (tx)
               call GRNXTF
            endif
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         end do
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do

c     ------------------------------------------------------------------
c     format statements
c     ------------------------------------------------------------------

9000  format (' FADC - Plane ',i1,', Wire ',i2)
9010  format (' Integral Content of Hit = ',i4)

c     ------------------------------------------------------------------
      end

*********************************************************************

c     ==================================================================
c     Fill H-BOOK histograms with FADC data (integrated signal)
c     ==================================================================

      subroutine fhfadat

c     ------------------------------------------------------------------
c --> declarations
c     ------------------------------------------------------------------
```

```
c --> h-book histogram parameters
      common /chbprm/
     +   hindex, chtitl, nx, xmin, xmax, ny, ymin, ymax
      character*40
     +   chtitl
      integer*4
     +   hindex, nx, ny
      real*4
     +   xmin, xmax, ymin, ymax
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> reduced fadc data
      common /cfared/
     +   baseln, thrshld, fared
      integer*4
     +   baseln, thrshld
      integer*2
     +   fared(5,12,512)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local
      integer*4
     +   pl, wi, n
      real*4
     +   x

c     ------------------------------------------------------------------
c --> calculations
c     ------------------------------------------------------------------

      do pl = 1, 5, 1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         do wi = 1, 12, 1
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            if (fared(pl,wi,3).gt.0) then
               n = hindex+100*pl+wi
               x = 1.*fared(pl,wi,1)
               call HF1 (n,x,1.)
            end if
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         end do
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do

c     ------------------------------------------------------------------
      end

*********************************************************************

c     ==================================================================
c     Fill H-BOOK histograms with Length of FADC hits
c     ==================================================================

      subroutine fhfalen

c     ------------------------------------------------------------------
c --> declarations
c     ------------------------------------------------------------------

c --> h-book histogram parameters
      common /chbprm/
     +   hindex, chtitl, nx, xmin, xmax, ny, ymin, ymax
      character*40
     +   chtitl
      integer*4
```

```
     +   hindex, nx, ny
      real*4
     +   xmin, xmax, ymin, ymax
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> reduced fadc data
      common /cfared/
     +   baseln, thrshld, fared
      integer*4
     +   baseln, thrshld
      integer*2
     +   fared(5,12,512)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local
      integer*4
     +   pl, wi, n
      real*4
     +   x

c     ------------------------------------------------------------------
c --> calculations
c     ------------------------------------------------------------------

      do pl = 1, 5, 1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         do wi = 1, 12, 1
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            if (fared(pl,wi,3).gt.0) then
               n = hindex+100*pl+wi
               x = 1.*fared(pl,wi,3)
               call HF1 (n,x,1.)
            end if
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         end do
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do

c     ------------------------------------------------------------------
      end

*********************************************************************

c     ==================================================================
c     Fill H-BOOK histograms with FADC dP/A data (integrated signal)
c     ==================================================================

      subroutine fhfadpa

c     ------------------------------------------------------------------
c --> declarations
c     ------------------------------------------------------------------

c --> h-book histogram parameters
      common /chbprm/
     +   hindex, chtitl, nx, xmin, xmax, ny, ymin, ymax
      character*40
     +   chtitl
      integer*4
     +   hindex, nx, ny
      real*4
     +   xmin, xmax, ymin, ymax
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> reduced fadc data
      common /cfared/
     +   baseln, thrshld, fared
```

```
      integer*4
     +   baseln, thrshld
      integer*2
     +   fared(5,12,512)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local
      integer*4
     +   pl, wi, n
      real*4
     +   x, y

c     ----------------------------------------------------------------
c --> calculations
c     ----------------------------------------------------------------

      do pl = 1, 5, 1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         do wi = 2, 10, 2
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            if (
     +         (fared(pl,wi-1,3).gt.0)
     +         .and.
     +         (fared(pl,wi,3).gt.0)
     +         .and.
     +         (fared(pl,wi+1,3).gt.0)
     +         .and.
     +         (fared(pl,wi,1).gt.0)
     +         ) then
               n = hindex+100*pl+wi
               x = (1.*fared(pl,wi+1,1)-1.*fared(pl,wi-1,1))
     +            /(1.*fared(pl,wi,1))
               call HF1 (n,x,1.)
c              - - - - - - - - - - - - - - - - - - - - - - - - - - - -
               if (
     +            (fared(pl+1,wi-1,3).gt.0)
     +            .and.
     +            (fared(pl+1,wi,3).gt.0)
     +            .and.
     +            (fared(pl+1,wi+1,3).gt.0)
     +            .and.
     +            (fared(pl+1,wi,1).gt.0)
     +            ) then
                  n = hindex+600+100*pl+wi
                  y = (1.*fared(pl+1,wi+1,1)-1.*fared(pl+1,wi-1,1))
     +               /(1.*fared(pl+1,wi,1))
                  call HF2 (n,x,y,1.)
               end if
c              - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            end if
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         end do
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do

c     ----------------------------------------------------------------
      end

*********************************************************************


c     ================================================================
c     Fill H-BOOK histograms with FERA - vs. FADC (integrated) - data
```

