

Interner Bericht
DESY F31-87-02
February 1987

DAS CRYSTAL-BALL MIKROPROZESSOR-SYSTEM

von

Hermann Meyer

Eigentum der	DESY	Bibliothek
Property of		Library
Zugang:	26. MRZ 1987	
Accession:		
Leihfrist:	7	
Loum pe:		

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“

Das Crystal-Ball Mikroprozessor-System

Diplomarbeit
von
Hermann Meyer

II. Institut für Experimentalphysik
Universität Hamburg
November 1986

Zusammenfassung

In dieser Arbeit wurde ein vernetztes Online-Multiprozessor-System entwickelt, das Aufgaben aus der Online-Datenverarbeitung des Crystal-Ball Experiments übernahm. Mit dem Crystal-Ball Experiment werden am Speicherring DORIS II des Deutschen Elektronen Synchrotron (DESY) die Υ -Resonanzen untersucht.

Die Aufgabe des Prozessorsystems ist die kontinuierliche graphische Darstellung von physikalischen Ereignissen. Dazu werden die Rohdaten der Ereignisse in physikalische Größen umgerechnet, danach analysiert und anschließend nach Ereignisgruppen selektiert.

Das Prozessor-System wurde in der Meßperiode im Frühjahr 1986 parallel zu der existierenden Online-Datenverarbeitung getestet. Der Vergleichsdatensatz umfasst 638 000 registrierte Ereignisse. Davon wurden 96,7% (29,0%) analysiert (in Klammern der Wert des Crystal-Ball Online-Systems).

Der Vergleich zwischen dem Anteil der analysierten Ereignisse des Multiprozessor-Systems (96,7%) und dem Crystal-Ball-Online-Systems (29,0%), zeigt den Vorteil des entwickelten Prozessorsystems. Dieser Vorteil beruht auf der Methode der horizontalen Datenverarbeitung, bei der Prozesse parallel durch mehrere Prozessoren verarbeitet werden können. Dagegen erfolgt die Prozessverarbeitung bei der bisherigen vertikalen Methode durch einen Prozessor, der zwischen den Programmen, die den einzelnen Prozess steuern, hin und her schalten muß. Diese Umschaltzeiten führen zu Zeitverlusten bei der Datennahme.

Inhalt

Zusammenfassung	1
1 Einleitung	4
2 Der Speicherring DORIS II	6
3 Das Crystal-Ball Experiment	11
3.1 Der Crystal-Ball Detektor	11
3.1.1 Der NaJ-Schauerzähler	11
3.1.2 Der Innendetektor	13
3.1.3 Weitere Detektorkomponenten	16
3.2 Das Teilchenverhalten im Crystal-Ball	16
4 Die physikalischen Grundlagen	19
4.1 Das Quark-Modell	19
4.2 Elektron - Positron - Reaktionen	21
5 Das Datenaufnahme-System	27
5.1 Die Hardware des Online-Systems	29
5.2 Die Triggerbedingungen	31
5.3 Die Software des Online-Systems	32
5.4 Das PADAC-System	35
6 Das Multi-Prozessor-System	36
6.1 Das FPSS	37
6.2 Das TRON-System	38
6.3 Der HOST-Rechner	40
6.4 Der PADAC-Microprozessor	41
6.4.1 Die TMS 9900 Architektur	41
6.4.2 Das Workspace-Konzept	42
6.4.3 Die Interrupt-Struktur	42
6.4.4 Der Direct Memory Access	46
7 Das Prozessor-Betriebs-System	48
7.1 Der Aufbau des MCX-Systems	50
7.2 Die Arbeits-Schleife des MCX-Systems	51
7.2.1 Der inaktive IDLE-Loop Zustand	51

7.2.2 Der aktive IDLE-Loop Zustand	53
7.3 Externe Anforderungen an das MCX-System	57
7.4 Die Unterprogramm-Technik	58
8 Das Crystal-Ball-Micro-System	60
8.1 Die CBMS-Hardware	60
8.2 Die CBMS-Software	61
8.2.1 Prozessor I	61
8.2.2 Prozessor II	65
8.3 Synchronisation und Kommunikation	71
8.3.1 Die externe Prozessor-Kommunikation	71
8.3.2 Die interne Prozessor-Kommunikation	72
8.4 Die physikalische Analyse der Daten	73
9 Schlußbetrachtung	75
A MCX-Kommandos/Direktiven	77
B Graphische Basisfunktionen in der GAPLIB	79
Literatur	81
Danksagung	83

Kapitel 1

Einleitung

Diese Arbeit entstand am Crystal-Ball Detektor am Speicherring DORIS II des Deutschen-Elektronen-Synchrotrons (DESY), einer Großforschungseinrichtung zur experimentellen Erforschung der Elementarteilchen.

Der Crystal-Ball Detektor wurde ab 1978 am SLAC in Kalifornien für die Quarkoniumsspektroskopie verwendet. Bis 1981 wurde das Quark-Antiquark-System $c\bar{c}$ am Speicherring SPEAR untersucht. In einer spektakulären Aktion wurde der Detektor 1982 nach Deutschland transportiert und nimmt seit August 1982 Daten für das Quark-Antiquark-System $b\bar{b}$.

Ziel dieser Arbeit war es, ein Multiprozessor-System zu entwickeln, das Aufgaben aus der Online-Datenverarbeitung des Crystal-Ball Experiments übernimmt. Die Aufgaben, die moderne Speicherringexperimente an ein Online-Datenverarbeitungs-System stellen, lassen sich in vier Punkten zusammenfassen:

- Online-Datennahme
- Experiment-Überwachung
- Selektion und Reduktion physikalischer Ereignisse
- Programm-Entwicklungen

In der Regel erfolgt die Bewältigung dieser Aufgaben durch einen Prozessor mit einem Multiuser/Multitasking-System. Die im Wechselspiel zeitintensiver Input/Output- und Task-Aktivitäten entstehenden Zeitverluste in der Datennahme, zeigen die Grenzen der "vertikalen" Datenverarbeitung auf. Anhand des Datenaufnahme-Systems des Crystal-Ball Experiments wird die vertikale Datenverarbeitung in Abschnitt 5 beschrieben.

Das im Rahmen dieser Arbeit entwickelte Konzept für eine "horizontale" Datenverarbeitung stellt eine Alternative zur Einprozessorverarbeitung dar. Hier stehen mehrere Prozessoren zur Verfügung, die die oben angeführten Aufgaben eines Online-Systems durch eine parallele Verarbeitung erfüllen.

Für das am Crystal-Ball Experiment eingesetzte Multiprozessor-System wurde als Aufgabenbereich die Selektion und Aufbereitung der physikalischen Ereignisse, mit anschließender graphischer Darstellung auf einem Bildschirm,

festgelegt. Dieses System wird in seinem Aufbau und seinen Funktionen in Abschnitt 8 beschrieben.

