Untersuchungen zur Datenqualität des Myon-Detektors im H1-Experiment

von
Jan Olzem

Diplomarbeit in Physik

vorgelegt der

Fakultät für Mathematik, Informatik und Naturwissenschaften

der

Rheinisch-Westfälischen Technischen Hochschule Aachen

im November 2000

angefertigt im
I. Physikalischen Institut
Prof. Dr. Ch. Berger

Inhaltsverzeichnis

Kapitel 1

Das H1-Experiment

1.1 Der Speicherring HERA

Die *Hadron-Elektron-Ringanlage* (HERA) am Deutschen Elektronen-Synchrotron DESY in Hamburg wurde in den Jahren 1984 - 1991 erbaut. In den beiden Speicherringen der 6,3 km Umfang messenden Anlage werden Elektronen[1] und Protonen gegenläufig beschleunigt und in zwei Strahlkreuzungspunkten zur Kollision gebracht. Mit den erreichten Strahlenergien von $27,5\ GeV$ der Elektronen und (seit August 1998) $920\ GeV$ der Protonen kann eine Schwerpunktsenergie von $\sqrt{s} \approx 319\ GeV$ erzielt werden, die maximal erreichte Luminosität beträgt dabei $\mathcal{L} = 1,4 \cdot 10^{31} cm^{-2} s^{-1}$ [2].

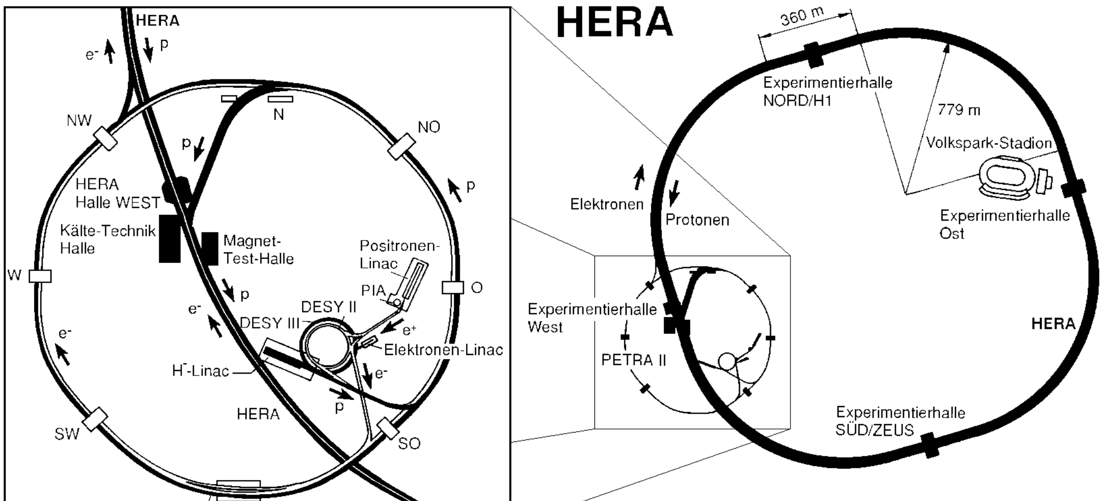

Abbildung 1.1: Schematische Darstellung der *Hadron-Elektron-Ringanlage*. Der Ausschnitt auf der linken Seite zeigt die Anordnung der Vorbeschleuniger.

[1] Aufgrund der höheren Lebensdauer des Strahls werden derzeit Positronen anstelle von Elektronen beschleunigt. Beide Teilchen sind jedoch im folgenden vereinfachend als Elektronen bezeichnet.

Nach einer für die Betriebspause im Winter 2000/2001 geplanten Überarbeitung der Anlage soll die zu erzielende Luminosität auf $\mathcal{L} = 7,5 \cdot 10^{31} cm^{-2}s^{-1}$ gesteigert werden [2], was weit über den ursprünglich avisierten Design-Wert von $\mathcal{L} = 1,5 \cdot 10^{31} cm^{-2}s^{-1}$ hinausreicht.

Die beiden Strahlröhren von HERA führen die Elektronen bzw. Protonen in derzeit je 174 sog. Paketen [2], die in zeitlichem Abstand von 96 ns aufeinanderfolgen, was einem räumlichen Abstand von ca. 29 m entspricht. In den Strahlkreuzungspunkten kommt es somit zu $10,4 \cdot 10^6$ Kollisionen je zweier Pakete pro Sekunde. Die Dauer von 96 ns zwischen je zwei Kollisionen der Pakete stellt im Bereich des H1-Triggersystems eine vielgebrauchte Einheit der Zeit dar und wird nachfolgend als *Zeitscheibe* bezeichnet. So liefert HERA mit der sog. *HERA clock* ein Zeitsignal der Periode 96 ns, mit dem die Auslese der einzelnen Detektoren synchronisiert werden kann.

Vier große Detektoren sind in den Experimentierhallen entlang des Speicherrings untergebracht. Bei zweien davon, dem H1-Experiment in der nördlichen Halle und dem ZEUS-Detektor in der Halle Süd, treffen die Elektronen- und Protonenstrahlen unter einem Winkel von nahezu 180° aufeinander. Sie sind als Vielzweck-Detektoren ausgelegt und dienen hauptsächlich der Untersuchung der Elektron-Proton-Streuung. In der östlichen bzw. westlichen Experimentierhalle befinden sich die Detektoren HERMES und HERA-B, in deren Wechselwirkungszonen Elektronen bzw. Protonen aus den Teilchenstrahlen bzw. deren Halos auf ein feststehendes Target treffen. Der HERMES-Detektor ist spezialisiert auf die Messung von Spinverteilungen der Quarks in den Nukleonen, während HERA-B dem Studium der CP-Verletzung im b-Quark-System dient.

1.2 Das H1-Experiment

1.2.1 Allgemeines

Der Aufbau des H1-Experimentes ist in Abb. 1.2 schematisch wiedergegeben. Sein Design zielt im wesentlichen auf die Identifizierung und Energiemessung des gestreuten Elektrons ab sowie auf eine präzise Vermessung der hadronischen Energieflüsse. Aufgrund der unterschiedlich hohen Energien der in der Wechselwirkungszone aufeinandertreffenden Teilchenstrahlen ist der H1-Detektor asymmetrisch aufgebaut. Zur Beschreibung der Geometrie wird ein Kugelkoordinatensystem verwendet, wobei die z-Achse in Richtung des einlaufenden Protonenstrahls zeigt und ihren Ursprung im nominellen Wechselwirkungspunkt hat. Der Polarwinkel wird mit ϑ, der Azimutwinkel mit ϕ bezeichnet.

Nachfolgend werden die wichtigsten Komponenten des H1-Experimentes kurz aufgeführt; die Zahlen hinter den Stichworten verweisen auf die Legende in Abb. 1.2. Eine ausführliche Beschreibung des Detektors ist in [1] zu finden.

Die Zentralen Spurkammern $\boxed{2}$ (*central tracking chambers*) dienen der präzisen Spur- und Vertexrekonstruktion in der zentralen Region des H1-Detektors. Bei diesen zylinderförmig um das Strahlrohr angeordneten Kammern handelt es sich um eine ganze Reihe von Einzelsystemen, (von innen nach außen) eine Vieldraht-Proportionalkammer (CIP), eine z-Driftkammer (CIZ), eine innere Jetkammer

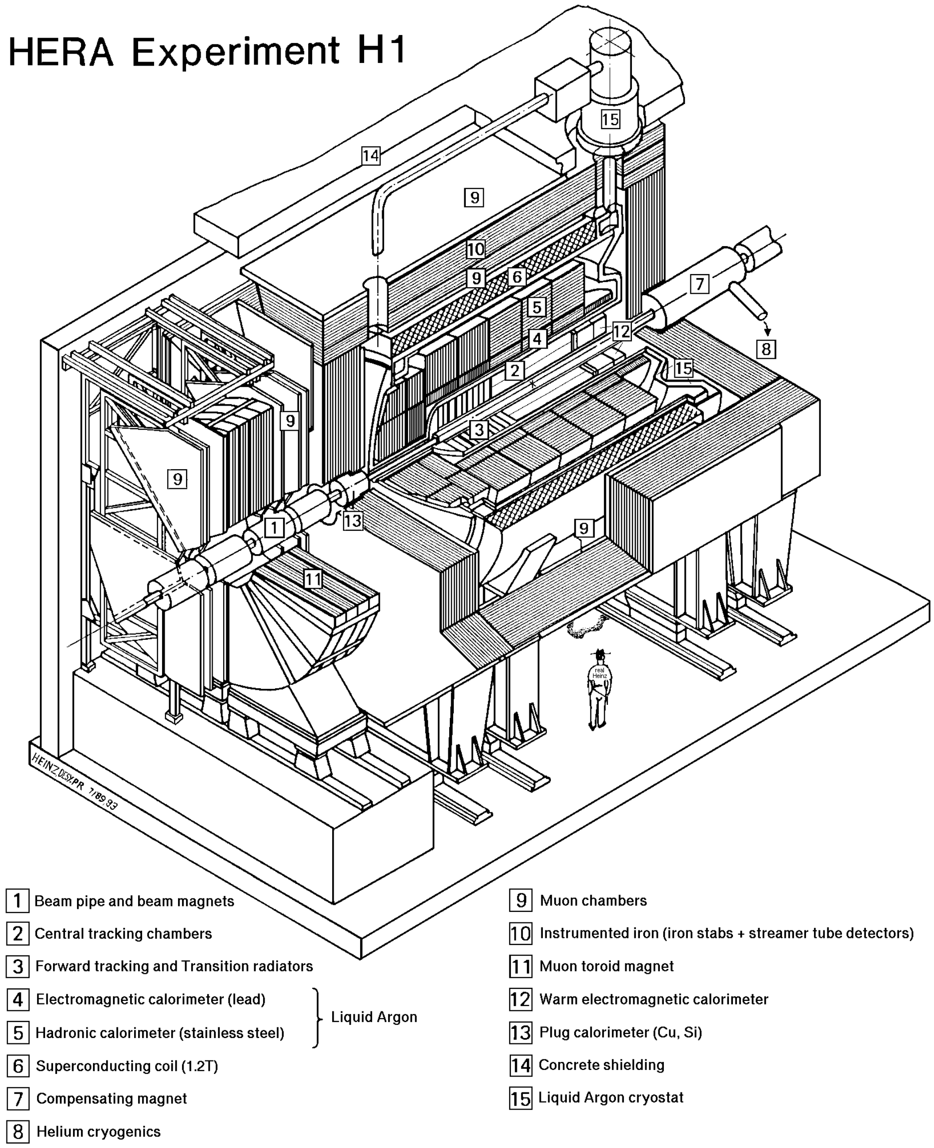

Abbildung 1.2: Der Aufbau des H1-Experimentes im Schnitt.

(CJC1), eine äußere z-Driftkammer (COZ), eine äußere Proportionalkammer (COP) sowie eine äußere Jetkammer (CJC2).

Die beiden z-Driftkammern weisen eine Ortsauflösung in z-Richtung von $\sigma_z \approx 350\ \mu m$ auf und erlauben somit eine sehr genaue Rekonstruktion der z-Position von Spuren und Vertices. Die beiden Jetkammern, aufgebaut aus je 30 Zellen à 24 Signaldrähten, besitzen eine Ortsauflösung von $\sigma_{r\phi} \approx 170\ \mu m$ in der r-ϕ-Ebene [1]. Infolge der Neigung der Zellen von 30° gegenüber dem Radiusvektor wird in beiden Kammern von einer Teilchenspur mindestens je eine der Zellen durchquert; durch einen sorgfältigen Vergleich der Signale beider Kammern während der Rekonstruktion kann die zeitliche Auflösung des zentralen Spurkammersystems auf bis zu $\sigma_t \approx 0,5\ ns$ gesteigert werden [10]. Von diesem Merkmal wird auch in der vorliegenden Analyse Gebrauch gemacht.

Das Flüssig-Argon-Kalorimeter (LAr) umfaßt einen elektromagnetischen $\boxed{4}$ und einen hadronischen Teil $\boxed{5}$. Die elektromagnetische Komponente ist, abhängig vom Polarwinkel, etwa 20 - 30 Strahlungslängen tief. Zwischen 2,4 mm dicken Bleiplatten als Absorbermaterial sind hier Flüssig-Argon-Ionisationskammern von $2,35\ mm$ Stärke installiert, die mit Hilfe flächiger Elektroden ausgelesen werden.

Die hadronische Komponente des LAr umschließt die elektromagnetische, enthält ebenfalls flüssiges Argon als aktiven Bestandteil und weist eine Tiefe von 4,7 bis 7 Absorptionslängen auf. Als Absorbermaterial wird hier allerdings Edelstahl eingesetzt, und zwar in einer wesentlich größeren Menge - die Dicke der Stahlplatten beträgt $19\ mm$ [1].

Die Supraleitende Spule $\boxed{6}$ umschließt wiederum das Kalorimeter und erzeugt im Bereich der Zentralen Spurkammern ein nahezu homogenes Magnetfeld der Stärke $1,2\ T$ [1]. Elektrisch geladene Teilchen, welche eine Bewegungskomponente senkrecht zur Strahlachse aufweisen, werden somit zum Zwecke der Impulsmessung auf eine gekrümmte Bahn gezwungen.

Das Vorwärtige Myon-Spektrometer $\boxed{9}$ befindet sich außerhalb der Spule und des weiter unten beschriebenen Instrumentierten Eisens im vorderen Teil des H1-Experimentes. Für die Bestimmung des Impulses von unter kleinen Polarwinkeln $(3° < \vartheta < 17°)$ entweichenden Myonen besitzt dieser Subdetektor einen eigenen Toroidmagneten $\boxed{11}$.

Vor und hinter dem Toroidmagneten sind jeweils drei großflächige Driftkammermodule installiert, deren Durchmesser von 4 m des ersten bis zu 6 m des letzten Moduls variiert. Jedes Modul besteht aus einer doppelten Lage sog. Driftzellen. Die Orientierung der Driftzellen innerhalb der Module (und damit der in ihnen enthaltenen sensitiven Drähte) ist derart gestaltet, daß zwei der Module im wesentlichen für die Messung des Polarwinkels ϑ prädestiniert sind, während die vier anderen hauptsächlich zur Messung des Azimutwinkels ϕ eingesetzt werden [3].

Die spezielle Geometrie des Vorwärtigen Muon-Spektrometers ermöglicht mit Hilfe einer Driftzeitmessung in den verschiedenen Modulen eine Bestimmung des Eintreffzeitpunktes eines Myons mit einer Genauigkeit von $\sigma_t \approx 5,6\ ns$ [10].

Der Zentrale Myon-Detektor , bestehend aus den Myonboxen $\boxed{9}$ sowie dem Instrumentierten Eisen $\boxed{10}$ wird im nächsten Abschnitt detailliert beschrieben.

1.2.2 Das H1-Triggersystem

Neben den $e^{\pm}p$ - Streuereignissen, die von physikalischem Interesse sind, werden vom Detektor auch eine Vielzahl von Untergrundereignissen registriert. Zu den wichtigsten Quel-

Strahl-Restgas-Wechselwirkung		50 kHz
kosmische μ (Barrel)		700 Hz
Photoproduktion	1,6 μb	25 Hz
$c\bar{c}$ total	1 μb	15 Hz
DIS (kleine Q^2)	150 nb	2,2 Hz
DIS (hohe Q^2, e in LAr)	1,5 nb	1,4 min^{-1}
DIS (*charged current*, $p_T > 25\ GeV$)	50 pb	3,0 h^{-1}
W-Produktion	0,4 pb	0,5 d^{-1}

Tabelle 1.1: Ereignisraten (und Wirkungsquerschnitte) verschiedener Typen von Signal- und Untergrundereignissen bei Design-Luminosität [1].

len dieses Untergrundes zählen die Synchrotronstrahlung der beschleunigten Elektronen, Wechselwirkungen der Protonen mit dem Restgas in der Strahlröhre oder der Strahlröhre selbst sowie Partikel der kosmischen Höhenstrahlung (diese sind hauptsächlich Myonen). Tabelle 1.1 gibt einen Überblick über die Ereignisraten verschiedener Typen von Signal- und Untergrundereignissen bei Design-Luminosität.

Eine vollständige Auslese aller Daten des H1-Detektors benötigt ca. $2,5\ ms$ pro Ereignis; während des Auslesevorgangs ist eine weitere Datennahme unterbunden. Insofern ist es notwendig, physikalisch interessante Ereignisse möglichst in Echtzeit zu identifizieren und von Untergrundereignissen abzutrennen, so daß die Auslese des Detektors im letzteren Falle weitgehend vermieden wird.

Zu diesem Zweck ist die Prozedur der Bewertung eines Ereignisses durch das Triggersystem hierarchisch gegliedert in drei Triggerstufen (*L1, L2, L4*). Jede der drei Triggerstufen, die parallel zur Datennahme (*online*) arbeiten, kann selbständig entscheiden, ob ein Ereignis verworfen oder für die weitere Bearbeitung an die nächsthöhere Stufe übergeben wird. Dies hat zur Folge, daß die Ereignisrate von einer Triggerstufe zur nächsten sukzessive abnimmt und die Komplexität der zur Verfügung stehenden Informationen gleichermaßen ansteigen kann.

Um die Subdetektor-Informationen bis zum Eintreffen der Entscheidung der ersten Triggerstufe zu speichern[2], legen alle Subdetektoren ihre Signale in sog. *pipelines*[3] ab, die mindestens eine Länge von 23 Zeitscheiben aufweisen müssen. Sind die pipelines gestoppt worden, können die Daten darin an einer bekannten Stelle wieder aufgefunden werden.

Trigger Level 1

Die erste Triggerstufe entscheidet synchron zur Datennahme mit der Frequenz der HERA clock ($10,4\ MHz$) und arbeitet somit totzeitfrei. Eine positive Entscheidung (*L1 KEEP*) von L1 führt zum Anhalten der pipelines und zur teilweisen Auslese des Detektors.

Aufgrund der Anforderungen an ihre Geschwindigkeit ist L1 eine reine Hardware-Triggerstufe. Sie bezieht die zur Entscheidung notwendigen Informationen aus einem Satz

[2] Aufgrund der Driftzeiten in den Kammern des LAr in der $\mathcal{O}(1\ \mu s)$ und der Signalwege ist das Eintreffen der Entscheidung von L1 an den Subdetektoren um bestenfalls 23 Zeitscheiben verzögert.

[3] In einer pipeline werden die jeweils ältesten Daten verworfen, um Platz für die neuesten zu schaffen.

von 200 *Triggerelementen*, die von den einzelnen Subdetektoren zur Verfügung gestellt werden. Diese Triggerelemente werden logisch miteinander zu 128 *Subtriggern* verknüpft, von denen jeder einzelne eine positive Entscheidung von L1 herbeiführen kann. Zur Anpassung an die veränderliche Intensität der Teilchenstrahlen während einer Runperiode können diese Subtrigger zusätzlich skaliert werden - ist der Skalierungsfaktor n_m, so wird nur jede n-te Ankunft des Subtriggers m auch validiert.

Trigger Level 2

Hat die Auslese des Detektors nach einem L1 KEEP-Signal begonnen, wird die Bearbeitung des Ereignisses in den beiden nächsten Triggerstufen in Gang gesetzt. Auch L2 benutzt speziell gefertigte Hardware, um eine Entscheidung innerhalb der geforderten Zeit von 20 μs herbeizuführen. Sie bedient sich dazu zweierlei unabhängiger Teilsysteme, des Topologischen Triggers (*L2TT*) sowie eines neuronalen Netzwerkes (*L2NN*), welche beide eine Mustererkennung im hochdimensionalen Raum der an L2 übermittelten Parameter durchführen. Fällt die Entscheidung positiv aus (*L2 KEEP*), so wird die Auslese des Detektors vervollständigt, und die nächste Stufe, L4, kann ihre Arbeit aufnehmen. Eine ursprünglich geplante Triggerstufe L3 existiert derzeit nicht, ist aber für die Wiederaufnahme des Strahlbetriebs nach der Umbauphase im Winter 2000/1 geplant [2].

Trigger Level 4

Die vierte Triggerstufe besteht aus einem cluster von RISC-Prozessoren, die eine teilweise Rekonstruktion der Ereignisse mit Hilfe der Rekonstruktionssoftware H1REC vornehmen. Somit stehen L4 eine Vielzahl physikalisch relevanter Größen zur Verfügung, die eine zielgerichtete Eliminierung von Untergrundereignissen ermöglichen. Wechselwirkungen der Strahlteilchen mit den Molekülen des Restgases oder der Apertur der Strahlführung sind somit wirkungsvoll zu unterdrücken. Insbesondere können zu diesem Zeitpunkt auch fehlerhaft erzeugte Triggersignale erkannt werden, indem L4 die 128 Subtrigger anhand zusätzlicher Informationen (beispielsweise rekonstruierter Energien und Impulse) nachträglich überprüft. Auf diesem Wege entstehen die von L4 *validierten Subtrigger*.

All dies führt zu einer Reduzierung der Rohdatenmenge um ca. 80% [10] und zur Speicherung der verbliebenen Ereignisse auf die Magnetbänder. Etwa 1% der Ereignisse wird in jedem Falle gespeichert zwecks Kontrolle der Leistung von L4. Die Tatsache, daß die Bearbeitungszeit in der vierten Triggerstufe stark von Umfang und Topologie der Ereignisse abhängt, führt auch zu deren teilweise nicht chronologischer Reihenfolge auf den Bändern.

1.3 Der Zentrale Myon-Detektor im H1-Experiment

Myonen mit einem Impuls ab etwa 1 bis 2 GeV/c, je nach Polarwinkel, können das Kalorimeter und die Spule durchdringen und den Zentralen Myon-Detektor erreichen. Dieser ist der am weitesten außen gelegene hermetische Detektor des H1-Experimentes. Er gliedert sich räumlich in vier Subdetektoren, die zylindrisch um die Strahlröhre angeordneten

Sektor	Module
BIEC	6 bis 11
BOEC	0 bis 5, 12 bis 15
BUBA	16 bis 24
BLBA	25 bis 31

Sektor	Module
FUBA	32 bis 40
FLBA	41 bis 47
FOEC	48 bis 53, 60 bis 63
FIEC	54 bis 59

Tabelle 1.2: Die Definition der acht Sektoren mit den zugehörigen Modulen [10].

forward barrel (*FBA*) und *backward barrel* (*BBA*) sowie die beiden Endkappen *forward endcap* (*FEC*) und *backward endcap* (*BEC*). Jeder der Subdetektoren ist wiederum in 16 *Module* segmentiert, die zwar unterschiedliche Abmessungen aufweisen, aber alle einen äquivalenten Aufbau zeigen (s. Abschnitt 1.3.1).[4]

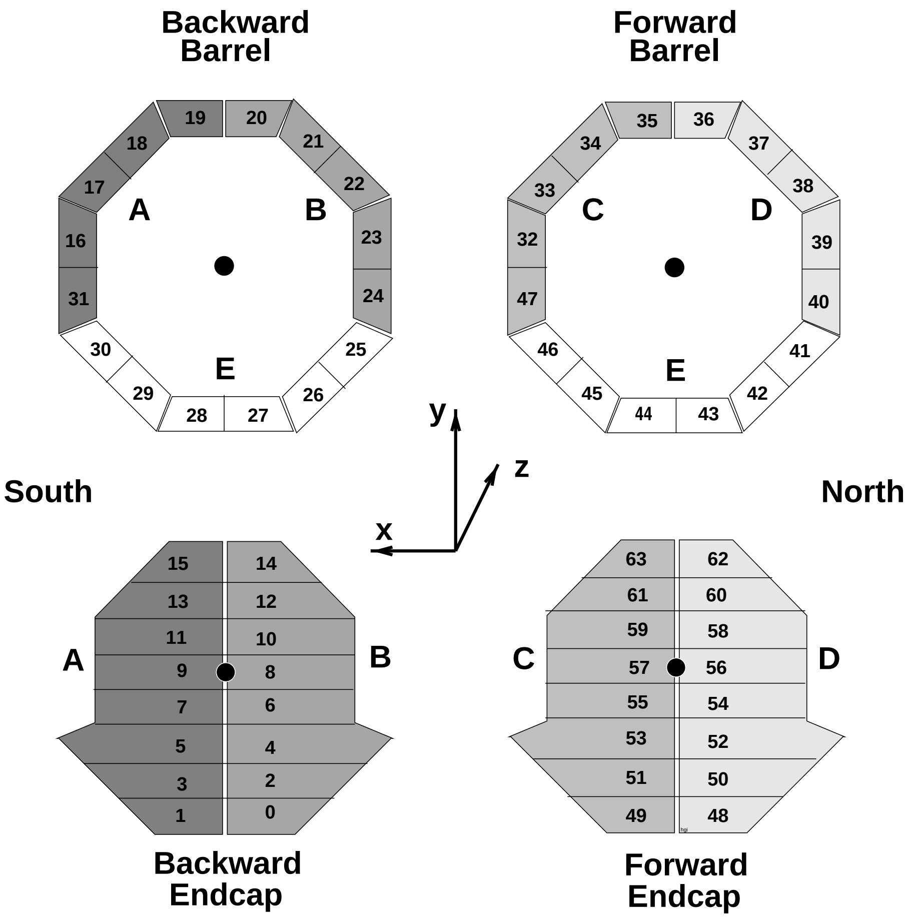

Abbildung 1.3: Anordnung der vierundsechzig Module des Zentralen Myon-Detektors und ihre Zugehörigkeit zu den logical hardware cluster A bis E.

[4]Eine weitere, unabhängige Unterteilung des Zentralen Myon-Detektors in die fünf *logical hardware cluster* A bis E ist technischer Natur (vergl. Abb. 1.3).

1.3.1 Das Instrumentierte Eisen

Die sensitiven Bauteile des Zentralen Myon-Detektors sind zwischen den massiven, 7,5 cm starken Stahlplatten des Magnetjochs installiert, welches zur Rückführung des magnetischen Flusses der supraleitenden Spule dient. Es gibt insgesamt pro Modul maximal 16 *Lagen* solcher Bauteile, von denen 9 zwischen den Stahlplatten und jeweils drei unterhalb und oberhalb eines Moduls angebracht sind. Die außerhalb des Eisenjochs installierten Lagen sind in aus Aluminium gefertigten Kästen untergebracht, den sog. *Myonboxen*. Jedoch sind nicht alle Module sowohl auf der Ober- als auch auf der Unterseite mit Myonboxen ausgestattet. Die logische Numerierung der Lagen, die für die spätere Diskussion wichtig ist, beginnt bei null und jeweils an der Innenseite eines Moduls. Sie ist in der Abb. 1.4 dargestellt und bleibt auch dann gültig, wenn innere oder äußere Myonboxen nicht vorhanden sind. Um die Spuren von Myonen in allen drei Dimensionen vermessen zu können, werden zwei verschiedene, miteinander gekoppelte Typen sensitiver Bauteile verwendet, Drahtkammern und Streifenelektroden.

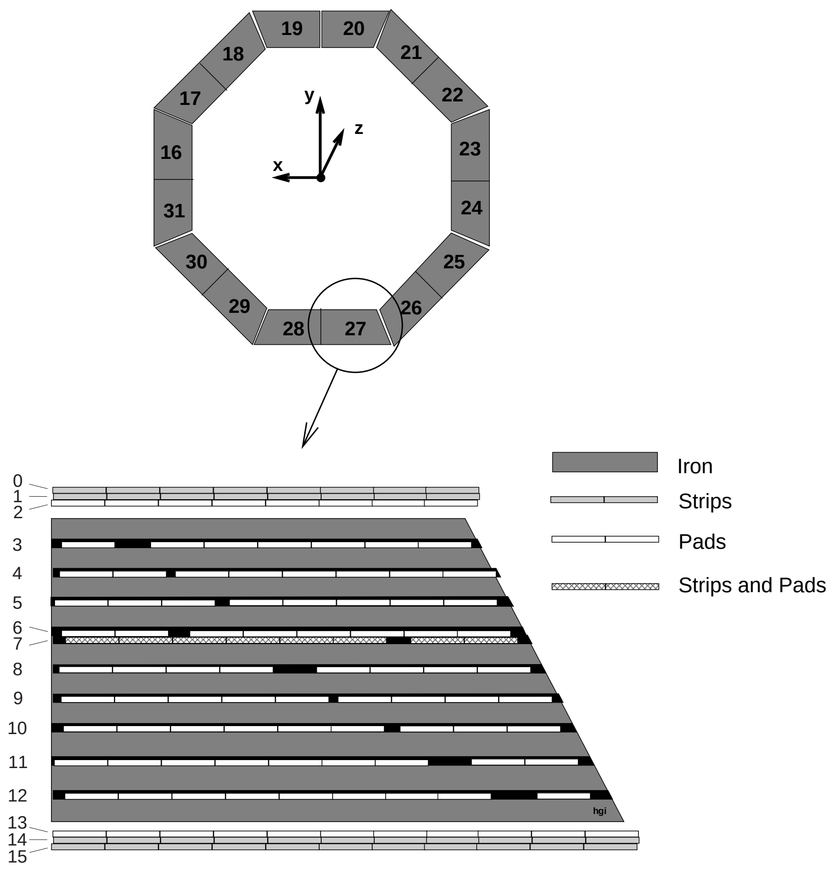

Abbildung 1.4: Schnitt durch ein Modul des rückwärtigen Barrelbereiches. Es sind die 16 Lagen sensitiver Bauteile zu erkennen, welche von 0 bis 15 numeriert sind und alle auch Drahtkammer-Elemente beinhalten. Die Legende verweist lediglich auf die zusätzliche Instrumentierung der Lagen mit pads und Streifenelektroden.

Der Zentrale Myon-Detektor besteht im wesentlichen aus den 103.000 *Drähten*, die eingebettet sind in sog. *Iarocci-Streamerrohrkammern* oder *Drahtkammern*. Bei diesen handelt es sich um mit einer dreikomponentigen Gasmischung gefüllte und aus Kunststoff bestehende Röhren, die eine quadratische Querschnittsfläche von $10 \times 10\ mm^2$ besitzen und an der Innenseite mit einer Graphitmasse niedrigen elektrischen Widerstandes beschichtet sind. Im Innern verläuft jeweils der dünne Kupfer-Beryllium-Draht als geerdete Anode. An der als Kathode ausgebildeten Wand einer Drahtkammer liegt während des Betriebs eine Spannung von ca. $-4,5\ kV$ an [5]. Jeweils acht der Drahtkammern bilden zusammen ein knapp 8 cm breites sog. *Profil*, welches dann den grundlegenden Baustein des Instrumentierten Eisens darstellt. Die Profile werden paarweise zu einem sog. *Element* zusammengefaßt und in einer Kunststoffhülle gasdicht verschlossen (s. Abb.1.5).

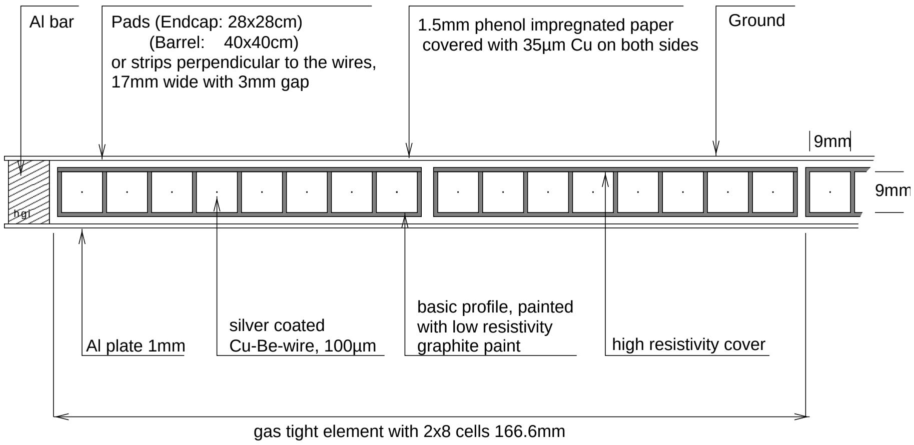

Abbildung 1.5: Schnitt durch ein Drahtkammer-Element.

Im übrigen weisen die Lagen innerhalb eines Moduls häufig eine unterschiedliche Breite auf. Da sie nur mit ganzen Elementen bestückt sind, ergeben sich nicht instrumentierte Bereiche (*dummies*). Diese sind dergestalt über die Lagen verteilt, daß keine Gerade durch den nominellen Wechselwirkungspunkt mehrere von ihnen durchquert [4].

Die dritte Dimension wird mit Hilfe der 28700 *Streifenelektroden* erschlossen, die in den beiden innersten und äußersten Lagen (sowie in Lage 7) senkrecht zu den Drähten auf die Elemente geklebt sind und dabei eine Breite von je 17 mm bedecken. Die Kunststoffabdeckung der Elemente besitzt einen hohen elektrischen Widerstand und erlaubt die Influenzwirkung auf die Kupferplatten der Elektroden. Deren Signale werden dann digital verarbeitet.

Damit der Zentrale Myon-Detektor auch als *Kalorimeter* fungieren kann, befinden sich auf jeder Lage eines Moduls großflächige Ausführungen der Streifenelektroden, die sog. *pads*, die allerdings im Gegensatz zu den Drähten und Streifenelektroden analog ausgelesen werden. Im Bereich der Kalorimetrie ist der Zentrale Myon-Detektor auf die Vermessung der Energien von aus dem Flüssig-Argon-Kalorimeter entweichenden Hadronen beschränkt und verbessert damit die Energieauflösung des hadronischen Kalorimeters. Es werden die differentiellen analogen Signale der sog. *Türme* (engl. *towers*, die pads der jeweils innersten fünf bzw. äußersten sechs Lagen) aufsummiert, verstärkt und anschließend integriert. Mit diesem Verfahren erreicht man eine Energieauflösung in der Größenordnung von $\sigma_E/E \approx 100\%/\sqrt{E/GeV}$ [6]. Die kalorimetrische Funktion der analog ausgelesenen pads ist jedoch nicht Gegenstand der Untersuchungen im Rahmen der vorliegenden Arbeit.

1.3.2 Der Readout

Die Drahtkammern und Streifenelektroden des Zentralen Myon-Detektors werden digital ausgelesen. Alle Kanäle eines Elementes sind direkt mit den sog. *wire/strip digital module boards* (*WDMB/SDMB*), der *unmittelbar auslesenden Elektronik*, verbunden. Die analogen Signale werden zunächst in einer Komparatorschaltung als solche identifiziert, digitalisiert und anschließend in einer 32 Zeitscheiben tiefen pipeline untergebracht. Die Synchronisation der Ausgabe des Komparators findet dabei durch Vergleich mit der HERA clock statt.

Im Falle einer positiven Entscheidung der ersten Triggerstufe werden die pipelines angehalten, und der *readout controller* (*ROC*) leitet deren Auslese ein. Jeder readout controller sendet zunächst über einen seriellen Bus eine Folge von Steuerimpulsen an die unmittelbar auslesende Elektronik, welche die pipelines weiterschalten, bis deren gewünschter Eintrag an der letzten Stelle liegt und dort ausgelesen werden kann. Ist die Auslese vollständig, kann durch Aussenden weiterer Steuerimpulse der jeweils nächste Eintrag der pipelines abgefragt werden. Auf diese Weise geschieht derzeit die Auslese von vier Zeitscheiben pro Ereignis (-1, 0, 1, 2).

1.3.3 Der Myon-Trigger

In Bezug auf den Myon-Trigger sind die Subdetektoren BEC, BBA, FBA und FEC des Zentralen Myon-Detektors jeweils weiterhin dividiert in einen oberen und unteren Bereich, *forward upper/lower barrel* (*FUBA/FLBA*) und *backward upper/lower barrel* (*BUBA/BLBA*), oder einen inneren und äußeren Bereich, *forward inner/outer endcap* (*FIEC/FOEC*) und *backward inner/outer endcap* (*BIEC/BOEC*). Diese werden im folgenden als *Sektoren* bezeichnet (vergl. Tab. 1.2).

Die Triggerelemente mit den logischen Nummern 56 bis 63 werden vom Zentralen Myon-Detektor an die Triggerstufe L1 geliefert. Dessen unmittelbare Ausleseelektronik wertet zu diesem Zweck die digitalen Signale der Drahtkammer-Elemente der einzelnen Module aus. Nicht alle Lagen werden hierbei berücksichtigt; lediglich drei der inneren (Lagen 3, 4 und 5) sowie zwei der äußeren (Lagen 8 und 12) können mit dem Myon-Trigger verbunden sein, was von der Position des Moduls im Detektor abhängig ist. Hierauf wird weiter unten noch genauer eingegangen. Die Signale, die zur Erzeugung der Triggerelemente benötigt werden, stammen aus den jeweils ersten Einträgen der pipelines der unmittelbar auslesenden Elektronik, stehen also nach sehr kurzer Zeit bereits zur Verfügung. Entscheidend für die vorliegende Analyse ist dabei die Tatsache, daß diese Quellsignale bereits einmal mit der HERA clock synchronisiert wurden.

Nach Bildung eines logischen ODER aller acht Signale eines Profils gelangen die beiden resultierenden Bits pro Element im nächsten Schritt in die sog. *layer boards* (*LB*).

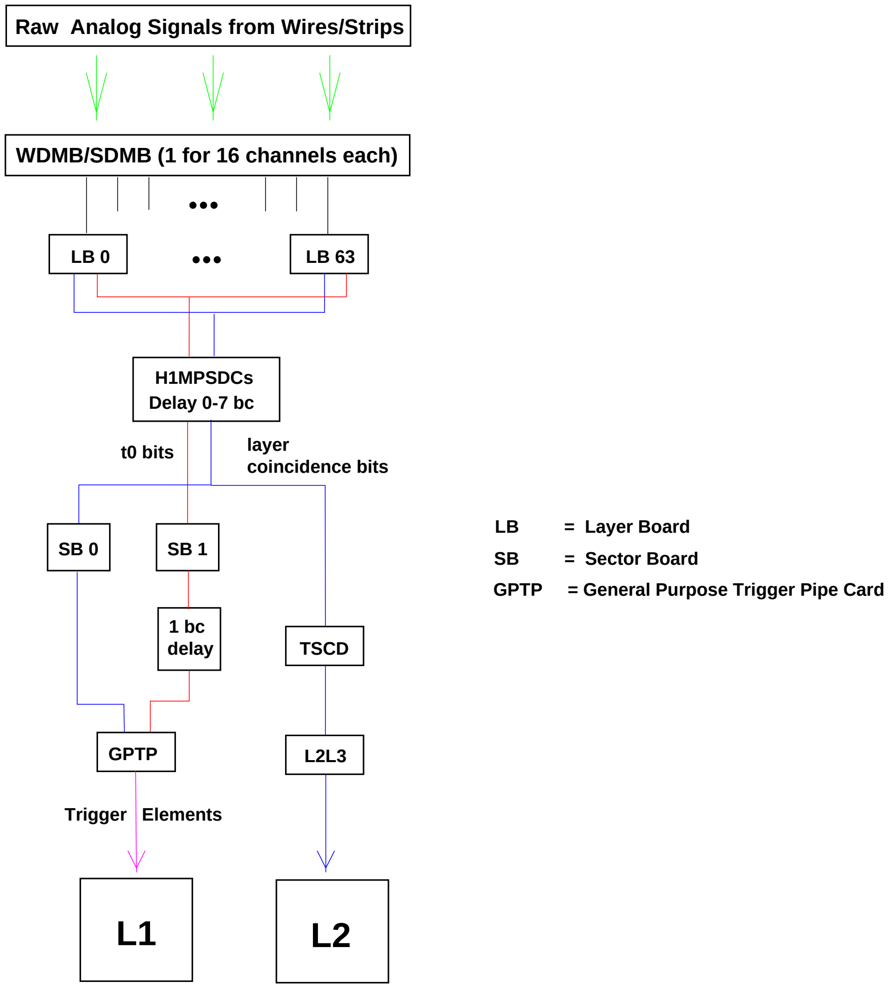

Abbildung 1.6: Der Fluß der Signale des Myon-Triggers von den Drähten bis hin zu den Triggerstufen L1 und L2 [14].

Layer Boards

Die layer boards bilden eine logische ODER-Verknüpfung der binären Signale der Drahtkammer-Elemente, so daß aus dem resultierenden Wert hervorgeht, ob eine Lage insgesamt von einem Teilchen getroffen wurde oder nicht. Die weitere Verknüpfung geschieht nach dem Prinzip der *n/m-Bedingung*, d.h. daß die layer boards ein positives Ausgangssignal liefern, wenn von m Lagen eines Moduls mindestens n auch getroffen wurden. Um welche und wie viele der Triggerlagen 3, 4, 5, 8 und 12 es sich bei m und n handelt, hängt von der Zugehörigkeit der Module zu den verschiedenen Subdetektoren ab (s. Tab. 1.3). So werden für ein positives Signal im Barrel-Bereich weniger getroffene Lagen gefordert als für die stark mit Untergrundteilchen belastete vorwärtige Endkappe.

Problematisch bei der Ermittlung der Lagenkoinzidenz ist die Tatsache, daß Spuren von ein und demselben Teilchen in unterschiedlichen Zeitscheiben des readout Signale hinterlassen können, und zwar infolge unterschiedlicher Driftzeiten in den Drahtkammern. Geht man von einer maximalen Driftzeit von ca. 140 ns in den Drahtkammern aus, so

BEC (außer Module 8 und 9)	3 aus {3, 4, 5, 8, 12}
Module 8, 9	2 aus {3, 4, 5} UND 1 aus {8, 12}
Barrel (außer Modul 33)	2 aus {3, 4, 5, 8}
Modul 33 (Run $<$ 251082)	2 aus {3, 4, 5, 8}
(Run $\geq$ 251082)	2 aus {3, 4, 5, 12}
FOEC	3 aus {3, 4, 5, 8, 12}
FIEC	4 aus {3, 4, 5, 8, 12}

Tabelle 1.3: Die verwendeten Lagenkoinzidenzbedingungen. Die linke Spalte enthält die Sektoren oder Module, die rechte die zu erfüllende Bedingung. Die Nummern der Triggerlagen sind in geschweifte Klammern gefaßt [14].

kann man die Ankunft aller von einer Teilchenspur verursachten Signale innerhalb zweier Zeitscheiben erwarten. Aus diesem Grund werden die einzeln für jede Triggerlage ermittelten Signale um jeweils eine Zeitscheibe gedehnt und mit einem logischen UND verknüpft (s. Abb. 1.7). Als *Eintreffzeitpunkt* t_0 wird die Zeitscheibe des ersten eintreffenden Signals markiert. Die layer boards liefern also pro Zeitscheibe zwei Signale (Bits) an die weitere elektronische Verarbeitungskette.

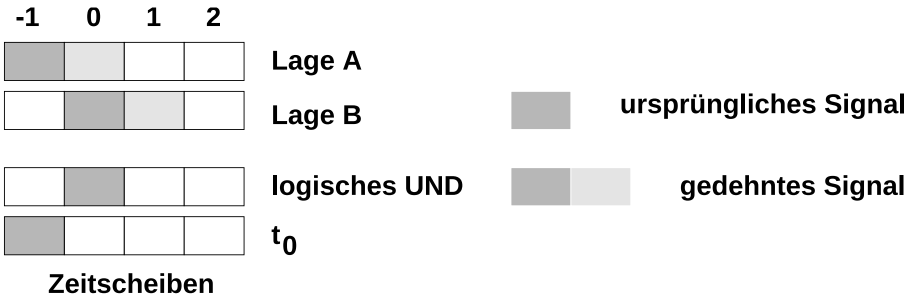

Abbildung 1.7: Zur Ermittlung der Lagenkoinzidenzen mit Hilfe der Signaldehnung. Das resultierende Koinzidenzsignal würde i.d.F. mit der Zeitscheibe 0 synchronisiert, das t_0-Signal mit der Zeitscheibe -1.

Sector Boards

Nach einer erneuten Synchronisation der zwei Gruppen à 64 Ausgabesignale der layer boards in den *H1 multi purpose signal delay cards* (*H1MPSDC*) werden sie im Speicher der beiden *sector boards* (*SB*) abgelegt. Im Zuge der weiteren Reduzierung der Trigger-Datenmenge geschieht an dieser Stelle eine Verknüpfung (logisches ODER) der Signale der layer boards[5], wobei die Koinzidenz- und t_0-Kanäle in je einem sector board getrennt behandelt werden. Beide sector boards stellen schließlich jedes für sich einen Satz von 16 Ausgabesignalen (*Triggersektoren*) zur Verfügung. Es handelt sich hierbei um zwei Bits pro Sektor; eines indiziert das Vorhandensein einer erfüllten Triggerbedingung in wenigstens einem Modul des Sektors, ein weiteres Bit indiziert Gleiches in wenigstens

[5]Hat also mindestens eines der Module in der Zeitscheibe n ein Lagenkoinzidenzsignal bzw. t_0-Signal erzeugt, so liefert das entsprechende sector board ein positives Ergebnis für Zeitscheibe n.

No.	Sektor	No.	Sektor	No.	Sektor	No.	Sektor
0	$n_{BIEC} > 0$	4	$n_{BUBA} > 0$	8	$n_{FUBA} > 0$	12	$n_{FIEC} > 0$
1	$n_{BIEC} > 1$	5	$n_{BUBA} > 1$	9	$n_{FUBA} > 1$	13	$n_{FIEC} > 1$
2	$n_{BOEC} > 0$	6	$n_{BLBA} > 0$	10	$n_{FLBA} > 0$	14	$n_{FOEC} > 0$
3	$n_{BOEC} > 1$	7	$n_{BLBA} > 1$	11	$n_{FLBA} > 1$	15	$n_{FOEC} > 1$

Tabelle 1.4: Die Definition der sechzehn Triggersektoren. In der jeweils rechten Spalte bedeutet n_{Sektor} die Zahl der Module, die im entsprechenden Sektor eine erfüllte Triggerbedingung aufweisen.

Sektor	TE-Nummer	TE-Name	TE-Definition
BIEC	56	Mu_BIEC	$n_{BIEC} > 0$
BOEC	57	Mu_BOEC	$n_{BOEC} > 0$
BEC	58	Mu_2_BIoOEC	$n_{BIEC} > 0$ ODER $n_{BOEC} > 0$
Barrel	59	Mu_Bar	$(n_{BUBA} + n_{BLBA} + n_{FUBA} + n_{FLBA}) > 0$
Barrel	60	Mu_3_Bar[6]	$(n_{BUBA} + n_{BLBA} + n_{FUBA} + n_{FLBA}) > 2$
FIEC	61	Mu_FIEC	$n_{FIEC} > 0$
FOEC	62	Mu_FOEC	$n_{FOEC} > 0$
FEC	63	Mu_2_FIoOEC	$n_{FIEC} > 0$ ODER $n_{FOEC} > 0$

Tabelle 1.5: Die Definition der acht Triggerelemente (TE) des Zentralen Myon-Detektors. In der rechten Spalte bedeutet n_{Sektor} die Zahl der Module, die im entsprechenden Sektor eine erfüllte Triggerbedingung aufweisen (vergl. Tab. 1.2).

zwei Modulen. Die sechzehn Triggersektoren sind demzufolge nicht mit den weiter oben beschriebenen Sektoren identisch, sondern bezeichnen vielmehr die Zahl der Module, die in diesen eine erfüllte Triggerbedingung aufweisen. Eine Übersicht gibt die Tab. 1.4.

Aufgrund der Driftzeiten in den Drahtkammern besitzen die Signale der Koinzidenz häufig eine größere zeitliche Ausdehnung als die t_0-Signale oder erreichen die Elektronik der Triggerstufe L1 eine Zeitscheibe zu spät (vgl. Abb. 1.7). Aus diesem Grunde werden die t_0-Signale gegenüber den Koinzidenzsignalen um eine Zeitscheibe verzögert, bevor sie die letzte Stufe des Myon-Triggers passieren können.

General Purpose Trigger Pipe Board

Im *general purpose trigger pipe board* (*GPTP*) findet schließlich der endgültige „Zusammenbau" der acht Triggerelemente statt, die anschließend an die Triggerstufe L1 übermittelt werden. Zunächst müssen dazu die Koinzidenz- und t_0-Informationen der beiden sector boards bitweise verglichen werden (logisches UND), nachdem sie in einer pipeline der Länge 32 abgelegt worden sind. Im nächsten Schritt erfolgt dann die Verknüpfung der Signale der einzelnen Sektoren zu den Triggerelementen. Deren Definition ist in Tab. 1.5 verzeichnet.

[6]Dieses Triggerelement existierte ursprünglich als Mu_2_Bar und wurde im Frühjahr 1999 geändert.

Das PQZP-System

Parallel zur Verarbeitung der Signale in den sector boards sind die Ausgabekanäle der multi purpose signal delay card auch mit den Speicherkarten des *PQZP*-Systems[7] verbunden, die bereits der Elektronik der Triggerstufe L2 zugeordnet sind. In diesen Speicherkarten wird die Information geführt, ob ein Modul eine Lagenkoinzidenz verzeichnet oder nicht.

Für die Erfassung der Triggersignale des Zentralen Myon-Detektors verfügt das PQZP-System über zwei Speicherkarten. Beide umfassen je eine pipeline mit einer Tiefe von 32 Zeitscheiben, unterscheiden sich jedoch in der Lage des als Zeitscheibe 0 ausgelesenen Eintrags der pipelines. Die Speicherkarte mit der logischen Nummer 0 führt den mit Zeitscheibe 0 korrespondierenden Eintrag an letzter Stelle, so daß 31 nachfolgende Zeitscheiben in der pipeline verzeichnet sind. Die Speicherkarte mit der logischen Nummer 1 hingegen führt den t_0-Eintrag an einer bestimmten Stelle innerhalb der pipeline[8] mit der Konsequenz, daß hier auch frühere Zeitscheiben zugänglich sind. Die Position des t_0-Eintrages in der Speicherkarte 1 ist bestimmt durch die Einstellung der Elektronik, kann also prinzipiell variiert werden.

Eine positive Entscheidung der ersten Triggerstufe bewirkt das Anhalten der pipelines, die Auslese der dem Interaktionszeitpunkt entsprechenden Speicherstelle (deren Position ist, einmal vorgegeben, konstant) und die Übermittlung der Informationen an die *L2L3-Speicherkarten* der Triggerstufe L2. Diese bilden nun zusätzlich zu den 8 Triggerlementen (bzw. den daraus abgeleiteten Subtriggern) den Beitrag des Myon-Detektors zur Datenmenge, mit deren Hilfe L2 die Triggerentscheidung fällt.

1.3.4 Das Monitoring des Myon-Detektors

Bislang stand zum Zwecke der Kontrolle der Funktion des Myon-Detektors das Programm *KOOL* [17] zur Verfügung, welches u.a. auch zur Online-Überwachung des H1-Experimentes während der Datennahme verwendet wird. KOOL erstellt eine ganze Reihe einfacher Zählhistogramme, auf die auch grundlegende Operationen angewendet werden können. So sind Raten von Triggerelementen, Subtriggern, usw. abrufbar sowie speziell auf den Myon-Detektor bezogene Daten wie beispielsweise die Anzahl der getroffenen Drähte, Streifenelektroden und Lagen.

Insbesondere aufgrund der komplexen Struktur der Signalverarbeitung innerhalb des Myon-Triggers und der vielfältigen Konfigurations- und Kalibrationsmöglichkeiten ist die permanente Kontrolle der Funktion des Myon-Detektors unbedingt notwendig.

Trotz Übersichtlichkeit und Mächtigkeit sind die von KOOL erstellten Histogramme nicht differenziert genug, um eine eingehende Überwachung zu ermöglichen. Daraus ergibt sich die Notwendigkeit einer eigens zu diesem Zweck erstellten Programmbibliothek, welche in den folgenden Kapiteln beschrieben ist.

[7] PQZP: parallel quickbus zero-suppression processor

[8] Während des Jahres 2000 befand sich dieser Eintrag an der Stelle 20 der pipeline.

Kapitel 2

Die Klassen-Bibliothek

Die Programmiersprache C++ stellt das Konzept der *Klasse* bereit, also einen benutzerdefinierten Datentyp, der neben einer beliebigen Gruppe von Variablen auch einen Satz von Methoden (Funktionen), die auf die Variablen angewendet werden können, beinhalten kann. Eine im Speicher eines Rechners realisierte Instanz einer solchen Klasse wird als *Objekt* bezeichnet. Als objektorientierte Programmiersprache bietet C++ ebenfalls das Prinzip der *Vererbung*, d.h. die Übertragung von Variablen und Methoden einer Klasse auf von ihr abgeleitete Tochterklassen.

Für die vorliegende Analyse der Rohdaten des Myon-Detektors wurde eine geeignete Bibliothek von C++-Klassen entworfen und angefertigt. Dabei sind im wesentlichen zwei Ziele verfolgt worden:

- Der Quellcode der einzelnen Klassen soll für sich genommen *vollständig* sein, d.h. eine Benutzung sollte grundsätzlich ohne weiteres Editieren der Quelldateien möglich sein. Damit einher geht auch die Möglichkeit, die Klassen und die darin implementierten Analyseschritte mit wenig Aufwand in ein lauffähiges Programm zu integrieren. Insgesamt bleibt aber eine Modifikation des Quellcodes zur Anpassung an veränderte Gegebenheiten problemlos durchführbar.

- Es wurde darauf geachtet, die Struktur der Klassen-Bibliothek entsprechend einem modularen Schema zu entwerfen, welches auch *Erweiterungen* zuläßt. Auf diese Weise kann die zur Analyse verfügbare Sammlung von Methoden ausgeweitet werden, da neue Klassen einfach in die bereits vorhandene Struktur einfügbar sind.

Bei der Erstellung der Klassenbibliothek wurde in hohem Maße von der Funktionalität des ROOT-Systems Gebrauch gemacht, welches im folgenden Abschnitt beschrieben ist. Ein Teil der Klassen entstand mit Hilfe des Programms `bos2oop` (s. Abschn. 2.2).

2.1 Das ROOT-System

Das ROOT-System entstand am CERN und stellt eine Programmierumgebung zur Datenanalyse dar, die insbesondere abgestimmt ist auf das Gebiet der Hochenergiephysik. Grundsätzlich handelt es sich dabei um eine Bibliothek von C++-Klassen, die durch ein interaktives Programm ergänzt werden. ROOT ist damit sowohl in kompilierte C++-Programme integrierbar als auch interaktiv oder als Makrointerpreter zu verwenden. C++ fungiert dabei als interaktive Sprache und auch als Makro-Sprache. Die von ROOT zur Verfügung gestellten Klassen lassen sich grob in folgende Gruppen zusammenfassen.

- *Basisklassen*, die Standards für Objektstrukturen definieren und Kompatibilität der Klassen untereinander ermöglichen, beispielsweise `TObject`
- *Container-Klassen* zur Verwaltung von Vektoren, Listen und ähnlichen häufig gebrauchten Gruppen-Datenstrukturen, wie etwa `TObjArray` und `TClonesArray`
- *Histogramm- und Minimierungsklassen*, welche die Erfassung von Variablen in Histogrammen bieten sowie die Anpassung von Funktionen, statistische Analysen u.ä.
- *Klassen zur Datenerfassung* wie die Klassen `TTree` und `TNtuple` zur flexiblen Speicherung großer Datenmengen
- *Klassen für 2- und 3-dimensionale Graphikelemente*
- *Schnittstellen-Klassen und C++-Interpreter-Klassen* zum Zwecke der Interaktion mit dem Benutzer
- *Schnittstellen-Klassen für das Betriebssystem*
- *Netzwerk-Klassen* zur Verwendung von ROOT auf untereinander vernetzten Rechnern
- *Klassen zur automatischen Dokumentationserstellung*

Das ROOT-System verwendet das Konzept des *tree*[1], eines hierarchisch aufgebauten Gefüges zur strukturierten Speicherung von Variablen. Ein tree besitzt eine beliebige Anzahl von *branches*[2], diese wiederum eine beliebige Anzahl von *leaves*[3], welche die eigentlichen Variablen darstellen und somit im Rahmen ihres jeweiligen branches übersichtlich gruppiert sind. Wesentlich hierbei ist, daß sowohl die branches als auch die leaves jeweils Vektoren mit dynamischer Länge bilden. Diese Struktur wird für jedes Ereignis dupliziert und an den bereits vorhandenen tree angefügt, nachdem die Variablen mit Werten belegt worden sind.

Ein Großteil der Klassen des ROOT-Systems ist abgeleitet von der Klasse `TObject`, die auf diese Weise Methoden für ein einheitliches Verhalten in Bezug auf Ein- und Ausgabe, Fehlerbehandlung, graphische Darstellung, usw. an ihre Tochterklassen vererbt. Hier liegt auch der Schlüssel für die Einbindung eigener Klassen in das ROOT-System, indem man diese als von `TObject` abgeleitete Klassen definiert. Zusätzlich zu dieser Vorgehensweise erstellt ROOT für zu implementierende fremde Klassen ein Verzeichnisobjekt (*dictionary*), welches deren wichtigste Merkmale wie Name, Speicherplatzbedarf, Mutterklassen, enthaltene Variablen und Methoden, usw. aufschlüsselt.

[1]engl. f. Baum
[2]engl. f. Äste
[3]engl. f. Blätter

Das ROOT-System bietet ferner einen Mechanismus zur automatischen Erstellung von Dokumentationen im HTML-Format, und zwar sowohl für implementierte fremde Klassen als auch für zu kompilierende Programme oder Makros. Die Dokumentation umfaßt außerdem Klassenverzeichnisse, Diagramme zur Visualisierung von Vererbungsreihen und Datentypenregister.

2.2 bos2oop

2.2.1 Das H1-Datenformat

Die Daten des H1-Detektors, insbesondere auch die Rohdaten, sind in FPack-Dateien auf Magnetbändern im BOS-Format[4] abgespeichert [19] (für eine Beschreibung des FPack-Systems s. [21]). Das BOS-Format wurde 1975 am DESY entwickelt, um eine der Schwachstellen der Programmiersprache FORTRAN zu umgehen, die in der unflexiblen Dimensionierung von Vektoren und Datenfeldern besteht. Demnach werden Sätze von Ereignisvariablen verwandter Natur in BOS-Datenbänken gruppiert, wobei diese ihrerseits ihre eigene Länge, d.h. die Zahl ihrer Datensätze, verwalten können.

Die logische Struktur einer BOS-Bank ist im Prinzip die einer Tabelle und definiert durch die Variablen (Spalten), deren Werte sie beinhaltet. Die physikalische Struktur jedoch ist aufgrund der sequentiellen Natur des Speichermediums eine andere. Die Werte der gespeicherten Variablen stehen auf dem Band hintereinander in einer Reihe; dies macht es notwendig, die Länge der Bank (dies entspricht der Zahl der Zeilen einer Tabelle) und die Zahl der enthaltenen Variablen zu Beginn eines jeden Datensatzes mit abzuspeichern.

Die Definition einer BOS-Bank, also der Satz von Variablen, die in der Bank gruppiert sind, wird durch eine Beschreibung mit Hilfe der *data description language* (*DDL*) festgelegt und steht damit sowohl bei der Speicherung der Daten als auch zur Auslese der Bänder zur Verfügung. Die Übergabe der von FPack ausgelesenen Werte an die auswertenden Programme geschieht durch Zwischenspeicherung in einem *common-Block*[5]. Deshalb ist eine Konvertierung der Rohdaten in das ROOT-Format derzeit nicht gänzlich ohne die Einbindung von FORTRAN-Routinen möglich.

Nach abgeschlossener Datenaufnahme stehen die Bänder für die Ereignisrekonstruktion zur Verfügung. Die Rekonstruktion von Spuren und Vertices, die Bestimmung von Teilchenimpulsen und -energien sowie die Ermittlung einer ganzen Reihe weiterer Parameter zur Verwendung in der physikalischen Analyse geschieht in diesem Schritt. Die Ergebnisvariablen werden wiederum in speziellen BOS-Bänken gruppiert und auf den sog. POT[6]- bzw. DST[7]-Bändern bereitgestellt.

Im Zuge der in Planung befindlichen Umstrukturierung der H1-Analyseprogramme hin zu einem C++-orientierten System ist auch eine weitgehende Neukonzeption des Datenformates vorgesehen [20]. Die BOS/FPack-basierten DST-Daten sollen hierbei durch ein vollkommen neues Format ersetzt werden. Eine Änderung des Formates der Rohdaten, welche künftig in die POT-Daten eingegliedert werden, ist jedoch nicht geplant.

[4]BOS: *bank operating system*

[5]Dies ist ein Speicherbereich, der verschiedenen FORTRAN-Subroutinen gemeinsamen Zugriff erlaubt.

[6]POT: *production ouput tapes*

[7]DST: *data summary tapes*

2.2.2 Die Verwendung von bos2oop

Die Vorgehensweise, Sätze von Variablen in einer übergeordneten BOS-Bank zusammenzufassen, ist im Grunde genommen durchaus vergleichbar mit dem Prinzip des objektorientierten Programmierens. Die Definition einer BOS-Bank findet dort ihre Entsprechung in dem Begriff der Klasse. In der vorliegenden Analyse wurde das Programm `bos2oop` [19] verwendet, mit dessen Hilfe eine problemlose Generierung von den BOS-Bänken entsprechenden C++-Klassenstrukturen möglich ist.

`bos2oop` bezieht die Information über Struktur und Umfang der zu erzeugenden Klassendefinition standardmäßig aus den in DDL verfaßten Definitionen der entsprechenden BOS-Bänke, die in der Datei `/h1/h1banks/h1banks.car` auf den H1-Rechnern verfügbar ist. Es besteht darüberhinaus die Möglichkeit, eine eigene DDL-Datei mit Bankdefinitionen zu erstellen. Auf diese Weise können den entstehenden Klassen einfache Funktionen hinzugefügt werden[8], die beispielsweise zur Dekodierung binär kodierter Variablen nützlich sind.

Die Namen der BOS-Bänke, deren C++-Pendants erzeugt werden sollen, sind dem Programm mittels einer Konfigurationsdatei mitzuteilen. Ferner kann optional eine *einhüllende* oder *übergeordnete Klasse* erzeugt werden, die die entsprechenden Variablen und Routinen für einen bequemen Zugriff auf die Daten der eingebundenen Klassen enthält, und zwar jeweils für ein vollständiges einzelnes Ereignis. Die einhüllende Klasse verfügt auch über einen Satz von Routinen, welche zur Konvertierung der Rohdaten in das ROOT-Format verwendet werden können. Diese ist im Anhang und in [22] detailliert beschrieben.

2.3 Die Struktur der Klassen-Bibliothek

Die Struktur der hier beschriebenen Klassen-Bibliothek, die den zu Beginn des Kapitels aufgeführten Zielen der einfachen Anwendbarkeit und Erweiterbarkeit genügt, gründet sich auf eine dreistufige Hierarchie. Sie umfaßt die von `bos2oop` erzeugten *Datenklassen*, die auswertenden *Basisklassen* und die *ausführenden Klassen* mit der Möglichkeit einer bankübergreifenden Bearbeitung der Daten. Zur Verwendung der Klassenbibliothek müssen die Rohdaten in das ROOT-Format übertragen werden. Parallel zu dieser Arbeit wurde außerdem eine ausführliche Dokumentation im HTML-Format zur Verwendung der Klassen-Bibliothek angefertigt, die auch den Quellcode selbst beinhaltet [22].

2.3.1 Datenklassen

Alle hier verwendeten Datenklassen wurden von `bos2oop` automatisch generiert. Die Bezeichnungen der Datenklassen richtet sich nach den Namen der BOS-Bänke, die ihnen jeweils zugrunde liegen. Im Falle von Bänken, die mehrere Datensätze (äquivalent den Zeilen einer Tabelle) enthalten können, wird die erzeugte Datenklasse nur je eine der Zeilen verwalten können, und ihre Bezeichnung erhält die Endung ,,Row". Lediglich zwei der

[8] Hierzu steht die von `bos2oop` interpretierte `INCLUDE`-Umgebung zur Verfügung.

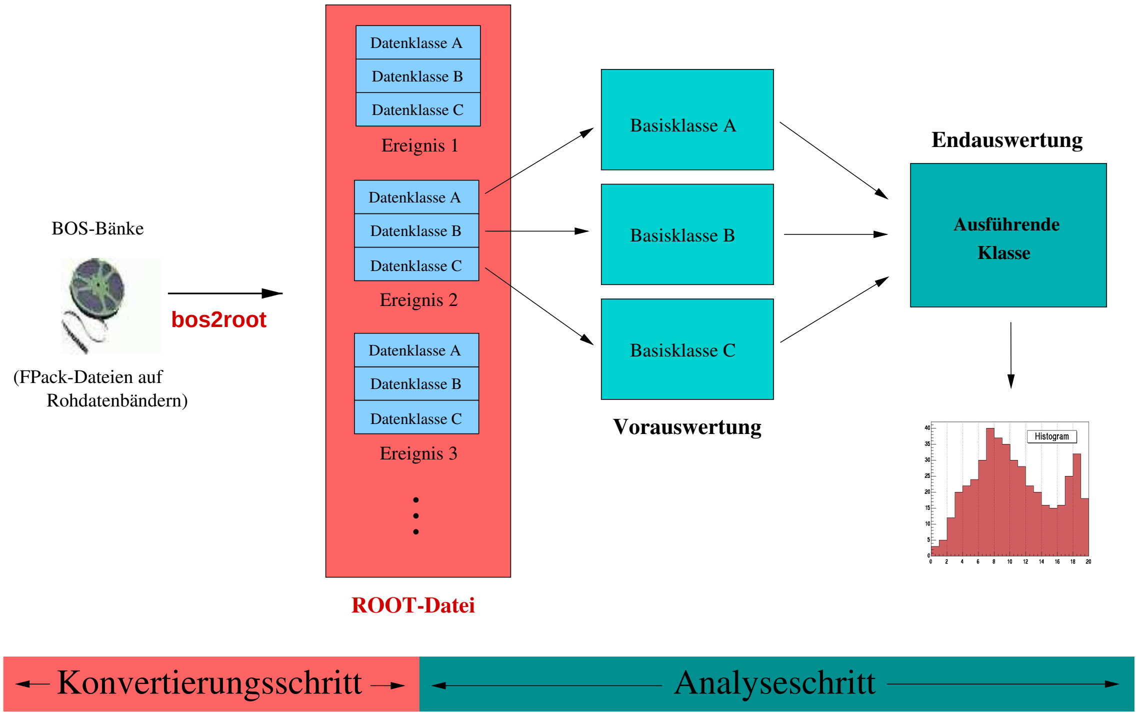

Abbildung 2.1: Grob-schematischer Ablauf der Datenanalyse mit den Methoden der Klassen-bibliothek. Die im Konvertierungsschritt erzeugte ROOT-Datei wird später von den Basisklassen und ausführenden Klassen ausgelesen.

hier ausgelesenen Bänke enthalten durchgehend nur eine Zeile pro Ereignis (die Bänke HEAD und YTL4, vergl. Tab. 2.1).

Zusätzlich zu den Variablen, die denjenigen der BOS-Bänke entsprechen, enthalten die Datenklassen eine Reihe von Methoden für den lesenden Zugriff, den schreibenden Zugriff während der Rohdatenkonvertierung, die Überprüfung der Variablenwerte und die Einbindung in das ROOT-System.

Die übergeordnete Klasse `TMUONS` enthält selbst keine Variablen von physikalischer Bedeutung, wohl aber eine Reihe von Vektoren der ROOT-Klasse `TClonesArray` für die einzelnen Datenklassen. Jede Komponente eines solchen Vektors beinhaltet genau eine Zeile von Daten der zugrundeliegenden BOS-Bank. Auf diese Weise wird der gesamte Inhalt eines Ereignisses der BOS-Bank in Form eines Vektors gespeichert. Die Zahl der Komponenten der Vektoren ist ebenfalls in `TMUONS` verzeichnet.

In den zur Rohdatenkonvertierung und -speicherung verwendeten ROOT-trees bilden die Datenklassen die branches (vergl. Abschn. 2.1).

2.3.2 Basisklassen

Während die automatisch generierten Datenklassen hauptsächlich eine speichernde Funktion haben, liegt der Schwerpunkt bei den Basisklassen auf den ersten Schritten der Daten-

[9] Die Klasse TResl ist zwar mit `bos2oop` erzeugt worden, entspricht aber keiner existierenden BOS-Bank. Sie nimmt lediglich Ergebnisse von Berechnungen während der Konvertierung der Rohdaten auf.

