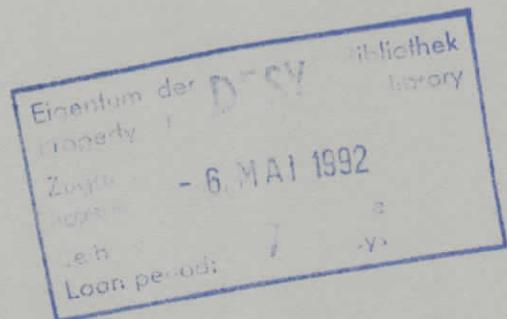


Interner Bericht
DESY F35D-92-01
April 1992

Überwachung und Auswertung des Datenflusses im ZEUS „Eventbuilder“

von

T. Schlichting



DESY behält sich alle Rechte für den Fall der Schutzrechtserteilung und für die wirtschaftliche Verwertung der in diesem Bericht enthaltenen Informationen vor.

DESY reserves all rights for commercial use of information included in this report, especially in case of filing application for or grant of patents.

**"Die Verantwortung für den Inhalt dieses
Internen Berichtes liegt ausschließlich beim Verfasser"**

Überwachung und Auswertung des Datenflusses im ZEUS „Eventbuilder“

**Diplomarbeit am Fachbereich Physik
der Universität Hamburg**

**vorgelegt von
Thorsten Schlichting ✓**

April 1992



Inhaltsverzeichnis

Einleitung	6
1 Das ZEUS Experiment	7
1.1 Der Aufbau des ZEUS-Detektors	7
1.2 Das ZEUS Trigger- und Datenerfassungssystem	9
1.2.1 Die Komponentensubsysteme	11
1.2.2 Die erste Triggerstufe	12
1.2.3 Die zweite Triggerstufe	12
1.2.4 Der Eventbuilder	12
1.2.5 Die dritte Triggerstufe	12
1.2.6 Die Datenaufzeichnung	12
1.2.7 Die zentrale Steuerung	13
2 Der Eventbuilder	14
2.1 Aufgaben und Anforderungen an den Eventbuilder	14
2.2 Die Funktionseinheiten des Eventbuilders	18
2.3 Der Aufbau des Eventbuilders	20
2.3.1 Der GSLT-Verteiler	20
2.3.2 Die Komponentenschnittstellen	21
2.3.3 Die TLT-Schnittstellen	22
2.3.4 Der Kreuzschienenverteiler	22
2.3.5 Die Steuereinheit	23
3 Möglichkeiten der Echtzeitüberwachung und Analyse des Eventbuilder-Betriebes	27
3.1 Zeitmessung in Transputernetzen	28
3.2 Kenngrößen externer Systeme	28
3.3 Kenngrößen der internen Funktionseinheiten	32
3.4 Analyse von Kenn- und Überwachungsgrößen	33
3.4.1 Echtzeitüberwachung	33
3.4.2 Statistiken über den Eventbuilderbetrieb	35
3.4.3 Leistungsoptimierung des Eventbuilders	35
3.4.4 Leistungsgrenzen des DAQ-Systems	37
4 Verhalten des Eventbuilders im Probetrieb des ZEUS-Detektors	38
4.1 Kosmische Strahlung	38
4.2 Ergebnisse des Funktionstests des ZEUS-Detektors	40
4.3 Ergebnisse der Eventbuilderüberwachung	42
4.3.1 Implementation der Kenngrößenmessung	42
4.3.2 Eichung der Zeitmesser im Transputernetz	45
4.3.3 Der Ausbaustand des Eventbuilders im Testbetrieb	46
4.3.4 Messungen am Eventbuilder während des Testbetriebes	47
4.4 Leistungsanalyse und Ermittlung von Grenswerten	56

5 Zusammenfassung	65
A Der Server	66
A.1 Funktionsprinzip des expert-Programms	66
A.2 Funktionsprinzip des Mitt-Programms	68
B Danksagung	69
C Erklärung	70

Abbildungsverzeichnis

1.1	Längsschnitt durch den ZEUS-Detektor	8
1.2	Zeitliche Zusammenhänge im ZEUS-Triggersystem	9
1.3	Vereinfachte Darstellung des Datenflusses im ZEUS-Trigger- und Datenerfassungssystem	10
2.1	Einbettung des Eventbuilders in seine Systemumgebung	15
2.2	Die Funktionseinheiten des Eventbuilders und deren Zusammenspiel	18
2.3	Der GSLT-Verteiler	20
2.4	Das 2TP-VME-Modul als Komponentenschnittstelle	21
2.5	Das 2TP-VME-Modul als TLT-Schnittstelle	22
2.6	Der Kreuzschienenverteiler	23
2.7	Die Steuereinheit	24
3.1	Beispiel für eine Textmeldung an der Systemspezialistenschnittstelle	35
3.2	Länge der Ereignisteildaten von GSLT und BMUO, gemessen während Run 1033	36
4.1	Differentielles Impulsspektrum kosmischer Myonen auf Meereshöhe [HAY62]	39
4.2	Graphische Darstellung eines rekonstruierten Ereignisses	41
4.3	Die Struktur der Überwachungsinformationen	43
4.4	Einbettung der Überwachungstransputer in den Eventbuilder	44
4.5	Zeiteichung der Transputer im Eventbuilder	45
4.6	Die zeitliche Entwicklung der GSLT-Frequenz $f_{T_{rig}}$ für Run 1033	48
4.7	Die zeitliche Entwicklung von $\Delta t_{T_{rig}}$	48
4.8	Die Verteilung von $\Delta t_{T_{rig},n}$	49
4.9	Die Verteilung der Abstände der positiven Triggerentscheidungen des GSLT $\Delta t_{T_{rig},n}$ die kleiner als 20 ns waren für Run 1033	49
4.10	Fehlermeldung Triggerentscheidung zu langsam abgenommen	50
4.11	Verteilung von d_K für die am Run #1033 beteiligten Komponenten	51
4.12	Antwortzeit von BMUO in Abhängigkeit der Ereignisteildatenlänge in Run #1033	52
4.13	Der Abstand Δt_K gegen den Abstand $\Delta t_{T_{rig}}$	53
4.14	Zeitlicher Verlauf der Füllstände von Puffern im Eventbuilder	54
4.15	Das Format der Daten- und Quittierungspakete auf einem Transputerlink	56
4.16	Schematische Darstellung der Übertragungsstrecken für Ereignisteildaten im Eventbuilder	57
4.17	Transfergeschwindigkeit auf den Datenlinks in Abhängigkeit der Kabellänge zwischen Kreuzschienenverteiler und Komponentenschnittstellen	57
4.18	Maximalfrequenz einer Komponentenschnittstelle in Abhängigkeit der Datenlänge	59
4.19	Mittlere Unbenutztheit von Links einer Komponentenschnittstelle	60
4.20	Mittlere Zeit, die die Links der Komponentenschnittstelle mit den längsten Ereignisteildaten ungenutzt blieben	61
4.21	Verteilungen der mittlere Zeit, die die Links der Komponentenschnittstelle mit den längsten Ereignisteildaten im Mittel ungenutzt blieben für verschiedene Datenlängen	62
4.22	Abstand der ersten beiden Maxima in der Verteilung der Zeit, die die Links der Komponentenschnittstelle mit den längsten Ereignisteildaten ungenutzt blieben in Abhängigkeit von der Datenlänge	63

4.23	Verlauf der minimalen Zeit, die die Links der Komponentenschnittstelle mit den längsten Ereignisteildaten unbenutzt blieben	63
4.24	Rate, mit der die Komponentenschnittstelle mit den längsten Ereignisteildaten ihre Daten liefern konnte in Abhängigkeit der Datenlänge	64
A.1	Die Proseßkonfiguration des Eventbuilderservers	67

Tabellenverzeichnis

1.1 Übersicht über die verschiedenen Komponentensubsysteme des ZEUS-Experiments . . .	11
2.1 Länge der Ereignisteildaten einzelner DAQ-Komponenten	16
3.1 Symbolverzeichnis von Kenngrößen externer Systeme	31
3.2 Symbolverzeichnis von Kenngrößen der internen Funktionseinheiten des Eventbuilders	34
4.1 Eichkonstanten für die Zeitmessung im Eventbuilder-Transputernets	47
4.2 Längen der einzelnen Ereignisteildaten von Run 1033	50
4.3 Signalverzögerungen auf den Übertragungsstrecken des Eventbuilders.	56

Einleitung

Die vorliegende Arbeit wurde im Rahmen des ZEUS-Experiments, welches am HERA-Speicherring durchgeführt wird, angefertigt. Der ZEUS-Detektor ist ein Universaldetektor. Eine große Anzahl von Spurkammern und Szintillationszählern liefern Daten auf etwa 250.000 Zählerkanälen. Da alle 96ns Teilchenpakete des HERA-Speicherringes innerhalb des Erfassungsbereichs des Detektors zur Kollision gebracht werden, kann eine enorme Datenmenge auftreten, die höchste Anforderungen an das Trigger- und Datenerfassungssystem stellt. Aus diesem Grund wird beim ZEUS-Experiment in großem Maße Parallelverarbeitung eingesetzt. So besteht der Detektor aus einer Reihe von Komponenten, z. B. der zentralen Spurkammer oder dem hochauflösenden Kalorimeter, die unabhängig voneinander die ihnen zugeordneten Kanäle auszulesen vermögen. Um die Datenmenge durch Auswahl physikalisch interessierender Reaktionen zu verringern, wird ein dreistufiges Triggersystem verwendet. In den ersten beiden Stufen werden die Daten der einzelnen Komponenten getrennt untersucht. Erst die dritte Triggerstufe benötigt zur Untersuchung das gemeinsame Vorliegen der Daten aller Komponenten. Deshalb werden die Daten von den verschiedenen Detektorkomponenten nach der zweiten Triggerstufe dem sogenannten „Eventbuilder“ übermittelt. Der Eventbuilder hat die Aufgabe, die Daten der verschiedenen Komponenten, welche zu demselben Ereignis gehören, zusammenzuführen und sie an die dritte Triggerstufe, eine Rechnerfarm, weiterzuleiten. Die Anforderungen an den Eventbuilder sind sehr hoch: Die erwartete Triggerrate hinter der zweiten Triggerstufe beträgt 100 Hz. Der erwartete Datenfluß durch den Eventbuilder beträgt damit etwa 100 Ereignisse/s, was bei einer mittleren Datenmenge von 143 KByte/Ereignis einem Datenfluß von 14,0 MByte/s entspricht. Die vorliegende Arbeit ist der näheren Untersuchung und Optimierung des Eventbuilderbetriebes gewidmet. Im nachfolgenden Kapitel wird zunächst ein Überblick über den Aufbau des ZEUS-Detektors und seines Trigger- und Datenerfassungssystems gegeben. Daran schließt sich ein Kapitel über den Aufbau des Eventbuilders an. Im weiteren Verlauf der Arbeit werden dann Konzepte zur Echtzeitüberwachung und Systemanalyse des Eventbuilders untersucht, bevor Beobachtungen während der Inbetriebnahme und des Probetriebes des Detektors vorgestellt und ausgewertet werden.

Kapitel 1

Das ZEUS Experiment

Das ZEUS-Experiment ist eines der beiden großen Experimente am Elektron-Proton-Speicherring HERA des Deutschen Elektronen Synchrotrons DESY in Hamburg. Der ZEUS-Detektor und sein Datenerfassungssystem (DAQ, Data acquisition) werden im folgenden vorgestellt.

1.1 Der Aufbau des ZEUS-Detektors

Der Detektor hat eine Abmessung von $11,6\text{ m} \times 10,8\text{ m} \times 20,0\text{ m}$ und ein Gesamtgewicht von 3600 t. Abbildung 1.1 zeigt einen Längsschnitt. Vom Wechselwirkungspunkt nach außen betrachtet erkennt man folgende Detektorkomponenten:

- Direkt am Strahlrohr befindet sich der Vertexdetektor (VXD). Es handelt sich um eine kleine Zeit-Expansionskammer, die dem Nachweis kurzlebiger Teilchen dient. Ihre räumliche Auflösung beträgt $30\mu\text{m}$.
- Die zentrale Spurkammer (CTD, Central Tracking Detector) besteht aus einer zylindrischen Driftkammer mit einem äußeren Radius von 85 cm und einer Gesamtlänge von 240 cm . Mit insgesamt 4608 Signaldrähten wird radial eine Auflösung von $100\mu\text{m}$ erreicht. Die Auflösung in Strahlrichtung liegt zwischen $1,0\text{ mm}$ und $1,4\text{ mm}$.
- Eine supraleitende Spule (COIL) mit einem Durchmesser von $1,7\text{ m}$ umgibt den CTD und den VXD und erzeugt ein für eine Impulsbestimmung benötigtes Magnetfeld von $1,8\text{ T}$. Um die Wechselwirkung mit Reaktionsprodukten minimal zu halten, besteht sie aus Aluminium mit einer NbTi/Cu-Einlage und ist nur $0,9$ Strahlungslängen dick.
- Eine weitere supraleitende Spule (COMPENSATOR) kompensiert den Einfluß der supraleitenden Spule auf den Speicherring.
- Entlang der Strahlachse folgen die Vorwärts- und Rückwärtsspurkammern (FTD, Forward Tracking Detector; RTD, Rear Tracking Detector). Dies sind ebene Driftkammern, die Teilchen in vom CTD nicht abgedeckten Winkelbereichen erfassen. Sie haben eine Ortsauflösung von $120\mu\text{m}$.
- Zwischen den drei Driftkammern, die den FTD bilden, befinden sich zwei Übergangsstrahlungsdetektoren (TRD, Transition Radiation Detector). Sie dienen der Unterscheidung von Hadronen und Elektronen.

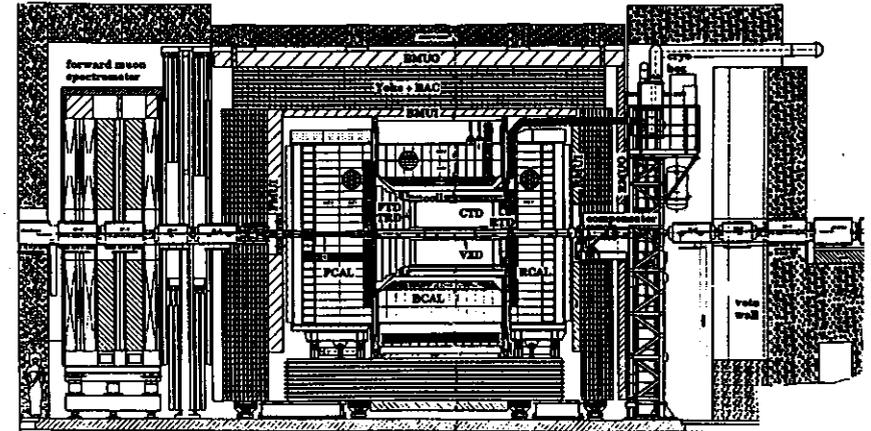


Abbildung 1.1: Längsschnitt durch den ZEUS-Detektor

- Ein hochauflösendes Uran-Szintillator Sandwichkalorimeter umgibt den zentralen Detektorbereich. Es erreicht für Elektronen eine Energieauflösung von $\frac{\sigma_E}{E} = \frac{18}{\sqrt{E}}\%$, für Hadronen $\frac{\sigma_E}{E} = \frac{35}{\sqrt{E}}\%$ und ist in drei Sektionen unterteilt: FCAL (Forward Calorimeter), BCAL (Barrel Calorimeter) und RCAL (Rear Calorimeter). Die von ihnen erreichte Raumwinkelabdeckung liegt über $99,5\%$ und die Emissionswinkel von Jets können mit einer Genauigkeit von besser als 10 mrad bestimmt werden. Zur besseren Unterscheidung zwischen Elektronen und Hadronen sind in das Kalorimeter in einer Tiefe von 3 Strahlungslängen die Siliziumdioden des Hadron Elektron Separators (HES, auf der Abbildung nicht zu erkennen) eingelassen. Die Hadronen-Mißidentifikationsrate beträgt weniger als 10^{-3} bei 20 GeV .
- Das Kalorimeter ist vom Eisenjoch (YOKE) umgeben, das aus Eisenplatten besteht, zwischen denen sich die Spurkammern des Backing Calorimeter (BAC) befinden. Hier werden Teilchen nachgewiesen, die im Uran-Kalorimeter nicht absorbiert wurden. Die Energieauflösung beträgt $\frac{\sigma_E}{E} = \frac{100}{\sqrt{E}}\%$ bei einer Ortsauflösung von etwa 1 mm .
- Vor und hinter dem Eisenjoch befinden sich Spurkammern, die dem Nachweis von Myonen dienen (BMUO, BMUI, RMUO, RMUI Barrel (Rear) Myon inner (outer) detector). Da das Eisen des Jochs mit Kupferspulen auf $1,6\text{ T}$ erregt wird, ist mit diesen Kammeren eine Impulsbestimmung möglich. Die Auflösung beträgt $\frac{\sigma_x}{p} < 30\%$ bei 20 GeV .
- In Protonenrichtung folgt das Vorwärts-Myonen-Spektrometer. Es besteht wie BMUI und RMUI aus einer inneren Driftkammer vor dem Eisenjoch (FMUI), außerhalb aber aus einem Driftkammersystem mit dazwischenliegenden torusförmig-magnetisierten Eisenplatten, womit es eine Auflösung von $\frac{\sigma_x}{p} < 23\%$ bei 100 GeV erreicht.

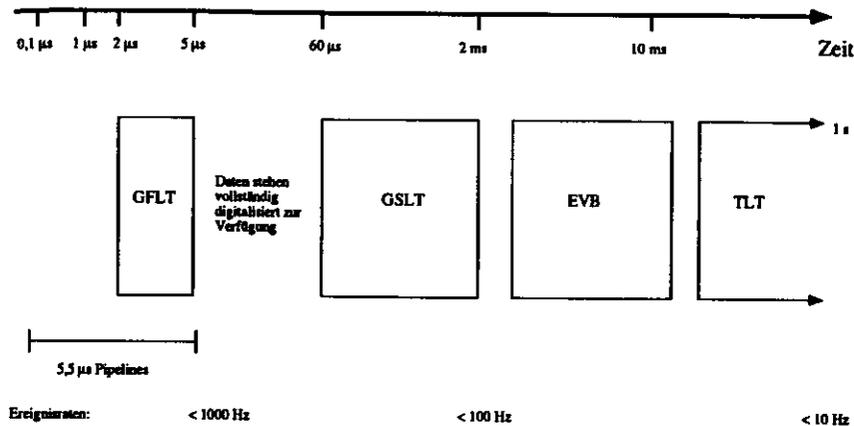


Abbildung 1.2: Zeitliche Zusammenhänge im ZEUS-Triggersystem

- In Elektronenrichtung folgt die Vetowand (VETO WALL), eine Eisenwand mit zwei Szintillatoren auf beiden Seiten. Sie soll den zentralen Detektorbereich vor Teilchen aus dem Strahl-Halo schützen.

Die folgenden beiden Detektorkomponenten sind in der Abbildung nicht zu erkennen, da sie sich nicht mehr in unmittelbarer Nähe der übrigen Detektorkomponenten befinden.

- Das Leading Proton Spectrometer (LPS) dient dem Nachweis von Protonen, die unter sehr kleinen Winkeln gestreut wurden. Es besteht aus 6 Siliziumdetektoren, die in Protonenrichtung zwischen 40m und 90m Entfernung vom Wechselwirkungspunkt aufgestellt sind.
- Der Luminositätsmonitor (LUMI) dient der Messung der Luminosität mit Hilfe von Elektron-Proton-Bremsstrahlung. Dazu werden die Elektronen mit einem Kalorimeter in 36m Entfernung und die Bremsstrahlungs-Photonen mit einem Kalorimeter in 108m Entfernung vom Wechselwirkungspunkt nachgewiesen.

1.2 Das ZEUS Trigger- und Datenerfassungssystem

Beim Betrieb des HERA-Speicherrings wird es am Wechselwirkungspunkt des ZEUS-Experiments alle 96 ns zu einer Begegnung von Elektronen- und Protonenpaketen kommen. Selbst wenn jeder der über 250000 Kanäle des Detektors nur ein Byte an Information liefern würde, erhielte man bei dieser Ereignisrate eine Datenmenge von über 2,5 TB/s. Da diese Datenmenge mit heutiger Technologie weder aufzeichnenbar noch sinnvoll nutzbar ist, zeigt sich hier eine der Hauptaufgaben des Triggersystems: die Datenreduktion.

Die meisten der anfallenden Daten haben ihren Ursprung nicht in Elektron/Proton-Wechselwirkungen, sondern in unerwünschten Nebenreaktionen wie Verlust von Teilchen aus

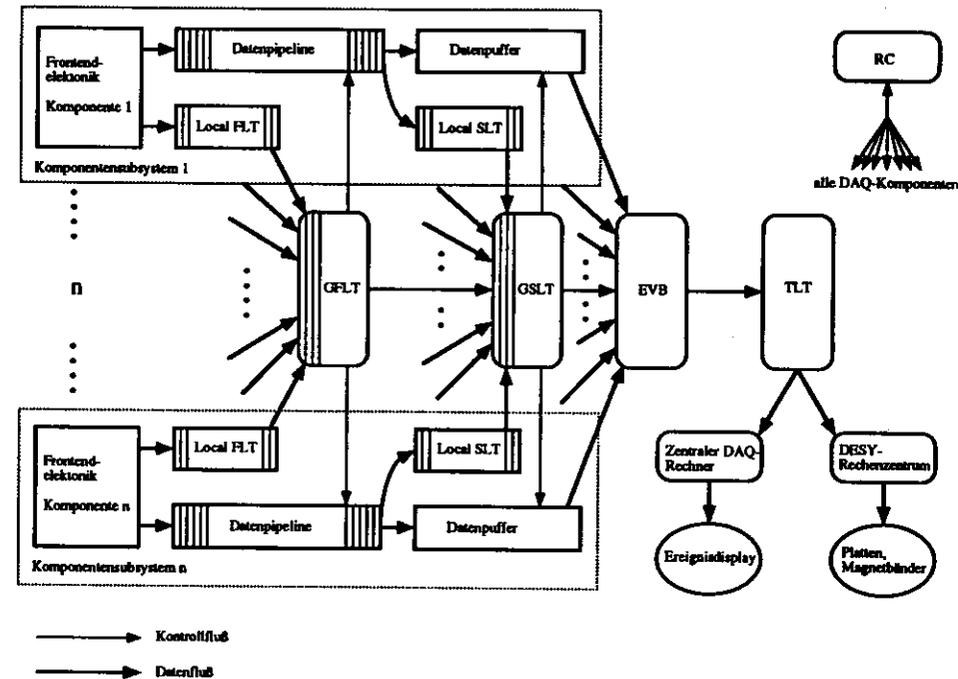


Abbildung 1.3: Vereinfachte Darstellung des Datenflusses im ZEUS-Trigger- und Datenerfassungssystem (Erläuterung siehe Text). Die Darstellung der Komponentensubsysteme ist nur für das hochauflösende Kalorimeter und Zentrale Spurkammer streng richtig, andere Komponentensubsysteme können z.T. eine andere Struktur besitzen.

Komponente		Anzahl der Auslesekanäle	Prozessortyp
Central Tracking Detector	CTD	4608	Transputer
Forward/Rear Tracking Detector	FTD/RTD	5778	Transputer
Barrel Calorimeter	BCAL	5184	Transputer
Forward Calorimeter	FCAL	4344	Transputer
Rear Calorimeter	RCAL	2336	Transputer
Transition Radiation Detector	TRD	2472	Transputer
Hadron Electron Separator	RHES	37304	Transputer
Backing Calorimeter	BAC	≈40000	Transputer
Vertex Detector	VXD	832	68 k
Beamline	BEAM	31	Transputer
Barrel/Rear Muon Detector	BMUO/RMUO	84336	Transputer
Forward Muon Detector	FMUO	18948	68 k
Leading Proton Spectrometer	LPS	52000	Transputer
Luminosity Monitor	LUMI		68 k
Vetowall	VETO		Transputer
Fast Clear	FCLR		68 k

Tabelle 1.1: Übersicht über die verschiedenen Komponentensubsysteme des ZEUS-Experiments

dem Strahl, Reaktionen mit den Restgas-Molekülen im Strahlrohr, oder auch in kosmischer Strahlung. Eine weitgehende Filterung der Primärdaten ist daher möglich und nötig. Diese Filterung setzt eine schnelle Analyse der Daten voraus, wozu ein hochparalleles mehrstufiges Triggersystem verwendet wird. Die zeitlichen Zusammenhänge zeigt Abb. 1.2.

Einen Überblick über das Trigger- und Datenerfassungssystem verschafft Abbildung 1.3. Sie zeigt den Datenfluß von der detektorkomponentenspezifischen Ausleseelektronik bis hin zur Datenaufzeichnung im DESY-Rechenzentrum. Die nun folgende Beschreibung des Systems bleibt im wesentlichen auf den Datenfluß beschränkt, Einzelheiten zu den verwendeten Triggeralgorithmen entnehme man etwa [Smith90].

1.2.1 Die Komponentensubsysteme

Die einzelnen Detektorkomponenten benötigen eine an ihren Aufbau angepasste Ausleseelektronik. Tabelle 1.1 zeigt eine Übersicht über die verschiedenen Komponentensubsysteme des Experiments. Aufgeführt sind der Prozessortyp, auf dem sie basieren, und die Anzahl ihrer Auslesekanäle.

Die Kanäle einer Komponente werden alle 96 ns ausgelesen und die Signale in eine Pipeline geschrieben, welche je nach Komponente digital (wie z. B. bei CTD) oder analog (wie bei RCAL, BCAL und FCAL) aufgebaut ist. Sie hat eine Tiefe von 52 Einträgen, wodurch sie Daten bis zu $52 \times 96\text{ ns} \approx 5\text{ }\mu\text{s}$ speichern kann. Während dieser Zeit hat die erste Triggerstufe über die weitere Prozessierung der Daten zu entscheiden.