Die notwendigen Komponenten eines Multiprozessor-Systems werden in Abschnitt 6 vorgestellt. Einen wichtigen Punkt nimmt hierbei das Betriebssystem der Prozessoren ein, da die Verarbeitung hoher Eingangsraten stark I/O- unterstützende Betriebssysteme notwendig macht. Das hierfür entwickelte Real-Time-System wird in Abschnitt 7 dargestellt.

Die physikalischen Grundlagen der Elementarteilchenphysik, die in die Aufbereitung der physikalischen Ereignisse einfließen, werden in Abschnitt 4 erläutert. Da die Online-Datennahme nicht isoliert vom Experiment steht, wird das Crystal-Ball Experiment in Abschnitt 3 in seiner Gesamtheit vorgestellt. Die Beschreibung des Speicherrings DORIS II geschieht in Abschnitt 2 und soll den Rahmen moderner Hochenergie-Experimente aufzeigen.

Den Abschluß dieser Arbeit bildet die Erläuterung der erzielten Ergebnisse mit dem eingesetzten Multiprozessor System, die während der Meßzeit im Frühjahr 1986 gewonnen wurden.

Kapitel 2

Der Speicherring DORIS II

Als Ausgangspunkt für viele Untersuchungen im Bereich der Elementarteilchenphysik dient die gegenseitige Vernichtung von Elektronen und Positronen. In einem Speicherring bewegen sich die Teilchen gegenläufig auf einer gemeinsamen Strahlachse, so daß im sogenannten Wechselwirkungspunkt als Schwerpunktsenergie die doppelte Strahlenergie zur Verfügung steht. Diese Schwerpunktsenergie steht der Erzeugung von Elementarteilchen nach dem Einsteinschen Äquivalenzprinzip

$$E = m \cdot c^2 \quad (2.1)$$

zur Verfügung. Bevor die Teilchen zur Kollision kommen, müssen sie erst erzeugt und beschleunigt werden.

Die Elektronen werden in einer Glühkathode erzeugt und in einem Linearbeschleuniger (LINAC I) auf 62 MeV beschleunigt und ins Synchrotron eingespeist. Dort werden sie auf die Energie des Speicherrings, z.B. 5000 MeV, gebracht und dann in den Ring "injiziert" wobei sie auf die schon im Ring befindlichen Elektronen "gepackt" werden.

Die Positronen werden durch Paarbildung erzeugt. Dieser Effekt tritt auf, wenn Elektronen auf ca. 300 MeV beschleunigt werden und dann auf ein Wolfram-Target gelenkt werden. Aus der entstehenden Bremsstrahlung werden Elektronen-Positronen-Paare erzeugt. Mit Hilfe von magnetischen Feldern werden die Positronen in einem kleinem Speicherring von ca. 30m Umfang, dem Positronen-Intensitäts-Akkumulator (PIA), zu kürzeren und intensiveren Bündeln konzentriert. Mit einer Energie von 480 MeV werden sie ins Synchrotron eingespeist und auf die gleiche Energie wie die Elektronen beschleunigt um dann ebenfalls in den Speicherring injiziert zu werden. Dieser Fullvorgang dauert einige Minuten und danach stehen im Speicherring zwei Teilchenpakete mit je 10^{11} Teilchen zur Verfügung.

Da die beiden Teilchenpakete, die auch als 'Bunche' bezeichnet werden, auf einer gemeinsamen Bahn gegeneinander umlaufen, kann die Strahlführung nur durch Magnetfelder bewirkt werden.

Für einen Speicherring lassen sich vier Kenngrößen festlegen:

- die Strahllebensdauer
- die Strahloptik
- die Luminosität und
- der Energieverlust

Um lange Strahllebensdauern zu erreichen, muß das Strahlrohr evakuiert werden, mit dem Ziel das möglichst wenig Teilchen durch Streuung an den Kernen des Restgases verloren gehen und damit die Strahlintensität nur langsam mit der Zeit abnimmt.

Der Verlauf der Zeitkonstante τ ist in Abbildung 2.1 dargestellt [19]. Danach sollte der Druck im Bereich von 10^{-9} mbar liegen, um eine genügend lange Lebensdauer zu erzielen.

Um die Streuung der Teilchen an der Strahlrohrwand gering zu halten, ist eine genügend freie Apertur, d.h. ein Mindestabstand Strahl - Wand, notwendig. Dieser Abstand kann in Standardbreiten angegeben werden, da die Bunche in transversaler Richtung als gaussförmig verteilt angesehen werden müssen.

Die Strahloptik legt die magnetischen Führungs- und Fokussierungsfelder und damit den Strahlorbit fest. Das Führungsfeld läßt sich aus der Gleichgewichtsbedingung zwischen Lorentzkraft und Zenrifugalkraft ableiten und wird durch Dipolmagnete erzeugt. Die Fokussierungsfelder werden durch Quadrupol- und Sextupolmagnete erzeugt.

Das vertikale Dipolfeld erzeugt die horizontale Ablenkung und legt damit die Sollbahn der Teilchen fest. Das Quadrupolfeld steigt außerhalb der Strahlachse linear an und beeinflusst damit die Teilchen, die sich nicht auf der Sollbahn bewegen. Allerdings führt die Entkopplung der vertikalen und der horizontalen Ebene dazu, dass der Quadrupol nur in einer Ebene fokussiert und in der anderen defokussiert. Durch Zusammenschalten zweier Magnete mit vertauschter Polarität ergibt sich in der Summe eine fokussierende Wirkung.

Da in einem Bunch eine Impulsverteilung vorliegt, werden die Teilchen mit einer Impulsabweichung in den Quadrupolfeldern geringfügig anders fokussiert. Diesen Effekt nennt man Chromatizität und er wird mit den Feldern der Sextupolmagnete korrigiert, da die vertikalen und horizontalen Feldkomponenten nicht entkoppelt sind.

Ein wichtiger Maschinenparameter ist der Q-Wert. Er ist ein Maß für die Teilchenoszillation in vertikaler und horizontaler Richtung um die Sollbahn. Entstehende Resonanzen führen dabei zu einem instabilen Maschinenzustand der zum Verlust des Teilchenstrahls führt.

Um eine effektive Elementarteilchenforschung betreiben zu können, muß eine genügend hohe Trefferwahrscheinlichkeit der Teilchen erreicht werden. Dieser Maschinenparameter wird als Luminosität bezeichnet.