Datenklasse	BOS-Bank
TMUONS	-
THead	HEAD
TIrweRow	IRWE
TIrseRow	IRSE
TTel1_0Row	TEL1(0)
TTl23_8Row	TL23(8)
TItkrRow	ITKR
TTscd_8Row	TSCD(8)
TTlv1Row	TLV1

Datenklasse	BOS-Bank
TTgpp_2048Row	TGPP(2048)
TItprRow	ITPR
TItwxRow	ITWX
TYtl4	YTL4
TCjccRow	CJCC
TCjgrRow	CJGR
TCjkvRow	CJKV
TMtctRow	MTCT
TResl[9]	-

Tabelle 2.1: Die in dieser Analyse verwendeten Datenklassen und die zugrundeliegenden BOS-Bänke. Die in Klammern gefaßten Zahlen geben die logischen Nummern der BOS-Bänke an, sofern diese mehrfach vorhanden sind.

Basisklasse	Verwendete Datenklassen
TMuIRWE	TIrweRow, THead
TMuTEL1	TTel1_0Row, THead, TResl (nur cosmic/beamhalo runs)
TMuTL23	TTl23_8Row
TMuYTL4	TYtl4
TMuITKR	TItkrRow
TMuTSCD	TTscd_8Row, THead
TMuTGPP	TTgpp_2048Row, THead
TMuITPR	TItprRow
TMuITWX	TItwxRow

Tabelle 2.2: Die Basisklassen und die Datenklassen, auf die sie zurückgreifen.

auswertung. Prinzipiell erfolgt die Datenauswertung weitestgehend durch die Basisklassen, sofern dafür nicht Variablen aus verschiedenen Datenklassen benötigt werden[10].

Bei der Erzeugung eines Objektes einer Basisklasse ist ein Zeiger[11] auf die Instanz der Klasse `TMUONS` an dieses zu übergeben. Somit erhält das neu entstandene Objekt Zugriff auf alle Daten eines Ereignisses und wird selbständig die entsprechenden Variablen der benötigten Datenklassen auslesen. Dies impliziert grundsätzlich, daß Basisklassen dahingehend konzipiert sind, lediglich ein einzelnes Ereignis zu bearbeiten, um anschließend aus dem Speicher des Rechners entfernt und wieder neu erzeugt zu werden.

Der Funktionsumfang einer Basisklasse ist stark von den in der zugrundeliegenden Datenklasse enthaltenen Informationen abhängig. Als Beispiele seien hier die Ermittlung von erfüllten Triggerbedingungen durch die Basisklasse `TMuIRWE` oder die Suche nach

[10]Eine Ausnahme hiervon bildet die Datenklasse *THead*, welche u.a. die Ereignis- und Runnummern verwaltet.

[11]In der C/C++-Nomenklatur bezeichnet ein Zeiger eine ganzzahlige Variable, die eine Speicheradresse beinhaltet.

Triggersignalen in den PQZP(8)-Speicherkarten durch `TMuTSCD` erwähnt. Eine vollständige Dokumentation ist in [22] zu finden.

2.3.3 Ausführende Klassen

Die oberste Stufe in der Hierarchie der Klassen-Bibliothek nehmen die ausführenden Klassen ein, welche dabei auf Methoden und Zwischenergebnisse einer oder auch mehrerer Basisklassen zurückgreifen. Jede ausführende Klasse ist auf eine bestimmte Aufgabe spezialisiert und kann dazu eine entsprechende Gruppe von Histogrammen, Bildschirmausgaben oder zusätzlichen Dateien als Ergebnis liefern. Die Ergebnisse der Bearbeitung werden fortlaufend in Histogrammen gesammelt, am Ende des Programmablaufs ggf. weiterverarbeitet und in einer Ausgabedatei gespeichert.

Ein Objekt einer ausführende Klasse bearbeitet grundsätzlich nur ein einzelnes Ereignis. Ist dessen Bearbeitung abgeschlossen, müssen dem Objekt die Daten eines neuen Ereignisses zugeführt werden. Eine externe Programmschleife ist also notwendig, um die Daten aller Ereignisse in das Ergebnis einfließen zu lassen. Jede der ausführenden Klassen umfaßt die drei speziellen Methoden `Init()`, `Include()` und `Finish()` (vergl. Abb. 2.2), mit deren Hilfe sie sich in ein zu kompilierendes Programm eingliedern läßt. Mit Hilfe der Methode `Init()` einer ausführenden Klasse, die vor der Programmschleife aufzurufen ist, kann die notwendige Initialisierung des Objektes durchgeführt werden, welches damit eine Reihe notwendiger Programmschritte selbständig veranlaßt:

- Die Auslese der benötigten branches des trees, welcher die konvertierten Rohdaten enthält
- Die Initialisierung der zu füllenden Histogramme
- Das Einrichten eines speziellen Unterverzeichnisses in der Ausgabedatei

Aus diesem Grunde müssen den zu erzeugenden Objekten der ausführenden Klassen Zeiger übergeben werden auf die Objekte des trees und der Ausgabedatei.

Ist das Objekt initialisiert, kann innerhalb der Programmschleife die eigentlich ausführende Methode `Include()` aufgerufen werden, welche die Berechnungen durchführt und fortlaufend die Histogramme aktualisiert. Die dazu notwendigen Informationen werden hauptsächlich von den Objekten der Basisklassen an die Objekte der ausführenden Klassen übergeben. Werden nur einige rohe Zahlenwerte aus den Datenklassen benötigt, die nicht erst durch die Methoden einer Basisklasse aufzubereiten sind, so können die Datenklassen an dieser Stelle auch direkt ausgelesen werden. Hierzu muß dem zu erzeugenden Objekt der ausführenden Klasse zusätzlich ein Zeiger auf das Objekt der übergeordneten Klasse übergeben werden.

Eine Reihe von Diagrammen, insbesondere diejenigen, die Effizienzwerte zeigen, entstehen durch Division zweier vollständiger Histogramme. Auch können bestimmte Prüfmethoden nur mit vollständig gefüllten Histogrammen durchgeführt werden. Zu diesem Zweck verfügen alle ausführenden Klassen über eine Methode `Finish()`, die nach der Programmschleife aufzurufen ist. Aus Kontinuitätsgründen ist `Finish()` auch dann vorhanden, wenn keine abschließenden Bearbeitungsschritte notwendig sind.

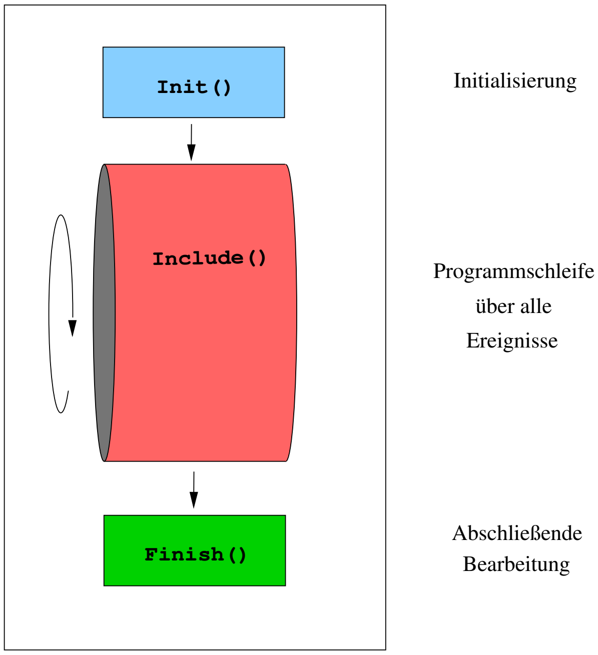

Abbildung 2.2: Das Einbinden einer ausführenden Klasse in ein zu kompilierendes Programm mit Hilfe der Methoden `Init()`, `Include()` und `Finish()`.

Zur einfachen Implementierung der ausführenden Klassen steht die Klasse `TMuLoop` zur Verfügung. Sie beinhaltet bereits die Programmschleife zur sukzessiven Bearbeitung aller Ereignisse des trees sowie die zum Aufruf der ausführenden Klassen notwendigen Befehle. `TMuLoop` speichert die Objekte der ausführenden Klassen in einem Vektor ab und ruft die drei Methoden zur Implementierung an den entsprechenden Stellen selbständig auf. Zur Speicherung der Objekte ist jeweils eine Programmzeile erforderlich. Welche Aufgaben ausgeführt werden sollen, kann nun einfach durch Entfernen oder Hinzufügen einer einzigen Zeile gesteuert werden.

Das Einlesen der Rohdaten kann mittels der Klasse `TMuFiles` geschehen. Hierzu sind die Namen der konvertierten Rohdatendateien in einer Liste zusammenzustellen und abzuspeichern. Die Methoden von `TMuFiles` sind in der Lage, diese Liste zu erfassen und einen Zeiger auf einen tree zurückzuliefern, welcher die gesamte Datenmenge der aufgelisteten Rohdatendateien vereint. Dabei stehen verschiedene Methoden bereit mit jeweils unterschiedlichem Funktionsumfang in Bezug auf die Erkennung von Dateipfaden.Solch ein Zeiger auf einen tree kann unmittelbar von den ausführenden Klassen verwendet werden.

Insgesamt bietet die Klassenbibliothek also eine Möglichkeit, verschiedene Analysen der Rohdaten des Zentralen Myon-Detektors mit einem lauffähigen Programm durchzuführen, welches mit wenigen Programmzeilen bereits voll funktionstüchtig ist.

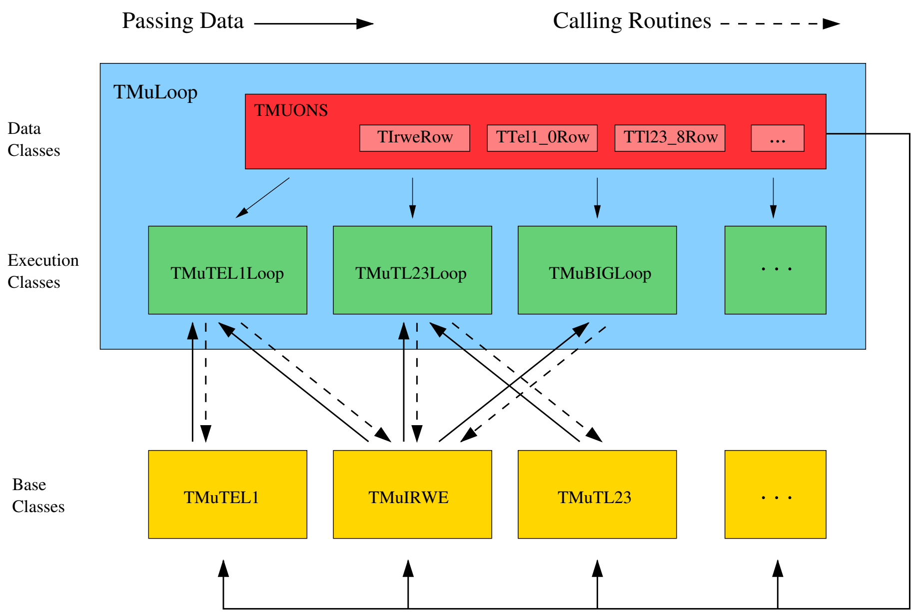

Abbildung 2.3: Der Datenfluß und der Aufruf von Methoden zwischen den Objekten der Klassen-Bibliothek während der Bearbeitung eines Ereignisses. Die Klasse `TMuLoop` enthält Objekte der ausführenden Klassen sowie der übergeordneten Datenklasse `TMUONS`. Bei Bedarf werden von den ausführenden Klassen Methoden der Basisklassen aufgerufen, welche sich der Daten in `TMUONS` bedienen.

Kapitel 3

Triggereffizienzen

Die Effizienz des Myon-Triggers ist definiert als diejenige Wahrscheinlichkeit, ein Triggersignal in der korrekten Zeitscheibe vorzufinden, falls ein Myon das Instrumentierte Eisen durchquert. Sie hat demnach neben einem logischen auch einen zeitlichen Aspekt, ein Triggersignal kann sowohl ausbleiben als auch mit zeitlicher Verschiebung generiert werden, was beides die Effizienz verringert.

Grundsätzlich wird die Effizienz eines Triggersystems ermittelt durch die Zählung derjenigen Ereignisse, in denen die Bedingung für ein Triggersignal erfüllt war und dieses auch tatsächlich zu einem bestimmten vorgegebenen Zeitpunkt registriert wurde. Histogramme, mit deren Hilfe diese Zählung durchgeführt wird, sind nachfolgend als *Konformitätshistogramme* bezeichnet. Die ermittelte Anzahl ist zu normieren auf die Zahl der Ereignisse, in denen erfüllte Triggerbedingungen vorlagen. Deren Zählung geschieht in einem *Bedingungshistogramm*. Dies kann für einzelne Detektorkomponenten separat geschehen; in der vorliegenden Analyse beziehen sich alle Triggereffizienzen auf die einzelnen Module des Zentralen Myon-Detektors.

In diesem Kapitel werden zunächst Verfahren vorgestellt zur Ermittlung sowohl der rein zeitlichen Triggereffizienz als auch der Effizienz der Elektronik des Myon-Triggers (Hardware-Effizienz). Parallel dazu sind die Klassen beschrieben, mit denen die einzelnen Verfahren durchgeführt werden können. Darauf aufbauend wird diskutiert, inwiefern die Teileffizienzen zu einer Gesamteffizienz zu kombinieren sind und somit eine bessere Eingrenzung der Ursachen verminderter Triggereffizienz möglich wird. Den Abschluß bildet ein Vergleich der Ergebnisse mit denen eines anerkannten Verfahrens zur Bestimmung der Triggereffizienz.

3.1 Die Bestimmung der zeitlichen Effizienz des Myon-Triggers

Die hier beschriebene Methode zur Ermittlung der zeitlichen Triggereffizienz wurde ausführlich in [10] vorgestellt. Im Zuge dieses Verfahrens werden die Daten kosmischer sowie von Halo-Myonen verwendet, um die Reaktion des Myon-Triggers mit möglichst hoher zeitlicher Auflösung zu analysieren.

Im folgenden wird zunächst diskutiert, in welcher Weise verschiedene Effekte die zeitliche Verteilung von Triggersignalen beeinflussen, die von Myonen im normalen Strahlbetrieb des H1-Experimentes ausgelöst werden. Ein Myon, welches seinen Ursprung in einer $e^{\pm}p$-Streureaktion hat und somit dem nominellen Wechselwirkungspunkt des H1-Experimentes entstammt, benötigt eine gewisse Zeitspanne, um die Module des Zentralen Myon-Detektors zu erreichen. Auch Driftzeit und Signalwege sind mitbestimmend für die Zeitscheibe, in welcher das Triggersignal letztendlich zu liegen kommt.

Wie weiterhin gezeigt wird, kann lediglich die Driftzeit nicht aus der Flugbahn des Myons rekonstruiert werden. Sie unterliegt einer zufälligen Verteilung, und ihren Einfluß auf die zeitliche Triggereffizienz gilt es zu ermitteln. Flug- und Signalzeiten sind bereits bei der Konfiguration der Elektronik des Myon-Triggers berücksichtigt.

Da im normalen Strahlbetrieb eine feste Phasenbeziehung der Wechselwirkungszeitpunkte zu dem Takt der HERA clock besteht, muß eine Untersuchung des zeitlichen Verhaltens des Myon-Triggers mit den Daten kosmischer und von Halo-Myonen durchgeführt werden. Zu diesem Zweck sind Korrekturen zur Flugzeit vorzunehmen, da die Konfiguration der Elektronik des Myon-Triggers auf normale Runperioden abgestimmt ist.

Eine Beschreibung der Klasse `TMuSpecialLoop`, die Teil der im letzten Kapitel vorgestellten Klassenbibliothek ist und zur Ermittlung der zeitlichen Triggereffizenz verwendet werden kann, sowie die Diskussion der Ergebnisse ist Gegenstand des nächsten Abschnittes.

3.1.1 Einfluß der Nachweisverzögerung

Durchquert ein Myon, welches dem nominellen Wechselwirkungspunkt des Experimentes entstammt, eine der Drahtkammern des Zentralen Myon-Detektors, so wird der darin entstehende Spannungspuls aufgrund der Flugzeit t_{Flug} vom Vertex bis zur Kammer, der Driftzeit t_{Drift} innerhalb der Kammer und der Verzögerung durch Signalwege t_{Signal} mit einer gewissen Verzögerung in der unmittelbar auslesenden Elektronik eintreffen. Die Summe von Driftzeit, Flugzeit und Verzögerung durch Signalwege wird im folgenden als *Nachweisverzögerung* t_N bezeichnet, welche insgesamt eine zeitlich kontinuierlich verteilte Größe darstellt,

$$t_N = t_{Drift} + t_{Flug} + t_{Signal} \quad . \tag{3.1}$$

Driftzeiten

Die Signalerzeugung in einer Drahtkammer geschieht durch Ionisation des Zählgases entlang der Spur eines durchquerenden Myons. Die auf diesem Wege entstehenden Elektronen werden im elektrischen Feld der Kammer beschleunigt und driften zum Anodendraht hin, wobei durch Stöße kaskadenartig weitere freie Ladungsträger erzeugt werden. Der Verlauf des elektrischen Feldes im Innern der Kammer bewirkt eine im Mittel konstante Driftgeschwindigkeit der Elektronen durch das Zählgas in der Größenordnung von etwa 50 $mm/\mu s$.

Die Driftzeit variiert infolgedessen mit dem lotrechten Abstand der Spur vom Draht im Innern der Kammer, so daß dieser Anteil der Verzögerung einer kontinuierlichen Verteilung unterliegt. In etwa 4% der Fälle erreichen die Driftzeiten 96 ns und mehr. Die längsten beobachteten Driftzeiten liegen bei ca. 140 ns [10].

Flug- und Signalzeiten

Da die Module des Zentralen Myon-Detektors nicht sphärisch um den Wechselwirkungspunkt herum angeordnet sind, benötigen Myonen unterschiedliche Flugzeiten, um in verschiedenen Richtungen das Instrumentierte Eisen zu erreichen. Die Flugzeiten variieren dabei nicht nur von Modul zu Modul, sondern auch für die einzelnen Lagen und ebenfalls für verschiedene Punkte in ein und derselben Drahtkammer. Für hochenergetische Myonen, deren Bahnen vom magnetischen Feld innerhalb des Detektors nicht zu stark gekrümmt werden, kann die Flugzeit leicht berechnet werden,

$$t_{Flug} = \frac{S}{c} = \frac{1}{c}\sqrt{x^2 + y^2 + z^2} \quad , \tag{3.2}$$

wenn die Fluggeschwindigkeit mit der Lichtgeschwindigkeit c approximiert wird. Die Koordinaten x, y und z der Orte, an welchen die Teilchenspuren die Streifenelektroden durchquert haben, sind in der BOS-Bank ISCR verfügbar.

Zusätzlich zu den Flugzeiten muß auch die Dauer berücksichtigt werden, die das Signal für die Fortpflanzung entlang der Drähte benötigt (*Signalzeit*). Die unmittelbar auslesende Elektronik ist an demjenigen Ende der Drahtkammern mit den Drähten gekoppelt, welches die größte Entfernung vom Wechselwirkungspunkt besitzt. Aus diesem Grunde sind Flug- und Signalzeit miteinander korreliert, die Signalzeit wird desto kleiner, je länger die Flugzeit ist und umgekehrt [10].

Aus Abb. 3.1 kann das Verhältnis der Summe von Flugzeit und Signalzeit zur maximalen Flugzeit abgeleitet werden,

$$\frac{t_{Flug} + t_{Signal}}{t_{Flug,max}} = \frac{1}{c}\frac{\sin\vartheta_m}{\sin\vartheta} + \frac{1}{v_{Signal}}\left(\cos\vartheta_m - \sin\vartheta_m \cot\vartheta\right) \quad , \tag{3.3}$$

wobei die Geschwindigkeit v_{Signal} des Signals entlang eines Drahtes zu etwa 20 cm/ns angenommen werden kann und $\vartheta_m = \vartheta_{min}$ ist im rückwärtigen Teil des Detektors (BEC, BBA) bzw. $\vartheta_m = \vartheta_{max}$ im vorwärtigen Teil (FEC, FBA). Sowohl die Flugzeit als auch die Verzögerung durch Signalwege sind in der Einstellung der Elektronik jedes Moduls berücksichtigt, um einer Verschiebung der Triggersignale hin zu späteren Zeitpunkten entgegenzuwirken. Zusammengenommen rangieren sie von etwa 15 ns bis hin zu 35 ns [10], abhängig von der Position einer Kammer (bzw. des umgebenden Moduls) im Detektor.

Die zeitliche Effizienz des Myon-Triggers

Alle Prozesse im Verlauf der Erzeugung von Triggersignalen im Zentralen Myon-Detektor werden nun durch den Takt der HERA clock bestimmt. Dies betrifft sowohl die pipelines, in denen Signale zwischengespeichert werden, als auch die Auslese der Detektorkomponenten selbst. In der unmittelbar auslesenden Elektronik geschieht aus diesem Grunde eine erste Synchronisation der Signale mit der HERA clock und in den H1 multi purpose signal delay cards eine erneute Resynchronisation. Problematisch für die Beurteilung der

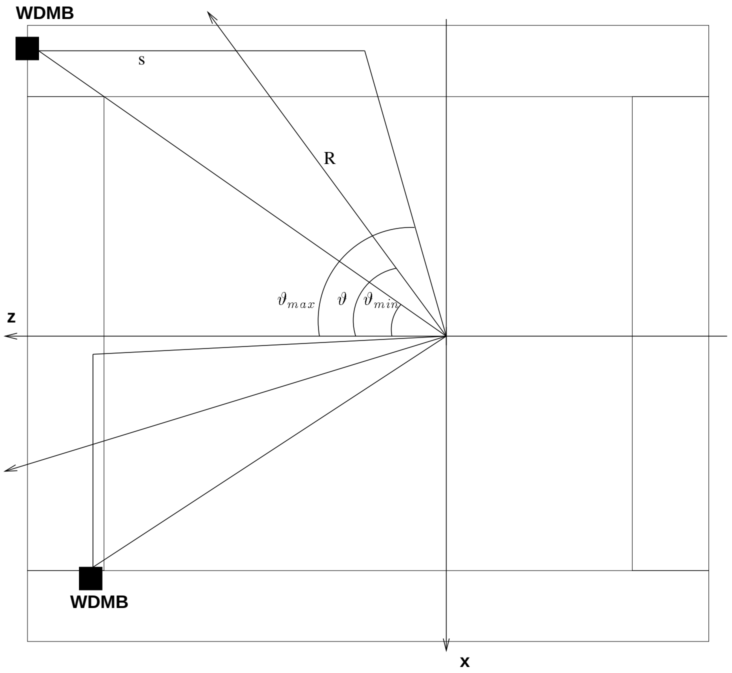

Abbildung 3.1: Zur Korrelation von Flugzeit und Signalzeit. Die Darstellung zeigt schematisch den Zentralen Myon-Detektor in Draufsicht [10]. Bei den WDMB handelt es sich um die unmittelbar auslesende Elektronik (vergl. Abschn. 1.3.1), s bezeichnet den vom Signal zurückzulegenden Weg. Weitere Erläuterungen im Text.

Triggereffizienz ist dabei, daß diese Synchronisationsvorgänge einen verbreiternden Effekt auf die Verteilung der resultierenden Triggersignale auf die Zeitscheiben haben. Diese Verteilung ist nicht mehr kontinuierlich, sondern diskret.

Von zwei analogen Signalen, die von demselben Myon in zwei unterschiedlichen Drahtkammern verursacht wurden, kann eines die unmittelbare Ausleseelektronik vor dem Eintreffen eines Taktes der HERA clock erreichen, das andere jedoch nach diesem Takt, falls es aufgrund einer längeren Driftzeit verzögert wurde. Beide Signale würden somit zwei aufeinanderfolgenden Zeitscheiben zugeschrieben, obwohl sie zeitgleich ausgelöst wurden. Die Wahrscheinlichkeit, ein von einem Myon verursachtes Triggersignal in der korrekten Zeitscheibe der unmittelbar auslesenden Elektronik vorzufinden, wird als *zeitliche Triggereffizienz* ε_t des Zentralen Myon-Detektors bezeichnet.

Insbesondere der Einfluß der Driftzeit auf die zeitliche Triggereffizienz ist in dieser Hinsicht von Interesse, da dieser Anteil der Nachweisverzögerung - im Gegensatz zu den Flug- und Signalzeiten - nicht aus der Flugbahn des Myons ermittelt und korrigiert werden kann.

3.1.2 Analyse mit Hilfe kosmischer und Halo-Myonen

Der *wahre Eintreffzeitpunkt* t_0 der Myonen im Zentralen Myon-Detektor fällt im normalen Strahlbetrieb mit einer festen Phase der HERA clock zusammen, welche mit der Frequenz der Kollisionen von Teilchenpaketen oszilliert. Die Reaktion des Myon-Triggers muß jedoch relativ zu jeder Phase der HERA clock ermittelt werden, um den Einfluß der Nachweisverzögerung auf die zeitliche Effizienz des Myon-Triggers zu bestimmen. Dies erfordert, das gesamte Zeitintervall zwischen zwei Takten der HERA clock zu erfassen.

Ein feste Phasenbeziehung ist also zu vermeiden. Eine Möglichkeit besteht darin, die Bestimmung der zeitlichen Effizienz mit Myonen der kosmischen Strahlung oder Myonen aus den Halos der Teilchenstrahlen durchzuführen, die den Detektor in zufälligen Zeitabständen durchqueren.

Kosmische Myonen und Halo-Myonen

Die Myonen der Höhenstrahlung, welche die Erdoberfläche erreichen, sind sekundäre Zerfallsprodukte energiereicher Teilchen kosmischen Ursprungs. Diese (meist Protonen und leichte Atomkerne) treffen in einer Höhe von etwa 15 km und mehr auf Atomkerne in den oberen Schichten der Atmosphäre, was zur Bildung hadronischer Schauerkaskaden führt. Die hierbei erzeugten geladenen Mesonen, insbesondere Pionen, zerfallen in Myonen und Neutrinos. Der weitaus größte Teil der geladenen sekundären Teilchen der Höhenstrahlung auf Meeresniveau wird von auf diesem Wege entstehenden Myonen gebildet [23]. Die Intensität $I(\theta)$ der Myonen variiert mit dem Winkel θ zur Senkrechten gemäß

$$I(\theta) = I_0 \cos^2 \theta \tag{3.4}$$

für Myonen mit einem Impuls bis hin zu etwa 100 GeV/c [24].

Halo-Myonen hingegen haben ihren Ursprung in der unmittelbaren Umgebung des Protonenstrahls, entstehen also während des Strahlbetriebs. Bei der Wechselwirkung von Protonen aus den Teilchenpaketen mit der Apertur der Strahlröhre entstehen hierbei hochenergetische Pionen, welche wiederum in Myonen und Neutrinos zerfallen. Aufgrund des bewegten Schwerpunktsystems dieser Reaktionen werden die Myonen in Vorwärtsrichtung emittiert; der Winkel ihres Impulsvektors zur Strahlachse ist aus diesem Grunde sehr klein.

Die Daten kosmischer sowie von Halo-Myonen werden in separaten Runperioden gewonnen. Zum Nachweis kosmischer Myonen wird der Strahlbetrieb unterbrochen, während die Datenaufnahme des Experimentes aktiv bleibt. Voraussetzung für die Speicherung eines Ereignisses ist das Signal des Subtriggers ST49 (vergl. Abschn. 1.2.2), welches Treffer in allen vier Lagen der Vieldraht-Proportionalkammer CIP signalisiert. Die Datennahme für Halo-Myonen geschieht hingegen während des Strahlbetriebs. Der hier geforderte Subtrigger trägt die Bezeichnung ST63 und signalisiert sowohl mindestens eine Spur im Vorwärtigen Myon-System als auch ein Signal von einer der *Veto-Wände*. Deren Funktion ist in [1] beschrieben.

Ermittlung der Flugzeitkorrektur

Im Falle kosmischer Myonen kann der Eintreffzeitpunkt mit Hilfe der Daten der Zentralen Spurkammern mit hoher Genauigkeit ermittelt werden, sofern die Teilchenspur dort registriert wird (vergl. Abschn. 1.2.1). Bei den meist nahezu parallel zur Strahlachse verlaufenden Spuren der Halo-Myonen besteht die Möglichkeit der Feststellung von t_0 innerhalb der Kammern des Vorwärtigen Myon-Systems. Beide Subdetektoren liefern auch die für die Datennahme der entsprechenden Runperioden notwendigen Triggersignale.

Im Gegensatz zu den Myonen, welche bei $e^{\pm}p$-Kollisionen im normalen Strahlbetrieb entstehen, ist nun aber nicht der nominelle Wechselwirkungspunkt des H1-Experimentes deren Ursprung. Vielmehr treten kosmische wie Halo-Myonen von außerhalb in das Experiment ein und durchqueren folglich den Zentralen Myon-Detektor, bevor sie im Zentralen Spurkammersystem bzw. dem Vorwärtigen Myon-System registriert werden. Die Flugzeit, die währenddessen verstreicht, muß entsprechend berücksichtigt werden. Hierzu ist ein Korrekturterm t_{korr} zu ermitteln. Es ist für kosmische Myonen im Barrelbereich sowie in den äußeren Endkappen

$$t_{korr} = 2\,t_{Flug} = \frac{2\,S}{c} \qquad \text{für den oberen Bereich } (0 \leq \phi < \pi) \quad , \tag{3.5}$$

$$t_{korr} = 0 \qquad \text{für den unteren Bereich } (\pi \leq \phi < 2\,\pi) \tag{3.6}$$

und für Halo-Myonen in den inneren Endkappen

$$t_{korr} = \frac{S - z}{c} \qquad . \tag{3.7}$$

Ermittlung des relativen Eintreffzeitpunktes

Der relative Eintreffzeitpunkt ist definiert als die Dauer zwischen dem Durchgang eines Myons durch das Instrumentierte Eisen und dem Beginn der Zeitscheibe des dadurch verursachten Triggersignals. Um diesen zu ermitteln, müssen zunächst eine Reihe von Anforderungen an die Datenmenge gestellt werden. So sind Ereignisse auszuschließen, die Triggersignale von mehr als zwei Modulen umfassen. Die Spur eines kosmischen oder eines Halo-Myons, das den Zentralen Myon-Detektor durchquert, wird einen Eintritts- und einen Austrittspunkt in das bzw. aus dem Detektorvolumen aufweisen. Sind mehrere Module getroffen, ist nicht mit ausreichender Sicherheit gewährleistet, daß das verursachende Teilchen die richtige Identität besitzt. Die Überprüfung der Zahl der triggernden Module geschieht mittels Abfrage der BOS-Bank TSCD, welche die an die Triggerstufe L2 übermittelten Lagenkoinzidenzsignale der einzelnen Module verzeichnet (s. Abschn. 1.3.3). Alternativ kann auch das Auftreten simulierter Lagenkoinzidenzen gefordert werden[1]. Darüberhinaus soll der in der z-Driftkammer bzw. dem Vorwärtigen Myon-System gemessene absolute Eintreffzeitpunkt innerhalb eines gewissen Intervalls liegen ($-0.8 < t_{0,rel} < 0.2$ für kosmische und $-0.25 < t_{0,rel} < 1.4$ für Halo-Myonen in der Einheit von Zeitscheiben).

Ereignisse, die diese Kriterien erfüllen, werden im weiteren auf die Aktivität in jedem einzelnen Modul hin untersucht. Eine Berechnung des relativen Eintreffzeitpunktes für eine Spur innerhalb eines Moduls ist nur dann sinnvoll, wenn für dieses (d.h. den umgebenden Sektor) eine erfüllte Triggerbedingung vorliegt, d.h. sowohl ein t_0-Signal als auch ein Lagenkoinzidenzsignal in den Rohdaten der GPTP-Karten verzeichnet ist. Diese Information liegt nur für alle Module eines Sektors gemeinsam vor. Um sie nach einem

[1] Hierzu wurden die FORTRAN-Routinen ITSMTP [10] bzw. SIMEXP [22] verwendet, welche die Simulation der GPTP-Karten durchführen bzw. die Ergebnisse aus den common-Blöcken exportieren. Die Simulation geschieht durch Auswertung der Signale des Readout und hat den Vorteil, daß die Ergebnisse modulweise vorliegen im Gegensatz zu den echten Rohdaten der GPTP-Karten, die nur jeweils einen Sektor insgesamt betreffen.

einzelnen Modul aufzuschlüsseln, muß wiederum zur Verifikation das Ergebnis einer Simulation herangezogen werden.

Anschließend kann die Berechnung des relativen Eintreffzeitpunktes für geeignete Spuren erfolgen. Zu diesem Zweck wird zunächst die Zeitscheibe des ersten Eintreffens von Lagenkoinzidenz- und t_0-Signal ($t_{C,TGPP}$ und $t_{0,TGPP}$) in den GPTP-Karten ermittelt. In den meisten Fällen werden diese beiden Werte nicht übereinstimmen. Für die Triggereffizienz von Bedeutung ist jedoch das später eintreffende Signal, d.h. das Maximum von ($t_{C,TGPP}$, $t_{0,TGPP}$) ist zu betrachten. Von der Zeitscheibe, in der dieses Signal registriert wurde, ist nun der Wert des gemessenen Eintreffzeitpunktes t_0 zu subtrahieren und der Flugzeit-Korrekturterm t_{korr} aufzuaddieren. Dabei ist zu beachten, daß die Tiefe t_{TGPP} der pipelines der GPTP-Karten, d.h. die Position des wahren t_0 in den pipelines, abhängig ist von der Konfiguration der elektronischen Bauteile. Sie muß ermittelt und ebenfalls vom Resultat subtrahiert werden. Zusammenfassend errechnen sich die relativen Eintreffzeitpunkte gemäß

$$t_{0,rel} = \max(t_{C,TGPP}, t_{0,TGPP}) - t_0 - t_{TGPP} + t_{korr} \quad . \tag{3.8}$$

Die relativen Eintreffzeitpunkte werden für jedes Modul einzeln in Histogrammen, aufgetragen über einen Zeitraum von ± 3 Zeitscheiben, gezählt. Diese enthalten somit die Verteilung $N(t_{0,rel})$ derjenigen relativen Eintreffzeitpunkte, die zu einem Triggersignal in der Zeitscheibe 0 geführt haben (Abb. 3.2 a)). Sie kann zur Berechnung der Wahrscheinlichkeit, ein Triggersignal in der Zeitscheibe des tatsächlichen Eintreffens eines Teilchens im Myon-Detektor zu beobachten, herangezogen werden. Zu diesem Zweck muß sie geeignet normiert werden.

Berechnung der Wahrscheinlichkeitsverteilungen

Zur Normierung der Verteilung $N(t_{0,rel})$ ist es notwendig, alle relativen Eintreffzeitpunkte, die in dem von ihr abgedeckten Zeitfenster liegen, zu bestimmen. Dazu gehören auch diejenigen, die nicht selbst zu einem Triggersignal in der Zeitscheibe 0 geführt haben. Da nun $N(t_{0,rel})$ aus einer Vielzahl von Zeitscheiben in unterschiedlichen Ereignissen ermittelt wird, stellt sie eine Größe dar, die für jede beliebige Zeitscheibe repräsentativ ist; sie beschreibt ebenso die Lage und Häufigkeit der relativen Eintreffzeitpunkte, die zu einem Triggersignal in benachbarten Zeitscheiben führen, sofern sie dementsprechend zeitlich versetzt ausgewertet wird.

Hier liegt nun der Schlüssel für die geeignete Normierung von $N(t_{0,rel})$, denn die relativen Eintreffzeitpunkte, die kein Triggersignal in der Zeitscheibe 0 verursacht haben, können wiederum als zugehörig zu $N(t_{n,rel})$ einer benachbarten Zeitscheibe n betrachtet werden. Es müssen also die Verteilungen $N(t_{n,rel})$, die zeitlich entsprechend versetzte Kopien der ermittelten Verteilung $N(t_{0,rel})$ darstellen, für ein geeignetes Intervall von n um $n = 0$ aufaddiert werden (vergl. Abb. 3.2 b)). Wird wie im vorliegenden Fall die Verteilung innerhalb eines Zeitfensters von 290 ns ermittelt, so reicht es aus, ein Intervall $-2 \leq n \leq 2$ zu wählen. Der Breite der Verteilung $N(t_{n,rel})$ ist damit ausreichend Rechnung getragen.

Das auf diese Weise entstehende Histogramm dient als Normierung für $N(t_{0,rel})$. Insgesamt kann die oben beschriebene Vorgehensweise der Normierung folgendermaßen ma-

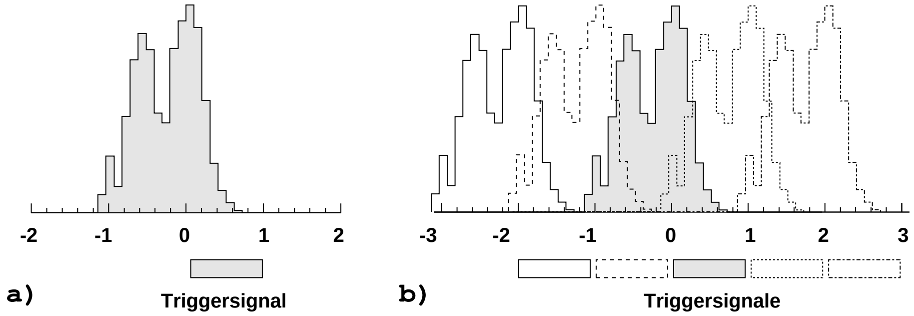

Abbildung 3.2: Zur Normierung der Verteilung $N(t_{0,rel})$. `a)` Ursprüngliche Verteilung. `b)` Bei der Addition der versetzten Verteilungen werden relative Eintreffzeitpunkte benachbarter Zeitscheiben in $N(t_{0,rel})$ quasi *hineingestreut.*

thematisch formuliert werden[2],

$$p(t_{0,rel}) = \frac{N(t_{0,rel})}{\sum_{t_{TGPP}=-n}^{t_{TGPP}=+n} N(t_{0,rel} + t_{TGPP})} \quad . \tag{3.9}$$

Die so erhaltene Verteilung $p(t_{0,rel})$ gibt für jeden Wert von $t_{0,rel}$ die Wahrscheinlichkeit an, daß das von einer zugehörigen Spur verursachte Triggerelement in der Zeitscheibe 0 zu liegen kommt. Im Idealfall hat sie die Form eines Rechtecks der Höhe 1 sowie der Breite einer Zeitscheibe und ist um $t_{0,rel} = 0$ zentriert. Tatsächlich aber kommt es zu einer gewissen Verschmierung der Idealform durch die Breite der Verteilung der Nachweisverzögerung; auch bewirkt die Nachweisverzögerung eine Verschiebung des Schwerpunktes der Verteilung. Die wahre Wahrscheinlichkeitsverteilung kann approximiert werden mit einer Funktion gemäß

$$\hat{\varepsilon}_t(t_{0,rel}) = \frac{1}{4} \cdot p_0 \cdot \mathrm{Erfc}\left(\frac{t_{0,rel} - \bar{t}_N - 0.5}{\sigma_N}\right) \cdot \mathrm{Erfc}\left(\frac{-t_{0,rel} - \bar{t}_N - 0.5}{\sigma_N}\right), \tag{3.10}$$

wobei $\mathrm{Erfc}(x)$ das komplementäre Gauß'sche Fehlerintegral bezeichnet:

$$\mathrm{Erfc}(x) = 1 - \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad . \tag{3.11}$$

Die freien Parameter $\bar{t}_N$ (mittlere Nachweisverzögerung), σ_N (Breite) und p_0 (Amplitude) können für jedes Modul mit Hilfe eines Minimierungsverfahrens angepaßt werden. Die so gewonnene *Effizienzfunktion* $\hat{\varepsilon}_t(t_{0,rel})$ gibt Aufschluß über die zeitliche Effizienz ε_t des Moduls, indem der Funktionswert an der Stelle $t_{0,rel} = 0$ ausgewertet wird,

$$\varepsilon_t = \hat{\varepsilon}_t(t_{0,rel} = 0) \quad . \tag{3.12}$$

Insgesamt existieren also zwei wesentliche Kategorien von Gründen, die zu einer Verminderung der zeitlichen Triggereffizienz eines Moduls führen. Einerseits kann es zu einer

[2] Entgegen der Darstellung in [10] findet eine zusätzliche Normierung des Nenners nicht statt.

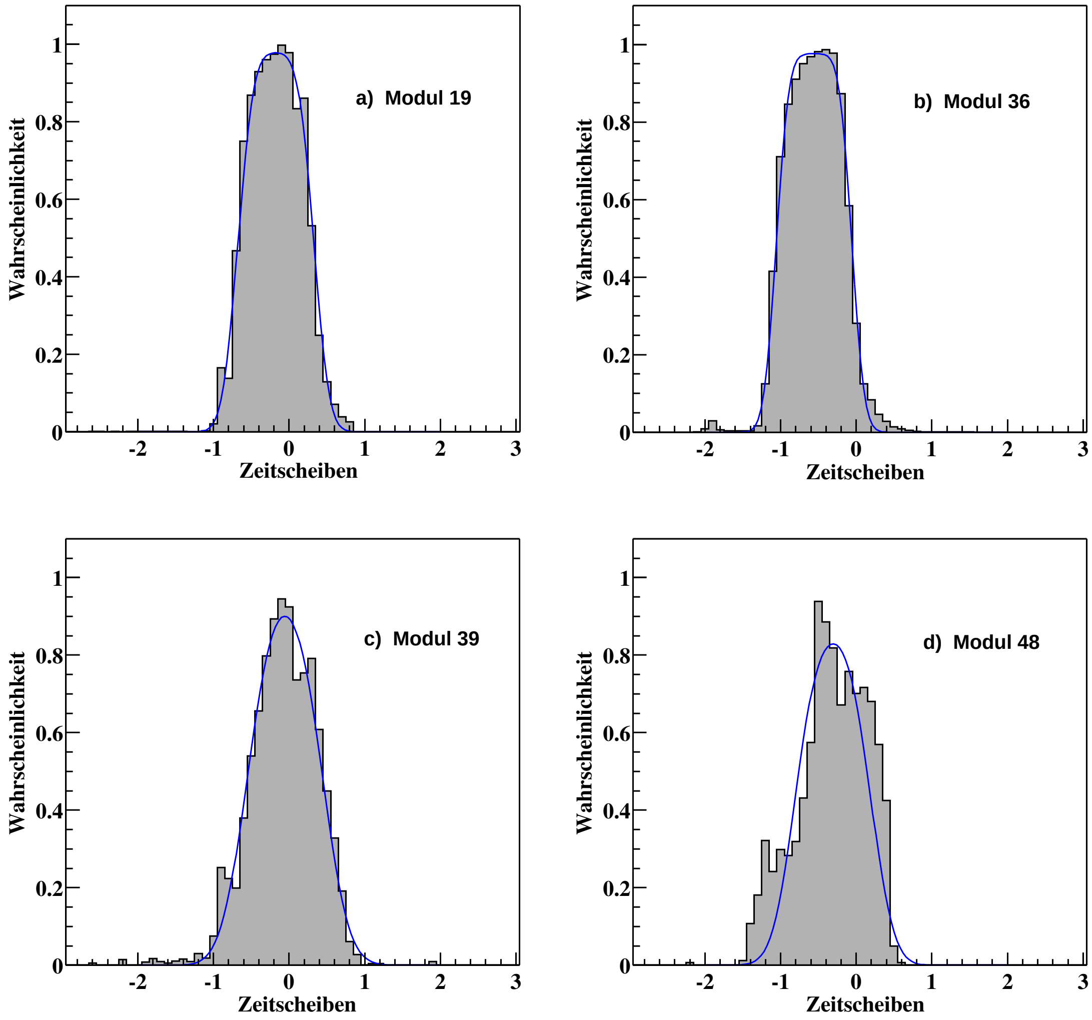

Abbildung 3.3: Die Wahrscheinlichkeitsverteilungen und die angepaßten Funktionen $\hat{\varepsilon}_t(t_{0,rel})$ für vier ausgewählte Module des Zentralen Myon-Detektors. `a)` Akzeptable Verteilung `b)` verschobene Verteilung `c)` verbreiterte Verteilung `c)` nicht modellmäßige Verteilung

Überlagerung von relativen Eintreffzeitpunkten kommen, die in benachbarten Zeitscheiben Triggersignale ausgelöst haben, mit der Verteilung für die Zeitscheibe 0. Im Falle solcher Ambiguitäten geht die Eindeutigkeit der Zuordnung verloren, und die Effizienz sinkt. Je größer die Breite σ_N von $N(t_{0,rel})$ ist, desto deutlicher tritt dieser Effekt in Erscheinung. Ein Beispiel hierfür zeigt die Abb. 3.3 `c)`.

Andererseits sind in manchen Fällen auch deutliche Verschiebungen der mittleren Nachweisverzögerung in negativer Richtung festzustellen. Daß die damit verbundenen weit zurückreichenden relativen Eintreffzeitpunkte überhaupt ein Triggersignal in der Zeitscheibe 0 verursachen können, ist auf den Einfluß der Synchronisations- und Resynchronisationsphasen zurückzuführen. Je später die Synchronisation eines Signals mit der HERA clock stattfindet, desto längere Driftzeiten sind möglich, ohne daß das Signal bereits der nachfolgenden Zeitscheibe zugeordnet wird. Der Funktionswert an der Stelle

$t_{0,rel} = 0$ befindet sich in diesem Falle auf der abfallenden Flanke des Graphen von Gl. (3.10) - die Effizienz sinkt drastisch (Abb. 3.3 `b)`).

Ein mittels der oben beschriebenen Methode bestimmter Effizienzwert erfaßt lediglich den rein zeitlichen Aspekt der Triggereffizienz, der durch die Verteilung der Nachweisverzögerung und die Phasen der Synchronisation mit der HERA clock bestimmt ist. Bleibt ein zu erwartendes Triggersignal völlig aus oder entsteht eine erhebliche Verzögerung erst im Verlauf der weiteren Verarbeitungskette, so kann dies keinen Einfluß haben auf das Verhalten von $p(t_{0,rel})$, wie es hier ermittelt wurde. Ein Beispiel für ein akzeptables Verhalten in Bezug auf die zeitliche Effizienz zeigt das Modul 19 (Abb. 3.3 `a)`), welches somit stellvertretend für den Großteil der Module steht. Daß in solchen Fällen die Triggereffizenz insgesamt dennoch sehr schlecht ausfallen kann, wird im nächsten Abschnitt deutlich.

3.1.3 Die Klasse `TMuSpecialLoop`

Die Klasse `TMuSpecialLoop`, die zum Umfang der in Kapitel 2 beschriebenen Klassenbibliothek gehört, kann zur Berechnung der zeitlichen Triggereffizienzen mittels kosmischer bzw. Halo-Myonen eingesetzt werden. `TMuSpecialLoop` benötigt hierzu mit einer speziellen Methode konvertierte Rohdatendateien, deren Erzeugung in Anhang A beschrieben ist. Aufgrund der Natur beider Myonquellen - um die Senkrechte verteilter Einfall kosmischer bzw. weitgehend zur Strahlachse parallele Flugbahnen der Halo-Myonen - sind diese jeweils nur für die Untersuchung einer begrenzten Zahl von Modulen verwendbar. Kosmische Myonen durchqueren meist den Barrelbereich des Zentralen Myon-Detektors und die äußeren Endkappen, Halo-Myonen demgegenüber bevorzugt die inneren Endkappen, also die Module 6 bis 11 sowie 54 bis 59. Insgesamt kann somit die zeitliche Effizienz für jedes Modul ermittelt werden. Die Berechnung ist im übrigen für kosmische und Halo-Myonen, d.h. für die entsprechenden Module, getrennt durchzuführen, da die Information betreffend den wahren Eintreffzeitpunkt jeweils aus einer anderen BOS-Bank entnommen werden muß. Die Ergebnisse können anschließend manuell kombiniert werden.

`TMuSpecialLoop` wird zunächst die Flugzeiten der Myonen bestimmen und anschließend deren relativen Eintreffzeitpunkte. Die Klasse ist modular aufgebaut, alle Aufgaben sind jeweils innerhalb einer eigenen Methode realisiert. Die gemessenen Eintreffzeitpunkte stammen im Falle kosmischer Myonen aus den BOS-Bänken CJCC und CJGR, welche die Ergebnisse der Spurrekonstruktion innerhalb des Zentralen Spurkammersystems beinhalten. Werden Halo-Myonen zur Bestimmung herangezogen, findet die Information aus der BOS-Bank MTCT Verwendung, die die Ergebnisse der Rekonstruktion im Vorwärtigen Myon-Spektrometer aufnimmt.

Liegen die Verteilungen der relativen Eintreffzeitpunkte für die einzelnen Module vor, so errechnet eine weitere Routine die zur Normierung notwendigen Histogramme und schließlich die Wahrscheinlichkeitsverteilungen mit den angepaßten Funktionen $\hat{\varepsilon}_t(t_{0,rel})$. Alle weiteren auszugebenden Daten können hieraus abgeleitet werden. `TMuSpecialLoop` ermittelt die zeitliche Triggereffizienz eines jeden Moduls durch Bestimmung des Funktionswertes der angepaßten $\hat{\varepsilon}_t(t_{0,rel})$ an der Stelle $t_{0,rel} = 0$ und visualisiert alle 64 Werte in einem Histogramm namens `Efficiency` (s. Abb. 3.4). Auch werden die statistischen Fehler der Effizienzwerte berechnet. Sei (für ein beliebiges Modul) k die Zahl der relativen Eintreffzeitpunkte an der Stelle $t_{0,rel} = 0$, die zu einem Triggersignal in der Zeitscheibe 0

Histogramm	Inhalt
`Efficiency`	Die Werte der zeitlichen Effizienz aller vierundsechzig Module
`ChiSquare`	Die Qualität der Anpassung von $\hat{p}_t(t_{0,rel})$ (χ^2 pro Freiheitsgrad), ebenfalls für alle Module[4]
`ChiSquareDist`	Die Verteilung von χ^2 pro Freiheitsgrad[4]
`TZero`	Die Lage der mittleren relativen Eintreffzeitpunkte aller vierundsechzig Module
`SingleEfficiency_0` ... `SingleEfficiency_63`	Die Wahrscheinlichkeitsverteilungen aller Module mit den angepaßten Funktionen $\hat{\varepsilon}_t(t_{0,rel})$

Tabelle 3.1: Übersicht der von `TMuSpecialLoop` erzeugten Ergebnishistogramme. Konformitäts- und Bedingungshistogramme sind nicht aufgeführt.

geführt haben, und n die Zahl aller relativen Eintreffzeitpunkte an der Stelle $t_{0,rel} = 0$. Dann errechnet sich der statistische Fehler $\delta\varepsilon_t$ des Effizienzwertes zu

$$\delta\varepsilon_t = \frac{\delta k}{n} = \frac{1}{n}\sqrt{k\left(1 - \frac{k}{n}\right)} \quad . \tag{3.13}$$

Zur Erfassung der zeitlichen Effizienz einzelner Triggerelemente wird weiterhin das Histogramm `ElementEfficiency` generiert. In diesem Falle sind die Effizienzen der entsprechenden Module jeweils durch Mittelwertbildung zur Gesamteffizienz der fünf einfachen[3] Triggerelemente zusammengefaßt.

Obwohl `TMuSpecialLoop` die Anpassung der Parameter in der Funktion $\hat{\varepsilon}_t(t_{0,rel})$ an die einzelnen Wahrscheinlichkeitsverteilungen bis zu zwanzigfach iterativ wiederholt, um die Qualität weitgehend zu optimieren, wird nicht für jedes Modul und in jedem Fall eine perfekte Anpassung erreicht werden können. Der Grund hierfür liegt in der nicht immer dem Modell entsprechenden Form der Wahrscheinlichkeitsverteilungen $p(t_{0,rel})$ (vergl. Abb. 3.3 `d)`). Um die Güte der errechneten Effizienzwerte schnell erfassen zu können, stellt `TMuSpecialLoop` zwei weitere Histogramme zur Verfügung. Beide enthalten die Werte des Parameters χ^2, der bei der Anpassung der Funktionen $\hat{\varepsilon}_t(t_{0,rel})$ an die Wahrscheinlichkeitsverteilungen ermittelt und minimiert wird[4]. Das Histogramm `ChiSquare` zeigt die Werte von χ^2 für alle 64 Module und kann somit deckungsgleich mit `Efficiency` dargestellt werden. Das Histogramm `ChiSquareDist` hingegen zeigt die Verteilung der Werte von χ^2 innerhalb eines Intervalls von $\chi^2 = 0$ bis $\chi^2 = 1,2$. Ein weiteres Histogramm namens

[3]Hierbei handelt es sich um die Triggerelemente Mu_BIEC, Mu_BOEC, Mu_Bar, Mu_FIEC und Mu_FOEC (s. Tabelle 1.5), die bereits einzelne erfüllte Triggerbedingungen signalisieren.

[4]Alle angegebenen Werte sind als χ^2 *pro Freiheitsgrad* zu interpretieren, wie sie von den Methoden der verwendeten ROOT-Klassen automatisch errechnet wurden. Aufgrund des hierbei benutzten Verfahrens fallen die Werte von χ^2 zu klein aus und sind hauptsächlich qualitativ von Bedeutung.

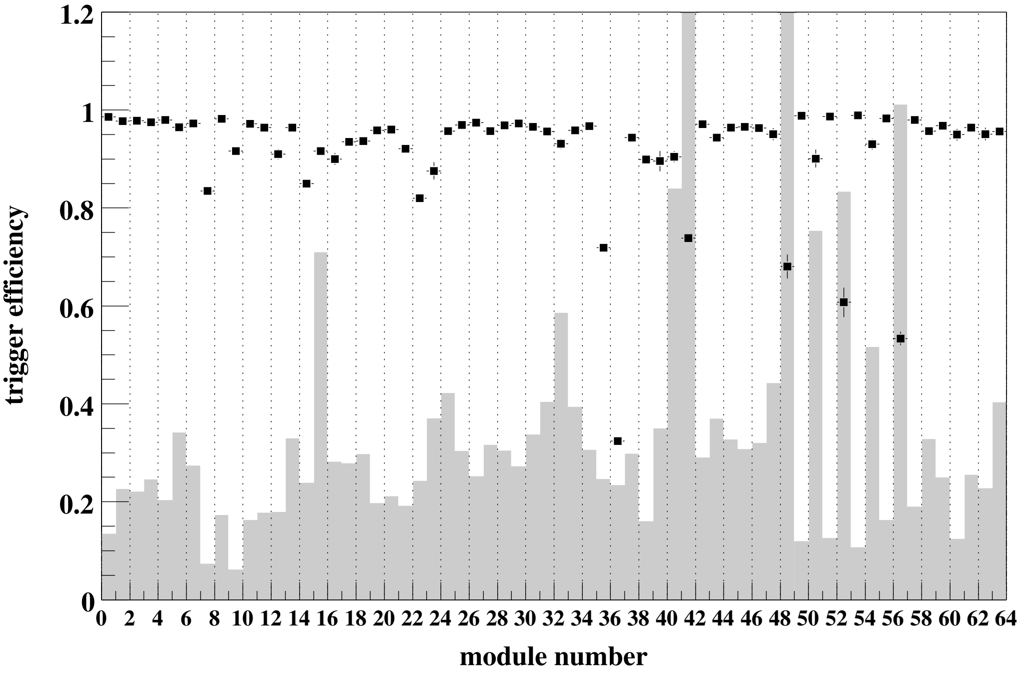

Abbildung 3.4 : Die zeitliche Effizienz der Module des Zentralen Myon-Detektors. Verwendet wurden die Daten kosmischer und von Halo-Myonen (1.776.700 bzw. 1.212.470 Ereignisse), detektiert während des Jahres 2000. Grau hinterlegt ist jeweils die Qualität der Anpassung von $\hat{\varepsilon}_t(t_{0,rel})$ (χ^2 pro Freiheitsgrad)[4].

`TZero` verzeichnet für jedes Modul die Lage des mittleren relativen Eintreffzeitpunktes (Abb. 3.5).

Alle Histogramme werden in der Ausgabedatei in einem Verzeichnis namens `SpecialHistograms` abgelegt. Darüberhinaus erzeugt `TMuSpecialLoop` innerhalb dieses Verzeichnisses die beiden Unterverzeichnisse `NumeratorHistos` bzw. `DenumeratorHistos`, welche die jeweils 64 Histogramme der Verteilungen $N(t_{0,rel})$ bzw. die Normierungshistogramme beinhalten. Einen Überblick der Struktur des Verzeichnisses `SpecialHistograms` gibt Abb. 3.6.

Wie in der Abb. 3.4 zu erkennen ist, kann die zeitliche Triggereffizienz der weitaus meisten Module durchaus als zufriedenstellend bezeichnet werden. Bei den Modulen mit den Nummern 35, 36, 41, 48, 52 sowie 56 sind jedoch Mängel erkennbar. Im Falle der vier letztgenannten Module ist dies offensichtlich zu einem gewissen Teil auf die nicht dem Modell entsprechende Form der Wahrscheinlichkeitsverteilung $p(t_{0,rel})$ zurückzuführen (vergl. Abb. 3.3 `d)`). Der Grund für diese Abweichungen ist nicht unmittelbar zu erkennen.

Zumindest bei den Modulen 35, 36 und 56 liegt die Ursache der verminderten zeitlichen Triggereffizienz auch in einer deutlichen zeitlichen Verschiebung der mittleren relativen Eintreffzeitpunkte in negativer Richtung. Die Abb. 3.5 und die Abbildungen der Wahrscheinlichkeitsverteilungen $p(t_{0,rel})$ in Anhang C geben hierüber Auskunft. Auch läßt sich der Abb. 3.5 entnehmen, daß die mittleren relativen Eintreffzeitpunkte der Module allgemein um $t_{0,rel} \approx -0,2$ verteilt sind.

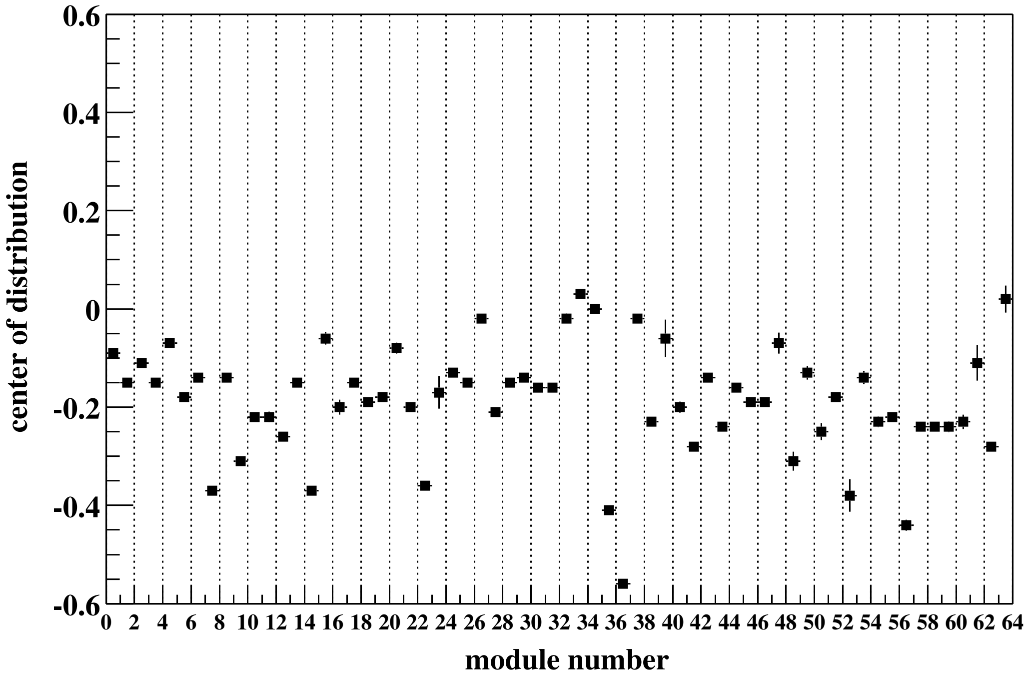

Abbildung 3.5: Die Lage der mittleren relativen Eintreffzeitpunkte für die Module des Zentralen Myon-Detektors in Einheiten von Zeitscheiben. Daten wie in der vorangehenden Abbildung.

3.2 Die Bestimmung der Hardware-Effizienz des Myon-Triggers

Neben einer zeitlich bedingten Reduzierung der Triggereffizienz aufgrund der Verteilung der Nachweisverzögerung können auch hardwarebedingte Fehlfunktionen die Effizienz des Myon-Triggers beeinflussen. Dabei wird beobachtet, daß Triggersignale ausbleiben, die zwingend hätten erzeugt werden müssen. Allerdings kann es im Verlauf der Signalverarbeitung auch zu erheblichen zeitlichen Verschiebungen der Triggersignale kommen. Diese Effekte werden durch die Hardware-Effizienz ε_h erfaßt, also der Wahrscheinlichkeit einer korrekten Erzeugung und Verarbeitung der Triggersignale durch die Elektronik des Myon-Triggers:

$$\varepsilon_h = \frac{\text{Zahl der korrekt erzeugten Triggersignale}}{\text{Zahl der erfüllten Triggerbedingungen}} \tag{3.14}$$

3.2.1 Effizienzbestimmung mit Hilfe der Triggerelemente

Wie bereits in Abschn. 1.3 beschrieben, liegt die Quelle der dem Myon-Trigger zur Verfügung stehenden Signale in der unmittelbar auslesenden Elektronik, wo sie aus den jeweils ersten Einträgen der pipelines in die Triggerelektronik gespeist werden. Da die Synchronisation mit der HERA clock an dieser Stelle bereits stattgefunden hat, ist die Information betreffend den Einfluß der Nachweisverzögerung nicht mehr in den Triggersi-

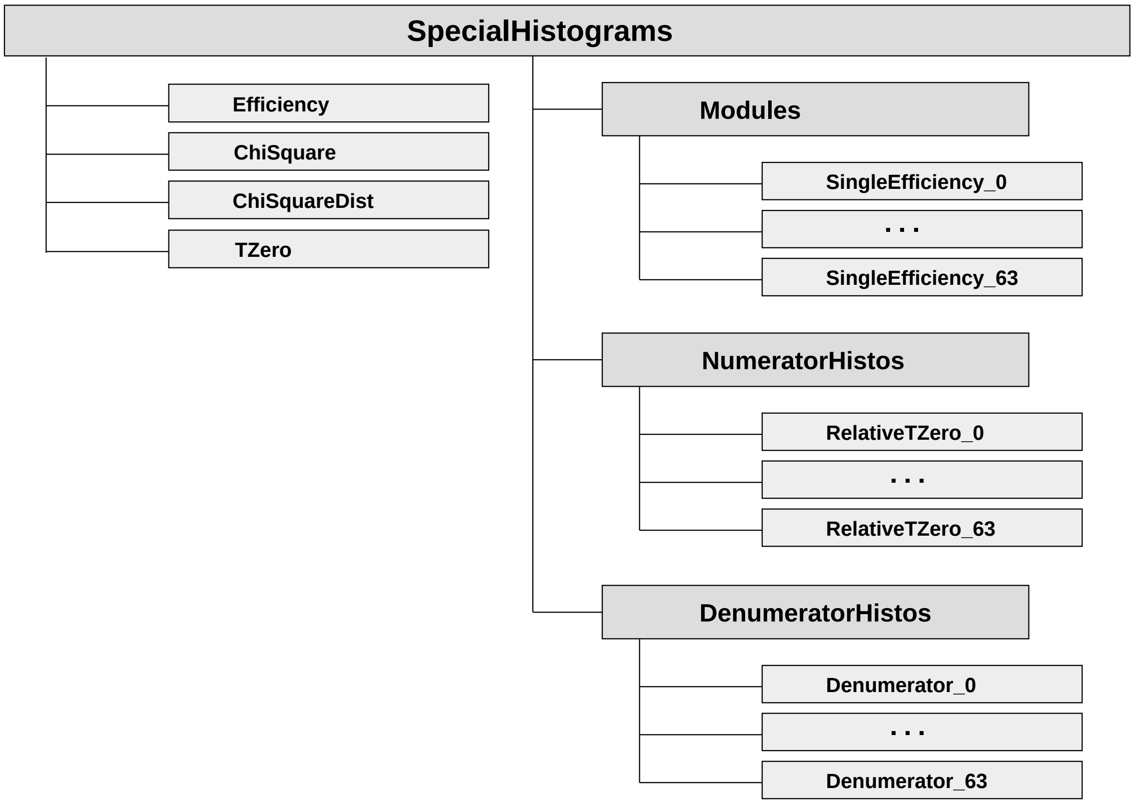

Abbildung 3.6: Die von `TMuSpecialLoop` in der Ausgabedatei angelegte Verzeichnisstruktur und die erzeugten Histogramme.

gnalen vorhanden. Vielmehr sind sie mit den Signalen des Readout identisch, schließlich stammen sie aus den gleichen pipelines. Diese Eigenschaft kann nun verwendet werden, um den Einfluß der nachfolgend auswertenden Elektronik des Myon-Triggers auf die Effizienz zu bestimmen.

Das Auslösen eines Triggersignales in einer gewissen Zeitscheibe n durch ein Modul des Zentralen Myon-Detektors ist an zwei Bedingungen geknüpft. Auf der einen Seite muß die entsprechende Lagenkoinzidenzbedingung erfüllt (s. Tab. 1.3), andererseits zumindest ein Draht in der Zeitscheibe n getroffen worden sein. Die Dehnung der Signale in den layer boards (vergl. Abb. 1.7) bringt es dann mit sich, daß auch in der Zeitscheibe $n+1$ getroffene Lagen zur Lagenkoinzidenz in n beitragen können. Beide Bedingungen lassen sich nun mit Hilfe der Signale des Readout simulieren.

Die zur Simulation nötige Information stammt aus der BOS-Bank IRWE[5], welche für ein Ereignis jeden getroffenen Draht mit Modul-, Lagen-, Element- und Drahtnummer verzeichnet. Auch ist die Zeitscheibe, in welcher das Signal des Drahtes die pipeline der zugehörigen unmittelbaren Ausleseelektronik erreicht hat, abfragbar. Diese kann eine der Zeitscheiben des Readout (-1, 0, 1, 2) sein.

Verglichen werden die Ergebnisse der Simulation mit den acht tatsächlich an L1 übermittelten Triggerelementen (s. Tab. 1.5). Sie sind verfügbar in der BOS-Bank

[5]IRWE: *iron response wire event*

TEL1(0)[6] zusammen mit den weiteren 192 Triggerelementen anderer Detektoren des H1-Experimentes, und zwar innerhalb eines Zeitfensters von ± 32 Zeitscheiben. Da die Simulation jeweils für ein einzelnes Modul durchgeführt werden kann, jedes der Triggerelemente aber eine ganze Gruppe von Modulen umfaßt, muß die Analyse auf Ereignisse beschränkt werden, in denen nur ein einzelnes Modul pro Triggersektor ein Signal erzeugt hat. Die Zuordnung des Triggerelementes zu einem Modul bleibt somit eindeutig. Darüberhinaus kann das Simulationsergebnis für die Zeitscheibe n auch mit den Triggerelementen anderer Zeitscheiben verglichen werden. Zeitliche Verschiebungen, die immer nur in Einheiten ganzer Zeitscheiben erfolgen können, sind somit erfaßbar.

3.2.2 Die Klasse `TMuTEL1Loop`

Eine Berechnung der Hardware-Effizienz der Module des Myon-Detektors ermöglicht die ausführende Klasse `TMuTEL1Loop`. Als auszuwertende Rohdatenmenge können die Aufzeichnungen normaler Runperioden Verwendung finden, eine Beschränkung auf kosmische oder Halo-Myonen ist nicht notwendig. Im vorliegenden Fall bilden die Rohdaten aus einem Zeitraum von April 2000 bis Ende August 2000 die Datenbasis aller Hardware-Effizienzen. Ein Viertel des Datenbestandes[7] dieser Periode fand Eingang in die Analyse.