1.2.2 Die erste Triggerstufe

Wie aus Abb. 1.3 ersichtlich, besteht die erste Triggerstufe (FLT, First Level Trigger) aus einem globalen sowie verschiedenen, bei den jeweiligen Detektorkomponenten befindlichen lokalen Teilen. Letztere führen eine parallele Ereignisanalyse auf Basis der ihnen zur Verfügung stehenden Daten durch und übermitteln ihre Resultate binnen $2\text{ }\mu\text{s}$ der globalen ersten Triggerstufe (GFLT), welche dann die Triggerentscheidung fällt. Diese steht genau $5\text{ }\mu\text{s}$ nach der Reaktion fest und wird dann an alle Komponenten verteilt. Die Rate der positiven Triggerentscheidungen beträgt 1000 Hz .

1.2.3 Die zweite Triggerstufe

Sobald die erste Triggerstufe eine Entscheidung getroffen hat, werden die Daten aus der Datenpipeline entweder verworfen oder gelangen, eventuell nach vorheriger Digitalisierung, zur lokalen zweiten Triggerstufe und in einen Datenpuffer.

Wiederum führen die lokalen Triggersubsysteme Berechnungen aufgrund der ihnen verfügbaren Daten aus und übermitteln die Ergebnisse an die globale Triggerstufe (GSLT), welche diese weiterverarbeitet und spätestens 3 ms nach der Reaktion eine Entscheidung über die Verwendung eines Ereignisses fällt. Der GSLT reduziert die Ereignisrate um einen Faktor 10 auf 100 Ereignisse/s .

1.2.4 Der Eventbuilder

Die Entscheidungen des GSLT werden zunächst an den Eventbuilder und von diesem weiter an die Detektorkomponenten verteilt.

Bislang sind die Daten eines Ereignisses noch auf die verschiedenen Komponentensubsysteme verteilt. Für die weitere Prozessierung werden Daten akzeptierter Ereignisse vom Eventbuilder eingelesen, in einer Datenstruktur zusammengeführt und an die dritte Triggerstufe übergeben. Da der Eventbuilder Gegenstand des nächsten Kapitels ist, folgt hier keine weitere Beschreibung.

1.2.5 Die dritte Triggerstufe

Die dritte Triggerstufe (TLT, Third Level Trigger) übernimmt die vollständigen Daten eines Ereignisses vom Eventbuilder und führt eine komplette Rekonstruktion und Analyse des Ereignisses durch. Auch die dritte Triggerstufe ist wieder parallel ausgelegt, sie besteht aus einer Workstationfarm, die über ein spezielles Bussystem an 6 Punkten mit dem Eventbuilder verbunden ist. Die Ereignisrate nach der dritten Triggerstufe wird etwa 5 Ereignisse/s betragen.

1.2.6 Die Datenaufzeichnung

Alle akzeptierten Ereignisse gelangen zum DESY-Rechenzentrum und werden in den dortigen Großrechner eingespeist. Dieser speichert die Daten auf Platte oder Band, von wo sie für die weitere Analyse abgerufen werden können. Die Aufzeichnungsgeschwindigkeit ist auf Seiten des Großrechners auf 500 KB/s begrenzt.

1.2.7 Die zentrale Steuerung

Das gesamte Experiment kann zentral von einem Punkt aus gesteuert werden. Hierzu existiert ein Steuerprogramm (Run Control, RC), welches über ein LAN¹ mit allen Komponenten des Trigger- und Datenerfassungssystems in Verbindung steht.

Mit seiner Hilfe können verschiedene Triggermodi ausgewählt werden, einzelne Komponenten de- oder aktiviert und das ganze System gestartet werden.

Kapitel 2

Der Eventbuilder

Der Eventbuilder als eine der grundlegenden Komponenten des ZEUS Trigger- und Datenerfassungssystems ist Gegenstand der Untersuchungen dieser Arbeit. Dieses Kapitel ist daher seiner genauen Beschreibung gewidmet. Dabei werden zunächst die Anforderungen, die an den Eventbuilder gestellt werden, vorgestellt. Sodann werden die Entwicklung und die Realisierung des Systems beschrieben.

2.1 Aufgaben und Anforderungen an den Eventbuilder

Jedes Ereignis, das im ZEUS-Detektor registriert wird, wird zunächst vom GFLT untersucht. Alle von diesem als potentiell physikalisch interessant erkannten Ereignisse erfahren eine erneute Untersuchung nach strengeren Maßstäben durch den GSLT. Für diese Untersuchung werden jedoch nur die Daten einiger Komponenten benötigt und dementsprechend besitzt der GSLT auch nur zu diesen Komponenten eine direkte Verbindung. Da die GSLT-Entscheidung jedoch an alle Komponentensubsysteme zu verteilen ist, übermittelt der GSLT diese zunächst an den Eventbuilder, der über Verbindungen zu allen verfügt und die weitere Verteilung übernimmt.

Falls nun auch der GSLT das Ereignis als potentiell physikalisch interessant eingestuft hat, beginnt die eigentliche Aufgabe des Eventbuilders, das Konstruieren von kompletten Ereignisdaten. Dazu werden von allen Komponentensubsystemen Daten erwartet —im folgenden als „Ereignisteildaten“ bezeichnet—, welche einzulesen und auf korrektes Datenformat¹ zu prüfen sind. Weiter sind die Daten der verschiedenen Komponentensubsysteme in einer gemeinsamen Datenstruktur² zusammenzufassen und an den TLT zu übergeben. Diese Datenstruktur soll im folgenden als „Ereignisdaten“ bezeichnet werden. Hierbei ist besonders zu beachten, daß der Datenfluß von den Komponentensubsystemen zum TLT nicht durch den Eventbuilder gebremst wird. Dies bedeutet aber, daß die Übertragungskapazität des Eventbuilders höher zu sein hat als die Einlese- und Verarbeitungskapazität des TLT.

Die Einbettung des Eventbuilders in seine Systemumgebung wird in Abbildung 2.1 dargestellt. Sie zeigt, daß eine Vielzahl von Schnittstellen, die den Eventbuilder mit dem Trigger- und Datenerfassungssystem verbinden, benötigt werden. Insgesamt lassen sich die Anforderungen an den Eventbuilder wie folgt klassifizieren:

¹Das einzulesende Datenformat ist ein abgewandeltes ZEBRA-Format. Vergleiche hierzu [Behrens90] und [ZEBRA86].

²Das Datenformat für die Ausgabe ist ADAMO, eine Beschreibung findet man in [ADAMO91]

¹Local Area Network

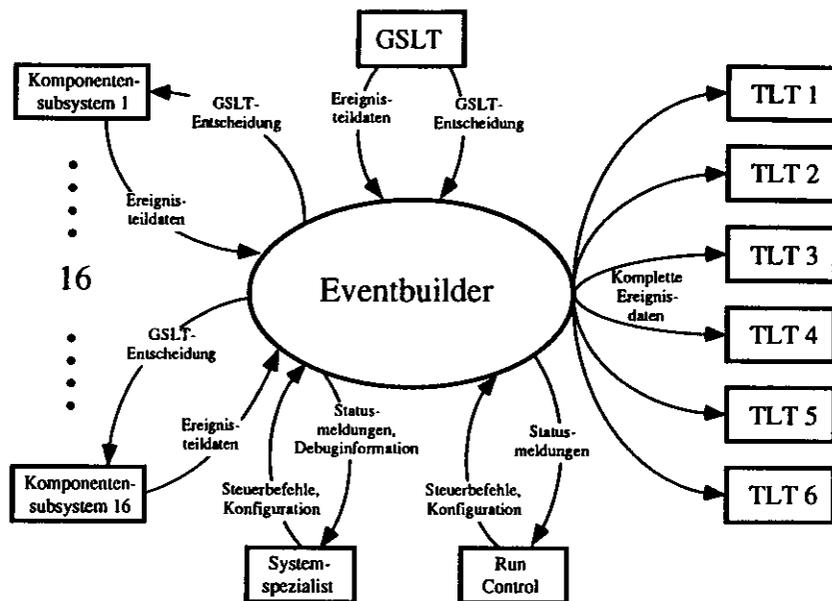


Abbildung 2.1: Einbettung des Eventbuilders in seine Systemumgebung

• **Anforderungen an die Schnittstellen zu den Komponentensubsystemen**

Für den Datenfluß im Eventbuilder stellen die Schnittstellen zu den Komponentensubsystemen die Datenquellen dar. Tabelle 2.1 zeigt die 17 Komponenten des Datenerfassungssystems, welche Ereignisteildaten an den Eventbuilder liefern.

Eine Sonderstellung hierbei nimmt der GSLT ein, welcher keine Detektorkomponente ist und somit über keine eigenen Auslesekanäle verfügt; im Zuge seiner Ereignisanalyse erzeugt er jedoch Daten, die genau wie die übrigen Ereignisteildaten vom Eventbuilder zu den Ereignisdaten zusammenzufassen sind. Da die Datenmenge des GSLT größenordnungsmäßig mit denen der übrigen Komponentensubsysteme übereinstimmt, besitzt er eine entsprechend aufgebaute Schnittstelle zum Eventbuilder.

Die Tabelle zeigt, daß von den Komponentensubsystemen dasjenige von CTD mit 30kB die größte Menge einzulesender Daten liefert. Bei einer durchschnittlichen Ereignisrate von 100 Ereignissen/s haben die Schnittstellen daher einen Datenfluß von bis zu

$$D_{KSS_{max}} := 3MB/s$$

zu bewältigen. Die meisten Ereignisteildaten werden jedoch eine Länge von 10kB nicht überschreiten, weshalb dort ein Datenfluß von

$$D_{KSS} := 1MB/s$$

Komponente		Datenlänge [KByte]
Central Tracking Detector	CTD	10 - 30
Forward/Rear Tracking Detector	FTD	10 - 15
Barrel Calorimeter	BCAL	5 - 20
Forward Calorimeter	FCAL	5 - 20
Rear Calorimeter	RCAL	5 - 10
Transition Radiation Detector	TRD	5 - 10
Hadron Electron Separator	HES	5 - 10
Backing Calorimeter	BAC	2
Vertex Detector	VXD	2
Beamline	BEAM	1
Barrel/Rear Muon Detector	BMUO	0.9
Forward Muon Detector	FMUO	0.5
Leading Proton Spectrom.	LPS	0.2
Luminosity Monitor	LUMI	0.2
Vetowall	VETO	0.01
Fast Clear	FCLR	1.2
Global 2nd Level Trigger	GSLT	10 - 20
Σ		62 - 143

Tabelle 2.1: Länge der Ereignisteildaten einzelner DAQ-Komponenten

ausreichend ist.

Räumlich verteilen sich die Schnittstellen zusammen mit den Komponentensubsystemen und großen Teilen der Ausleseelektronik im sog. Rucksack, einem dreistöckigen Stahlgebäude, welches direkt an den Detektor anschließt. Um von den Schnittstellen Verbindungen bis zum Ausgang des Rucksacks herzustellen, treten Kabellängen von bis zu 30m auf.

Elektrisch sollen diese Schnittstellen eine Übergabe unabhängig von dem vom Komponentensubsystem verwendeten Prozessortyp erlauben. Da die Komponentensubsysteme entweder auf Transputern oder auf 68k/VME-Systemen basieren, haben die Schnittstellen eine Übergabe entweder über VMEbus³ oder über Transputerlinks⁴ zu ermöglichen.

• **Anforderungen an die Schnittstellen zum TLT**

Für den Datenfluß im Eventbuilder stellen die Schnittstellen zum TLT die Datensinken dar. Der TLT besteht aus 36 Arbeitsplatzrechnern, die in 6 Gruppen zu je 6 Rechnern eingeteilt sind und sich in einem Computerraum in einer Entfernung von 60m zum Rucksack befinden. Der Eventbuilder benötigt zu jeder dieser Gruppen eine Schnittstelle auf VMEbus Basis.

³VMEbus ist ein von einer Gruppe von Rechnerherstellern 1981 entworfene und inzwischen zu einem weit verbreiteten Industriestandard gewordene Rechnerarchitektur [Peterson89].

⁴Mit „Link“ werden die auf dem Chip integrierten seriellen Schnittstellen einer speziellen Mikroprozessorfamilie, den Transputern, bezeichnet. Diese Mikroprozessoren wurden von INMOS Ltd. entwickelt und zeichnen sich zum einen durch eben diese seriellen Schnittstellen, zum anderen durch einen Befehlssatz aus, der eine einfache Implementierung einer von [Hoare78] vorgeschlagenen Methode der Prozesssynchronisation durch Ein-/Ausgabeanweisungen ermöglicht.

Wie aus Tabelle 2.1 zu entnehmen, beträgt die maximale Datengröße eines Ereignisses 143kB. Bei 100 Ereignissen/s entspricht das an jeder der 6 Schnittstellen einer benötigten Datenrate von

$$D_{TLT} := 2,4MB/s$$

Da die 36 TLT-Rechner die Ereignisse unabhängig voneinander verarbeiten und je nach Ereignistyp unterschiedlich lange rechnen, kann es vorkommen, daß an den Schnittstellen unterschiedlich schnell Daten abgenommen werden. Dies bedeutet für den Eventbuilder, daß die tatsächlich zu erbringende Datenrate wesentlich größer als D_{TLT} sein sollte, um so diese Schwankungen ausgleichen zu können. Zusätzlich sollte bereits bei der Verteilung der Daten auf die einzelnen TLT-Gruppen deren Auslastung berücksichtigt werden, so daß ein gleichmäßiger Datenfluß gewährleistet ist.

- **Anforderungen an die Schnittstellen zur Verteilung der GSLT-Entscheidung**
Der GSLT hat für alle vom GFLT akzeptierten Ereignisse eine Triggerentscheidung zu treffen. Daher liefert er sowohl positive als auch negative Entscheidungen mit einer Rate von 1000 Entscheidungen/s über einen Transputerlink an den Eventbuilder. Die Entscheidungen sind vom Eventbuilder einzulesen und schnellstmöglich an die 16 Komponentensubsysteme (der GSLT selbst benötigt sie nicht mehr) über Transputerlinks zu verteilen. Da eine Triggerentscheidung eine Länge von 16 Byte hat [Smith90], entspricht dies einer Transferrate von

$$D_{GSLT} := 16KB/s$$

Diese Transferrate ist für einen Transputerlink völlig unkritisch. Da jedoch die Komponentensubsysteme bis zum Eintreffen einer Entscheidung des GSLT alle Ereignisteildaten zu puffern haben, ist, um den benötigten Pufferspeicher nicht unnötig zu vergrößern, der zeitliche Abstand zwischen Einlesen und Verteilen der Entscheidung so gering wie möglich zu halten.

Da der Datentransfer auf Transputerlink byteorientiert erfolgt und jedes weitere Byte nur gesendet wird, wenn der Empfang des vorherigen bestätigt wurde⁵, haben die Komponentensubsysteme die Möglichkeit, die Übertragungsgeschwindigkeit zu verlangsamen bzw. den Transfer ganz zu unterbinden. Im Zuge der Systemintegration hat der Eventbuilder solche Subsysteme zu identifizieren, damit Abhilfe geschaffen werden kann.

- **Weitere Schnittstellen**
Um den Eventbuilder steuern und konfigurieren zu können, wird eine Verbindung zur zentralen Experimentsteuerung (Run Control) benötigt. Dazu wird üblicherweise ein Programmpaket namens ZIP [Park89] benutzt, welches diese Verbindung auf VMS-Systemen⁶ herstellt. Zur Entwicklung des Eventbuilders ist es außerdem unerlässlich dem Programmierer eine Schnittstelle zur Verfügung zu stellen, die ihm eine Steuerung und Überwachung des Eventbuilders unabhängig von Run Control ermöglicht.

- **Anforderungen an die Pufferkapazität**
Die Daten der verschiedenen Komponentensubsysteme können erst dann in einer gemeinsamen Datenstruktur zusammengefaßt und an den TLT übergeben werden, wenn

alle Ereignisteildaten im Eventbuilder vorliegen. Da nun aber die einzelnen Komponentensubsysteme ihre Daten jeweils unterschiedlich schnell nach Empfang der GSLT-Entscheidung zur Verfügung stellen können, muß der Eventbuilder bis zum Eintreffen der Daten der langsamsten Komponente die aller übrigen Komponenten in internen Speichern zwischenspeichern können.

2.2 Die Funktionseinheiten des Eventbuilders

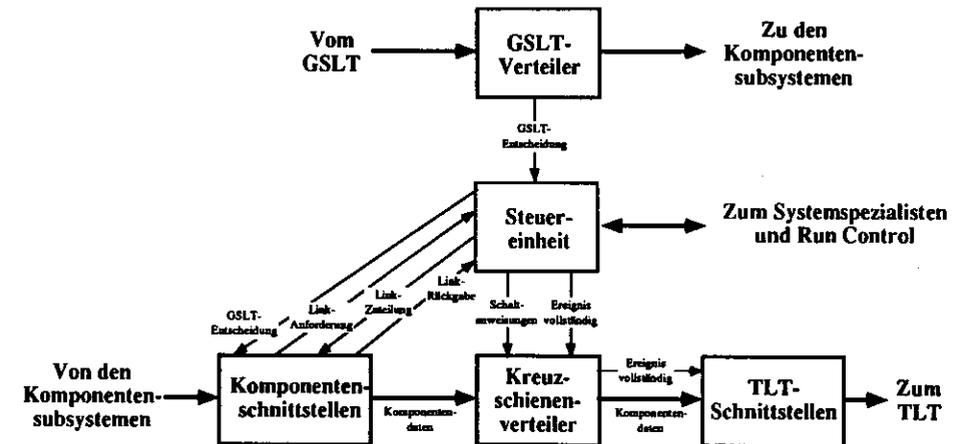


Abbildung 2.2: Die Funktionseinheiten des Eventbuilders und deren Zusammenspiel

Die Entwicklung des Eventbuilders wurde von einer von der Kollaboration entwickelten Prozessorkarte auf Transputerbasis, welche auch VMEbus Zugriffsmöglichkeiten bietet, maßgeblich beeinflusst. Diese Karte, das sog. 2TP-VME-Modul, eröffnet durch die hohe Rechenleistung der Transputer zum einen die Möglichkeit, Datenformate schon während der Datenerfassung zu überprüfen, zum anderen bieten die seriellen Schnittstellen der Transputer eine Möglichkeit, die Daten einfach und kostengünstig über die nicht unerhebliche Entfernung von 90m zwischen Komponentenschnittstellen im Rucksack und TLT-Schnittstellen im Computerraum zu transferieren.

Wie aus dem ersten Abschnitt bekannt ist, gibt es 17 Komponentenschnittstellen und 6 TLT-Schnittstellen. Die Aufgabe des Eventbuilder erfordert es nun, daß jede dieser Komponentenschnittstellen mit jeder TLT-Schnittstelle verbunden ist. Da handelsübliche Transputer jedoch

⁵ sog. Handshake oder Quittierungsbetrieb

⁶ VMS ist das von Digital entwickelte Betriebssystem für deren VAX-Rechnersysteme

nur 4 serielle Schnittstellen besitzen, ist es technisch nicht möglich, diese Verbindungen direkt herzustellen. Verschiedene Studien ([Hagge90],[Woeniger89]) zeigen, daß die dynamische Generierung dieser Verbindungen mit Hilfe eines Kreuzschienenverteilers den höchstmöglichen Datendurchsatz bei gleichzeitig geringsten Kosten und geringster Störanfälligkeit gibt: Es werden alle Kabel optimal genutzt und die Aufgaben defekter Kabel können innerhalb gewisser Grenzen (und auf Kosten des Datendurchsatzes) aufgrund der hohen Redundanz von anderen Kabeln mit übernommen werden.

Durch die Benutzung eines Kreuzschienenverteilers ergibt sich die Notwendigkeit einer weiteren Funktionseinheit des Eventbuilders, einer Steuereinheit. Deren Aufgabe besteht im Schalten des Kreuzschienenverteilers und enthält außerdem die im vorigen Abschnitt erwähnten Schnittstellen zu Run Control und den Systemspezialisten.

Der GSLT-Verteiler, der die GSLT-Entscheidung an alle Komponentensubsysteme übermittelt, vervollständigt schließlich die Auflistung der Funktionseinheiten des Eventbuilders. Alle Funktionseinheiten sowie deren Verbindungen sind in Abbildung 2.2 dargestellt.

Das Zusammenspiel dieser Funktionseinheiten wird über asynchronen Nachrichtenaustausch (sog. „message passing“) koordiniert. Insgesamt ergibt sich für den Eventbuilder folgendes Funktionsprinzip:

Zunächst empfängt der GSLT-Verteiler eine Triggerentscheidung. Diese sendet er sofort weiter an die Komponentensubsysteme und im Falle einer positiven Triggerentscheidung an die Steuereinheit, welche die Entscheidung dann an die Komponentenschnittstellen übermittelt.

Die Komponentenschnittstellen warten nun auf das Eintreffen der zugehörigen Ereignisteil- daten von den Komponentensubsystemen. Sobald diese eintreffen, werden sie in einen lokalen Speicher kopiert und ihre Datenstruktur auf Fehler hin untersucht. Weiterhin wird geprüft, ob die Datenstruktur den durch die Triggerentscheidung bedingten Anforderungen genügt. Anschließend wird die Steuereinheit von dem Vorhandensein der Daten in Kenntnis gesetzt, indem ihr eine Anforderung für eine Verbindung zum TLT („Link-Anforderung“) übermittelt wird.

Die Steuereinheit stellt nun fest, auf welcher TLT-Schnittstelle die Daten des betreffenden Ereignisses gesammelt werden, um dann schnellstmöglich eine Verbindung zu schalten. Steht für ein Ereignis noch nicht fest, über welche TLT-Schnittstelle es ausgegeben werden soll, so wählt die Steuereinheit unter Berücksichtigung der momentanen Last auf dem TLT eine Schnittstelle aus, auf der das Ereignis dann konstruiert wird. Die Komponentenschnittstelle wird über die geschaltete Verbindung durch eine „Link-Zuteilung“-Nachricht informiert.

Erhält eine Komponentenschnittstelle ein „Link-Zuteilung“, so transferiert sie die dazugehörigen Ereignisteil- daten über die entsprechende Verbindung durch den Kreuzschienen- verteiler zu einer TLT-Schnittstelle. Das Ende dieses Transfers wird der Steuereinheit durch eine „Link-Rückgabe“- Meldung angezeigt, woraufhin die geschaltete Verbindung wieder abgebaut werden kann.

Sobald die Steuereinheit feststellt, daß von allen Komponentenschnittstellen die Daten zu einem Ereignis zu einer TLT-Schnittstelle gelangt sind, sendet sie der TLT-Schnittstelle durch den Kreuzschienenverteiler eine „Ereignis vollständig“- Nachricht, welche daraufhin die Daten in einer gemeinsamen Datenstruktur zusammenzufassen und an den TLT zu übergeben hat. Abbildung 2.2 faßt die Funktionsweise des Eventbuilders nochmals zusammen.

2.3 Der Aufbau des Eventbuilders

Der Funktionseinheiten des Eventbuilders sind zum überwiegenden Teil aus Transputern aufgebaut, die über ihre seriellen Schnittstellen miteinander kommunizieren. Die meisten dieser Transputer befinden sich auf den vorher schon erwähnten Prozessorkarten von Typ 2TP-VME, welcher innerhalb des ZEUS-Experiments zum Standard geworden ist. Lediglich einige besondere Bestandteile des Eventbuilders benutzen eine gesonderte Hardware. Im Folgenden werden nun die verschiedenen Funktionseinheiten, deren Hardwareaufbau sowie die Struktur der auf ihnen laufenden Programme vorgestellt.

2.3.1 Der GSLT-Verteiler

Wie schon erläutert wurde, liest der GSLT-Verteiler zunächst eine Triggerentscheidung vom GSLT ein und verteilt sie an die Komponentensubsysteme weiter. Den Aufbau des Verteilers zeigt Abbildung 2.3. Er besteht aus einem Transputer aus der T4-Familie. Dieser ist mit einer besonderen, beim DESY entwickelten Hardware ausgestattet, welche einen Eingangslink auf bis zu 32 Ausgangslinks auffächern kann. Zur Zeit sind davon jedoch nur 16 in Benutzung.

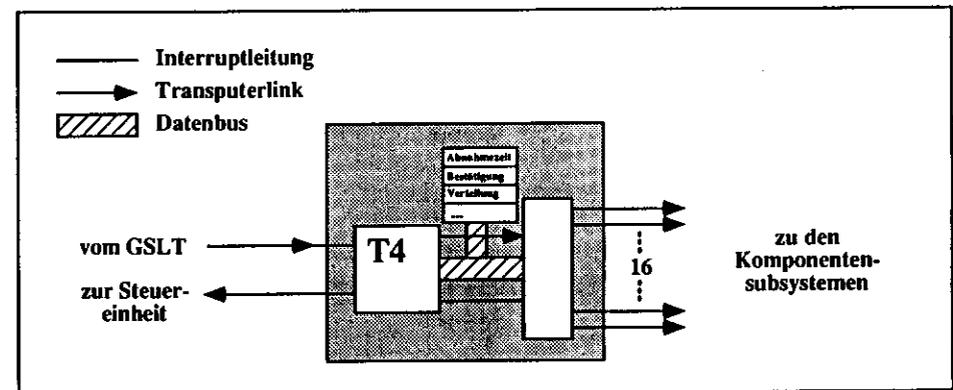


Abbildung 2.3: Der GSLT-Verteiler

Diese Hardware enthält verschiedene Register: Mit einem Register kann eingestellt werden, welche der 32 Ausgangslinks benutzt werden sollen. In einem anderen Register wird eine Verzögerungszeit ausgewählt. Innerhalb dieser Verzögerungszeit müssen alle Links den Empfang eines Bytes quittiert haben, andernfalls wird eine Unterbrechungsanforderung (sog. Interrupt oder Event) an den T4 gesendet, welcher dann einem weiteren Register entnehmen kann, von welchen Ausgangslinks die Bestätigungen noch fehlten. Durch diesen Aufbau ist es möglich, diejenigen Komponentensubsysteme zu identifizieren, welche die GSLT- Entscheidungen nicht oder zu langsam abnehmen.

Da die GSLT-Entscheidungen parallel an alle Komponentensubsysteme verteilt werden, verzögert der GSLT-Verteiler die Versendung der Entscheidung nur um die Zeitspanne, die er selbst zum Einlesen benötigt. Dadurch ist die schnellstmögliche Verteilung gewährleistet.