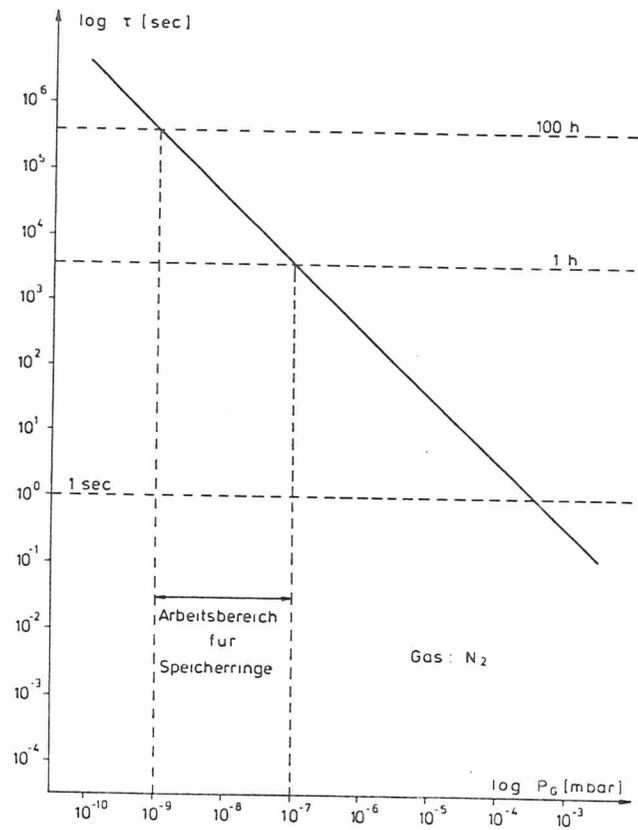


Abbildung 2.1: Die Strahllebensdauer

Sie ist definiert als

$$L = \frac{N}{\sigma} \quad (2.2)$$

wobei N die Ereignisrate für eine bestimmte Reaktion und σ der entsprechende Wirkungsquerschnitt ist. Diese Gleichung läßt sich auch aus den Speicherringparametern entwickeln [19]:

$$L = \frac{1}{4\pi \cdot e^2 \cdot f} \cdot \frac{I^+ \cdot I^-}{\sigma_x \cdot \sigma_z} \quad (2.3)$$

($\sigma_{x,z}$ = Strahlquerschnitt, I = Teilchenströme, f = Umlauffrequenz)

Eine hohe Luminosität kann durch die Zunahme der Teilchenströme und der Verringerung der Strahlquerschnittsfläche erreicht werden. Die Grenze für die Erhöhung der Teilchenströme liegt a) in der verfügbaren Leistung der Beschleunigerstrecken und b) den Raumladungseffekten der Bunche selbst. Dieser Effekt beschreibt die gegenseitige Beeinflussung der Bunche durch das sie umgebende elektromagnetische Feld. Die Verringerung der Strahlquerschnitte ist durch die Quadrupolfelder begrenzt.

Die Berechnung der Luminosität aus den Strahlparametern ist im Prinzip möglich, aber aufgrund der großen Fehler bei der Bestimmung der Strahlgröße ($\sigma_{x,z}$) sehr ungenau. In der Praxis verwendet man im allgemeinen die aus der QED bekannte Bhabha-Streuung als Eichreaktion.

Die wesentlichste Einschränkung der maximalen Strahlenergie von Speicherringen ist der Energieverlust durch die Synchrotronstrahlung. Da jede beschleunigte Ladung elektromagnetische Wellen abstrahlt, gilt dies auch für die radial beschleunigten Teilchen in den Dipolmagneten des Speicherrings. Das Spektrum reicht dabei vom sichtbaren Licht bis zur Röntgenstrahlung. Für die abgestrahlte Leistung im Speicherring gilt [19]:

$$P_\gamma = \left(\frac{2r_e}{3(m_0c^2)^3} \right) \cdot \frac{E^4}{\rho^2} \quad (2.4)$$

(r_e = kl. Elektronenradius, m_0c^2 = Ruheenergie, E = Strahlenergie, ρ = Krümmungsradius)

Dieser Energieverlust muß ausgeglichen werden, was in sogenannten Cavities geschieht. Dies sind Hohlraumresonatoren, in denen durch elektromagnetische Anregung ein E-Feld parallel zur Strahlrichtung schwingt. Der Energiebereich ist begrenzt und liegt bei DORIS II bei max. 5.6 GeV. In der Tabelle 2 sind die wesentlichen Parameter des Speicherrings DORIS II zusammengefasst. Die Abbildung 2.2 zeigt DORIS II in Zusammenhang mit dem gesamten Beschleunigersystem bei DESY.

Parameter von DORIS II	
Umfang	288 m
mittlerer Krümmungsradius	25,8 m
Maximalenergie	5,6 GeV
Energieunschärfe bei $E_{Strahl}=5,0$ GeV	$\sim 10,0$ MeV
typische Luminosität	$\geq 10^{31} \text{cm}^{-2} \text{s}^{-1}$
Anzahl der e^+e^- -Bunche	1
Bunchgröße ($\sigma_x \cdot \sigma_y \cdot \sigma_z$)	$2 \text{ cm} \times 1 \text{ mm} \times 0,1 \text{ mm}$

Tabelle 2.1: DORIS II

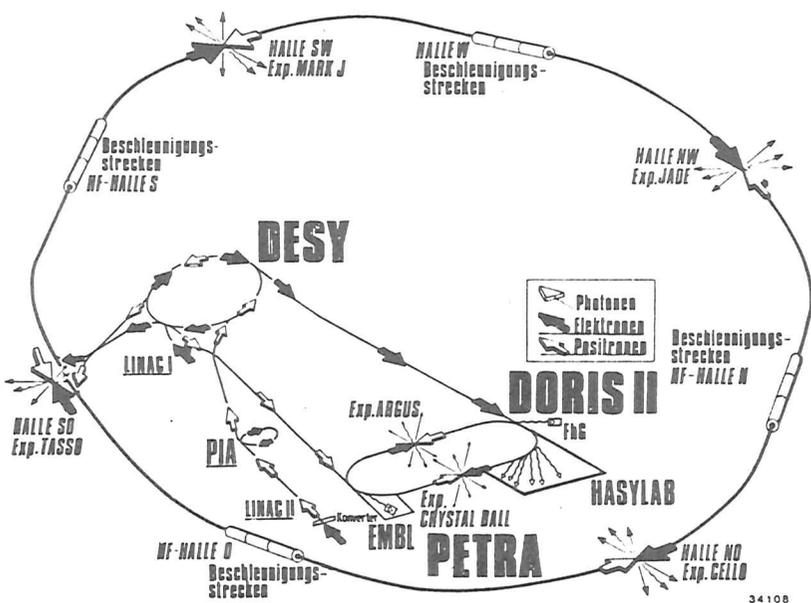


Abbildung 2.2: Der Speicherring DORIS II

Kapitel 3

Das Crystal-Ball Experiment

Da jedes Online-System im Kontext zum Experiment steht, soll nachfolgend der Crystal-Ball Detektor in seinen Komponenten, sowie das Teilchenverhalten im Detektor, erläutert werden.

3.1 Der Crystal-Ball Detektor

Der Detektor befindet sich in einer der beiden Wechselwirkungszone des Speicherrings DORIS II. Der Crystal-Ball ist ein nichtmagnetischer Detektor, d. h. er arbeitet ohne Magnetfeld zur Impulsmessung geladener Teilchen. Seine Hauptkomponenten bestehen aus :

- dem NaJ-Schauerzähler
- dem Innendetektor mit vier Proportionalkammern
- den beiden NaJ-Endkappen
- den Luminositätszählern
- den Time-of-Flight-Countern (TOF)

In Abbildung 3.1 sind die einzelnen Detektorkomponenten skizziert.