`TMuTEL1Loop` bedient sich der Methoden der Basisklasse `TMuIRWE`, welche auf der Grundlage der BOS-Bank IRWE verschiedene Aussagen über das zu erwartende Triggerverhalten der einzelnen Module treffen. So wird zunächst ermittelt, ob für eine bestimmte Zeitscheibe n eine Lagenkoinzidenzbedingung in einem Modul vorlag. Um der Signaldehnung Rechnung zu tragen, werden dabei alle diejenigen Drähte berücksichtigt, die in den Zeitscheiben n und $n+1$ einen Treffer verzeichnen. Auch kann festgestellt werden, wie viele und welche Module innerhalb eines Sektors eine solche Triggerbedingung erfüllen. Der Wert von n ist beschränkt auf das Intervall [-1, 1], da der Readout nur Signale aus vier Zeitscheiben zur Verfügung stellt.

Ist nun für ein Modul eine erfüllte Triggerbedingung festgestellt worden, und trifft dies auf kein weiteres Modul im gleichen Triggersektor zu, so ist das Auftreten des entsprechenden Triggerelementes zu erwarten. Bleibt dieses aus oder erscheint mit zeitlicher Verschiebung, kann das verursachende Modul eindeutig bestimmt werden. Zu diesem Zweck werten die Methoden der Basisklasse `TMuTEL1` die Variablen der BOS-Bank TEL1(0) aus und liefern die Information, ob bzw. wann das Triggerelement aufgezeichnet wurde. Innerhalb der Klasse `TMuTEL1Loop` geschieht die Berechnung der Hardware-Effizienzen durch Zählung derjenigen Ereignisse, in denen in einem Modul eine Triggerbedingung vorlag und das Triggerelement auch tatsächlich (ggf. in einer bestimmten Zeitscheibe) beobachtet wurde. Diese Werte sind zu normieren auf die Zahl der Ereignisse, in denen insgesamt eine Triggerbedingung in diesem Modul vorlag. Beide Zählungen werden separat in zwei verschiedenen Histogrammen durchgeführt, welche jeweils alle 64 Module berücksichtigen.

Um verschiedene Aspekte der Beeinflussung der Triggereffizienz durch die elektronischen Bauteile gleichermaßen darstellen zu können, erzeugt `TMuTEL1Loop` standardmäßig eine ganze Reihe von Effizienzhistogrammen. Zunächst findet sich eine Gruppe von Hi-

[6]TEL1: *trigger elements level 1*
[7]Jedes vierte Rohdatenband wurde konvertiert.
[8]Alle Triggerbedingungen werden detektiert anhand der Signale des Readout.

Histogramm	Inhalt
`TEL1EffHistoM2`	Die Wahrscheinlichkeit eines Triggerelementes in Zeitscheibe -2, ausgelöst durch eine erfüllte Triggerbedingung[8] in Zeitscheibe 0
`TEL1EffHistoM1`	Gleiches für ein Triggerelement in Zeitscheibe -1
`TEL1EffHisto0`	Gleiches für ein Triggerelement in Zeitscheibe 0
`TEL1EffHisto1`	Gleiches für ein Triggerelement in Zeitscheibe 1
`TEL1EffHisto2`	Gleiches für ein Triggerelement in Zeitscheibe 2
`TEL1EffHistoALL`	Die Wahrscheinlichkeit eines Triggerelementes in irgendeiner Zeitscheibe, ausgelöst durch eine erfüllte Triggerbedingung in Zeitscheibe 0
`TEL1EffHistoR3BC`	Die Wahrscheinlichkeit eines Triggerelementes in Zeitscheibe 0, ausgelöst durch eine erfüllte Triggerbedingung in einer der Zeitscheiben -1, 0 oder 1
`BIEC` ... `2_FIoOEC`	Die zeitliche Verteilung je eines der acht Triggerelemente in den Zeitscheiben -2 bis 2

Tabelle 3.2: Übersicht der von `TMuTEL1Loop` erzeugten Effizienz- und Verteilungshistogramme. Die Effizienzhistogramme umfassen vierundsechzig unabhängige Einträge für alle Module. Konformitäts- und Bedingungshistogramme sind nicht aufgeführt.

stogrammen, welche die Wahrscheinlichkeit beinhalten, ein für Zeitscheibe 0 zu erwartendes Triggerelement jeweils in den Zeitscheiben -2 bis 2 aufzufinden (i. d. Reihenfolge sind dies die Histogramme `TEL1EffHistoM2`, `TEL1EffHistoM1`, `TEL1EffHisto0`, `TEL1EffHisto1` und `TEL1EffHisto2`). Als zusätzliche Bedingung wird gefordert, daß kein Triggerelement in der Zeitscheibe 0 vorhanden war (selbstverständlich mit Ausnahme von `TEL1EffHisto0`, s. Abb. 3.7).

Die Wahrscheinlichkeit eines völligen Fehlens eines zu erwartenden Triggerelementes kann dem Histogramm `TEL1EffHistoALL` entnommen werden (Abb. 3.7). Erfaßt wird hier die Häufigkeit von erfüllten Triggerbedingungen in jeweils einem Modul in Verbindung mit einem Triggerelement in *irgendeiner* der $\pm$ 32 Zeitscheiben der BOS-Bank TEL1(0). Zur Normierung wird die Zahl der erfüllten Triggerbedingungen pro Modul verwendet.

Ein weiteres Effizienzhistogramm namens `TEL1EffHistoR3BC` verdeutlicht das Ergebnis einer abweichenden Zählung. Ist in einem Modul in Zeitscheibe -1, 0 oder 1 eine Triggerbedingung erfüllt worden und findet sich das Triggerelement in der Zeitscheibe 0, so wird dieser Fall im Konformitätshistogramm gezählt. Als Normierung dient nun die Häufigkeit des Vorhandenseins einer erfüllten Triggerbedingung in einer der Zeitscheiben -1, 0 oder 1.

Neben den Effizienzhistogrammen ist auch eine Übersicht der zeitlichen Verteilung der Triggerelemente standardmäßig Teil der Ergebnisse der Klasse `TMuTEL1Loop`. Für jedes Triggerelement wird die Häufigkeit seines Auftretens, aufgeschlüsselt nach den Zeitscheiben -2 bis 2, gezählt und in ein Histogramm eingetragen. Es werden acht solcher Histogramme generiert, deren Namen sich an den Bezeichnungen der acht Triggerelemente orientieren (vergl. Tab. 1.5 und Abb. 3.8).

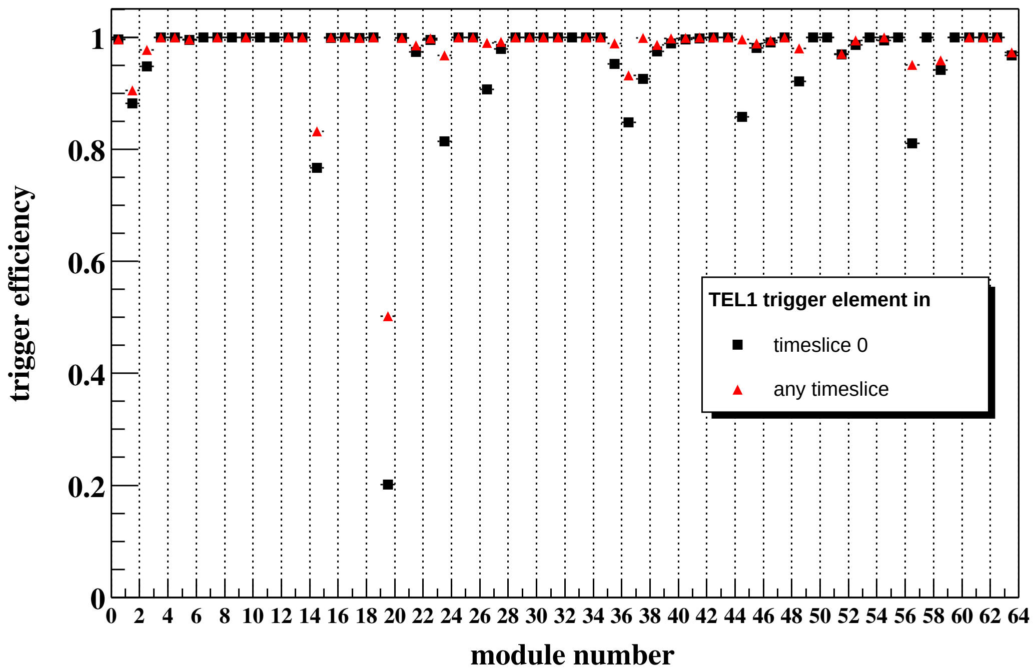

Abbildung 3.7: Die Wahrscheinlichkeiten für die vierundsechzig Module, ein in Zeitscheibe 0 zu erwartendes Triggerelement in Zeitscheibe 0 (Quadrate) bzw. überhaupt (Dreiecke) vorzufinden.

Aufgrund der Ergebnisse der Analyse mit Hilfe der Triggerelemente wird nun deutlich, daß die Hardware-Effizienz der Module deutlich von der zeitlichen Triggereffizienz zu unterscheiden ist. Abb. 3.7 zeigt eine Überlagerung der Histogramme `TEL1EffHisto0` bzw. `TEL1EffHistoALL`; es sind dies die Wahrscheinlichkeiten, ein Triggerelement in Zeitscheibe 0 bzw. in irgendeiner Zeitscheibe vorzufinden, sofern in Zeitscheibe 0 eines zu erwarten war. Da für eine korrekte Funktion des Triggersystems während normaler Runperioden das Vorhandensein des Triggerelementes in der Zeitscheibe des Wechselwirkungszeitpunktes entscheidend ist, muß erstere als die eigentliche Hardware-Effizienz betrachtet werden. Eine Reihe von Modulen weist in mindestens einem der beiden Fälle deutliche Fehlfunktionen auf. Insbesondere Modul 19 liefert etwa 50% der Triggerelemente überhaupt nicht an die Triggerstufe L1 ab, die aufgrund der Signale des Readout eigentlich hätten erzeugt werden müssen. Die zeitliche Effizienz von Modul 19 hingegen ist durchaus zufriedenstellend (s. Abschn. 3.1). Auf die Gründe für dieses Verhalten sowie mögliche Ansätze zur Lösung des Problems wird weiter unten eingegangen.

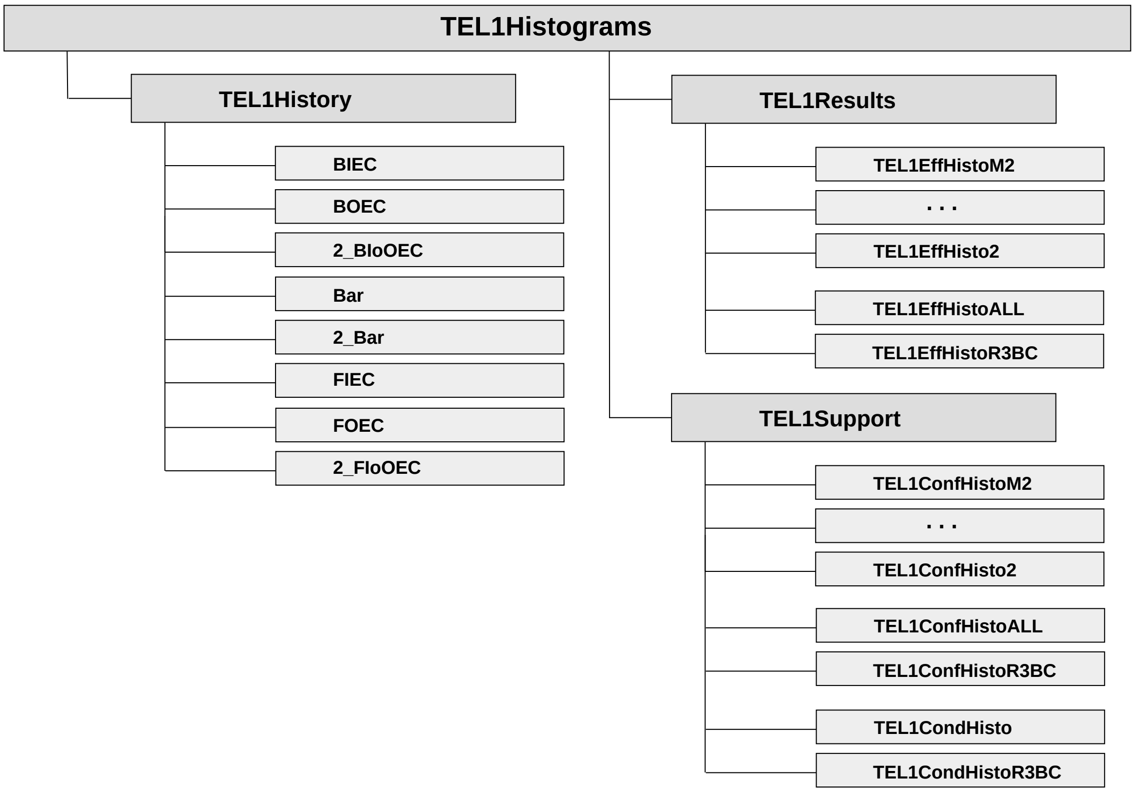

Abbildung 3.8: Die Histogramme innerhalb der Verzeichnisstruktur, die `TMuTEL1Loop` in der Ausgabedatei erzeugt.

3.2.3 Von L4 verifizierte Triggerelemente

Von den vom Myon-Detektor mit einem oder mehreren Triggerelementen bedachten Ereignissen ist ein gewisser Teil als physikalisch nicht interessant einzustufen[9]. Bei diesen handelt es sich in den meisten Fällen um die Aufzeichnung der Spuren von Strahl-Restgas-Wechselwirkungen und kosmischer Myonen. Auch bilden die Spuren von Myonen aus den Halos der Teilchenstrahlen u.a. solche Ereignisse. Soll die Analyse auf Daten physikalisch relevanter Prozesse beschränkt werden, so kann die Bewertung durch die Triggerstufe L4 als Kriterium hinzugezogen werden.

Wie im ersten Kapitel erläutert, erzeugt L4 einen Satz von 128 validierten Subtriggern für die Zeitscheibe 0, die in der BOS-Bank YTL4 für jedes Ereignis abgerufen werden können. Um die Rate der in die Berechnung eingehenden kosmischen oder Halo-Myonen erheblich zu verringern, können nun validierte Subtrigger gefordert werden, die nicht auf Triggerelementen des Myon-Systems basieren. In einem solchen Fall muß also weitere Aktivität an anderer Stelle im Detektor beobachtet worden sein. Konsequenterweise sollten mehrere solcher validierter Subtrigger in die Analyse einfließen, was ebenfalls die statistische Genauigkeit der Aussage erhöht.

[9]Diese Bewertung hängt genau genommen vom Zweck der Runperiode ab. Hier sind normale Runperioden gemeint.

3.2.4 Die Klasse `TMuYTL4Loop`

Für die Durchführung des oben beschriebenen Verfahrens steht die Klasse `TMuYTL4Loop` als Teil der Klassenbibliothek zur Verfügung. Sie bedient sich derjenigen 21 von L4 validierten Subtriggern, welche die höchsten Raten aufweisen (für eine detaillierte Aufzählung s. [22]). Die Ermittlung der Hardware-Effizienz geschieht analog zu derjenigen im vorangegangenen Abschnitt; jedoch besteht für die Zählung eines Ereignisses sowohl im Konformitäts- als auch im Bedingungshistogramm die zusätzliche Bedingung, daß mindestens einer der erwähnten 21 validierten Subtrigger vorzufinden war. Formal drückt sich diese Forderung in einem zusätzlichen Faktor zur Hardware-Effizienz ε_h aus,

$$\varepsilon_h' = \varepsilon_h \cdot \varepsilon_{ver} \quad , \tag{3.15}$$

mit der *Verifikationseffizienz* ε_{ver}. Diese wird bestimmt durch das Verhältnis der Zahl derjenigen Ereignisse, in welchen L4 einen der geforderten unabhängigen validierten Subtrigger geliefert hat, zur Gesamtzahl aller untersuchten Ereignisse.

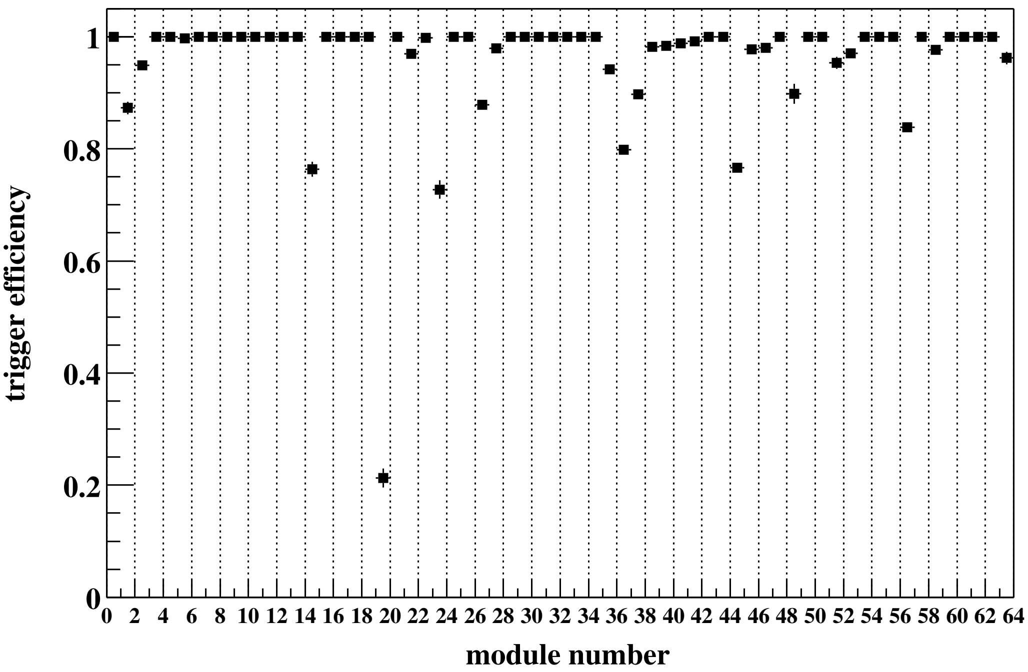

Abbildung 3.9: Die von der Klasse `TMuYTL4Loop` ermittelten Hardware-Effizienzen ε_h' auf der Grundlage von Triggerelementen in Verbindung mit von L4 validierten Subtriggern.

Ist `TMuYTL4Loop` aktiviert worden, so erzeugt das Objekt innerhalb der Ausgabedatei ein Verzeichnis namens `YTL4Histograms`. Darin werden zwei Unterverzeichnisse angelegt, `YTL4Results` mit dem Effizienzhistogramm `TrigEff_L4` sowie `YTL4Support`, welches das Konformitäts- sowie das Bedingungshistogramm enthält (s. Abb. 3.10).

Das Ergebnis der Berechnung ist in Abb. 3.9 aufgeführt. Nur wenige Module weisen unter der zusätzlichen Bedingung eine veränderte Hardware-Effizienz auf, wie der direkte

Histogramm	Inhalt
TrigEff_L4	Die Wahrscheinlichkeit eines Triggerelementes in Zeitscheibe 0 in Koinzidenz mit einem der von L4 validierten Subtrigger, ausgelöst durch eine erfüllte Triggerbedingung[8] in Zeitscheibe 0

Tabelle 3.3: Das von `TMuYTL4Loop` erzeugte Effizienzhistogramm.

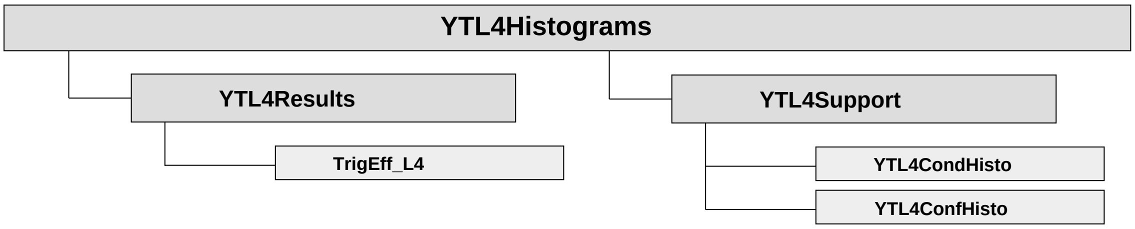

Abbildung 3.10: Die Struktur des von `TMuYTL4Loop` in der Ausgabedatei erzeugten Verzeichnisbaums mit den drei Histogrammen.

Vergleich mit Abb. 3.7 (Quadrate) zeigt. So fällt die Effizienz für die Module 23, 36 und 44 um 12% bzw. 6% niedriger aus. Diese Module besitzen offensichtlich eine geringfügig höhere Triggereffizienz bezüglich des Nachweises kosmischer Myonen, was mit deren im Mittel größeren Impulsen begründet werden kann. Ansonsten ist ε_{ver} kaum von 1 verschieden.

Die von `TMuYTL4Loop` ermittelte Effizienz ε'_h eines Moduls kann nun als die eigentliche Hardware-Effizienz betrachtet werden. Sie bezeichnet die Wahrscheinlichkeit, ein anhand der Signale des Readout in der Zeitscheibe 0 zu erwartendes Triggerelement auch tatsächlich vorzufinden. Durch den Anteil der L4-Verifikation ist sie darüberhinaus weitgehend auf physikalisch interessante Ereignisse beschränkt. Allerdings können mit `TMuYTL4Loop` keine der Zeitscheibe 0 vorausgehende oder nachfolgende Zeitscheiben betrachtet werden, wie dies im letzten Abschnitt bei den Ergebnissen der Analyse mit der Klasse `TMuTEL1Loop` vorgestellt wurde.

3.3 Die Kombination von zeitlicher und Hardware-Effizienz

3.3.1 Ermittlung der Gesamteffizienz

Eine Verringerung sowohl der zeitlichen Triggereffizienz ε_t aus Gl. (3.12) der Module des Zentralen Myon-Detektors als auch deren Hardware-Effizienz ε_h führen beide zu einer verminderten Wahrscheinlichkeit, ein physikalisch interessantes Ereignis als solches zu identifizieren. Insofern liegt es nahe, die kombinierte Gesamteffizienz ε folgendermaßen zu faktorisieren,

$$\varepsilon = \varepsilon_t \cdot \varepsilon_h \cdot \varepsilon_{ver} \quad . \tag{3.16}$$

Dabei wird allerdings nicht berücksichtigt, ob eine Korrelation von ε_t und ε_h besteht, also eine Auswirkung einer verminderten zeitlichen Effizienz auf den Wert von ε_h. Offensichtlich treten tatsächlich eine verminderte zeitliche Effizienz in Verbindung mit einer ebenfalls verringerten Hardware-Effizienz auf (Modul 36 und Modul 56). Aber auch unkorreliertes Verhalten ist zu beobachten, insbesondere das Modul 19 betreffend.

Ein bislang nicht betrachteter Effekt, welcher die Gesamteffizienz beeinträchtigt, besteht im Einfluß defekter *(„toter")* Triggerlagen. Hierfür sind die in diesem Kapitel vorgestellten Methoden der Klassenbibliothek nicht sensibel. Defekte Triggerlagen, welche überhaupt keine Treffer signalisieren, wurden in drei verschiedenen Modulen nachgewiesen (vergl. Abschn. 5.1).

3.3.2 Vergleich mit dem DST-Verfahren

Eine weitere Methode zur Ermittlung der Triggereffizienz der Module des Zentralen Myon-Detektors bildet das DST-Verfahren. Dieses bedient sich dabei nicht der Rohdatenmenge, sondern der vollständig rekonstruierten Daten der auf den DST-Bändern gespeicherten BOS-Bänke. Das DST-Verfahren wird derzeit zur Erstellung der BOS-Bank ITTE[10] verwendet, welche im Rahmen des H1-Experimentes die offiziellen Werte der Triggereffizienzen für die Korrektur von Monte-Carlo-Simulationen bereitstellt [25].

Die Grundlage des Verfahrens bilden die Variablen der BOS-Bank DMUO[11], welche u.a. auch die im Verlauf einer Spur getroffenen Lagen eines Moduls beinhaltet. Ausgehend von dieser Information kann das Vorliegen einer Lagenkoinzidenz geprüft werden, wobei die Zeitpunkte (Zeitscheiben) der Treffer in den Lagen unberücksichtigt bleiben. Lediglich Ereignisse, die genau ein detektiertes Myon enthalten, finden hierbei Beachtung. Auf diese Weise bleibt die Zuordnung einer Spur zu einem Triggerelement in jedem Falle eindeutig.

Weist ein Modul eine erfüllte Lagenkoinzidenzbedingung auf, so wird die BOS-Bank TEL1(0) auf ein Triggerelement in der Zeitscheibe 0 und dem entsprechenden Sektor hin untersucht. Darüberhinaus ist einer der 21 bereits oben beschriebenen von L4 validierten Subtrigger zu fordern, welche vom Myon-Trigger unabhängig sind. In einem solchen Fall wird das Ereignis in einem Konformitätshistogramm gezählt. Als Normierung dient die Zahl der Ereignisse mit einer Lagenkoinzidenz in dem entsprechenden Modul in Verbindung mit einem dieser von L4 validierten Subtrigger.

Prinzipiell kann die auf diesem Wege errechnete Effizienz ε_{DST} dargestellt werden als Produkt eines zeitlichen Anteils $\varepsilon_{DST,t}$ und einer Verifikationseffizienz $\varepsilon_{DST,ver}$,

$$\varepsilon_{DST} = \varepsilon_{DST,t} \cdot \varepsilon_{DST,ver} \quad . \tag{3.17}$$

Der zeitliche Anteil hierbei wird bestimmt durch das Verhältnis der Zahl $N(0)$ der Triggerelemente in der Zeitscheibe 0 zur Anzahl $N(-1,0,1,2)$ der Triggerelemente in allen vier Zeitscheiben des Readout insgesamt,

$$\varepsilon_{DST,t} = \frac{N(0)}{N(-1,0,1,2)} \quad . \tag{3.18}$$

[10] ITTE: *iron trigger timing efficiency*
[11] DMUO: *DST muon tracks*

Der Anteil der Verifikation berücksichtigt zu erwartende, aber ausbleibende Triggerelemente und auch von L4 nicht validierte Subtrigger. Er ist bestimmt durch die Anzahl der Ereignisse mit einer erfüllten Lagenkoinzidenzbedingung in Zeitscheibe 0, mindestens einem unabhängigen von L4 validierten Subtrigger sowie einem Triggerelement in Zeitscheibe 0, dividiert durch die Anzahl der Ereignisse mit einer erfüllten Lagenkoinzidenzbedingung in Zeitscheibe 0 und einem solchen L4-Subtrigger,

$$\varepsilon_{DST,ver} = \frac{N(\text{Lagenkoinzidenz \& L4-Verifikation \& Triggerelement in 0})}{N(\text{Lagenkoinzidenz \& L4-Verifikation})} \quad . \tag{3.19}$$

Insofern liegt ein Vergleich der Ergebnisse des DST-Verfahrens mit den Histogrammen der vorliegenden Analyse nahe, wie er in Abb. 3.11 dargestellt ist.

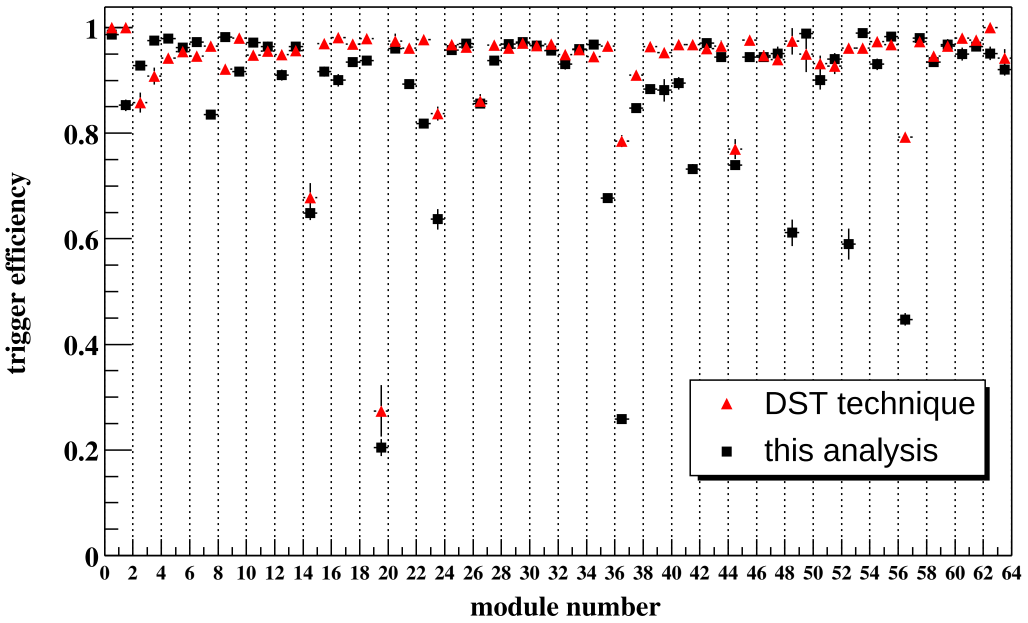

Abbildung 3.11: Vergleich der entsprechend Gl. (3.16) kombinierten zeitlichen und Hardware-Effizienzen dieser Analyse (Quadrate) mit dem Ergebnis des DST-Verfahrens (Dreiecke).

Auffällig ist, daß die mit dem DST-Verfahren gewonnenen Werte nicht in allen Fällen eine zufriedenstellende Übereinstimmung mit den Ergebnissen dieser Analyse zeigen. Insbesondere diejenigen Module, deren Gesamteffizienz offensichtlich durch eine verringerte zeitliche Effizienz beeinträchtigt sind (Module 35, 36, 41, 48, 52 und 56), werden mit dem DST-Verfahren durchweg besser beurteilt. Dies ist in gewisser Hinsicht zu erwarten, da sich das DST-Verfahren, ähnlich den Methoden der Klasse `TMuYTL4Loop`, ausschließlich der Signale des Readout bedient. Eine Berücksichtigung des Zeitverhaltens des Myon-Triggers, also des Einflusses der Nachweisverzögerung, findet nicht statt.

Deutlicher ist die Übereinstimmung, zumindest im Rahmen der jeweiligen Fehler, bei den hardwarebedingt eingeschränkt funktionstüchtigen Modulen wie etwa den Modulen

14 und 19. Sowohl der zeitliche als auch der logische Aspekt (Ausbleiben zu erwartender Triggerelemente) der Hardware-Effizienz kommen in beiden Verfahren vollständig zur Geltung.

Kapitel 4

Diagnoseverfahren

Um die Gründe für die teilweise erheblich verringerte Triggereffizienz einiger Module des Zentralen Myon-Detektors näher zu bestimmen, sind eine Reihe von weiteren Analysen notwendig. Insbesondere eine Verringerung der Hardware-Effizienz im Verlauf der Verarbeitung der Triggersignale kann auf diese Weise untersucht werden. In diesem Kapitel werden verschiedene Methoden aufgezeigt, mit deren Hilfe sich Teile der elektronischen Verarbeitungskette auf ihre Funktionstüchtigkeit hin inspizieren lassen. Parallel dazu sind die ausführenden Klassen der Klassenbibliothek beschrieben, die zur Durchführung dieser Untersuchungen Verwendung finden können.

4.1 Die Statistik getroffener Triggerlagen

Einen ersten Ansatz zur Diagnose verringerter Effizienzen stellt die Statistik getroffener Triggerlagen dar. Um einen Überblick der Aktivität dieser Lagen zu erhalten, kann zunächst die Anzahl der Treffer darin separat ermittelt werden. Soll das Ergebnis jedoch im Hinblick auf die Untersuchung der Hardware-Effizienz eines Moduls verwertbar sein, ist eine weitergehende Differenzierung nötig; es bietet sich an, hier zwischen den effizienten und den ineffizienten Ereignissen zu unterscheiden. Effizient bedeutet dabei, ein Triggerelement tatsächlich vorzufinden, sofern anhand der Signale des Readout eines hätte generiert werden müssen.

4.1.1 Die Klasse `TMuLayerLoop`

Eine solche Untersuchung kann mit Hilfe der ausführenden Klasse `TMuLayerLoop` durchgeführt werden, welche sich mit den Rohdaten beliebiger Runperioden begnügt, sofern die BOS-Bänke IRWE und TEL1 darin verfügbar sind. Die Ergebnisse speichert `TMuLayerLoop` im Verzeichnis `LayerHistograms` der Ausgabedatei ab. Es werden zwei Gruppen à vierundsechzig Histogramme erstellt, die jeweils einen Eintrag für jede der fünf Triggerlagen beinhalten.

Die erste Gruppe, welche im Unterverzeichnis `LayersOnIfNoTrigger` abgelegt wird, umfaßt die Histogramme `LayerNoTrigger_<Modulnummer>` (wobei `<Modulnummer>` von 0 bis 63 rangiert). Diese enthalten jeweils für ein Modul die Gesamtzahl der Treffer in den einzelnen Triggerlagen, sofern das Modul in diesem Ereignis ineffizient war - in keiner

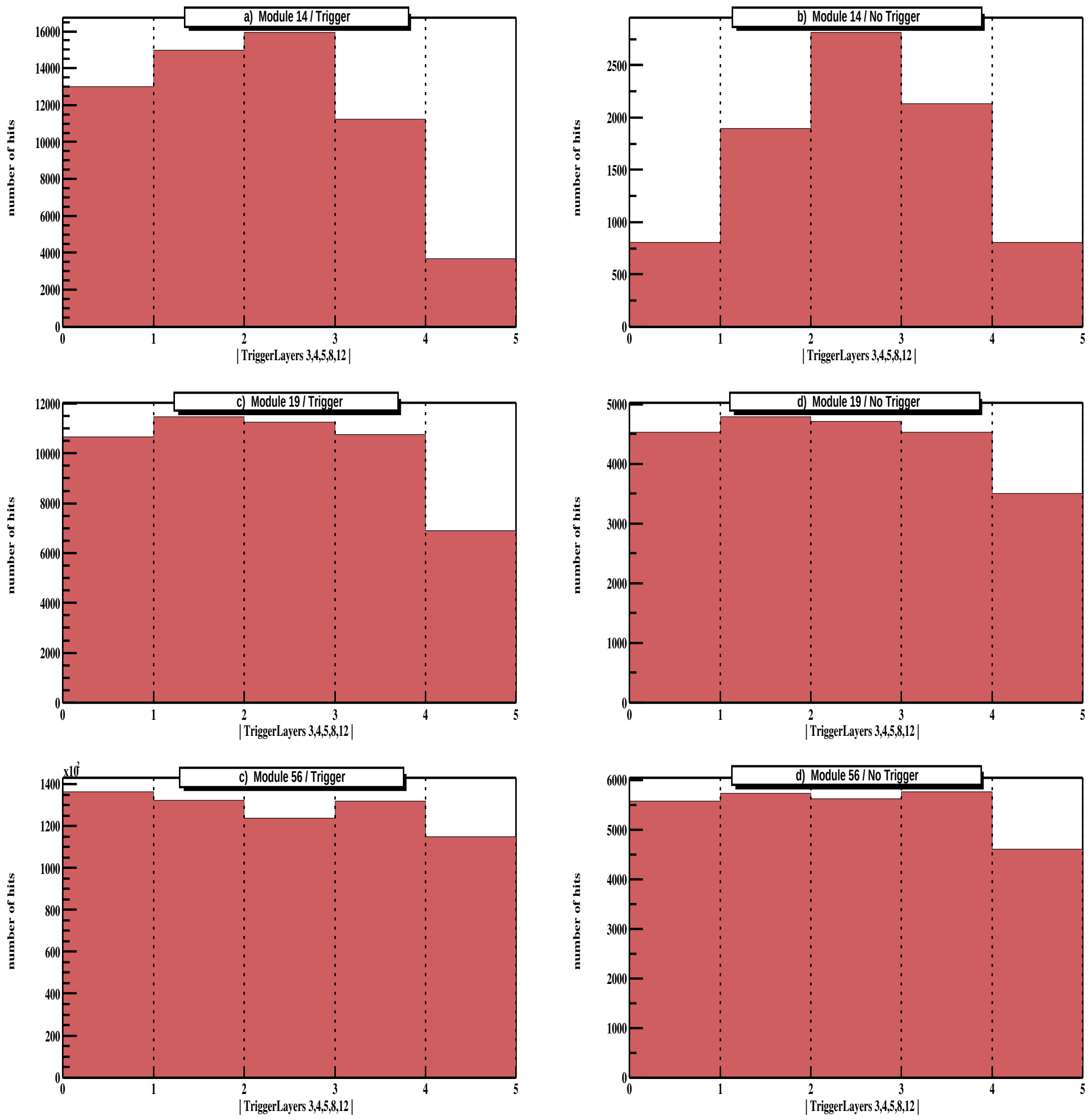

***Abbildung 4.1*:** Die Ergebnisse der ausführenden Klasse `TMuLayerLoop` für drei ausgewählte Module. Auf der linken Seite sind jeweils die Histogramme der effizienten, auf der rechten die der ineffizienten Ereignisse zu sehen.

der Zeitscheiben ist ein Triggerelement verzeichnet worden, obwohl dem Readout zufolge im Modul eine Lagenkoinzidenz zu einem Zeitpunkt vorhanden war. Um die mehrfache Ambiguität der Triggerelemente bezüglich der Module aufzulösen, können lediglich Ereignisse in der Auswertung Verwendung finden, während denen nur einzelne Module pro Sektor erfüllte Triggerbedingungen aufweisen. Es werden alle vier Zeitscheiben des Readout berücksichtigt, d.h. Treffer in allen diesen Zeitscheiben gezählt.

Die zweite Gruppe (im Unterverzeichnis `LayersOnIfTrigger`) bilden die vierundsechzig Histogramme `LayerIfTrigger_<Modulnummer>`. Diese werden analog zum vorangegangenen Fall behandelt, jedoch finden hier nur Ereignisse Eingang in die Analyse, die im obigen Sinne als effizient gelten, d.h. die im Readout detektierte Lagenkoinzidenz hat

Histogramm	Inhalt
LayerIfTrigger_0 ... LayerIfTrigger_63	Die Zahl der Treffer in den einzelnen Triggerlagen (Zeitscheiben -1 bis 2), gezählt während effizienter Ereignisse (Triggerbedingung und Triggerelement detektiert).
LayerNoTrigger_0 ... LayerNoTrigger_63	Die Zahl der Treffer in den einzelnen Triggerlagen (Zeitscheiben -1 bis 2), gezählt während ineffizienter Ereignisse (Triggerbedingung detektiert, aber kein Triggerelement).

Tabelle 4.1: Übersicht der von `TMuLayerLoop` generierten Zählhistogramme.

auch tatsächlich zu einem Triggerelement in irgendeiner der Zeitscheiben geführt. Auch hier werden in allen vier Zeitscheiben des Readout getroffene Lagen berücksichtigt.

Die Ergebnisse für die drei Module mit der niedrigsten Hardware-Effizienz sind in Abb. 4.1 dargestellt (vergl. Abb. 3.7). Auf der linken Seite sind jeweils die Histogramme plaziert, die im Falle effizienter Ereignisse gefüllt werden, auf der rechten Seite diejenigen der ineffizienten Ereignisse. Grundlage der hier vorgestellten Analyse ist das Rohdatenmaterial des Zeitraumes von April 2000 bis Ende August 2000. Die Module 19 und 56 zeigen hier nur sehr unerhebliche Unterschiede, lediglich bei Modul 14 ist zu erkennen, daß bei ausbleibendem Triggerelement die inneren Triggerlagen deutlich seltener getroffen werden als im Falle effizienter Ereignisse. Eine Begründung für dieses Verhalten kann nur schwer gegeben werden, zumal eine ganze Reihe von Faktoren wie etwa Teilchenimpuls und Spurwinkel den Nachweis der Myonen beeinflussen. Elf der Module (mit den Nummern 6, 8, 10, 11, 32, 49, 50, 53, 55, 57 und 59) wiesen überhaupt keine ineffizienten Ereignisse auf, bei sechs weiteren Modulen war deren Häufigkeit verschwindend gering.

Zusammenfassend kann aus diesen Ergebnissen noch kein aussagekräftiger Hinweis abgeleitet werden, welcher Mechanismus für die verringerten Hardware-Effizienzen verantwortlich ist, insbesondere aufgrund fehlender Gemeinsamkeiten im Verhalten der betroffenen Module.

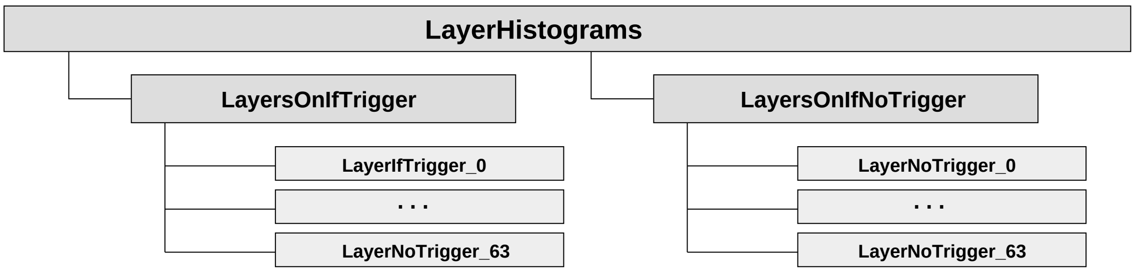

Abbildung 4.2: Übersicht des Verzeichnisses `LayerHistograms`, wie es von der Klasse `TMuLayerLoop` in der Ausgabedatei erzeugt wird.

4.2 Die Effizienz der Signalübermittlung an L2

Nach der Resynchronisation der Signale in den H1 multi purpose signal delay cards (s. Abschn. 1.3.3) zweigen die Koinzidenzkanäle zu den Speicherkarten des PQZP-Systems ab, bevor sie die sector boards erreichen. Das PQZP-System handhabt die Aquisition der für die Entscheidung der Triggerstufe L2 benötigten Daten. Im Falle einer positiven Entscheidung von L1 werden die pipelines gestoppt und die zur Zeitscheibe 0 gehörenden Signale in den Speicherkarten an die L2L3-Karten übermittelt. Die Inhalte sowohl der PQZP- als auch der L2L3-Karten stehen im Rahmen des Rohdatenmaterials für jedes Ereignis zur Verfügung, und zwar in den BOS-Bänken TSCD(8) bzw. TL23.

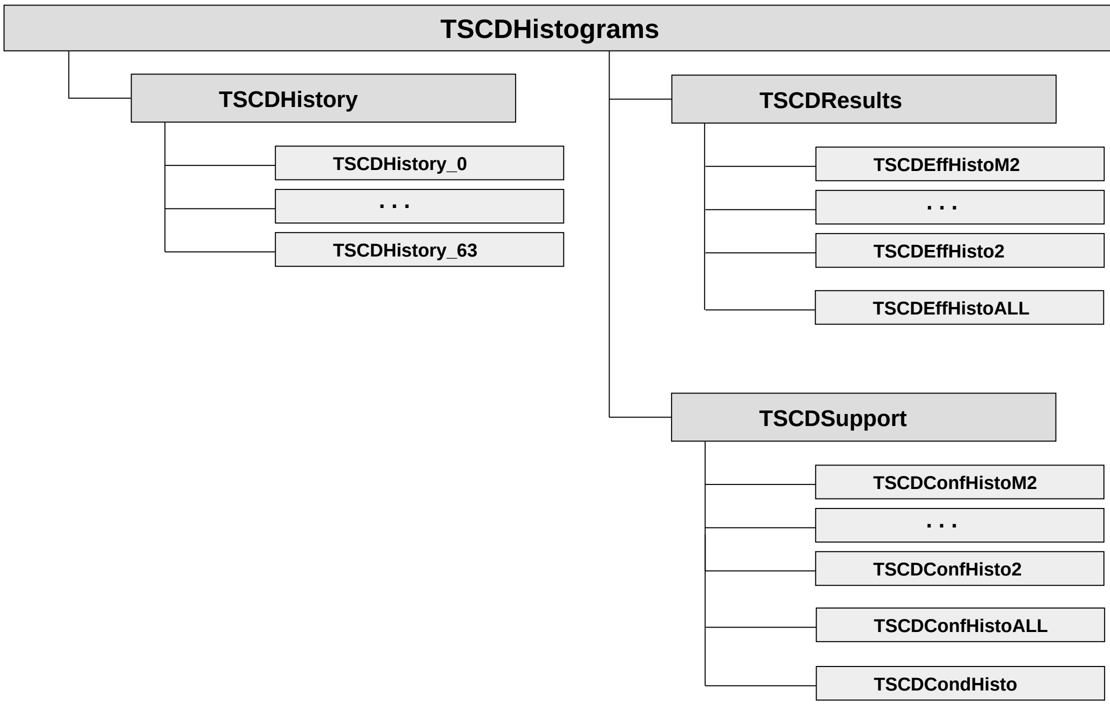

Abbildung 4.3: Übersicht der Struktur des von der Klasse `TMuTSCDLoop` in der Ausgabedatei erzeugten Unterverzeichnisses.

4.2.1 Bestimmung anhand der Koinzidenzsignale in den PQZP-Karten

Die das PQZP-System erreichenden Koinzidenzsignale können nun mit denjenigen des Readout verglichen werden. Weist der Readout für eine Zeitscheibe n eine erfüllte Lagenkoinzidenz auf, so sollte diese Information auch in den Speicherkarten an der entsprechenden Stelle der pipeline zu finden sein. Ist dies der Fall, wird das Ereignis im Konformitätshistogramm gezählt. Die Effizienz der Signalübermittlung an L2 errechnet sich dann aus der Häufigkeit solcher Konformitäten dividiert durch die Zahl der insgesamt in Zeitscheibe n detektierten Lagenkoinzidenzen.

Eine Verringerung der Effizienz der Signalübermittlung an L2 kann durch zweierlei Kategorien von Fehlfunktionen verursacht werden. Einerseits kann das zu erwartende Koinzidenzsignal in einer anderen als der vorgesehenen Zeitscheibe in den PQZP-Karten eintreffen, andererseits aber auch ganz ausbleiben. In einem derartigen Fall kommen mehrere Teilsysteme der elektronischen Verarbeitungskette für einen Defekt oder eine unkorrekte Einstellung in Frage; es sind dies die unmittelbare Ausleseelektronik, die layer boards sowie die multi purpose signal delay cards.

4.2.2 Die Klasse `TMuTSCDLoop`

Histogramm	Inhalt
`TSCDEffHistoM2`	Die Wahrscheinlichkeit eines PQZP-Koinzidenzsignals in Zeitscheibe -2, ausgelöst durch eine Lagenkoinzidenz[1] in Zeitscheibe 0
`TSCDEffHistoM1`	Gleiches für ein PQZP-Koinzidenzsignal in Zeitscheibe -1
`TSCDEffHisto0`	Gleiches für ein PQZP-Koinzidenzsignal in Zeitscheibe 0
`TSCDEffHisto1`	Gleiches für ein PQZP-Koinzidenzsignal in Zeitscheibe 1
`TSCDEffHisto2`	Gleiches für ein PQZP-Koinzidenzsignal in Zeitscheibe 2
`TSCDEffHistoALL`	Die Wahrscheinlichkeit eines PQZP-Koinzidenzsignals in irgendeiner Zeitscheibe, ausgelöst durch eine Lagenkoinzidenz in Zeitscheibe 0

Tabelle 4.2: Übersicht der von `TMuTSCDLoop` erzeugten Effizienzhistogramme. Alle Histogramme umfassen vierundsechzig unabhängige Einträge für alle Module. Konformitäts- und Bedingungshistogramme sind nicht aufgeführt.

Die oben beschriebene Vorgehensweise zur Ermittlung der Signalübermittlungseffizienz an L2 ist in der Klasse `TMuTSCDLoop` implementiert. Auch diese Klasse verarbeitet Rohdaten normaler Runperioden. Grundlage der hier durchgeführten Untersuchung ist ebenfalls das Datenmaterial des Zeitraumes von April 2000 bis Ende August 2000.

Zunächst untersucht `TMuTSCDLoop` den Inhalt der BOS-Bank IRWE und bedient sich dazu der Methoden der Klasse `TMuIRWE`, um erfüllte Lagenkoinzidenzbedingungen in den Modulen zu erkennen. Ist eine solche Bedingung erfüllt, so wird der Inhalt der pipeline der PQZP-Speicherkarte 1 auf das Vorhandensein der entsprechenden Koinzidenzsignale hin überprüft. Da die Koinzidenzsignale auch mit zeitlicher Verschiebung in den Speicherkarten zu finden sein können, werden mehrere Einträge der pipelines für die Zeitscheiben -2 bis 2 untersucht. Treten die Signale in beiden Bänken auf, so wird das Konformitätshistogramm für die entsprechende Zeitscheibe gefüllt; abschließend findet eine Normierung aller Werte auf die Zahl der insgesamt anhand des Readout in der entsprechenden Zeitscheibe ermittelten Lagenkoinzidenzen statt. Für die Untersuchung des Inhalts der BOS-Bank TSCD(8) enthält die Basisklasse `TMuTSCD` alle erforderlichen Methoden. Eine Übersicht der erzeugten Effizienzhistogramme gibt die Tabelle 4.2.

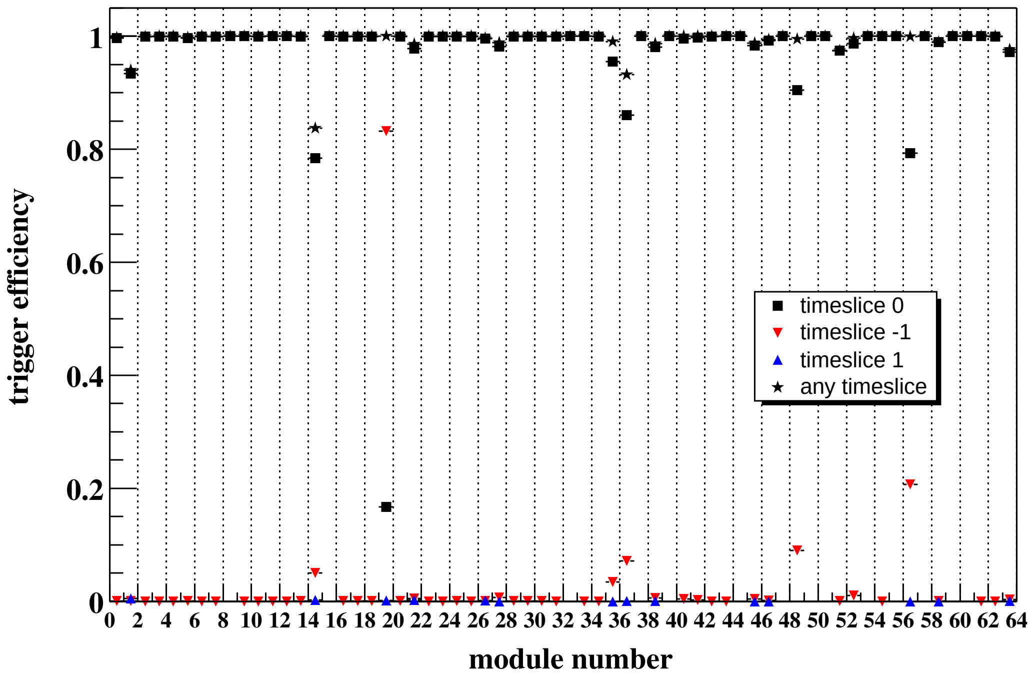

Abbildung 4.4: Die Wahrscheinlichkeit eines PQZP-Lagenkoinzidenzsignals in der Zeitscheibe 0 (Quadrate), den Zeitscheiben -1 oder 1 (jwls. Dreiecke) sowie in irgendeiner der Zeitscheiben (Sterne), falls in der Zeitscheibe 0 eine Lagenkoinzidenzbedingung erfüllt war. Nur von null verschiedene Werte sind eingezeichnet.

Der Eintrag der pipeline der PQZP-Karte 1, welcher als Zeitscheibe 0 ausgelesen wird, ist abhängig von der Konfiguration der Karte und somit langfristig variabel. Seine Position kann allerdings aus dem Inhalt der BOS-Bank TSCD(8) errechnet werden, wofür die Klasse `TMuTSCD` ebenfalls entsprechende Methoden umfaßt. Diese werden von `TMuTSCDLoop` selbständig aufgerufen, und das Ergebnis erscheint zur Kontrolle auf dem Bildschirm. Es sollte in jedem Falle auf einen vernünftigen Wert hin überprüft werden (vergl. Abschn. 1.3.3).

Das Verzeichnis innerhalb der Ausgabedatei, welches `TMuTSCDLoop` erzeugt, trägt den Namen `TSCDHistograms`. Es gliedert sich in drei Unterverzeichnisse, einerseits `TSCDResults`, welches die Effizienzhistogramme beinhaltet, und `TSCDSupport` zur Speicherung der Konformitäts- und Bedingungshistogramme.

`TMuTSCDLoop` erfaßt darüberhinaus alle Koinzidenzsignale eines jeden Moduls, welche die PQZP-Speicherkarten erreichen, und zählt diese, aufgeschlüsselt nach Zeitscheiben (bzw. den Einträgen der pipeline), in vierundsechzig Histogrammen namens TSCDHistory_<Modulnummer>. Diese werden im Unterverzeichnis TSCDHistory abgelegt. Die Struktur des Verzeichnisses TSCDHistograms mit den darin enthaltenen Histogrammen veranschaulicht Abb. 4.3.

Die Ergebnisse der Analyse sind in der Abb. 4.4 dargestellt. Die Signalübermittlungseffizienz der verschiedenen Module für ein Lagenkoinzidenzsignal in der Zeitscheibe 0 (Quadrate) weist ein ähnliches Muster auf wie die allgemeine Hardware-Effizienz (vergl. Abb. 3.7). Unterdurchschnittliche Werte zeigen wiederum die Module 14, 19, 36, 48 sowie 56. Offensichtlich ist die Ursache für deren verminderte Effizienzwerte in den ersten Gliedern der elektronischen Verarbeitungskette zu suchen, da die Lagenkoinzidenzsignale bereits am Ausgang der multi purpose signal delay cards nach L2 abzweigen.

In auffälliger Weise zeigt sich in diesen Fällen eine Häufung von Lagenkoinzidenzsignalen in der Zeitscheibe -1 (Dreiecke), und zwar desto deutlicher, je geringer der Wert der Effizienz für Zeitscheibe 0 ausfällt. Dies läßt den Schluß zu, daß es sich bei dem Grund für die verringerten Effizienzwerte um ein zeitliches Problem handelt; ein gewisser Teil der Lagenkoinzidenzen trifft verfrüht ein. Dies wird insbesondere deutlich, wenn zusätzlich die Wahrscheinlichkeit betrachtet wird, ein Lagenkoinzidenzsignal überhaupt in einer der Zeitscheiben zu beobachten. Für die Module 19, 48 und 56 ist diese nahezu 1. Es werden also alle Signale erzeugt, nur erreichen sie die PQZP-Karten in vielen Fällen zu einem vorgezogenen Zeitpunkt.

4.2.3 Bestimmung anhand der Koinzidenzsignale in den L2L3-Karten

Histogramm	Inhalt
TL23EffHisto	Die Wahrscheinlichkeit eines L2L3-Koinzidenzsignals in Zeitscheibe 0, ausgelöst durch eine Lagenkoinzidenz[1] in Zeitscheibe 0
TL23TrigHisto	Gesamtzahl der an die L2L3-Karte übermittelten Triggerelemente für jedes der 64 Module.

Tabelle 4.3: Die von TMuTL23Loop erzeugten Ergebnishistogramme. Beide Histogramme umfassen vierundsechzig unabhängige Einträge für alle Module. Konformitäts- und Bedingungshistogramm sind nicht aufgeführt.

Auch der Inhalt der entsprechenden L2L3-Karte des PQZP-Systems kann mit den Signalen des Readout verglichen werden. Diese Information ist in der BOS-Bank TL23 verfügbar. Die L2L3-Karte der Triggerstufe L2 nimmt im Falle einer positiven Entscheidung der Triggerstufe L1 das Lagenkoinzidenzsignal der Zeitscheibe 0 der PQZP-Speicherkarten[2] auf (vergl. Abschn. 1.3.3). Aus diesem Grunde sollten die hier ermittel-

[1] Alle Lagenkoinzidenzen werden detektiert anhand der Signale des Readout.

[2] Dieses Signal wird der PQZP-Speicherkarte 0 entnommen.

ten Effizienzwerte mit denjenigen für die Zeitscheibe 0 des vorangegangenen Abschnitts übereinstimmen.

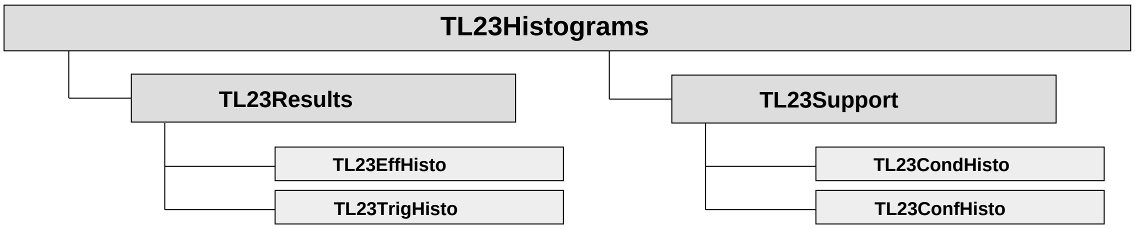

Abbildung 4.5: Übersicht des von `TMuTL23Loop` in der Ausgabedatei angelegten Verzeichnisses.

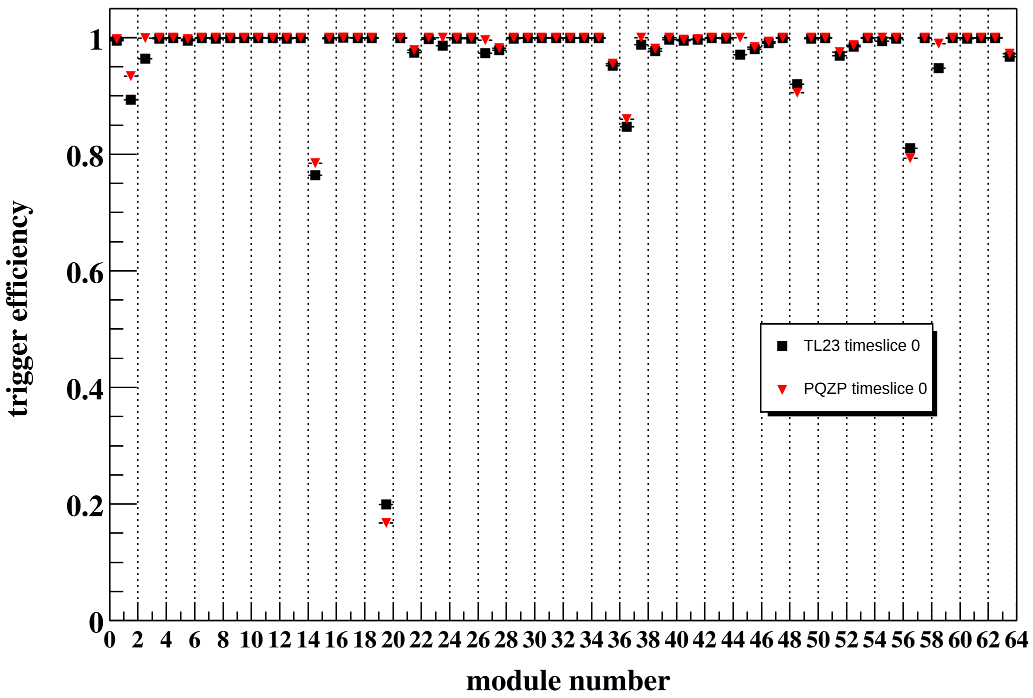

Abbildung 4.6: Die mit Hilfe der Signale der L2L3-Karten errechneten Signalübermittlungseffizienzen (Quadrate) im Vergleich zu denjenigen aus den Daten der PQZP-Karten ermittelten (Dreiecke).

4.2.4 Die Klasse `TMuTL23Loop`

Die Ermittlung der Signalübermittlungseffizienz an L2 mit Hilfe des Inhaltes der L2L3-Karten verläuft analog zu derjenigen mit Hilfe der PQZP-Speicherkarten. Auch stützt sich die Analyse auf dasselbe Datenmaterial. Allerdings verfügen die L2L3-Karten nicht über pipelines, so daß jeweils nur das der Zeitscheibe 0 zugehörige Signal ausgelesen und verglichen werden kann. Somit entfällt auch die Lokalisierung des entsprechenden

Eintrags. Die Methoden zur Auslese der BOS-Bank TL23 entstammen der Basisklasse `TMuTL23`.

`TMuTL23Loop` erzeugt in der Ausgabedatei das Verzeichnis `TL23Histograms`, welches wiederum in die Unterverzeichnisse `TL23Results` (Ergebnishistogramme) und `TL23Support` (Konformitäts- und Bedingungshistogramm) gegliedert ist. Zusätzlich zum Effizienzhistogramm zählt `TMuTL23Loop` für jedes Modul die Gesamtzahl der die L2L3-Karten erreichenden Koinzidenzsignale und legt diese Werte in einem separaten Histogramm `TL23TrigHisto` ab. Eine Übersicht des Verzeichnisses gibt Abb. 4.5, die erzeugten Histogramme sind nochmals in Tab. 4.3 aufgeführt.

Wie die Abb. 4.6 zeigt, stimmen die mit Hilfe der Klasse `TMuTL23Loop` bestimmten Effizienzen (Quadrate) nicht exakt mit denjenigen des vorangegangenen Abschnitts für die Zeitscheibe 0 überein (Dreiecke). Die Abweichungen betragen in einigen Fällen bis zu 4%. Der Grund hierfür ist nicht bekannt; beide Gruppen von Effizienzwerten basieren auf derselben Rohdatenmenge.

4.2.5 Vergleich der Signale in den PQZP-Karten mit den Eingangssignalen der GPTP-Elektronik

Da die in den layer boards erzeugten Lagenkoinzidenzsignale wie in Kapitel 1 beschrieben sowohl zu den PQZP-Speicherkarten als auch zu den GPTP-Karten hin abzweigen, liegt ein Vergleich der jeweils dort eintreffenden Signale nahe. Auf diesem Wege können Aussagen getroffen werden hinsichtlich der Funktion eines Teils der elektronischen Verarbeitungskette. Es sind dies die sector boards, die delay card zur Verzögerung der t_0-Signale sowie die GPTP-Karte selbst. Ein Vergleich mit den Signalen des Readout findet somit nicht statt.

Die Eingangssignale in die pipeline der GPTP-Karte sind in der BOS-Bank TGPP(2048) verfügbar, und zwar jeweils für die sechzehn Triggersektoren in einem Zeitfenster von 32 Zeitscheiben. Dabei werden sowohl die t_0-Signale als auch die Koinzidenzsignale der jeweiligen Zeitscheiben in der Bank abgelegt. Wie bei der PQZP-Speicherkarte 1 ist die Lage des als Zeitscheibe 0 ausgelesenen pipeline-Eintrages variabel mit der Konfiguration der elektronischen Bauteile.

Ist eine solche Koinzidenz in der pipeline der GPTP-Karte verzeichnet, so ist zu überprüfen, ob mindestens eines der zum entsprechenden Triggersektor zugehörigen Module ein Signal ebenfalls in der PQZP-Speicherkarte verursacht hat (die Information bezüglich der Lagenkoinzidenz liegt in der BOS-Bank TSCD(8) modulweise vor). Obwohl prinzipiell möglich, ist es nicht notwendig, diesen Vergleich in mehreren Zeitscheiben parallel durchzuführen; eine Beschränkung auf die Zeitscheibe 0 ist durchaus sinnvoll.

4.2.6 Die Klasse `TMuL1L2Loop`

Der oben beschriebene Vergleich der Signale in den PQZP-Karten mit den Eingangssignalen der GPTP-Elektronik kann mit Hilfe der ausführenden Klasse `TMuL1L2Loop` durchgeführt werden. `TMuL1L2Loop` erzeugt in der Ausgabedatei das Verzeichnis `L1L2Histograms`, welches wiederum die beiden Unterverzeichnisse `L1L2Results` (Ergebnishistogramme) sowie `L1L2Support` (Konformitäts- und Bedingungshistogramme) umfaßt (s. Abb. 4.7).

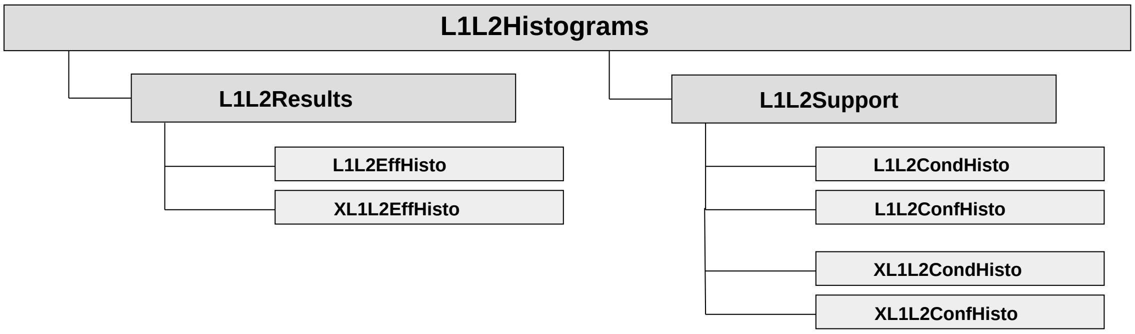

Abbildung 4.7: Die Struktur des Verzeichnisses, in welchem `TMuL1L2Loop` seine Histogramme ablegt.

Histogramm	Inhalt
`L1L2EffHisto`	Die Wahrscheinlichkeit, ein GPTP-Koinzidenzsignal in der Zeitscheibe 0 anhand der Koinzidenzsignale der PQZP-Speicherkarten (Zeitscheibe 0) verifizieren zu können.
`XL1L2EffHisto`	Die Wahrscheinlichkeit, ein GPTP-Koinzidenzsignal in einer der Zeitscheiben -1, 0, 1 anhand der Koinzidenzsignale der PQZP-Speicherkarten (Zeitscheibe 0) verifizieren zu können.

Tabelle 4.4: Die von `TMuL1L2Loop` erzeugten Ergebnishistogramme. Beide Histogramme umfassen sechzehn unabhängige Einträge für die Triggersektoren. Konformitäts- und Bedingungshistogramme sind nicht aufgeführt.

Das Ergebnis des Vergleichs wird auch hier in Form einer Effizienz dargestellt. Existiert ein Lagenkoinzidenzsignal in der pipeline der GPTP-Karte (Zeitscheibe 0) und hat mindestens eines der Module des entsprechenden Triggersektors ein solches Signal auch in der Zeitscheibe 0 der PQZP-Speicherkarte verursacht, so wird dies als Konformität gezählt. Acht der Triggersektoren erfordern zwei oder mehr getroffene Module. Die abschließende Normierung besteht in der sektorweisen Division durch die Anzahl aller im Triggersektor registrierten Lagenkoinzidenzsignale in der Zeitscheibe 0.

Ein weiteres Histogramm dient der Erfassung potentieller zeitlicher Verschiebungen der beiden Signalgruppen gegeneinander. Zu diesem Zweck werden die drei Zeitscheiben -1, 0, 1 der GPTP-pipeline betrachtet. Findet sich in einer dieser Einträge ein Lagenkoinzidenzsignal, so wird wiederum in den Daten der PQZP-Speicherkarten das entsprechende Signal in der Zeitscheibe 0 überprüft. Ist es dort vorhanden, so besteht Konformität. Die

Normierung ist die gleiche wie im vorangegangenen Fall. Beide Ergebnishistogramme sind in Tab. 4.4 nochmals aufgeführt.

Im Gegensatz zu den Inhalten der PQZP-Speicherkarten kann der als Zeitscheibe 0 interpretierte Eintrag der GPTP-pipeline nicht anhand der Rohdaten ermittelt werden. Er ist somit im Falle einer Änderung manuell anzupassen[3].