```
c     ==================================================================

      subroutine fhffdat

c     ------------------------------------------------------------------
c --> declarations
c     ------------------------------------------------------------------

c --> h-book histogram parameters
      common /chbprm/
     +   hindex, chtitl, nx, xmin, xmax, ny, ymin, ymax
      character*40
     +   chtitl
      integer*4
     +   hindex, nx, ny
      real*4
     +   xmin, xmax, ymin, ymax
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> fera data evaluation parameters
      common /cfeprm/
     +   fepl, fewi,
     +   felcut, feucut, fenorm, feback
      integer*2
     +   fepl(60), fewi(60),
     +   felcut(5,12), feucut(5,12), feback(2)
      real*4
     +   fenorm(5,12)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> sorted fera data
      common /cfesrt/
     +   fera
      integer*2
     +   fera(5,12)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> reduced fadc data
      common /cfared/
     +   baseln, thrshld, fared
      integer*4
     +   baseln, thrshld
      integer*2
     +   fared(5,12,512)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local:
      integer*4
     +   pl, wi, n
      real*4
     +   bg, x, y

c     ------------------------------------------------------------------
c     calculations
c     ------------------------------------------------------------------

c --> determine FERA background
      pl = feback(1)
      wi = feback(2)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      if (
     +   (pl.gt.0)
     +   .and.
     +   (wi.gt.0)
     +   ) then
         bg = fera(pl,wi)*fenorm(pl,wi)
      else
         bg = 0.
```

```
      end if
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      do pl = 1, 5, 1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         do wi = 1, 12, 1
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            if (
     +         (fera(pl,wi).ge.felcut(pl,wi))
     +         .and.
     +         (fera(pl,wi).le.feucut(pl,wi))
     +         .and.
     +         (fared(pl,wi,3).gt.0)
     +         ) then
               n = 600+hindex*100*pl*wi
               x = (1.*fared(pl,wi,1))
               y = (fera(pl,wi)*fenorm(pl,wi))-bg
               call HF2 (n,x,y,1.)
            end if
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         end do
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do

c     ------------------------------------------------------------------
      end

*********************************************************************

c     ==================================================================
c     Fill H-BOOK histograms with dP/A data - FERA vs. FADC (integrated)
c     ==================================================================

      subroutine fhffdpa

c     ------------------------------------------------------------------
c --> declarations
c     ------------------------------------------------------------------

c --> h-book histogram parameters
      common /chbprm/
     +   hindex, chtitl, nx, xmin, xmax, ny, ymin, ymax
      character*40
     +   chtitl
      integer*4
     +   hindex, nx, ny
      real*4
     +   xmin, xmax, ymin, ymax
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> fera data evaluation parameters
      common /cfeprm/
     +   fepl, fewi,
     +   felcut, feucut, fenorm, feback
      integer*2
     +   fepl(60), fewi(60),
     +   felcut(5,12), feucut(5,12), feback(2)
      real*4
     +   fenorm(5,12)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> sorted fera data
      common /cfesrt/
     +   fera
      integer*2
     +   fera(5,12)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
c --> reduced fadc data
      common /cfared/
     +   baseln, thrshld, fared
      integer*4
     +   baseln, thrshld
      integer*2
     +   fared(5,12,512)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --> local:
      integer*4
     +   pl, wi, n
      real*4
     +   bg, x, y

c     ---------------------------------------------------------------
c     calculations
c     ---------------------------------------------------------------