3.1.1 Der NaJ-Schauerzähler

Dieser Teil des Detektor besteht aus 672 radial angeordneten Natriumjodid $[NaJ(Tl)]$ -Kristallen, die mit Thallium als Szintillationssubstanz dotiert sind. Jeder einzelne Kristall hat die Form eines Pyramidenstumpfes mit gleichseitiger dreieckiger Grundfläche und einer Höhe von 40 cm. An der Grundfläche eines jeden Kristalls schließt sich eine Photomultiplieröhre an. Die in Abbildung 3.2 dargestellten Kristalle bilden eine Kugel mit einem Durchmesser von 65 cm.

Die Segmentierung des Zentraldetektors führt zu einer Einteilung in 20 gleichseitige Dreiecke, die sogenannten "Major Triangles". Diese sind ihrerseits

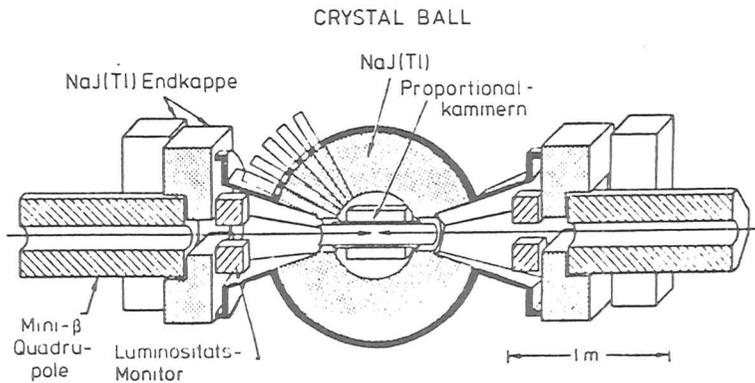


Abbildung 3.1: Die Detektorkomponenten des Crystal-Ball

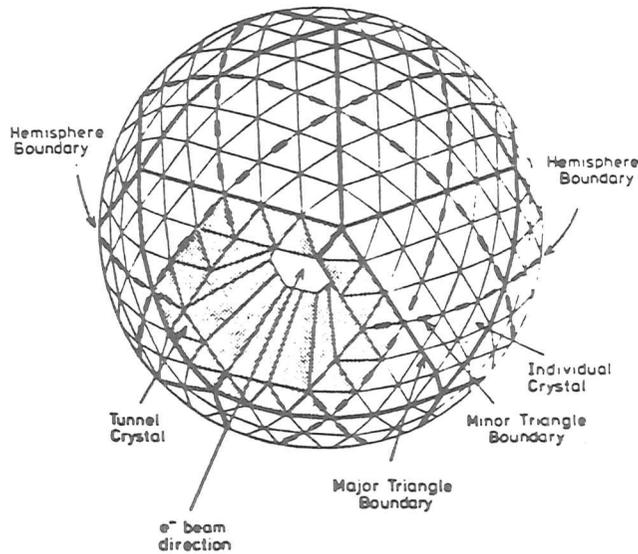


Abbildung 3.2: Der NaJ-Schauerzähler

in je 4 Dreiecke, den "Minor Triangles" unterteilt und diese bestehen schließlich aus 9 Einzelkristallen. Mechanisch ist der Zentraldetektor an der Äquator-ebene in zwei Hemisphären unterteilt, die mittels einer Hydraulik auseinander gefahren werden können. Für den Durchlaß des Strahlrohres wurden an zwei gegenüberliegenden Punkten je 24 Kristalle ausgelassen. Dieser Bereich heißt Tunnelregion. Mit dieser geometrischen Anordnung der Kristalle lassen sich 93 % des Raumwinkelbereichs abdecken.

Eine entscheidende Größe für jeden Detektor ist die Energie- und Ortsauflösung. Bei diesem Detektor beträgt die Energieauflösung

$$\frac{\sigma(E)}{E} = \frac{2.7\%}{E^{\frac{1}{4}}} \quad (3.5)$$

(Teilchenenergie in GeV)

Der Energieabgabeprozess von Elektronen, Positronen und γ -Quant wird in Abschnitt 3.2 auf Seite 16 näher beschrieben.

Die Ortsauflösung des Detektors ergibt sich aus seiner Geometrie. Für die Winkelauflösung des Polarwinkels θ (Ortsvektor zur z-Achse) ergibt sich 6 Grad, da genau 30 Kristalle am Äquator 180 Grad ergeben. Die Meßgenauigkeit läßt sich auf 2 Grad in θ steigern, da man bei Schauern über mehrere Kristalle den Energieschwerpunkt um den Faktor 3 genauer bestimmen kann.

Die Auflösung des Azimutwinkel ϕ (Projektion des Ortsvektors in die x-y Ebene) ist proportional zu $\frac{1}{\sin \theta}$ und wird damit ungünstiger, je dichter ein Schauer in der Tunnelregion auftritt. Für die x-y Ebene ($\sin \theta = 1$) ist die Auflösung in ϕ genau der in θ , also 6 Grad. Direkt am Tunnel ergibt sich nur eine Auflösung von 12 Grad in ϕ , bzw. von 4 Grad für den Energieschwerpunkt, da hier nur 30 Kristalle für 360 Grad vorliegen.

Für den Schutz des Zentraldetektors sind zwei Maßnahmen zu nennen.

- Um die Strahlenbelastung gerade beim Füllen des Speicherrings so gering wie möglich zu halten, da genau in dieser Phase ein Teil der neu injizierten Teilchen sofort aus der Bahn kommen, wird der Ball mechanisch in der Äquatorebene auseinander gefahren und ein Bleischild eingebracht.
- Da das Natriumjodid sehr stark hygroskopisch reagiert und damit seine optischen Eigenschaften irreparabel geschädigt werden, befindet sich der ganze Detektor in einem Trockenhaus. Die Luftfeuchtigkeit in diesem Raum wird stets auf -60°C Taupunkttemperatur und 20°C Raumtemperatur gehalten.