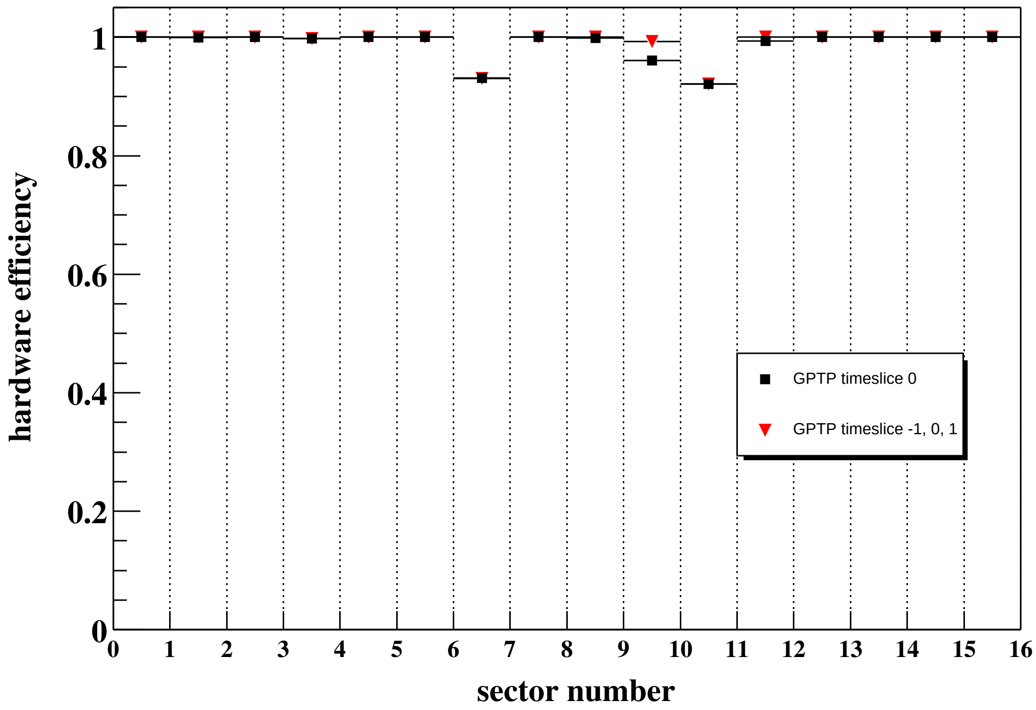

Abbildung 4.8: Die von der ausführenden Klasse `TMuL1L2Loop` ermittelten Wahrscheinlichkeiten, ein GPTP-Lagenkoinzidenzsignal (Quadrate: Zeitscheibe 0, Dreiecke: Zeitscheiben -1, 0 oder 1) anhand der Daten der PQZP-Speicherkarte 1 (nur Zeitscheibe 0) zu verifizieren. Auf der Abszisse sind die sechzehn Triggersektoren aufgetragen (vergl. hierzu Tab. 1.4).

Wie die Abb. 4.8 verdeutlicht, sind die Ergebnisse des Vergleichs der Signale in den PQZP-Karten mit den Eingangssignalen der GPTP-Elektronik (Quadrate) durchaus zufriedenstellend. Bis auf die einfachen Triggersektoren des unteren Barrelbereiches ($n_{BLBA} > 0$ und $n_{FLBA} > 0$) sowie den Triggersektor $n_{FUBA} > 1$ betragen alle Effizienzen nahezu 100%. Lediglich im letzteren Falle ist die verringerte Effizienz dabei auf ein zeitliches Problem zurückzuführen; die Effizienzwerte für ein erweitertes Zeitfenster (Dreiecke) beweisen, daß dort sehr wohl in jedem Falle Lagenkoinzidenzsignale in beiden Karten generiert werden, jedoch nicht immer in der korrekten Zeitscheibe.

[3]Während des Jahres 2000 befand sich dieser Eintrag an Position 19 der pipeline.

4.3 Die Analyse der die GPTP-Elektronik erreichenden Signale

Die im Vorfeld beschriebenen Untersuchungen decken bereits einen Großteil der Hardware des Myon-Triggers bis hin zu den Eingangssignalen der GPTP-Karte ab. Deren Funktion selbst, d.h. die Generierung der an die Triggerstufe L1 zu übermittelnden Triggerelemente, kann damit jedoch nicht einer Prüfung unterzogen werden.

4.3.1 Die Klasse `TMuTGPPLoop`

Die Klasse `TMuTGPPLoop` simuliert die Verarbeitung der t_0- und Lagenkoinzidenzsignale zu den acht Triggerelementen des Zentralen Myon-Detektors, wie sie innerhalb der GPTP-Karte vorgenommen wird. Im Zuge dieser Operation entsteht zunächst das logische UND der beiden Eingangssignale für jeden Triggersektor in der Zeitscheibe 0. Anschließend werden die so erhaltenen sechzehn Bits gemäß den Angaben in der Tab. 1.5 zu den Triggerelementen verknüpft.

Das Ergebnis dieser Simulation und der Vergleich mit den tatsächlich an L1 übermittelten Triggerelementen kann wieder in Form einer Effizienz angegeben werden. Diese stellt die Wahrscheinlichkeit dar, ein bestimmtes Triggerelement tatsächlich vorzufinden, sofern es aufgrund der Eingangssignale der GPTP-Karte zu erwarten war. Auch in diesem Falle

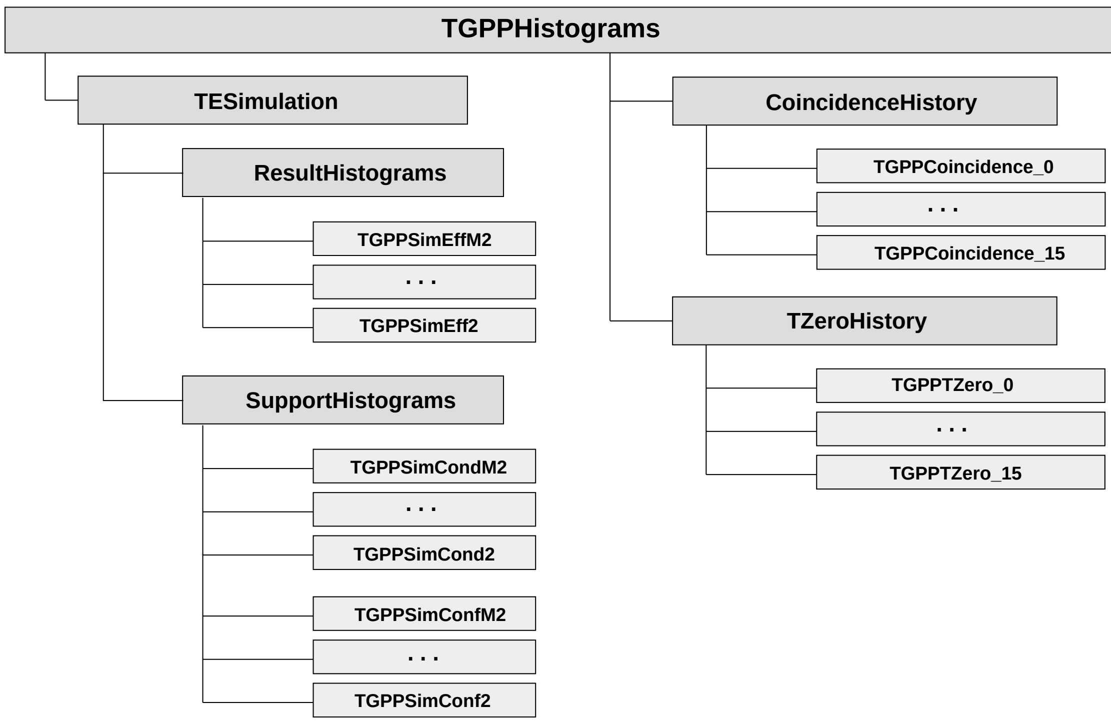

Abbildung 4.9: `TMuTGPPLoop` legt seine Histogramme in das Verzeichnis `TGPPHistograms` der Ausgabedatei ab.

besteht die Möglichkeit, ein Triggerelement in verschiedenen Zeitscheiben aufzusuchen, um etwaige zeitliche Verschiebungen zu erfassen.

Alle Histogramme, die `TMuTGPPLoop` generiert, finden sich nach Abschluß der Bearbeitung im Verzeichnis `TGPPHistograms` der Ausgabedatei. Das Unterverzeichnis `TESimulation` beherbergt dabei die Ergebnishistogramme (Unterverzeichnis `ResultHistograms`) sowie die Konformitäts- und Bedingungshistogramme (Unterverzeichnis `SupportHistograms`).

`TMuTGPPLoop` zählt weiterhin die für die einzelnen Triggersektoren eintreffenden t_0- und Lagenkoinzidenzsignale, separat und nach Zeitscheiben aufgeschlüsselt, über ein Zeitfenster von 32 Zeitscheiben. Die so erhaltenen beiden Gruppen zu je sechzehn Histogrammen werden in den Unterverzeichnissen `CoincidenceHistory` und `TZeroHistory` abgelegt.

Histogramm	Inhalt
`TGPPSimEffM2`	Die Wahrscheinlichkeit, ein Triggerelement in Zeitscheibe -2 vorzufinden, falls aufgrund der GPTP-Eingangssignale eines in Zeitscheibe 0 zu erwarten ist.
`TGPPSimEffM1`	Die entsprechende Wahrscheinlichkeit für ein Triggerelement in Zeitscheibe -1.
`TGPPSimEff0`	Die entsprechende Wahrscheinlichkeit für ein Triggerelement in Zeitscheibe 0.
`TGPPSimEff1`	Die entsprechende Wahrscheinlichkeit für ein Triggerelement in Zeitscheibe 1.
`TGPPSimEff2`	Die entsprechende Wahrscheinlichkeit für ein Triggerelement in Zeitscheibe 2.

Tabelle 4.5: Die von `TMuTGPPLoop` erzeugten Effizienzhistogramme zur Erzeugung der Triggerelemente und ihr Inhalt.

Die Funktion der GPTP-Karte die Generierung der Triggerelemente betreffend kann nur als einwandfrei bezeichnet werden (s. Abb. 4.10, Quadrate). Bis auf das Triggerelement mit der Nummer 60 (dies entspricht mehr als 2 getroffenen Modulen im Barrelbereich) liegt die Effizienz bei 100%. Auch sind die Wahrscheinlichkeiten eingetragen, das entsprechende Triggerelement in einer der beiden benachbarten Zeitscheiben -1 oder 1 aufzufinden (jeweils Dreiecke). Auffällig ist hier, daß das Triggerelement Nummer 60 sehr häufig auch in der Zeitscheibe -1 erscheint.

4.4 Schlußfolgerung

Auf der Grundlage der in den vorangegangenen Abschnitten vorgestellten Untersuchungsergebnisse können nun die Ursachen für die verringerten Triggereffizienzen einiger Module weitgehend eingegrenzt werden.

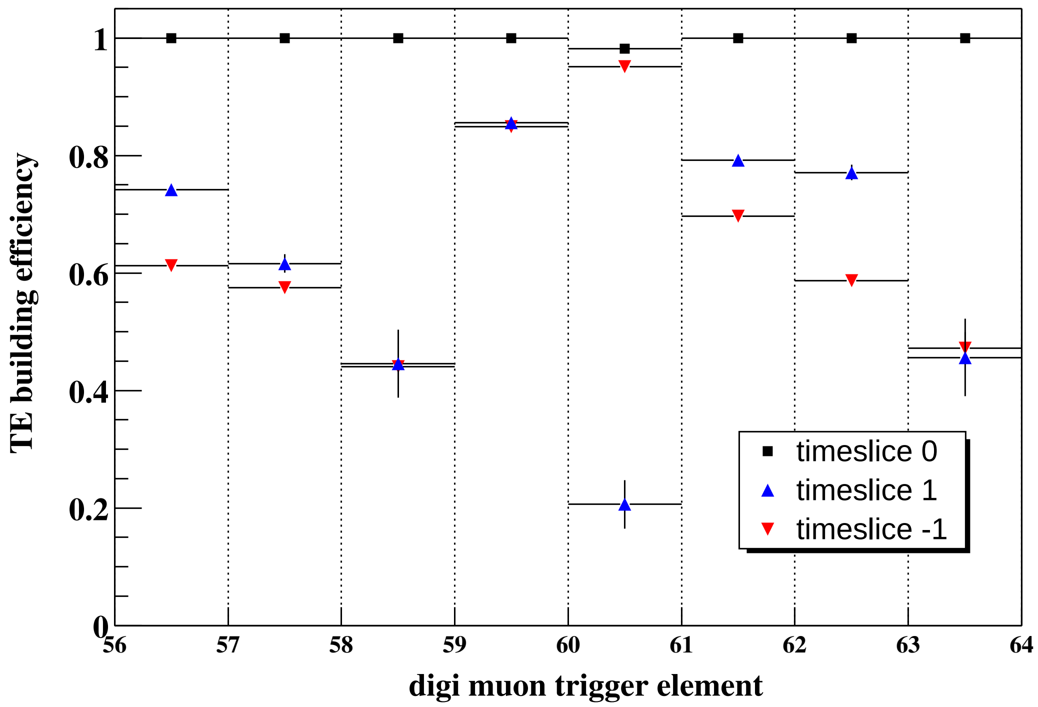

Abbildung 4.10: Die Effizienz GPTP-Karte im Hinblick auf die Generierung der Triggerelemente. Deren Nummern sind in Tab. 1.5 aufgeschlüsselt. Quadrate: Triggerelement in Zeitscheibe 0, jeweils Dreiecke: Triggerelement in Zeitscheibe -1 bzw. 1.

Bereits bei der Betrachtung der Signalübertragungseffizienzen an L2 der betroffenen Module wurde deutlich, daß es offenbar nur in einer geringen Anzahl von Fällen tatsächlich zu einem Verlust erzeugter Lagenkoinzidenzsignale kommt. Vielmehr treffen diese häufig mit einer zeitlichen Verschiebung in L2 und in der GPTP-Karte ein. Es gilt nun zu ermitteln, inwiefern diese Verschiebung die Generierung eines Triggerelementes in der GPTP-Karte unterbinden kann.

Die Abb. 4.11 zeigt den Inhalt der GPTP-Karte zweier ausgewählter Ereignisse in jeweils einem Diagramm. Je eine Spalte steht hierin für einen Triggersektor, die 32 Einträge der GPTP-pipeline sind als Zeilen wiedergegeben[4]. Beide Ereignisse sind dahingehend ausgewählt, daß nur in Modul 19, welches dem Triggersektor BUBA (Spalte Nr. 4) zugeordnet ist, eine erfüllte Triggerbedingung vorlag. Zusätzlich wurden Treffer in jedem der anderen Module des Triggersektors 4 ausgeschlossen. Auf diese Weise ist sichergestellt, daß die in den Diagrammen verzeichneten t_0-Signale (`T`) und Lagenkoinzidenzsignale (`C`) dort von Modul 19 verursacht wurden.

Der als Zeitscheibe 0 ausgelesene Eintrag der GPTP-pipeline ist derjenige an Position (Zeile im Diagramm) 19. Bei der Betrachtung des Diagramms `a)` der Abb. 4.11 wird nun der Grund für das häufige Ausbleiben der von Modul 19 zu erwartenden Triggerelemente

[4]Solche Diagramme können mit der Methode `Print()` der Basisklasse `TMuTGPP` auf dem Bildschirm ausgegeben werden.

Triggersektor

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
1	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
2	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
3	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
4	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
5	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
6	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
7	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
8	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
9	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
10	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
11	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
12	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
13	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
14	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
15	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
16	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
17	--	--	--	--	C-	--	--	--	--	--	--	--	--	--	--	--
18	--	--	--	--	C-	--	--	--	--	--	--	--	--	--	--	--
19	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
20	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
21	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
22	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
23	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
24	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
25	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
26	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
27	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
28	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
29	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
30	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
31	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

a) TGPP input (run 278823, event 100675)

Triggersektor

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	--	--	--	--	-T	--	--	--	--	--	--	--	-T	--	--	--
1	--	--	--	--	-T	--	--	--	--	--	--	--	-T	--	--	--
2	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
3	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
4	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
5	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
6	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
7	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
8	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
9	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
10	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
11	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
12	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
13	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
14	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
15	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
16	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
17	--	--	--	--	CT	--	--	--	--	--	--	--	--	--	--	--
18	--	--	--	--	CT	--	--	--	--	--	--	--	--	--	--	--
19	--	--	--	--	-T	-T	--	--	--	--	--	--	--	--	--	--
20	--	--	--	--	-T	-T	--	--	--	--	--	--	--	--	--	--
21	--	--	--	--	-T	--	-T	--	--	--	--	--	--	--	--	--
22	--	--	--	--	-T	--	-T	--	--	--	--	--	--	--	--	--
23	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
24	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
25	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
26	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
27	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
28	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
29	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
30	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--
31	--	--	--	--	-T	--	--	--	--	--	--	--	--	--	--	--

b) TGPP input (run 278802, event 87009)

Abbildung 4.11: Diagramme der Eingangssignale der GPTP-Karte 1 zweier Ereignisse, wie sie die Routine `Print()` der Klasse `TMuTGPP` ausgibt. Die Spalten bezeichnen die 16 Triggersektoren (vergl. Tab. 1.4), die Zeilen die 32 Einträge der pipeline. Es bedeuten T: t_0-Signal, C: Koinzidenzsignal. Der als Zeitscheibe 0 ausgelesene Eintrag hat die Position (Zeile) 19, spätere Zeitscheiben liegen unterhalb davon.

deutlich[5]. Ein verfrühtes Eintreffen der Koinzidenzsignale führt zu dem Umstand, daß t_0- und Lagenkoinzidenzsignale nicht mehr in der Zeitscheibe 0 zusammentreffen - aber nur in einem solchen Falle würde das Triggerelement durch Anwendung der logischen UND-Operation gebildet. Das Diagramm 4.11 a) steht diesbezüglich beispielhaft für eine Vielzahl anderer Ereignisse, die an dieser Stelle nicht wiedergegeben werden können.

Für die Module 48 und 56 sowie eingeschränkt auch für das Modul 35 trifft die gleiche Argumentation zu. In diesen Fällen läßt sich ebenfalls eine deutliche Korrelation zwischen verfrüht eintreffenden Lagenkoinzidenzsignalen und ausbleibenden Triggerelementen feststellen.

Weiterhin stellt sich die Frage, auf welche Weise diese zeitliche Verschiebung zustande kommt und wie sie zu beheben ist. Prinzipiell erscheinen drei Ursachen naheliegend.

- *Eine unkorrekte Konfiguration der Elektronik.* Unter den vielfältigen Möglichkeiten, den Signalfluß innerhalb der Verarbeitungskette des Myon-Triggers zeitlich zu beeinflussen, kommen nur jene in Frage, die eine Verzögerung von t_0- und Lagenkoinzidenzsignalen unabhängig voneinander erlauben. Am Ausgang der layer boards besteht die Möglichkeit einer Verzögerung in Schritten von 24 ns [10], so daß auf die Verteilung der Signale in die Zeitscheiben im Zuge der folgenden Resynchronisation eingewirkt werden kann. Dies deckt sich ebenfalls mit den Ergebnissen der

[5] Die Übereinstimmung von Positionsnummer und Modulnummer ist rein zufällig und nicht von Bedeutung.

Untersuchungen dieses Kapitels, wonach die Ursache der Ineffizienz im ersten Teil der Verarbeitungskette zu suchen ist.

- *Ein Defekt der Elektronik.* Dieser wäre ebenfalls in der unmittelbar auslesenden Elektronik, den layer boards oder der multi purpose signal delay card zu suchen. Eine kontinuierlich auftretende zeitliche Verschiebung kann auch in diesem Fall durch die Anpassung der Signalverzögerung behoben werden.

- *Ein Fehler des Readout.* In den vorangegangenen Abschnitten wurde stillschweigend eine korrekte Verarbeitung der Drahtkammersignale im Rahmen des Readout vorausgesetzt. Da die Hardware-Effizienz aber auf einem Vergleich mit diesen beruht, liegt auch hier eine theoretisch mögliche Fehlerquelle vor. Hinweise darauf sind allerdings nicht festzustellen.

Anders gelagert ist der Sachverhalt in Bezug auf die Module 14 und 36. Hier gehen tatsächlich in einem Teil der Ereignisse Lagenkoinzidenzsignale verloren, die anhand des Readout eindeutig zu rekonstruieren sind. Da diese in solchen Fällen auch nicht in den PQZP-Speicherkarten verzeichnet werden, ist der Defekt wiederum in den ersten Gliedern der Verarbeitungskette zu suchen.

Abschließend sei noch auf ein Phänomen hingewiesen, welches im Diagramm 4.11 `a)` erkennbar ist und als t_0-*Rauschen* bezeichnet wird. Während dieses Ereignisses signalisierte die Elektronik Zeitscheibe für Zeitscheibe eine neu getroffene Triggerlage in den Modulen des Triggersektors 4. Da den Signalen des Readout zufolge lediglich Modul 19 getroffen wurde, ist dies physikalisch unmöglich. Welches elektronische Bauelement für diese Fehlfunktion verantwortlich ist, bleibt ungewiß. Das t_0-Rauschen wurde im Rahmen dieser Analyse lediglich im Triggersektor 4 beobachtet, Lagenkoinzidenzsignale waren davon nicht betroffen.

Problematisch hierbei ist, daß diese falsch erzeugten t_0-Signale in die Generierung der Triggerelemente einfließen, wie sich zeigen läßt, und somit das Ergebnis verfälschen. Einen Einfluß auf die Hardware-Effizienz, wie sie mit den Methoden der Klassenbibliothek ermittelt wurde, übt dieses Phänomen nicht aus; es werden hierbei generell mehr Triggerelemente erzeugt als im korrekten Fall, niemals aber weniger.

Kapitel 5

Weitere Analysen

5.1 Die Bestimmung von Lageneffizienzen

Durchquert ein geladenes Teilchen die Lagen eines Moduls im Zentralen Myon-Detektor, so sollten alle getroffenen Lagen ein entsprechendes Signal an die unmittelbar auswertende Elektronik liefern. Eine Lage gilt im Rahmen eines Ereignisses als ineffizient, wenn sie nachweislich getroffen wurde, das Signal aber ausblieb.

5.1.1 Prinzipielle Vorgehensweise

Ob eine Lage getroffen wurde oder nicht, kann mit Hilfe der Informationen ermittelt werden, die in der BOS-Bank ITKR[1] abgelegt werden. ITKR verzeichnet Parameter und Ergebnisse der Spurrekonstruktion, die von dem Programm ITREC durchgeführt wird, wobei für jede Spur ein Datensatz existiert. Neben Größen wie Teilchenenergie, räumlichen und zeitlichen Koordinaten der Spur, Qualität der Spuranpassung (Kovarianzmatrix, etc.), Index der getroffenen Lagen und verwandten Daten ist dort auch ein Zeiger verfügbar, der im Index der BOS-Bank ITPR[2] auf einen der Spur zugeordneten Datensatz weist.

ITPR verzeichnet ebenfalls einen Datensatz pro rekonstruierter Spur und enthält zunächst die Zahl der getroffenen Drähte, die zur Rekonstruktion herangezogen wurden. Weiterhin sind Zeiger vermerkt, die wiederum auf einen Datensatz der BOS-Bank ITWX[3] verweisen. ITWX liegt parallel zur BOS-Bank IRWE und liefert für jeden Datensatz einen Zeiger auf denjenigen Eintrag in IRWE, welcher entlang der Spur auf den aktuellen Eintrag folgt.

Die Vorgehensweise läßt sich also folgendermaßen zusammenfassen. Von einem Eintrag in ITKR (d.h. einer rekonstruierten Spur) ist zunächst der Zeiger s auf den entsprechenden Datensatz der ITPR zu extrahieren. Dieser enthält einen weiteren Zeiger d, der auf denjenigen Datensatz (d.h. getroffenen Draht) in IRWE verweist, welcher als erster in die Spurrekonstruktion einbezogen wurde. d referenziert gleichzeitig aber auch den Datensatz in ITWX, welcher auf den nächsten berücksichtigten getroffenen Draht verweist.

[1] ITKR: *Iron Track Reconstruction*
[2] ITPR: *Iron Track Pointer*
[3] ITWX: *Iron Track Wire Relation*

Dessen Index ist nun für d einzusetzen und der zweite Schritt zu wiederholen, usw., so daß auf diese Weise der Verlauf der Spur Draht für Draht nachvollzogen werden kann. Verweist ITWX wieder auf den ersten getroffenen Draht, so wurde die Spur vollständig durchlaufen.

Aus der nun bekannten Teilchenspur kann die Effizienz einer Folge von Lagen im Rahmen des Ereignisses ermittelt werden. Mit Hilfe zweier Methoden der Klasse `TMuITKR` lassen sich die oberste sowie die unterste getroffene Lage ermitteln. Damit ist jetzt ein Kriterium für die Beurteilung der dazwischen befindlichen Lagen geschaffen, denn diese sollten nun ebenfalls ein positives Signal geliefert haben, was überprüft werden kann. Dabei muß darauf geachtet werden, daß die Spur lediglich ein einzelnes Modul des Zentralen Myon-Detektors durchquert hat. Ein offensichtlicher Nachteil dieser Methode besteht allerdings darin, daß über die jeweils oberste und unterste Lage eines Moduls keinerlei Aussage getroffen werden kann. Ihre Effizienz ist so nicht zu bestimmen, weshalb in diesem Falle auf erweiterte Kriterien zurückzugreifen ist.

Bei der Durchführung einer solchen Analyse treten jedoch eine Reihe von Schwierigkeiten auf. Wie den Konstruktionsplänen des Zentralen Myon-Detektors im Anhang zu entnehmen ist, sind die Myonboxen, welche (sofern vorhanden) die innersten drei sowie die äußersten drei Lagen eines Moduls beherbergen, nicht in jedem Falle exakt bei dem Modul installiert. Vielmehr sind die Myonboxen in einigen Fällen in vertikaler Richtung oder auch horizontal senkrecht zur Strahlachse versetzt angebracht (schematisch dargestellt in Abb. 5.1 `a)`). In einem solchermaßen konstruierten Modul kann ein Myon prinzipiell zwei Lagen

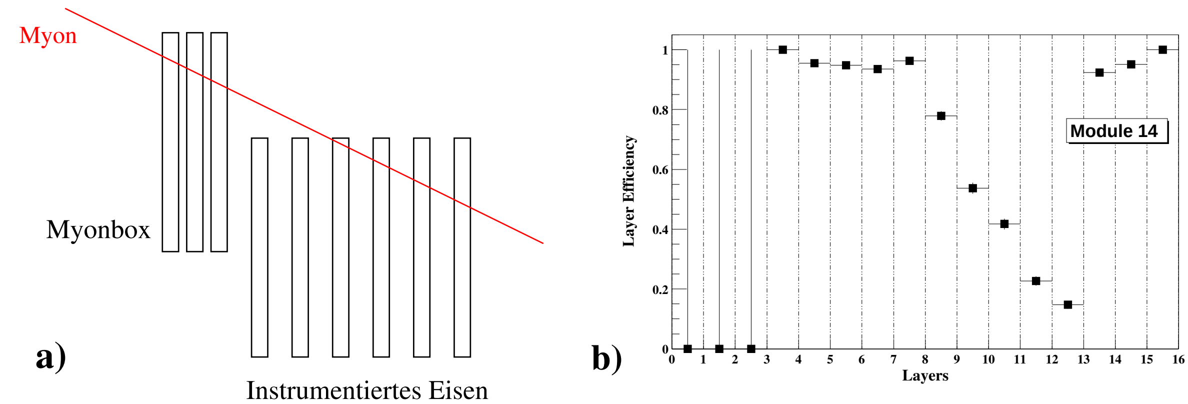

Abbildung 5.1: `a)` Schematische Darstellung einer versetzt angebrachten Myonbox und einer das Modul durchquerenden Myon-Spur. Dargestellt sind die einzelnen Lagen. `b)` Die dadurch verursachte scheinbare Verringerung der Lageneffizienzen am Beispiel von Modul 14. Es ist die Lageneffizienz über die Lagennummer aufgetragen.

durchqueren, eine oder mehrere dazwischenliegende Lagen aber möglicherweise nicht, wie in der Abbildung dargestellt ist. Entsprechend der oben beschriebenen Vorgehensweise würde diese nicht getroffene Lage nun als ineffizient gezählt werden, obwohl dafür keinerlei Anhaltspunkt besteht. Es zeigt sich, daß aufgrund der uneinheitlichen Installation der Myonboxen an den einzelnen Modulen diese Fehlerquelle nicht präzise ausgeschlossen werden kann. Insofern und da die Lagen innerhalb der Myonboxen keine Signale an den Myon-Trigger liefern, muß auf deren Betrachtung verzichtet werden.

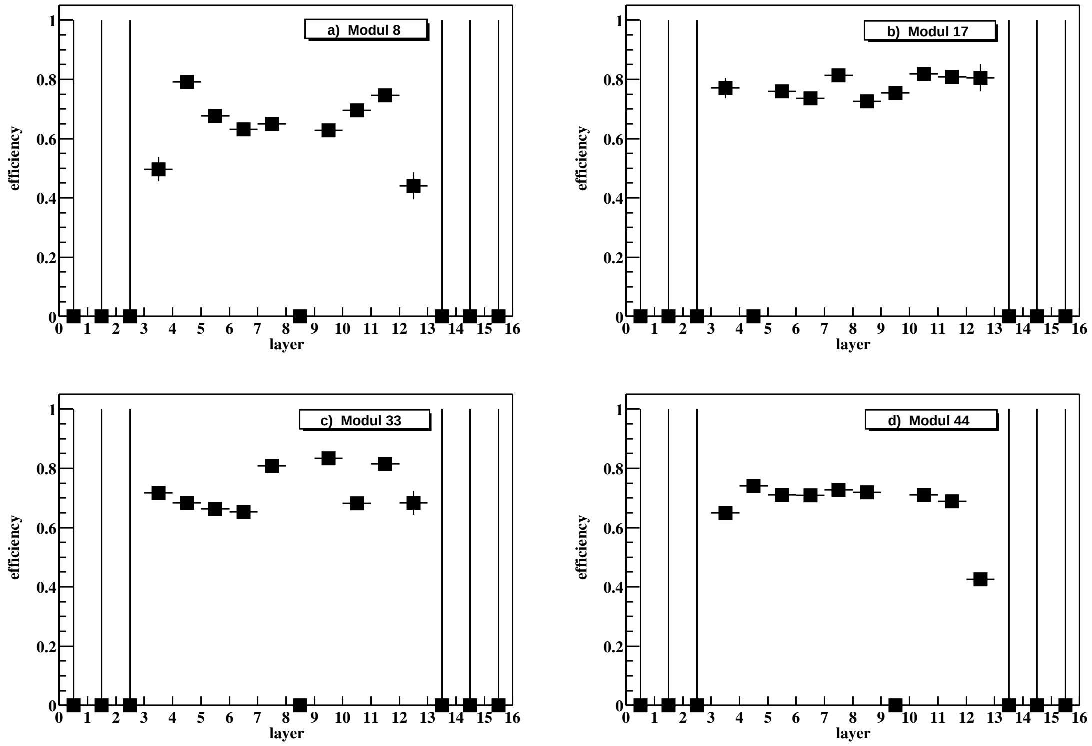

Abbildung 5.2: Die Lageneffizienzen für vier ausgewählte Module. Es sind deutlich die defekten Lagen zu erkennen, deren Effizienz gleich null ist. Die Lagen 0 bis 2 sowie 13 bis 15 sind nicht berücksichtigt. Auch zeigt sich die Schwäche des Kriteriums bezgl. der Lagen 3 und 12.

Eine weitere Schwierigkeit besteht, wie bereits weiter oben angedeutet, in der Begutachtung der Lagen 3 und 12, also der innersten bzw. äußersten Lage innerhalb des Instrumentierten Eisens. Über diese kann mit Hilfe des erwähnten Kriteriums keinerlei Aussage getroffen werden. Tatsächlich stellt es sich als unmöglich heraus, für jedes einzelne Modul zufriedenstellende Zahlenwerte der Effizienz dieser Lagen zu finden.

5.1.2 Die Klasse `TMuITKRLoop`

Die oben beschriebene Methode ist in der ausführenden Klasse `TMuITKRLoop` realisiert. `TMuITKRLoop` sollte nach Möglichkeit mit den Daten kosmischer sowie Halo-Myonen betrieben werden, welche die bessere Spurqualität gegenüber den Daten normaler Runperioden aufweisen. Das Objekt der Klasse liefert als Ergebnis einen Satz von 64 Histogrammen, welche jeweils die Lageneffizienzen der zugehörigen Module zeigen. Diese Effizienzhistogramme mit der Bezeichnung `LayerEffHisto_<Modulnummer>` (wobei `<Modulnummer>` von 0 bis 63 rangiert) werden intern wieder durch Division zweier Histogramme gebildet; eines davon (Bedingungshistogramm) zählt für jede Lage des Moduls die Ereignisse, in welchen sie ein positives Signal hätte liefern müssen, das andere wird gefüllt, wenn zusätzlich das Signal von dieser Lage auch registriert wurde (Konformitätshistogramm). Die Bedingungshistogramme sind mit `LayerCondHisto_<Modulnummer>` bezeichnet, die

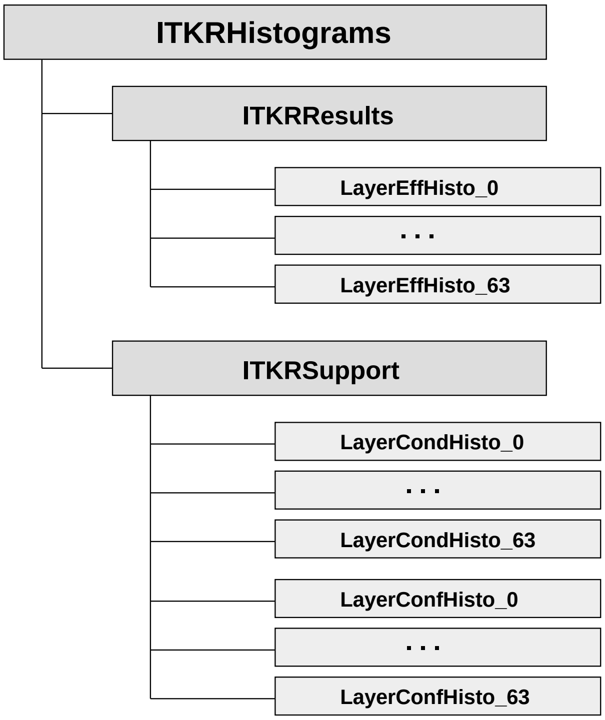

Abbildung 5.3: Der von `TMuITKRLoop` in der Ausgabedatei erzeugte Verzeichnisbaum und die darin gespeicherten Histogramme.

Konformitätshistogramme mit `LayerConfHisto_<Modulnummer>`. Die Division von Konformitätshistogramm und Bedingungshistogramm wird Lage für Lage durchgeführt und liefert die Werte des Effizienzhistogramms. Eine davon abweichende Vorgehensweise ist zur Erfassung der Effizienz der Lagen 3 und 12 implementiert. Hier kann nur indirekt darauf geschlossen werden, ob eine Lage von einem Myon getroffen wurde oder nicht. Als Kriterium hierfür wird überprüft, ob in der jeweils benachbarten Lage 4 bzw. 11 ein Treffer in einer der vier Zeitscheiben des Readout verzeichnet ist. Jedoch muß der Verlauf der Spur dahingehend eingeschränkt werden, daß nur die Drähte in der Mitte dieser Lagen (hier diejenigen von Element 3) getroffen wurden. Meldet nun zusätzlich die zentrale Lage 7 einen Treffer, so kann geometrisch auf die Durchquerung der äußeren Lagen geschlossen werden.

Dieses Kriterium weist jedoch einen Mangel auf, denn es kann auf diese Weise nicht mehr auf die Effizienz der Lage insgesamt geschlossen werden, sondern nur auf einen gewissen Teilbereich. Befindet sich dort beispielsweise ein nicht instrumentierter Abschnitt der Lage, so wird die Effizienz über Gebühr verringert. Tatsächlich konnte kein akzeptables Kriterium zur Begutachtung dieser Lagen gefunden werden, so daß ihre Effizienz in einigen Fällen nur qualitativ von Bedeutung ist.

Ist `TMuITKRLoop` in ein lauffähiges Programm integriert, wird in der obersten Ebene der Ausgabedatei ein Verzeichnis namens `ITKRHistograms` erzeugt, in welchem das Objekt die gesamten Ergebnisse der Berechnung ablegt. `ITKRHistograms` selbst gliedert sich wiederum in zwei Unterverzeichnisse; `ITKRResults` nimmt die Effizienzhistogramme auf, `ITKRSupport` die Konformitäts- und Bedingungshistogramme (s. Abb. 5.3).

Die Lageneffizienzen von vier ausgewählten Modulen sind in der Abb. 5.2 wiedergegeben. Deutlich sind diejenigen Lagen zu erkennen, die überhaupt keine Treffer signalisiert haben und deren Effizienz somit gleich null ist. Bei dreien der Module ist jeweils eine Triggerlage betroffen. Dies ist besonders problematisch im Hinblick auf die im dritten Kapitel besprochene Triggereffizienz, die durch derartige Defekte beeinträchtigt wird. Die Methoden der Klassenbibliothek sind nicht sensibel für diesen Effekt.

5.2 Das Auffinden von Ereignissen mit hoher Myon-Aktivität

Von Zeit zu Zeit registriert das H1-Experiment Ereignisse mit besonders hoher Aktivität im Zentralen Myon-Detektor. Durchschnittlich weist ca. jedes 800. Ereignis normaler Runperioden, welches Treffer in den Modulen des Zentralen Myon-Detektor beinhaltet, mehr als 450 getroffene Drähte im Instrumentierten Eisen auf; ein solches wird als *big muon event* bezeichnet. Eine Abschätzung der Rate geschah mit Hilfe der Daten von vier Halo-Myon-Runperioden. Während der Dauer von zusammengenommen 60.991 Sekunden wurden insgesamt dreiundzwanzig Ereignisse mit mehr als 450 Treffern im Zentralen Myon-Detektor registriert. Dies ergibt eine Rate von $(0,00038 \pm 0,00008)\ s^{-1}$, was etwa einem Ereignis in 44 Minuten entspricht. Als Quelle solcher Ereignisse kommen Schauer kosmischer Myonen in Frage, ein Beispiel hierfür ist in Abb. 5.4 dargestellt.

5.2.1 Die Klasse `TMuBIGLoop`

Die ausführende Klasse `TMuBIGLoop` kann zur Identifizierung von Ereignissen mit mehr als 450 getroffenen Drähten eingesetzt werden. Darüberhinaus ist sie auch zur Suche nach verrauschten Drähten verwendbar, worauf weiter unten eingegangen wird. `TMuBIGLoop` zählt die Treffer pro Ereignis und sammelt die Ergebnisse in einer Reihe verschiedener Histogramme (s. Tab. 5.1). Das Objekt der Klasse beschränkt sich dabei nicht auf Ereignisse mit hoher Aktivität im Zentralen Myon-Detektor; alle Ereignisse der zu untersuchenden Datenmenge werden bearbeitet. Weiterhin wird, falls ein Ereignis mehr als 450 Treffer aufweist, eine Meldung auf dem Bildschirm ausgegeben. `TMuBIGLoop` erstellt ebenfalls eine Datei namens `bigmuon.list`, in welche die Nummern von Runperiode und Ereignis in tabellarischer Form niedergeschrieben werden. Diese kann beispielsweise in Verbindung mit dem Programm `h1ed`[4] zur Visualisierung der Ereignisse dienen (vergl. Abb. 5.4).

`TMuBIGLoop` legt in der Ausgabedatei das Verzeichnis `BigMuonEvents` an, welches sich weiterhin in die Unterverzeichnisse `ClusterwiseHistos` (5 Histogramme `HitsPerClusterA` bis `HitsPerClusterE`) und `ModulewiseHistos` (64 Histogramme `HitsPerEvent_0` bis `HitsPerEvent_64`) gliedert. Eine Übersicht der Verzeichnisstruktur gibt die Abb. 5.5. Die Klasse kann mit den Rohdaten normaler Runperioden verwendet werden.

[4]`h1ed`: H1 Event Display

[5]Die Zugehörigkeit der Module zu den Clustern ist aus Abb. 1.3 ersichtlich.

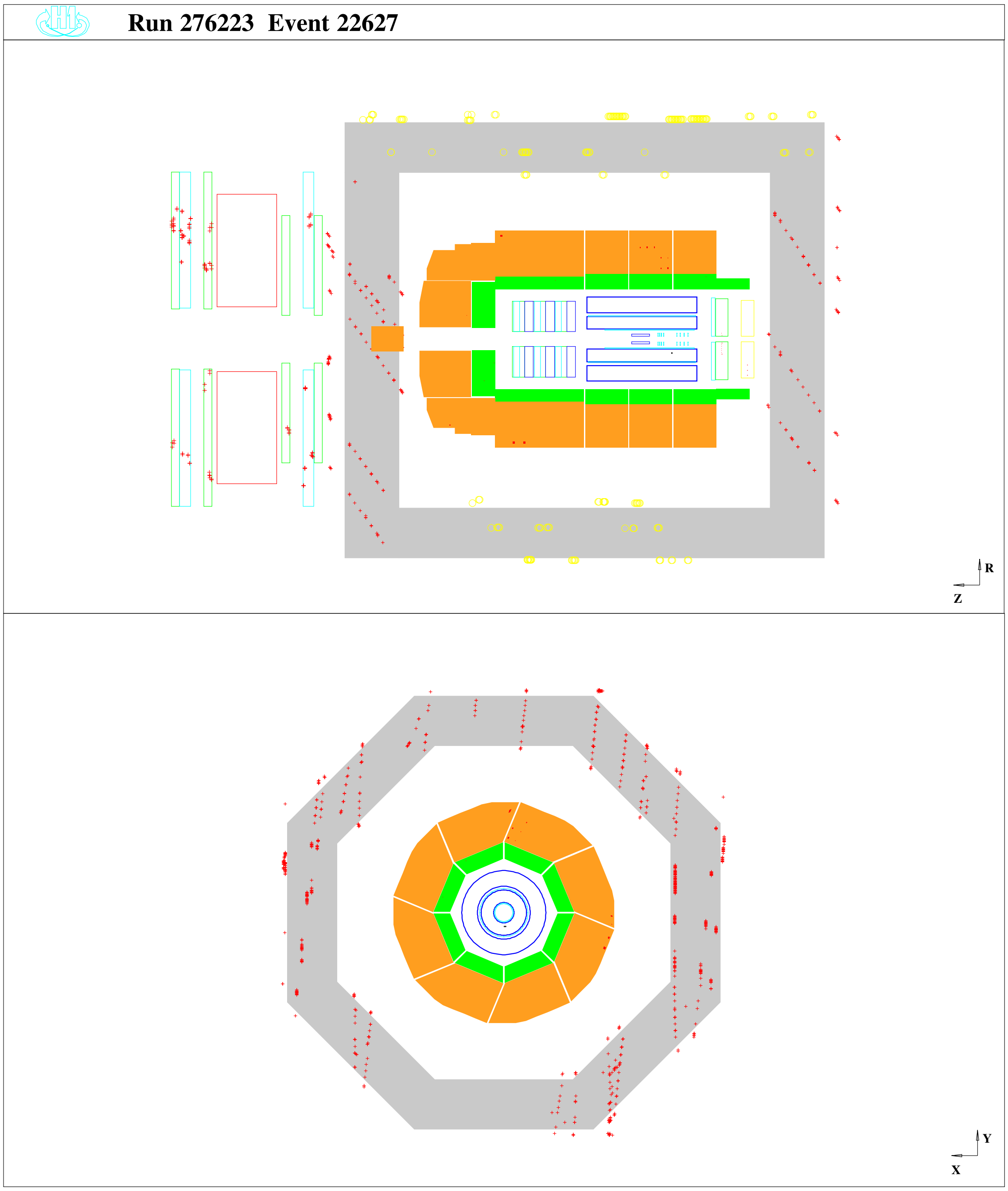

Abbildung 5.4: Ein typisches Beispiel eines Schauers kosmischer Myonen. Dieser durchquerte am 12. Juli 2000 das Experiment und verursachte eine hohe Aktivität im Myon-Detektor. In diesem Falle meldeten 1053 Drähte einen Treffer.

In Abb. 5.6 ist die Verteilung der getroffenen Drähte auf die vierundsechzig Module dargestellt. In diesem Histogramm sind lediglich die während der Ereignisse mit hoher Aktivität getroffenen Drähte berücksichtigt. Es zeigt sich eine Verteilung mit einigen signifikanten Häufungen und einem dominanten Eintrag für das Modul 29. Der Grund für die Vielzahl getroffener Drähte in Modul 29 ist jedoch völlig unklar. Es handelt sich hierbei

Histogramm	Inhalt
`HitsPerModule`	Die Gesamtzahl der Treffer für jedes der Module
`HitsPerEvent`	Die Zahl der Ereignisse mit n Treffern, aufgetragen über n
`HitsPerClusterA` ... `HitsPerClusterE`	Die Zahl der Ereignisse mit n Treffern im Cluster[5] A ... E, aufgetragen über n
`HitsPerEvent_0` ... `HitsPerEvent_63`	Die Zahl der Ereignisse mit n Treffern im Modul 0 ... 63, aufgetragen über n

Tabelle 5.1: Übersicht der von `TMuBIGLoop` erzeugten Zählhistogramme.

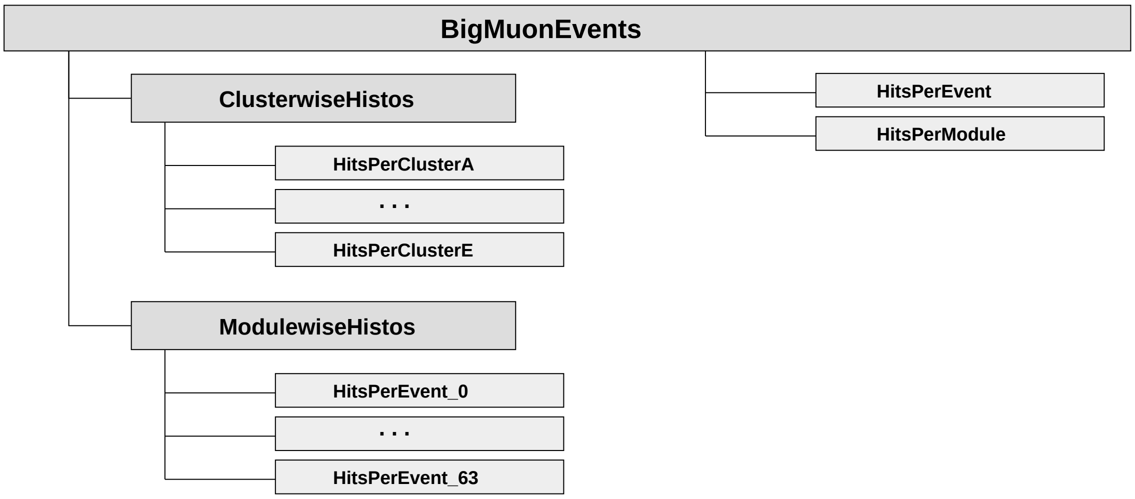

Abbildung 5.5: Übersicht des von `TMuBIGLoop` in der Ausgabedatei angelegten Verzeichnisses `BigMuonEvents`, der Unterverzeichnisse sowie der erzeugten Histogramme.

nicht um den Effekt eines verrauschten Drahtes (s. nächster Unterabschnitt). Ein Vergleich mit dem Histogramm für Modul 29 in der Abb. C.36 des Anhangs zeigt, daß in diesem Modul überdurchschnittlich häufig große Zahlen von Treffern pro Ereignis auftreten[6]. Möglicherweise liegt hier eine Fehlfunktion der unmittelbar auslesenden Elektronik vor.

Die Abb. 5.7 zeigt hingegen die Verteilung der Häufigkeit getroffener Drähte auf die Module, wie sie aus allen Ereignissen der untersuchten Datenmenge gewonnen wurde (keine Beschränkung auf Ereignisse mit hoher Aktivität). Es fällt sofort die große Zahl der Treffer in den inneren Endkappen auf, insbesondere in der stark mit Untergrundteilchen belasteten FIEC. Daß die rückwärtige innere Endkappe (BIEC) im Gegensatz zum Barrelbereich ebenfalls eine deutliche Häufung von Treffern aufweist, ist durch die dort

[6]Dieses Histogramm wird in Abschn. 5.5 erläutert.

fehlende Abschirmung durch das Flüssig-Argon-Kalorimeter und die Supraleitende Spule begründet.

Weitere Häufungen treten, unterschiedlich ausgeprägt, auch an den jeweils obersten und untersten Modulen des Barrelbereiches auf. Der Grund hierfür ist in der Vielzahl der kosmischen Myonen zu suchen, die in das Experiment gelangen und deren Verteilung bei einem Winkel von 0° zur Senkrechten ein Maximum aufweist.

Keine physikalische Erklärung hingegen liegt für die unerwartet große Häufigkeit von Treffern in den Modulen 54 und 62 vor. Vielmehr wird hier ein Defekt innerhalb der Instrumentierung sichtbar, der mit der Klasse `TMuBIGLoop` zuverlässig detektiert werden kann.

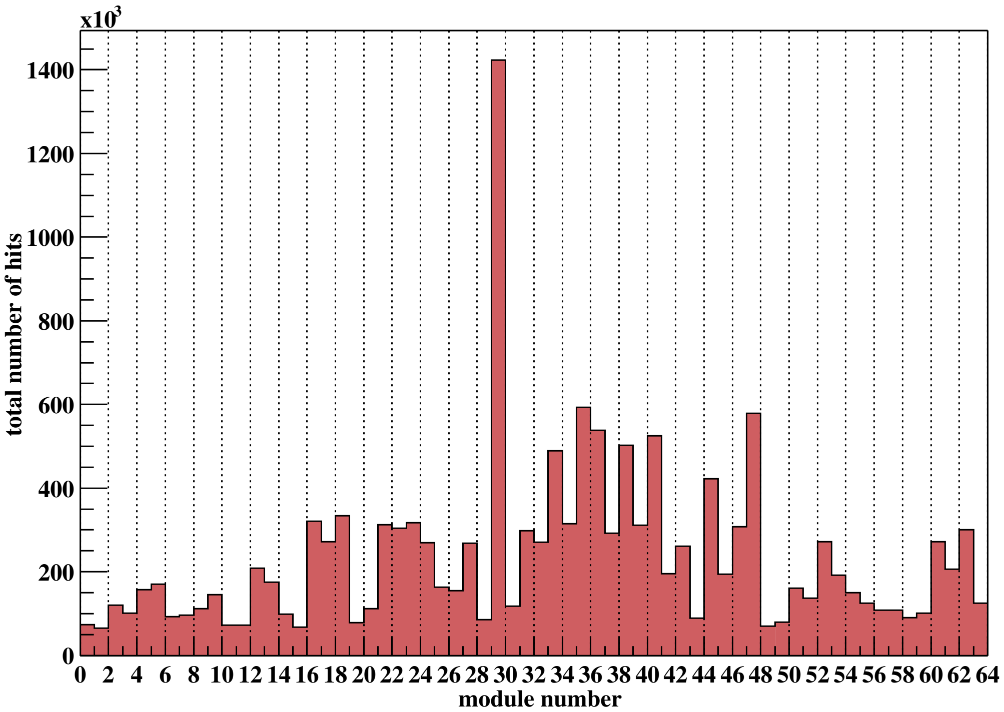

Abbildung 5.6: Die Verteilung der Häufigkeit getroffener Drähte auf die 64 Module. Es wurden nur *big muon events* berücksichtigt, d.h. mehr als 450 Drähte wurden während der hier betrachteten Ereignisse getroffen.

5.2.2 Der Nachweis verrauschter Drähte mit `TMuBIGLoop`

Die Abb. 5.8 zeigt das Histogramm `HitsPerEvent_62`, wie es von `TMuBIGLoop` erzeugt wird, in logarithmischer Skalierung. Dargestellt ist die Häufigkeit von Ereignissen mit n getroffenen Drähten in Modul 62, aufgetragen über n. Deutlich tritt die übermäßige Anzahl von Ereignissen hervor, in deren Verlauf genau vier der Drähte von Modul 62 einen Treffer signalisieren.

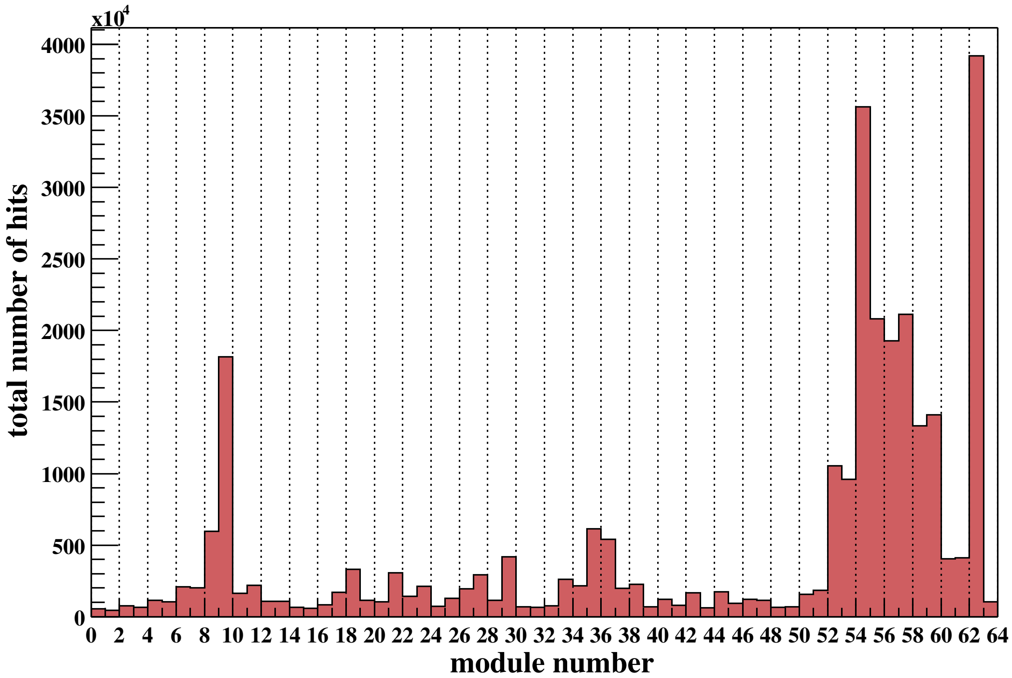

Abbildung 5.7: Die Verteilung der Häufigkeit getroffener Drähte auf die 64 Module. Es wurden alle Ereignisse der untersuchten Datenmenge berücksichtigt.

Wie bereits im einleitenden Kapitel beschrieben, werden bei der Auslese des Detektors vom Readout die Zeitscheiben -1 bis 2 berücksichtigt. Liefert nun ein Draht aufgrund eines Defektes ein permanentes Signal an die Elektronik, so wird dies in allen vier ausgelesenen Zeitscheiben registriert. Da nur in einem kleinen Teil der Fälle Myonen ein bestimmtes Modul durchqueren, wird die Zahl der Ereignisse mit vier Treffern dominieren, was sich in den entsprechenden Zählhistogrammen deutlich abzeichnet.

Solch ein als Rauschen bezeichnetes Verhalten kann durch eine defekte Aufhängung und ein damit verbundenes Durchhängen des Drahtes verursacht werden. Auch Beschädigungen von Elementen oder Kabeln sowie ein Fehlverhalten der unmittelbar auslesenden Elektronik kommen als Gründe hierfür in Frage.

Um die Hinweise auf verrauschte Drähte auszuwerten, nimmt `TMuBIGLoop` abschließend eine Durchmusterung der modulweise erzeugten Histogramme vor. Es wird hierbei die Häufigkeit des Auftretens von Ereignissen mit bestimmten Anzahlen[7] von Treffern im Vergleich zur durchschnittlichen Anzahl bestimmt. Überschreitet diese einen gewissen Schwellenwert, so kann das Vorhandensein eines verrauschten Drahtes als gesichert gelten, und eine entsprechende Meldung wird auf dem Bildschirm ausgegeben. Das betroffene Modul ist somit bestimmt.

Wie sich zeigen läßt, ist das hier angewendete Kriterium, einen verrauschten Draht in einem Modul zu identifizieren, sehr zuverlässig. Auch bei einer geringen Zahl ausgewerteter Ereignisse sind die betreffenden Einträge der Histogramme signifikant gegenüber

[7]Es sind dies alle ganzzahligen Vielfachen von vier.

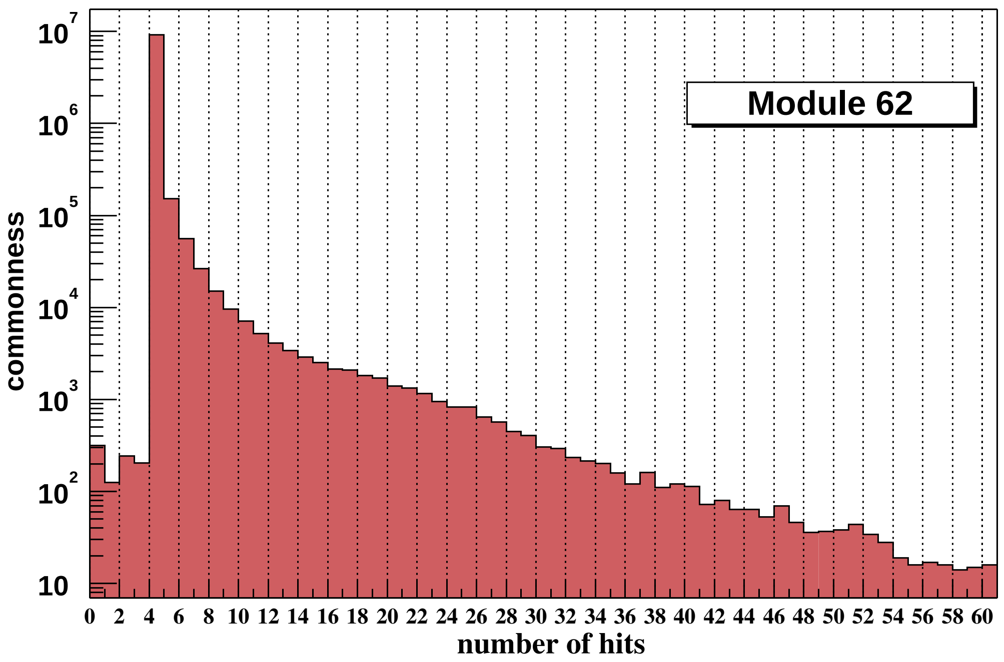

Abbildung 5.8: Die Häufigkeit von Ereignissen mit einer bestimmten Anzahl getroffener Drähte in Modul 62 (Ausschnitt). Der hohe Wert an der Stelle 4 ist ein sicherer Hinweis auf einen verrauschten Draht in diesem Modul.

dem Durchschnitt erhöht. Die Anpassung des Schwellenwertes ist auf empirischem Wege geschehen; eine weitergehende Optimierung ist durchaus denkbar. Ein zusätzlicher Vorzug der Methode ist ihre Schnelligkeit, da keinerlei aufwendige Algorithmen Verwendung finden, sondern lediglich simple Zählhistogramme erstellt und ausgewertet werden.

5.3 Die Lokalisierung verrauschter Drähte

Die im vorangegangenen Abschnitt vorgestellte Methode zur Identifizierung verrauschter Drähte kann zu deren genauer Lokalisierung weiter ausgebaut werden. Zu diesem Zweck sind die Signale des Readout zu zählen, die von den einzelnen Lagen bzw. den einzelnen Drähten eines betroffenen Moduls verursacht wurden. Ist ein Draht defekt und liefert ein permanentes Signal an die unmittelbar auswertende Elektronik, so resultiert dies wiederum in einer unangemessen großen Häufigkeit gemeldeter Treffer. Solche Fälle sind leicht und zuverlässig detektierbar.

5.3.1 Die Klasse `TMuNoisyLoop`

Sind ein oder mehrere verrauschte Drähte von `TMuBIGLoop` in einem der Module erkannt worden, so kann die Klasse `TMuNoisyLoop` Verwendung finden, um diese exakt zu lokalisieren. `TMuNoisyLoop` nimmt keine vollständige Durchmusterung aller Module des Zentralen

Histogramm	Inhalt
`Overview_<Modulnummer>`	Die Gesamtzahl der getroffenen Drähte für jede der bis zu 16 Lagen von Modul **<Modulnummer>**
`Layer_0` ... `Layer_15`	Die Gesamtzahl der Treffer für jeden Draht der jeweiligen Lage, aufgeschlüsselt nach Elementen und Drahtnummern

Tabelle 5.2: Übersicht der von `TMuNoisyLoop` erzeugten Zählhistogramme.

Myon-Detektors vor, sondern beschränkt sich auf ein einzelnes vorzugebendes Modul. Auf diese Weise bleibt die Zahl der erzeugten Histogramme in einem vertretbaren Rahmen.

`TMuNoisyLoop` erzeugt das Verzeichnis `NoisyHistograms` in der Ausgabedatei, sofern das Objekt der Klasse aktiviert wurde. In diesem Verzeichnis findet sich zunächst das Histogramm `Overview_<Modulnummer>`[8], welches die Verteilung der Treffer auf die bis zu 16 Lagen des Moduls zeigt (s. Abb. 5.9 `a)`).

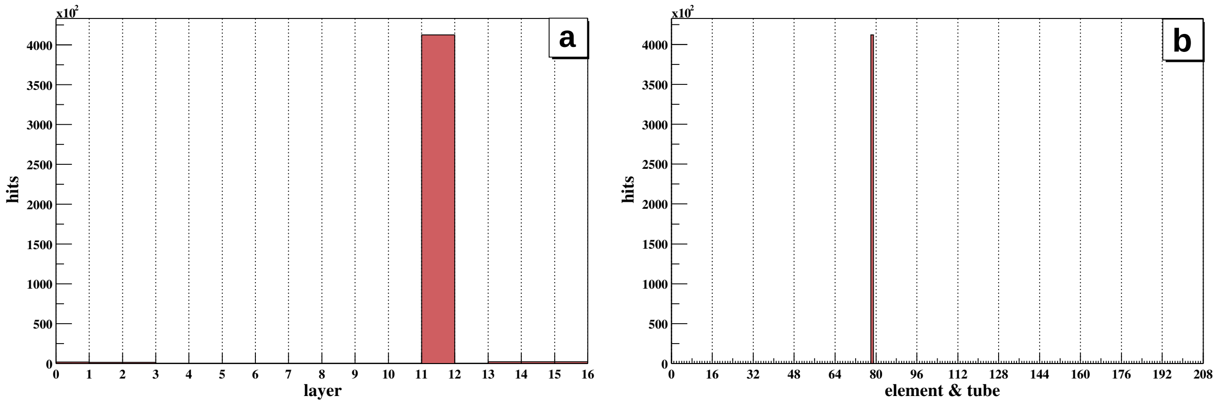

Abbildung 5.9: Zur Lokalisierung verrauschter Drähte am Beispiel von Modul 62. Die linke Abbildung `a)` zeigt eine Übersicht der 16 Lagen mit der Zahl der Treffer, die rechte Abbildung `b)` die einzelnen Drähte in Lage 11.

Anhand dieses Histogramms kann die betroffene Lage identifiziert werden; in diesem Falle ist es die Lage 11, wie einwandfrei zu erkennen ist. Im Unterverzeichnis `Layers` befinden sich 16 weitere Histogramme, eines für jede Lage, in welchen die Gesamtzahl der Treffer für jeden einzelnen Draht abgelesen werden kann. Das entsprechende Histogramm für Lage 11 zeigt Abb. 5.9 `b)`. Erneut tritt einer der Einträge des Histogramms deutlich hervor, und es kann nun dessen genaue Position abgelesen werden. Die Abszisse ist in Abständen von je 16 Einheiten (Drähten) beschriftet, was einem Element entspricht. So-

[7] Die Zugehörigkeit der Module zu den Clustern ist aus Abb. 1.3 ersichtlich.

[8] `<Modulnummer>` bezeichnet die Nummer desjenigen Moduls, welches `TMuNoisyLoop` zur Untersuchung vorgegeben wurde.

mit können nun alle die Position des Drahtes kennzeichnenden Angaben durch Abzählen ermittelt werden: Modul 62, Lage 11, Element 4, darin Draht 14.

Zur Kontrolle ermittelt auch `TMuNoisyLoop` diese Werte und gibt sie abschließend auf dem Bildschirm aus. Diese sind jedoch lediglich als Hinweis zu verwenden, auch kann so nur ein einzelner Draht aufgefunden werden. Eine Durchsicht der Histogramme bleibt in jedem Falle notwendig.

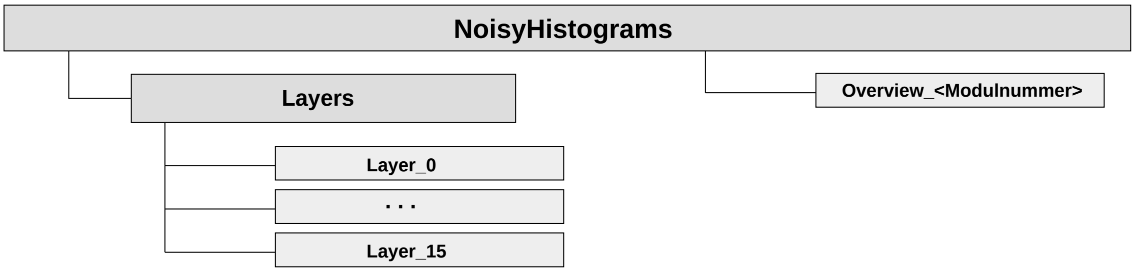

Abbildung 5.10: Übersicht des Verzeichnisses `NoisyHistograms`, wie es von der Klasse `TMuNoisyLoop` in der Ausgabedatei angelegt wird.

`TMuNoisyLoop` kann für seine Aufgabe die Rohdaten beliebiger Runperioden verwenden, sofern die BOS-Bank IRWE damit zur Verfügung steht. Beim Aufruf der Klasse ist die Nummer des zu untersuchenden Moduls der Methode `Init()` als Parameter zu übergeben. Die bereits besprochene Klasse `TMuLoop` zur einfachen Implementierung der hier beschriebenen Analyseverfahren enthält zu diesem Zweck bereits einen vorbereiteten Befehlsaufruf.

Die Histogramme, die mit Hilfe von `TMuNoisyLoop` erstellt werden können, erwiesen sich darüberhinaus auch als überaus nützlich zur Begutachtung des Zustandes der Module generell. Steht eine hinreichend große Datenmenge zur Verfügung, so besteht hier die Möglichkeit, einen Überblick über die Gesamtheit der Drähte und ihre Funktion zu erhalten. Inaktive Profile, Elemente oder gar Lagen lassen sich auf diese Weise schnell erkennen. Die tatsächliche Ausstattung der Module mit Myonboxen ist bereits an den jeweiligen Übersichtshistogrammen ablesbar - dies ist eine ansonsten recht schwierig verifizierbare Information.

5.4 Die zeitliche Verteilung von Koinzidenzen

Bislang sind erfüllte Triggerbedingungen anhand des Readout ausschließlich in der Zeitscheibe 0 betrachtet worden[9]. Es kann generell von Interesse sein, die Häufigkeit erfüllter Triggerbedingungen in benachbarten Zeitscheiben zu erfassen. Darüberhinaus besteht die Möglichkeit, Lagenkoinzidenzen zu detektieren, die keinerlei zeitlichen Einschränkungen unterworfen sind; solche erstrecken sich über alle vier Zeitscheiben des Readout und sind damit keine Koinzidenzen im ursprünglichen Sinne mehr.

Wie später deutlich wird, spiegeln sich auch bei diesen Betrachtungen etwaige zeitliche Ineffizienzen mancher Module in den Ergebnissen wider. Bei einwandfreier Triggeref-

[9] Eine Ausnahme bildet das Histogramm `TEL1EffHistoR3BC` der Klasse `TMuTEL1Loop`.

fizienz bietet die Analyse auch einen Überblick der mittleren zeitlichen Verteilung von Myondurchquerungen im Zentralen Myon-Detektor.

5.4.1 Die Klasse TMuShiftLoop

Anhand von Rohdaten normaler Runperioden, welche die BOS-Bank IRWE umfassen, kann die Klasse `TMuShiftLoop` die zeitliche Verteilung von Lagenkoinzidenzen für jedes der Module ermitteln. `TMuShiftLoop` erzeugt in der Ausgabedatei das Verzeichnis `ShiftHistograms` und legt dort vierundsechzig Histogramme namens `ShiftTZero_<Modulnummer>` ab.