c --> determine FERA background
      pl = feback(1)
      wi = feback(2)
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      if (
     +   (pl.gt.0)
     +   .and.
     +   (wi.gt.0)
     +   ) then
         bg = fera(pl,wi)*fenorm(pl,wi)
      else
         bg = 0.
      end if
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      do pl = 1, 5, 1
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         do wi = 2, 10, 2
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c --------> fill histogram with dP/A-signal
            if (
     +         (fera(pl,wi-2).le.feucut(pl,wi-2))
     +         .and.
     +         (fera(pl,wi-1).ge.felcut(pl,wi-1))
     +         .and.
     +         (fera(pl,wi).ge.felcut(pl,wi))
     +         .and.
     +         (fera(pl,wi+1).ge.felcut(pl,wi+1))
     +         .and.
     +         (fera(pl,wi+2).le.feucut(pl,wi+2))
     +         .and.
     +         (((fera(pl,wi)*fenorm(pl,wi))-bg).gt.0.)
     +         .and.
     +         (fared(pl,wi-1,3).gt.0)
     +         .and.
     +         (fared(pl,wi,3).gt.0)
     +         .and.
     +         (fared(pl,wi+1,3).gt.0)
     +         .and.
     +         (fared(pl,wi,1).gt.0)
     +         ) then
               n = 600+hindex+100*pl+wi
               x = (1.*fared(pl,wi+1,1)-1.*fared(pl,wi-1,1))
     +            /(1.*fared(pl,wi,1))
               y = ((fera(pl,wi+1)*fenorm(pl,wi+1))
     +            - (fera(pl,wi-1)*fenorm(pl,wi-1)))
     +            / ((fera(pl,wi)*fenorm(pl,wi))-bg)
```

```
              call HF2 (n,x,y,1.)
            end if
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
          end do
c         - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      end do

c     ------------------------------------------------------------------
      end

*********************************************************************

c     ==================================================================
c     List Data of 2-dim H-Book Histograms
c     ==================================================================

      subroutine listhbi

c     ------------------------------------------------------------------
c     declarations
c     ------------------------------------------------------------------

c --> local:
      logical
     +   ex, hexist
      character*1
     +   decide
      character*4
     +   chid
      character*80
     +   chtitl
      integer*4
     +   id, nx, ny, nwt, loc
     +   , idvect(60), n, i, j, k
      real*4
     +   xmin, xmax, ymin, ymax, x

c     ------------------------------------------------------------------
c     calculations
c     ------------------------------------------------------------------

c --> List the Identifiers of All 1-dim H-Book Histograms in idvect
      call HID1(idvect, n)
      id = -1
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      do while (id.ne.0)
c -----> read ID of histogram to list (id = 0: return)
10        call obey ('VMFCLEAR')
          write (6,9000)
9000      format (
     +       ' ',60('-')
     +       / ' List of all 1dim Histograms:'
     +       / ' ',60('-')
     +       )
          write (6,9001) (idvect(i), i=1,n)
9001      format (10i6/)
          write (6,9002)
9002      format (
     +       ' ',60('-')
     +       / ' Enter the ID of the Histogram to List [0=none]:'
     +       )
          rewind 5
          read (5,*,err=10,end=10) id
          ex = hexist(id)
```

```
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         if (ex) then
c --------> read histogram parameters
            call HGIVE (id,chtitl,nx,xmin,xmax,ny,ymin,ymax,nwt,loc)
c --------> initiate output-file
            call intconv (id,chid)
            call obey ('FILEDEF 22 DISK HIST'//chid//' OUTPUT A2')
            rewind 22
c --------> write listing to file
            write (22,*) nx
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            do i = 1, nx, 1
               call hix(id,i,x)
               write (22,*) x, hi(id,i)
            end do
c           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         endif
c        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      enddo

c     ------------------------------------------------------------------
      end

*********************************************************************

c     ==================================================================
c     List Data of 2-dim H-Book Histograms
c     ==================================================================

      subroutine listhb2

c     ------------------------------------------------------------------
c     declarations
c     ------------------------------------------------------------------

c --> local:
      logical
     +   ex, hexist
      character*1
     +   decide
      character*4
     +   chid
      character*80
     +   chtitl
      integer*4
     +   id, nx, ny, nwt, loc
     +   , idvect(60), n, i, j, k
      real*4
     +   xmin, xmax, ymin, ymax, x, y

c     ------------------------------------------------------------------
c     calculations
c     ------------------------------------------------------------------

c --> List the Identifiers of All 2-dim H-Book Histograms in idvect
      call HID2 (idvect, n)
      id = -1
c     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      do while (id.ne.0)
c -----> read ID of histogram to list (id = 0: return)
10       call obey ('VMFCLEAR')
         write (6,9000)
9000     format (
     +      ' ',60('-')
```