3.1.2 Der Innendetektor

Der Innendetektor besteht aus vier Kammern und diese wieder aus je einer Doppellage Proportionalröhren, deren Anodendrähte an beiden Enden ausgelesen werden. Die Kammern sind mit einem Gasgemisch aus 79% Argon, 20% CO_2 und 1% Methan bei Normaldruck gefüllt. Die vier Kammern überdecken

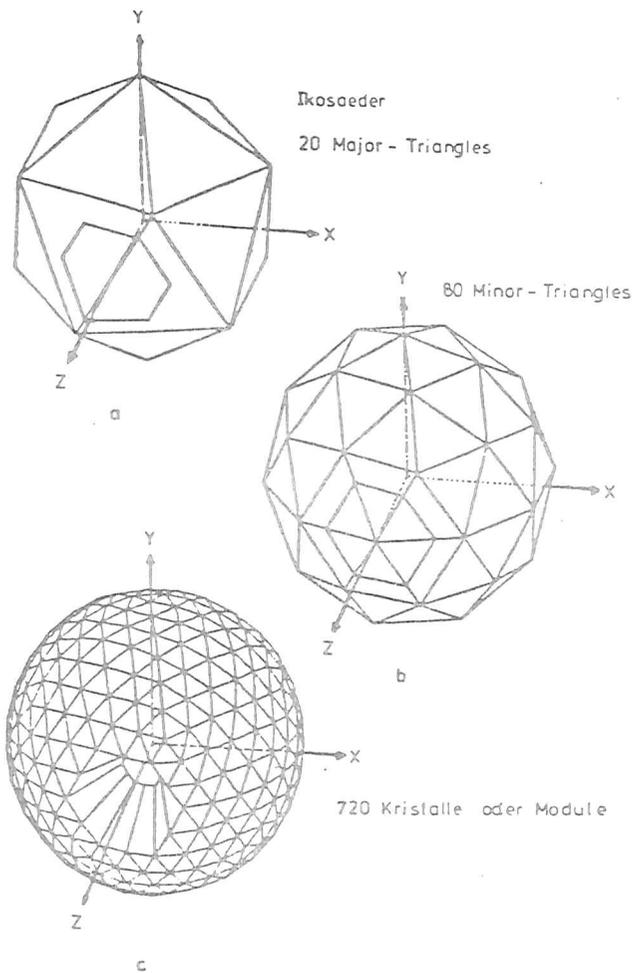


Abbildung 3.3: Die Geometrie des Zentral-Detektors

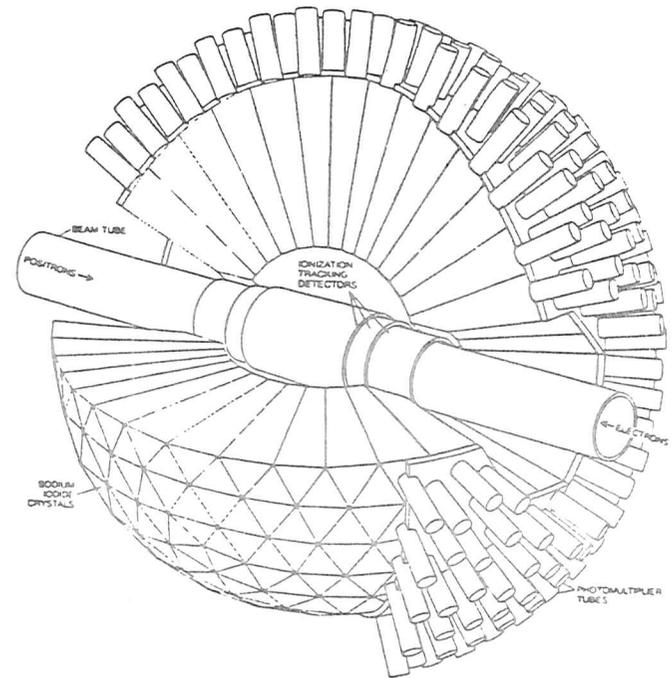


Abbildung 3.4: Der Innen-Detektor

einen Raumwinkelbereich (vom Strahlrohr gesehen) von 98% , 96% , 85% und 75% (siehe Abbildung 3.4).

Die Aufgabe der vier Kammern ist es, mit ihrer Information die Spuren geladener Teilchen zu rekonstruieren. Durchquert ein geladenes Teilchen eine Kammer, so ionisiert es auf seinem Weg Gasmoleküle. Die entstehenden Elektronen driften in dem elektrischen Feld zu den Anodendrähten. Auf diesem Weg werden weitere Gasmoleküle ionisiert, so daß ein verstärkter Stromimpuls auf den Drähten registriert wird.

Da die Anodendrähte in z-Richtung orientiert sind, kann durch Ladungsvergleich der z-Durchgang des Teilchens ermittelt werden.

$$Z = (1 + R) \cdot \frac{Q_L - Q_R}{Q_L + Q_R} \cdot \frac{L}{2} \quad (3.6)$$

(R Drahtwiderstand, L Kammerlänge)

Haben mehrere Drähte reagiert, so kann die Spur eines geladenen Teilchens rekonstruiert werden. Das Ortsauflösungsvermögen liegt bei [18]:

$$\Delta\phi \approx 50\text{mrad}$$

$$\Delta z \approx 1\text{cm}$$

3.1.3 Weitere Detektorkomponenten

Im Bereich hinter den beiden Tunnelregionen befinden sich die Endkappen sowie die Minibetamagnete. Diese speziellen Quadrupolmagnete gehören zum Speicherring und dienen der besseren Fokussierung der Teilchenstrahlen.

Die Endkappen bestehen wie der Zentraldetektor aus NaJ(Tl)-Kristallen. Durch ihre Anordnung wird der erfaßte Raumwinkelbereich auf 98% erhöht. Ihre wesentliche Aufgabe ist der Nachweis von Teilchen, die sich unter kleinem Winkel zur Strahlrichtung bewegen.

Zur Bestimmung der sogenannten "Kleinwinkel-Bhabha-Luminosität" stehen innerhalb der Tunnelregion vier Luminositätsmonitore zur Verfügung (siehe Abbildung 3.1). Jeder Monitor ist sandwichartig aufgebaut, d.h. mehrere Lagen Blei gefolgt von je einer Lage Szintillationsmaterial, wodurch es möglich ist, die Energie der Teilchen zu bestimmen. Mit Hilfe einer fest verdrahteten elektronischen Logik können die bevorzugt unter kleinem Winkel auftretenden Bhabha-Ereignisse durch Koinzidenzen schnell identifiziert werden.

Zur Erkennung von kosmischer Strahlung befinden sich auf dem Trockenhaus großflächige Plastiksintillatoren, die sogenannten 'Time-of-Flight-Counter' (TOF). Treffen sich im Wechselwirkungspunkt die beiden Teilchenpakete, so benötigen die entstehenden Teilchen eine ganz bestimmte Zeit, bis sie die TOF's erreicht haben. Passieren nun Teilchen die Zähler nicht innerhalb des gesetzten Zeitfensters, so wird angenommen, sie sind kosmischen Ursprungs.

3.2 Das Teilchenverhalten im Crystal-Ball

Dringt ein Teilchen in einen Kristall des Detektors ein, so bildet sich ein elektromagnetisches Schauer aus. Eintretende Photonen wechselwirken durch drei Reaktionen mit der Materie [9]:

- Photoeffekt
- Compton-Effekt
- Paarzeugung

Die beiden ersten Prozesse spielen im Energiebereich $E_\gamma \geq 2m_e c^2$ eine untergeordnete Rolle. Für die eindringenden und aus dem Paareffekt entstehenden Elektronen und Positronen lassen sich zwei Energiebereiche für den Energieverlust in Materie unterscheiden.

1. Ionisation:

Liegt die Energie der Teilchen unter einigen 10 MeV, so steht der Energieverlust durch Ionisation und Anregung der Elektronen der NaJ-Atome im Vordergrund. Dieser Prozeß wird durch die Bethe-Bloch-Gleichung beschrieben.