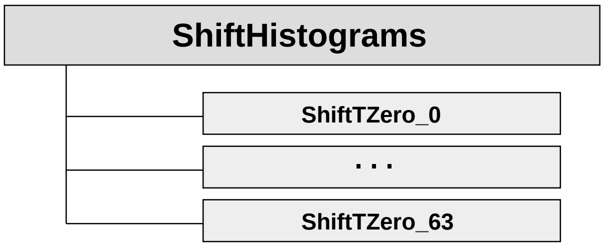

Abbildung 5.11: Das von `TMuShiftLoop` in der Ausgabedatei angelegte Verzeichnis `ShiftHistograms`.

Jedes der Histogramme umfaßt fünf Einträge; diejenigen mit der Bezeichnung -1, 0, 1 zeigen die Häufigkeit von Lagenkoinzidenzen in den entsprechenden Zeitscheiben, der letzte Eintrag die Häufigkeit von Koinzidenzen ohne zeitliche Beschränkung (der Eintrag mit der Bezeichnung zwei ist unbenutzt). Alle Lagenkoinzidenzen werden anhand der Signale des Readout und mit Hilfe der Methoden der Klasse `TMuIRWE` ermittelt.

Die Ergebnisse für vier ausgewählte Module zeigt die Abb. 5.12. Bei zweien der Module ist eine deutliche Neigung hin zu Lagenkoinzidenzen in der Zeitscheibe -1 zu beobachten. In auffälliger Weise ist dies bei nahezu allen Modulen der Fall, die eine signifikante Verschiebung ihres mittleren relativen Eintreffzeitpunktes in negativer Richtung aufweisen (vergl. Abb. 3.5, Abschn. 3.1). Auf die hier zugrundeliegende Korrelation wird im nächsten Abschnitt noch detaillierter eingegangen.

In allen Fällen entfällt eine gewisse Anzahl der detektierten Lagenkoinzidenzen auf die Zeitscheiben -1 und 1. Sofern nicht die zeitliche Triggereffizienz dort eine Rolle spielt, können diese zu einem großen Teil als kosmische oder Halo-Myonen interpretiert werden, deren Eintreffen keinerlei feste Phase zur Strahlkreuzungsfrequenz von HERA aufweisen. Der Anteil der Lagenkoinzidenzen ohne zeitliche Beschränkung bleibt durchgehend in einer Größenordnung von etwa einem Prozent und darunter.

5.5 Die zeitliche Verteilung getroffener Drähte

Um den Einfluß des Readout auf die zeitliche Verteilung von Koinzidenzen erfassen zu können, bietet sich eine nach Zeitscheiben differenzierte Zählung der während eines Ereignisses getroffenen Drähte an. Mit einer solchen Vorgehensweise kann die Wirkung der

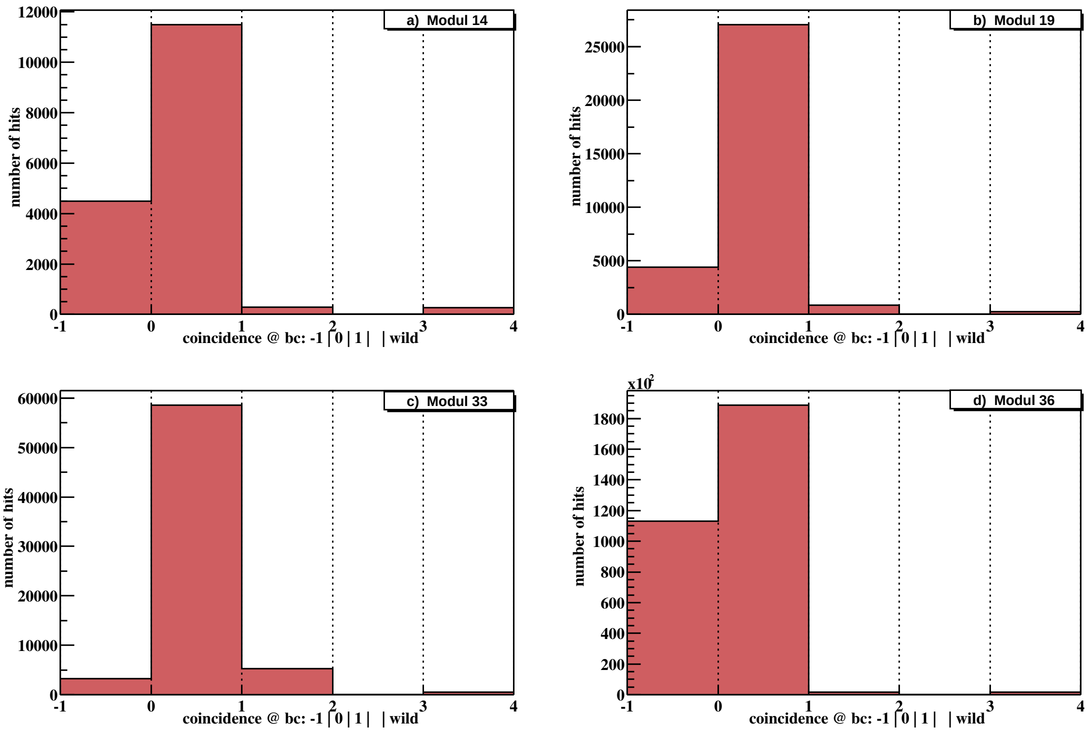

Abbildung 5.12: Die zeitliche Verteilung von Lagenkoinzidenzen in vier ausgewählten Modulen. Der letzte Eintrag jedes Histogramms zeigt die Anzahl von Koinzidenzen ohne zeitliche Beschränkung, der jeweils zweite von rechts wurde der Übersichtlichkeit halber frei gelassen.

Synchronisation der analogen Signale mit der HERA clock begutachtet werden, wie sie in der unmittelbar auslesenden Elektronik geschieht. Ein Vergleich mit den Signalen innerhalb des Myon-Triggers findet hier nicht statt.

5.5.1 Die Klasse `TMuHitsLoop`

Die ausführende Klasse `TMuHitsLoop` erstellt, sofern aktiviert, für jedes der Module ein Histogramm und zählt die pro Zeitscheibe getroffenen Drähte. Dabei werden alle vier Zeitscheiben des Readout berücksichtigt. Anschließend stellt `TMuHitsLoop` die Histogramme in dem Verzeichnis `HitsHistograms` der Ausgabedatei zur Verfügung.

Die Abb. 5.13 zeigt die Ergebnisse der Berechnung wiederum beispielhaft für vier Module. Von besonderem Interesse sind hier zunächst einmal die Fälle `a)` (Modul 33) und `b)` (Modul 36). Insbesondere das letztere Modul wies bereits bei der Ermittlung der zeitlichen Verteilung von Lagenkoinzidenzen eine deutliche Verschiebung in negativer Richtung auf, Modul 33 hingegen zeigt diese Besonderheit nicht (vergl. Abb. 5.12). Dieses Muster ist nun in den Histogrammen der Verteilung der getroffenen Drähte in die Zeitscheiben wiederzufinden, worin offensichtlich auch der Grund für die angesprochene zeitliche Verschiebung der Lagenkoinzidenzen zu suchen ist. Diese Korrelation ist bei nahezu allen übrigen Modulen ebenfalls zu beobachten, und kann mit der zeitlichen Verschiebung der

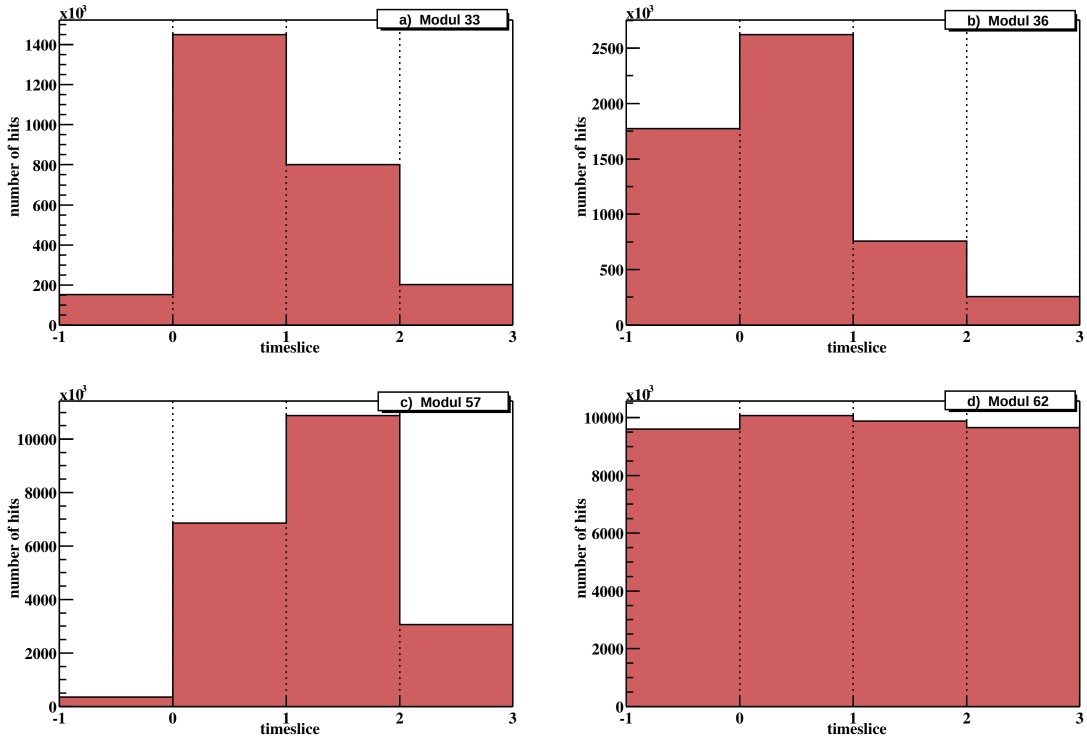

Abbildung 5.13: Die zeitliche Verteilung getroffener Drähte in vier ausgewählten Modulen, jeweils in den vier Zeitscheiben des Readout.

mittleren relativen Eintreffzeitpunkte, wie sie in Abb. 3.5, Abschn. 3.1 besprochen wurde, in Beziehung gesetzt werden.

Eine derartige Verschiebung des mittleren relativen Eintreffzeitpunktes ist dahingehend zu interpretieren, daß auch recht weit zurückliegende Durchquerungen von Myonen im Detektor ein Signal in der Zeitscheibe 0 verursachen können. Anders ausgedrückt ist die Dauer zwischen der Durchquerung und der Zeitscheibe des Signals also dementsprechend groß. Daß solche Signale nicht mit der folgenden Zeitscheibe synchronisiert werden ist eine Folge der Phase der HERA clock relativ zum wahren Eintreffzeitpunkt; je später der Takt der HERA clock eintrifft, desto größer ist die Zahl derjenigen Signale, die eine Synchronisation mit der früheren Zeitscheibe erfahren. In diesen Zeitscheiben werden somit mehr Treffer registriert, was bei Modul 36 deutlich zu erkennen ist. Einen Einfluß auf die Hardware-Effizienz besitzt dieser Effekt prinzipiell nicht, da er gleichermaßen auch auf die Signale der Triggerelektronik wirkt.

Weiterhin wird die zeitliche Verteilung der Signale auch verbreitert. Im Gegensatz zu dem sehr scharfen Maximum, welches die wahren Eintreffzeitpunkte während normaler Runperioden aufweisen (s. Abb. 5.14), ist dasjenige der diskreten Verteilung in die Zeitscheiben naturgemäß breiter.

Obige Überlegungen verlieren allerdings ihre Relevanz im Hinblick auf die Ergebnisse für Module der vorwärtigen Endkappe, insbesonderen diejenigen in der Nähe der Strahlröhre (FIEC). Die Abb. 5.13 c) für das Moul 57 macht dies beispielhaft deutlich.

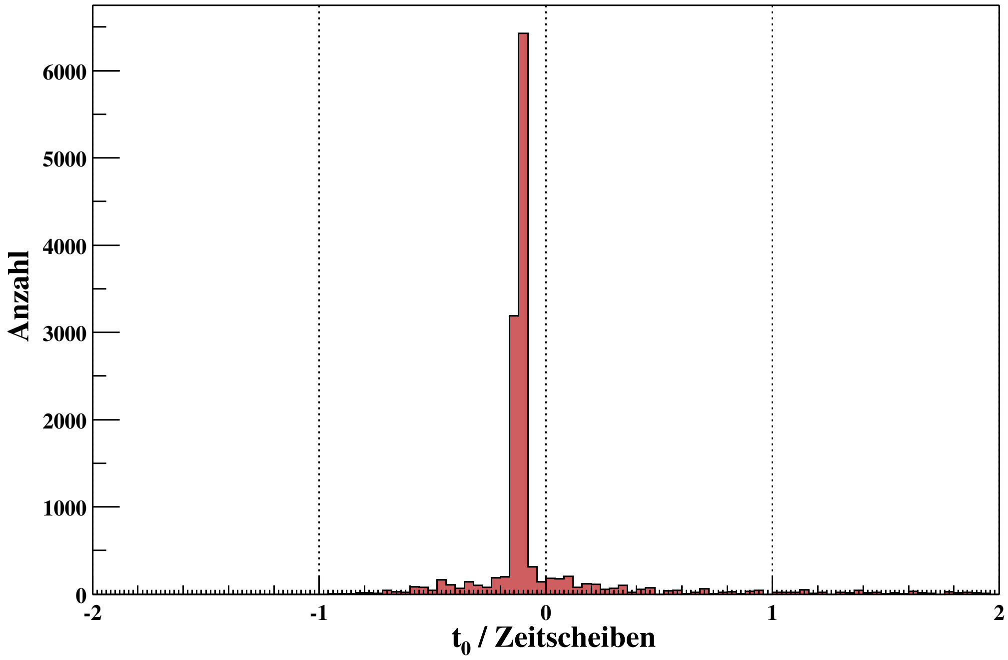

Abbildung 5.14 : Die Verteilung der wahren Eintreffzeitpunkte während einer normalen Runperiode, wie sie in den Zentralen Spurkammern gemessen wurden.

Die weitaus meisten Drähte werden in der Zeitscheibe 1 getroffen, überdurchschnittlich viele auch in der Zeitscheibe 2; die gesamte Verteilung erscheint zu späteren Zeitscheiben hin verschoben zu sein.

Dies ist auf die große Zahl langlebiger Zerfallsprodukte (häufig Pionen) zurückzuführen, die bei der $e^{\pm}p$-Streuung entstehen und aufgrund des bewegten Schwerpunktsystems von Lepton und Proton bevorzugt in die Richtung des Protonenstrahls emittiert werden. In diesem Muster sind die angesprochenen Effekte nicht mehr erkennbar. Aus diesem Grund bestehen, wie bereits erwähnt, für den vorwärtigen Teil des Zentralen Myon-Detektors strengere Anforderungen an eine zu einem Triggersignal führende Lagenkoinzidenz.

Abschließend sei noch kurz auf das Histogramm der Abb. 5.13 `d)` eingegangen, welches die zeitliche Verteilung der Treffer für Modul 62 zeigt. In Abschn. 5.3 wurde bereits erwähnt, daß sich während der Durchführung der Analyse ein verrauschter Draht im Modul befand. In der Abbildung ist nun deutlich zu erkennen, mit welcher Gleichmäßigkeit ein solcher Defekt scheinbare Treffer in den vier Zeitscheiben des Readout erzeugt.

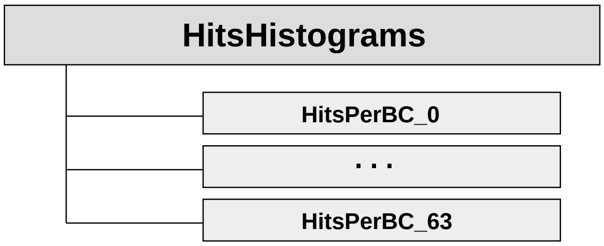

Abbildung 5.15: Übersicht des Verzeichnisses `HitsHistograms` mit den darin erzeugten Histogrammen.

Kapitel 6

Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde eine C++-Klassenbibliothek vorgestellt, welche eine Sammlung von Methoden zur Überwachung der Funktion des Zentralen Myon-Detektors umfaßt. Ein Schwerpunkt liegt dabei auf der Betrachtung des Myon-Triggersystems. Bei der Erstellung der Klassenbibliothek wurde insbesondere Wert gelegt sowohl auf eine einfache Bedienbarkeit als auch auf die Möglichkeit, die Funktionalität der Klassen zu erweitern und ggf. veränderten Bedingungen anpassen zu können.

Die zeitliche Effizienz des Myon-Triggers

Mit Hilfe der Klassenbibliothek wurde im Rahmen dieser Arbeit sowohl der zeitliche als auch der Hardware-Aspekt der Triggereffizienz des Zentralen Myon-Detektors differenziert ermittelt. Hierbei können folgende Ergebnisse festgehalten werden.

- Die zeitliche Triggereffizienz der weitaus meisten Module des Zentralen Myon-Detektors kann als gut bezeichnet werden. Es gibt jedoch einige wenige Ausnahmen.
- In dieser Hinsicht fällt hier insbesondere das Modul 36 auf, welches eine zeitliche Verschiebung des mittleren relativen Eintreffzeitpunktes von mehr als einer halben Zeitscheibe aufweist. Die Signalwege zwischen der unmittelbar auslesenden Elektronik dieses Moduls und der nachgeschalteten Elektronik sind lang im Vergleich zu fast allen anderen Modulen. Aus diesem Grunde wurden die Signale dieses Moduls bereits zeitlich um -0.3 Zeitscheiben verschoben [26].
- Geringer beeinträchtigte zeitliche Triggereffizienzen weisen weiterhin die Module 35, 41, 48, 52 und 35 auf. Bei den letztgenannten vier ist dies hauptsächlich auf die unregelmäßige Form der in Abschn. 3.1 beschriebenen Wahrscheinlichkeitsverteilungen zurückzuführen, was sich negativ auf die Qualität der Funktionsanpassung auswirkt. Ein Grund für diese Abweichungen ist nicht genau erkennbar.

Die Hardware-Effizienz des Myon-Triggers

- Die Hardware-Effizienz der meisten Module des Zentralen Myon-Detektors kann als sehr gut bezeichnet werden. Allerdings bestehen auch hier einige wenige Ausnahmen.

- Betreffend die Hardware-Effizienz treten bei sechs der Module gewisse Mängel auf. Allen voran weist das Modul 19 eine um etwa 80% verringerte Effizienz auf. Anhand der Ergebnisse der Effizienzbestimmung der Signalübermittlung an die Triggerstufe L2 kann gezeigt werden, daß die verringerten Effizienzen der Module 19, 35, 48 und 56 auf ein zeitliches Problem zurückzuführen sind. Es werden alle Lagenkoinzidenzen korrekt signalisiert, aber nicht in jedem Fall in der richtigen Zeitscheibe. Eine Betrachtung der die GPTP-Karte erreichenden Signale zeigt, daß die Lagenkoinzidenzsignale zumeist eine Zeitscheibe verfrüht dort eintreffen, was die Generierung des entsprechenden Triggerelementes unterbindet. Die Ineffizienzen lassen sich möglicherweise durch eine gezielte Verzögerung der Lagenkoinzidenzsignale am Ausgang der sector boards beheben. Alternativ kann die Anfertigung und Installation eines elektronischen Bauteils in Erwägung gezogen werden, welches gezielt die gewünschten zeitlichen Verzögerungen bewirkt.

- Die Module 14, 21 und 36 sind in ähnlicher Weise betroffen. Auch hier erreicht ein Teil der Lagenkoinzidenzsignale verfrüht die GPTP-Elektronik, jedoch werden insgesamt nur ca. 84% (Modul 14), 98% (Modul 21) bzw. 93% (Modul 36) der Lagenkoinzidenzsignale in einer der Zeitscheiben überhaupt erzeugt. Der Grund hierfür bleibt unklar.

Die Gesamteffizienz

Zeitliche und Hardware-Effizienz wurden darüberhinaus multiplikativ zu einer Gesamteffizienz kombiniert. Ein Vergleich des Ergebnisses mit denen des DST-Verfahren, welches offiziell zur Ermittlung der Triggereffizienz des Zentralen Myon-Detektors eingesetzt wird, zeigt eine Reihe signifikanter Unterschiede. So fällt die Beurteilung der Module, deren zeitliche Triggereffizienz beeinträchtigt ist, beim DST-Verfahren durchweg besser aus. Der Grund hierfür liegt in der Tatsache, daß das DST-Verfahren im Gegensatz zur hier ermittelten kombinierten Gesamteffizienz vollständig auf den Signalen des Readout basiert. Der Einfluß der Nachweisverzögerung bleibt dabei unberücksichtigt.

Defekte Lagen

Im Rahmen der Klassenbibliothek steht auch eine Gruppe weitergehender Analysemethoden zur Verfügung. So können die Effizienzen der inneren Lagen der Module bestimmt werden, was zur Erkennung defekter Lagen führte. Die Ergebnisse wurden mit Hilfe weiterer Methoden der Klassenbibliothek verifiziert und lassen sich wie folgt zusammenfassen.

- Es finden sich in drei Modulen defekte Triggerlagen, wie in der Tabelle 6.1 verzeichnet ist. Dies ist von erheblicher Bedeutung für die Betrachtung der Triggereffizienz, welche auf diese Weise in den betroffenen Modulen deutlich herabgesetzt wird. Problematisch hierbei ist, daß dies nicht von den hier vorgestellten Methoden zur Bestimmung der Hardware-Effizienz erfaßt werden kann. Eine Korrektur der entsprechenden Werte ist also im Nachhinein vorzunehmen. Der Defekt der Lage 8 in Modul 33 war bereits bekannt, so daß deren Funktion an die Lage 12 übertragen

Modul	Lage
8	8
17	4
33	8
44	9

Tabelle 6.1: Tabelle der als defekt erkannten Lagen. Äußere Lagen in ggf. vorhandenen Myonboxen wurden nicht untersucht.

wurde. Ein ähnliches Vorgehen empfielt sich, sofern eine Reparatur nicht möglich ist, auch im Falle von Modul 17. Die Konfiguration von Modul 8 hingegen kann aufgrund der angewendeten Lageneffizienzbedingung (vergl. Tabelle 1.3) so nicht sinnvoll modifiziert werden.

- Die hier verwendete Methode zur Ermittlung der Lageneffizienzen kann weiter verbessert werden. Insbesondere im Hinblick auf die Einbeziehung der äußeren Lagen erscheint eine Erweiterung notwendig. Dies erfordert eine explizite Berücksichtigung der Geometrie jedes einzelnen Moduls und dessen individueller Ausstattung mit Myonboxen. Auf diesem Wege kann auch eine höhere Genauigkeit der Effizienzbeurteilung für die Lagen 3 und 12 erreicht werden.

Verrauschte Drähte

Mit weiteren Methoden der Klassenbibliothek konnten verrauschte Drähte detektiert und einwandfrei lokalisiert werden. Zu Beginn der winterlichen Betriebspause im September 2000 zeigte noch ein einzelner verbliebener Draht einen solchen Defekt (Modul 62, Lage 11, Element 4, darin Draht 14).

Die Zahl getroffener Drähte pro Ereignis

Bei der Betrachtung der Zahl getroffener Drähte pro Ereignis offenbart sich ein Problem mit Modul 29, welches überdurchschnittlich häufig eine hohe Zahl von Treffern aufweist. Sowohl die zeitliche Triggereffizienz als auch die Hardware-Effizienz dieses Moduls ist nahezu einwandfrei. Auch konnten keinerlei weitere Hinweise auf einen Defekt in Modul 29 bzw. der zugehörigen Elektronik festgestellt werden. Der Grund für das auffällige Verhalten bleibt demnach unklar.

Anhang A

Konvertierung der Rohdaten in das ROOT-Format

A.1 Der Aufruf von `bos2oop`

Die Namen der BOS-Bänke, deren entsprechende Datenklassen erzeugt werden sollen, sind zunächst in eine ansonsten leere Konfigurationsdatei einzutragen, wobei folgendes Beispiel das Schema verdeutlicht[1].

```
BANKS
  MUONS (
    HEAD IRWE IRSE TGPP[2048] TSCD[8] TL23[8] TEL1[0]
    TLV1 YTL4
  )
END
```

Das Schlüsselwort `BANKS` ist darin zwingend notwendig. `MUONS` stellt den sog. *wrapper* dar, der `bos2oop` veranlassen wird, die einhüllende Klasse namens `TMUONS` zu erzeugen. Diese dient später als übergeordnete Datenstruktur und enthält die entsprechenden Variablen und Routinen für den bequemen Zugriff auf die Daten der eingebundenen Klassen, und zwar jeweils für ein einzelnes Ereignis. Es können auch BOS-Bänke außerhalb des wrappers angegeben werden, bzw. kann der wrapper auch ganz ausgelassen werden. In diesem Falle werden die daraus entstehenden Klassen nicht durch das auf diesem Wege kompilierte Konvertierungsprogramm `bos2root` berücksichtigt (s. nächster Abschnitt), d.h. die Daten der zugehörigen BOS-Bänke werden sich nicht in den von `bos2root` erzeugten ROOT-Dateien wiederfinden, ohne daß dies manuell nachgetragen wird. Zusammengefaßt besitzt der Aufruf von `bos2oop` die folgende Syntax:

```
bos2oop [Optionen] [-d <DDL-Datei>] <Konfigurationsdatei>
```

Ein solcher Aufruf resultiert in der Erzeugung eines Verzeichnisses `MUONS`, der Datenklassen und der wrapper-Klasse TMUONS. Auch werden die bereits vorhandenen speziellen *makefiles* entsprechend angepaßt. In diesem Verzeichnis kann nun durch Aufruf

[1] Die hier aufgeführten Bänke wurden bei der Analyse normaler Runperioden ausgewertet.

Rohdaten und Verwendungszweck	BOS-Bänke	Rekonstruktions-programme
kosmische und Halo-Myonen (Lageneffizienzen)	HEAD, IRWE, ITKR, ITWX, ITPR	ITREC
kosmische Myonen (zeitliche Triggereffizienzen)	HEAD, IRWE, TGPP(2048), TSCD(8), TEL1(0), CJCC, CJKV, CJGR, RESL, ISCR	CJCREC
Halo-Myonen (zeitliche Triggereffizienzen)	HEAD, IRWE, TGPP(2048), MTCT, TSCD(8), TEL1(0), RESL, ISCR	MTREC
normale Runperioden (sonstige)	HEAD, IRWE, IRSE, TGPP(2048), YTL4, TSCD(8), TL23(8), TEL1(0)	-

Tabelle A.1: Die für die verschiedenen Analysezwecke benötigten BOS-Bänke und aufzurufenden Rekonstruktionsprogramme.

des *make*-Befehls die shared library `libbos2root.so` erzeugt werden, anschließend im übergeordneten Verzeichnis das Konvertierungsprogramm `bos2root`.

A.2 Die Konvertierung mit `bos2root`

`bos2root` kann mit der folgenden Syntax aufgerufen werden:

```
bos2root [<Dateiname>] <  <Steuerdatei>
```

`<Dateiname>` bezeichnet den Namen der zu erzeugenden ROOT-Datei; wird dieser Parameter nicht angegeben, erzeugt `bos2root` eine Datei namens `Event.root`. Um später von ROOT gelesen werden zu können, muß `Dateiname` generell die Endung „`.root`" besitzen. Auch ist darauf zu achten, daß die shared library `libbos2root.so` sich innerhalb der Reichweite des Konvertierungsprogrammes befindet, d.h. die Umgebungsvariable `LD_LIBRARY_PATH` entsprechend gewählt ist. Selbstverständlich kann `bos2root` nur erfolgreich auf Rechnern benutzt werden, die Zugriff auf die Verzeichnisse `/acs` des DESY besitzen, die Magnetbänder von dort aus also erreichbar sind[2]. Als Steuerdatei ist eine standardmäßig für die Auslese der Magnetbänder gebrauchte FPack-Steuerdatei zu

[2]Dies sind z.Zt. der *dice-cluster*, der *desy-cluster* sowie die „*PC-farm*".

verwenden. In dieser muß das auszulesende Magnetband, die Zahl der zu konvertierenden Ereignisse sowie ggf. eine Reihe von Parametern für die aufgerufenen Rekonstruktionsprogramme angegeben bzw. angepaßt werden. Welche Rekonstruktionsprogramme notwendig sind, kann der Tabelle A.1 entnommen werden.

Weiterhin muß für die Konvertierung der Daten kosmischer und Halo-Myonen zum Zwecke der Bestimmung zeitlicher Triggereffizienzen eine speziell überarbeitete Version von `bos2root` verwendet werden. Um den Umfang der erzeugten ROOT-Dateien in diesen Fällen auf ein vertretbares Maß zu verringern, sind erste Schritte der Auswertung in das Konvertierungsprogramm integriert. Zu diesem Zweck ist die Datenklasse `TResl` erzeugt worden, welche die Ergebnisse dieser Berechnungen aufnimmt. Für eine detaillierte Beschreibung sei auf [22] verwiesen.

Die Größe der erzeugten ROOT-Dateien variiert stark mit der Zahl der ausgelesenen BOS-Bänke und in geringerem Umfang auch mit den verschiedenen Magnetbändern selbst. Für die Konvertierung von Daten normaler Runperioden in der oben angegebenen Konfiguration sowie denjenigen kosmischer Myonen sind erfahrungsgemäß etwa 10 bis 20 Megabyte pro Band einzukalkulieren, für die Daten von Halo-Myonen bis hin zu 100 Megabyte und mehr.

A.3 Die automatisierte Konvertierung

Es besteht die Möglichkeit, die Konvertierung der Rohdaten im Verlauf der Datennahmeperioden weitgehend zu automatisieren. Die UNIX-Betriebssysteme der H1-Rechner bieten hierzu das Programm `cron`, welches regelmäßig zu beliebigen, fest vorzugebenden Zeitpunkten das Konvertierungsprogramm aufrufen kann. Die Rohdaten normaler Runperioden, die in der vorliegenden Analyse Verwendung fanden, sind zu einem Großteil auf diese Weise in das ROOT-Format übertragen worden.

Die Grundlage der automatisierten Konvertierung bilden eine Reihe von *shell-script*-Programmen unter Verwendung der Befehle `awk` und `sed`. Die folgenden Arbeitsschritte werden dabei ausgeführt.

- Das Erkennen neu gewonnener Rohdatenbänder durch Vergleich deren Entstehungszeitpunktes mit dem Datum und der Uhrzeit des vorangegangenen Konvertierungsvorgangs
- Die Erzeugung einer entsprechenden Dateiliste
- Die Erstellung der FPack-Steuerdateien, welche einzeln für jedes Band vorgenommen wird
- Das Anfügen des Befehlsaufrufes für `bos2root` mit der zugehörigen Steuerdatei in die gewünschte Warteschlange (*queue*) zur Bearbeitung
- Abschließend das Kopieren der ROOT-Dateien auf eine lokale Festplatte ausreichender Kapazität

Der Quellcode der shell-script-Programme und weitere benötigte Dateien sind in [22] einzusehen.

Im vorliegenden Fall wurde das Konvertierungsprogramm mittels `cron` zu fünf verschiedenen Zeitpunkten am Tage aufgerufen, um die Belastung der Rechner nicht auf bestimmte Uhrzeiten zu konzentrieren. Auch wurden nicht alle Magnetbänder ausgelesen. Zum Zwecke der permanenten Überwachung der Funktion des Zentralen Myon-Detektors ist es ausreichend, jeweils ein Viertel der anfallenden Datenmenge normaler Runperioden zur Auswertung zu bearbeiten. Die von Anfang April 2000 bis Anfang September 2000 insgesamt konvertierte Datenmenge beläuft sich auf ca. 44 Gigabyte.

Anhang B

Konstruktionspläne

In diesem Kapitel des Anhangs sind vier Konstruktionspläne abgedruckt, welche die Ausstattung der Module mit Myonboxen dokumentieren [6]. Bei den Modulen der Endkappen sind, sofern vorhanden, lediglich die äußeren Myonboxen (Lagen 13 bis 15) dargestellt.

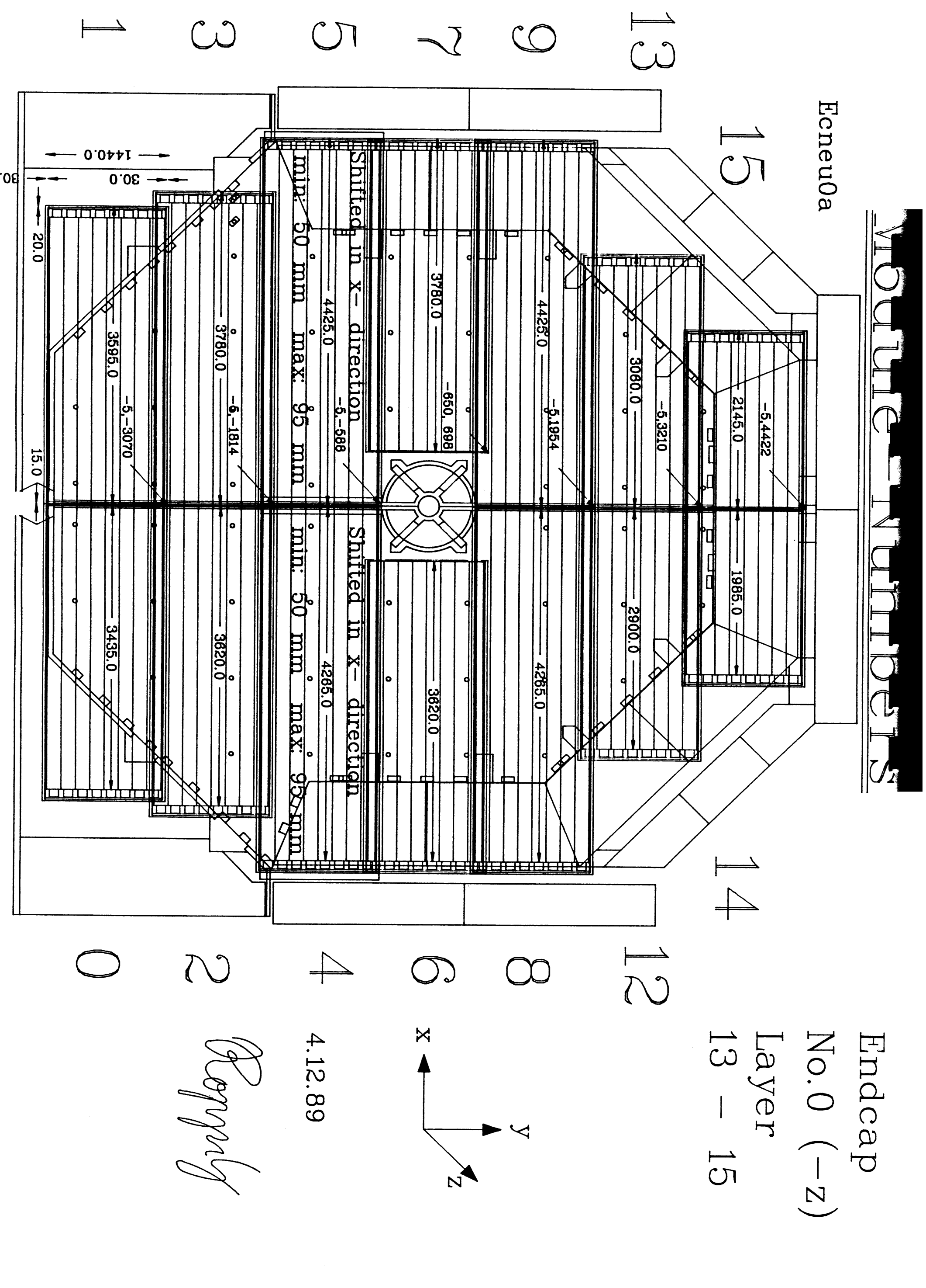

Abbildung B.1: Die Ausstattung der rückwärtigen Endkappe (BEC) mit äußeren Myonboxen (Lagen 13 bis 15) [7].

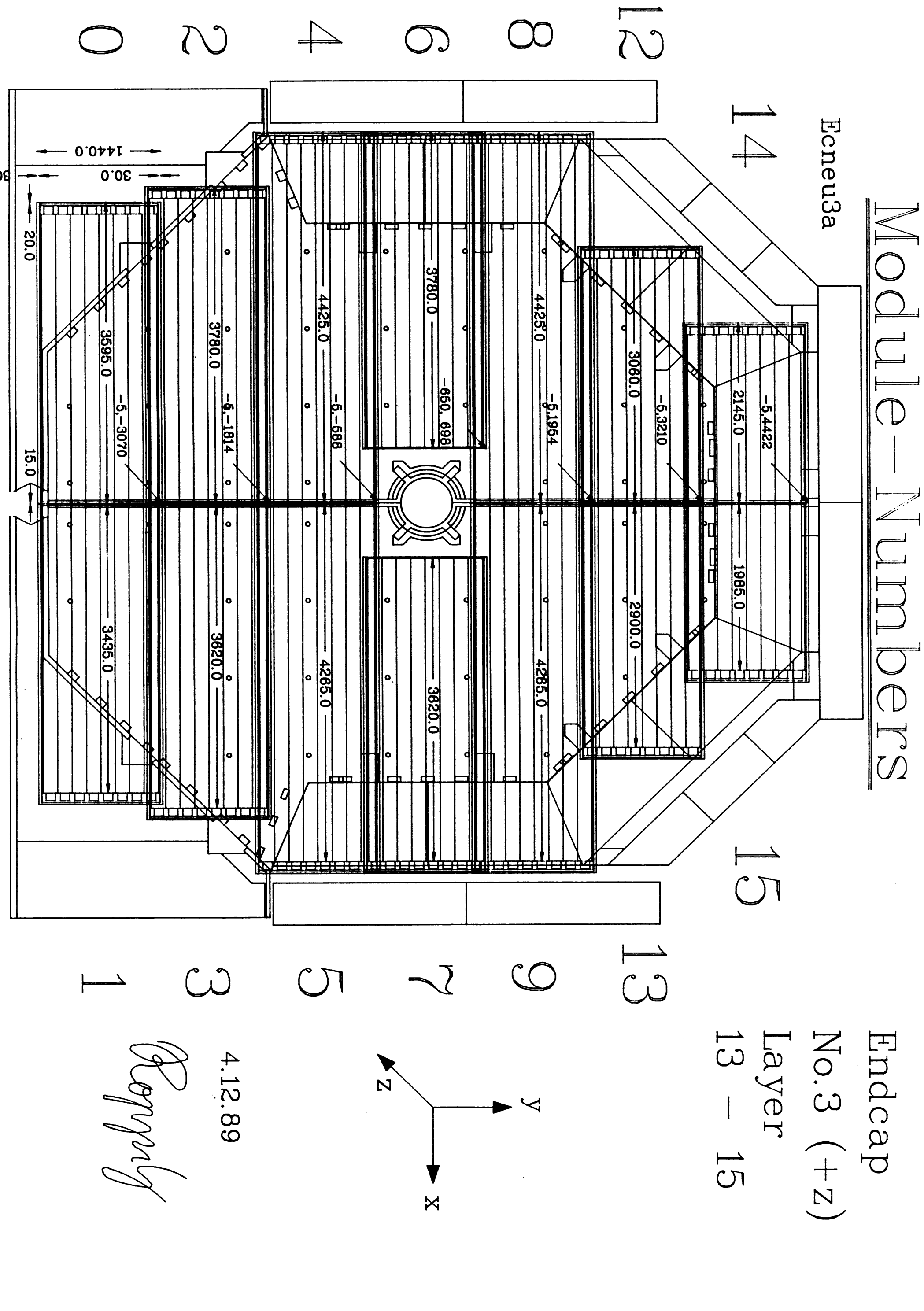

Abbildung B.2: Die Ausstattung der vorwärtigen Endkappe (FEC) mit äußeren Myonboxen (Lagen 13 bis 15) [7].

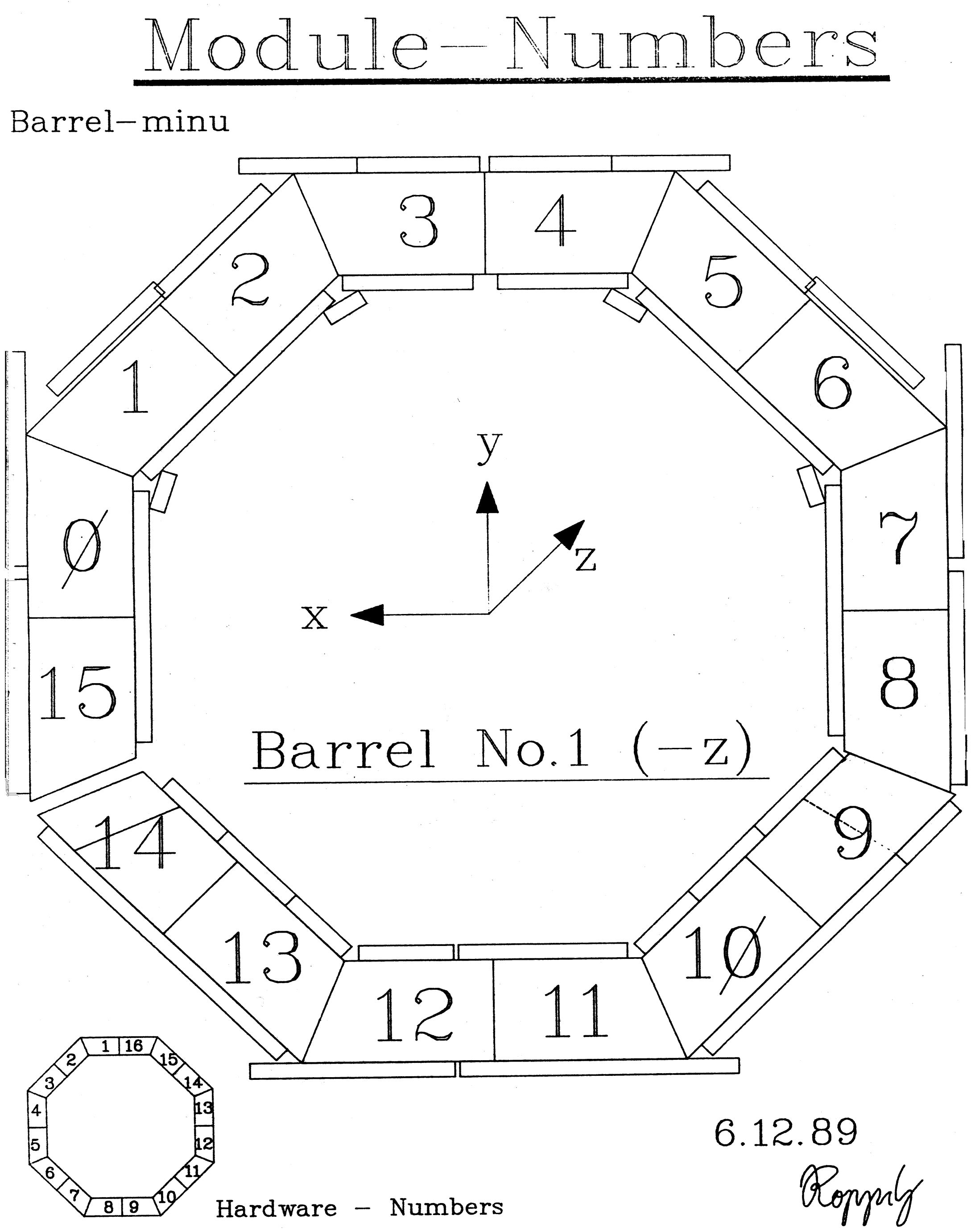

Abbildung B.3: Die Ausstattung des rückwärtigen Barrelbereiches (BBA) mit Myonboxen (Lagen 0 bis 2 und 13 bis 15) [8].

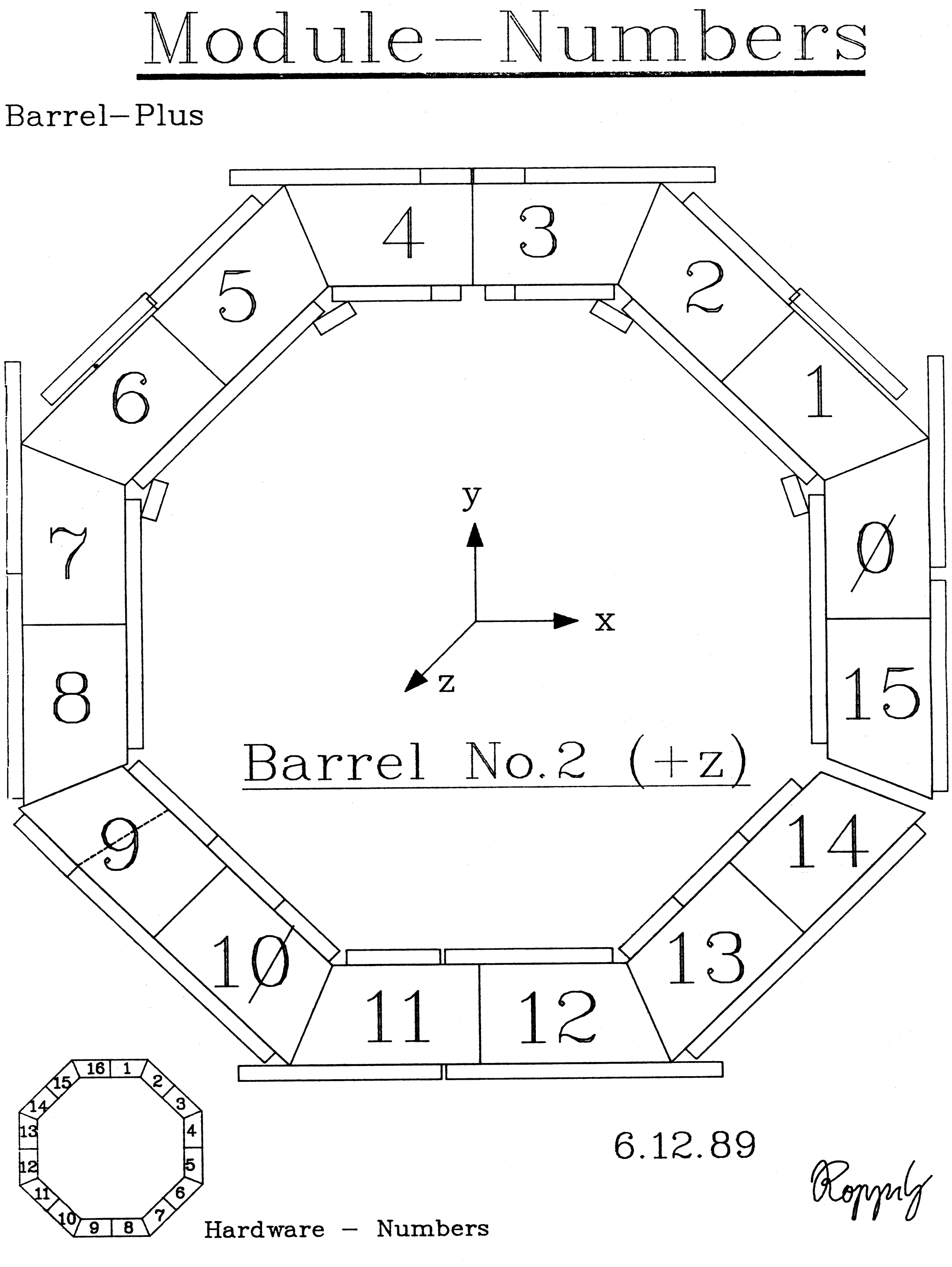

Abbildung B.4: Die Ausstattung des vorwärtigen Barrelbereiches (FBA) mit Myonboxen (Lagen 0 bis 2 und 13 bis 15) [8].

Anhang C

Weitere Histogramme

In diesem Kapitel des Anhangs sind diejenigen Histogramme abgedruckt, die im Text aus Platzgründen nicht untergebracht werden konnten. Soweit nicht anders angegeben, basieren alle Ergebnisse auf dem Rohdatenmaterial von April 2000 bis Ende August 2000.

C.1 Die zeitlichen Triggereffizienzen der Module

Auf den folgenden Seiten finden sich die Histogramme der Wahrscheinlichkeitsverteilungen $p(t_{0,rel})$ mit den angepaßten Funktionen $\hat{\varepsilon}_t(t_{0,rel})$, wie sie von der ausführenden Klasse `TMuSpecialLoop` im Verzeichnis `/SpecialHistograms/Modules` der Ausgabedatei erzeugt werden. Grundlage dieser Histogramme sind 1.776.700 bzw. 1.212.470 Ereignisse kosmischer bzw. von Halo-Myonen des Jahres 2000.

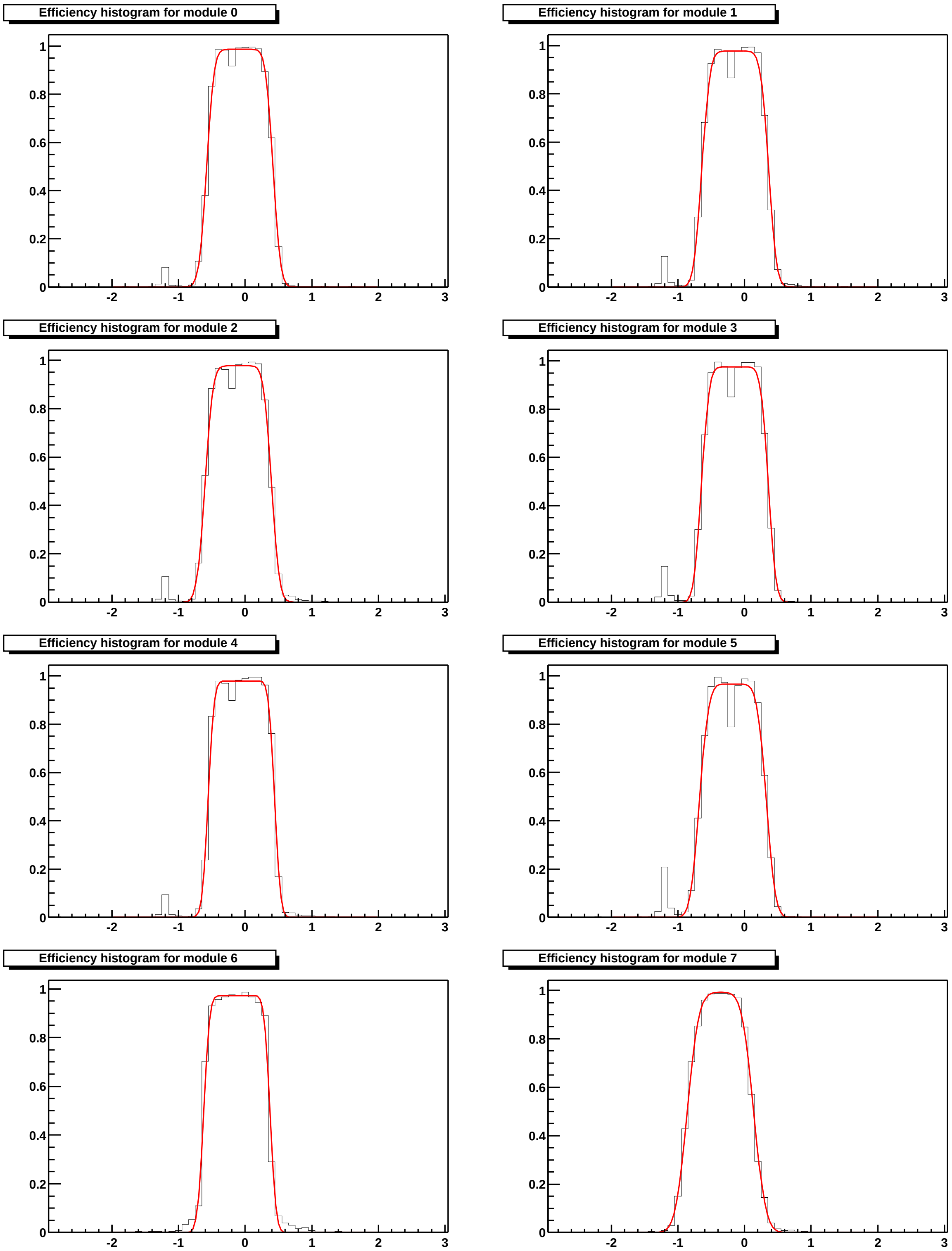

Abbildung C.1: Die Histogramme `SingleEfficiency_<Modulnummer>` für die Module 0 bis 7.

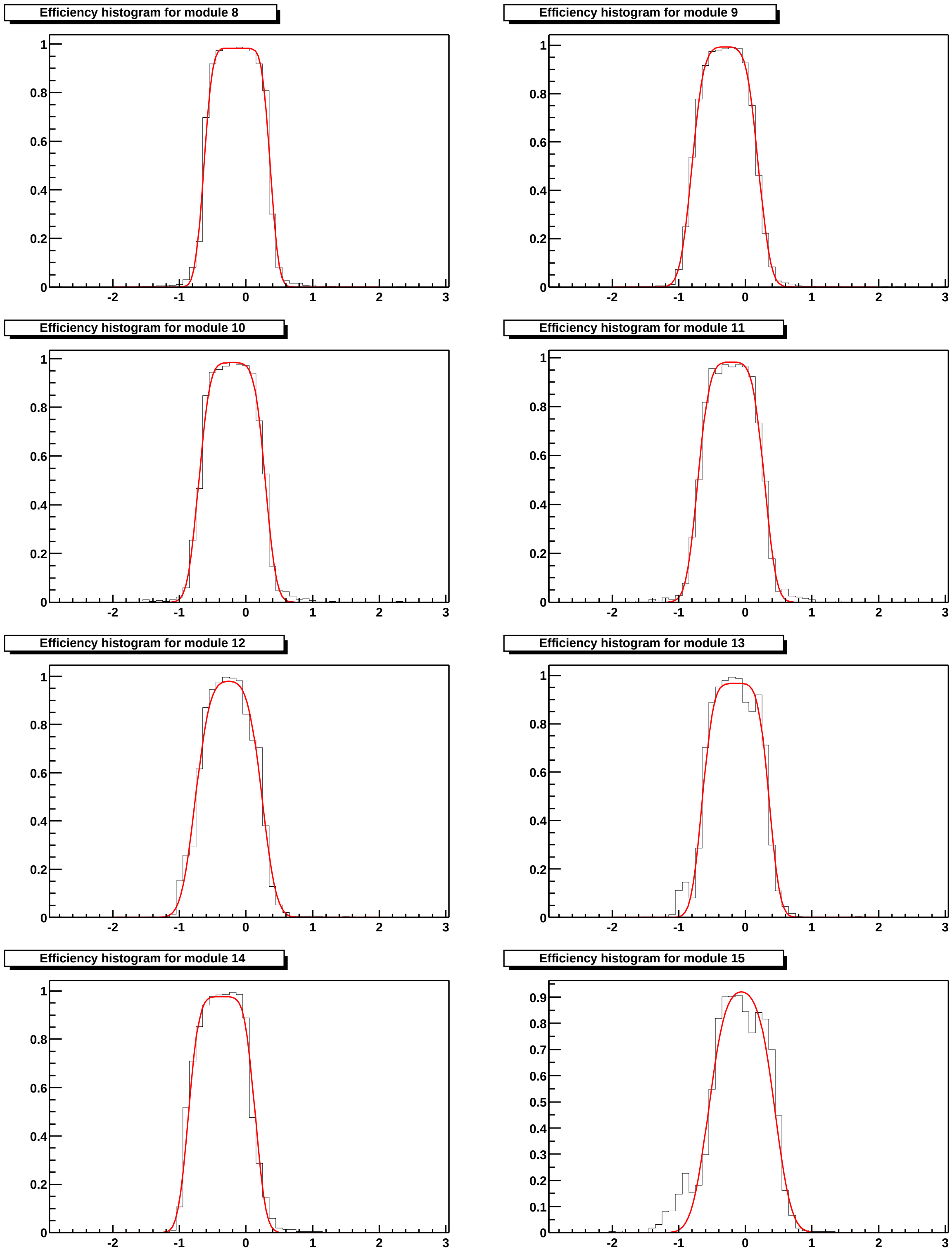

Abbildung C.2: Die Histogramme `SingleEfficiency_<Modulnummer>` für die Module 8 bis 15.

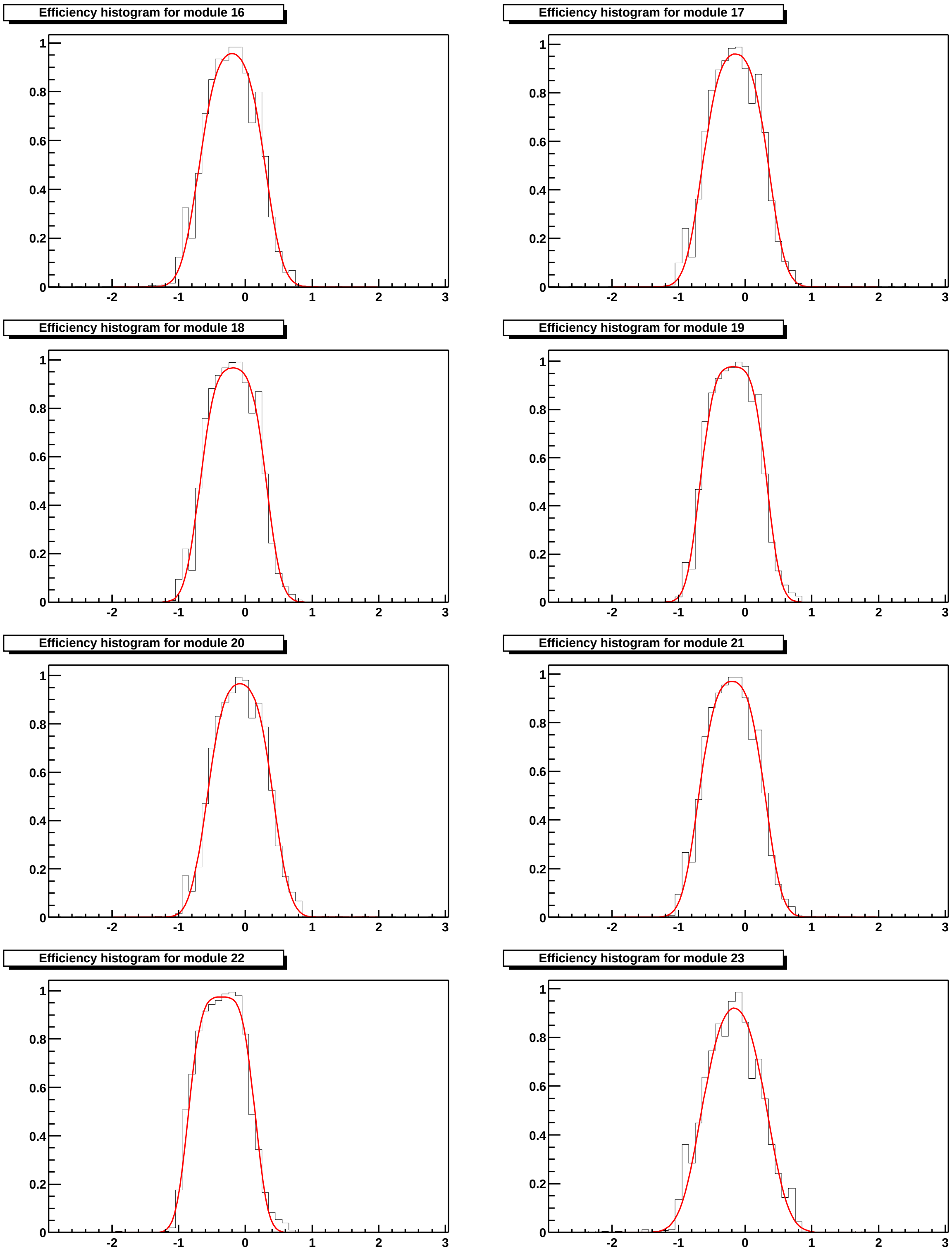

Abbildung C.3: Die Histogramme `SingleEfficiency_<Modulnummer>` für die Module 16 bis 23.

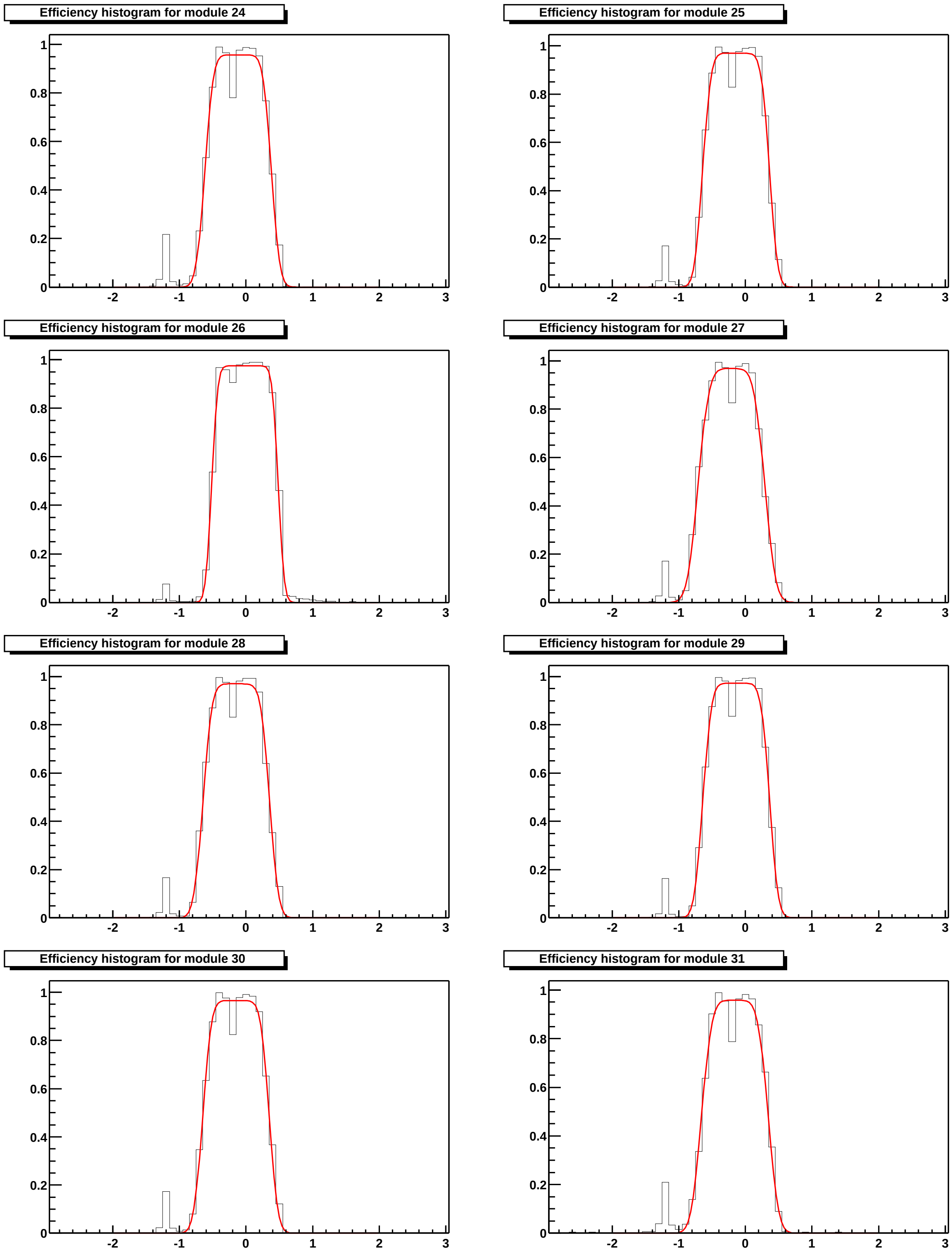

Abbildung C.4: Die Histogramme `SingleEfficiency_<Modulnummer>` für die Module 24 bis 31.

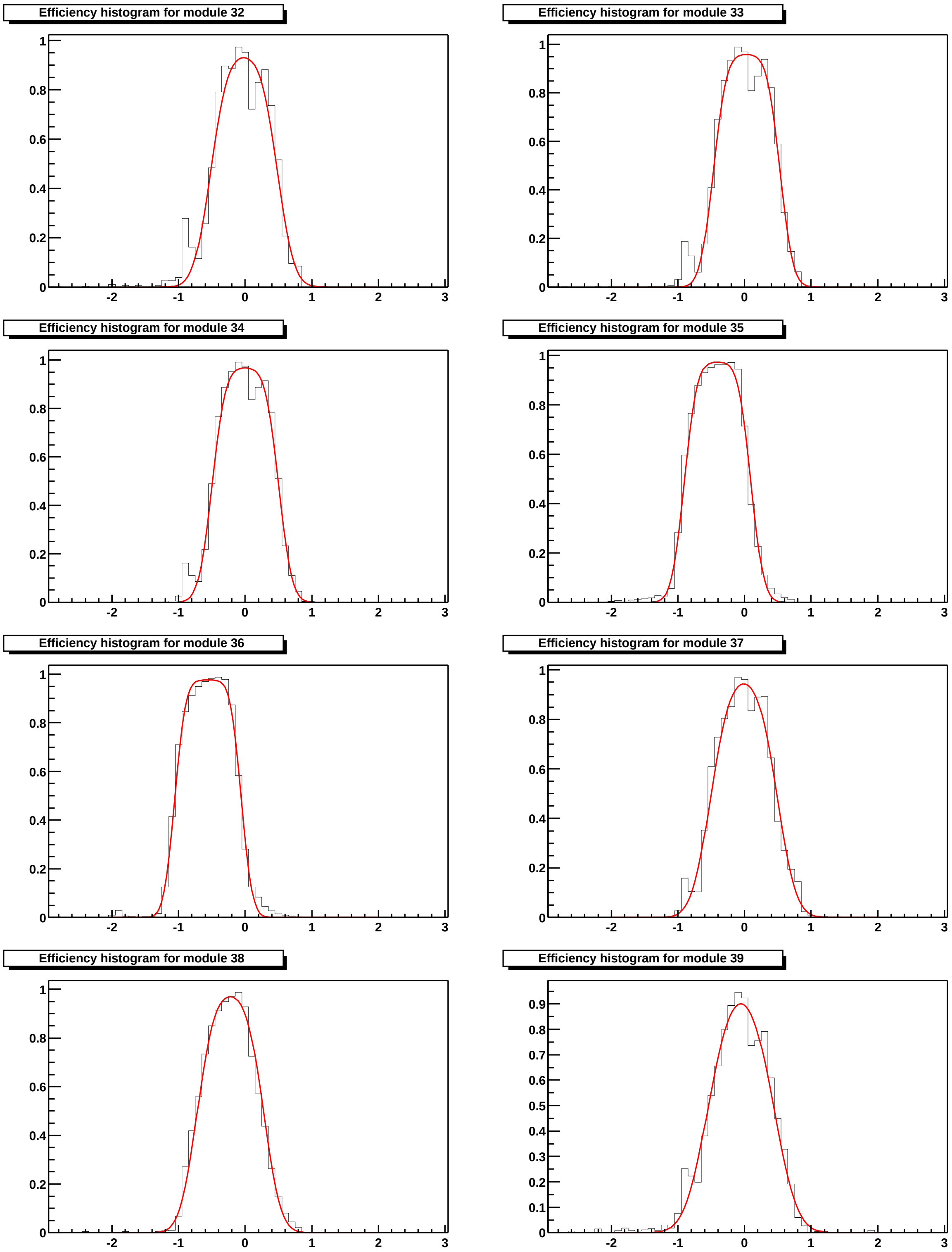

Abbildung C.5: Die Histogramme `SingleEfficiency_<Modulnummer>` für die Module 32 bis 39.