```
     +      / ' List of 2-dim Histograms:'
     +      / ' ',60('-')
     +      )
        write (6,9001) (idvect(i), i=1,n)
9001    format (10i6/)
        write (6,9002)
9002    format (
     +      ' ',60('-')
     +      / ' Enter the ID of the Histogram to List [0=none]:'
     +      )
        rewind 5
        read (5,*,err=10,end=10) id
        ex = hexist(id)
c       - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
        if (ex) then
c -------> read histogram parameters
           call HGIVE (id,chtitl,nx,xmin,xmax,ny,ymin,ymax,nwt,loc)
c -------> initiate output-file
           call intconv (id,chid)
           call obey ('FILEDEF 22 DISK HIST'//chid//' OUTPUT A2')
           rewind 22
           write (22,*) nx
           write (22,*) ny
c          - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
           do i = 1, nx, 1
c             - - - - - - - - - - - - - - - - - - - - - - - - - - - -
              do j = 1, ny, 1
                 call hijxy(id,i,j,x,y)
                 write (22,*) x, y, hij(id,i,j)
              end do
c             - - - - - - - - - - - - - - - - - - - - - - - - - - - -
           end do
c          - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
        endif
c       - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     enddo

c    ------------------------------------------------------------
     end
```

Es folgt ein Beispiel für das Format, in dem IDEFIX den Runfile einliest:

```
Example for ONLINE data file
============================

Time: 20:21:31
Date: 11-26-89
run number....................   408
no. of events................    20
no. of FERAs.................    32
FERA thresholds..............
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1
no. of TDCs..................     0
TDC thresholds...............

no. of 2249 ADCs.............     0
rate for 2249 ADCs...........     0
2249 ADC thresholds..........
```

```
no. of FADCs..................    24
FADC baselines................
    35    35    35    35    35    35    35    35    35    35    35    35
    35    35    35    35    35    35    35    35    35    35    35
start-address for sampling....     0
start-address for scanning....     0
stop-address for sampling.....  1023
stop-address for scanning..... 12287
mode of sampling..............    43
mode of scanning..............    17
threshold for hitdetector (a).    30
threshold for hitdetector (p).    30
channels read left of hit.....    15
channels read right of hit....    15
***********************************************************************
event number:................     1
no. of entries:..............   349
 -1000    -1   252    -2   213    -3    89    -4   108    -5   115    -6
    -7    16    -8    20    -9   124   -10   119   -11   112   -12   110
   183   -14   314   -15    16   -16    13   -17   202   -18   116   -19
   -20   134   -21   124   -22   116   -23    53   -24    31   -25   124
   109   -27   146   -28   139   -29   141   -30   201   -31    31   -32
 -1111 -4000   -75    37  2312  1545  1545  2057  2312  2312  2054  1541
  1543  1544  2826  7177 21259-32502 29703 23304 15623  9480  4359  2312
  1801  2058  2571  2315  1288  3088  2571  1544  2057  2057  2057  2310
  1800  1799 -1099    35  1544  1543  1030  1540   519   519  1543  1286
  1542  1543  1798  1798 10249 23819 27401 16905  7690  4617  4619  4364
  1801  1800  1799  1801  1028  2573  1032   259  1288  1288  1291  1546
 -3034    61  2053  2055  1800  1289  1543  2054  2310  2055  2313  1801
  1289  1545  1546  2057  1802  1546  1546  1545  1287  1541  1798  1286
  2312  2055  1803  1545  1289  1545  1546  2314  1544  1800  1289  1290
  2309  2564  2568  2824  3081  3081  3084  2313  1545  1801  1290  2570
  2570  3082  2826  2567  2566  2312  2569  2568  2821  2568  2569 -5196
   774  1797  2054  1030  1544  1799  1798  1801  1033  1802  2060     6
  2348  1884  2173  1159  1142  1113  2111  2093  2079  1557  1552  1290
  2317  1544   775  1547  1287  1539  1540  1028  1541  1541  1540  1539
  1281 -6218    35  1028  1031  1030  1540  1799  1544  1290   776   776
  1290  1032  5380 16387 22276 16128  8705  3847  2315  2573  3338  2825
  2309  2309  2053  1540   259  2056  2057  1541  1544  1286  1285  1282
    61  1538  2051  3331  2564  2310  1543  2053  2054  1542  2053  3330
  1285  2309  2309  2049  2306  2054  3333  2052  1286  1540  2053  2055
  2054  2053  2306  2050  1793  1025  1282  2048  2052  1285  1283  1289
  2056  2055  1798  1802  2824  1543  2054  1541  2309  1792  1540  1543
  1802  1800  1797  1797  1542  1799  2567  1798  1285  1285 -4444
***********************************************************************
event number:................     2
no. of entries:..............   388

etc.
```

Zwei Parameter-files werden von IDEFIX benötigt. Von dem File FERA* IFINPUT bzw. FADC* IFINPUT werden Parameter zur Analyse der ADC bzw. FADC-Daten eingelesen, das andere File (IFHBOO* IFINPUT dient zum Initialisieren der H-BOOK-Spektren. Es folgen jeweils ein Beispiel für das Format dieser Files:

```
*********************************************************
*  COMMENT LINES MUST START WITH A '*' IN THE FIRST ROW.  *
*  THERE MUST BE ONE LINE BETWEEN THE LAST COMMENT LINE   *
*  AND THE START OF THE DATA BLOCKS.                      *
```

```
*******************************************************
*
*************************************************************
*  PLANE AND WIRE OF FERA CHANNEL FOR BACKGROUND SUBTRACTION  *
*  NO BACKGROUND SUBTRACTION IF PLANE =0 OR WIRE = 0          *
*************************************************************
*
*        | PLANE|  WIRE|
         +------+------+
         |     0|     0|
*        +------+------+
*
*****************************************************
*  PARAMETERS FOR EVALUATION OF FERA DATA :           *
*   - CHAN: NUMBER OF CHANNEL                         *
*   - PLANE: NUMBER OF PLANE RELATED TO THIS CHANNEL  *
*   - WIRE: NUMBER OF WIRE RELATED TO THIS CHANNEL    *
*   - LCUT: LOWER CUT FOR THIS CHANNEL                *
*   - UCUT: UPPER CUT FOR THIS CHANNEL                *
*   - NORM: NORMALIZATION FACTOR FOR THIS CHANNEL     *
*****************************************************
*
* |  CHAN| PLANE|  WIRE|  LCUT|  UCUT|  NORM|
  +------+------+------+------+------+------+
  |     1|     1|    12|     1|  9999| 1.000|
  |     2|     1|    10|     1|  9999| 1.000|
  |     3|     1|     8|     1|  9999| 1.000|
      :      :      :      :      :      :
  |    59|     0|     0|  9999|     1| 1.000|
  |    60|     0|     0|  9999|     1| 1.000|


*******************************************************
*  COMMENT LINES MUST START WITH A '*' IN THE FIRST ROW.  *
*  THERE MUST BE ONE LINE BETWEEN THE LAST COMMENT LINE   *
*  AND THE START OF THE DATA BLOCKS.                      *
*******************************************************
*****************************************************
*  PARAMETERS FOR INITIATION OF H-BOOK HISTOGRAMS :  *
*****************************************************

   OFFSET FOR HISTOGRAM ID ..|     0|
   TITLE FOR ALL HISTOGRAMS .|FERA - SPECTRA                        |
   NUMBER OF BINS (X)........|   500|
   XMIN .....................|    0.|
   XMAX .....................| 1000.|
   NUMBER OF BINS (Y)........|    40|
   YMIN .....................|    0.|
   YMAX .....................| 2100.|
```

Danksagung

Mein Dank gilt

- Herrn Dr. Eduard Roderburg für die außerordentlich gute und immer fröhliche Betreuung,
- Herrn Prof. Dr. K. Kilian für die Überlassung des Themas und für interessante, auch über das Thema der Arbeit hinausreichende, Diskussionen,
- Herrn Prof. Dr. F. Hinterberger für die Übernahme des Korreferats,
- Herrn Dr. habil. Walter Oelert dafür, daß er mich nach Jülich 'gelockt' hat, für tatkräftige Unterstützung während der gesamten Arbeit und für das hervorragende (Arbeits)klima in der 'Mittelenergiephysik'-Gruppe,
- allen Mitgliedern dieser Gruppe — Guido Decker, Toni Empl, Christoph Lippert, Thomas Sefzick, Dr. Georg Sehl, Ralph Thomas und Michal Ziółkowski (in alphabetischer Reihenfolge) — für die gute Zusammenarbeit und unzählige kleine und große Hilfen,
- Ciara Mullan and Brian o'Gallachoir for their help during the test-run in Hamburg,
- den Betreibern des Teststrahls 21 am DESY für ihre Gastfreundschaft,
- Herrn Prof. Dr. Walenta und seinen Mitarbeitern für die unbürokratische Überlassung der Analogelektronik und das Ausleihen zusätzlicher FADC-Module,
- Herrn Hadamek und seinen Mitarbeitern für die zuverlässige Ausführung der mechanischen Arbeiten,
- Herrn Bojowald für seine Hilfe bei der Entwicklung der Hochspannungs-Steuermodule,
- sowie allen, die ich vergessen habe.

FORSCHUNGSZENTRUM JÜLICH GmbH

KFA

Jül-2420
Januar 1991
ISSN 0366-0885