2. Strahlungsbereich:

Hochenergetische Elektronen (Positronen) erzeugen im Coulombfeld der NaJ-Atome Bremsstrahlung. Über diesen Energieverlust läßt sich eine Strahlungslänge X_0 definieren, in der die Energie um den Faktor e abnimmt:

$$E = E_0 e^{-\frac{x}{X_0}} \quad (3.7)$$

Für NaJ(Tl) beträgt die Strahlungslänge $X_0 = 2.52\text{cm}$. Bei einem 40cm langem Kristall sind es damit 16 Strahlungslängen und damit ist eine fast vollständige Energieabgabe der Teilchen gewährleistet.

Liegt die Energie der Bremsstrahlungsquanten weit über 1 MeV, so entstehen neue Elektron/Positron-Paare und danach wieder Bremsstrahlung. Dieser Lawineneffekt wird als elektromagnetisches Schauer bezeichnet. Aus der Strahlungslänge läßt sich für Photonen die mittlere Weglänge errechnen, bevor es ein Paar erzeugt:

$$X_\gamma = \frac{7}{9} X_0 \quad (3.8)$$

Die Energieabgabe von Muonen und Hadronen durch Bremsstrahlung ist dagegen nur sehr klein, da der Energieverlust proportional $1/m^4$ ist. Damit verlieren die schweren Teilchen im wesentlichen über die Ionisation ihre Energie. Für die Hadronen läßt sich eine Absorptionslänge μ definieren:

$$N(x) = N_0 e^{-\mu x} \quad (3.9)$$

(N = Anzahl der Teilchen)

Die 40 cm langen NaJ-Kristalle haben eine mittlere freie Weglänge von ungefähr $1/\mu$.

Die laterale Ausdehnung eines Schauers wird durch den Molière-Radius r_m angegeben. Der Molière-Radius ist proportional der Strahlungslänge X_0 und ergibt sich für NaJ(Tl) zu $r_m = 4.35\text{cm}$. Innerhalb dieses Kreises von $2 \cdot r_m$ wird 95% der gesamten Schauerenergie deponiert. An der Geometrie des Crystal-Ball zeigt sich, daß die 12 Nachbar-Kristalle in diesem Radius liegen (siehe Abb. 3.5) und sich damit die gesamte Schauerenergie aus der Summe der Energien von 13 Kristallen einfach berechnen läßt.

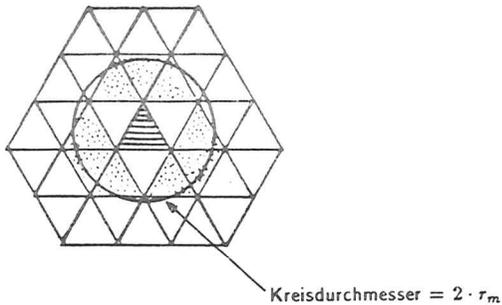


Abbildung 3.5: Die Ausdehnung eines Schauers

Kapitel 4

Die physikalischen Grundlagen

In diesem Abschnitt sollen die physikalischen Grundlagen der Quarkoniumsphysik insoweit erläutert werden, wie sie zum allgemeinen Verständnis des Crystal-Ball Experiments beitragen. Ebenso sollen die physikalischen Begriffe, die in den Abschnitten dieser Arbeit auftauchen, hier ihren Bezug haben.

4.1 Das Quark-Modell

Ausgehend von den Erkenntnissen der Elementarteilchenphysik der beiden letzten Jahrzehnte, läßt sich die Materie aus zwei fundamentalen Gruppen von Fermionen aufbauen, den Quarks und den Leptonen. Es wird von je sechs Vertretern einer Gruppe und deren Antiteilchen ausgegangen (siehe Tabelle 4.1), die strukturlos sind, bei einer heute möglichen Auflösung von $< 10^{-16} \text{cm}$. Die Existenz des τ -Neutrino ν_τ und des top-Quark müssen eingeschränkt werden, da sie bis heute noch nicht überzeugend nachgewiesen wurden.

Die Leptonen			Ladung	Leptonenzahl
e	μ	τ	1	+1
ν_e	ν_μ	ν_τ	0	-1

Die Quarks			Ladung	Baryonenzahl
up	charm	top	+1/3	+1/3
down	strange	bottom	-1/3	+1/3

Tabelle 4.1: Die fundamentalen Bausteine der Materie

Im Gegensatz zu den Leptonen treten die Quarks nur in gebundenen Zuständen auf, wobei zwischen Baryonen, die aus drei Quarks zusammengesetzt sind, und den Mesonen, die aus einem Quark-Antiquarkpaar bestehen, unterschieden wird. Baryonen und Mesonen bilden die Gruppe der Hadronen.

Jedes fundamentale Teilchen wird durch Quantenzahlen beschrieben. Nach dem Pauli-Prinzip, darf ein durch diese Quantenzahlen beschriebener Zustand für Fermionen, nur einfach besetzt sein, d.h. seine Gesamtwellenfunktion muß antisymmetrisch sein. Dies führte aber für das Ω^- aus der Familie der Baryonen zu einem Widerspruch, da es aus drei s-Quarks besteht und somit die Gesamtwellenfunktion

$$\Psi_{total} = \Psi_{Raum} \times \Psi_{Spin} \times \Psi_{Flavour}$$

nicht antisymmetrisch ist [3]. Es wurde eine neue Quantenzahl "Farbe" definiert, die in den drei Zuständen "rot", "grün" und "blau" auftritt. Da die experimentell beobachtbaren gebundenen Zustände der Quarks "farblos" sind, wurde dies zum Postulat erhoben, d.h. daß die beobachtbaren Hadronen Farbsingulets sind.

Aufgrund ihrer Eigenschaften treten die Teilchen in Wechselwirkung miteinander. Heute sind vier Wechselwirkungen bekannt (siehe Tabelle 4.2). Die Quarks nehmen an allen Wechselwirkungen teil, die Leptonen nicht an der starken Kraft, bzw. die neutralen Leptonen auch nicht an der elektromagnetischen Wechselwirkung.

Die fundamentalen Wechselwirkungen			
Wechselwirkung	Austauschteilchen	Symbol	rel. Stärke
Gravitation	Graviton	G	10^{-40}
Elektr.-Magnetismus	Photon	γ	$\approx 1/137$
Schwache WW	Intermediäre Vektorbosonen	Z^0, W^\pm	$\approx 10^{-5}$
Starke WW	Gluonen	g	≈ 1

Tabelle 4.2: Die fundamentalen Wechselwirkungen

Beschrieben werden diese Wechselwirkungen durch die Quantenfeldtheorie. Sie ordnet jeder Wechselwirkung ein Boson als Austauschteilchen zu. Die Stärke der Wechselwirkung wird durch die Kopplungskonstante bestimmt.