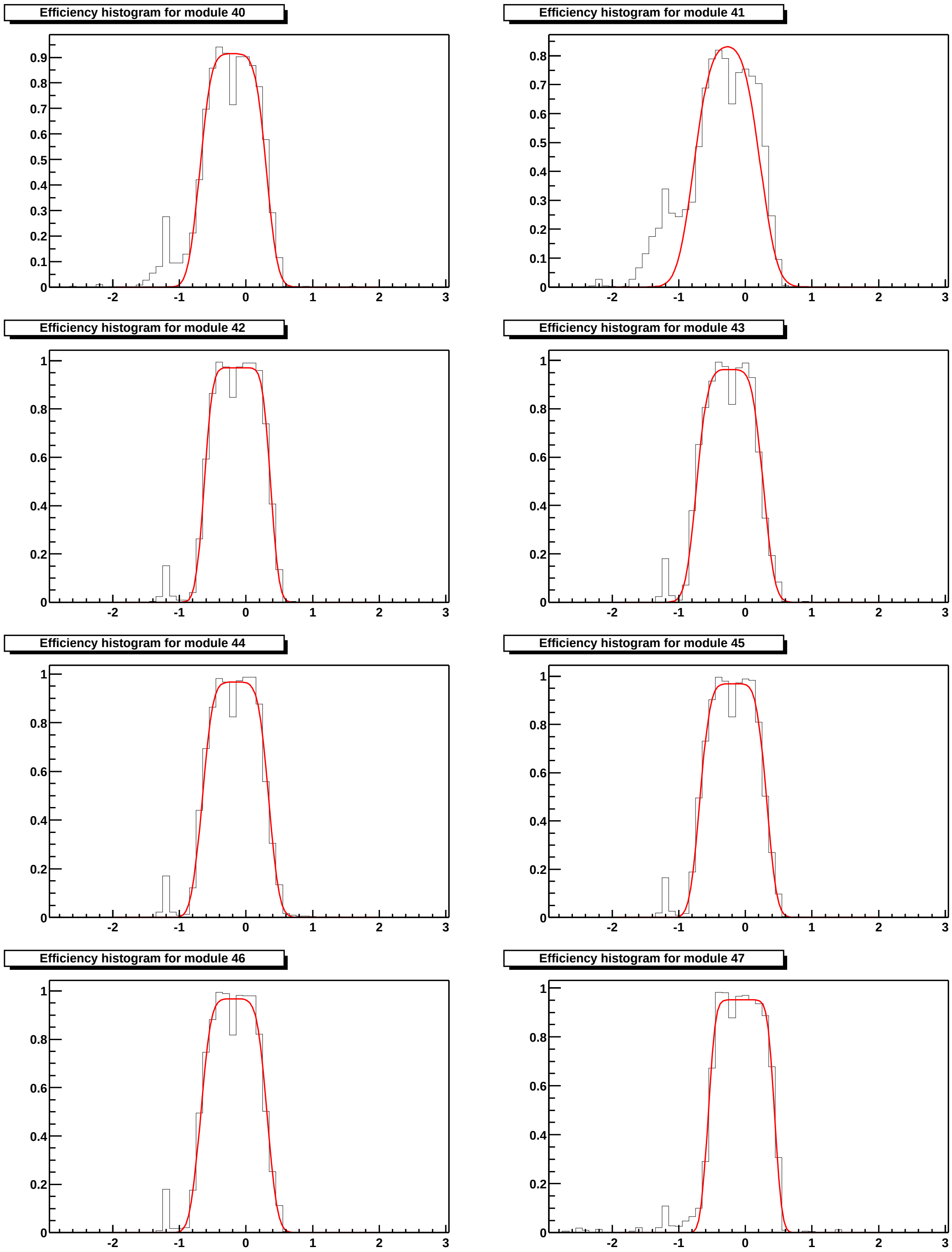

Abbildung C.6: Die Histogramme `SingleEfficiency_<Modulnummer>` für die Module 40 bis 47.

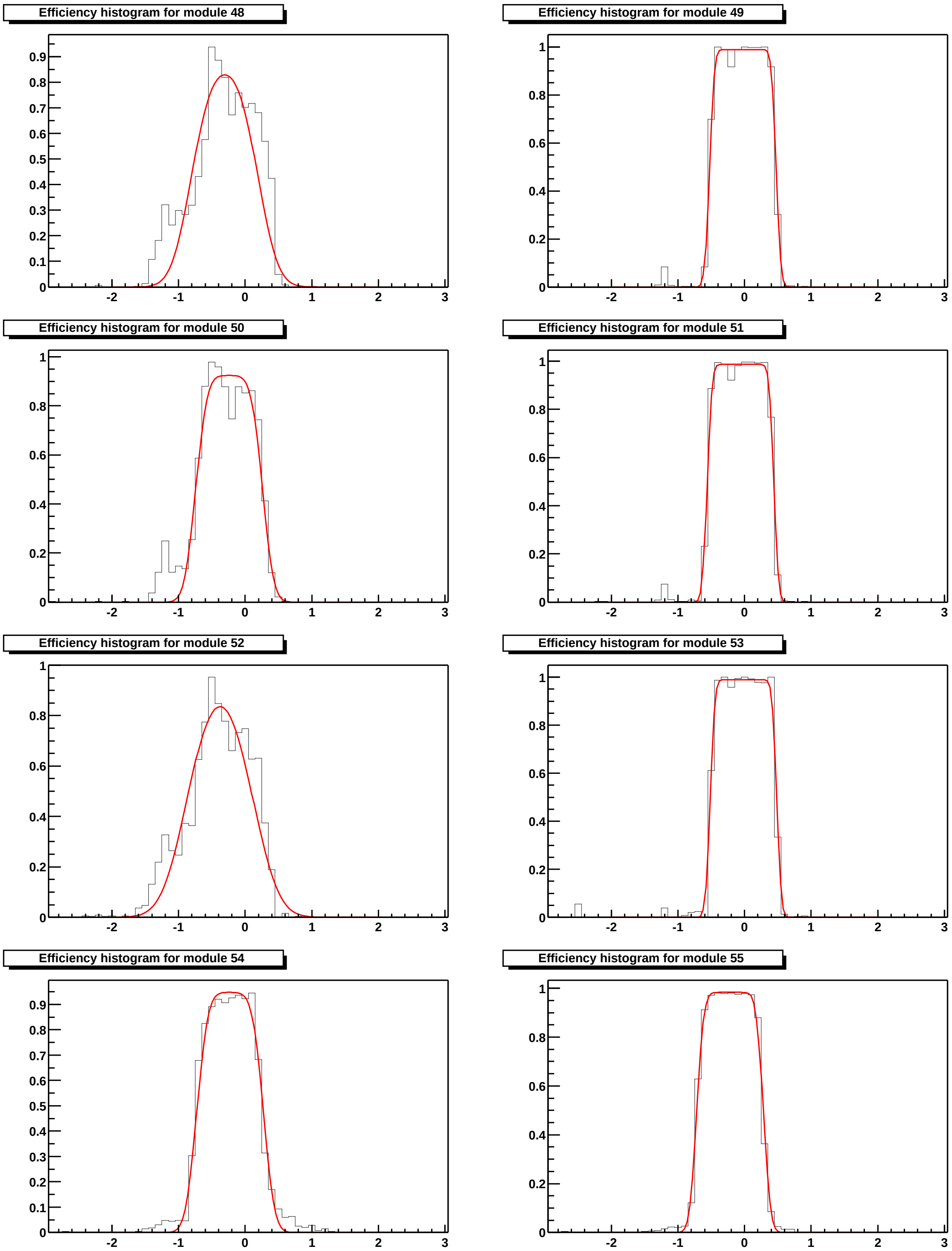

Abbildung C.7: Die Histogramme SingleEfficiency_<Modulnummer> für die Module 48 bis 55.

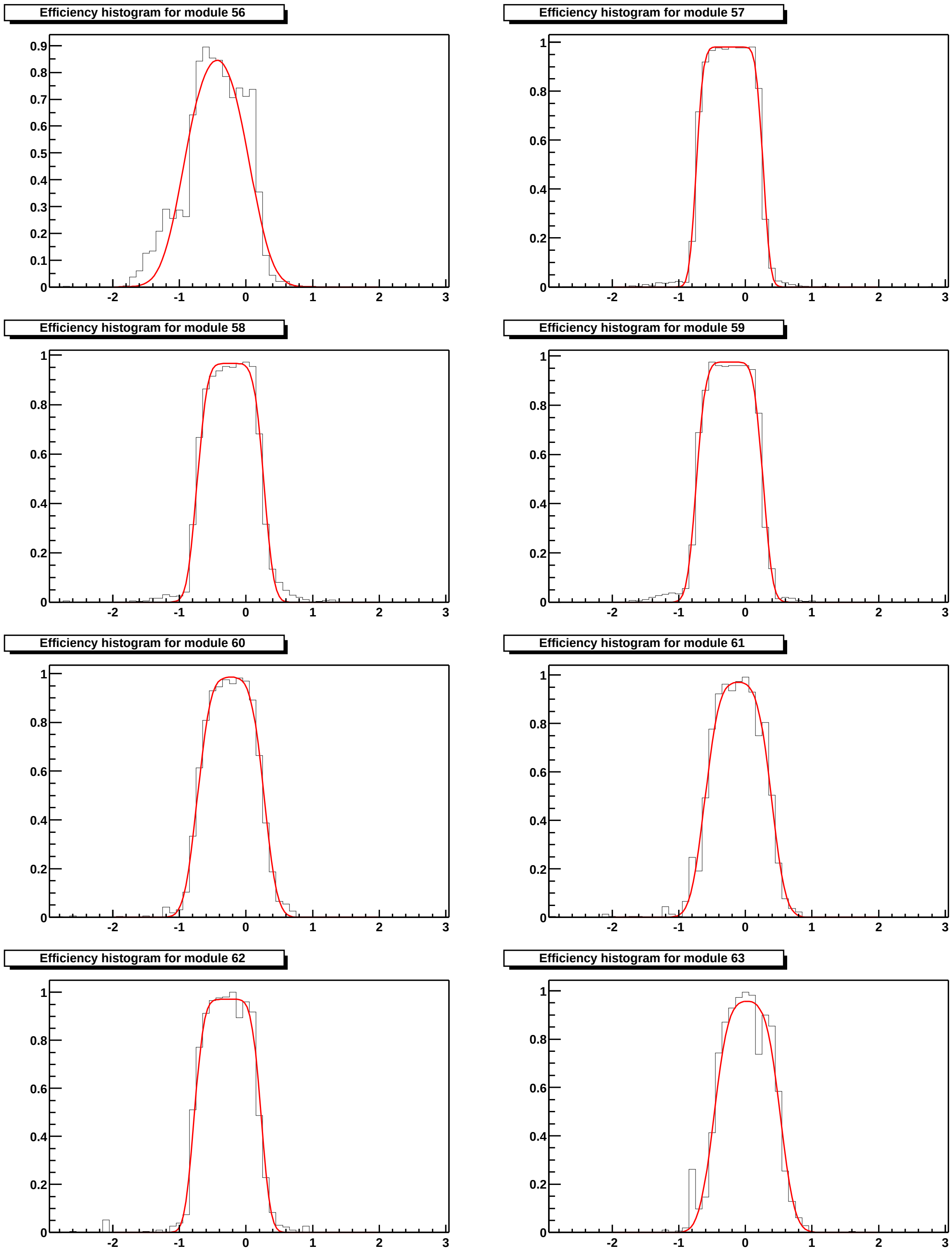

Abbildung C.8: Die Histogramme SingleEfficiency_<Modulnummer> für die Module 56 bis 63.

C.2 Zahl der getroffenen Triggerlagen während effizienter Ereignisse

Die vierundsechzig Histogramme `LayerIfTrigger_<Modulnummer>` der ausführenden Klasse `TMuLayerLoop` sind im folgenden abgedruckt. Sie tragen jeweils einen Eintrag für eine der fünf Triggerlagen.

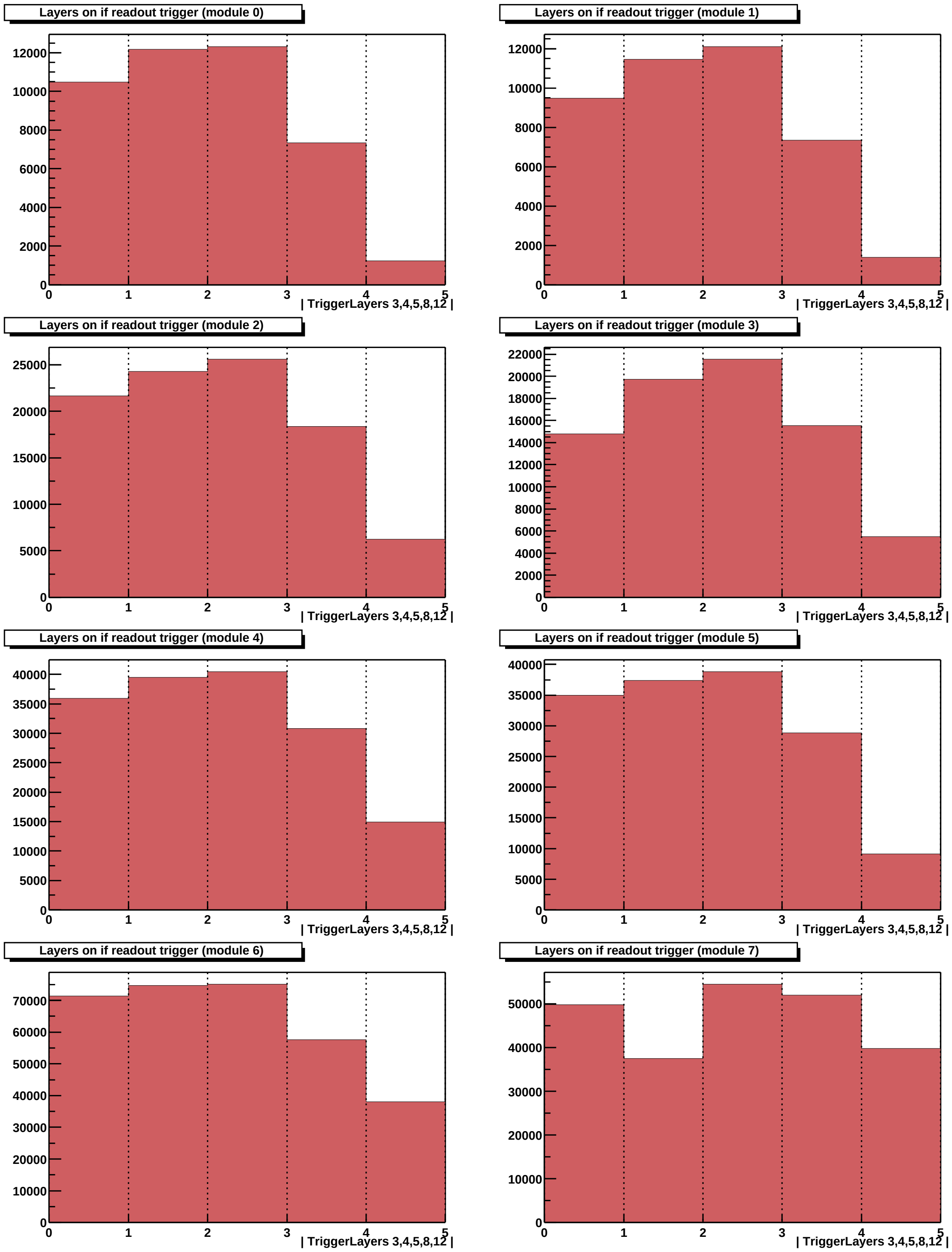

Abbildung C.9: Die Histogramme `LayerIfTrigger_<Modulnummer>` für die Module 0 bis 7.

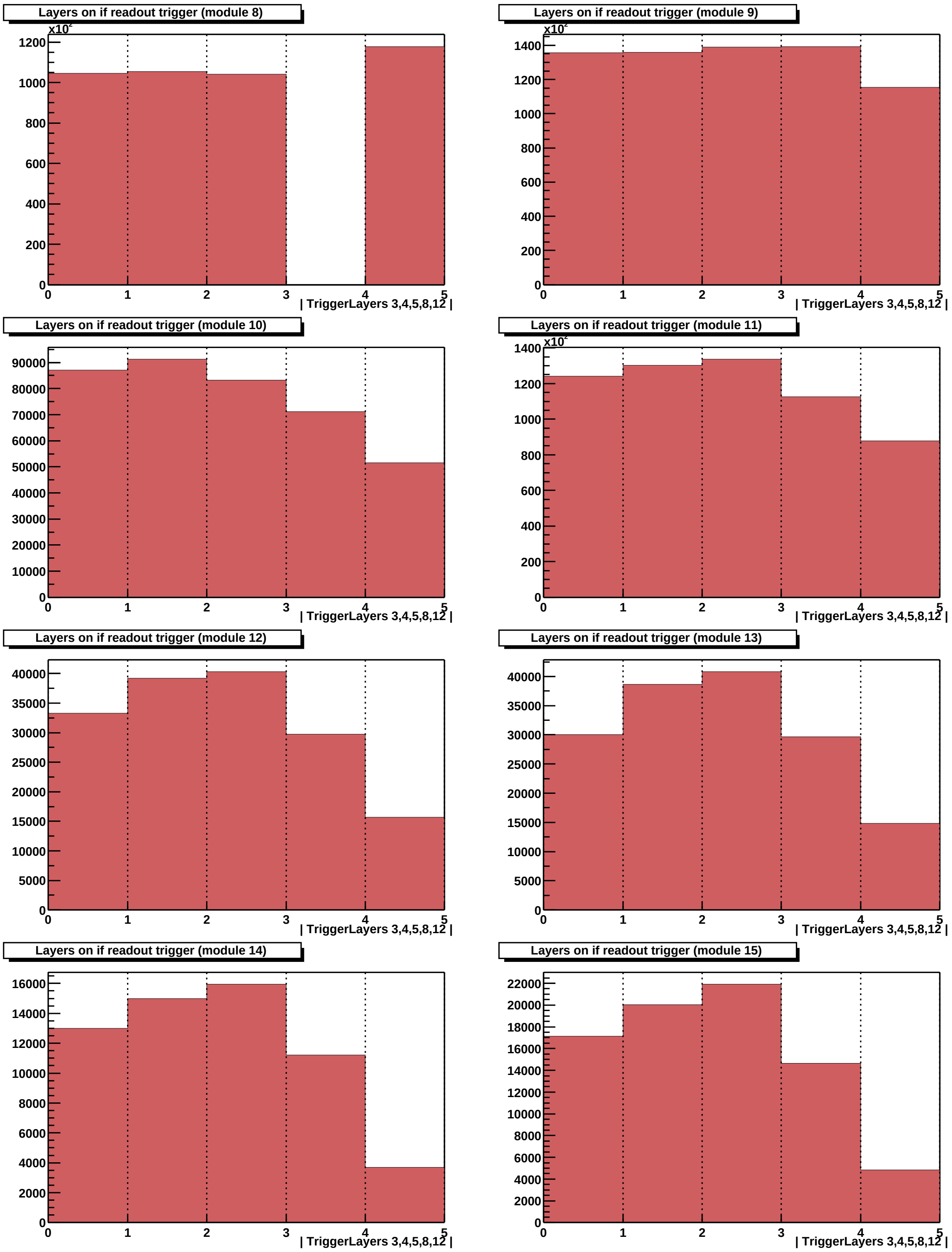

Abbildung C.10: Die Histogramme `LayerIfTrigger_<Modulnummer>` für die Module 8 bis 15.

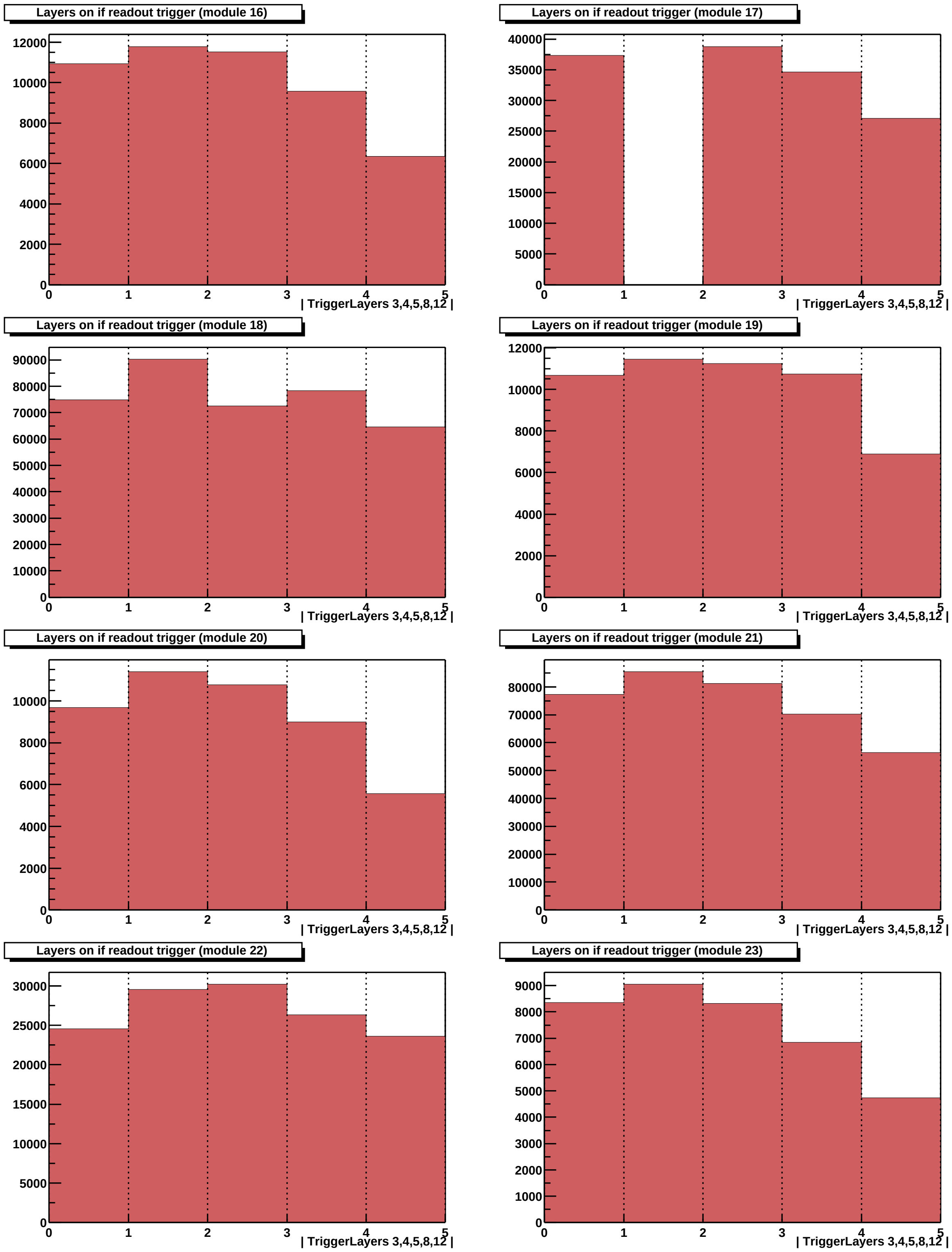

Abbildung C.11: Die Histogramme `LayerIfTrigger_<Modulnummer>` für die Module 16 bis 23.

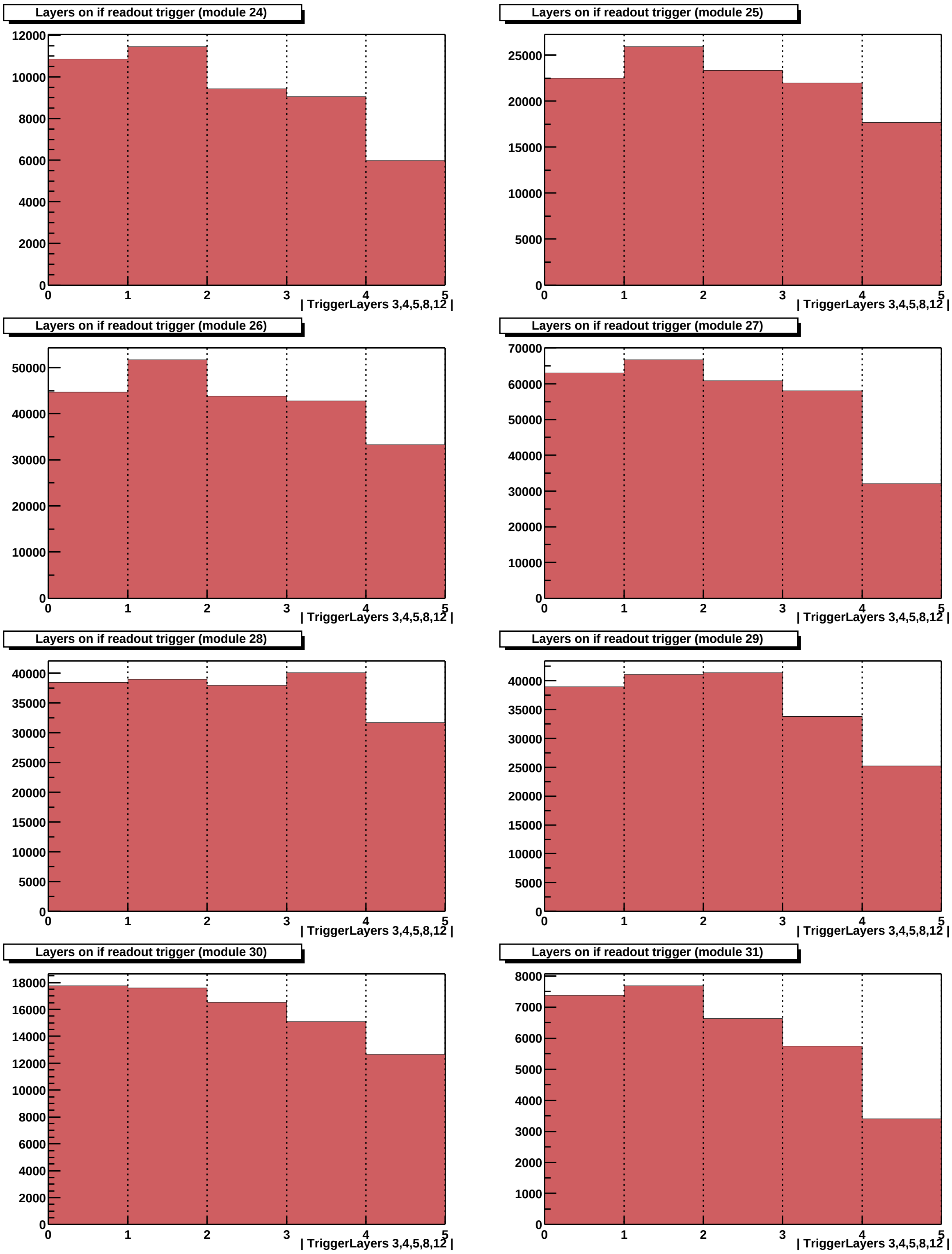

Abbildung C.12: Die Histogramme `LayerIfTrigger_<Modulnummer>` für die Module 24 bis 31.

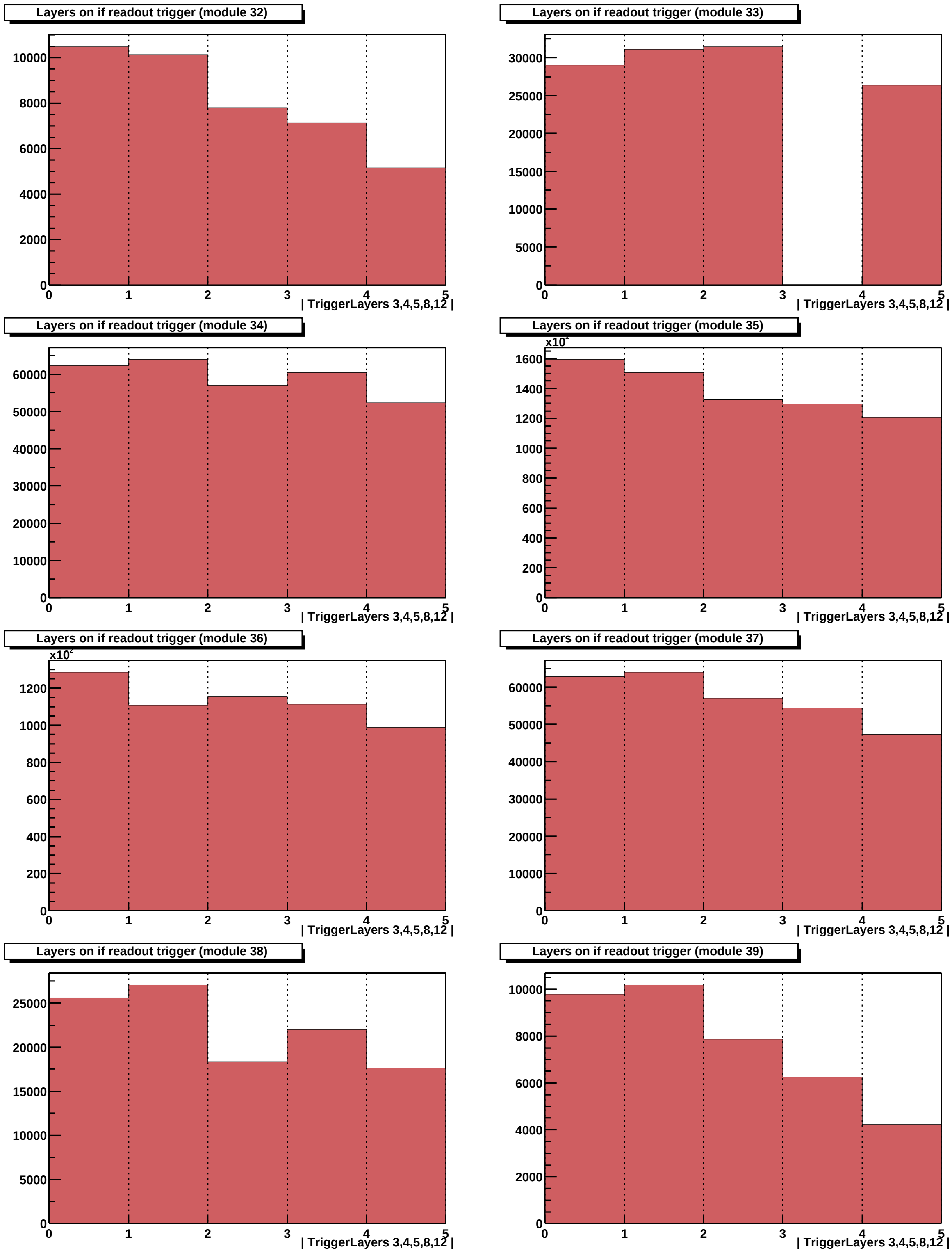

Abbildung C.13: Die Histogramme `LayerIfTrigger_<Modulnummer>` für die Module 32 bis 39.

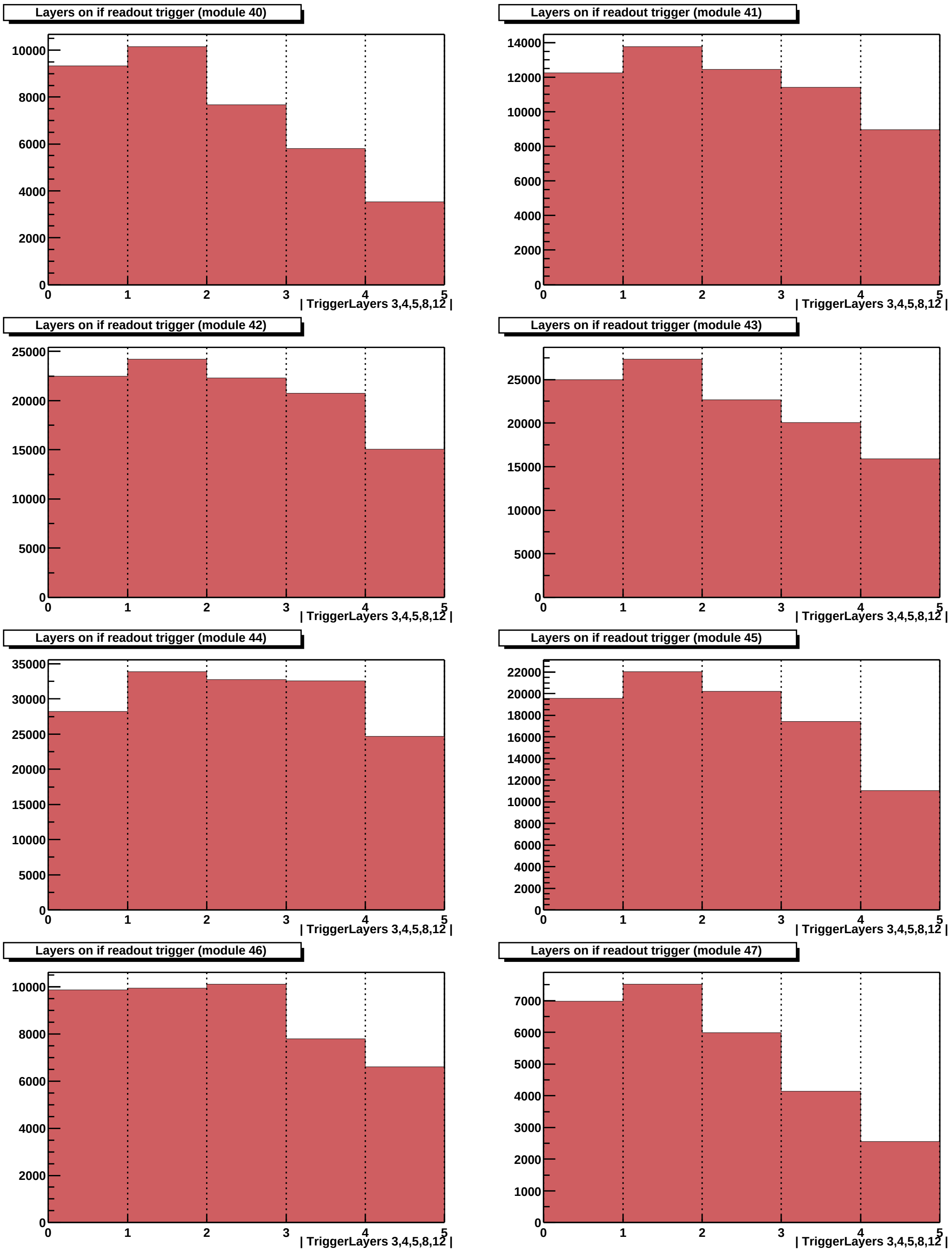

Abbildung C.14: Die Histogramme `LayerIfTrigger_<Modulnummer>` für die Module 40 bis 47.

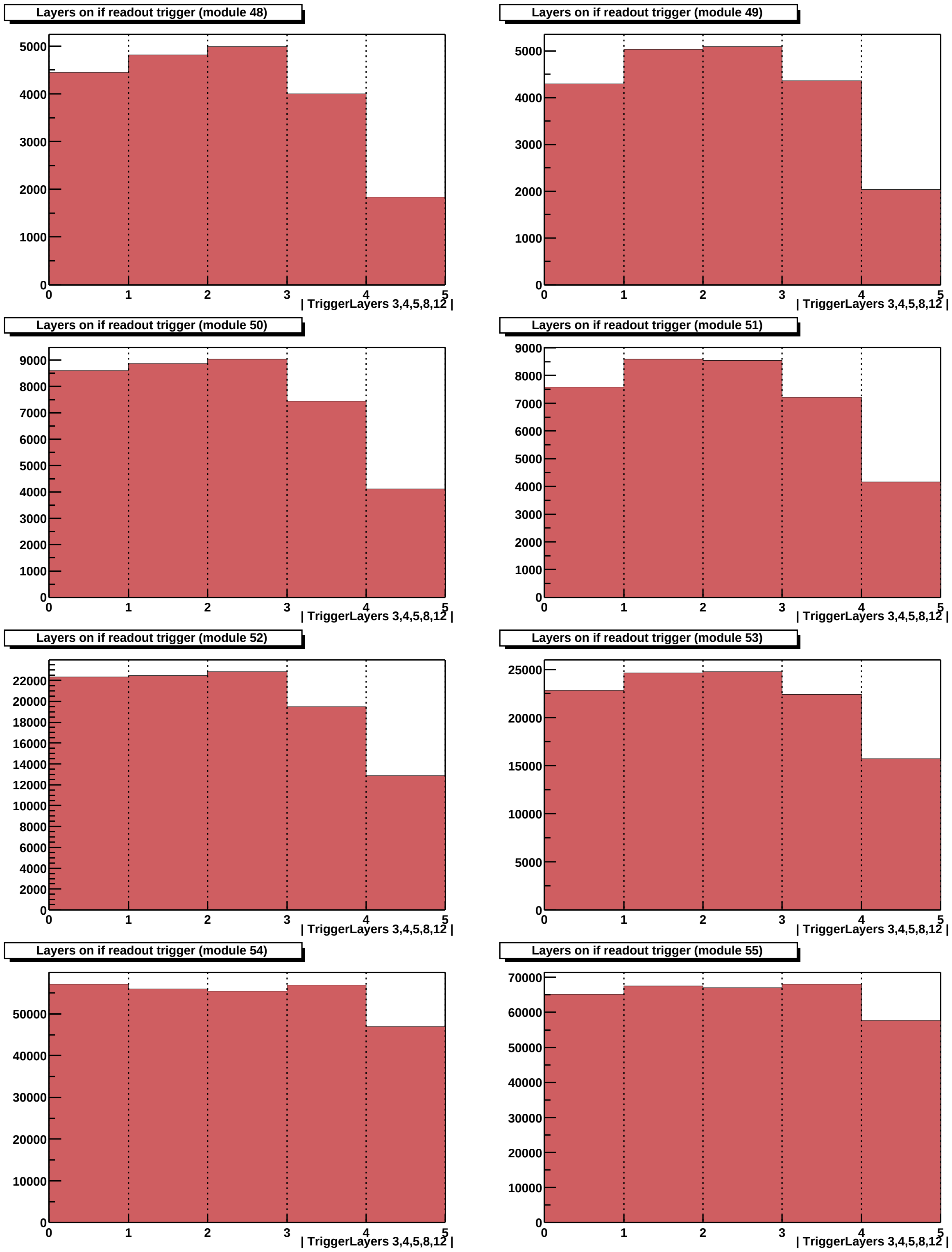

Abbildung C.15: Die Histogramme `LayerIfTrigger_<Modulnummer>` für die Module 48 bis 55.

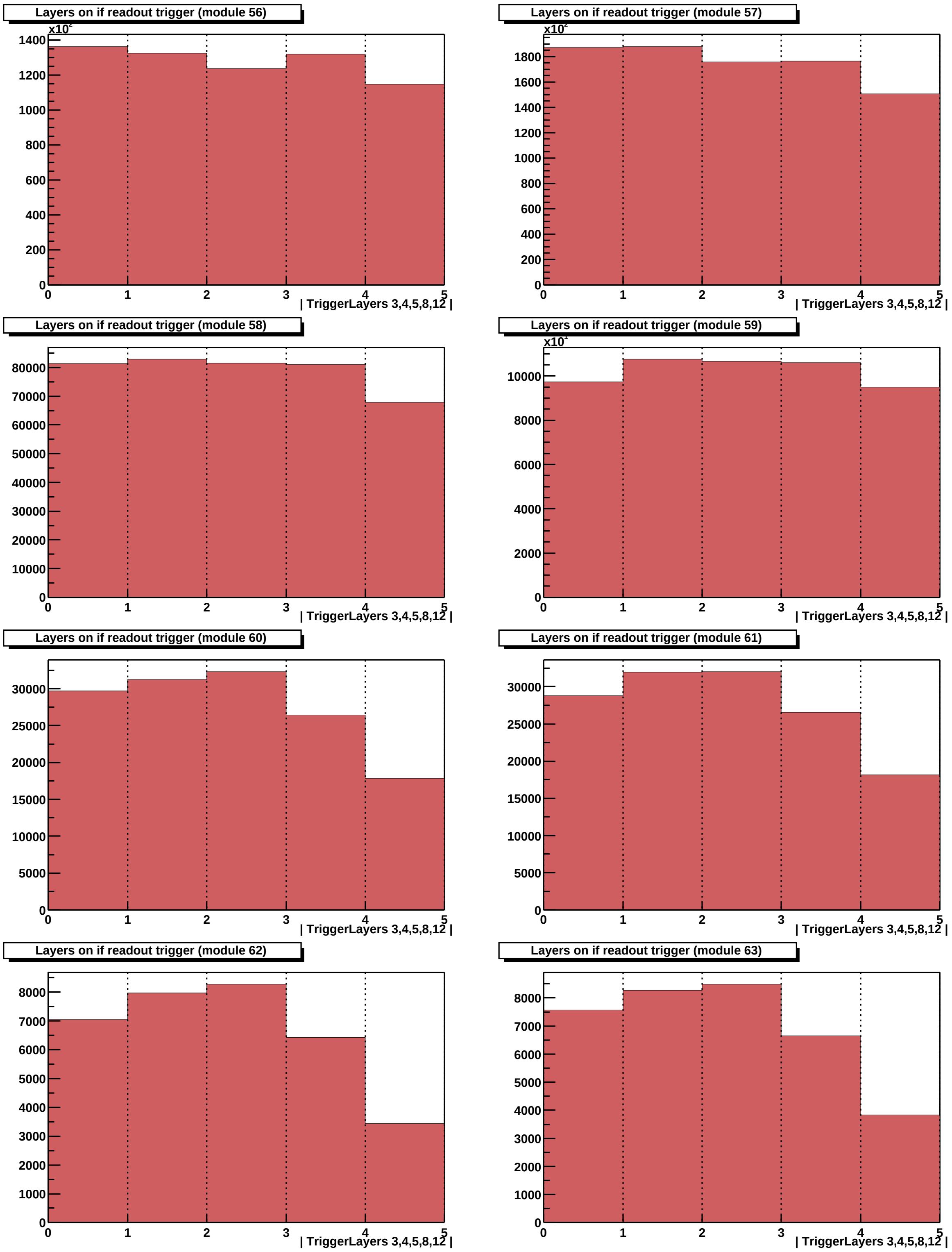

Abbildung C.16: Die Histogramme `LayerIfTrigger_<Modulnummer>` für die Module 56 bis 63.

C.3 Zahl der getroffenen Triggerlagen während ineffizienzter Ereignisse

Im Falle ineffizienzter Ereignisse werden die getroffenen Triggerlagen in den Histogrammen `LayerNoTrigger_<Modulnummer>` der ausführenden Klasse `TMuLayerLoop` gezählt. Hinsichtlich der Form sind diese Histogramme mit denen des vorangegangenen Abschnitts identisch. Ist ein Histogramm nicht abgedruckt, so hat das entsprechende Modul keinerlei ineffizientes Verhalten gezeigt.

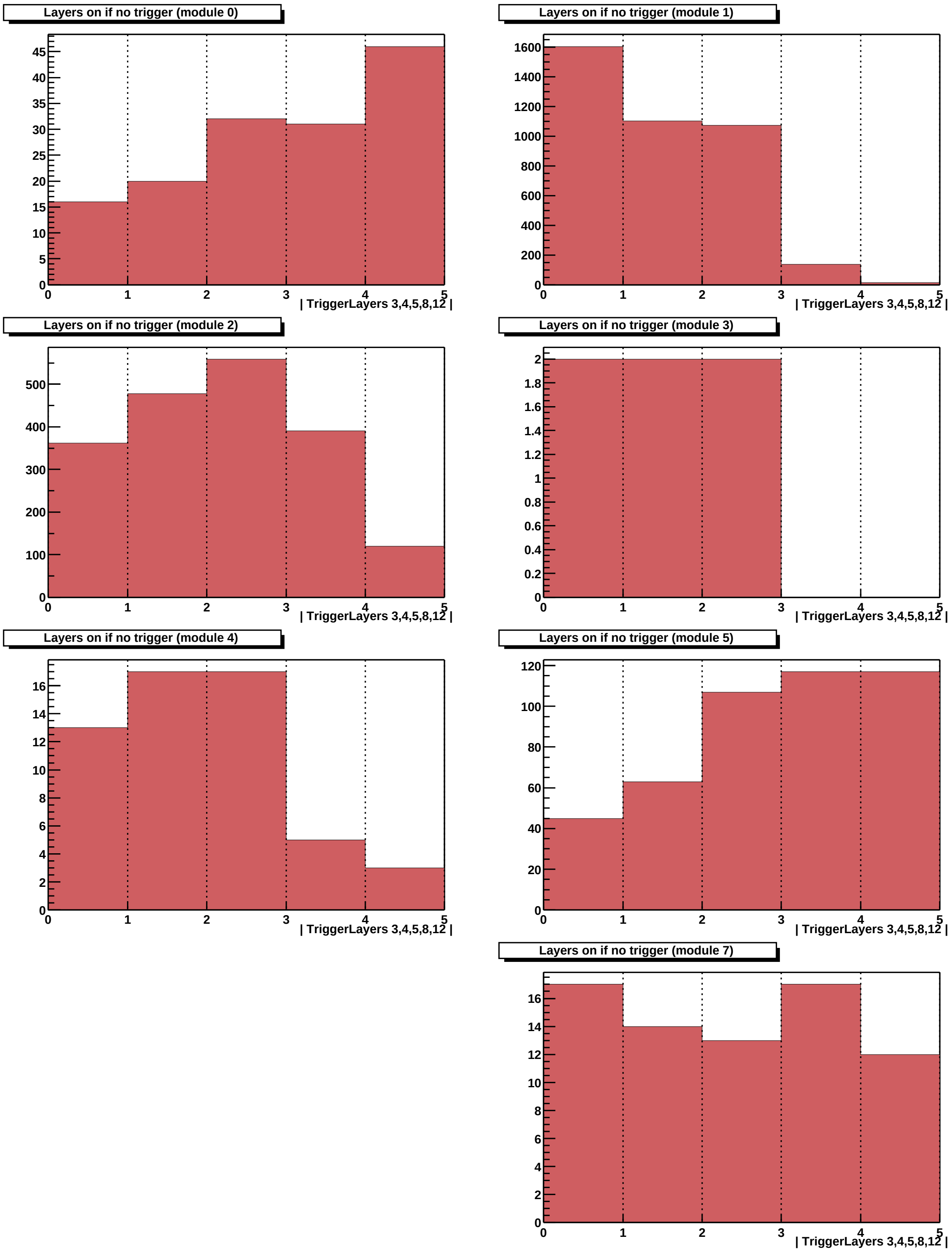

Abbildung C.17: Die Histogramme `LayerNoTrigger_<Modulnummer>` für die Module 0 bis 7.

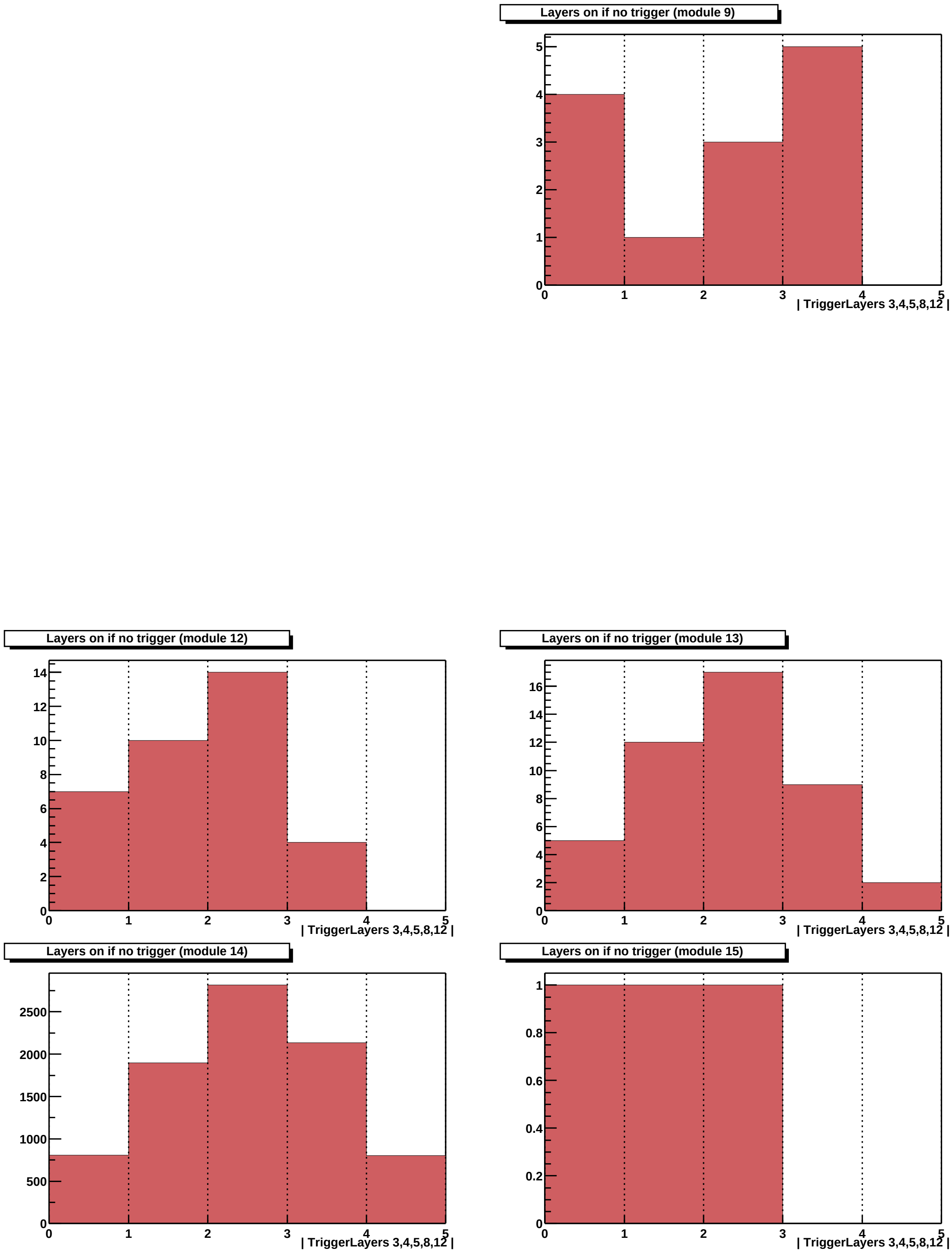

Abbildung C.18: Die Histogramme `LayerNoTrigger_<Modulnummer>` für die Module 8 bis 15.

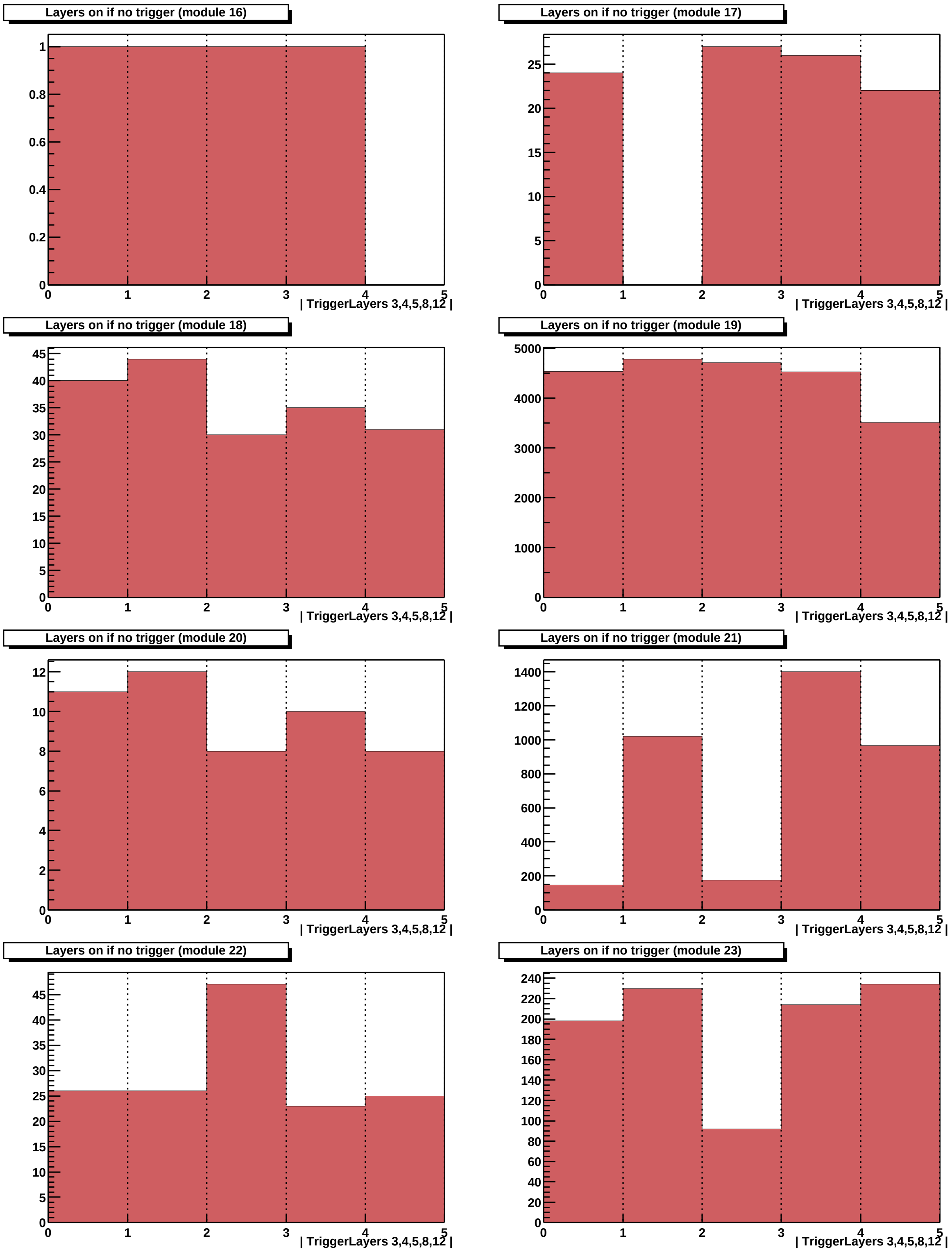

Abbildung C.19: Die Histogramme `LayerNoTrigger_<Modulnummer>` für die Module 16 bis 23.

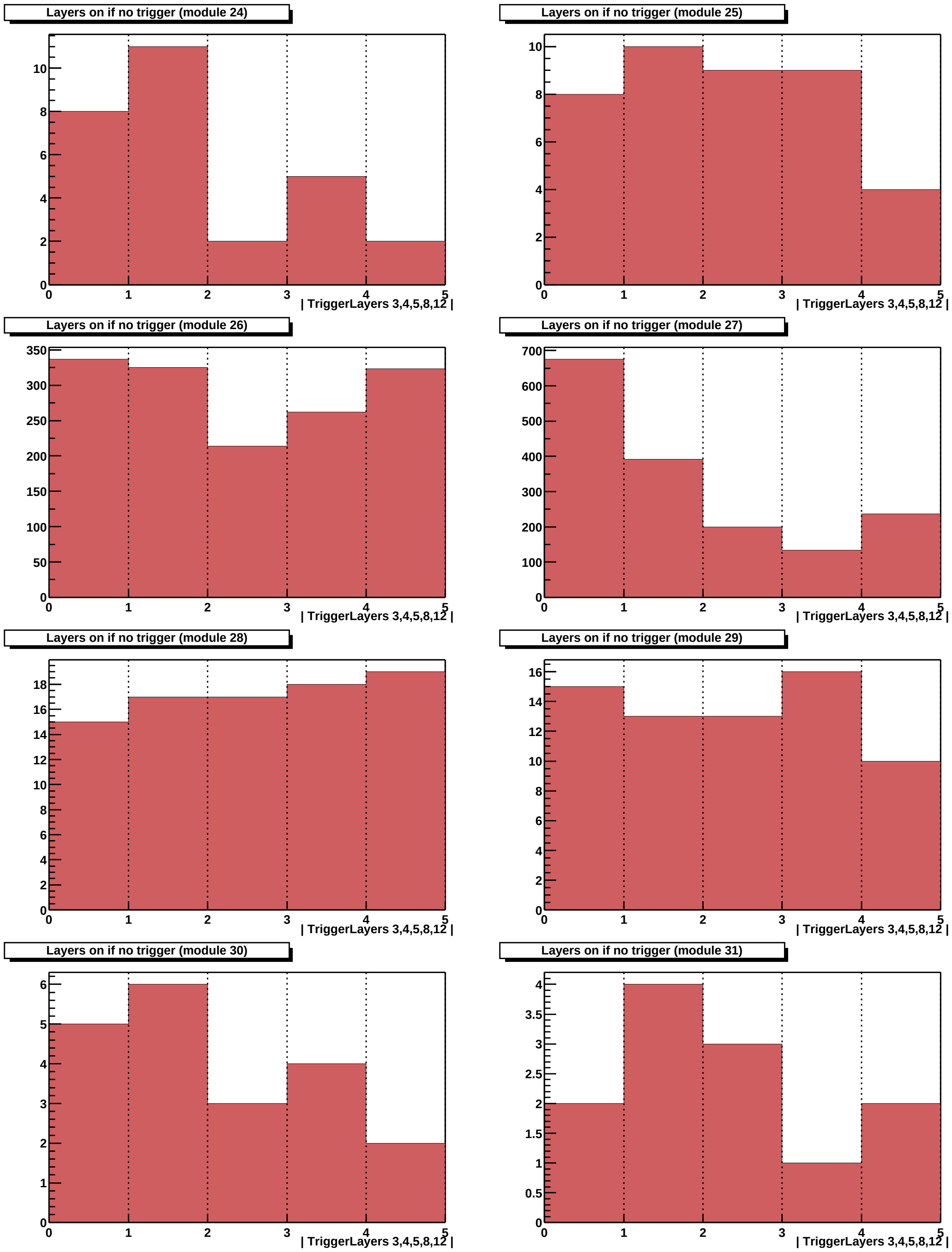

Abbildung C.20: Die Histogramme `LayerNoTrigger_<Modulnummer>` für die Module 24 bis 31.

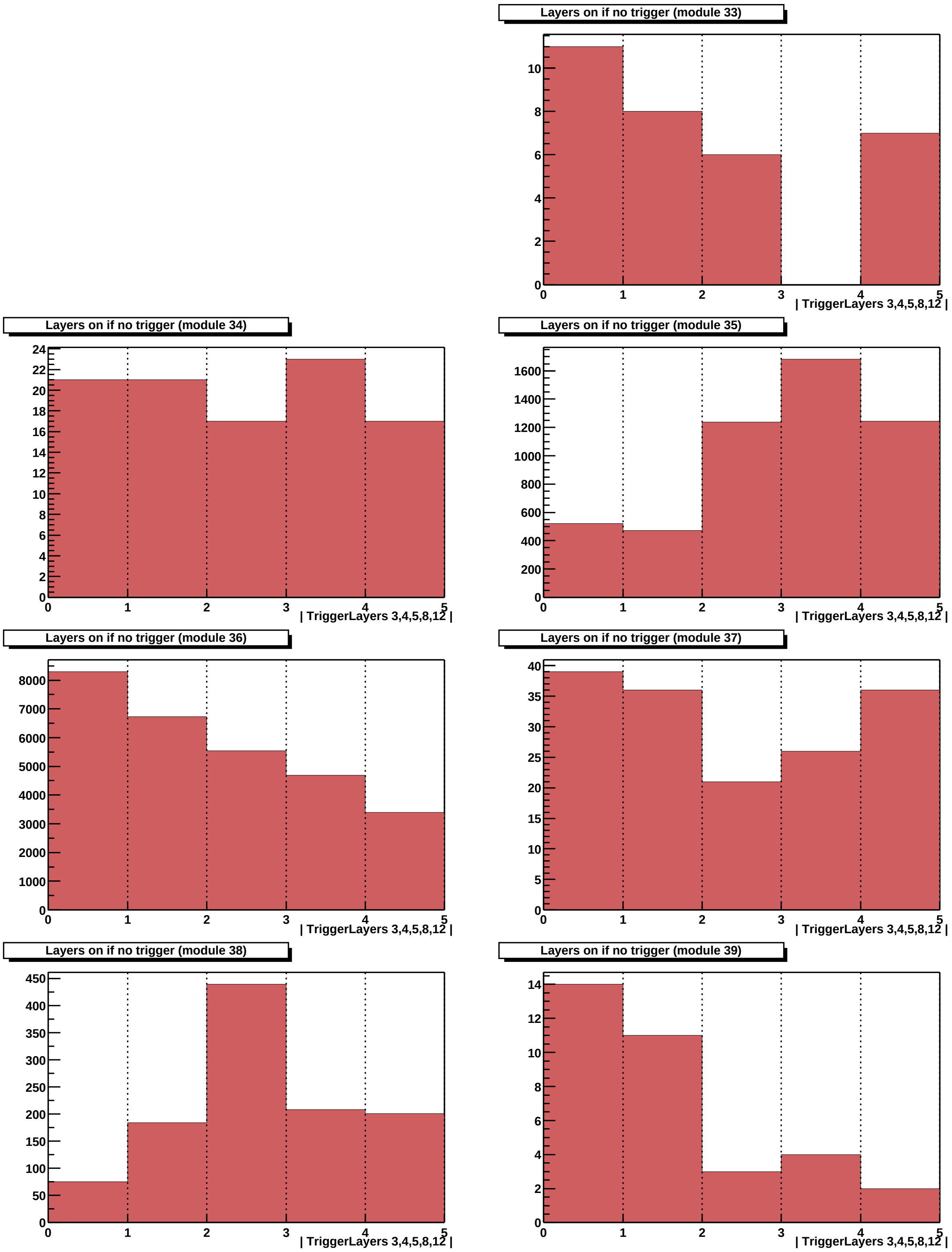

Abbildung C.21: Die Histogramme `LayerNoTrigger_<Modulnummer>` für die Module 32 bis 39.

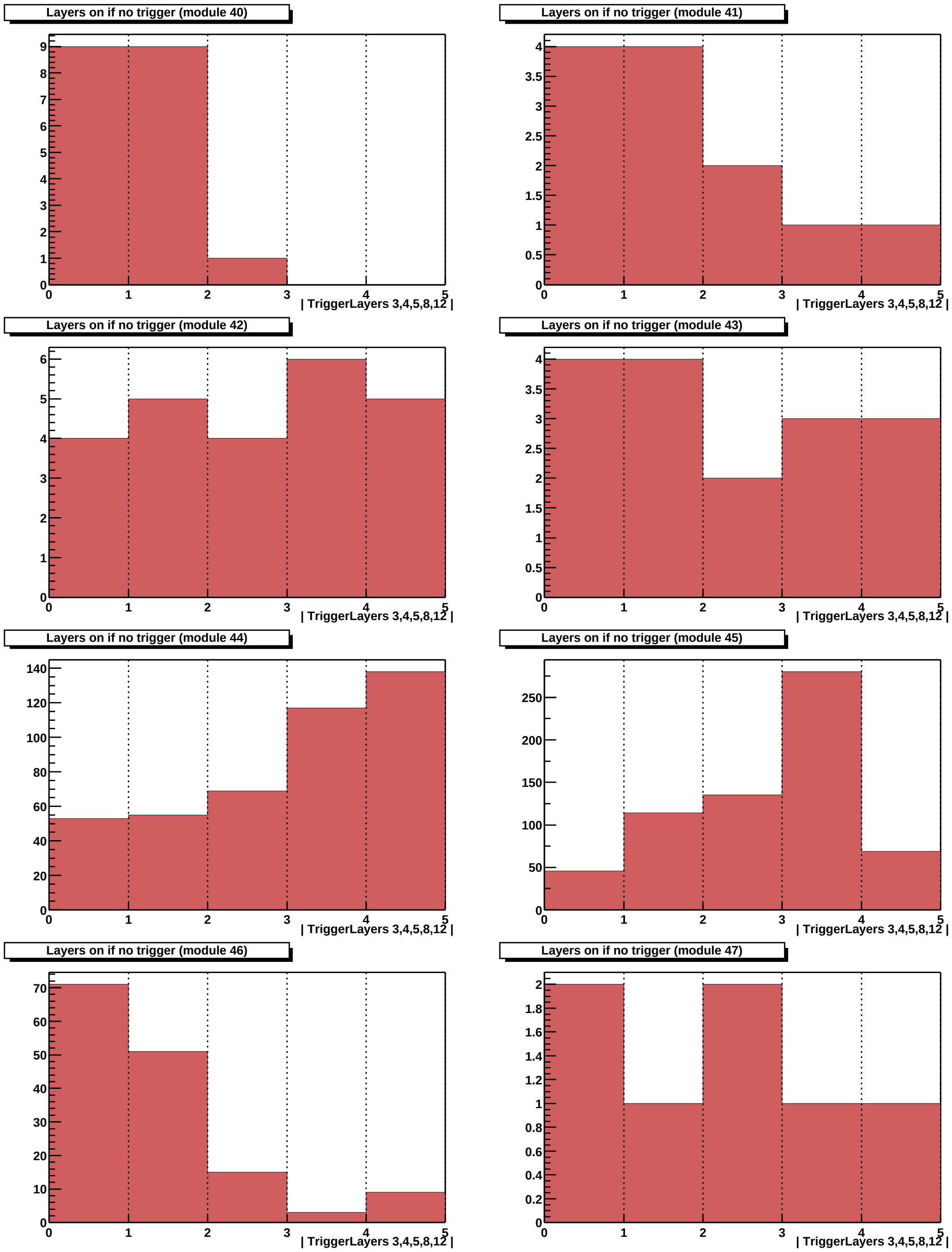

Abbildung C.22: Die Histogramme LayerNoTrigger_<Modulnummer> für die Module 40 bis 47.

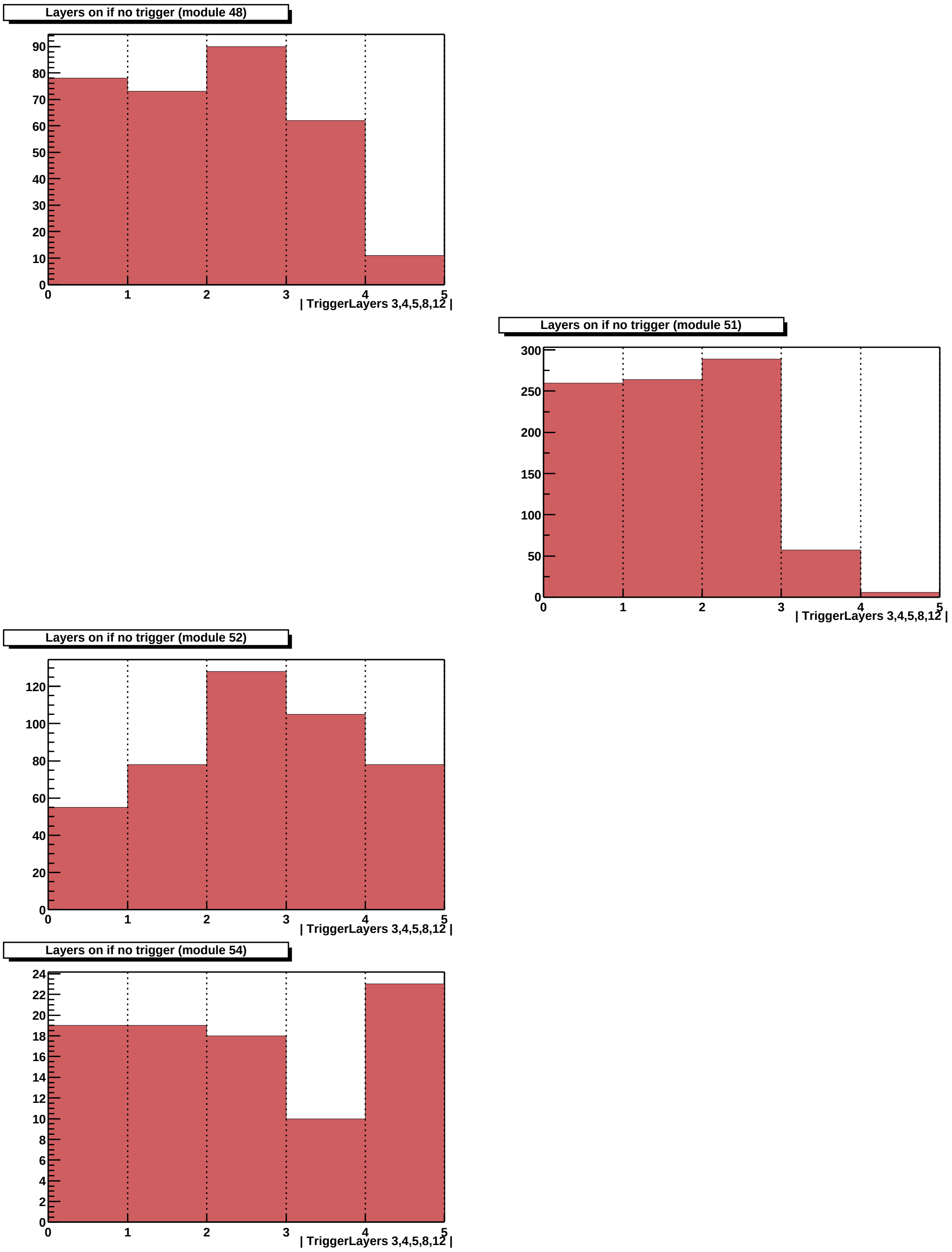

Abbildung C.23: Die Histogramme `LayerNoTrigger_<Modulnummer>` für die Module 48 bis 55.