2.3.2 Die Komponentenschnittstellen

Abbildung 2.4 zeigt den Einsatz eines 2TP-VME-Moduls als Komponentenschnittstelle. Jedes 2TP-VME-Modul enthält zwei Prozessoren vom Typ T800, ein VMEbus-Interface sowie einen allen drei Bestandteilen gemeinsamen Speicher, ein sogenanntes „Triple Ported Memory“ (TPM). Lediglich das TPM sowie der mit „X“ bezeichnete Transputer bilden eine Komponentenschnittstelle des Eventbuilders, der andere Transputer steht dem Komponentensubsystem zur freien Verfügung. Damit ist es sowohl Komponentensubsystemen auf 68k/VME-Basis als auch solchen, die auf Transputern basieren, möglich, ihre Daten im TPM abzulegen, um sie so an den Eventbuilder zu übergeben.

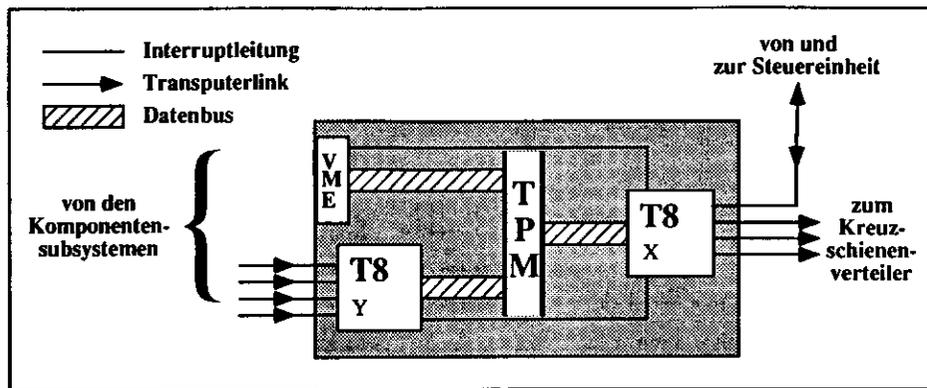


Abbildung 2.4: Das 2TP-VME-Modul als Komponentenschnittstelle

Das TPM wird in Form eines Ringpuffers verwaltet. Der genaue Aufbau des Übergabeprotokolls ist bei [Behrens91] beschrieben. Hat ein Komponentensubsystem seine Ereignisteildaten im TPM abgelegt, so signalisiert es dem „X“-Transputer das Vorliegen neuer Daten über ein Unterbrechungssignal (sog. Interrupt oder Event).

Dieser wiederum kopiert die Daten zunächst in einen internen, 3 MB großen Puffer, überprüft die Einhaltung des Datenformates und sendet, sobald er von der Steuereinheit die zu diesem Ereignis gehörende Triggerentscheidung erhalten hat, eine „Link-Anforderung“-Meldung an diese zurück.

Die Steuereinheit sendet ihrerseits dann eine „Link-Zuteilung“-Meldung, aus der hervorgeht, über welchen der 3 Links zum Kreuzschienenverteiler die Daten aus dem internen Puffer zu senden sind.

Nachdem die Daten von einer TLT-Schnittstelle empfangen worden sind, wird dies mit einer Quittierungsmeldung bestätigt. Diese Meldung enthält außerdem Informationen über die Auslastung der betreffenden TLT-Schnittstelle, die zusammen mit der „Link-Rückgabe“-Meldung an die Steuereinheit übermittelt werden.

2.3.3 Die TLT-Schnittstellen

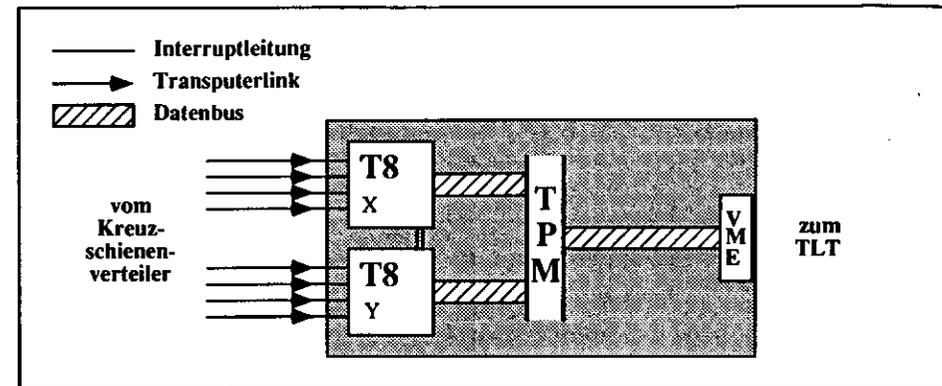


Abbildung 2.5: Das 2TP-VME-Modul als TLT-Schnittstelle

Abbildung 2.5 zeigt ein 2TP-VME-Modul als TLT-Schnittstelle. Die beiden Transputer empfangen unabhängig voneinander Ereignisteildaten, die ihnen von den Komponentenschnittstellen durch den Kreuzschienenverteiler zugesandt werden. Dabei gelangen Ereignisteildaten von einem bestimmten Ereignis immer zu demselben Transputer. Dieser schreibt die Daten zunächst in einen privaten, 3 MB großen Puffer und sendet anschließend eine Meldung über die interne Pufferauslastung an die Komponentenschnittstelle zurück.

Nachdem alle Teildaten eines Ereignisses empfangen wurden, wird von der Steuereinheit durch den Kreuzschienenverteiler eine „Ereignis vollständig“-Meldung geschickt. Diese Meldung veranlaßt den empfangenden Transputer, die Datenteile aus dem internen Puffer zu einer ADAMO-Datenstruktur zusammenzufassen. Sodann erfolgt das Kopieren der Daten in das TPM und damit die Übergabe an den TLT.

2.3.4 Der Kreuzschienenverteiler

Der Kreuzschienenverteiler erfüllt die Aufgabe, Verbindungen zwischen einem beliebigen Link einer Komponentenschnittstelle und einem beliebigen Link einer TLT-Schnittstelle herzustellen.

Die hierzu verwendete Schaltung ist bereits hinreichend in [Hagge90] beschrieben, deshalb sei hier nur erwähnt, daß sie, wie aus Abbildung 2.6 zu erkennen, von einem T2-Transputer gesteuert wird, der über 2 Links mit der Steuereinheit verbunden ist. Von dieser erhält er die Befehle zum Auf- und Abbau der Verbindungen.

Der Kreuzschienenverteiler ist innerhalb des Rucksacks über konventionelle Kabel mit den Komponentenschnittstellen verbunden, die Verbindung zu den TLT-Schnittstellen im Computerraum hingegen wird über 40m lange Glasfaserkabel hergestellt.

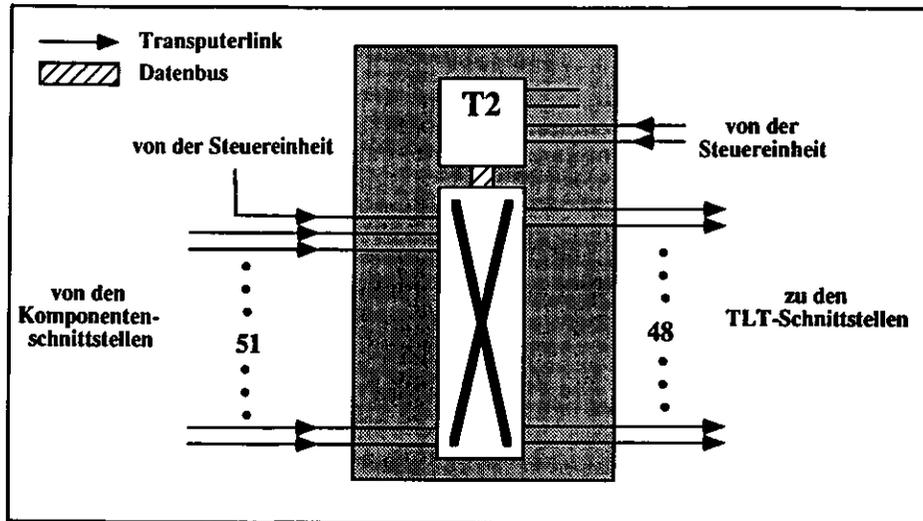


Abbildung 2.6: Der Kreuzschienenverteiler

2.3.5 Die Steuereinheit

Kommen wir nun zur größten und kompliziertesten Funktionseinheit des Eventbuilders, der Steuereinheit. Wie Abbildung 2.7 zeigt, besteht sie aus 16 Transputern und einer Workstation. Zur besseren Übersicht werden die einzelnen Bestandteile getrennt voneinander vorgestellt.

Der TTZZ

Wie bereits im vorherigen Abschnitt gezeigt wurde, benötigt die Steuereinheit Verbindungen zu allen 17 Komponentenschnittstellen. Da handelsübliche Transputer lediglich 4 Links besitzen ist eine Auffächerung des eigentlichen Ausgangs der Steuereinheit erforderlich. Hierfür wurde ein „ternärer Baum“ (TTZZ⁷) aus 12 Transputern aufgebaut (vgl. Abb. 2.7), dessen Ausgänge mit den Komponentenschnittstellen verbunden sind.

Die auf diesem Baum laufende Software ermöglicht es der Steuereinheit, Meldungen an alle oder auch nur an zuvor ausgewählte Komponentenschnittstellen zu senden bzw. von ihnen zu empfangen. Besonders wichtig ist dabei, daß auch Fehler im Komponentensubsystem (z.B. Stromausfall) erkannt werden und den Betrieb des Eventbuilders nicht unterbrechen.

Der Mixer

Dem Eingang des TTZZ vorgelagert ist der Mixer, dessen Aufgabe in der Aufnahme von GSLT-Entscheidungen sowie deren Einspeisung in den Nachrichtenfluß des Eventbuilders besteht. Der Mixer ist zusammen mit dem Supervisor auf einem 2TP-VME-Modul implementiert, wobei allerdings dessen TPM und VMEbus nicht benutzt werden (Abb. 2.7).

⁷Transputer Tree from Zenith for ZEUS

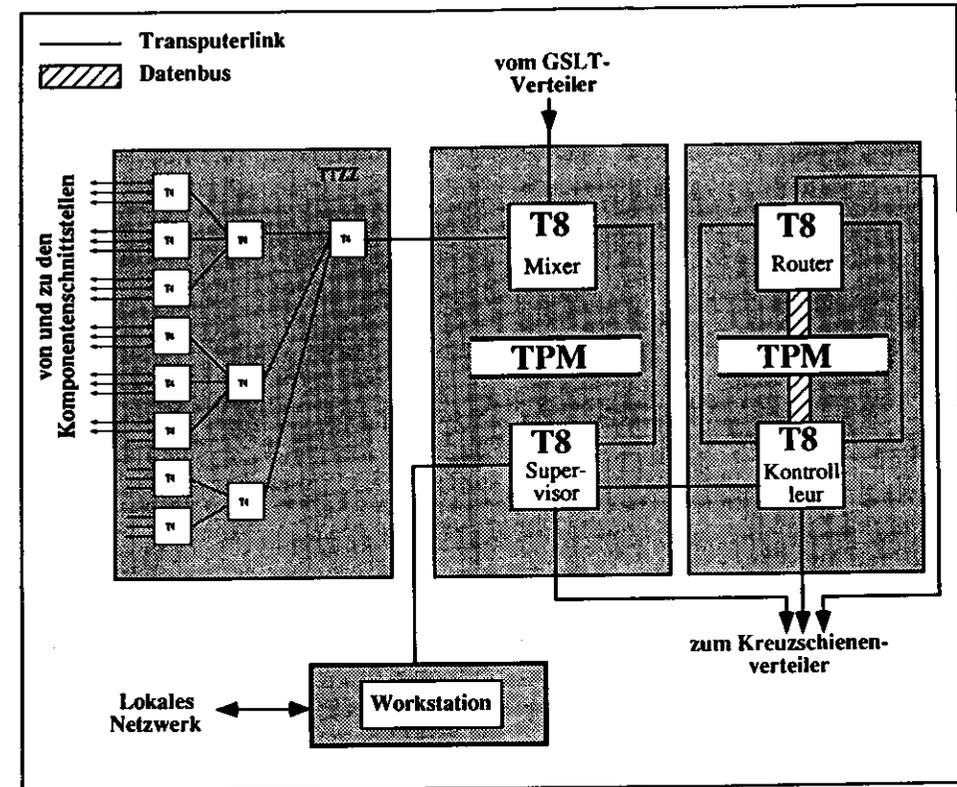


Abbildung 2.7: Die Steuereinheit

Der Supervisor

Der Supervisor-Transputer ist mit dem Mixer, dem Kreuzschienenverteiler, dem Kreuzschienenkontrollleur und der Workstation verbunden. Er ist die Instanz der Steuereinheit, die die eigentliche Steuerung des Eventbuilders vornimmt. Dies beinhaltet sowohl die Koordination der Konstruktion von Ereignissen als auch die Eingliederung in die Kommunikation mit der zentralen Experimentsteuerung.

Alle Link-Anforderungs- und Freigabemeldungen sowie die entsprechenden Zuteilungen durch den Kreuzschienenrouter (s.u.) werden durch den Supervisor geleitet. Dadurch kann der Supervisor ermitteln, welche Ereignisse auf welchem TLT-Knoten gerade konstruiert werden und welche der benötigten Ereignisteildaten dort bereits vorliegen. Erkennt der Supervisor, daß alle Teildaten eines Ereignisses auf einem TLT-Knoten vorliegen, so sendet er an diesen eine „Ereignis vollständig“-Meldung, um so die „ADAMOisierung“, d.h. Konstruktion und Formatierung des vollständigen Ereignisses einzuleiten.

Um das gesamte Trigger- und Datenerfassungssystem des ZEUS-Experiments kontrollieren und koordinieren zu können, wurde mit dem „Run Sequencing“ [Youngman91] festgelegt, unter welchen Bedingungen und in welcher Reihenfolge welche Komponente des Systems welchen Zustand anzunehmen hat; Die Einhaltung dieses „Run Sequencing“ sowie der dazu notwendige Botschaftenaustausch mit Run Control wird ebenfalls vom Supervisor koordiniert.

Kreuzschienenrouter und Kreuzschienenkontrolleur

Mit einem weiteren 2TP-Modul ist das TransputerNetz der Steuereinheit vollständig. Seine beiden Transputer werden als „Kreuzschienenkontrolleur“ und „Kreuzschienenrouter“ bezeichnet; sie haben die Aufgabe, von den Komponentenschnittstellen⁸ einlaufende Link-Anforderungen und Freigaben zu analysieren, eine geeignete Verbindung zu ermitteln und diese auf- bzw. abzubauen.

Eintreffende Meldungen werden zunächst vom Kreuzschienenrouter angenommen. Falls es sich um die erste Link-Anforderung eines Ereignisses handelt, legt er fest, auf welchem der 12 TLT-Schnittstellentransputer das Ereignis konstruiert werden soll. Dazu verwendet er Informationen über den Füllzustand des internen Puffers der verschiedenen TLT-Schnittstellen, die ihm zusammen mit vorherigen Link-Freigabe-Meldungen übermittelt wurden. Dadurch ist es möglich, Ereignisdaten stets auf den Transputer zu dirigieren, der momentan am wenigsten belastet ist (sog. Load Balancing). Ansonsten übermittelt er dem Kreuzschienenkontrolleur Komponentenschnittstelle und TLT-Schnittstellentransputer, zwischen denen eine Verbindung herzustellen ist. Dieser prüft dann, ob die Komponentenschnittstelle und der TLT-Schnittstellentransputer momentan über freie Links verfügen. Sobald diese vorhanden sind, schickt er eine Schaltanweisung an den Kreuzschienenverteiler, der die entsprechende Verbindung aufbaut. Weiter setzt er eine Link-Zuteilungsmeldung an die Komponentenschnittstelle ab.

Empfängt der Kreuzschienenrouter eine Link-Freigabemeldung, so leitet es diese an den Kreuzschienenkontrolleur weiter, der die Verbindung dann wieder abbauen kann.

Die Workstation

Ebenfalls ein Bestandteil der Steuereinheit ist eine Workstation⁹, der eine spezielle Schnittstelle Kommunikation über Transputerlinks ermöglicht. Dieser Rechner ist über LAN mit den anderen Workstations und Steuerrechnern des ZEUS-Experiments verbunden. Ein spezielles Programm („Server“) stellt nun die Verbindung zur zentralen Experimentsteuerung (Run Control) her. Zusätzlich können noch bis zu 5 weitere Systemspezialisten mit Hilfe des Programmes „expert“ eine Verbindung zum Server herstellen. Durch dieses Programm sind Run Control und die Systemspezialisten in der Lage, per Kommando die Steuerprogramme in die Transputer des Eventbuilders zu laden. Dazu benötigt der Server zunächst eine Liste der auszulesenden Komponentenschnittstellen und der aktiven TLT-Schnittstellen, denn im Zuge der Systemintegration kommt es häufig vor, daß einige Detektorkomponenten nicht an der Auslese teilnehmen können. Anhand dieser Liste erzeugt der Server dann ein Bootprogramm, welches in zwei Stufen in das TransputerNetz geladen wird. Durch die jeweils neue Erzeugung eines Bootprogramms ist es möglich, bestimmte Komponenten- oder TLT-Schnittstellen beim

⁸Die Nachrichten werden von den Komponentenschnittstellen durch TTZZ, Mixer und Supervisor an den Kreuzschienenrouter und den Kreuzschienenkontrolleur gesendet.

⁹SGI 4D/25 unter UNIX

booten überhaupt nicht anzusprechen, insbesondere stört es dann nicht, wenn diese gar nicht eingeschaltet sind.

In der ersten Stufe nun wird nur ein kurzes Programm gebootet, welches den Kreuzschienenverteiler in einen definierten Ausgangszustand bringt, damit dann das Booten des eigentlichen Programms in der zweiten Stufe durch den Kreuzschienenverteiler möglich wird.

Nach dem Laden der Steuerprogramme stellt der Server die Schnittstelle zwischen dem TransputerNetzwerk und der Außenwelt dar. Durch ihn ist der Supervisortransputer in der Lage, Nachrichten mit Run Control und den Systemexperten auszutauschen.

Kapitel 3

Möglichkeiten der Echtzeitüberwachung und Analyse des Eventbuilder-Betriebes

Wie aus den vorherigen Kapiteln deutlich wurde, ist das ZEUS-Datenerfassungssystem ein sehr großes, komplexes und heterogenes Rechnersystem, an dessen Planung, Realisierung und Programmierung eine große Anzahl von Wissenschaftlern und Technikern beteiligt ist. Wie bei jedem großen technischen System gibt es in der Entstehungs- und Betriebsphase unzählige Schwierigkeiten und Probleme, die nur durch fortwährende Iteration von Erprobung und anschließender Veränderung des Systems gelöst werden können.

Die Anzahl der Iterationen und damit die benötigte Zeit zur Fertigstellung sowie die benötigten Betriebsmittel können um so mehr verringert werden, je genauer die Ursachen von Problemen und je besser die Zusammenhänge innerhalb des Systems verstanden sind, weil damit viel gezieltere Veränderungen am System möglich werden.

Ein Mittel zum besseren Verständnis ist die gezielte Überwachung, mit der man sich einen Einblick in die internen Abläufe und Zusammenhänge zu verschaffen sucht. Dazu zerlegt man das Gesamtsystem in einzelne Funktionseinheiten und untersucht das Zusammenspiel dieser Einheiten. Überwachung, ausgehend von einer einzelnen Funktionseinheit, wie sie der Eventbuilder innerhalb des DAQ-Systems darstellt, kann nun in zwei Richtungen erfolgen:

Zum einen kann sie extern orientiert sein, d. h. die Überwachung zielt auf die an den Eventbuilder angrenzenden Funktionseinheiten und dient der Überprüfung der an sie gestellten Anforderungen. Zum anderen kann sie intern orientiert sein; in diesem Fall dient sie der Überprüfung der an den Eventbuilder selbst gestellten Anforderungen und damit einer Leistungs- und Zuverlässigkeitsbestimmung. Weiter dient sie dem besseren Verständnis der internen Abläufe und ermöglicht so eine gezielte Leistungsoptimierung.

Um ein System überwachen zu können, benötigt man eine Reihe von Kenngrößen, mit deren Hilfe interne Vorgänge beschrieben und parametrisiert werden. Die für die Überwachung des Eventbuilders und der unmittelbar mit ihm verbundenen Systeme gewählten Kenngrößen werden im folgenden dargestellt. Da viele dieser Kenngrößen auf Zeitmessungen beruhen, folgt zunächst ein Abschnitt über Zeitmessung in Transputernetzen.

3.1 Zeitmessung in Transputernetzen

Transputer besitzen zur Zeitmessung zwei Zähler, Z_{low} und Z_{high} , wobei Z_{low} alle $64\mu s$ und Z_{high} jede Mikrosekunde um eins inkrementiert werden. Jeder Prozeß innerhalb des Transputers kann abhängig von seiner Priorität¹ allerdings nur auf einen der beiden Zähler zugreifen. Um welchen es sich handelt wird dabei von der jeweiligen Prozeßpriorität festgelegt. Da die Prozesse im Eventbuilder zum überwiegenden Teil niedrig priorisiert sind, kann nur der $64\mu s$ -Zähler verwendet werden.

Bei der Auswertung von Zählerständen ist zu beachten, daß Zähler von unterschiedlichen Transputern eines Netzes nicht ohne weiteres miteinander verglichen werden können: Die Zähler eines Transputers werden immer dann auf null gesetzt, wenn dieser mit einem Programm geladen wird. Da die Transputer innerhalb eines Netzes alle nacheinander mit einem Programm geladen werden, haben ihre Zähler nie den gleichen Zählerstand. Weiterhin beziehen die Transputer ihren Systemtakt —und damit den Takt für die Zähler— aus ihnen eigenen Schwingkreisen, die durch Schwingquarze stabilisiert werden. Da diese Quarze nicht mit exakt gleicher Frequenz schwingen, entstehen Gangunterschiede zwischen den Zählern verschiedener Transputer.

Eine „Absolutzeit“²-Information ist in einem Transputernetz also nur dann eindeutig bestimmbar, wenn die Gangunterschiede der Zähler verschiedener Transputer relativ zueinander sowie deren Zählerstände zu einem festen Zeitpunkt bekannt und hinreichend konstant sind. In den folgenden Abschnitten definierte Zeiten sind grundsätzlich als Absolutzeiten zu verstehen. Ihre Messung wird im Zusammenhang mit der Erfassung der Kenn- und Überwachungsgrößen erläutert.

3.2 Kenngrößen externer Systeme

Die unmittelbar an den Eventbuilder anschließenden Systeme sowie deren Schnittstellen wurden bereits im vorigen Kapitel dargestellt. Von diesen Schnittstellen sind diejenigen zu Run Control und den Systemspezialisten von geringem Interesse, da sie während der eigentlichen Datenerfassungsphase nicht benötigt werden. Wichtig sind dagegen die Schnittstellen zum GSLT, zu den Komponentensubsystemen und zum TLT, da ihre Funktion für einen Datentransport durch den Eventbuilder unerlässlich ist. Folgende Kenngrößen beziehen sich auf diese Schnittstellen:

- **Die GSLT-Frequenz**

Der GSLT liefert seine Entscheidungen an den Eventbuilder. Die Anzahl der einlaufenden positiven Triggerentscheidungen pro Zeit bestimmen die Rate, mit der der Eventbuilder Ereignisteildaten zu Ereignisdaten zusammenfassen muß. Damit ist diese Rate eine der wichtigsten Kenngrößen für den Eventbuilderbetrieb, denn sie bestimmt die zu erbringende Datentransfer- und Rechenleistung.

Die Frequenz f_{Trig} ist definiert als

$$f_{Trig} := \frac{\text{Anzahl der positiven Triggerentscheidungen des GSLT in einem Zeitintervall}}{\text{Länge des Zeitintervalls}}$$

¹Transputer sind in der Lage, mehrere Prozesse pseudoparallel auszuführen. Dabei werden die Prozesse in einem priorisierten Zeitscheibeverfahren der CPU zur Abarbeitung übergeben (vgl. [NMOS88],[Hagge90]).

²Mit Absolutzeit ist hier der Bezug zu einer einzelnen festen Uhr gemeint.

Die Triggerentscheidungen treffen vom GSLT nicht mit konstantem Abstand ein. Dadurch kommt es zu lokalen Schwankungen in der Frequenz, je nachdem, wie groß das Zeitintervall ist, über das gemittelt wird. Auf den Eventbuilder hat nun aber eine kurzfristig hohe Frequenz Auswirkungen, da der Durchsatz durch den Eventbuilder nicht kurzzeitig beliebig erhöht werden kann. Vielmehr muß der Eventbuilder die Daten puffern, um die sich so ansammelnden Daten in einer Phase niedrigerer Frequenz abzutransportieren. Um solche Schwankungen in der Frequenz erkennen zu können, muß die Mittelung über hinreichend kleine Zeitintervalle erfolgen.

Im Eventbuilder können keine Frequenzen direkt gemessen werden. Vielmehr wird der Zeitpunkt aufgezeichnet, zu dem eine positive GSLT-Entscheidung eingelesen wurde. Die Ereignisse, auf die sich diese Entscheidung bezieht, sind, bei null beginnend, fortlaufend durchnummeriert. Bezeichne $t_{Trig,n}$ den Zeitpunkt, zu dem die GSLT-Entscheidung für das n -te Ereignis dem Eventbuilder übermittelt wurde. Dann ist

$$f_{Trig} = \frac{|\{n | t_{Trig,n} \in [t_0, t_1]\}|}{t_1 - t_0}$$

die GSLT-Frequenz während des Zeitintervalls von t_0 bis t_1 . Der Abstand zweier aufeinanderfolgender Triggerentscheidungen

$$\Delta t_{Trig,n} := t_{Trig,n} - t_{Trig,n-1}$$

zeigt, ob und wie stark es zu Schwankungen in der Rate kommt.