Für die elektromagnetische Wechselwirkung ergibt sie sich aus Quantenelektrodynamik (QED) [8]:

$$\alpha = \frac{e^2}{hc} \approx \frac{1}{137}$$

Die Kopplungskonstante α_s , für die starke Wechselwirkung folgt aus der Quantenchromodynamik (QCD) [8]:

$$\alpha_s = \frac{12\pi}{33 - 2N_f} \cdot \frac{1}{\ln(q^2/\Lambda^2)}$$

(q = Viererimpulsübertrag, N_f = Anzahl der Quarksorten,

Λ = (einige) 100 MeV)

Die Gleichung zeigt, daß für große Werte von q^2 und damit für kleine Abstände, die Kopplungskonstante zwischen den Quarks klein wird. Diesen Umstand

bezeichnet man als "asymptotische Freiheit", da sich die Quarks als quasi freie Teilchen verhalten. Umgekehrt nimmt die Bindung zu, wenn der Abstand größer wird, bis zu dem Punkt, an dem aus der entstandenen hohen Energiedichte neue Quarkpaare entstehen. Dieser Einschluß wird "Quark-Confinement" genannt. Die Ursache für dieses Verhalten der Quarks ist in den Austauschteilchen, den Gluonen, zu suchen. Im Gegensatz zu den Photonen, die als Austauschteilchen der elektromagnetischen Wechselwirkung elektrisch neutral bleiben, nehmen die Gluonen eine Farbladung an.

4.2 Elektron - Positron - Reaktionen

Die auftretenden Prozesse bei der e^+e^- -Vernichtung lassen sich im wesentlichen in zwei Gruppen einteilen:

- Bhabhastreueung sowie Leptonen- und Photonerzeugung
- Hadronenerzeugung

Die in der ersten Gruppe angegebenen Prozesse unterliegen der elektromagnetischen Wechselwirkung und lassen sich durch die QED beschreiben. Die Bhabha-Streuung

$$e^+e^- \rightarrow e^+e^- (\gamma)$$

wird in der niedrigsten Ordnung der Störungstheorie durch zwei Feynman-Graphen (siehe Abb. 4.1) beschrieben.

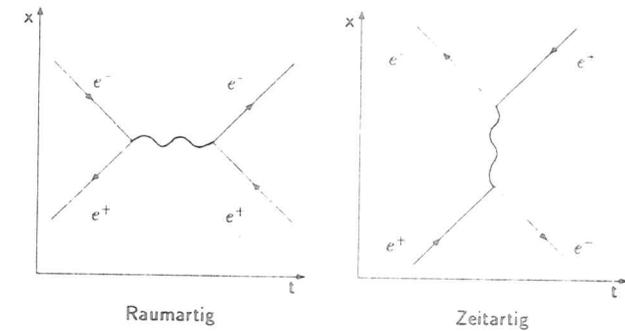
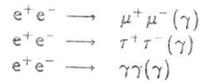


Abbildung 4.1: Die Bhabha-Streuung

Aus der Kenntnis des Wirkungsquerschnittes läßt sich die Luminosität berechnen [7]. Damit ist diese Reaktion eine wichtige Eichreaktion.

Für die elektromagnetische Paarerzeugung lassen sich folgende Prozesse beobachten:



Bei diesen Prozessen vernichten sich ein Elektron und ein Positron zu einem virtuellen Photon, aus dem wiederum ein Leptonenpaar entsteht (siehe Abb. 4.2).

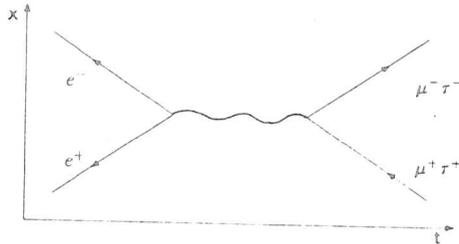


Abbildung 4.2: Die Leptonen-Erzeugung

Der Wirkungsquerschnitt dieser Reaktionen bestimmt sich aus folgender Gleichung :

$$\sigma(e^+ e^- \rightarrow \mu^+ \mu^-) \approx \frac{4\pi}{3s} \cdot \alpha^2 = \frac{87.3}{s} \text{ nbarn}$$

(\sqrt{s} = Schwerpunktsenergie, α = Feinstrukturkonstante)

Bei der Hadronenerzeugung müßen zwei Fälle unterschieden werden.

- ohne Resonanzanregung
- mit Resonanzanregung

Im Fall der Hadronenerzeugung ohne Resonanzanregung koppelt das virtuelle Photon nach den Regeln der QED an ein Quark-Antiquark-Paar. Da nur Farbsingulets zugelassen sind, müßen aus dem Vakuum $q\bar{q}$ -Paare erzeugt werden, so daß im Endzustand "farblose" Quarkkombinationen existieren (siehe Abb. 4.3). Beobachtet werden zwei "Jets", d.h. Bündel von Hadronen, die in der ursprünglichen Richtung des Quark und Antiquark fliegen, da die Hadronen nur einen geringen Impuls quer zur Richtung der ursprünglichen Quarks haben. Diese Kenntnis über die Fragmentierung von Quarkpaaren in Hadronen, sowie die mittlere Zahl geladener Teilchen, dient der Klassifikation der beobachteten Ereignisse im Crystal-Ball (siehe Abschnitt 8.4, Seite 73).

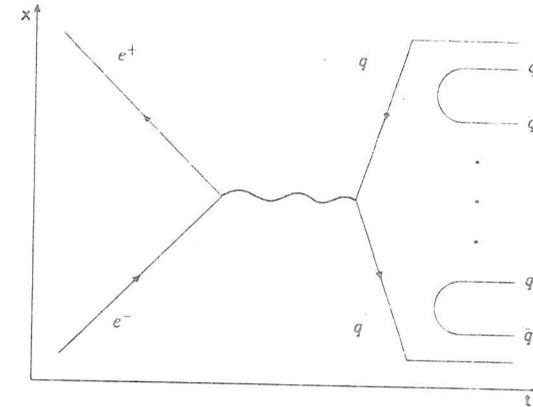


Abbildung 4.3: Die Quark-Erzeugung

Bei der Bestimmung des Wirkungsquerschnittes für die Vernichtung von $e^+ e^-$ -Paaren in Hadronen wird über alle Quarksorten aufsummiert:

$$\sigma(e^+ e^- \rightarrow \text{Hadronen}) = \frac{4\pi\alpha^2}{3s} \cdot \sum_{\text{Farben}} \cdot \sum_{\text{Flavour}} \cdot Q_q^2$$

(Q_q = Quarkladung, \sqrt{s} = Schwerpunktsenergie, α = Feinstrukturkonstante)

Aus dem Wirkungsquerschnitt für die Leptonen- und Hadronen- Erzeugung wird der R-Wert gebildet:

$$R = \frac{\sigma(e^+ e^- \rightarrow \text{Hadronen})}{\sigma(e^+ e^- \rightarrow \text{Leptonen})} = 3 \sum Q_q^2$$

Der Faktor 3 ist ein weiterer Hinweis auf die Existenz der drei Farben in dem Quarkmodell.