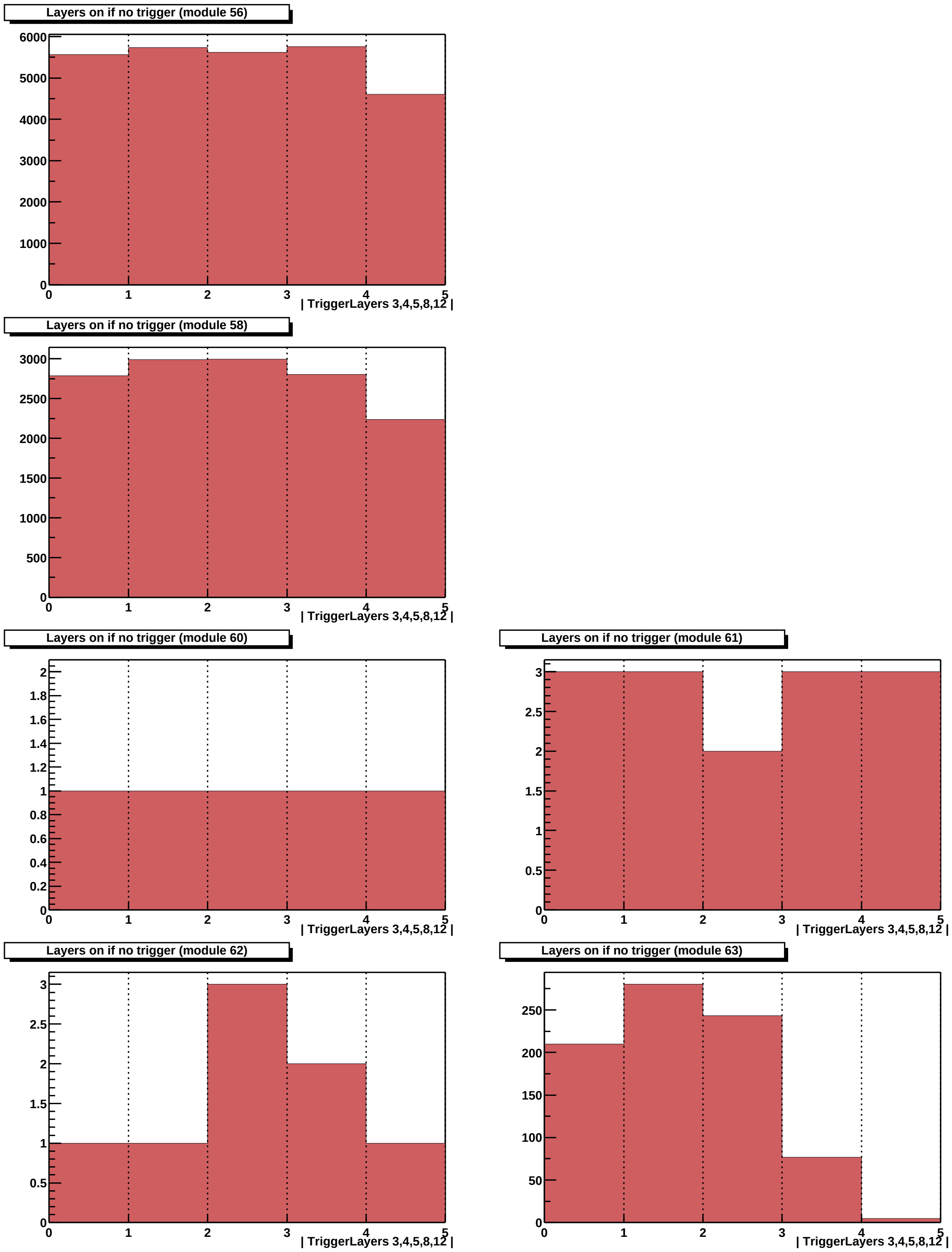

***Abbildung C.24*:** Die Histogramme `LayerNoTrigger_<Modulnummer>` für die Module 56 bis 63.

C.4 Lageneffizienzen

Dieser Abschnitt umfaßt den vollständigen Satz von vierundsechzig Histogrammen `LayerEffHisto_<Modulnummer>`, wie sie von der Klasse `TMuITKRLoop` erzeugt wurden. Es fanden ca. 600.000 Ereignisse kosmischer und Halo-Myonen, aufgenommen während des Jahres 2000, Eingang in die Analyse. Die Effizienzen der Lagen 0 bis 2 sowie 13 bis 15 können von `TMuITKRLoop` nicht ermittelt werden und sind somit in den Histogrammen nicht eingetragen. Bei den Lagen 3 und 12 ist die quantitative Aussage nicht in jedem Falle zweifelsfrei.

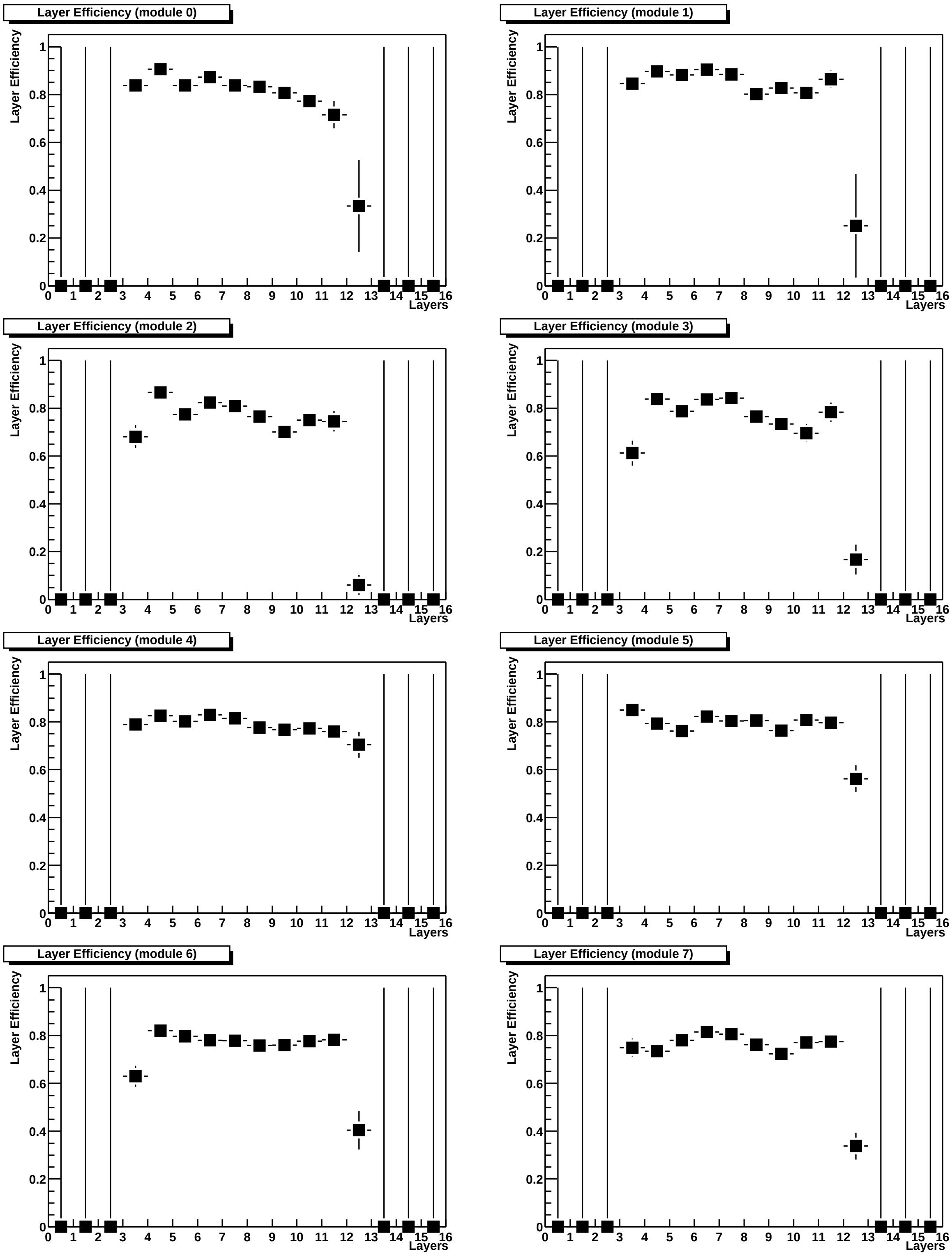

Abbildung C.25: Die Histogramme `LayerEffHisto_<Modulnummer>` für die Module 0 bis 7.

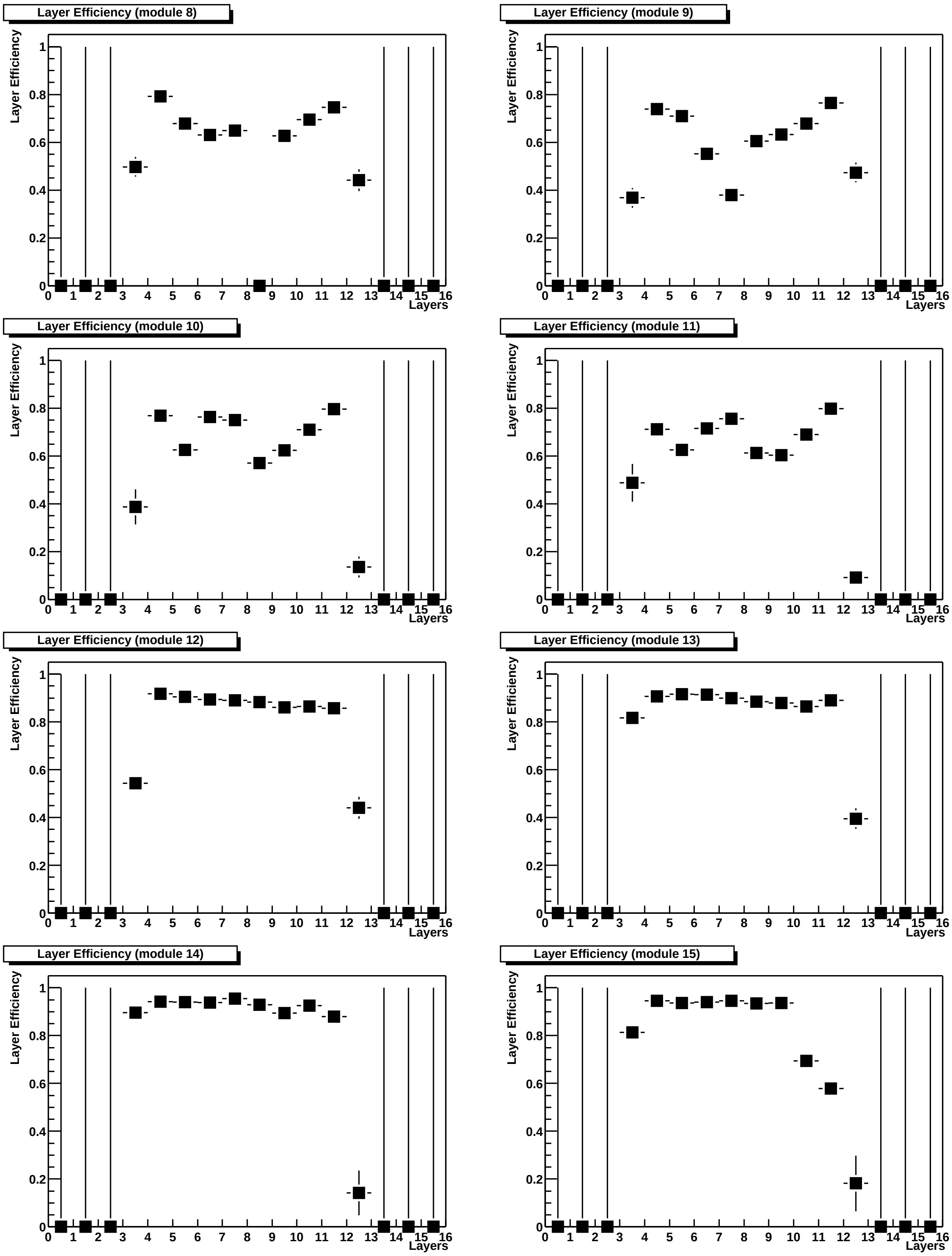

Abbildung C.26: Die Histogramme `LayerEffHisto_<Modulnummer>` für die Module 8 bis 15.

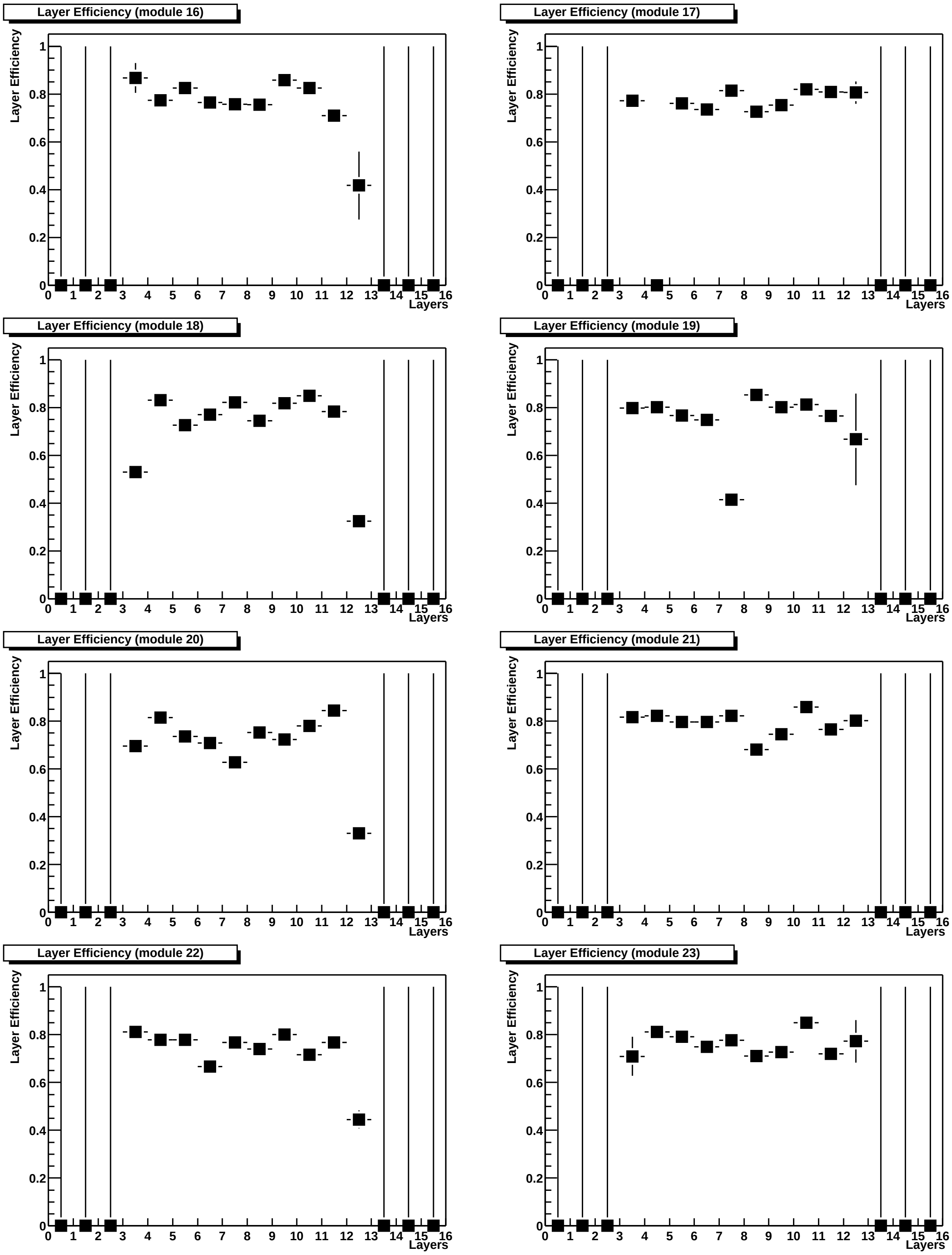

Abbildung C.27: Die Histogramme `LayerEffHisto_<Modulnummer>` für die Module 16 bis 23.

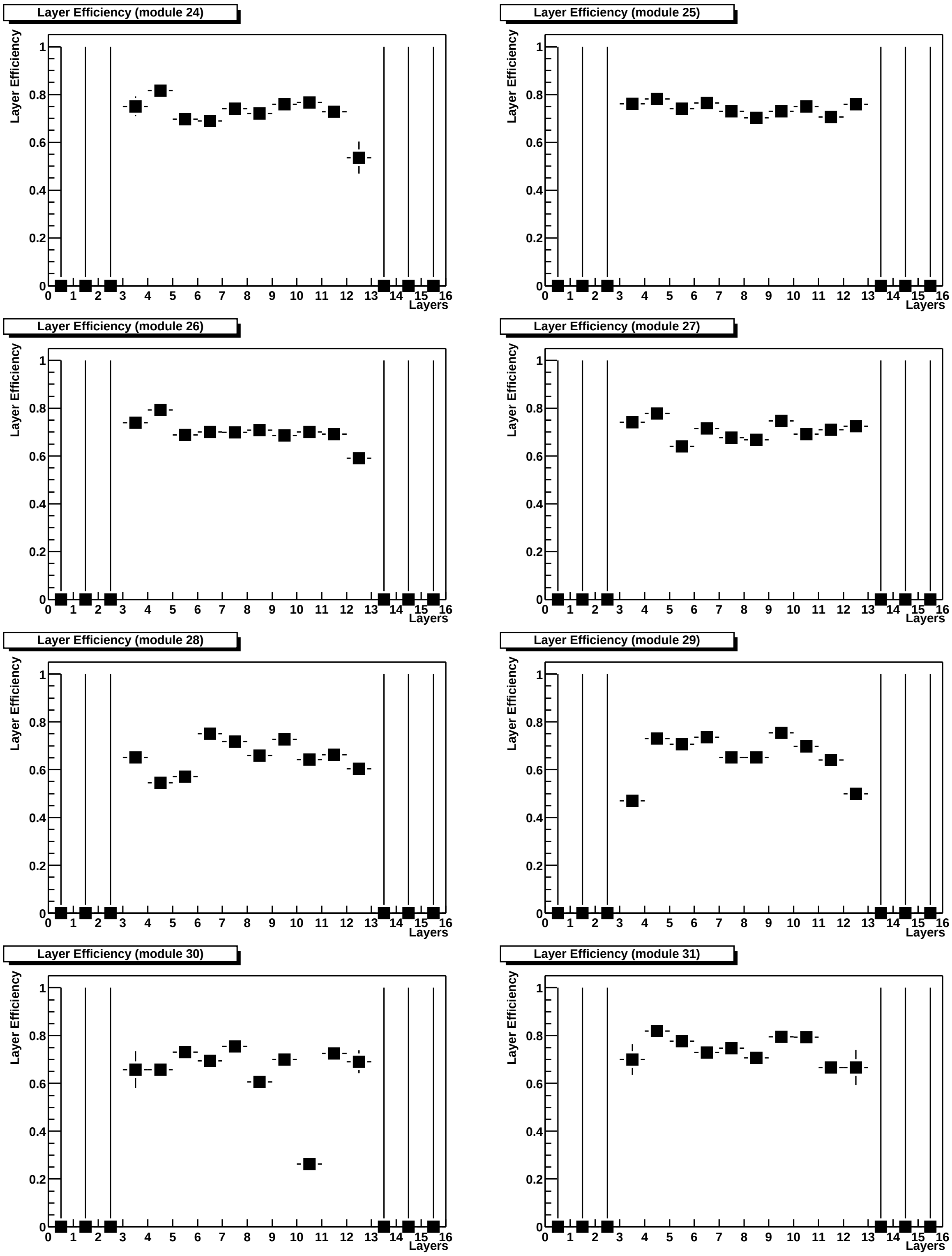

Abbildung C.28: Die Histogramme `LayerEffHisto_<Modulnummer>` für die Module 24 bis 31.

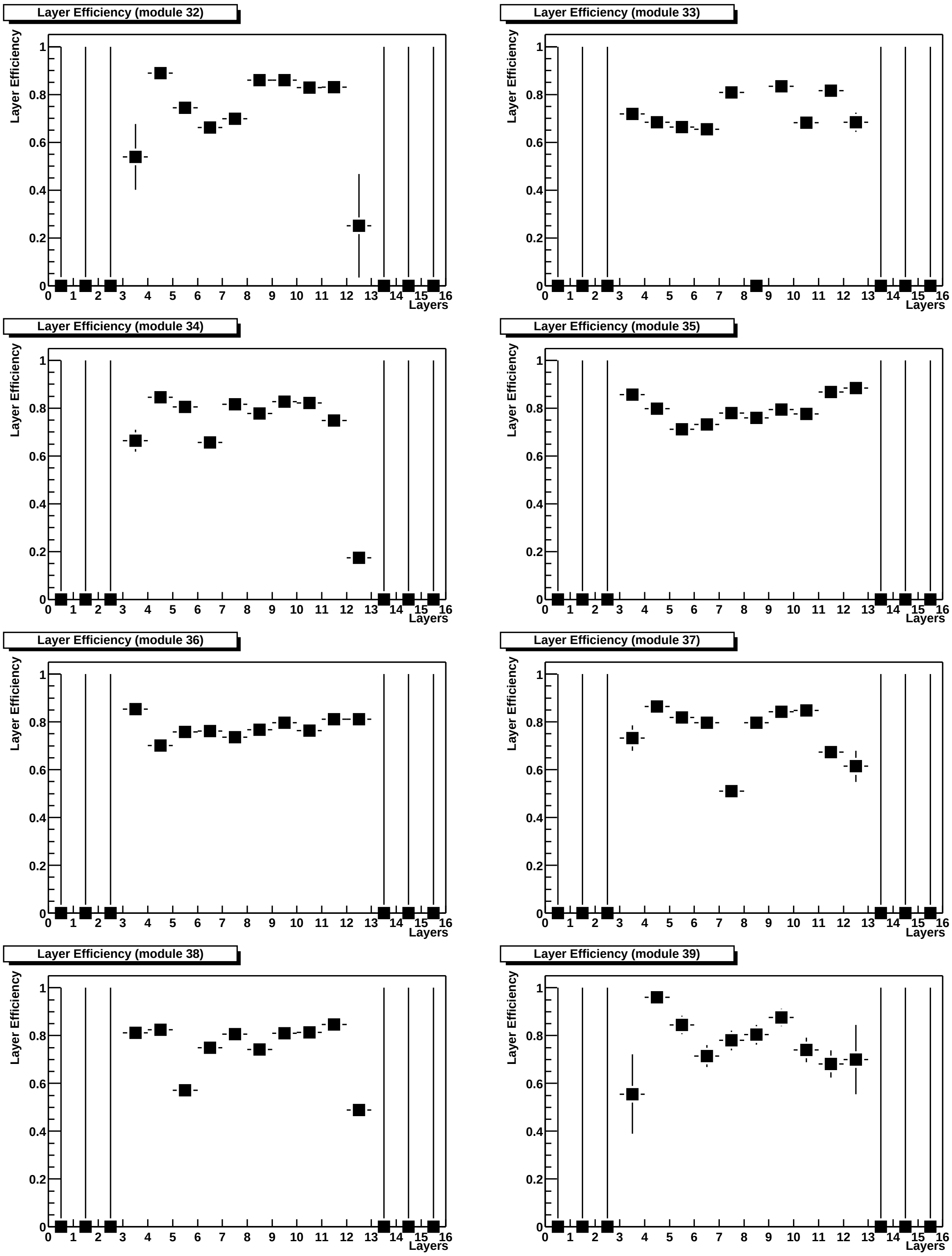

Abbildung C.29: Die Histogramme LayerEffHisto_<Modulnummer> für die Module 32 bis 39.

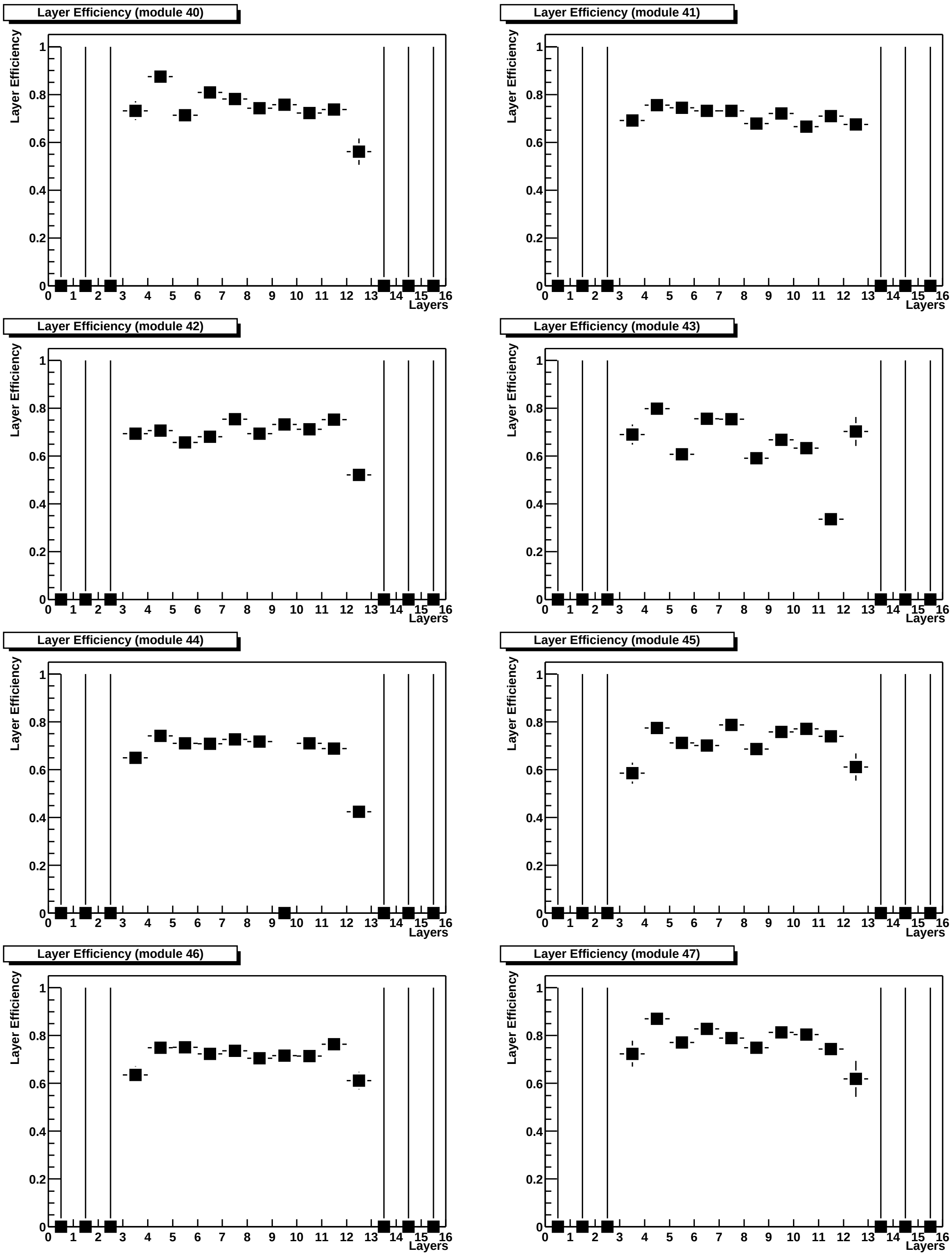

Abbildung C.30: Die Histogramme LayerEffHisto_<Modulnummer> für die Module 40 bis 47.

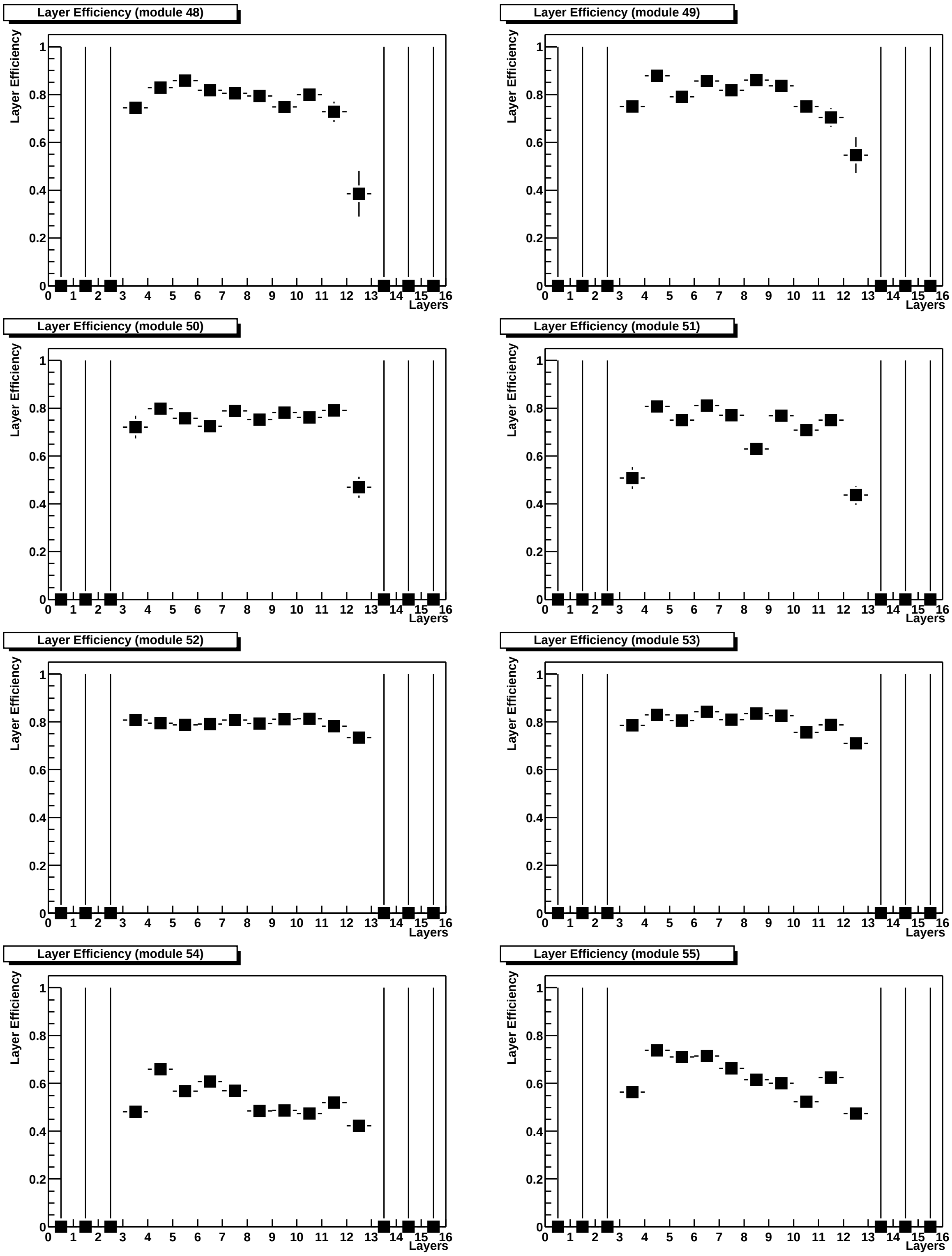

Abbildung C.31: Die Histogramme `LayerEffHisto_<Modulnummer>` für die Module 48 bis 55.

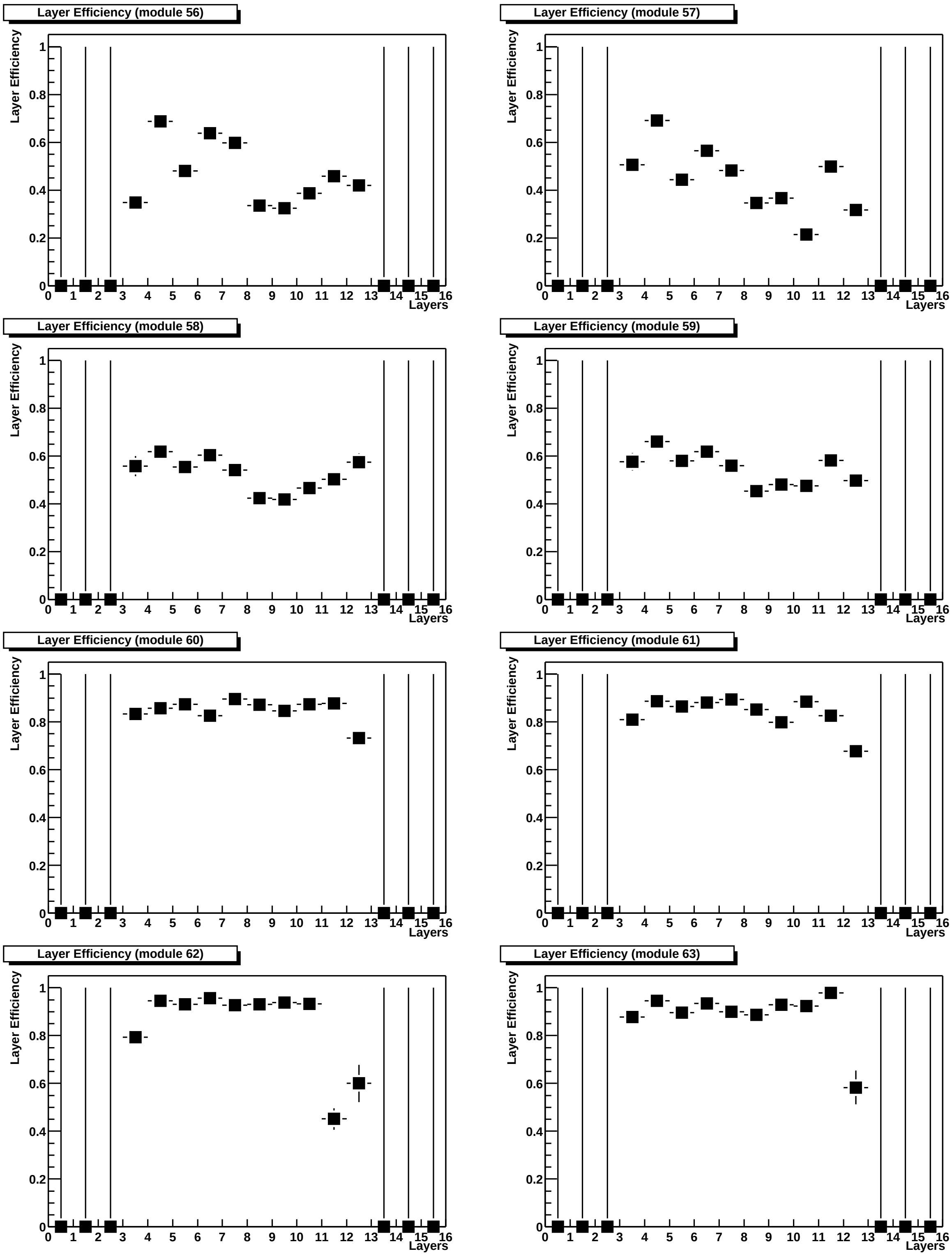

Abbildung C.32: Die Histogramme `LayerEffHisto_<Modulnummer>` für die Module 56 bis 63.

C.5 Zahl der getroffenen Drähte pro Ereignis

Die Klasse `TMuBIGLoop`, welche Ereignisse mit hoher Aktivität im Zentralen Myon-Detektor aufsuchen und verrauschte Drähte detektieren kann, erstellt für ihre Aufgaben die vierundsechzig Histogramme `HitsPerEvent_<Modulnummer>`. Diese sind im folgenden vollständig aufgeführt. Darüberhinaus ist das Histogramm `HitsPerEvent` abgedruckt, welches die Häufigkeit gewisser Anzahlen getroffener Drähte pro Ereignis im Zentralen Myon-Detektor aufschlüsselt. Für alle Histogramme wurde wegen der großen Zahl ausgewerteter Ereignisse eine logarithmische Darstellung gewählt.

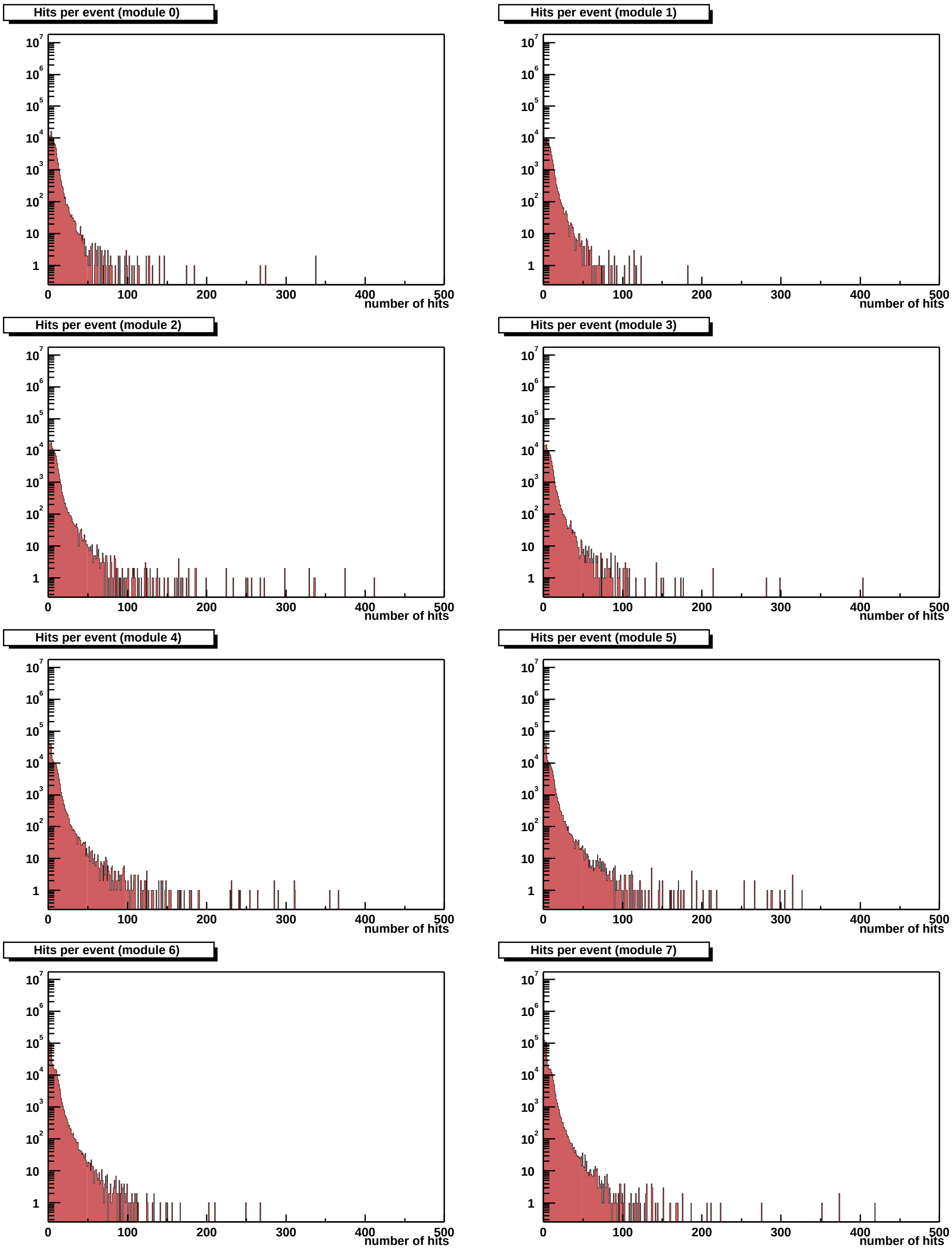

Abbildung C.33: Die Histogramme `HitsPerEvent_<Modulnummer>` für die Module 0 bis 7.

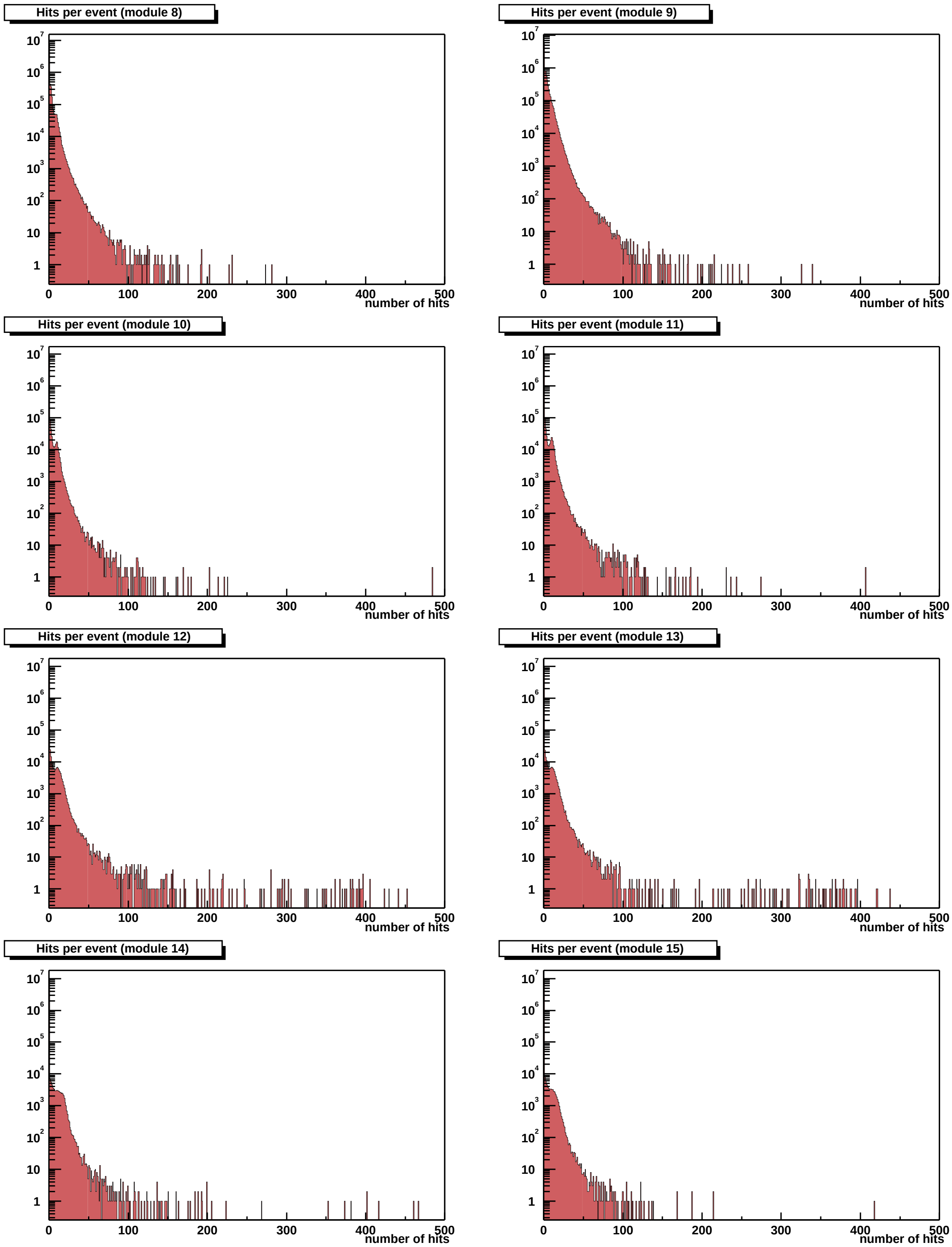

Abbildung C.34: Die Histogramme `HitsPerEvent_<Modulnummer>` für die Module 8 bis 15.

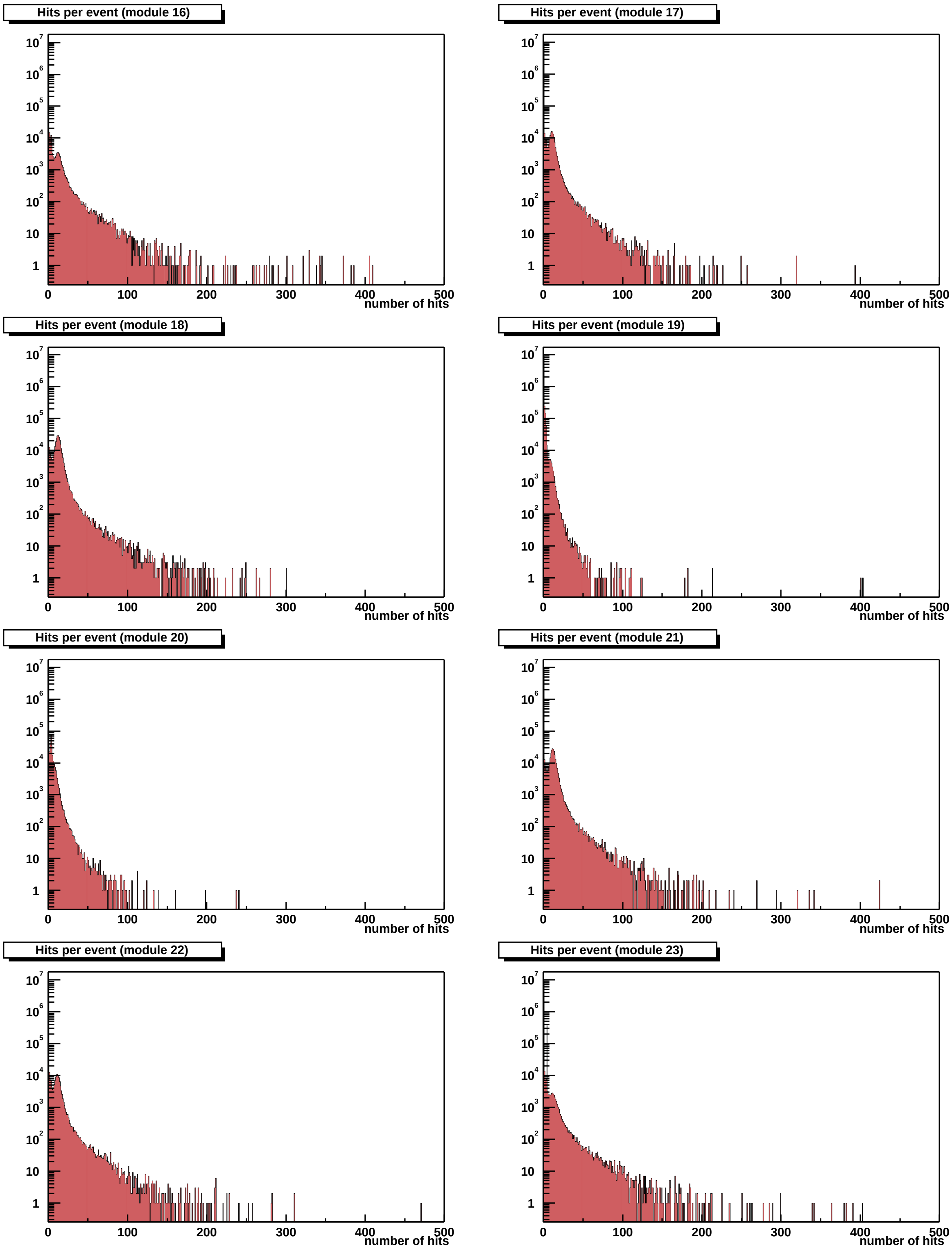

Abbildung C.35: Die Histogramme `HitsPerEvent_<Modulnummer>` für die Module 16 bis 23.

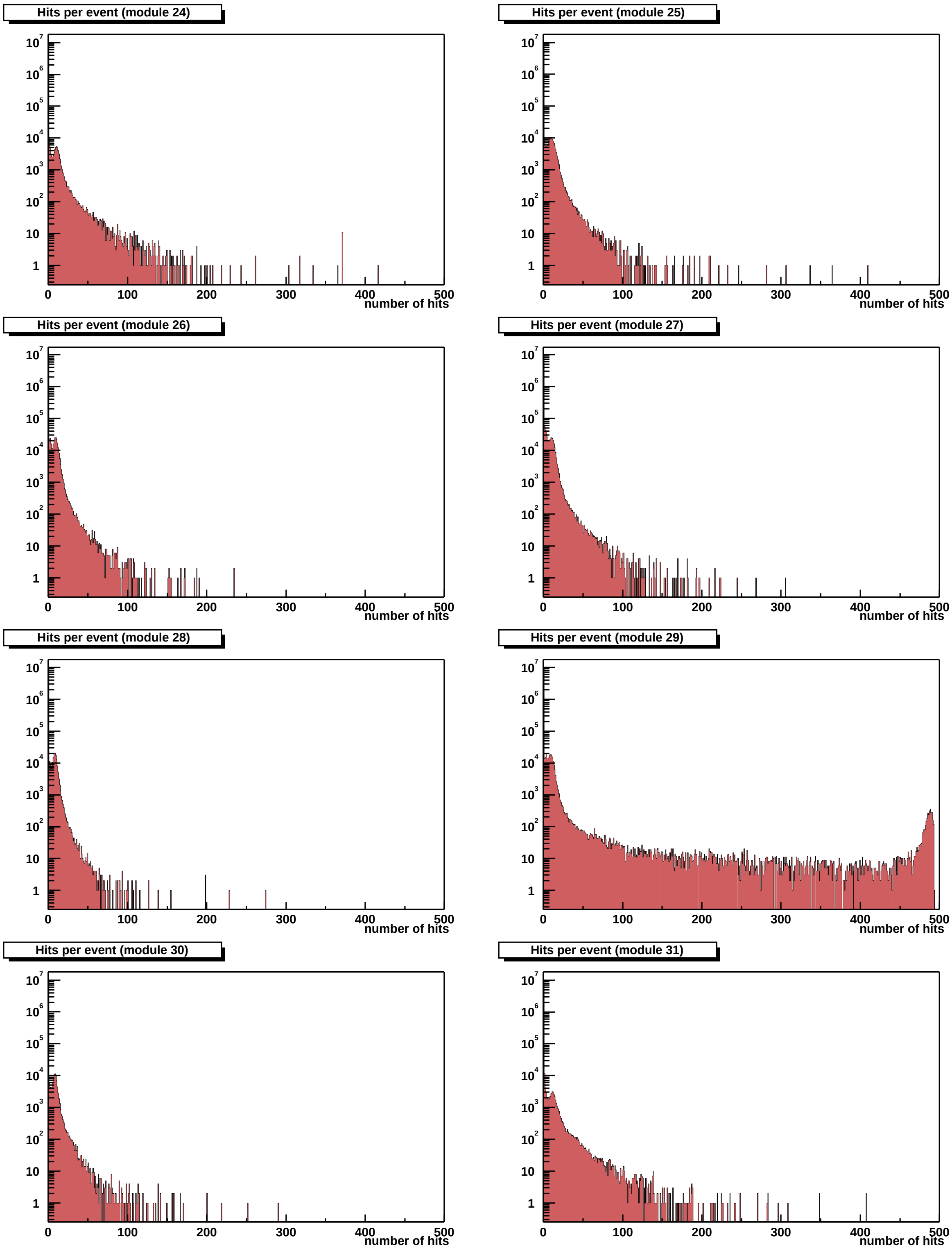

Abbildung C.36: Die Histogramme HitsPerEvent_<Modulnummer> für die Module 24 bis 31.

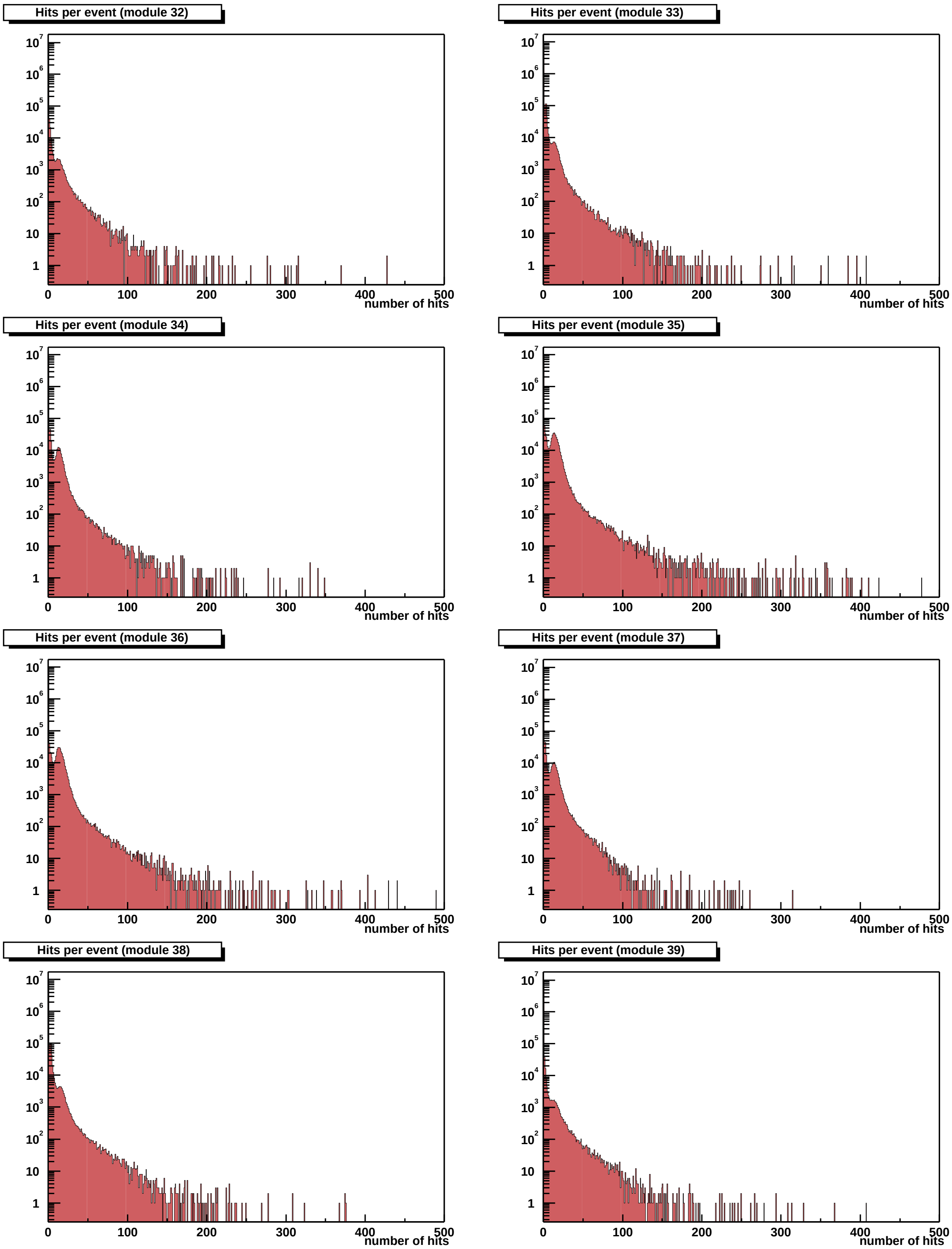

Abbildung C.37: Die Histogramme `HitsPerEvent_<Modulnummer>` für die Module 32 bis 39.

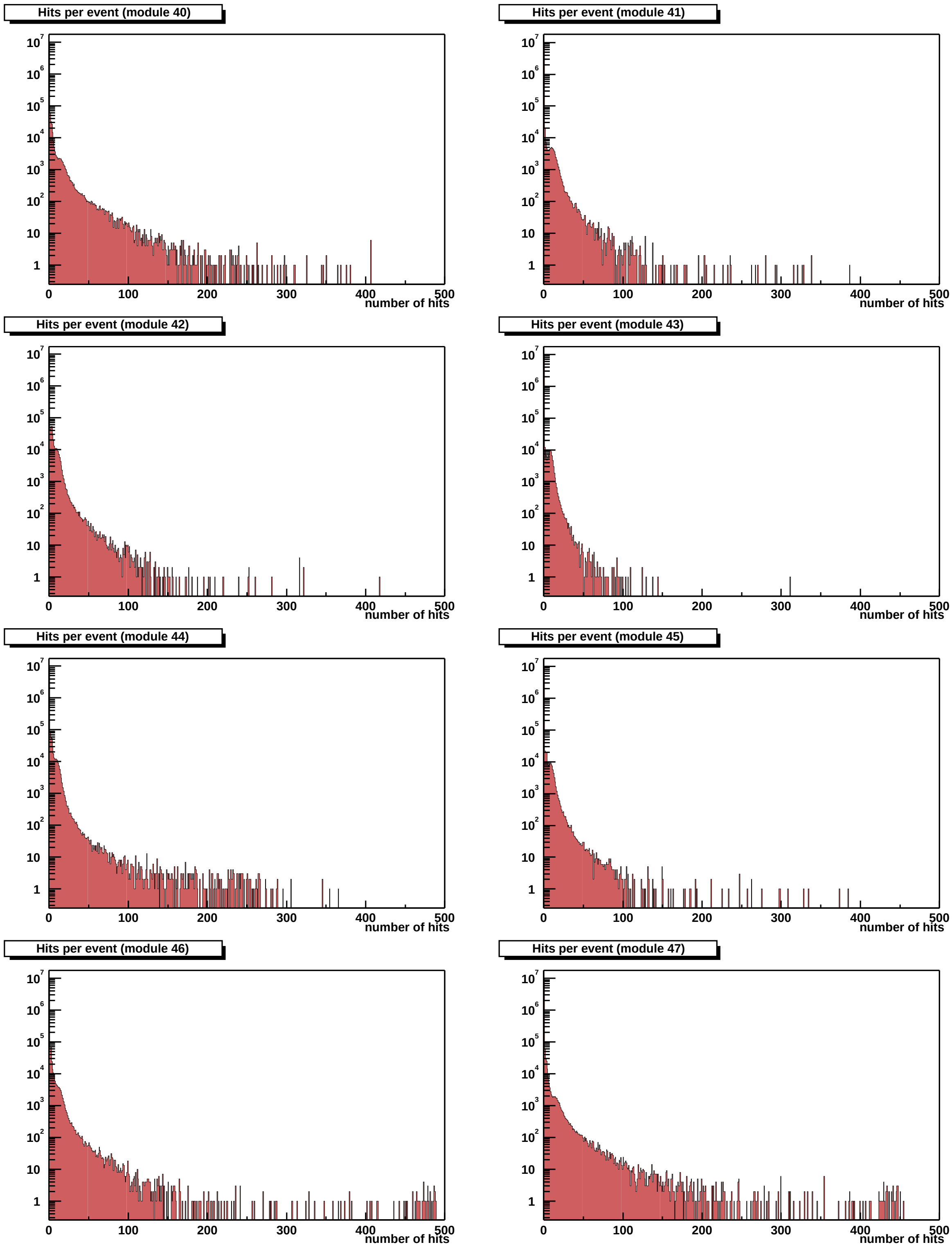

Abbildung C.38: Die Histogramme `HitsPerEvent_<Modulnummer>` für die Module 40 bis 47.

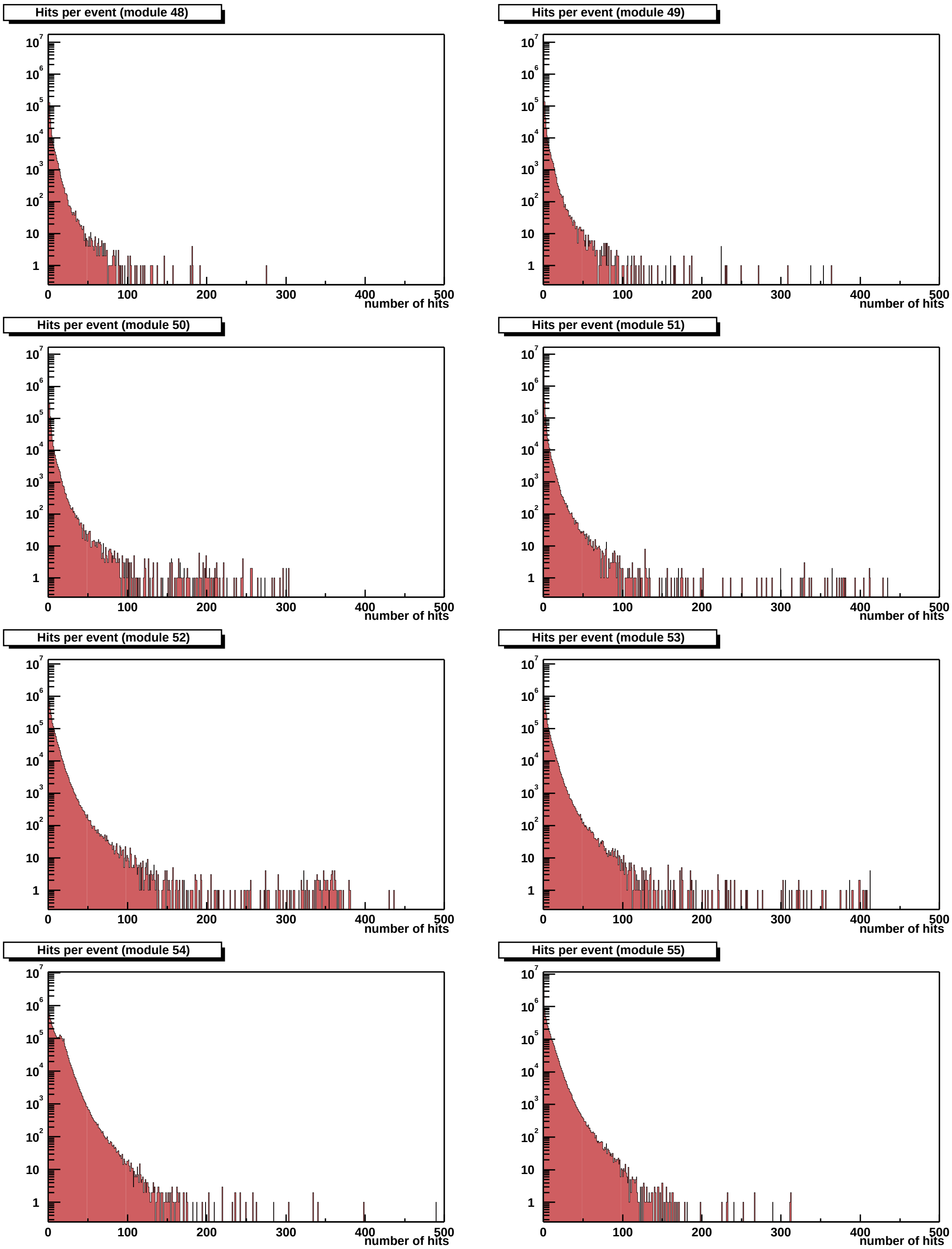

Abbildung C.39: Die Histogramme HitsPerEvent_<Modulnummer> für die Module 48 bis 55.

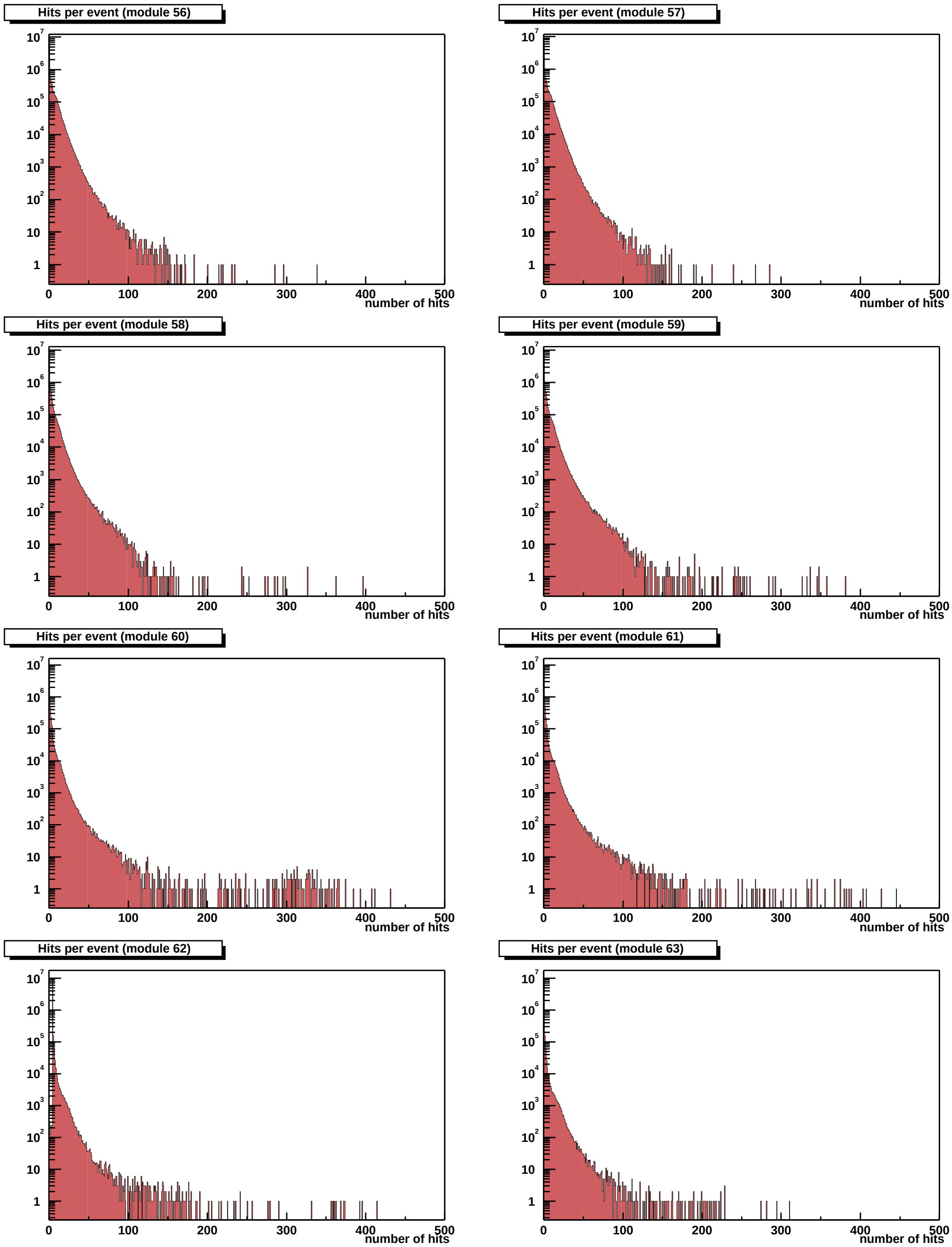

Abbildung C.40: Die Histogramme HitsPerEvent_<Modulnummer> für die Module 56 bis 63.

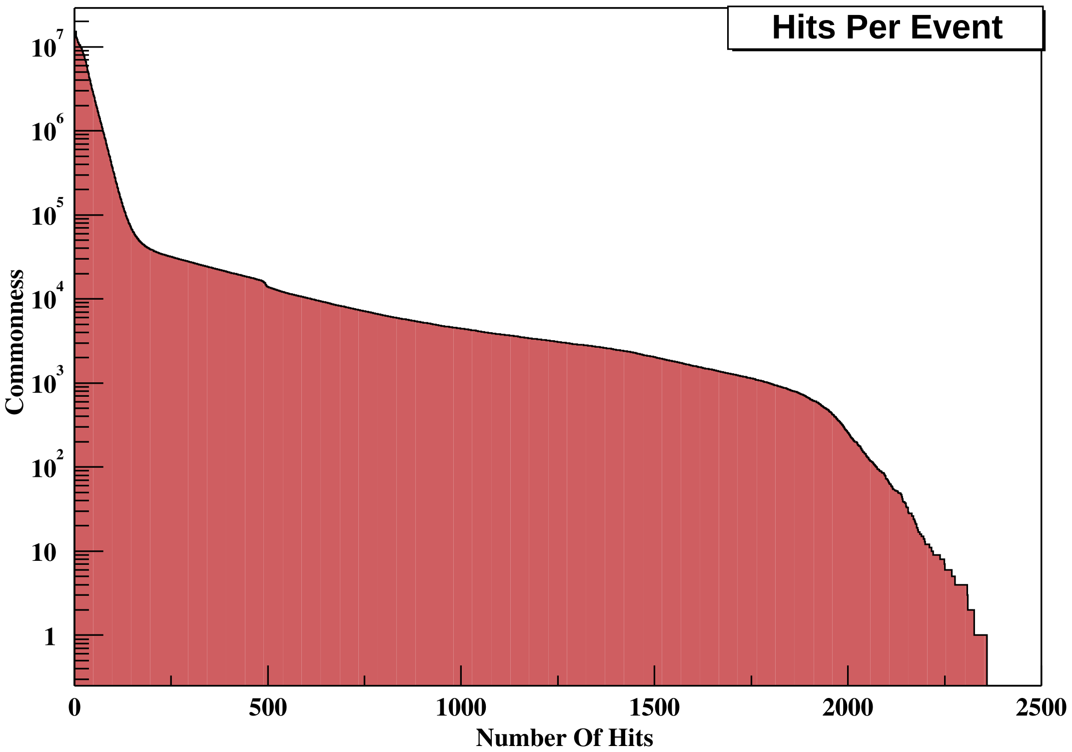

Abbildung C.41: Das Histogramm `HitsPerEvent`.