• Abnahmzeit der Komponentensubsysteme für Triggerentscheidungen

Über spezielle Transputerlinks verteilt der Eventbuilder die GSLT-Entscheidungen an die Komponentensubsysteme. Diese sind verpflichtet, die Entscheidungen so schnell wie möglich einzulesen, damit der GSLT nicht durch zu langsam oder gar nicht abgenommene Entscheidungen blockiert wird. Die vom Eventbuilder verwendete Hardware bietet keine praktikable Möglichkeit, die Einlesezeit einzelner Komponentensubsysteme zu bestimmen. Es ist allerdings möglich, eine Zeitspanne $t_{Verteil}$ vorzugeben, innerhalb derer alle Komponentensubsysteme die GSLT-Entscheidung zu akzeptieren haben. Dazu werden die 16 Byte einer Triggerentscheidung einzeln nacheinander übertragen. Der Empfang eines jeden Bytes muß von allen Komponentensubsystemen innerhalb von $t_{Verteil}$ bestätigt worden sein. Diejenigen Komponentensubsysteme, welche den Empfang nicht oder zu spät quittiert haben, werden von der weiteren Verteilung ausgeschlossen³. Durch Variation von $t_{Verteil}$ kann untersucht werden, welche Komponentensubsysteme welche Verzögerung für den GSLT bedeuten.

• Länge der Ereignisteildaten

Die Komponentensubsysteme liefern für jede positive Triggerentscheidung des GSLT Ereignisteildaten an den Eventbuilder⁴. Die Länge $L_{K,n}$ der Daten des Komponentensubsystems K für Ereignis n beeinflusst das Leistungsvermögen des Eventbuilders, da bei längeren Daten mehr Zeit für deren Bearbeitung und Übertragung zu den TLT-Schnittstellen benötigt wird.

• Antwortverhalten der Komponentensubsysteme auf positive Triggerentscheidungen

³Dies hat im allgemeinen einen Abbruch der Datenauslese des Experiments zur Folge.

⁴Sollte eine Komponente ereignisbedingt keine Daten erfaßt haben, so hat sie dies durch eine leere Datenstruktur dem Eventbuilder mitzuteilen.

Interessant ist der Zeitpunkt, zu dem die einzelnen Komponentensubsysteme ihre Daten an den Eventbuilder übergeben, denn der zeitliche Abstand zwischen dem schnellsten und langsamsten Komponentensubsystem bestimmt die Größe des vom Eventbuilder benötigten Pufferspeichers: Mit Einlesen der ersten Teildaten eines Ereignisses ist für dieses Ereignis ausreichend Pufferplatz im Eventbuilder zu reservieren, der erst mit Abschluß der Konstruktion dieses Ereignisses wieder freigegeben werden kann. Je länger nun die Konstruktion eines Ereignisses dauert, desto mehr Ereignisse befinden sich gleichzeitig im Eventbuilder und desto mehr Speicherplatz wird zu deren Pufferung benötigt. Die Antwortverzögerung $d_{K,n}$ des Komponentensubsystems K ergibt sich mit $t_{K,n}$ als dem Zeitpunkt der Übermittlung der Ereignisteildaten zu Ereignis n an den Eventbuilder zu

$$d_{K,n} := t_{K,n} - t_{Trig,n}$$

Weiter kann durch Betrachtung von

$$\Delta t_{K,n} := t_{K,n} - t_{K,n-1}$$

festgestellt werden, ob für ein Komponentensubsystem eine untere Grenze für die Zeit existiert, mit der es aufeinanderfolgend Daten liefern kann. Diese Größe ist insbesondere bei starken Schwankungen in der GSLT-Frequenz interessant, weil bei großem $\Delta t_{K,n}$ und kleinem $\Delta t_{Trig,n}$ der Wert von $d_{K,n}$ und damit der benötigte Pufferspeicher stark zunimmt.

• Datendurchsatz an den Komponentenschnittstellen

Für die Beurteilung des Leistungsvermögens der Komponentenschnittstellen ist deren Datendurchsatz D_K definiert als

$$D_K := \frac{\sum_{n \in \{n | t_{K,n} \in [t_0, t_1]\}} L_{K,n}}{t_1 - t_0}$$

• Füllzustand des Übergabepuffers der Komponentensubsysteme

Wie im vorigen Kapitel erläutert, bestehen die Schnittstellen zu den Komponentensubsystemen aus einem Ringpuffer, der sich im TPM eines 2TP-Moduls befindet. Der prozentuale Füllzustand $P_{K,n}$ dieses Puffers nach Einlesen der Ereignisteildaten von Ereignis n ist ein Maß dafür, ob Daten schnell genug von den Komponentenschnittstellen abgenommen werden können.

• Füllzustand des Übergabepuffers zum TLT

Auch die Schnittstellen zum TLT sind als Ringpuffer im TPM von 2TP-Modulen realisiert. Der Füllzustand $P_{TLT,n}$ des Ringpuffers in Prozent nach Übergabe von Ereignisdaten n an den TLT ist nun ein Maß dafür, ob der TLT Daten schnell genug auslesen und verarbeiten kann.

Die Definitionen der vorgestellten Kenngrößen für externe Systeme sind in Tabelle 3.1 noch einmal zusammengefaßt. Die Tabelle zeigt außerdem, inwieweit diese Größen einer direkten Messung zugänglich sind.

3.3 Kenngrößen der internen Funktionseinheiten

Die Funktionseinheiten des Eventbuilders wurden im vorherigen Kapitel erläutert. Folgende Kenngrößen sind für die Überwachung dieser Funktionseinheiten nötig:

- **Pufferauslastung der Komponentenschnittstellen**
Die Komponentenschnittstellen kopieren die Ereignisteildaten zunächst in einen internen Pufferspeicher, bis eine Linkzuteilung durch die Steuereinheit erfolgt. Die prozentuale Pufferauslastung $I_{K,n}$ nach Einlesen der Ereignisteildaten von Ereignis n ist damit ein Maß dafür, ob die Linkzuteilung und der anschließende Transfer der Daten zu den TLT-Schnittstellen schnell genug erfolgt.
- **Pufferauslastung der TLT-Schnittstellen**
Ereignisteildaten werden von den Komponentenschnittstellen in den internen Puffer der TLT-Schnittstellen übertragen. Dort verbleiben sie, bis alle Teildaten eines Ereignisses vorliegen, die „ADAMOisierung“ abgeschlossen und im TPM genügend Platz für die Übergabe der Daten an den TLT vorhanden ist.
Die prozentuale Pufferauslastung $I_{TLT,n}$ des TLT-Schnittstellentransputers TLT_n nach Transfer der letzten Ereignisteildaten von Ereignis n ist somit ein Maß dafür, ob dieser Vorgang schnell genug erfolgen kann.

- **Transferrate auf den Datenübertragungslinks**
Die Transferrate auf den Datenübertragungslinks gibt an, wieviel Daten pro Zeiteinheit über einen Link übertragen werden könnten, d. h. die maximal erreichbare Transfergeschwindigkeit von einer Komponentenschnittstelle zu einer TLT-Schnittstelle. Sei $j \in \{0, 1, 2\}$ die Nummer des Links der Komponentenschnittstelle, über den ein Transfer erfolgt⁴ und k gebe an, um den wievielten Transfer auf diesem Link es sich handelt. Weiter bezeichne $n(K, j, k)$ die Nummer des Ereignisses, dessen Teildaten im k -ten Transfer auf Link j der Komponentenschnittstelle K übertragen wurden. Mit $t_{K,j,k}^{Start}$ als dem Startzeitpunkt und $t_{K,j,k}^{Stop}$ als dem Endzeitpunkt dieses Transfers errechnet sich die Transferrate für diese Ereignisteildaten dann zu

$$T_{K,j,k} := \frac{L_{K,n(K,j,k)}}{t_{K,j,k}^{Stop} - t_{K,j,k}^{Start}}$$

- **Schaltverzögerung durch die Steuereinheit**
Die effektive Transferrate auf den Datenübertragungslinks wird niedriger als die maximale Transferrate sein, weil, bedingt durch Verzögerungen innerhalb der Steuereinheit, Links nach Ende eines Datentransfers nicht sofort für einen weiteren Transfer genutzt werden können; vielmehr muß zunächst eine „Link-Rückgabe“-Meldung an die Steuereinheit und anschließend eine „Link-Zuteilung“-Meldung von der Steuereinheit erfolgen. Die Zeitspanne $t_{K,j,k}^{frei}$, die der Link j vor dem k -ten Transfer ungenutzt war, ist

$$t_{K,j,k}^{frei} := t_{K,j,k}^{Start} - t_{K,j,k-1}^{Stop}$$

- **Ereignisverteilung auf die TLT-Schnittstellen**
Die Verteilung der Ereignisteildaten auf die verschiedenen TLT-Transputer sollte

⁴Von den vier Links einer Komponentenschnittstelle werden nur drei für den Datentransfer verwendet, ein Link wird für die Verbindung zur Steuereinheit benötigt.

Größe	Messung bzw. Berechnung	Bedeutung
$t_{Trig,n}$	Meßgröße	Zeitpunkt des Eintreffens der positiven Triggerentscheidung für Ereignis n .
f_{Trig}	$\frac{ \{n t_{Trig} \in [t_0, t_1]\} }{t_1 - t_0}$	GSLT-Frequenz im Intervall $[t_0, t_1]$.
$\Delta t_{Trig,n}$	$t_{Trig,n} - t_{Trig,n-1}$	Abstand zweier positiver GSLT-Entscheidungen.
$t_{Verteil}$	Systemparameter	Obere Grenze für die Zeit, innerhalb derer ein Komponentensubsystem den Empfang eines Bytes einer GSLT-Entscheidung zu quittieren hat.
$t_{K,n}$	Meßgröße	Zeitpunkt des Eintreffens der Ereignisteildaten von Komponente K für Ereignis n .
$\Delta t_{K,n}$	$t_{K,n} - t_{K,n-1}$	Abstand zweier aufeinanderfolgender Ereignisteildaten.
$d_{K,n}$	$t_{Trig,n} - t_{K,n}$	Anwortzeiten der Komponentensubsysteme auf eine GSLT-Entscheidung.
$L_{K,n}$	Meßgröße	Länge der Ereignisteildaten von Komponente K zu Ereignis n .
D_K	$\frac{\sum_{n \in \{n t_{K,n} \in [t_0, t_1]\}} L_{K,n}}{t_1 - t_0}$	Datendurchsatz der Komponentenschnittstelle K während des Intervalls $[t_0, t_1]$.
$P_{K,n}$	Meßgröße	Prozentualer Füllzustand des TPM der Komponentenschnittstelle von Komponente K zum Zeitpunkt $t_{K,n}$.
$P_{TLT,n}$	Meßgröße	Prozentualer Füllzustand des TPM der TLT-Schnittstelle TLT_n nach Empfang der letzten Ereignisteildaten zu Ereignis n .

Tabelle 3.1: Symbolverzeichnis von Kenngrößen externer Systeme

möglichst gleichmäßig erfolgen, um alle vorhandenen Links optimal zu nutzen. Durch Aufzeichnung von $u_{TLT}(n)$ als der Nummer des TLT-Schnittstellentransputers, auf dem Ereignis n konstruiert wurde, kann diese Verteilung später untersucht werden. Für eine Untersuchung des Leistungsvermögens der Schnittstellen zum TLT sind — analog zum vorherigen Kapitel — die Ausgabefrequenz von Ereignisdaten auf einem TLT-Schnittstellentransputer, f_{TLTu} mit u als der Nummer des zu untersuchenden Transputers, sowie der Datendurchsatz zu diesem Knoten, D_{TLTu} , einzuführen. Sei zunächst

$$L_n = \sum_K L_{K,n}$$

die Länge der Ereignisdaten von Ereignis n . Weiterhin bezeichne

$$\mathcal{N}_{TLTu} = \{m | u_{TLT}(m) = u \wedge \inf(\{t_{K,j,k}^{Start} | n(K,j,k) = m\}) \in [t_0, t_1]\}$$

die Menge aller Ereignisse, die im Zeitintervall $[t_0, t_1[$ auf TLT-Schnittstellentransputer u konstruiert werden. Dann ist

$$f_{TLTu} = \frac{|\mathcal{N}_{TLTu}(t_0, t_1)|}{t_1 - t_0}$$

und

$$D_{TLTu} = \frac{\sum_{n \in \mathcal{N}_{TLTu}} L_n}{t_1 - t_0}.$$

Die verschiedenen, für die Untersuchung der internen Funktionseinheiten des Eventbuilders benötigten Kenngrößen sind mit ihrer Definition in Tabelle 3.2 nochmals zusammengefaßt.

3.4 Analyse von Kenn- und Überwachungsgrößen

Die in den vorherigen Abschnitten definierten Kenngrößen bieten vielfältige Analysemöglichkeiten. So liefern sie nicht nur Informationen über den momentanen Betriebszustand des Eventbuilders, sondern zeigen auch das Zusammenspiel des Eventbuilders mit den übrigen Komponenten des DAQ-Systems. Ebenso bilden sie die Grundlage einer Leistungsoptimierung des Eventbuilders. Dieser Abschnitt nennt verschiedene Bereiche, für die eine Analyse von Überwachungsgrößen benötigt wird.

3.4.1 Echtzeitüberwachung

Die Funktion des Eventbuilders ist das Konstruieren und Formatieren von Ereignisdaten. Entsprechend sind f_{TLTu} und D_{TLTu} diejenigen Kenngrößen, die während des Betriebes die ordnungsgemäße Funktion des Eventbuilders belegen können. Regelmäßige Ausgaben von Momentanwerten dieser Größen auf einem Terminal im Kontrollraum des Experiments erlauben es den dort anwesenden Operateuren bereits, sich davon zu überzeugen.

Im Fall einer Störung gilt es, die betreffende Fehlerquelle möglichst schnell auffindig zu machen. Hierfür ist die gezielte Abfrage von Kenngrößen — vornehmlich der Füllzustand verschiedener Puffer — von der Konsole des Systemspezialisten aus unerlässlich. Als Beispiel für die

Größe	Messung bzw. Berechnung	Bedeutung
$I_{K,n}$	Meßgröße	Prozentualer Füllzustand des internen Pufferspeichers der Komponentenschnittstelle von Komponente K zum Zeitpunkt $t_{K,n}$.
$I_{TLTu,n}$	Meßgröße	Prozentualer Füllzustand des internen Pufferspeichers des TLT-Schnittstellentransputers $TLTu$ nach Empfang der letzten Ereignisteil- daten zu Ereignis n .
$n(K,j,k)$	Meßgröße	Nummer des Ereignisses, welches im k -ten Transfer über Link j der Komponentenschnittstelle K übertragen wurde.
$t_{K,j,k}^{Start}$	Meßgröße	Beginn dieser Übertragung
$t_{K,j,k}^{Stop}$	Meßgröße	Ende dieser Übertragung
$t_{K,j,k}^{frei}$	$t_{K,j,k}^{Start} - t_{K,j,k-1}^{Stop}$	Zeitspanne, die Link j des Komponentenschnittstelle K vor dem k -ten Transfer ungenutzt war.
$T_{K,j,k}$	$\frac{L_{K,n(K,j,k)}}{t_{K,j,k}^{Stop} - t_{K,j,k}^{Start}}$	Transferrate von Ereignisteil- daten für Ereignis n von Komponenten K .
$u_{TLT}(n)$	Meßgröße	Nummer des TLT-Schnittstellentransputers, auf dem Ereignis n konstruiert wurde.
L_n	$\sum_K L_{K,n}$	Länge des Ereignisses n .
\mathcal{N}_{TLTu}	$\{m u_{TLT}(m) = m \wedge \inf(\{t_{K,j,k}^{Start} n(K,j,k) = m\}) \in [t_0, t_1]\}$	Menge aller Ereignisse, die im Intervall $[t_0, t_1[$ auf TLT-Schnittstellentransputer $TLTu$ konstruiert wurden.
f_{TLTu}	$\frac{ \mathcal{N} }{t_1 - t_0}$	Frequenz der Ereigniskonstruktion auf TLT-Schnittstellentransputer $TLTu$ während des Intervalls $[t_0, t_1[$.
D_{TLTu}	$\frac{\sum_{n \in \mathcal{N}_{TLTu}} L_n}{t_1 - t_0}$	Datendurchsatz am TLT-Schnittstellentransputer $TLTu$ während des Intervalls $[t_0, t_1[$.

Tabelle 3.2: Symbolverzeichnis von Kenngrößen der internen Funktionseinheiten des Eventbuilders

```

03/24/92 21:10:27 schli> status ctd
03/24/92 21:10:27 GSLT : internal component interface status
latest trigger      event # : 23833
latest data accepted event # : 23826
tpm head : 6272
tail : 3032
status input_component_data      = WAITING_FOR_INTERRUPT
status survey_component_interface = IDLE
internal data buffer : 4 of 16 pages allocated to 4 events

```

Abbildung 3.1: Beispiel für eine Textmeldung an der Systemspezialistenschnittstelle. Hier: Abfrage des Status der Komponentenschnittstelle zu CTD.

Darstellung von Kenngrößen an der Konsole zeigt Abbildung 3.1, wie der Status einer Komponentenschnittstelle (hier CTD) auf Anfrage dem Operateur präsentiert wird: aufgeführt sind neben der Pufferauslastung $I_{K,n}$, auch die Nummer der letzten GSLT-Entscheidung und der letzten Ereignisteildaten, die dem Eventbuilder übermittelt wurden, sowie die Zustände der verschiedenen Prozesse, die auf dem betreffenden Schnittstellenrechner laufen.

3.4.2 Statistiken über den Eventbuilderbetrieb

Mittel- und langfristige Auswertungen des Eventbuilderbetriebes geben Aufschluß über die verschiedenen Betriebsituationen, mit denen der Eventbuilder am häufigsten konfrontiert wird und auf die hin er zu optimieren ist.

Die vom Eventbuilder zu erbringende Leistung wird vor allem durch die an seinen Eingängen auftretenden Datenströme bestimmt. Demnach wird eine Betriebsituation charakterisiert durch Größen wie $f_{Tr,i}$ und $L_{K,n}$, wobei im wesentlichen deren Mittelwerte und Schwankungen relevant sind. Eine mögliche Langzeitanalyse kann demnach durchgeführt werden, wenn für jede Messung die Größen $f_{Tr,i}$ und $L_{K,n}$ (eventuell auch L_n) histogrammiert und dann deren Mittelwerte, Schwankungen, Maxima und Minima aufgezeichnet werden.

Ein Beispiel für solche Histogramme zeigt Abbildung 3.2. Dort sind während eines Testbetriebes des ZEUS-Detektors (vgl. nächstes Kapitel) die Längen der Ereignisteildaten der Detektorkomponenten BMUO (Barrel-Myon-Detektor) und GSLT (Zweite Triggerstufe), $L_{BMUO,n}$ und $L_{GSLT,n}$ überwacht worden. Während GSLT annähernd eine stets gleiche Menge an Daten an den Eventbuilder übertrug, variierte die Länge der Ereignisteildaten von BMUO in einem weiten Bereich. Mit der Kenntnis dieser Verteilung können nun interne Betriebsparameter wie z. B. die Größe und Aufteilung interner Pufferspeicher optimal gewählt und angepaßt werden.

3.4.3 Leistungsoptimierung des Eventbuilders

Ziel der Analyse des Eventbuilderbetriebes ist es, ihn an die normale Betriebsituation im Trigger- und Datenerfassungssystem optimal anzupassen. Um dies erreichen zu können, ist es notwendig, die Abläufe innerhalb des Eventbuilders zu verstehen. So kann man herausfinden, welche Größen die Leistung des Eventbuilders momentan begrenzen und wie er zu manipulieren ist, damit die Leistung gesteigert wird.

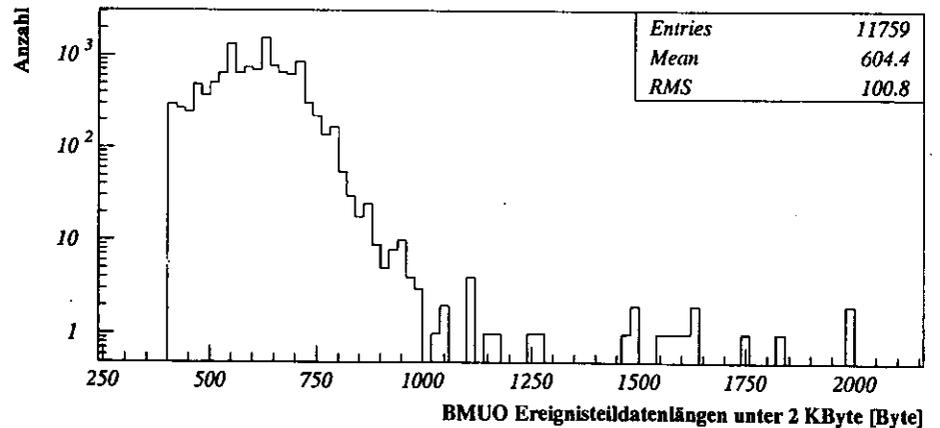
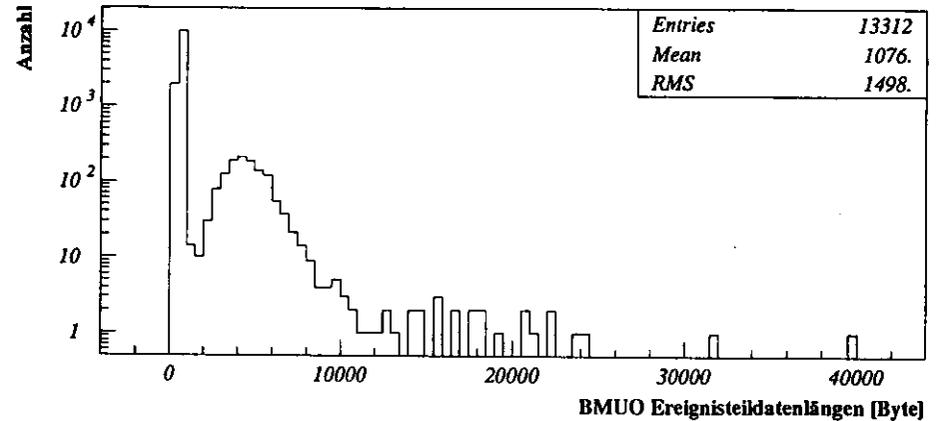
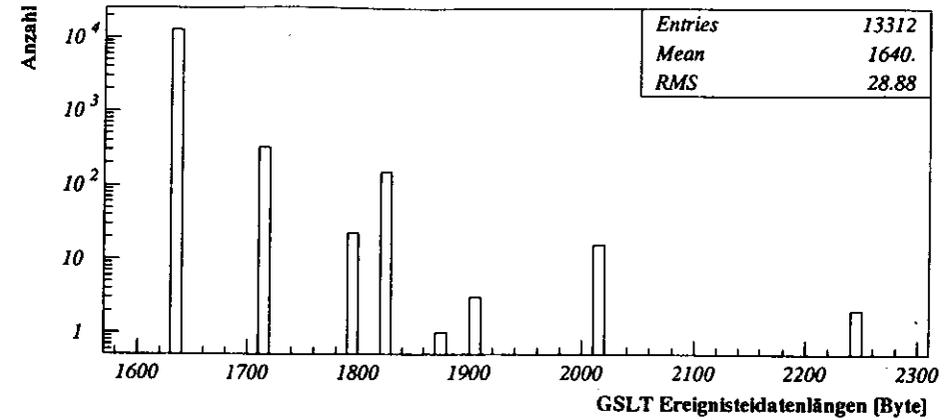


Abbildung 3.2: Länge der Ereignisteildaten von GSLT und BMUO, gemessen während Run 1033.

Maßgeblich für das Leistungsvermögen des Eventbuilders ist die Rate, mit der er vollständig formatierte Ereignisse dem TLT übermitteln kann. Um hier eine hohe Ausgabefrequenz erreichen zu können, ist es notwendig, daß die Datenübertragungskapazitäten innerhalb des Eventbuilders größer sind als die angestrebte Ausgangsrate, d. h.

$$\sum_K \bar{T}_K > \sum_u \bar{D}_{TLTu}$$

Für die Leistungsoptimierung ist es daher wichtig zu untersuchen inwieweit D_{TLTu} oder T_K von der Betriebssituation des Eventbuilders abhängen und wo ihre systembedingten Grenzwerte liegen. Analysen, die sich mit einer Optimierung befassen, werden im nächsten Kapitel vorgestellt. Soweit möglich wurde dabei auf Daten, die während eines Testbetriebes des ZEUS-Experiments aufgezeichnet wurden, zurückgegriffen. Wenn nötig wurden diese Daten noch durch Simulationen ergänzt.

3.4.4 Leistungsgrenzen des DAQ-Systems

Bedingt durch seine zentrale Position innerhalb des Trigger- und Datenerfassungssystems kann der Eventbuilder ermitteln, wie die verschiedenen Bestandteile dieses Systems auf schwankende Betriebsbedingungen reagieren. Interessant ist hier vor allem, inwieweit kurzzeitige Steigerungen der Triggerrate von den Komponentensubsystemen verkraftet werden oder ob anhand dieser Schwankungen möglicherweise obere Schranken für die Frequenz des Trigger- und Datenerfassungssystems ermittelt werden können.

Im nächsten Kapitel werden daher auch Untersuchungen vorgestellt, die die Ausgangsfrequenzen der verschiedenen Komponentensubsysteme in Abhängigkeit von der Triggerfrequenz analysieren.

Kapitel 4

Verhalten des Eventbuilders im Probetrieb des ZEUS-Detektors

Im Zuge der Inbetriebnahme des ZEUS-Detektors sowie seines Trigger- und Datenerfassungssystems werden verschiedene Systemtests durchgeführt. Jede Inbetriebnahme der Auslese- und Aufzeichnungskette von den entsprechenden Detektorcomponenten bis hin zum DESY-Rechenzentrum wird als „Run“ bezeichnet und fortlaufend durchnummeriert. Die „Runnummer“ dient bei der späteren Auswertung zur eindeutigen Identifikation der Experimentbedingungen und wird auch im folgenden Kapitel mehrfach Verwendung finden.

In der Woche vom 2. bis zum 8. Dezember 1991 fand zum zweiten Mal ein intensiver Systemtest im Schichtbetrieb rund um die Uhr statt, in dem neben Test- und Kalibrationsdaten im wesentlichen Daten von der Wechselwirkung kosmischer Strahlung mit dem Detektor aufgezeichnet wurden. Nach einer kurzen Beschreibung über die Entstehung und Zusammensetzung kosmischer Strahlung werden in diesem Kapitel einige der durch die Auswertung der Daten erhaltene Bilder der rekonstruierten Wechselwirkungen im Detektor vorgestellt.

An dieser Datenerfassung war auch der Eventbuilder beteiligt, und das mit seiner Hilfe in dieser Woche ermittelte Verhalten des Trigger- und Datenerfassungssystems wird im folgenden vorgestellt. Da auf Grundlage dieser Daten allein noch keine Aussage über die Grenzen der Leistungsfähigkeit des Eventbuilders getroffen werden kann, folgt ein Abschnitt, in dem die durch simulierte Belastungen bestimmten Grenzen der Leistungsfähigkeit des Eventbuilders diskutiert werden.

4.1 Kosmische Strahlung

Die im Probetrieb des Detektors nachgewiesenen Teilchen haben ihren Ursprung in Schauerprozessen in der Erdatmosphäre. Diese Schauer entstehen, wenn hochenergetische Teilchen aus dem Weltall in den höheren Schichten der Erdatmosphäre auf Luftmoleküle treffen. Diese sog. Primärteilchen, meist Protonen mit einer Energie von bis zu 10^{11} GeV , lösen durch Wechselwirkung mit den Kernen der Moleküle einen Kaskadenprozeß aus, bei dem aus dem getroffenen Kern eine große Anzahl von Sekundärteilchen freigesetzt werden. Sowohl das Primärteilchen, das noch einen großen Teil seiner ursprünglichen Energie besitzt, als auch die Sekundärteilchen, im wesentlichen Protonen, Neutronen und Pionen, treffen auf weitere Kerne, so daß sich der Prozeß wiederholt und sich die Anzahl der Teilchen im Schauer schnell erhöht.

Die entstandenen ungeladenen Pionen zerfallen nach einer mittleren Lebensdauer von 10^{-16} s ,

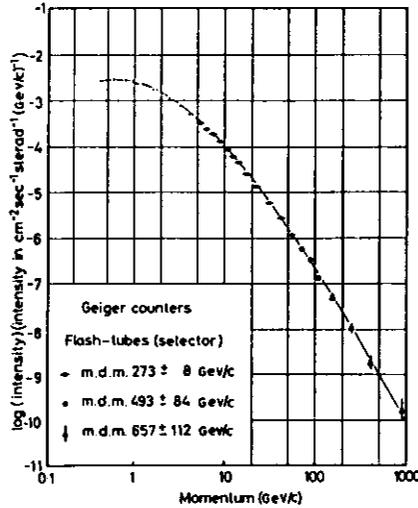
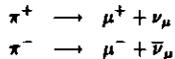


Abbildung 4.1: Differentielles Impulsspektrum kosmischer Myonen auf Meereshöhe [HAY62]

in zwei γ -Quanten, die Ursache für die elektromagnetische Komponente des Schauers sind: durch Paarerzeugung entstehen aus den γ -Quanten Elektron-Positron-Paare, die wiederum so hochenergetisch sind, daß sie ihre Energie bevorzugt durch Bremsstrahlung abgeben. Die so entstandenen γ -Quanten können diesen Prozeß wiederholen, wodurch sich die Anzahl der Photonen, Elektronen und Positronen schnell vervielfacht. Werden die ebenfalls entstandenen geladenen Pionen nicht innerhalb ihrer mittleren Lebensdauer von $\approx 10^{-8}$ s absorbiert, zerfallen sie nach folgendem Schema:



Der hierdurch entstehende Myonenanteil des Schauers erhöht sich mit zunehmendem Alter des Schauers, bis schließlich auf Meereshöhe 75 % der Teilchen Myonen sind. Dies ist darauf zurückzuführen, daß sich Myonen in ihrem Energieabgabeverhalten deutlich von den anderen Schauerkomponenten unterscheiden. Während die Lufthülle ein Target von 12 Kernwechsellängens darstellt [Gruppen85] und damit hadronische und elektromagnetische Schauer von einigen Hundert GeV absorbiert werden, geben Myonen nur wenig Energie ab. Näherungsweise gilt [PDG90]:

$$-\frac{dE}{dx} = a(E) + b(E) \cdot E$$

$a(E)$ steht für die Verluste durch Ionisation, die durch die Bethe-Bloch-Formel beschrieben werden können, in $b(E) \cdot E$ sind die Verluste durch Bremsstrahlung, Elektron-Positron-Paarerzeugung und photonukleare Reaktionen zusammengefaßt. Berücksichtigt man die mittlere Energie der Myonen von 2 GeV und das differentielle Spektrum, das mit E^{-2} fällt (Abbildung 4.1), so läßt sich feststellen, daß die Verluste durch Bremsstrahlung vernachlässigbar

sind. Durch ihre wesentlich größere Masse verlieren Myonen sehr viel weniger Energie durch Bremsstrahlung als Elektronen. Erst mit zunehmender Energie oberhalb einiger hundert GeV erreichen die Verluste durch Bremsstrahlung die Größenordnung der Verluste durch Ionisation. Dasselbe gilt für Energieverluste durch Elektron-Positron-Paarerzeugung und photonukleare Reaktionen: auch sie erreichen relevante Größenordnungen erst deutlich jenseits der mittleren Energie von kosmischen Myonen. Somit kann der Term $b(E)$ für kosmische Myonen vernachlässigt werden, und ausschließlich die Verluste durch Ionisation müssen betrachtet werden.

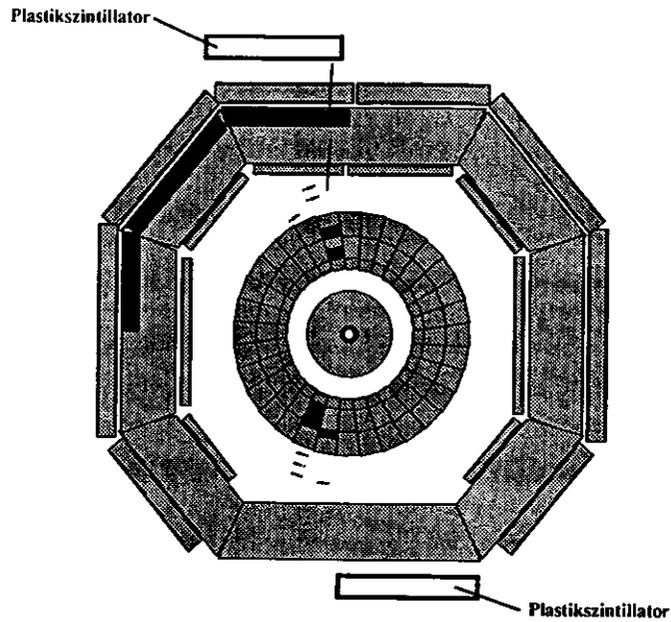
Die meisten kosmischen Myonen liegen energetisch im Bereich des breiten Minimums der Bethe-Bloch-Funktion. Dies bedeutet, daß sich kosmische Myonen annähernd wie minimal ionisierende Teilchen verhalten: mit konstantem, minimalem Energieverlust dE/dx passieren sie auch massive Uran- und Eisen-Blöcke. Durchqueren kosmische Myonen den ZEUS-Detektor, werden sie mit annähernd gleicher Energie und annähernd gleichem Energieverlust dE/dx in allen auf ihrem Weg liegenden Komponenten beobachtbar sein. Somit eignen sie sich nicht nur zur Kalibrierung einzelner Komponenten, sondern auch zum Test des Gesamtsystems.

4.2 Ergebnisse des Funktionstests des ZEUS-Detektors

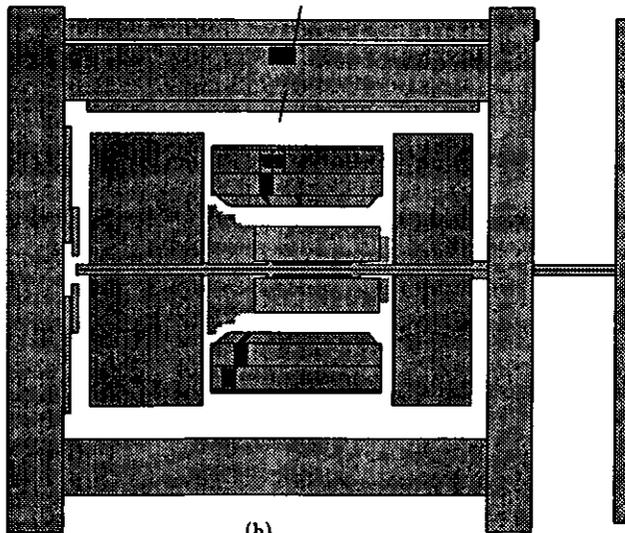
Während des Probetriebes gelang es, eine Vielzahl von Daten über im Detektor registrierte kosmische Myonen zum DESY-Rechenzentrum zu übermitteln und dort aufzuzeichnen.

Da das Registrieren kosmischer Strahlung nicht zu den späteren Aufgaben des Detektors gehört, war eine Erweiterung im Triggersystem nötig. Hierzu wurden über dem Detektor mehrere Schichten von Plastikszintillatoren angebracht und in Koinzidenz geschaltet (Vergleiche Abbildung 4.2). Das resultierende Koinzidenzsignal wurde dann der ersten Triggerstufe zur Verfügung gestellt, welche dieses mit den unterschiedlichen Triggersignalen der am Probetrieb teilnehmenden Detektorkomponenten verknüpfte. Ein weiterer Plastikszintillator befand sich unter dem Detektor und lieferte ebenfalls Signale an die erste Triggerstufe. Da die erste Triggerstufe variabel konfiguriert werden kann, wurde auf verschiedene Bedingungen hin getriggert, so z. B. auf ein Signal des oberen Szintillators verbunden mit einem Signal im Kalorimeter.

Die so erhaltenen Daten wurden von den Komponentensubsystemen über den Eventbuilder und die dritte Triggerstufe ins DESY-Rechenzentrum transferiert und später mit Hilfe der ZEUS-Analysesoftware ausgewertet. Ein Beispiel einer graphischen Darstellung zeigt Abbildung 4.2. Das Myon drang von oben in den äußeren Barrel-Muon-Detektor ein. Die Spur wurde rekonstruiert und als Gerade eingezeichnet. Sodann passierte es das Backing-Kalorimeter, wo noch keine Spurrekonstruktion stattfand. Zur Kennzeichnung wurde deshalb die gesamte getroffene Kammer, die im Querschnitt ein Viertel des Detektors abdeckt, schwarz eingefärbt. Weiter durchquerte das Myon den inneren Barrel-Muon-Detektor, wobei die Spur wieder teilrekonstruiert und als Gerade dargestellt wird. Schließlich gelangte es in die Sektionen des Barrel-Kalorimeters. In der Darstellung wurden die getroffenen Sektionen proportional zur Energiedeposition mit schwarzen Rechteckflächen gekennzeichnet. Zu beachten ist bei den in Abbildung 4.2(a) und 4.2(b) dargestellten Energiewerten, daß es sich nicht um Beträge an bestimmten Längs- oder Querschnitten durch den Detektor handelt, sondern vielmehr um Summen über alle Barrel-Kalorimetersektionen entlang der Strahlachse (4.2(a)) bzw. über alle Sektionen ober- und unterhalb des Strahlrohres (4.2(b)).



(a)



(b)

Abbildung 4.2: Mit Hilfe der ZEUS-Analyseprogramme sichtbar gemachte Energieeinträge kosmischer Myonen. [Leich92]

4.3 Ergebnisse der Eventbuilderüberwachung

4.3.1 Implementation der Kenngrößenmessung

Alle Kenngrößen bzw. die zu ihrer Bestimmung benötigten Informationen werden auf den Komponentenschnittstellen sowie dem GSLT-Verteilertransputer und den TLT-Schnittstellentransputern gemessen. Bei deren Erfassung ist zu beachten, daß diese den Betriebsablauf des Eventbuilders so wenig wie möglich beeinflusst, um die Kenngrößen nicht zu verfälschen oder die Leistung des Eventbuilders zu verschlechtern. Aus diesem Grund wurde die Erfassung aller Daten ereignisorientiert aufgebaut, d.h. es werden keine Informationen in konstanten Zeitintervallen gesammelt, sondern Daten werden nur zu Vorgängen gespeichert, die sich auf ein bestimmtes Ereignis beziehen. So werden z. B. Pufferauslastungen (P_K, I_K, \dots) nur registriert, wenn in den betreffenden Speicher neue Daten geschrieben werden.

Um die Informationen, die auf verschiedenen Transputern vorliegen, zusammenzuführen, wurde der in Abschnitt 2.3.5 vorgestellte Mixer-Transputer ausgewählt, da er sich an einem zentralen Punkt im Eventbuilder befindet und alle Meldungen, die beim Betrieb auftreten, durch ihn geleitet werden. Da das Auftreten dieser Meldungen ebenso ereignisorientiert ist wie das Erfassen von Überwachungsgrößen, bietet es sich an, die ohnehin ausgetauschten Meldungen um einige Bytes zu erweitern. Die gewonnen Überwachungsdaten können dann einfach in diese Meldungen mit aufgenommen und übertragen werden, ohne das zusätzlicher, das System belastender Nachrichtenaustausch anfällt. Als Beispiel sei hier das Protokoll der Link-Rückgabe-Meldung aufgeführt:

Struktur

Erläuterung

```
typedef struct {
    int TLT_node_id;
    int link_no;
    int internal_buffer_load;
    int head_tail_diff;
    int event_output;
    int released_buffer;
} TLT_NODE_PERFORMANCE;
```

Dieses sind die Überwachungsinformationen einer TLT-Schnittstelle

u
P_{TLT}
I_{TLT}

```
typedef struct {
    int data_length;
    int head_tail_diff;
    int internal_buffer_usage;
    struct {
        int event_from_tpm;
        int link_request_sent;
        int link_ready_received;
        int link_release_sent;
    } time_of;
    int comp_link_no;
    int tlt_link_no;
} COMP_IF_MONITORING_DATA;
```

Dieses sind die Überwachungsinformationen einer Komponentenschnittstelle

I_k
P_k
I_k

I_{k,p}

{Start
{Stop

```
typedef struct {
    int length;
    int tag;
    int component_id;
    int event_no;
    int component_link_no;
    int TLT_node_id;
    int TLT_buffer_load;
} LINK_RELEASE_MESSAGE;
```

} Dieser Teil der Meldung wird zum Transport im Eventbuilder benötigt

} Dieser Teil der Meldung enthält die Informationen der Link-Freigabemeldung

```
typedef struct {
    LINK_RELEASE_MESSAGE release;
    COMP_IF_MONITORING_DATA monitoring_data;
    TLT_NODE_PERFORMANCE tlt_node_performance;
} LINK_RELEASE_MONITORING_MESSAGE;
```

Dieser Teil wird zum Supervisor weitergeleitet

} Dieser Teil der Meldung verbleibt im Mixer

Abbildung 4.3: Die Struktur der Überwachungsinformationen

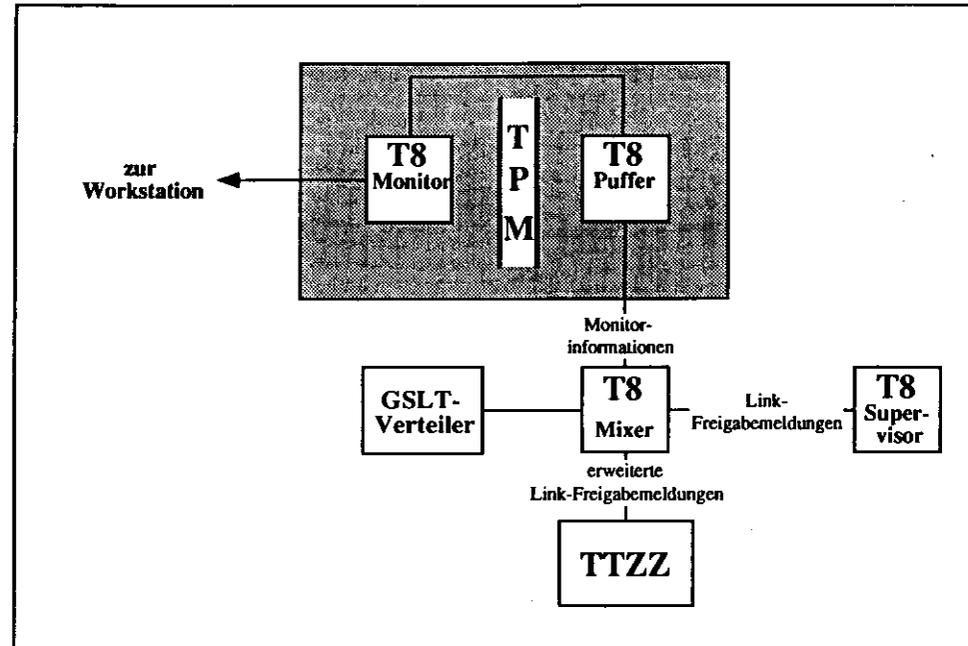


Abbildung 4.4: Einbettung der Überwachungs-Transputer in den Eventbuilder (Vergleiche Abschnitt 2.3).

Die Überwachungsinformationen verbleiben stets im Mixer, sie werden, um die Länge der Meldungen möglichst kurz zu halten, vor der Weitersendung an die Komponentenschnittstellen oder den Supervisor entfernt.

Um die Rechenzeit- und Speicherbelastung der Eventbuilder-Transputer durch die Auswertung der Überwachungsdaten möglichst gering zu halten, wurde für die Überwachung ein zusätzliches 2TP-Modul installiert. Abbildung 4.4 zeigt, wie es mit dem Mixer verbunden ist. Von den beiden Transputern des 2TP-Moduls wird einer als Puffer verwendet, der die aus dem Nachrichtenstrom herausgefilterten Daten empfängt und zur Auswertung bereithält. Der andere Transputer („Monitor“) kann auf diese gepufferten Daten zugreifen, um sie entweder „online“, d. h. sofort, zu analysieren oder aber um sie zur späteren Auswertung an die ebenfalls angeschlossene Workstation zu transferieren.

Bei dem Transfer vom Mixer zum Puffer ist zu beachten, daß zum Aufsetzen eines Transfers Rechenzeit benötigt wird, der Transfer selbst den Transputer aber kaum belastet¹. Um den Betriebsablauf im Mixer geringstmöglich zu behindern, gilt es also, Überwachungsdaten in

¹Transfers werden per „DMA“ („direct memory access“) durchgeführt, sind sie erst einmal gestartet, so können sie unabhängig von der betreffenden CPU abgewickelt werden.

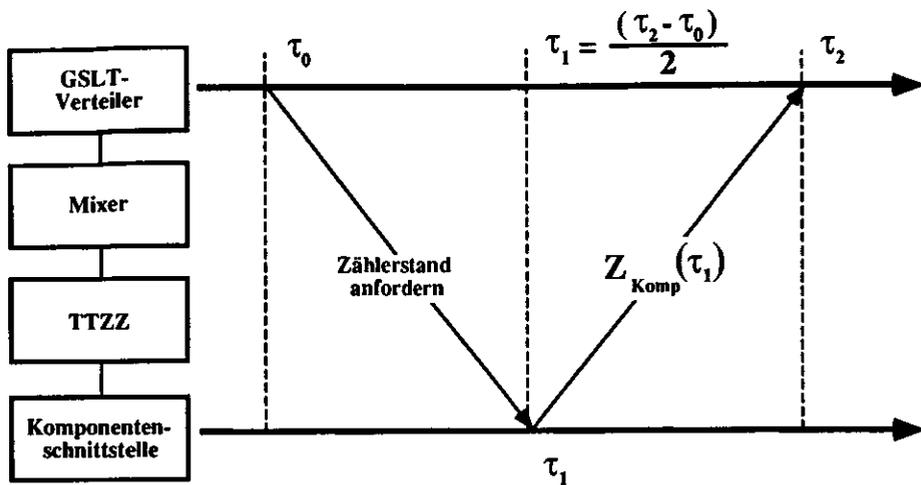


Abbildung 4.5: Zeiteichung der Transputer im Eventbuilder

möglichst großen Blöcken (und daher mit entsprechend wenigen Transfers) vom Mixer zum Puffer zu transferieren. Hierfür werden im Mixer zwei gleichartige Tabellen mit je 1024 Einträgen angelegt: eintreffende Überwachungsdaten werden dann zunächst in einer Tabelle abgelegt. Wenn diese voll ist, wird sie in einem Transfer zum Puffer übertragen, während neu eintreffende Überwachungsdaten in die andere Tabelle eingetragen werden. Die derzeitige Version der Eventbuilderüberwachung kann lediglich auf der Workstation Graphiken, die den Eventbuilderbetrieb charakterisieren, erzeugen. Für die Zukunft ist allerdings geplant, den Monitortransputer mit einer Grafikkarte und einem Bildschirm auszustatten, um dort Kenndaten direkt anzeigen zu können.

4.3.2 Eichung der Zeitmesser im Transputernetz

Die Zähler zur Zeitmessung verschiedener Transputer können nicht ohne weiteres miteinander verglichen werden, da zum einen ein gleichzeitiger Start bei null nicht gegeben ist, zum anderen Gangunterschiede durch unterschiedliche Schwingquarze auftreten können (Vergleiche Abschnitt 3.1).

Die Zähler des Transputernetzes des Eventbuilders sind also zu kalibrieren². Hierfür sind ihre Startwerte α_K zu einem gegebenen Zeitpunkt τ_0 sowie ihre Gangunterschiede ϵ_K zu bestimmen. Als Startpunkt τ_0 der Zeitmessung wird der Nullstand des Zählers auf dem Transputer des GSLT-Verteilers gewählt. Die Werte dieses Zählers werden auch für alle weiteren Zeitmessungen als maßgeblich betrachtet, so daß er die Absolutzeit τ innerhalb des Eventbuilders definiert. Bezeichnet $Z_K(\tau)$ den Zählerstand eines Transputers einer Komponentenschnitt-

² Da Zeitmessungen lediglich auf den Transputern der Komponentenschnittstellen und des GSLT-Verteilers durchgeführt werden, beschränkt sich auch die Kalibrierung auf diese Transputer.

stelle zur Zeit τ , $Z_0(\tau)$ den Zählerstand des Transputers des GSLT-Verteilers, so gilt unter der Annahme, daß sich die Zähler zweier Transputer nur um einen Anfangswert und einen konstanten Faktor unterscheiden:

$$Z_K(\tau) = (1 + \epsilon_K)Z_0(\tau) + \alpha_K$$

Da per Definition $Z_0(\tau) = \tau$, läßt sich so aus jeder Zeitmessung auf einem Komponentenschnittstellentransputer die Absolutzeit gemäß

$$\tau = \frac{Z_K(\tau) - \alpha_K}{1 + \epsilon_K}$$

rekonstruieren.

Für die Kalibrierung wurde ein Programm geschrieben, welches auf den Transputern des GSLT-Verteilers und der Komponentenschnittstellen läuft. Zu einem Zeitpunkt τ_0 wird vom GSLT-Verteiler eine Meldung durch den Mixer und den TTZZ an einen Komponentenschnittstellentransputer gesendet. Von diesem Transputer wird bei Empfang dieser Meldung zum Zeitpunkt τ_1 der Zählerstand $Z_K(\tau_1)$ an den GSLT-Verteiler zurückgesandt. Diese Meldung trifft dort zum Zeitpunkt τ_2 ein (Vergleiche Abbildung 4.5). Da der Transfer einer Meldung vom GSLT-Verteiler zur Komponentenschnittstelle genauso lange dauert wie der Transfer in umgekehrter Richtung, ist $\tau_1 = \frac{\tau_2 + \tau_0}{2}$. Damit ergibt sich α_K zu

$$\alpha_K = Z_K(\tau_1) - (1 + \epsilon_K)\tau_1 = Z_K(\tau_1) - (1 + \epsilon_K)\frac{\tau_2 + \tau_0}{2}$$

Wiederholt man diesen Vorgang zu späteren Zeitpunkten τ'_0, τ'_1 und τ'_2 , so läßt sich ϵ_K berechnen zu

$$\epsilon_K = \frac{Z_K(\tau'_1) - Z_K(\tau_1)}{\tau'_1 - \tau_1} - 1$$

Wird dieser Vorgang mehrmals für alle Komponentenschnittstellen und über einen längeren Zeitraum wiederholt, so kann man mit Hilfe einfacher linearer Regression aus den so gewonnenen Daten α_K und ϵ_K hinreichend genau bestimmen. Ein Ergebnis zeigt Tabelle 4.1. Zu beachten ist dabei, daß das Schwingverhalten der Quarze zur Taktfrequenzerzeugung und damit die Gangunterschiede der verschiedenen Zähler Z_K temperaturabhängig ist. Deshalb sollte die Bestimmung von α_K und ϵ_K stets unmittelbar vor oder nach dem zu untersuchenden Eventbuilderbetrieb durchgeführt werden.

4.3.3 Der Ausbauzustand des Eventbuilders im Testbetrieb

Viele Bestandteile des ZEUS-Detektors und seines Trigger- und Datenerfassungssystems hatten zum Zeitpunkt des einwöchigen Testbetriebes im Dezember 1991 noch nicht ihre endgültige Ausbaustufe erreicht. So waren von den 17 Komponentensubsystemen, die Daten an den Eventbuilder liefern werden, erst sieben einsatzbereit, nämlich der 2nd Level Trigger (GSLT), das Barrel-, Rückwärts- und Vorwärts-Kalorimeter (BCAL, RCAL, FCAL), die zentrale Spurkammer (CTD), das Backing-Kalorimeter (BAC) und der Barrel-Muon-Detektor (BMUO). Die Triggerrate $f_{Trigger}$, mit denen das Auslesesystem betrieben werden konnte, variierte zwischen 0.5 Hz und 6 Hz.

Der Eventbuilder selbst hatte ebenfalls noch nicht seine endgültige Ausbaustufe erreicht. So waren lediglich zwei von sechs TLT-Schnittstellen in Betrieb. Innerhalb dieser Schnittstellen wurde nur jeweils einer der beiden Transputer benutzt, und von den vier Links dieser Transputer wurde einer zur Ausgabe von Status- und Fehlermeldungen eingesetzt, so daß für den Datentransfer zum TLT nur jeweils drei, insgesamt also sechs Links eingesetzt werden konnten.