C.6 Die zeitliche Verteilung von Lagenkoinzidenzen

Die nachfolgend abgebildeten Histogramme `ShiftTZero_<Modulnummer>`, wie sie von der ausführenden Klasse `TMuShiftLoop` erzeugt wurden, geben die zeitliche Verteilung von Lagenkoinzidenzen in den Zeitscheiben -1, 0 und 1 wieder. Der Eintrag auf der rechten Seite beinhaltet jeweils die Häufigkeit von Lagenkoinzidenzen ohne zeitliche Beschränkung, der zweite Eintrag von rechts ist der Übersichtlichkeit halber freigelassen.

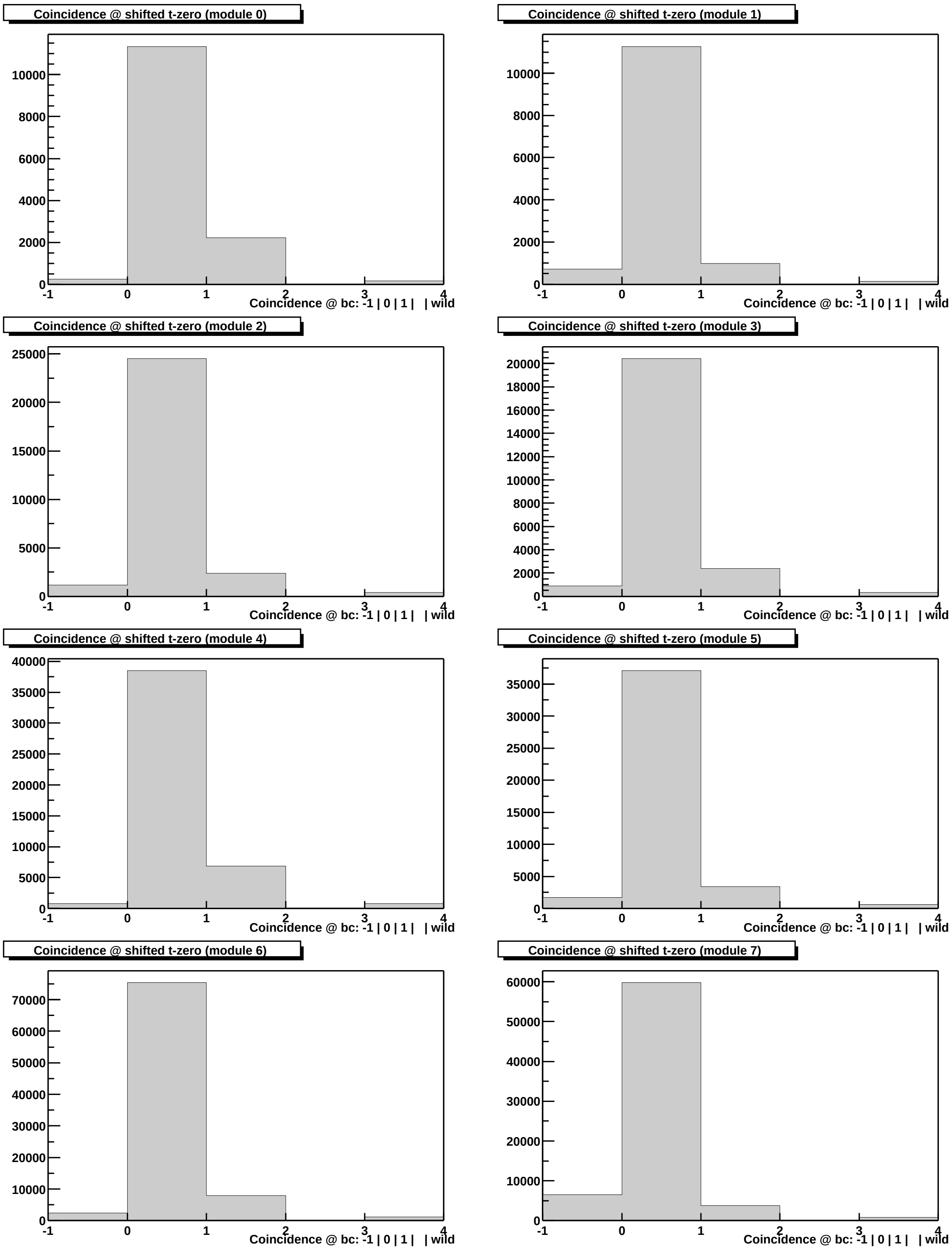

Abbildung C.42: Die Histogramme `ShiftTZero_<Modulnummer>` für die Module 0 bis 7.

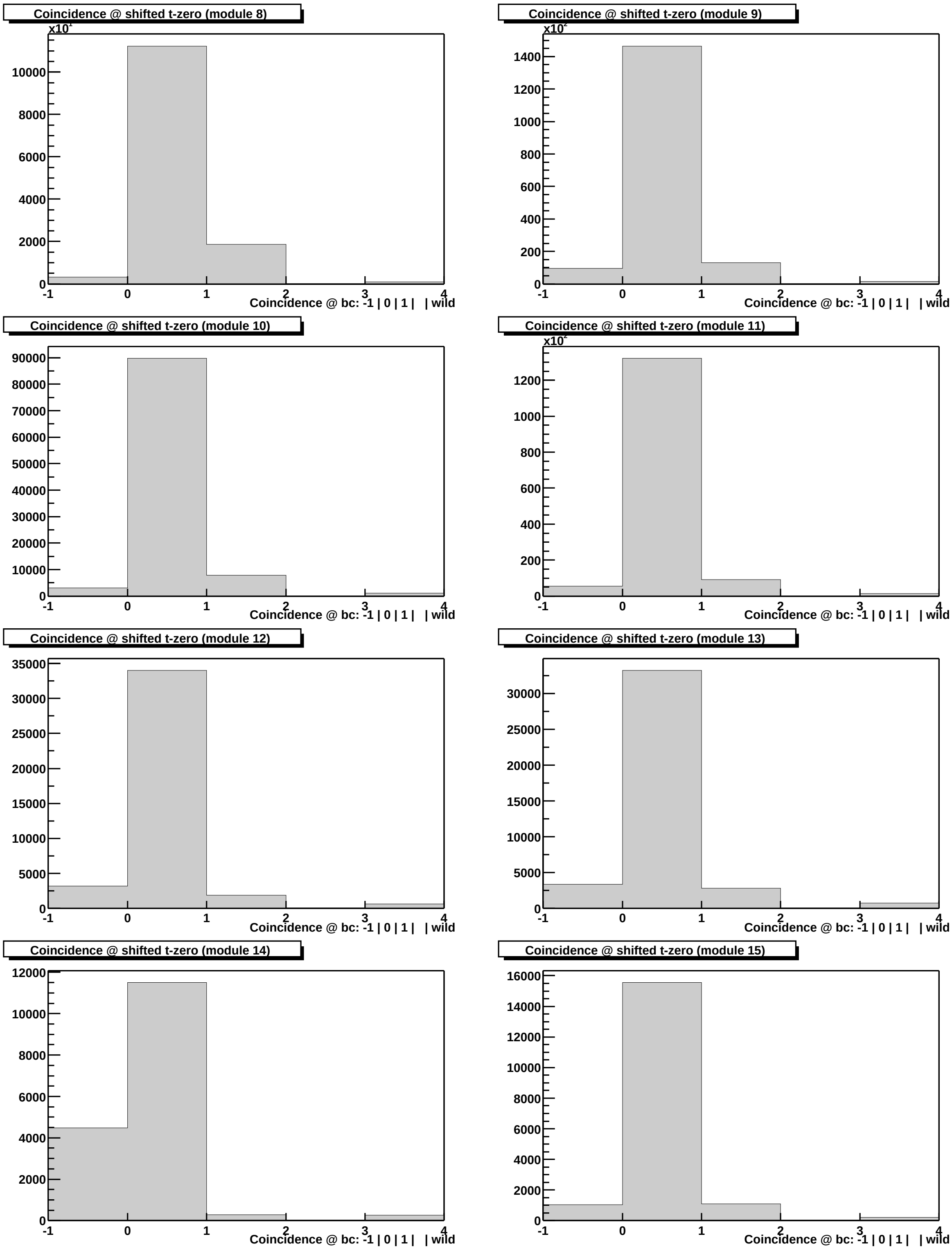

Abbildung C.43: Die Histogramme ShiftTZero_<Modulnummer> für die Module 8 bis 15.

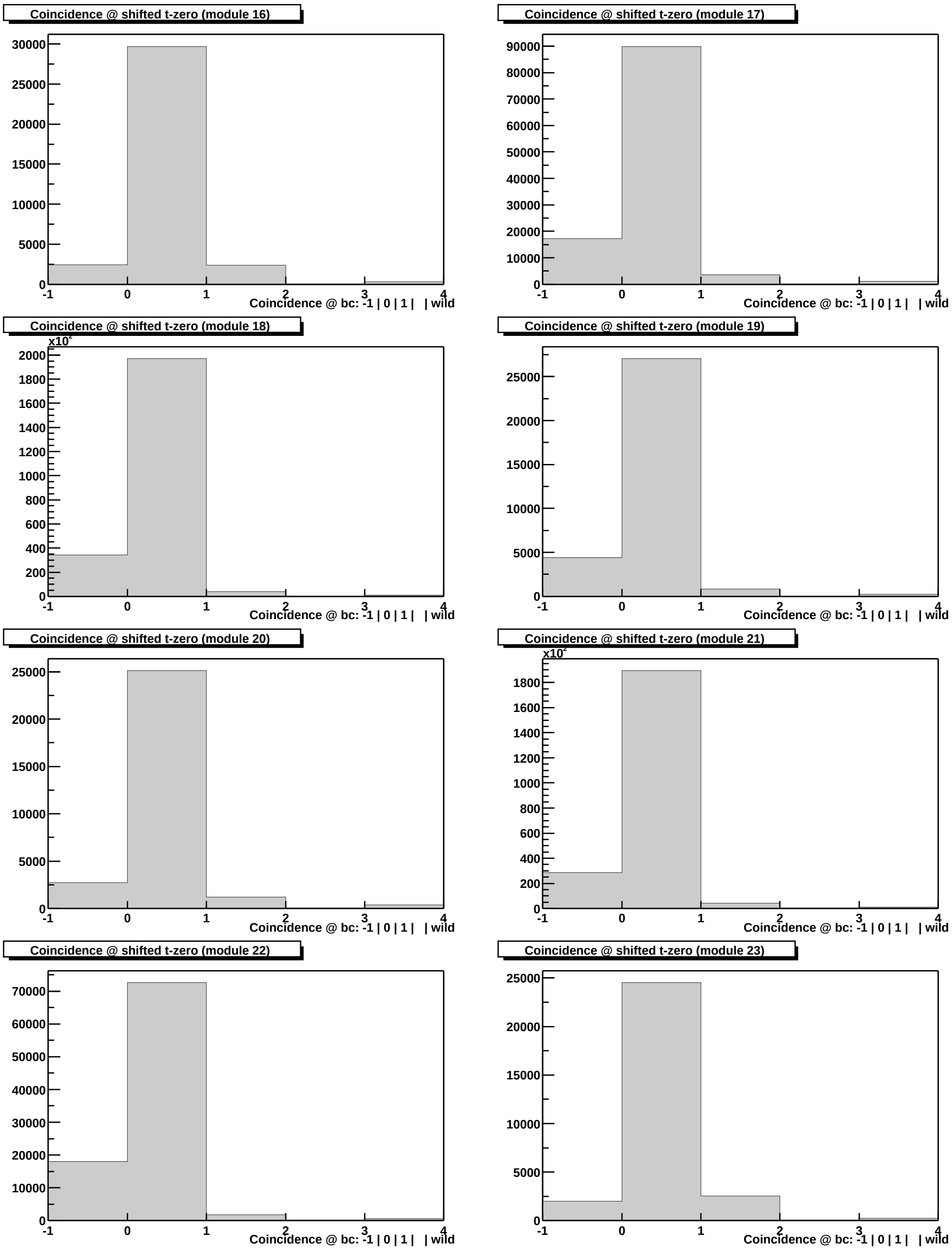

Abbildung C.44: Die Histogramme `ShiftTZero_<Modulnummer>` für die Module 16 bis 23.

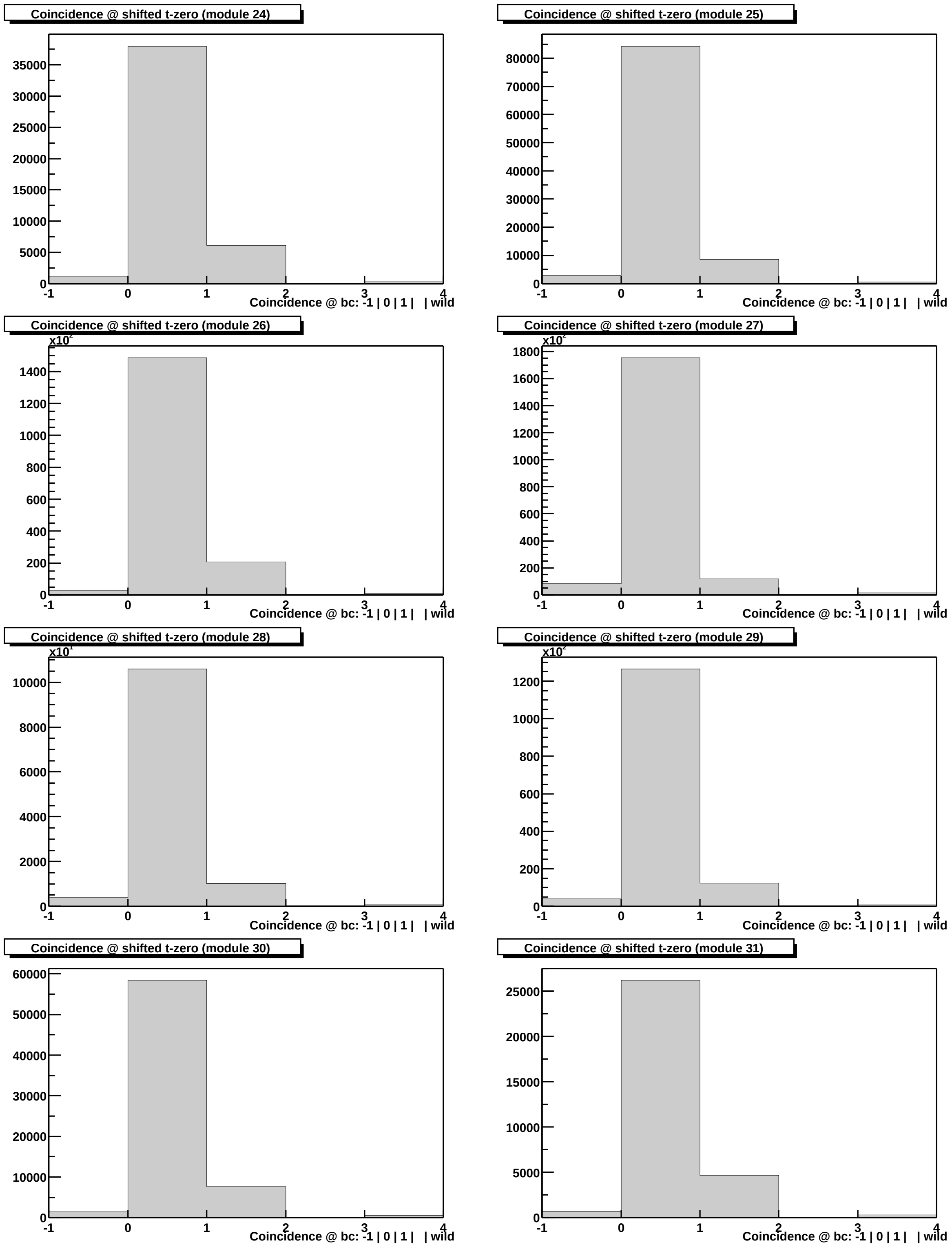

Abbildung C.45: Die Histogramme `ShiftTZero_<Modulnummer>` für die Module 24 bis 31.

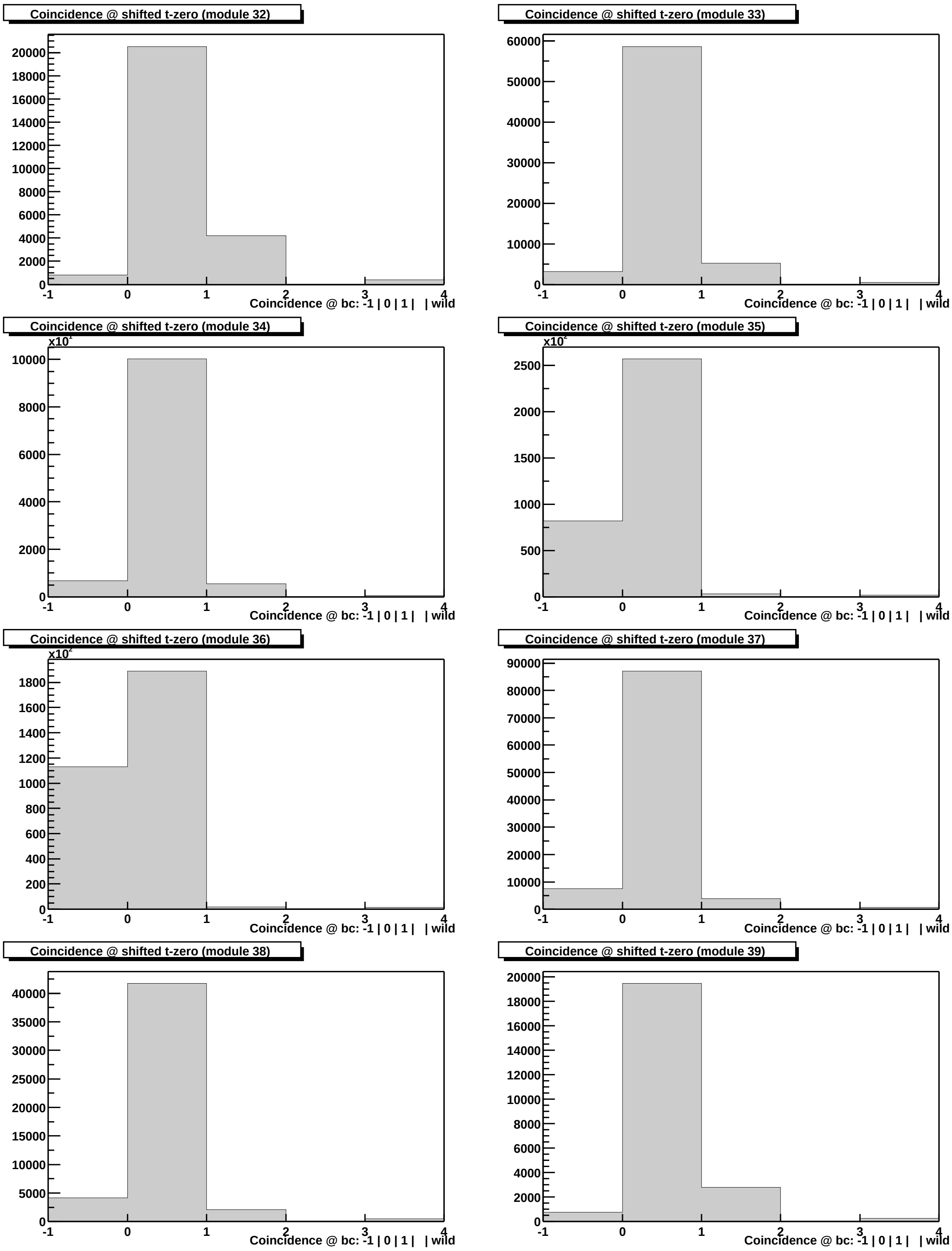

Abbildung C.46: Die Histogramme `ShiftTZero_<Modulnummer>` für die Module 32 bis 39.

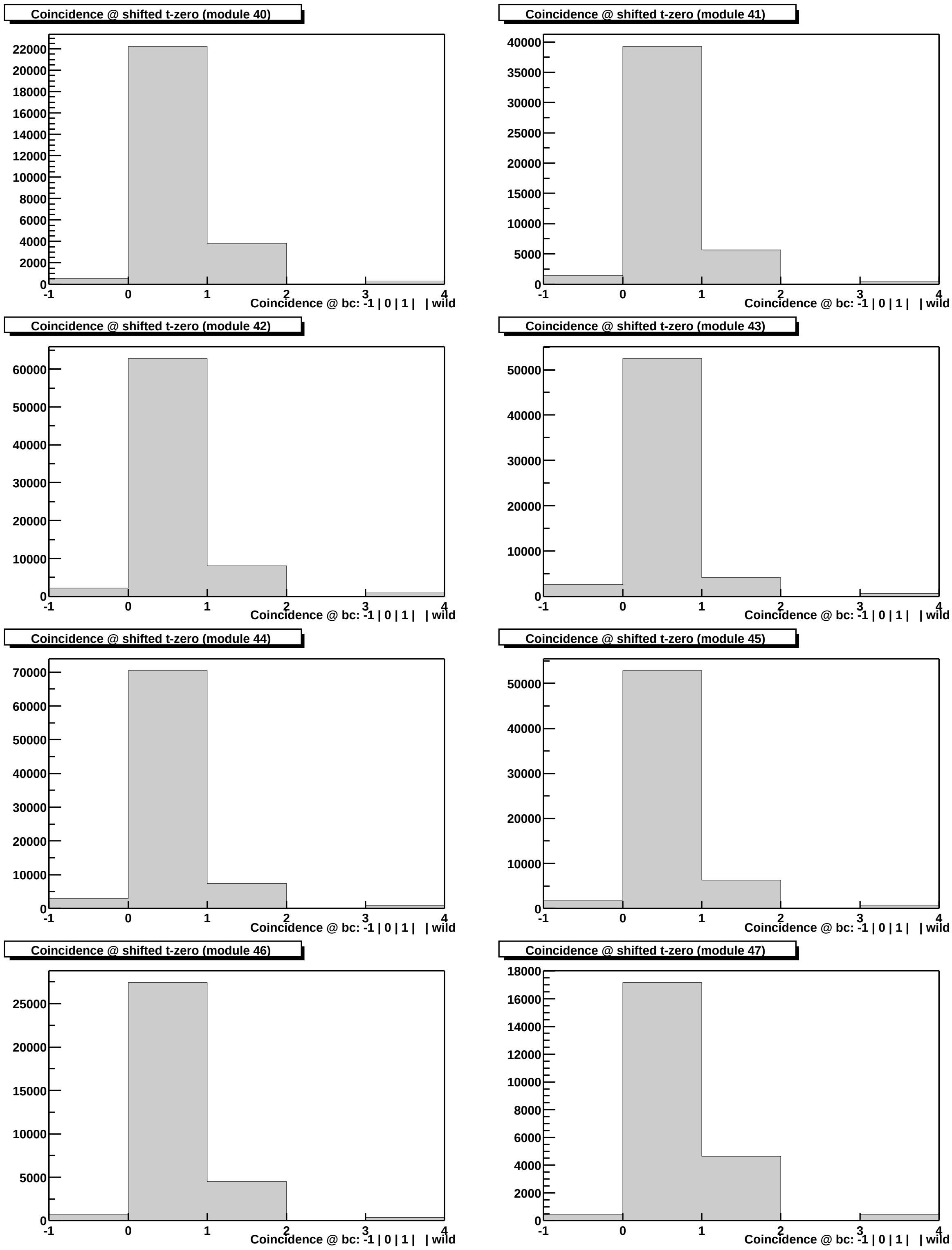

Abbildung C.47: Die Histogramme `ShiftTZero_<Modulnummer>` für die Module 40 bis 47.

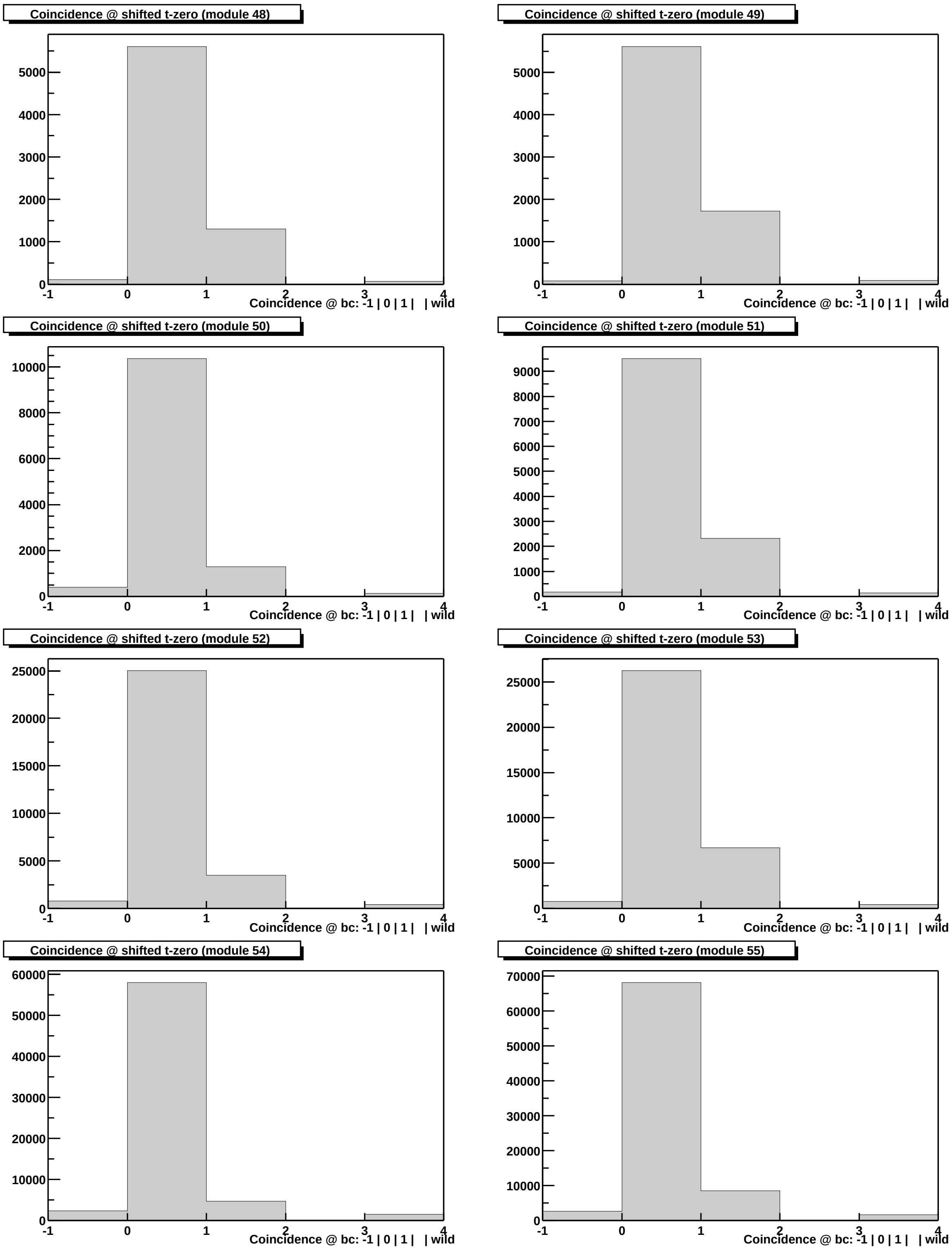

Abbildung C.48: Die Histogramme ShiftTZero_<Modulnummer> für die Module 48 bis 55.

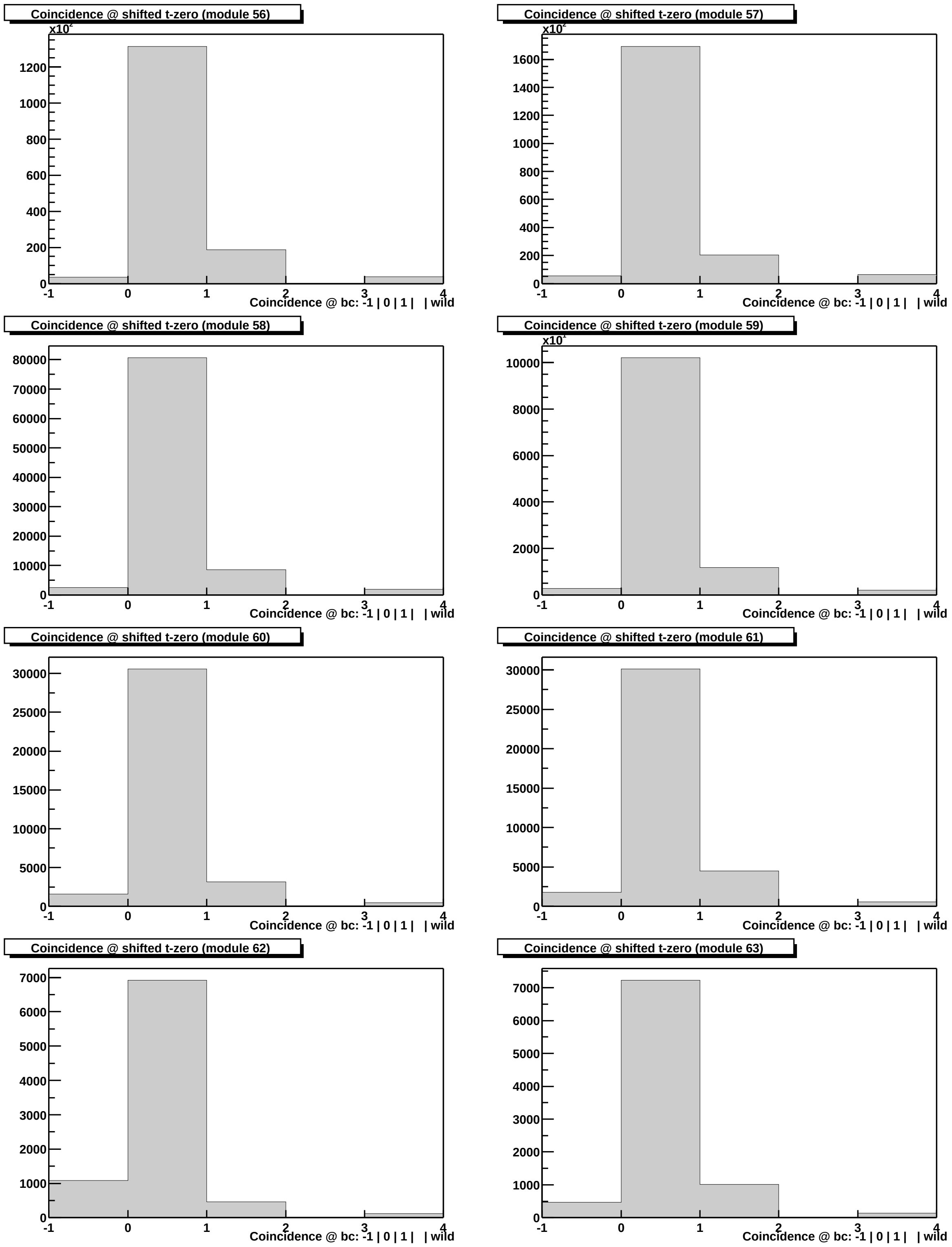

Abbildung C.49: Die Histogramme `ShiftTZero_<Modulnummer>` für die Module 56 bis 63.

C.7 Die zeitliche Verteilung getroffener Drähte

In diesem Abschnitt sind die Histogramme `HitsPerBC_<Modulnummer>` abgebildet. Sie sind mit der ausführenden Klasse `TMuHitsLoop` erzeugt worden und beinhalten jeweils die Gesamtzahl der Treffer im jeweiligen Modul, aufgeschlüsselt nach den Zeitscheiben -1, 0, 1, 2.

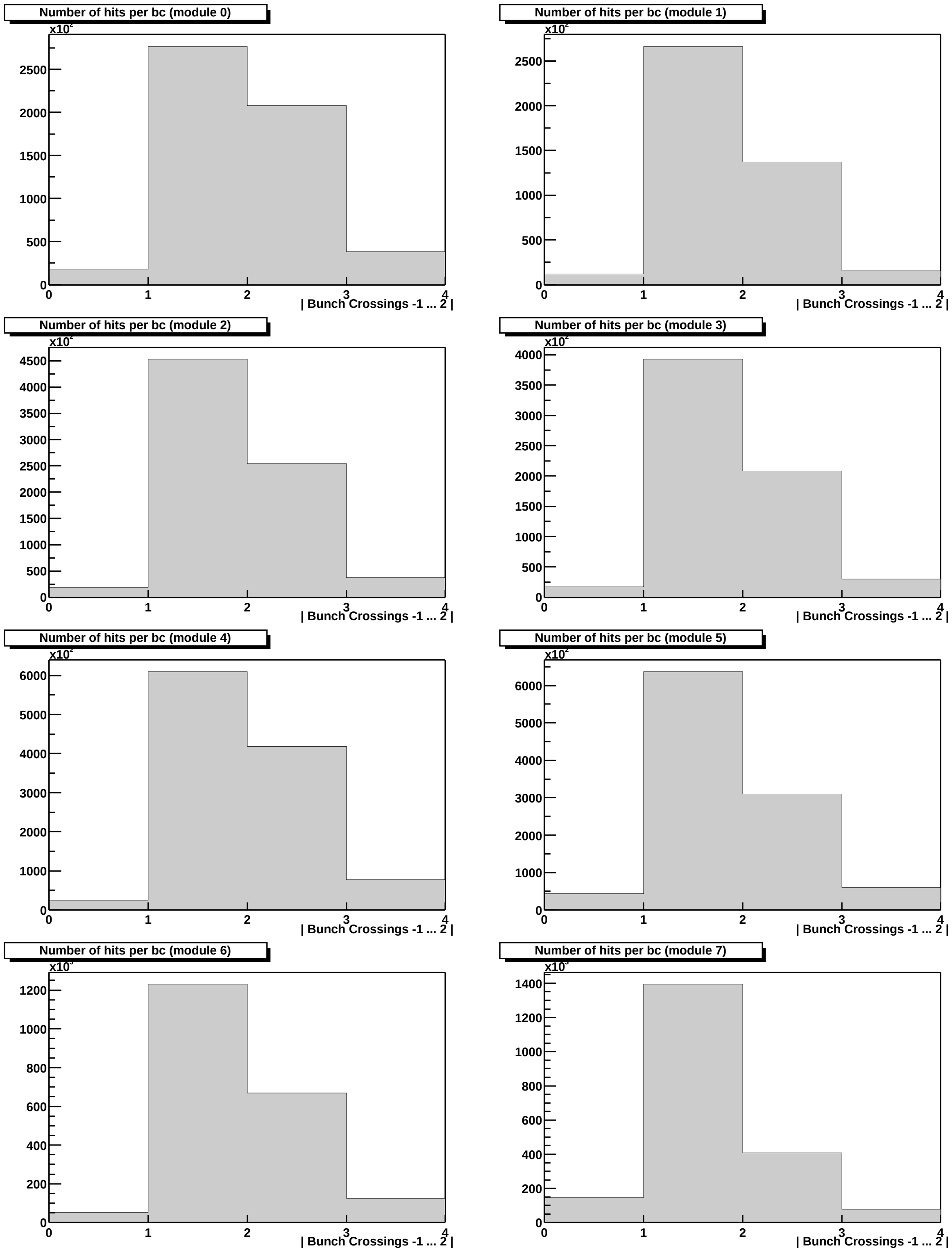

Abbildung C.50: Die Histogramme HitsPerBC_<Modulnummer> für die Module 0 bis 7.

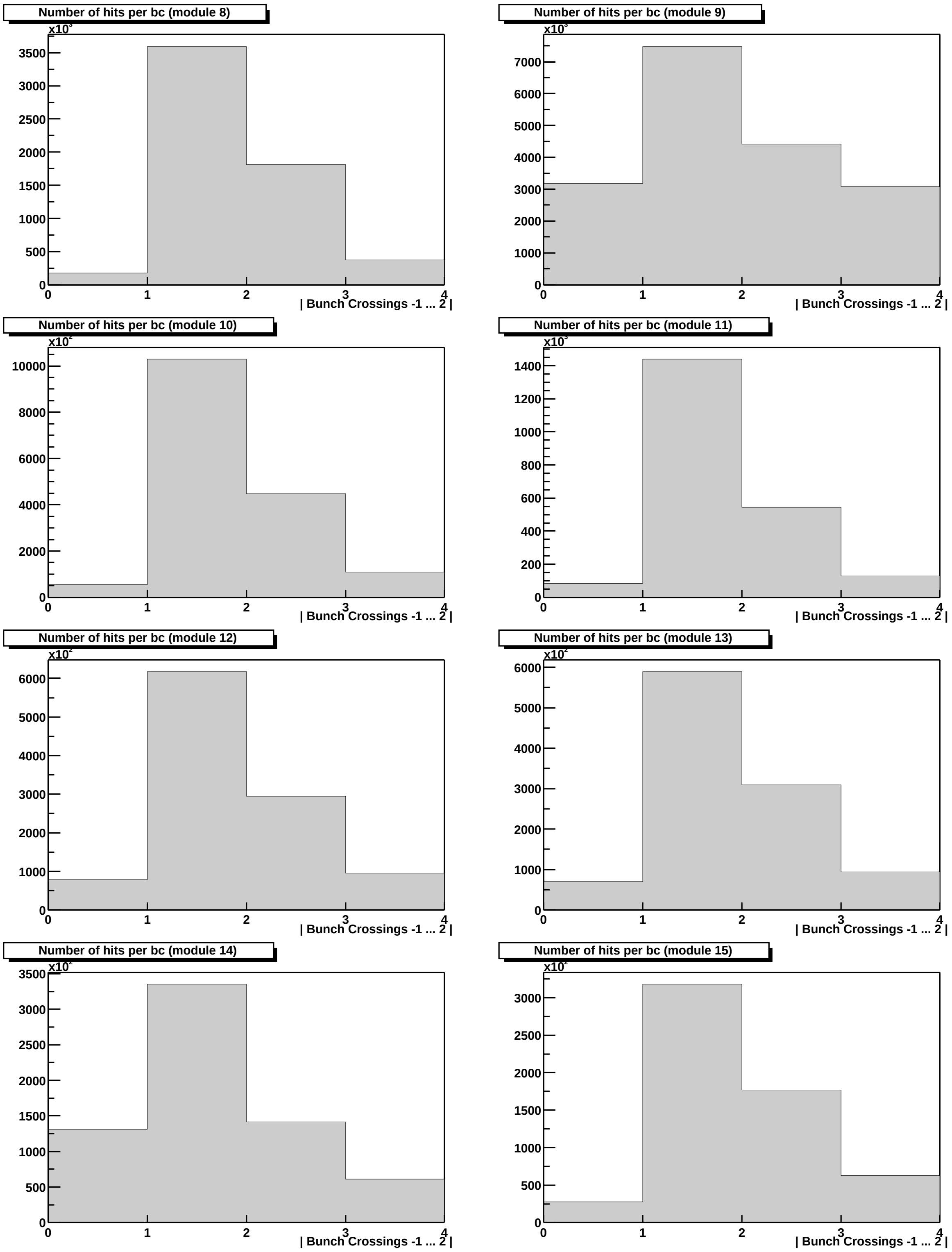

Abbildung C.51: Die Histogramme `HitsPerBC_<Modulnummer>` für die Module 8 bis 15.

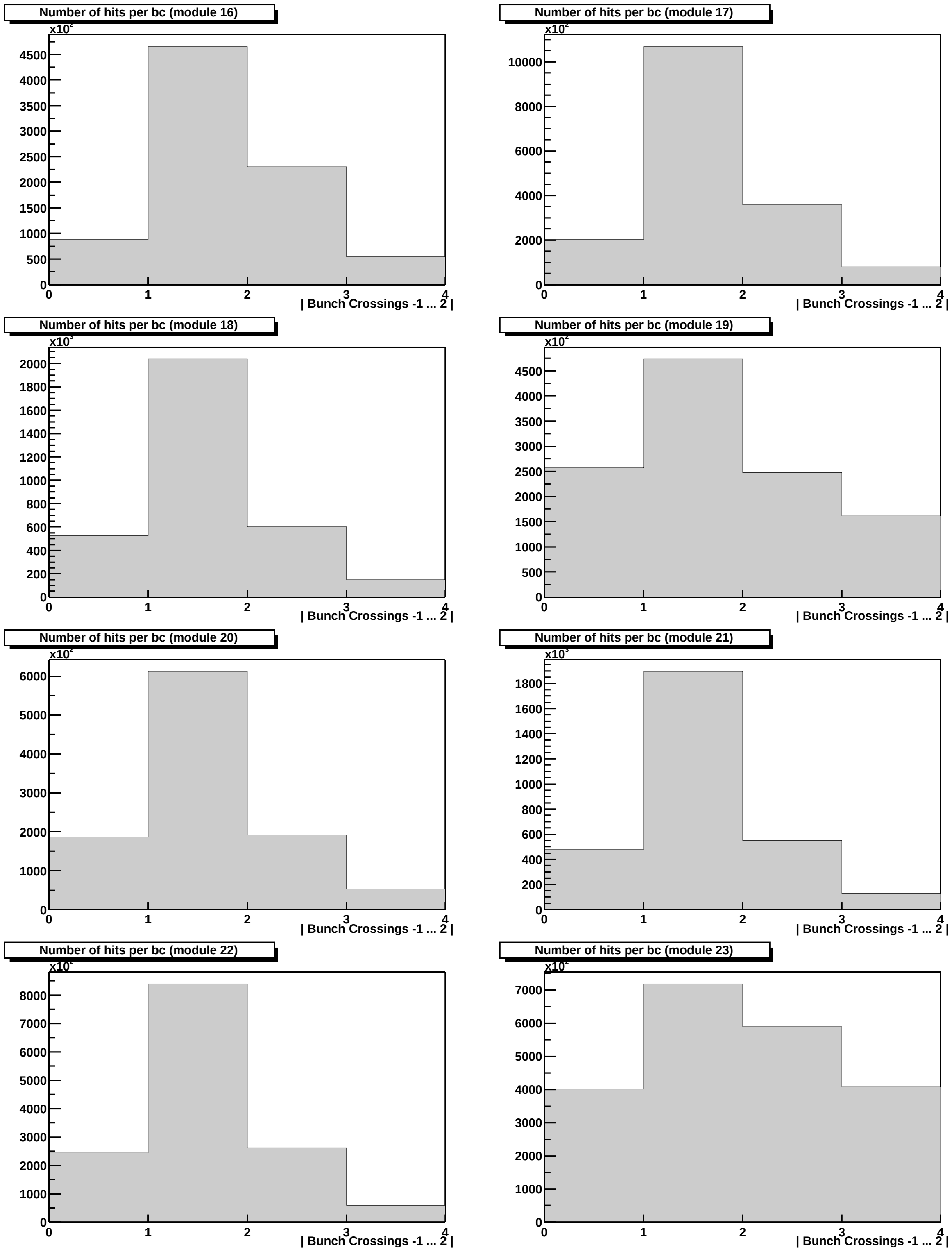

Abbildung C.52: Die Histogramme `HitsPerBC_<Modulnummer>` für die Module 16 bis 23.

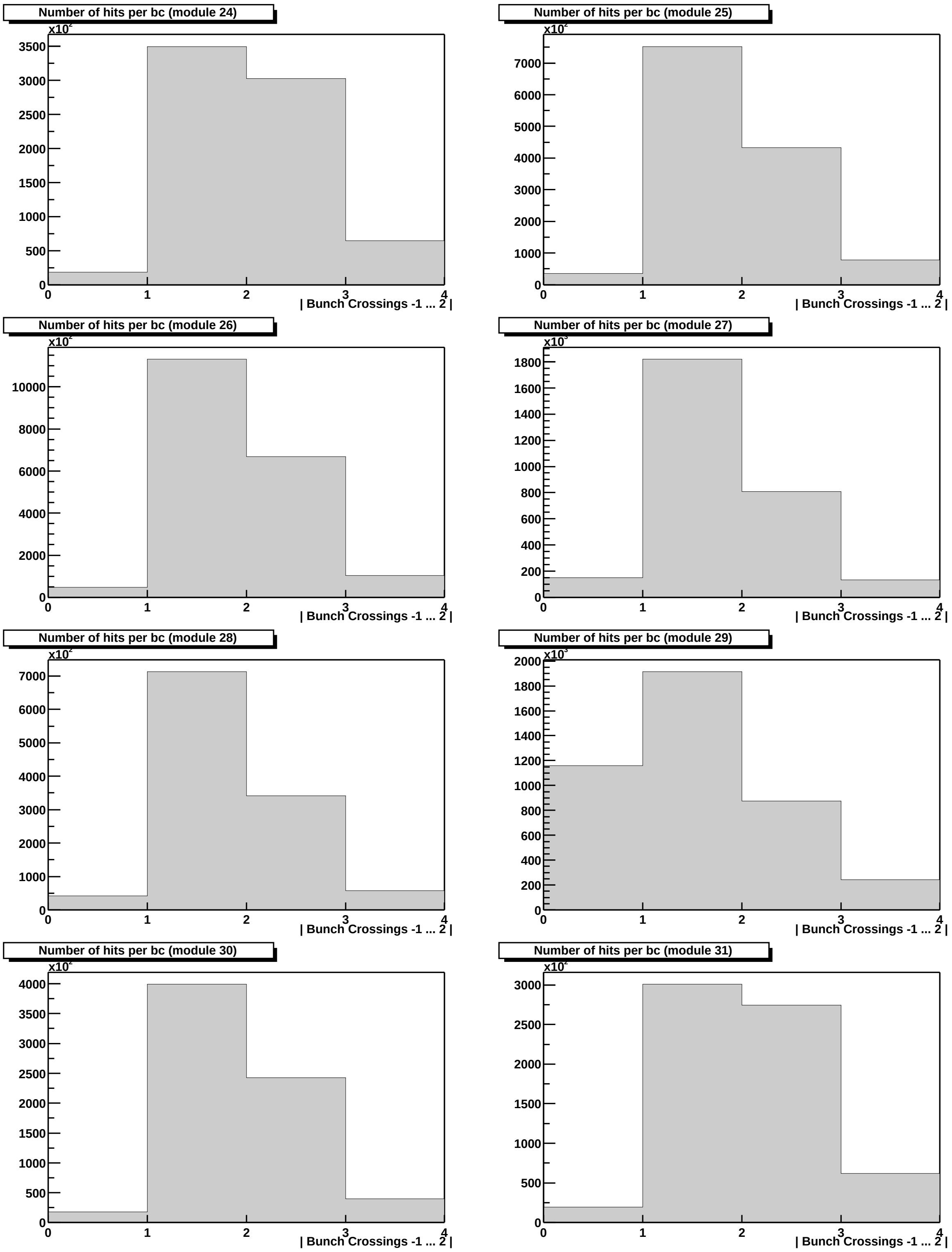

Abbildung C.53: Die Histogramme HitsPerBC_<Modulnummer> für die Module 24 bis 31.

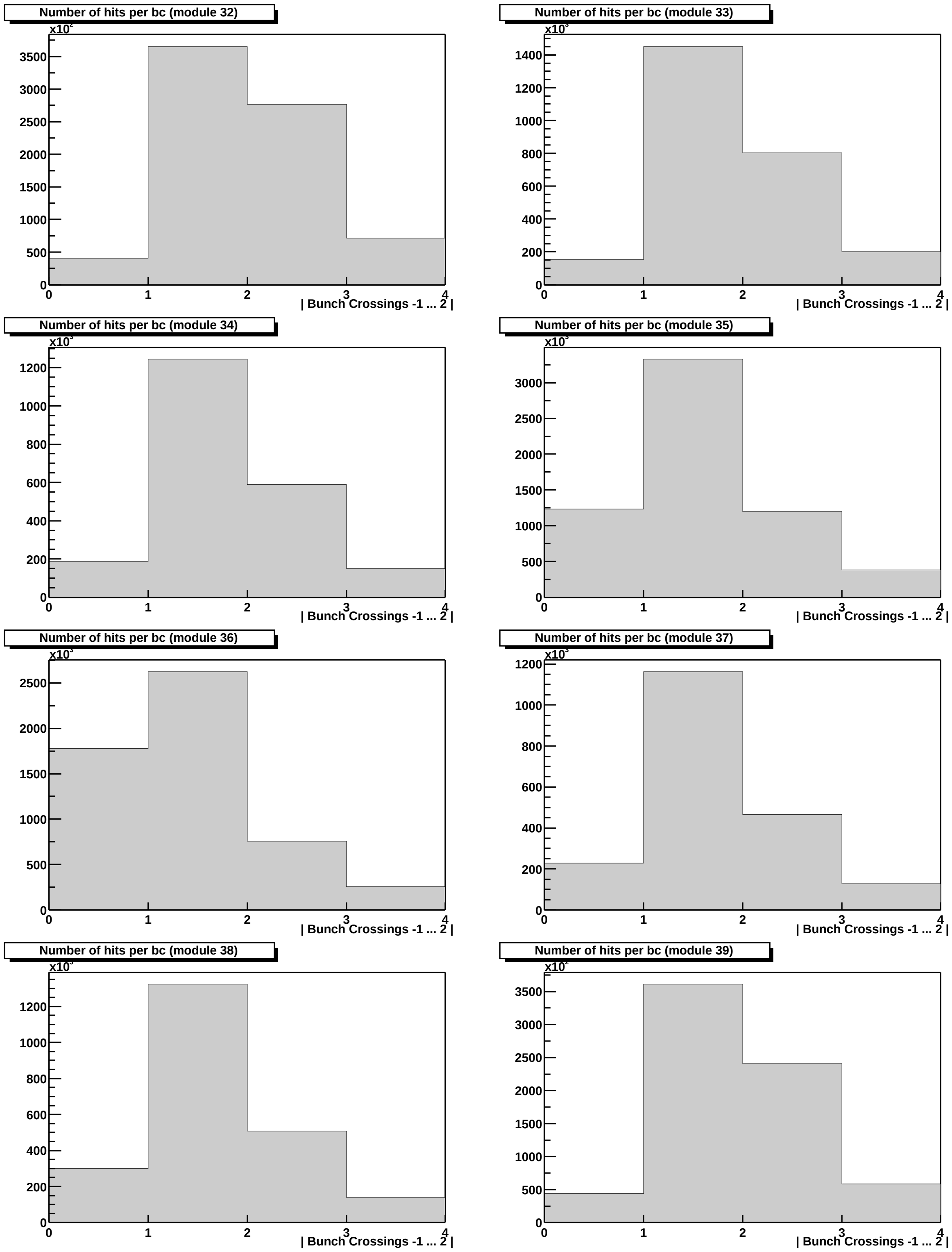

Abbildung C.54: Die Histogramme `HitsPerBC_<Modulnummer>` für die Module 32 bis 39.

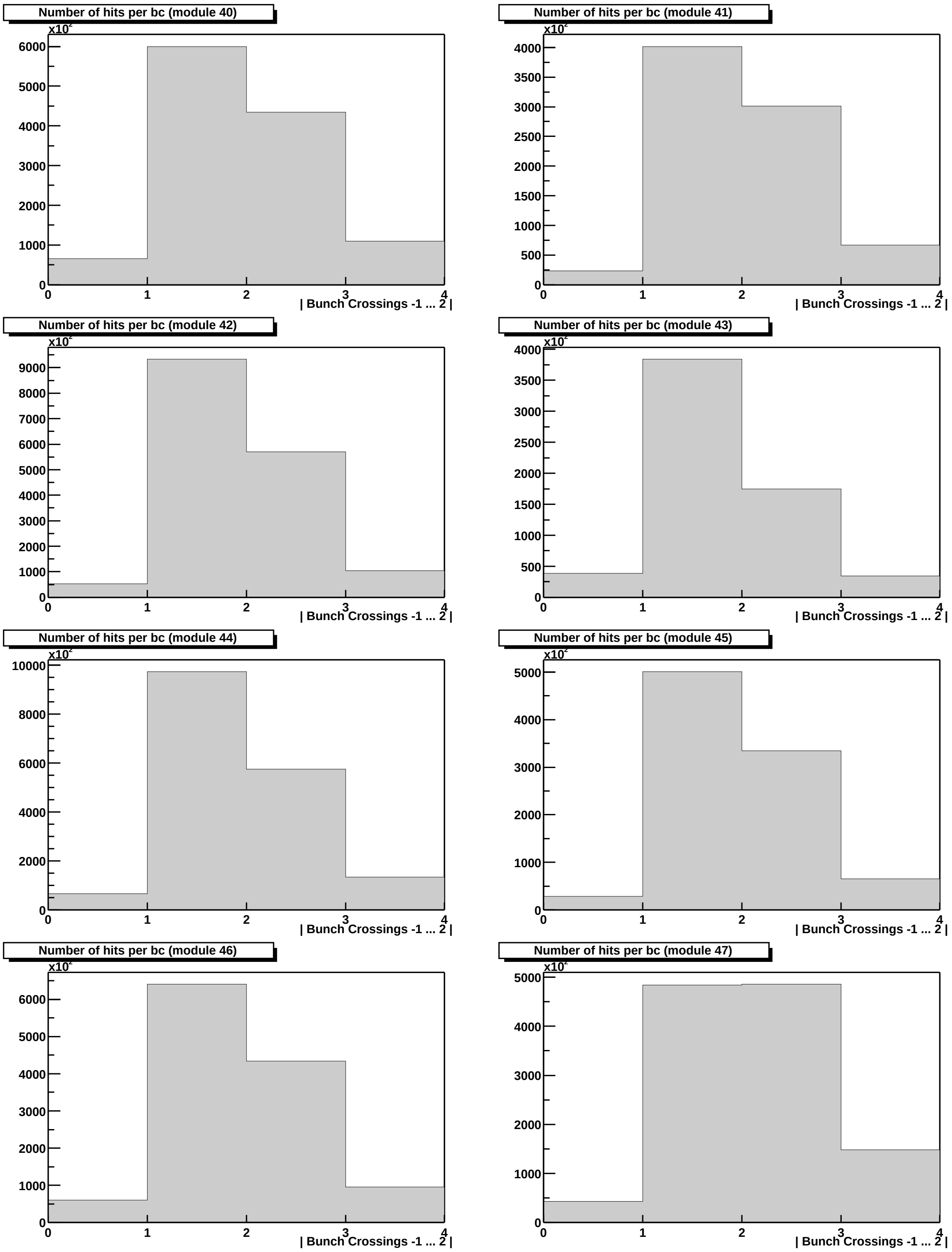

Abbildung C.55: Die Histogramme `HitsPerBC_<Modulnummer>` für die Module 40 bis 47.

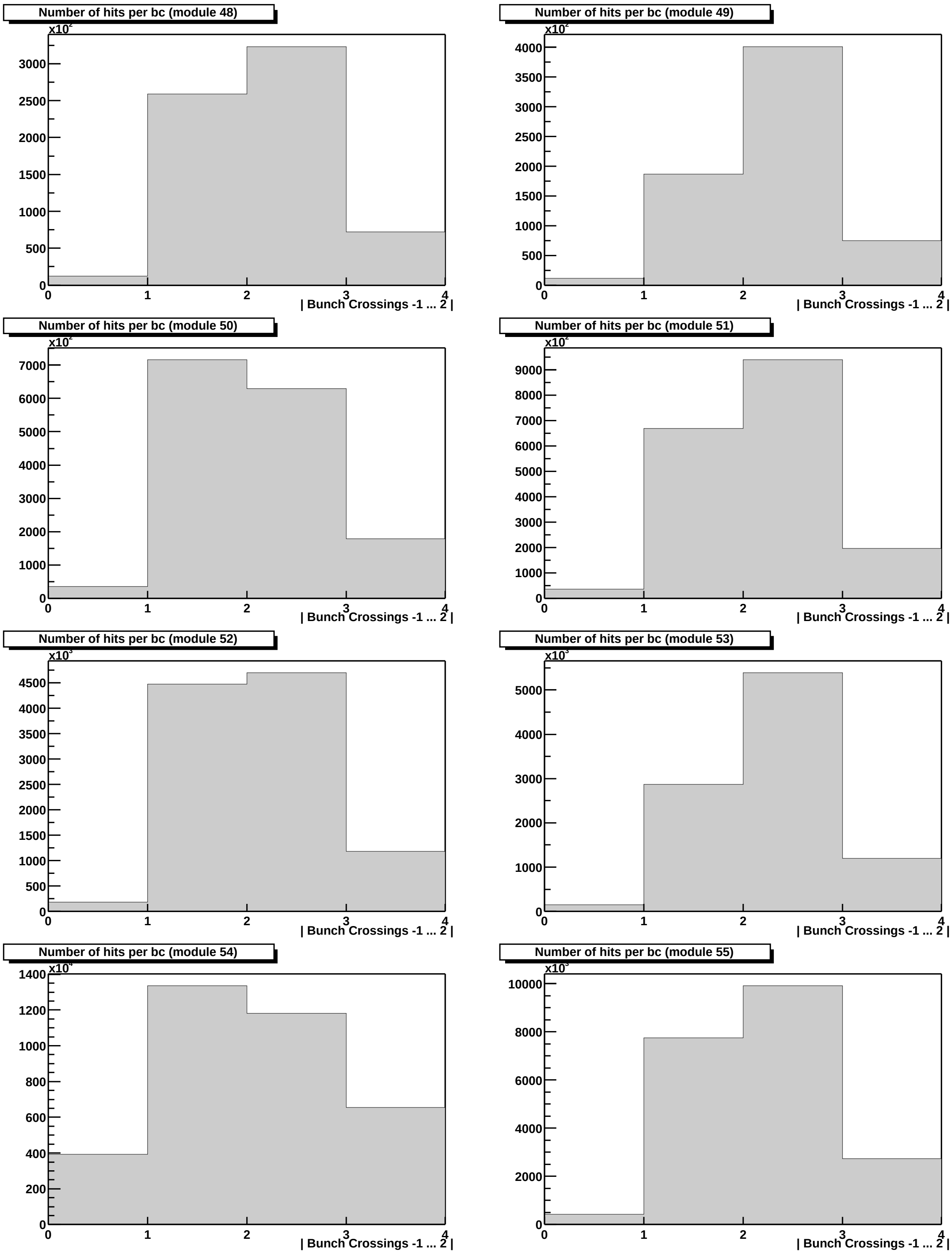

Abbildung C.56: Die Histogramme `HitsPerBC_<Modulnummer>` für die Module 48 bis 55.

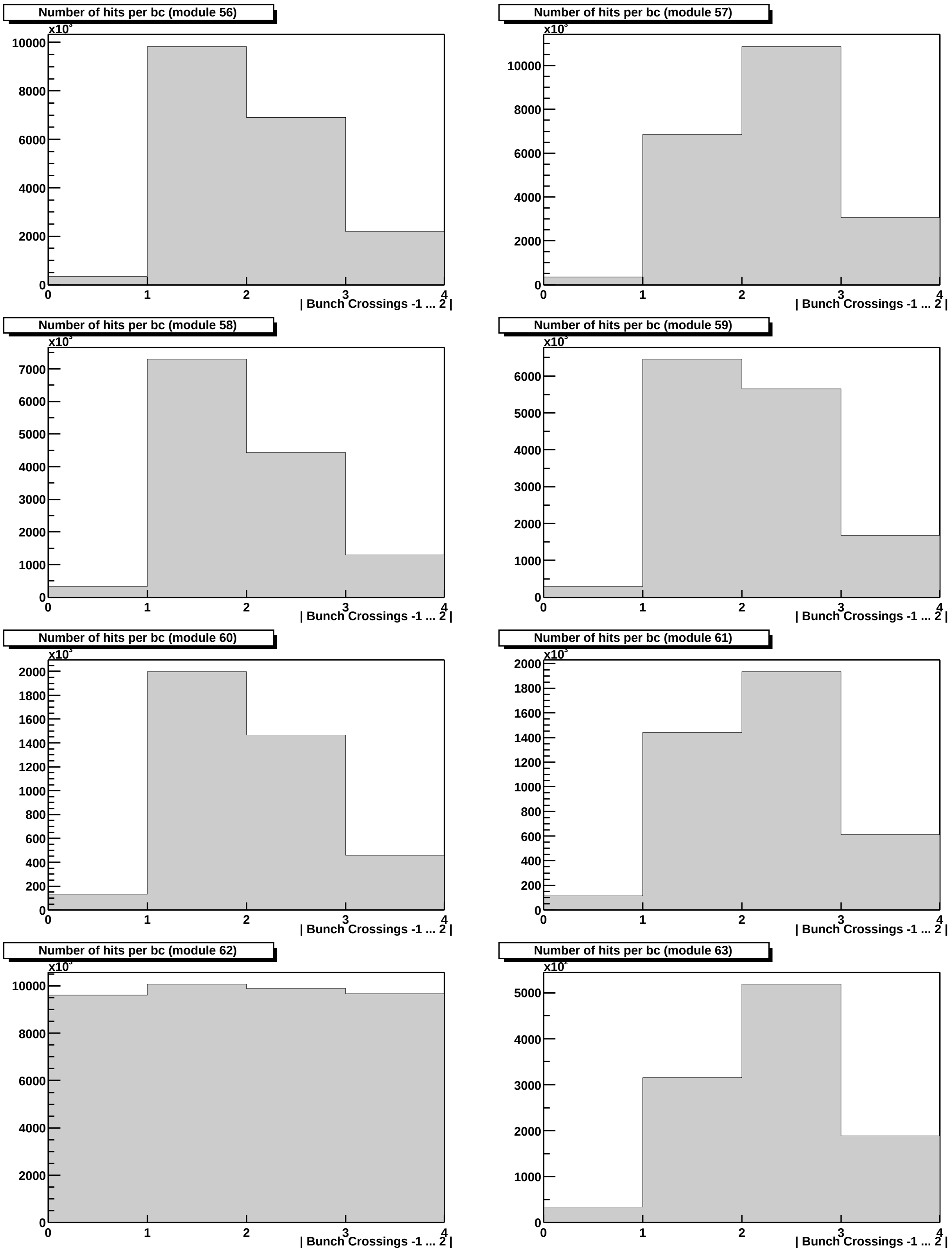

Abbildung C.57: Die Histogramme `HitsPerBC_<Modulnummer>` für die Module 56 bis 63.

Literaturverzeichnis

[1] H1 Collaboration. *The H1 Detector at HERA*. Internal Report, DESY H1-96-01, März 1996.

[2] H. Abramowicz, A. Caldwell. *HERA Collider Physics*. DESY 98-192, März 1999.

[3] H. Cronström et al. *The H1 forward muon spectrometer.* Nucl. Instr. & Meth. **A 340** (1994) 304 - 308.

[4] H. Itterbeck. *Untersuchungen am Myonsystem des H1-Detektors*. RWTH Aachen, PITHA 93/28, Februar 1993.

[5] R. Flachsenberg. *Aufbau einer Testapparatur für die Großserie der Streamerkammer-Detektoren für das H1-Experiment.* RWTH Aachen, April 1989.

[6] Prof. Dr. Braunschweig. Pers. Mitteilung.

[7] B. Koppitz. *ENDCAP-INSTRUMENTATION.* Mai 1990.

[8] B. Koppitz. *Barrel-μ-chambers*. H1-10/90-153, Oktober 1990.

[9] B. Koppitz. *Endcap-μ-chambers*. H1-06/91-182, Juni 1990.

[10] H. Itterbeck. *Techniques and Physics of the Central-μ-Trigger System of the H1-Detector at HERA*. RWTH Aachen, PITHA 97/38, September 1997.

[11] C. Keuker. *The Central Muon Data Acquisition of the H1 Experiment and its Application.* RWTH Aachen, PITHA 98/01, Januar 1998.

[12] C. Keuker, C. Kleinwort, H. Itterbeck, R. Starosta. *Improvement of the trigger timing of the H1 Digital Muon System.* DESY H1-02/95-427, Februar 1995.

[13] J. Tutas. *A Level 1 Trigger from the Limited Streamer Tube system.* DESY H1-07/91-185, Juli 1991.

[14] *The H1 Central Muon Detector Homepage.*
`http://www-h1.desy.de:80/h1/www/h1det/muon/c_muon/digmuhome.html`

[15] J. K. Köhne at al. *Realization of a second level neural network trigger for the H1 experiment at HERA.* Nucl. Inst. & Meth. **A 389** 128 (1997).

[16] R. Prosi. *Pedestrian's Guide to the L4 Filter Farm.* DESY H1-IN-412, November 1994.

[17] *The KOOL Homepage.*
`http://www-h1.desy.de:80/h1/iww/icas/imanuals/kool/kool.html`

[18] R. Brun, F. Rademakers. *ROOT Architectural Overview.* Mai 1996.
`http://root.cern.ch/root/Architecture.html`

[19] T. Hadig. *Bos2OOP program manual including: Description of H1 Data Format, Version 0.0δ.* November 2000.
`http://www-h1.desy.de/~hadig/www/projects/bos2root/`

[20] *The Future H1 ROOT Analysis Framework.*
`http://www-h1.desy.de:80/icas/oop/`

[21] V. Blobel et al. *FPack - F-package for input/output.* September 1994.
`http://www-h1.desy.de:80/icas/imanuals/fpack/`

[22] J. Olzem. *Muon Rawdata Analysis with ROOT.* September 2000.
`http://www-h1.desy.de/~olzem/TMuAnalysis.html`

[23] C. Caso et al. The European Physical Journal **C3** (1998) 1.

[24] O. C. Allkofer. *Introduction to Cosmic Radiation.* Thiemig 1975.

[25] B. Leißner. Dissertation in Vorbereitung.

[26] C. Kleinwort. Pers. Mitteilung.

Danksagung

Abschließend möchte ich mich bei all jenen bedanken, die zum Gelingen dieser Arbeit beigetragen haben, ganz besonders

bei Prof. Dr. Berger für die Möglichkeit, am H1-Experiment aktiv mitarbeiten zu können, und die vertrauensvolle Unterstützung während der gesamten Zeit.

bei Prof. Dr. Braunschweig für die Übernahme des Koreferats und die überaus hilfreiche Beratung in den technischen Aspekten des H1-Experimentes.

bei Boris Leißner für die umfangreiche Betreuung, von den allerersten Schritten in C++ bis hin zur *H1-Virtuosität*. Es war bestimmt nicht einfach, aber immer eine Anleitung zu professionellem Arbeiten.

bei Jürgen Scheins für viele tiefe Einblicke in das Wesen des Myon-Systems und die Einweisung in die fortgeschrittene Skript-Programmierung, bei Thomas Hadig, ohne dessen Hilfe ich vor so manchen Softwareproblemen definitiv hätte kapitulieren müssen, und auch bei Dr. Hans-Ulrich Martyn, Dr. Klaus Rabbertz, Sascha Caron, Thomas Kluge und Claus Kleinwort für ihre Hilfsbereitschaft.

bei Herrn Kukulies, Herrn Braun und bei Nina Loktionova für die Zähmung der widerspenstigen Rechner. Es hätten durchaus einige freie Tage werden können.

bei meinen Mit-Diplomanden und allen Mitarbeitern des I. Physikalischen Instituts generell. Besonders Roman Adolphi, Anja Vest, Ingo Strauch, Gilles Frising, Adil Aktas, Arno Heister, Dr. Frank Raupach, Roland Wagner, Niels v. Eldick, Dr. Thomas Kirn, Martin Hennecke und auch alle anderen waren und sind für die Atmosphäre verantwortlich, die das Büro zum Erstwohnsitz machen kann, ob in Hamburg oder in Aachen.

bei meinen Eltern, die vieles erst möglich gemacht haben.