Komponentenschnittstelle	$\alpha_K [64\mu s]$	ϵ_K
GSLT	4,014	$7,653 \cdot 10^{-6}$
BCAL	3,038	$12,995 \cdot 10^{-6}$
RCAL	3,744	$1,562 \cdot 10^{-6}$
FCAL	3,591	$9,898 \cdot 10^{-6}$
CTD	4,100	$2,001 \cdot 10^{-6}$
BAC	2,672	$0,544 \cdot 10^{-6}$
BMUO	3,968	$8,856 \cdot 10^{-6}$
FMUO	2,716	$4,471 \cdot 10^{-6}$

Tabelle 4.1: Eichkonstanten für die Zeitmessung im Eventbuilder-Transputernetz (9.12.1991 12:23)

4.3.4 Messungen am Eventbuilder während des Testbetriebes

Im folgenden werden die Resultate der Eventbuilderüberwachung aus dem Testbetrieb vorgestellt. Dies wird exemplarisch an Run #1033 vorgeführt.

An diesem Run waren die Detektorkomponenten BCAL, RCAL, FCAL, BAC und BMUO beteiligt, und auch der GSLT lieferte Daten an den Eventbuilder. Dieser Run wurde ausgewählt, weil zum einen eine große Anzahl von Komponentensubsystemen an der Datenerfassung beteiligt war, der Betrieb andererseits über einen langen Zeitraum stabil aufrechterhalten werden konnte und daher der erwarteten Betriebsituation bei Beginn des echten Experimentbetriebes entspricht.

Während der an den Testbetrieb anschließenden weiteren Integration des Trigger- und Datenerfassungssystems trat während eines Probebetriebes eine Situation auf, die ebenfalls analysiert wurde (Run #1216). Sie zeigt, wie sich bei Blockade des TLT die verschiedenen Puffer des Eventbuilders füllen.

Beobachtungen an den externen Systemen

Abbildung 4.6 zeigt die Rate der positiven Entscheidungen des GSLT für Run #1033, jeweils gemittelt über 10s. Wie man erkennt, begann der GSLT 250s nach dem Booten des Transputernetzes mit dem Senden von Triggerentscheidungen. Diese lieferte er mit einer Rate von 5,3Hz (gemittelt über den ganzen Run) an den Eventbuilder. In der Abbildung erkennt man, dass diese Rate zwar schwankt, während der 45 min, die dieser Run dauerte, jedoch keine steigende oder fallende Tendenz aufwies.

Der zeitliche Abstand Δt_{Trig} , welcher den Abstand der einzelnen Entscheidungen voneinander angibt, wird in Abbildung 4.7 in Form eines Scatter-Plots dargestellt. Man erkennt, daß der Abstand keine periodischen oder anderweitig auffälligen Schwankungen aufweist. In Abbildung 4.8 ist die gesamte Häufigkeitsverteilung dieses Abstandes dargestellt, und Abbildung 4.9 zeigt den Bereich der geringsten Abstände noch einmal in feinerer Auflösung. Man erkennt, daß, obwohl die mittlere Frequenz 5,3Hz betrug, der Abstand in großem Maße variieren kann. Der minimale Abstand zweier Triggerentscheidungen in diesem Run betrug 2,5 ms, was einer Spitzenfrequenz von 400Hz entspricht.

In der feinen Auflösung erkennt man eine Lücke zwischen 4,8 und 6,25 ms. Diese Lücke wurde im Zuge des einwöchigen Testbetriebes des öfteren beobachtet. Nachdem der Eventbuilder eine GSLT-Entscheidung eingelesen hat, verteilt er diese an die Komponentensubsysteme

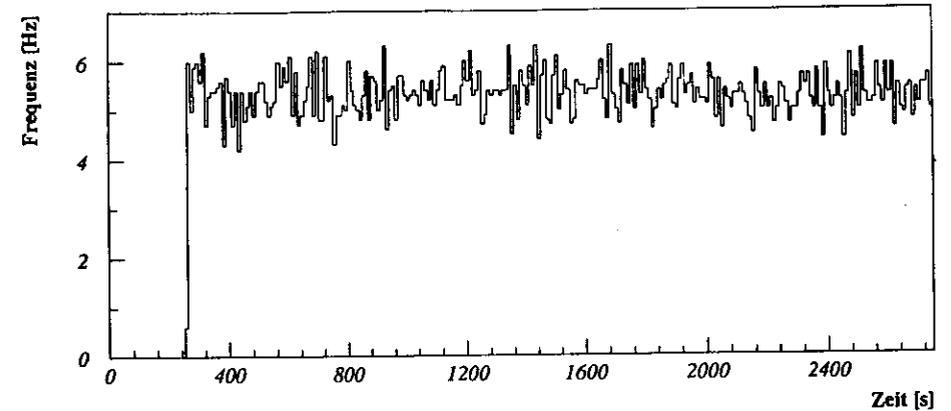


Abbildung 4.6: Die zeitliche Entwicklung der GSLT-Frequenz f_{Trig} für Run 1033, gemittelt über 10s.

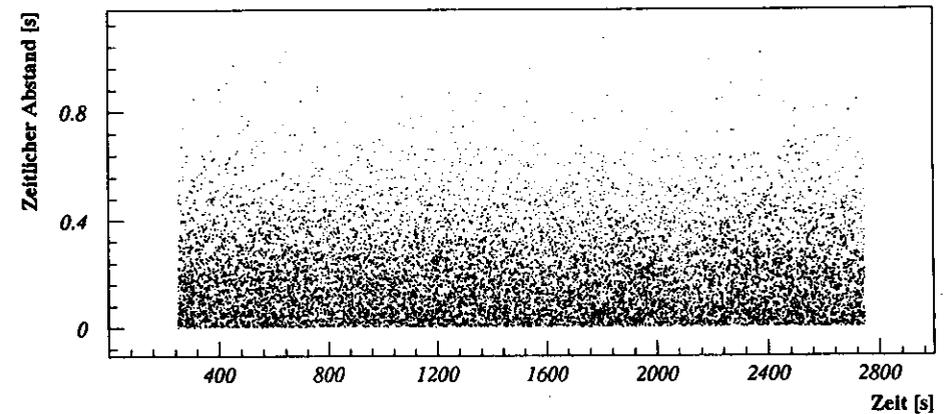


Abbildung 4.7: Die zeitliche Entwicklung von Δt_{Trig}

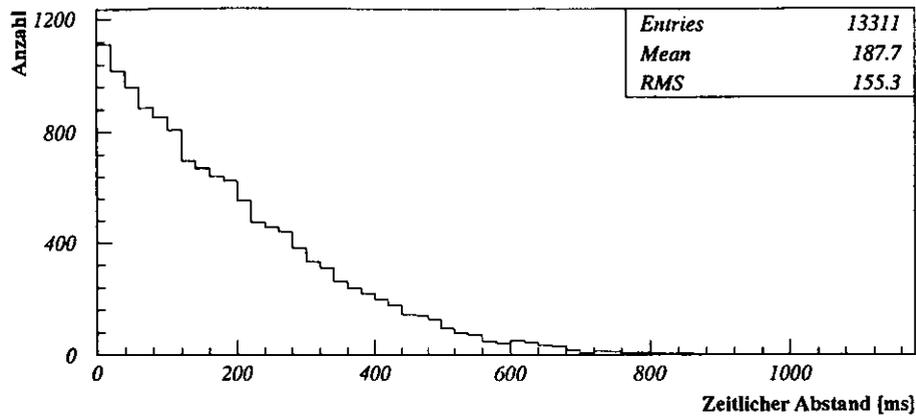


Abbildung 4.8: Die Verteilung von $\Delta t_{Trig,n}$

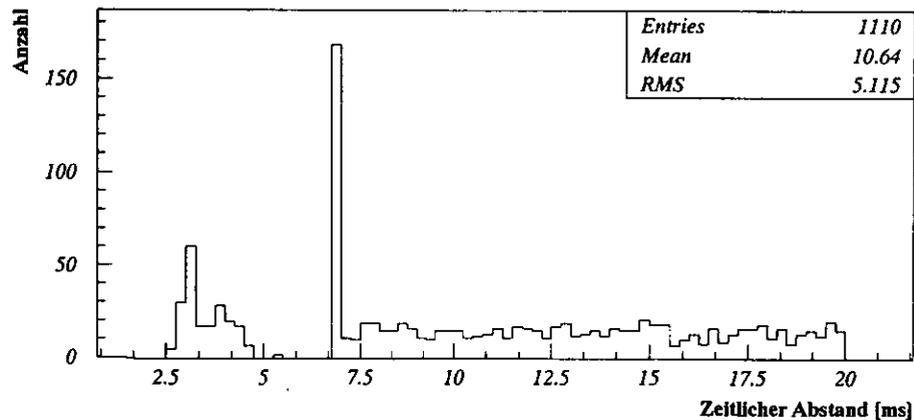


Abbildung 4.9: Die Verteilung der Abstände der positiven Triggerentscheidungen des GSLT $\Delta t_{Trig,n}$ die kleiner als 20 ms waren für Run 1033.

```
12/03/92 11:22:32 ERROR : SLT_FANOUT aborted; triggertype = c0000000
12/03/92 11:22:32 ERROR                                GFLT_no = -1
12/03/92 11:22:32 ERROR                                GSLT_no = 0
12/03/92 11:22:32 ERROR CTD excluded from run when not accepting SLT-decision
```

Abbildung 4.10: Beim Systemspezialisten und der zentralen Experimentsteuerung erscheinende Fehlermeldung, die angibt, daß ein Komponentensubsystem eine Triggerentscheidung nicht innerhalb von $t_{Verteil}$ quittiert hat.

Komponentensubsystem	Datenlänge in Bytes		
	Minimal	Maximal	Durchschnittlich
GSLT	1632	2240	1636
BCAL	42420	42420	42420
RCAL	14472	14472	14472
FCAL	44064	44064	44064
BAC	1472	1472	1472
BMUO	396	39788	976

Tabelle 4.2: Längen der einzelnen Ereignisteildaten von Run 1033

weiter. Daher wäre es denkbar, daß diese Lücke dadurch entsteht, daß einzelne Komponentensubsysteme eine Triggerentscheidung nur verzögert einlesen und dadurch den Eventbuilder am rechtzeitigen Einlesen der nächsten Entscheidung hindern. Da der Eventbuilder die Annahmezeit der einzelnen Komponentensubsysteme für diese Entscheidung jedoch überwacht, scheidet diese Möglichkeit aus und die Ursache für diese Lücke ist beim GSLT oder den ihm vorgelagerten Systemen zu suchen.

Während des Testbetriebes kam es einige Male vor, daß ein Komponentensubsystem die GSLT-Entscheidung nicht in der geforderten Abnahmezeit von $t_{Verteil} = 100\mu s$ abnehmen konnte. Dies wurde von der entsprechenden Hardware des Eventbuilders erkannt und erzeugte an der Schnittstelle zum Systemspezialisten und bei der zentralen Experimentsteuerung eine Fehlermeldung (Vergleiche Abbildung 4.10).

In der Meldung enthalten sind der Name des entsprechenden Komponentensubsystems und sowohl Entscheidungsnummer des GSLT als auch die des GFLT, da die Komponentensubsysteme auch die negative Triggerentscheidungen des GSLT innerhalb von $t_{Verteil}$ einlesen müssen. Da negative Triggerentscheidungen keine GSLT-Nummer besitzen, stellt die GFLT-Nummer oft die einzige Identifizierungsmöglichkeit der Triggerentscheidung dar.

Nachdem der Eventbuilder die Triggerentscheidungen an die Komponentensubsysteme verteilt hat, liefern diese für jede positive Entscheidung Daten an den Eventbuilder zurück. Abbildung 4.11 zeigt hier die Verteilungen des zeitlichen Abstandes d_K des Eintreffens der Daten zum Verteilen der entsprechenden Triggerentscheidung. Wie man erkennt, liefert der GSLT mit 6 ms am schnellsten Daten, während BCAL und FCAL mit 22 ms am längsten für das Abliefern der Daten benötigen. Dabei variieren die Komponentensubsysteme mit den größten Datenlängen, nämlich BCAL, RCAL und BMUO, (Vergleiche Tabelle 4.2) die Abgabezeit für die Ereignisteildaten. Besonders auffällig ist, daß BMUO die meisten Ereignisteildaten in weniger als 10 ms liefert, für einige wenige jedoch bis zu 60 ms benötigt. Wenn man zur Klärung dieses Verhaltens d_K über der Datenlänge aufträgt (Abbildung 4.12), erkennt man,

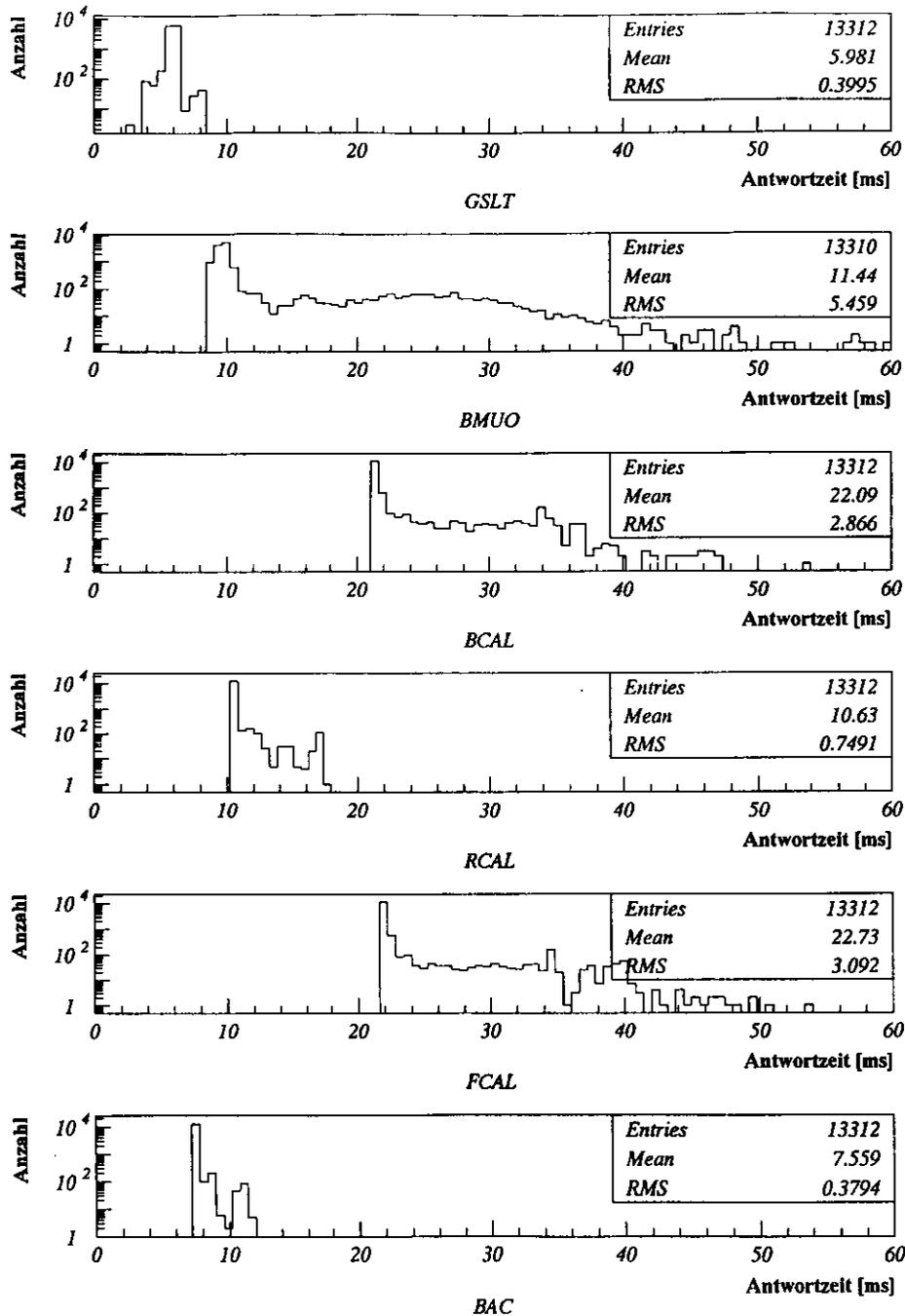


Abbildung 4.11: Verteilung von d_K für die am Run #1033 beteiligten Komponenten.

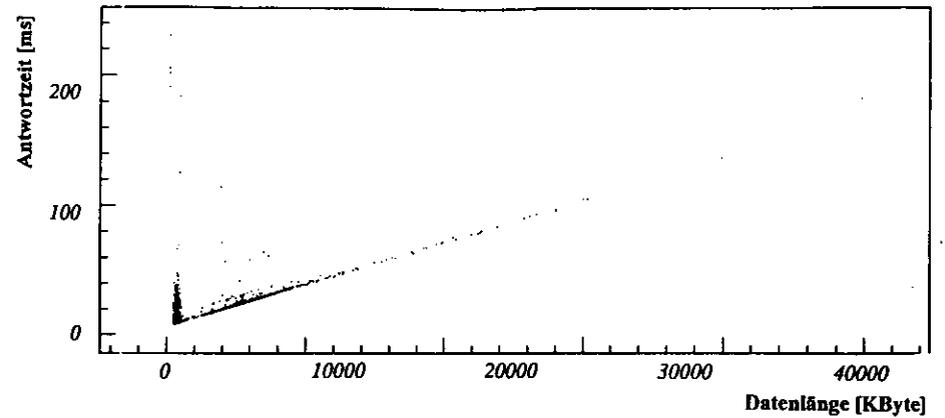


Abbildung 4.12: Antwortzeit von BMUO in Abhängigkeit der Ereignisteildatenlänge in Run #1033

daß für Datenlängen über 1600 Byte ein linearer Zusammenhang zwischen der Datenlänge und der Antwortzeit besteht. Für Datenlängen unter diesem Wert scheint dagegen kein Zusammenhang zwischen der Antwortzeit und der Datenlänge zu bestehen, da zu einer Datenlänge mehrere verschiedenen Antwortzeiten zu erkennen sind.

Wenn man die Datenlängen der verschiedenen Ereignisteildaten mit der Vorgabe zur Planung des Eventbuilders aus Tabelle 2.1 vergleicht, fällt auf, daß die Ereignisteildaten von BCAL, RCAL und FCAL ihre als maximale Länge bezeichnete Größe bereits um bis zu 10 KByte überschritten haben.

Aus der Information, wann Komponentensubsysteme ihre Daten abliefern, können weitere Schlüsse gezogen werden: Abbildung 4.13 trägt $\Delta t_{Trig,n}$ gegen $\Delta t_{K,n}$ auf, also den Abstand zweier positiver Triggerentscheidungen gegen den Abstand der dazugehörigen Ereignisteildaten. Grundsätzlich erwartet man, daß dieser Abstand gleich ist, d. h. eine Gerade mit der Steigung eins.

Sollte jedoch bei kurzen Triggerabständen Δt_{Trig} ein Komponentensubsystem dem GSLT nicht mehr folgen können, so wird in dem Diagramm eine untere Grenze Δt_K^{min} für den Abstand zweier Ereignisteildaten sichtbar werden, indem die Gerade bei Δt_K^{min} abknickt und in für kleinere Δt_K in eine Waagerechte übergeht. Nimmt Δt_K später wieder zu, so werden die „fehlenden“ Komponentendaten mit kurzfristig höherer Rate als f_{Trig} an den Eventbuilder übermittelt werden, was in dem Diagramm zu Einträgen unterhalb der Geraden führt.

Der Wert, den Δt_K^{min} für die einzelnen Komponenten annimmt, ist insofern kritisch, als die im Detektorbetrieb vorgesehene Rate positiver GSLT-Entscheidungen f_{Trig} von 100 Hz es erfordert, daß alle Komponenten ihre Ereignisteildaten in Abständen von weniger als 10 ms an den Eventbuilder übermitteln können. Die Abbildung zeigt, daß dies in dem beobachteten Betrieb nicht gegeben war, da FCAL und BCAL mit Abständen von jeweils 14 ms zu langsam waren. Die Triggerrate, die das Datenerfassungssystem tatsächlich bewältigen kann, läßt sich zu

$$f_{Trig}^{max} = \frac{1}{\sup(\{\Delta t_K^{min}\}_{alle K})}$$

errechnen und betrug im untersuchten Testbetrieb des Detektors etwa 71 Hz.

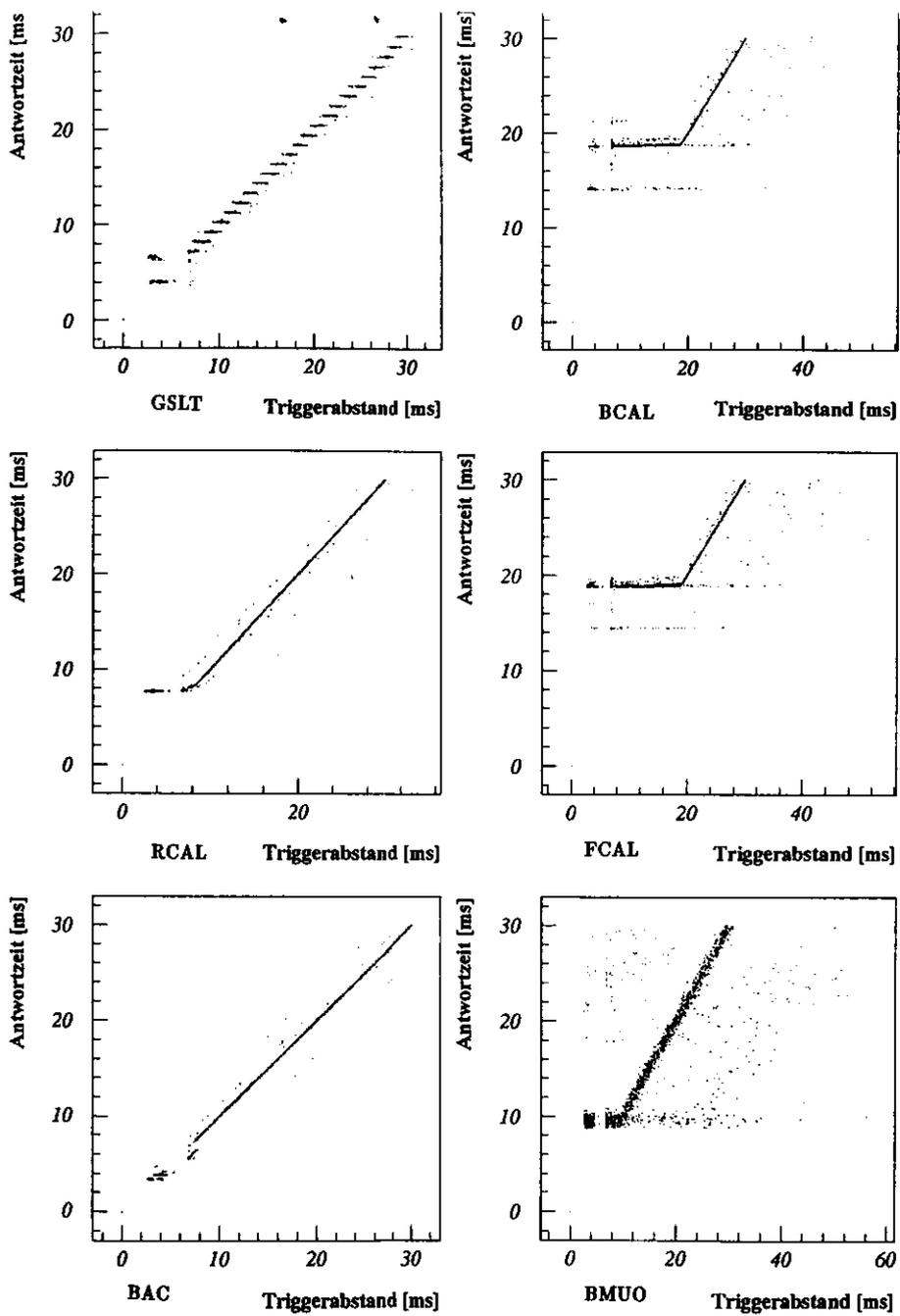


Abbildung 4.13: Der Abstand Δt_K aufeinander folgender Ereignisteildaten von verschiedenen Komponentensubsystemen aufgetragen gegen den Abstand Δt_{Trig} der entsprechenden Triggerentscheidungen.

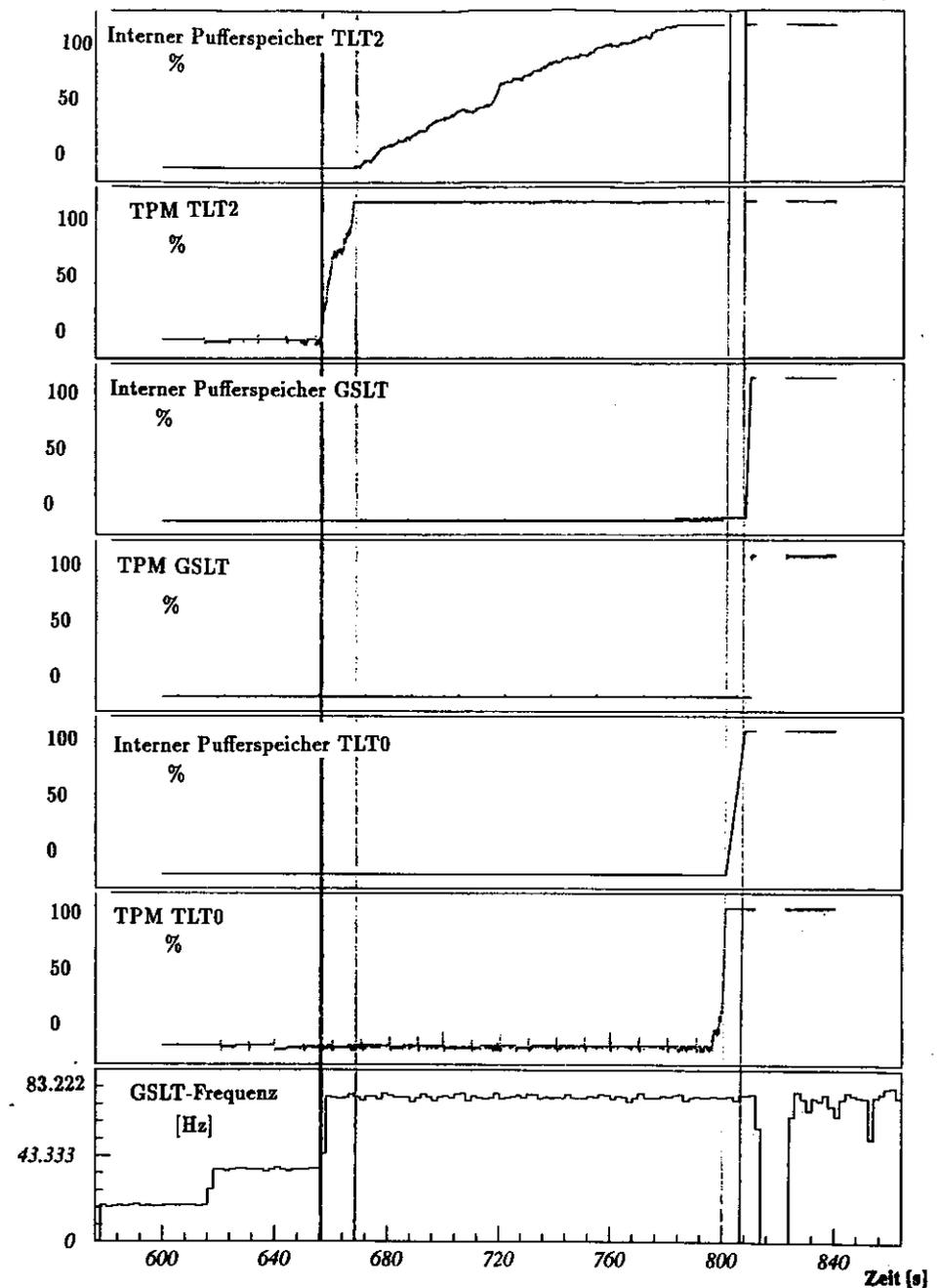


Abbildung 4.14: Der zeitliche Verlauf der Füllzustände der Puffer im Eventbuilder. (Erläuterung siehe Text)

Füllzustände der Puffer zur Erkennung von Betriebsstörungen

Während eines Systemtests, in dem absichtlich verschiedene äußere Parameter verändert wurden (Run #1216), kam es zu einem zwischenzeitigen Stillstand des Eventbuilderbetriebes. An dem Betrieb waren BAC und GSLT beteiligt, und der 3rd Level Trigger (TLT) übernahm an zwei Schnittstellen Daten vom Eventbuilder. Nachdem die GSLT-Frequenz f_{TLT} erhöht und die Datenabnahme D_{TLT} des TLT bewußt verringert wurde, füllten sich die verschiedenen Puffer innerhalb des Eventbuilders. Als die Puffer gefüllt waren, blockierte der Eventbuilder seine Eingänge solange, bis der TLT die zwischengespeicherten Ereignisdaten angenommen hatte. Danach lief er — unter Vollast — weiter.

Abbildung 4.14 zeigt den Verlauf der GSLT-Frequenz, gemittelt über 2s. Darüber ist die TPM-Auslastung P_{TLT} , also die Auslastung des Ringpuffers im „Triple Ported Memory“ zur Datenübergabe an der zweiten TLT-Schnittstelle, aufgetragen. Man erkennt, daß zu dem Zeitpunkt, zu dem die GSLT-Frequenz auf annähernd 75 Hz steigt, das TPM sich rapide füllt, da der TLT nicht mehr schnell genug Daten ausgelesen hat. Sofort nachdem das TPM vollständig gefüllt war, steigt der Füllzustand des internen Pufferspeichers I_{TLT} an dieser Schnittstelle an. Da dieser 3 MByte groß ist, das TPM aber nur 512 KByte, dauert es hier länger, bis der Puffer vollgelaufen ist. Nach 800s stellt dann der TLT das Auslesen von Daten an der anderen Schnittstelle ein, woraufhin schlagartig alle Puffer volllaufen und schließlich die Auslese und damit der GSLT gestoppt wird. Man beachte, daß die Puffer in den Komponentenschnittstellen erst dann beginnen vollzulaufen, wenn die Puffer auf den TLT-Schnittstellen vollständig gefüllt sind. Ebenso beginnt das TPM zu den Komponentensubsystemen erst dann vollzulaufen, wenn der interne Puffer vollständig gefüllt ist. Aus diesem Verhalten ist eindeutig zu erkennen, daß der Eventbuilderbetrieb unterbrochen wurde, weil dem Eventbuilder die Daten nicht mehr schnell genug abgenommen wurden.

4.4 Leistungsanalyse und Ermittlung von Grenzwerten

Im bisherigen Testbetrieb des ZEUS-Detektors ist keine Situation aufgetreten, die eine angemessene Bewertung des Leistungsvermögens des Eventbuilders in seiner endgültigen Ausbaustufe zuläßt. Deshalb wurden einige Belastungen simuliert, um dadurch an die Belastungsgrenzen zu stoßen.

Hierzu wurde ein Programm geschrieben, welches die Ringpuffer im TPM der Komponentenschnittstellen mit Ereignisteildaten wählbarer Länge zu füllen vermag. Dieses Programm wird zunächst vor einem Simulationsbetrieb auf den Komponentenschnittstellentransputern zu Ausführung gebracht. Während eines Simulationsbetriebes existiert nun auf den Komponentenschnittstellentransputern ein zusätzlicher Prozess, welcher bei Eintreffen einer GSLT-Entscheidungsmeldung (Vergleiche Abbildung 2.2 im vorausgegangenen Kapitel.) einen Zeiger auf diesen Ringpuffer erhöht und so die Übergabe von neuen Daten an den Eventbuilder simuliert. Da keine Daten kopiert, sondern lediglich ein Zeiger verschoben wird, ist der Einfluß auf die zur Verfügung stehende Rechenzeit innerhalb des Transputers vernachlässigbar gering und somit die Simulation dem echten Betrieb bestmöglich nachempfunden.

Der Datentransfer durch den Eventbuilder findet in drei Stufen statt: Zunächst werden die Ereignisteildaten aus dem TPM der Komponentenschnittstellen in deren lokalen Puffer kopiert. Von dort werden sie in den lokalen Puffer eines TLT-Schnittstellentransputers transferiert. Nach erfolgter „ADAMOisierung“ schließlich werden sie aus dem Pufferspeicher in das TPM kopiert. Die Kopiergeschwindigkeit in und aus dem TPM ist bekannt, sie liegt bei etwa 10 MByte/s [NIKHEF90]. Nachfolgend wird eine Messung beschrieben, die diesen Wert indirekt

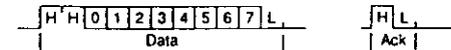


Abbildung 4.15: Das Format der Daten- und Quittierungspakete auf einem Transputerlink, dabei steht „H“ für High-Pegel und „L“ für Low-Pegel.

Komponente	Verzögerung [ns]
Wandlung TTL → RS422	14 – 20
Kabel zwischen Komponentensubsystem und Kreuzschienenverteiler	30 – 100
Wandlung RS422 → TTL	17 – 25
Kreuzschienenverteiler	75
Wandlung TTL → RS422	14 – 20
Wandlung RS422 → optisch	22 – 28
Glasfaserkabel	160
Wandlung optisch → RS422	36 – 45
Wandlung RS422 → TTL	17 – 25
Summe	385 – 498

Tabelle 4.3: Signalverzögerungen auf den Übertragungstrecken des Eventbuilders.

bestätigt. Unbekannt war dagegen die Transferringeschwindigkeit auf den Transputerlinks zwischen Komponentenschnittstellen und TLT-Schnittstellentransputern und mußte gemessen werden.

Wie bereits an anderer Stelle erwähnt, erfolgt die Datenübertragung auf den Transputerlinks seriell und byteorientiert mit einer Geschwindigkeit von 20 MBits/s: Jedes Byte wird vom Sender in Form von 11 Bits seriell übermittelt. Der Empfänger erkennt nun nach Empfang der ersten drei Bits, daß eine Übertragung stattfindet und sendet noch während er die restlichen Bits empfängt, ein Quittierungspaket aus zwei Bits an den Sender zurück (Siehe Abbildung 4.15). Empfängt dieser die Quittierung, bevor er das Senden der 11 Bits abgeschlossen hat, so kann er unmittelbar das nächste zu transferierende Byte anschließen. Damit erreicht man eine theoretische Übertragungsrate von $20/11 = 1.7$ MByte/s. Durch die Signallaufzeit auf dem Übertragungsweg werden die Daten und die Quittierungsmeldung jedoch verzögert, so daß die tatsächlich erreichte Übertragungsrate — abhängig von Länge und Aufbau des Übertragungsweges — weit unter dem theoretischen Höchstwert liegen kann. Den Datenübertragungsweg von einer Komponentenschnittstelle zu einer TLT-Schnittstelle ist in Abbildung 4.16 skizziert. Die mit ihm verknüpfte Signalverzögerung zeigt Tabelle 4.3. Da die Kabellängen zwischen Kreuzschienenverteiler und den verschiedenen Komponentenschnittstellen und damit auch die Signallaufzeiten unterschiedlich sind, besitzen die Komponentenschnittstellen individuell verschiedene Transferringeschwindigkeiten.

Für deren Bestimmung wurden in einem Simulationslauf 64 KByte große Ereignisteildaten von den verschiedenen Komponentenschnittstellen zu den TLT-Schnittstellen transferiert und aus der Differenz zwischen der Link-Zuteilungs- und der Link-Freigabe-Meldung die Transferringeschwindigkeit bestimmt. Das Ergebnis zeigt Abbildung 4.17, wo die Transferringeschwindigkeit

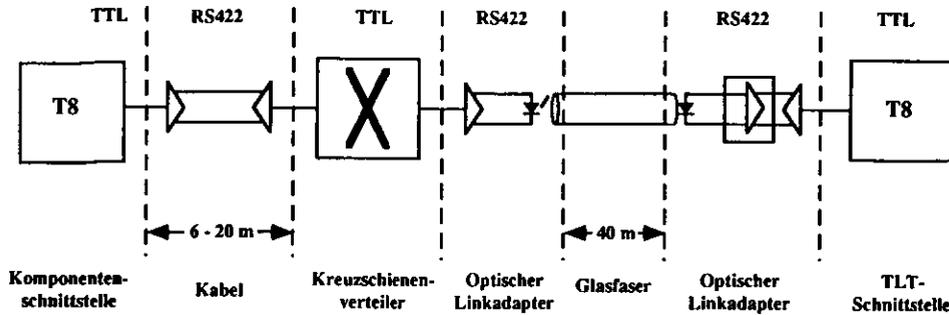


Abbildung 4.16: Schematische Darstellung der Übertragungsstrecken für Ereignisteildaten im Eventbuilder. Über den Übertragungsstrecken sind die jeweils verwendeten Protokolle (TTL, RS422) angegeben.

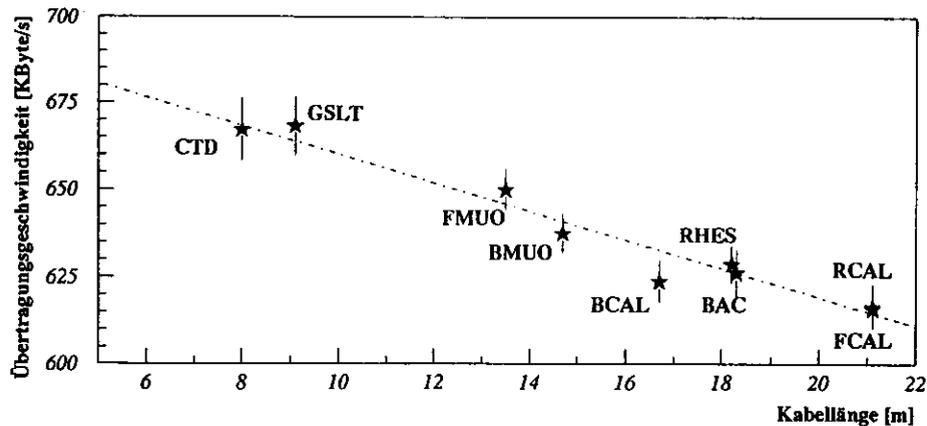


Abbildung 4.17: Die Transferegeschwindigkeit T auf den Datenlinks von den Komponentenschnittstellen zu den TLT-Schnittstellen in Abhängigkeit der Kabellänge zwischen Kreuzschienenverteiler und Komponentenschnittstelle. Diese ist keineswegs mit der gesamten Kabellänge zu verwechseln, Erläuterung siehe Text.

T in Abhängigkeit von der Kabellänge zwischen Kreuzschienenverteiler und Komponentenschnittstelle dargestellt ist. Die höchste Transferegeschwindigkeit wird vom GSLT aus mit 668 KByte/s erzielt, die niedrigste von FCAL aus mit $T = 616$ KByte/s. Dies ist insofern ungünstig, als der Testbetrieb gezeigt hat, daß FCAL die längsten Ereignisteildaten liefert und somit die höchste Transferegeschwindigkeit benötigt.

Diese so bestimmten Größen stellen den optimalen Fall dar, in dem die Links ständig für den

Datentransfer genutzt werden. Im Betrieb werden jedoch ständig wechselnde Verbindungen zwischen einer Komponentenschnittstelle und den verschiedenen TLT-Schnittstellen hergestellt. Zwischen dem Beenden eines Transfers und dem Beginn eines neuen Transfers kommt es zu Übertragungspausen auf den Links, welche den Datendurchsatz senken.

Für die maximale Frequenz f_K , mit der eine Komponentenschnittstelle Ereignisteildaten liefern kann, gilt mit N_{Links} als der Anzahl der Links, die von einer Komponentenschnittstelle zum Kreuzschienenverteiler führen, und der Annahme, daß alle diese Links gleiche Transferleistung erbringen

$$f_K = \frac{N_{Links}}{\frac{L_K}{T_K} + t_K^{frei}}$$

Entsprechendes gilt für die erreichbare Frequenz an den TLT-Schnittstellen, wobei $N_{TLT-Links}$ die Anzahl der Links aller TLT-Schnittstellentransputer bezeichne:

$$f_{TLT} = \frac{N_{TLT-Links}}{\sum_{\text{alle Komponenten}} \left(\frac{L_K}{T_K} + t_K^{frei} \right)}$$

Die Frequenz, mit der der Eventbuilder Ereignisse verarbeiten kann, kann nun nicht größer sein als die Frequenz, mit der die langsamste Komponentenschnittstelle Ereignisteildaten liefern kann. Genauso kann sie nicht größer sein als die Frequenz, mit der die TLT-Schnittstellen Daten empfangen können.

Betrachtet man den ungünstigsten Fall mit $T = 600$ KByte/s (siehe Abbildung 4.17) und $t_K^{frei} = 10$ ms sowie den Datenlängen aus Tabelle 2.1, so zeigt sich, daß erst bei weniger als 6 TLT-Schnittstellentransputern die Frequenz f_{TLT} kleiner als f_K und damit zum limitierenden Faktor wird. Somit kann zur Abschätzung der oberen Leistungsgrenze des Eventbuilders mit guter Näherung die Frequenz verwendet werden, mit der die Komponentenschnittstelle mit den größten Ereignisteildaten diese noch über ihre drei Links zu transferieren vermag. Dies ist für verschiedene T_K in Abbildung 4.18 aufgetragen.

Da die Transferegeschwindigkeit T_K auf den Links hardwaremäßig vorgegeben ist, sich also nicht je nach Auslastung und Betriebszustand des Eventbuilders ändert, ist bei vorgegebener Datenlänge L_K der einzig unbekannt Parameter die Zeit t_K^{frei} , die ein Link unbenutzt ist. Ihre Größe kann nur sinnvoll bestimmt werden, wenn sichergestellt ist, daß der interne Puffer der Komponentenschnittstelle stets gefüllt ist und somit stets Daten zur Transferierung zur Verfügung stehen.

Diese Zeit setzt sich aus einer Reihe von unterschiedlichen Bestandteilen zusammen:

- Zunächst geht die Rechenzeit ein, die der Komponentenschnittstellentransputer benötigt, um eine Link-Zuteilungsmeldung zu empfangen und den Transfer zu starten, sowie, nach Ende des Transfers, die Zeit zum Absenden der Link-Freigabemeldung (Vergleiche Abschnitt 2.3.2). Deren Größe kann von der Länge und der Struktur der Ereignisteildaten beeinflusst werden.
- Durch Belastung der Steuereinheit mit zu vielen Meldungen kann es zu Verzögerungen im Weiterleiten von Meldungen kommen.
- Der Algorithmus der Steuereinheit, nach dem Links vergeben werden, könnte durch ungünstige Linkvergabe für gegenseitige Aussperrungen sorgen, so daß eine Komponentenschnittstelle immer erst auf des Freiwerden eines Links auf der TLT-Seite warten muß.

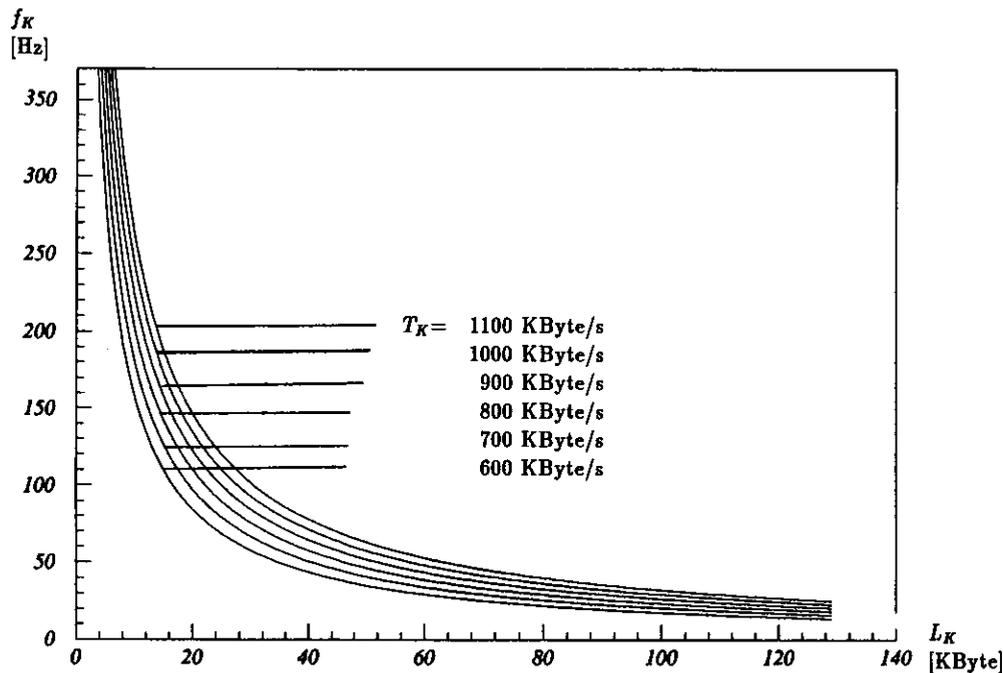


Abbildung 4.18: Die maximale Frequenz f_K , mit der eine Komponentenschnittstelle Ereignisteildaten liefern kann in Abhängigkeit der Datenlänge L_K . Variiert wurde T_K ; $t_K^{frei} = 2.3ms$ nach erfolgter Modifikation des Komponentenschnittstellenprogramms.

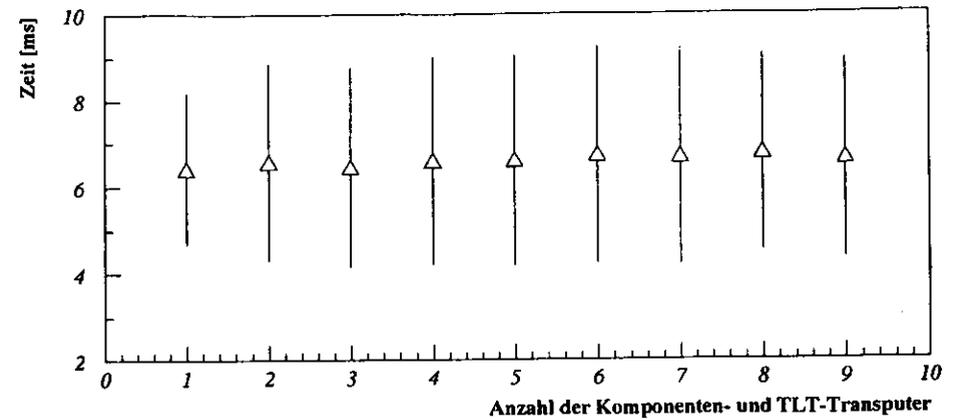


Abbildung 4.19: Mittlere Zeit, die die Links der Komponentenschnittstelle mit den längsten Ereignisteildaten (CTD mit 30 KByte) ungenutzt blieben. Variiert wurde die Anzahl der TLT- und Komponentenschnittstellen in einem Simulationslauf.

Der Algorithmus, nach dem die Steuereinheit Linkzuteilungen vornimmt, wurde untersucht, indem der Eventbuilderbetrieb mit unterschiedlich vielen angeschlossenen Komponenten simuliert wurde. Dabei wurde darauf geachtet, daß das Verhältnis der Anzahl der für den Datentransport zur Verfügung stehenden Links zwischen Komponentenschnittstellen und Kreuzschienenverteiler bzw. Kreuzschienenverteiler und TLT-Schnittstellen der endgültigen Ausbaustufe des Eventbuilders angeglichen war³. Zunächst wurden lediglich Daten von der Komponente, die die umfangreichsten Ereignisteildaten an den Eventbuilder übermitteln wird (CTD, 30 KByte), zu einer TLT-Schnittstelle geschickt. Danach wurden, geordnet nach der Länge der betreffenden Ereignisteildaten, schrittweise weitere Komponenten in die Simulation einbezogen. Für jeden dieser Simulationsläufe wurde t_{CTD}^{frei} gemessen. Dabei war die benutzte Triggerrate so hoch, daß auf den Komponentenschnittstellen stets Ereignisteildaten im Puffer vorlagen und somit Schwankungen in t_{CTD}^{frei} einzig auf die Steuereinheit zurückgeführt werden konnten. Abbildung 4.19 zeigt, daß die Zeit t_{CTD}^{frei} , in der die Links unbenutzt bleiben, nicht von der Anzahl der Komponenten- und TLT-Schnittstellen abhängt. Somit arbeitet der Algorithmus der Steuereinheit optimal.

Die Belastung der Steuereinheit mit Meldungen läßt sich untersuchen, indem während einer Simulation mit kleinen Ereignisteildaten die GSLT-Frequenz variiert wird. Bei durchgeführten Untersuchungen konnten GSLT-Frequenzen von bis zu

$$f_{Trig}^{max} = 254Hz$$

vom Eventbuilder bewältigt werden. Dieser Wert reproduziert bereits während eines früheren Testbetriebes gemachte Beobachtungen und stellt daher die obere Grenze für die Leistung der Steuereinheit dar.

³In seiner endgültigen Ausbaustufe wird der Eventbuilder über 51 (48) Links zwischen Kreuzschienenverteiler und Komponentenschnittstellen (TLT-Schnittstellen) verfügen. Momentan verfügen Komponenten- und TLT-Schnittstellen über jeweils drei Links zum Kreuzschienenverteiler. Daher wurde bei der Simulation stets mit der gleichen Anzahl von Komponenten- und TLT-Schnittstellen gearbeitet.

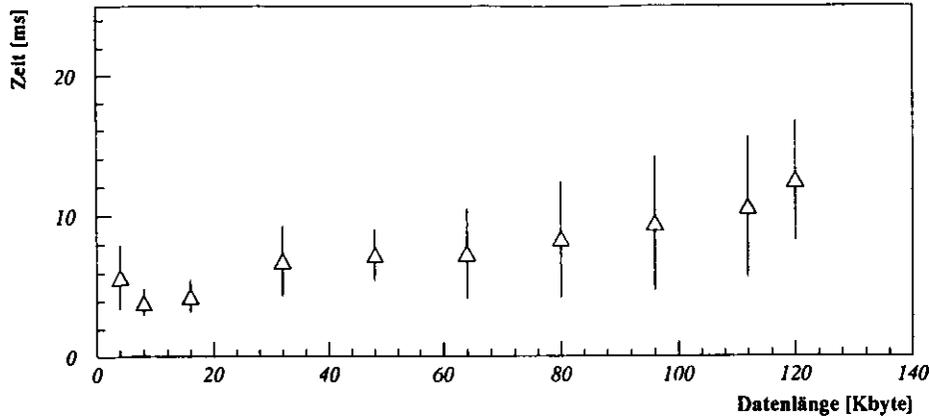


Abbildung 4.20: Mittlere Zeit, die die Links der Komponentenschnittstelle mit den längsten Ereignisteildaten ungenutzt blieben. Variiert wurde die Länge der Daten.

Schließlich wurde der Einfluß des Umfanges des Ereignisteildaten der verschiedenen Komponenten auf die Transferleistung innerhalb des Eventbuilders untersucht. Hierfür wurden von einer Komponente Ereignisteildaten verschiedener Länge an den Eventbuilder übermittelt, während alle anderen Komponenten Daten fester Länge (1 KByte) an den Eventbuilder übertragen. Das Resultat zeigt Abbildung 4.20. Die Zeit, in der Links nicht zum Datentransfer genutzt werden können, nimmt offensichtlich mit wachsender Länge der Ereignisteildaten zu. Um dieses Phänomen zu verstehen, wurde t_K^{frei} für verschiedene $L_{K,n}$ histogrammiert (Abbildung 4.21). Die Histogramme zeigen mehrere ausgeprägte Maxima, deren Abstand mit zunehmender Länge der Ereignisteildaten wächst. Trägt man den Abstand zweier solcher Maxima in Abhängigkeit von $L_{K,n}$ auf (Abbildung 4.22), so zeigt sich, daß dieser linear mit der Datenlänge zunimmt. Die beobachtete Steigung entspricht genau 10 MByte/s und damit der Rate, mit der Daten aus dem TPM von der Komponentenschnittstelle eingelesen werden können. Offensichtlich kann eine Komponentenschnittstelle, während sie Daten aus dem TPM in den internen Puffer kopiert, keine Meldungen empfangen oder senden. Die Zeit zwischen zwei Transfers auf einem Link wird daher jeweils um ein Vielfaches der Kopierzeit erhöht.

Eine genaue Untersuchung der an einer Komponentenschnittstelle beteiligten Prozesse hat inzwischen ergeben, daß eine Unterbrechung des Datentransfers aus dem TPM in den internen Puffer tatsächlich nicht möglich ist, jedoch durch Umformulierung des Programmcodes und geeignete Priorisierung der Prozesse erzwungen werden kann. Abbildung 4.23 zeigt daher die ersten Maxima, d. h. die minimale Zeit, die ein Link zwischen zwei Transfers ungenutzt blieb, in Abhängigkeit von der Datenlänge. Man erkennt, daß für große Datenlängen die Werte konstant bleiben, sich bei kleineren Datenlängen jedoch erhöhen. Dies ist auf die bereits genannte obere Grenze von 254 Hz für f_{TPM} zurückzuführen, welche in Abbildung 4.24 noch einmal deutlich zu erkennen ist. Aufgetragen ist dort die Frequenz, mit der die Komponentenschnittstelle mit den längsten Ereignisteildaten Daten liefern konnte gegen die Datenlänge.

Für die Zeit t_K^{frei} , die ein Link zwischen zwei Transfers ungenutzt ist, kann man also — nach erfolgter Änderung des Komponentenschnittstellenprogramms — unabhängig von der

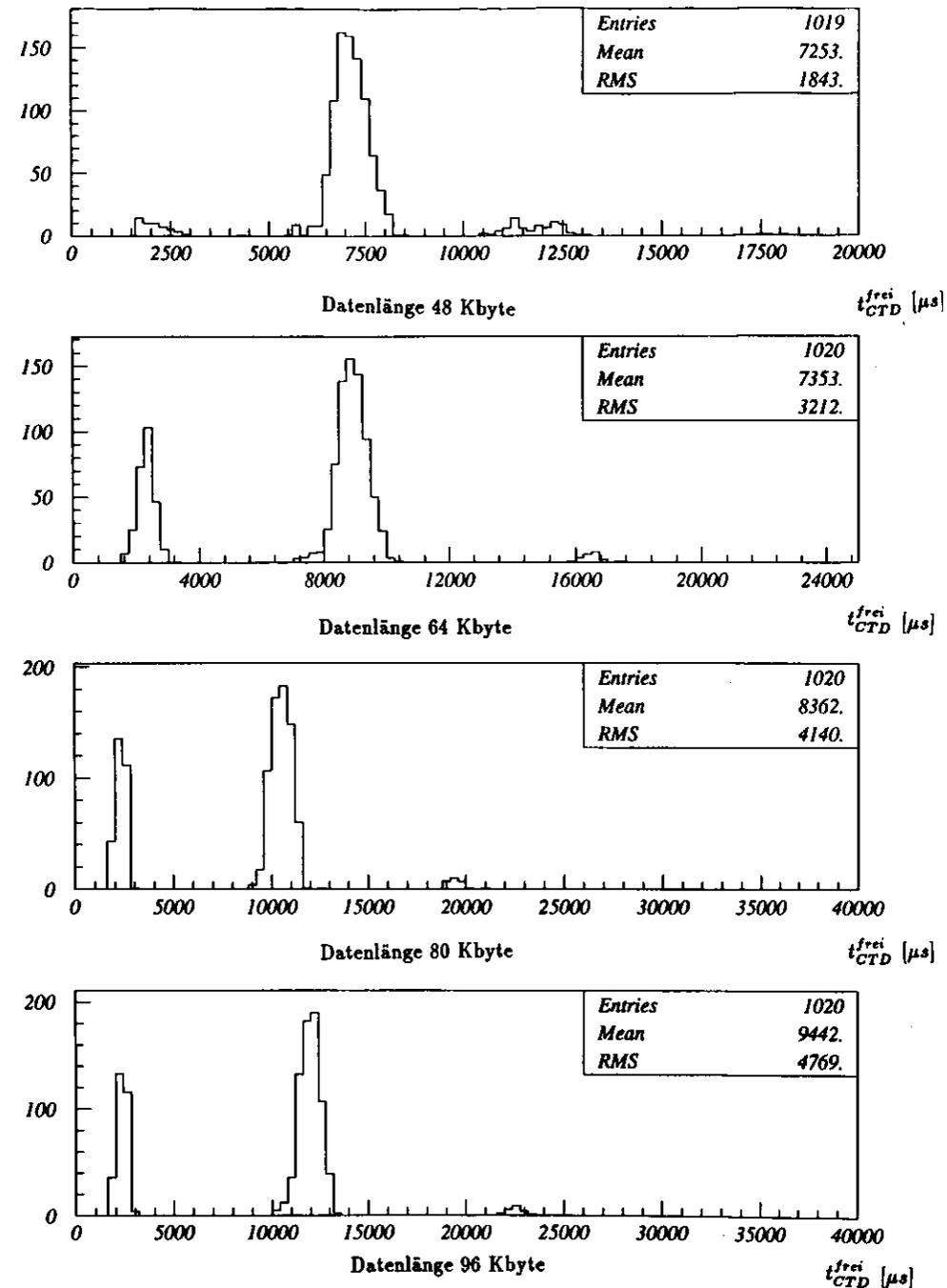


Abbildung 4.21: Verteilungen der mittleren Zeit, die die Links der Komponentenschnittstelle mit den längsten Ereignisteildaten im Mittel ungenutzt blieben für verschiedene Datenlängen. Die Daten beruhen auf einem Simulationslauf mit 6 Komponentenschnittstellen.

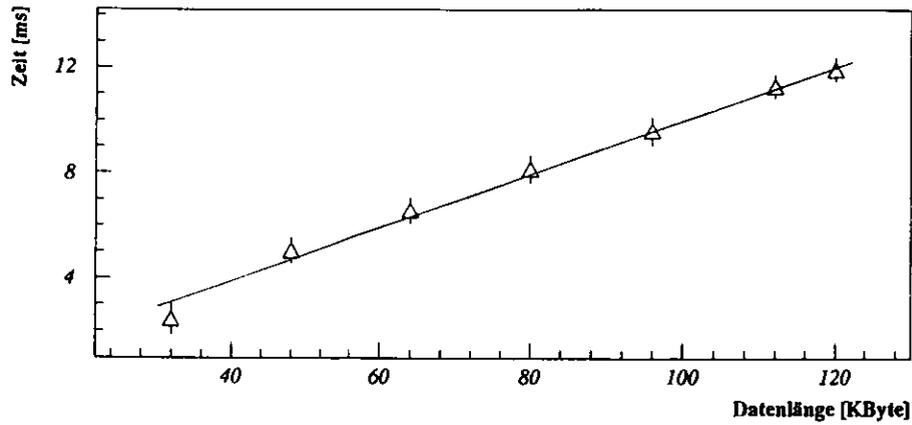


Abbildung 4.22: Abstand der ersten beiden Maxima in der Verteilung der Zeit, die die Links der Komponentenschnittstelle mit den längsten Ereignisteildaten ungenutzt blieben in Abhängigkeit von der Datenlänge.

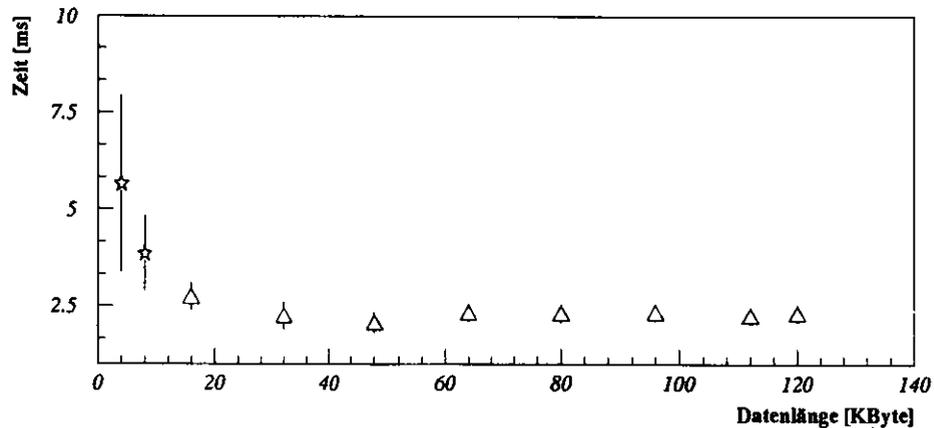


Abbildung 4.23: Verlauf der minimalen Zeit, die die Links der Komponentenschnittstelle mit den längsten Ereignisteildaten unbenutzt blieben. Bei den mit Sternen gekennzeichneten Meßwerten war der interne Puffer nicht mehr ausreichend gefüllt. Die Daten beruhen auf einem Simulationslauf mit 6 Komponentenschnittstellen.

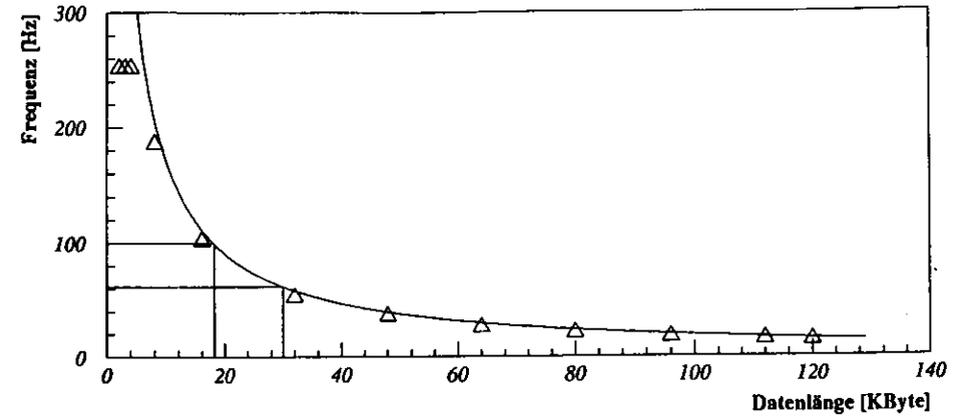


Abbildung 4.24: Rate, mit der die Komponentenschnittstelle mit den längsten Ereignisteildaten ihre Daten liefern konnte in Abhängigkeit der Datenlänge. Der eingezeichnete Kurvenverlauf zeigt den theoretischen Verlauf bei einer Transfergeschwindigkeit $T_K = 660$ KByte/s und $t_K^{frei} = 2.3$ ms. Die Daten beruhen auf einem Simulationslauf mit 6 Komponentenschnittstellen.

Rechenbelastung auf den Komponentenschnittstellentransputern und der Steuereinheit einen konstanten Wert von $2,3$ ms annehmen. Somit hängt die maximal erreichbare Ereignisrate des Eventbuilders für Triggerfrequenzen f_{T_i} unter 254 Hz nur von der Transfergeschwindigkeit T_K auf den Datenlinks und der Länge L_K der größten Ereignisteildaten ab. Unter der Voraussetzung, daß die größten Ereignisteildaten von der zentralen Spurkammer (CTD) geliefert werden und eine Länge von 30 KByte besitzen, beträgt die erreichbare maximale Ereignisrate 64 Hz.

Kapitel 5

Zusammenfassung

Im Zuge dieser Arbeit wurde eine Reihe von Kenngrößen definiert und gemessen, mit deren Hilfe sich sowohl Abläufe innerhalb des ZEUS-Eventbuilders beschreiben und analysieren lassen, als auch die an ihn angrenzenden Systeme des ZEUS Trigger- und Datenerfassungssystems auf ihr Systemverhalten hin untersucht werden können. Dazu wurde eine umfangreiche Software entwickelt, die die Erfassung und Auswertung der Kenngrößen möglich macht.

Während des Probetriebes des ZEUS-Detektors mit kosmischer Strahlung im Dezember 1991 wurde das Verhalten des Trigger- und Datenerfassungssystems untersucht und mit den an es gestellten Anforderungen verglichen. Dabei zeigte sich, daß das Vorwärts-, Rückwärts- und Barrel-Kalorimeter (FCAL, RCAL, BCAL) die bei der Planung als maximal angegebene Datenlänge um jeweils 10 Kbyte überschritten haben. Weiter zeigte sich, daß die schnellstmögliche Rate, mit der FCAL und BCAL Daten liefern konnten, unter der geforderten Rate von 100 Hz lag und 71 Hz betrug.

Da aus dem Probetrieb nicht auf die Leistungsgrenzen des Eventbuilders geschlossen werden konnte, wurde ein Simulationsverfahren entwickelt, mit dem der Eventbuilder beliebigen Belastungen ausgesetzt werden kann. Durchgeführte Simulationen zeigten, daß die Höchstfrequenz, mit der der Eventbuilder Ereignisteildaten einlesen, formatieren und an die dritte Triggerstufe (TLT) übergeben kann, nur von der Transfergeschwindigkeit auf den Datenlinks und der Länge der größten Ereignisteildaten bestimmt wird.

Da die bei der Planung des Eventbuilders geforderte maximale Länge der Kabel zwischen Komponenten- und TLT-Schnittstellen von 30 m [Gather88] nicht eingehalten, sondern weit überschritten wurde, liegt die Transfergeschwindigkeit mit 616–667 KByte/s deutlich unter dem geplanten Wert von 1,1 MByte/s. Mit der momentan erreichten Transfergeschwindigkeit und einer maximalen Länge der Ereignisteildaten von 30 KByte beträgt die höchste erreichbare Ereignistransferrate daher nur 64 Hz. Eine Steigerung der Transfergeschwindigkeit ist jedoch mit entsprechendem Hardwareaufwand möglich und schon in [Hagge90] beschrieben worden. Schließlich konnte durch die Auswertung interner Kenngrößen des Eventbuilders im Rahmen dieser Diplomarbeit die Ursache einer mit steigender Datenlänge zunehmenden Unbenutztheit von Datenlinks geklärt und eine Verbesserungsmöglichkeit aufgezeigt werden. Die sich daraus ergebende Programmveränderung verringerte die Zeit, die ein Datenlink zwischen zwei Transfers ungenutzt blieb, von typischen 6,4 ms auf einen Höchstwert von nur noch 2,3 ms.

Anhang A

Der Server

Die wichtigsten Bestandteile des Eventbuilders wurden auf Basis eines Transputernetzes realisiert. Da diese Transputer über kein Betriebssystem verfügen, welches einen eigenständigen Betrieb ermöglicht, wurde eine UNIX-Workstation vom Type SGI 4D/25 als Entwicklungs- und Bootplattform ausgewählt.

Um das Transputernetz zu booten und eine Kommunikation über LAN zur zentralen Experimentsteuerung und zu Systemexperten zu ermöglichen, wurde ein Programm („Server“) geschrieben. Dieses wird auf der Workstation abgearbeitet und wickelt den gesamten Nachrichtenaustausch zwischen Transputernetz und Workstation sowie Workstation und Systemspezialisten/Experimentsteuerung ab. Die Kommunikation zwischen einem Systemspezialisten und dem Server läuft über das Programm „Expert“, die Kommunikation zwischen der zentralen Experimentsteuerung und dem Server läuft über das Programm „Mtt“ (Mailbox To TCP/IP). Die Prozeßkonfiguration ist in Abbildung A.1 noch einmal dargestellt.

A.1 Funktionsprinzip des expert-Programms

Die Kommunikation zwischen einem Systemspezialisten und dem Server wird durch Aufruf des Programms „Expert“ gestartet und findet grundsätzlich über TCP/IP statt. Daher ist es nicht nötig, das „Expert“ auf der gleichen Workstation aufgerufen wird, auf der der „Server“ läuft; im Prinzip kann dazu jeder Rechner verwendet werden, der eine TCP/IP-Verbindung zur Eventbuilderworkstation herstellen kann und über das „Expert“-Programm verfügt. Mit Hilfe des „Expert“-Programms ist es möglich, Befehle an den „Server“ zu senden. Gleichzeitig bekommt der Aufrufer alle momentan anfallenden Textmeldungen angezeigt. Da sich bis zu fünf Systemspezialisten gleichzeitig mit dem „Server“ durch das „Expert“-Programm verbinden können, wurde auch eine Kommunikation auf Textbasis zwischen diesen vorgesehen. Dies erwies sich während der Systementwicklung als sehr hilfreich, da nicht immer alle Systementwickler am gleichen Ort verfügbar waren und so eine sehr komfortable Form der Ferndiagnose und Fernberatung möglich wurde.

Nach dem Starten versucht das expert-Programm zunächst, per UNIX-Standardfunktion `rcmd()` eine Verbindung zum TCP/IP-Port¹ `'evbserv'` der Workstation herzustellen.

Wenn die `rcmd()`-Funktion nach spätestens 30s feststellt, daß keine Verbindung zustandekommt, geht das „Expert“-Programm davon aus, daß momentan kein „Server“-Prozeß existiert und versucht, diesen zu starten. Dazu versucht es, —wieder per `rcmd()`-Aufruf,— eine

¹TCP/IP-Ports werden in UNIX durch das file `/etc/services` mit symbolischen Namen versehen. Die momentane Nummer des `evbserv`-Ports ist 4321

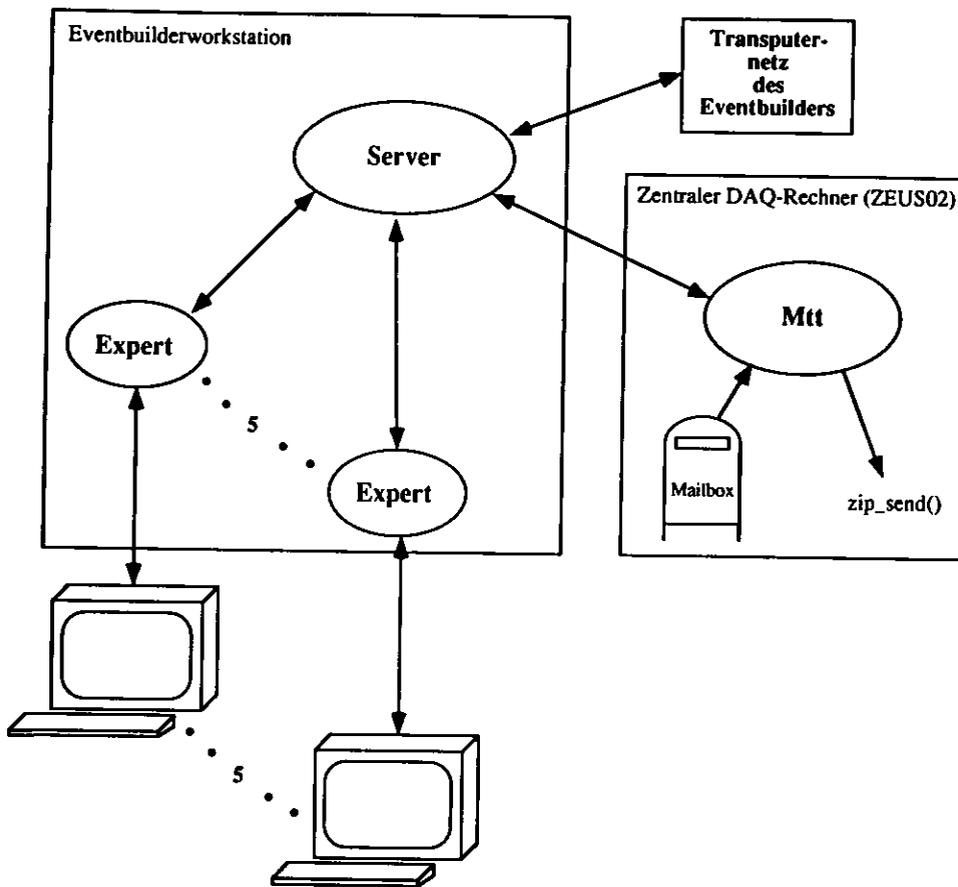


Abbildung A.1: Die Prozesskonfiguration des Eventbuilderservers

Verbindung zum Internet-Daemon (inetd) der Workstation herzustellen, welcher sodann einen Remote-Shell-Daemon (rshd) startet. Letzterer authentifiziert die Benutzerdaten, die ihm von der rcmd()-Funktion des „Expert“-Programms übermittelt wurden. Sofern dem Aufrufer des „Expert“-Programms im file `~ovbserv/.rhosts` die entsprechenden Rechte eingeräumt wurden, startet der rshd dann den „Server“, welcher die TCP/IP-Verbindung des rshd zum „Expert“-Programm erbt.

Sofern schon der erste Versuch des Verbindungsaufbaus des „Expert“-Programms erfolgreich war, der „Server“-Prozess also schon existierte, nimmt dieser die Authentifizierung nach dem gleichen Schema wie der Remote-Shell-Daemon vor. Hierzu benutzt er die Standardfunktion `'ruserok()'`. Anschließend ist die Verbindung hergestellt.

Das Remote-Shell-Protokoll erfordert es, daß der Ausgangsport der TCP/IP-Verbindung im privilegierten Bereich von 0 - 1024 liegt. Um solche einen Port binden zu dürfen, benötigt das „Expert“-Programm gewisse Sonderrechte. Auf einem UNIX-Rechner muß es S-Bit zu root sein, auf einem VMS-Rechner benötigt es das SYSPRV oder BYPASS Privileg.

A.2 Funktionsprinzip des Mtt-Programms

Das Funktionsprinzip des „Mtt“-Programms und die Abläufe beim Starten gleichen denen im vorherigen Abschnitt beschriebenen Abläufen des „Expert“-Programms. Der einzige Unterschied besteht darin, daß es Textmeldungen nicht einfach auf dem Schirm ausgibt, sondern mit Hilfe des ZIP-Programmpaketes [Park89] in VMS-Mailboxen verschickt und so eine Umsetzung der TCP/IP-Meldungen vornimmt. Ebenso liest es keine Befehle von der Tastatur ein, sondern besitzt eine eigene Mailbox, deren Inhalt sofort über TCP/IP an den „Server“-Prozess auf der Workstation gesendet wird. Das „Mtt“-Programm selbst kommt auf dem zentralen Experimentrechner, einer VAX 8700 zur Ausführung.

Anhang B

Danksagung

An dieser Stelle möchte ich mich bei allen denen bedanken, die mir die Erstellung dieser Diplomarbeit ermöglicht haben. Mein besonderer Dank gilt Herrn Prof. Dr. E. Lohrmann, der mir diese interessante und abwechslungsreiche Diplomarbeit im Rahmen des ZEUS-Experiments ermöglichte.

Weiter gilt mein Dank meinem Betreuer, Herrn Dr. W. Vogel, der mich über die gesamte Dauer der Arbeit auf das hervorragendste unterstützt und beraten hat.

Sehr zu danken habe ich auch der ZEUS-Eventbuildergruppe, namentlich Lars Hagge, Ulf Behrens, Johannes Mainusch, Torsten Woeniger und Frank-Olaf Lohmann. Die freundschaftliche Atmosphäre innerhalb der Gruppe wird mir in angenehmer Erinnerung bleiben.

Schließlich bedanke ich mich bei den zahlreichen Kollegen des Deutschen Elektronen-Synchrotrons und der ZEUS-Kollaboration für viele große und kleine Hilfestellungen und nicht zuletzt bei den Damen des ZEUS-Sekretariats, die mir stets großzügig ihren Rechner zur Erstellung der vielen Bilder überlassen haben.

Anhang C

Erklärung

Hiermit versichere ich, die vorliegende Arbeit unter Angabe aller verwendeten Quellen und Hilfsmittel selbständig angefertigt zu haben.

Hamburg, im April 1992

(Thorsten Schlichting)

Literaturverzeichnis

- [ADAMO91] S. M. Fisher, P. Palazzi
The ADAMO Data System
Programmers Manual — Version 3.2
CERN/ECP 1991
- [Behrens90] Ulf Behrens et al.
Data Flow from the components to the Event Builder
ZEUS-note 90-127, DESY Hamburg, December 1990
- [Behrens91] Ulf Behrens et al.
Interfacing Component Subsystems with the Eventbuilder
ZEUS-note 91-72, DESY Hamburg, July 1991
- [Gather88] Karl Gather
Conclusions at the Oxford meeting
ZEUS-note 88-124, DESY Hamburg, November 1988
- [Gruppen85] C. Grupen
Kosmische Strahlung
Physik in unserer Zeit 3(1985)69
- [Hagge90] Lars Hagge
Anwendungen von Transputern in der Hochenergiephysik
Diplomarbeit am Fachbereich Informatik, Universität Hamburg, Oktober 1990
- [HAY62] P. J. Hayman, A. W. Wolfendale
The Momentum Spectrum of Cosmic Ray Muons near Sea Level in the Momentum Range 5-1000 GeV/c
Proc. Phys. Soc. 80(1962), 710
- [Hoare78] C. A. R. Hoare
Communicating Sequential Processes
Comm. ACM 21(1978)666
- [INMOS88] INMOS Limited
TRANSPUTER INSTRUCTION SET
A compiler writer's guide
Prentice Hall International (UK) Ltd. 1988
ISBN 0-13-929100-8
INMOS document number 72 TRN 119 05
- [Leich92] Angelika Leich
ZEUS Ereignis-Grafikprogramm
Ich danke Angelika Leich für das Bild
- [NIKHEF90] NIKHEF ZEUS-group
2TP.VME User Software Manual (Version 2.3)
NIKHEF-H, Amsterdam, October 1990
- [Park89] Il Park
The InterProcess Communication in ZEUS central data acquisition system (ZIP)
ZEUS-note 89-124, DESY Hamburg, December 1989
- [Park90] Il Park
Control Flow for Tasks on ZEUS CDAQ
ZEUS-note 90-65, DESY Hamburg, June 1990
- [PDG90] Particle Data Group
Particle Properties Data Booklet
from 'Review of Particle Properties'
Physics Letters B239, April 1990
- [Peterson89] Wade D. Peterson
The VMEbus Handbook
The VMEbus International Trade Association, Scottsdale, USA, 1989
- [Smith90] W. H. Smith et al.
The ZEUS trigger system
ZEUS-note 89-084, version 2.0, DESY Hamburg, August 6, 1990
- [Woeniger89] Torsten Woeniger
Concepts for a Transputer Based Eventbuilder
ZEUS-note 89-112, DESY Hamburg, November 1989
- [Youngman91] C. Youngman
Run Sequencing
ZEUS-note 91-12, DESY Hamburg, February 1991
- [ZEBRA86] ZEBRA Reference Manual
Program Library, DD Division, CERN, Geneva 1986