Die Abbildung 4.4 zeigt den Verlauf des R-Wertes gegen die Schwerpunktsenergie [8]. Erwartungsgemäß tritt an den Energiewerten ein Sprung auf, der zur Erzeugung einer neuen Quarksorte ausreicht. Zusätzlich treten noch lokale Resonanzen auf. An diesen Punkten reicht die Energie eines Quark-Antiquark-Paares nicht aus, um durch Fragmentation neue Quarkpaare zu erzeugen. Das entstandene Quarkpaar geht in einen gebundenen Zustand ein. Diese Resonanzen werden als Quarkonium-Zustände bezeichnet.

Am Crystal-Ball-Experiment wird das Quarkonium $b\bar{b}$, das als Υ bezeichnet wird, untersucht. Den Graph für diese Resonanzenerzeugung zeigt Abbildung 4.6. Die Resonanzkurve für den $\Upsilon(1S)$ -Zustand zeigt Abbildung 4.5.

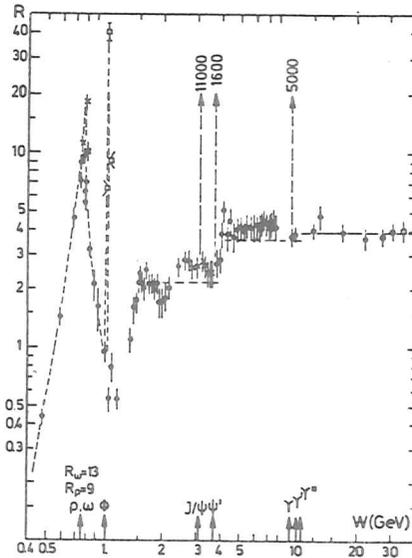


Abbildung 4.4: Energieabhängigkeit des R-Wertes

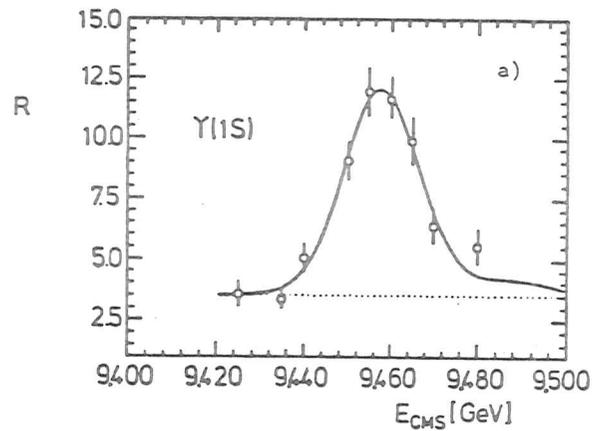


Abbildung 4.5: Resonanzkurve des $\Upsilon(1S)$ -Zustands

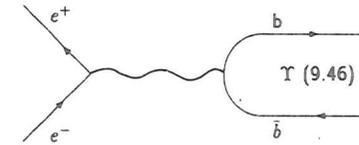


Abbildung 4.6: Die Υ -Erzeugung

Ein Potentialansatz für das Bottomonium-System führt zu verschiedenen Zuständen. Die Klassifikation erfolgt in der aus der Atomphysik üblichen Notation:

- n = Radialquantenzahl
- L = Bahndrehimpuls
- S = Gesamtspinquantenzahl

Für den Gesamtdrehimpuls \vec{J} , der Parität P und der Ladungsparität C gilt folgender Zusammenhang:

$$\vec{J} = \vec{S} + \vec{L}; P = (-1)^{L+1}; C = (-1)^{L+S}$$

Damit gilt für die Terme die folgende Notation:

$$n^{2S+1} L_J$$

Die Abbildung 4.7 zeigt das Niveauschema der Υ -Familie.

Der Zerfall des Υ -Systems zeigt drei Möglichkeiten:

- Vernichtung des $q\bar{q}$ -Paares in Gluonen oder ein virtuelles Photon
- Starker Zerfall durch Erzeugung von $q\bar{q}$ -Paaren
- Übergang der angeregten Zustände in den Grundzustand

Der am häufigsten auftretende Zerfall für den $\Upsilon(1S)$ -Zustand ist der Drei-Gluonen-Zerfall (82%). Da Gluonen Farbe tragen, ist der Ein-Gluonen-Zerfall verboten (Farbsingulett). Der Zwei-Gluonen-Zerfall verletzt die C-Parität. Bei dem Drei-Gluonen-Zerfall fragmentieren die Gluonen zu Hadronen, wobei drei Jets zu beobachten sind.

Der Zerfall in ein Photon ist seltener, da die Wahrscheinlichkeit proportional der Kopplungskonstante ist ($\alpha_s \gg \alpha$).

Für den starken Zerfall gilt die empirisch gefundene OZI-Regel (Okuba, Zweig, Iizuka), die die lange Lebensdauer der Resonanzen ($\tau \sim 10^{-20}$ sec) erklärt. Danach sind Zerfälle stark unterdrückt, bei dem sich das Quark-Antiquark-Paar erst vernichten muß, gegenüber denen, bei denen die Quarks erhalten bleiben und sich neue anlagern.

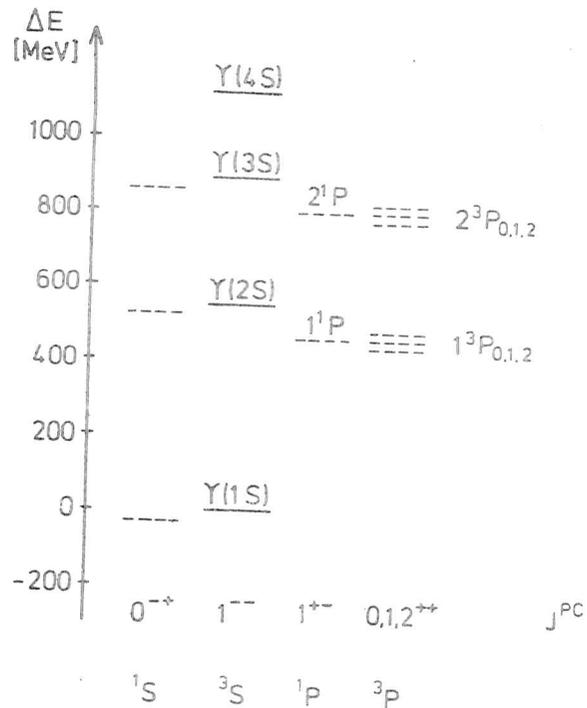


Abbildung 4.7: Niveauschema der Υ -Familie

Die Übergänge zwischen den einzelnen angeregten Zuständen erfolgt über die Fragmentation von leichten Hadronen (z.B. Pionen) oder durch elektromagnetische Übergänge zwischen den Zuständen mit unterschiedlicher Ladungspazität (z.B. S- und P- Zustand).

Kapitel 5

Das Datenaufnahme-System

In jedem Hochenergie-Experiment kommt der Datennahme besondere Bedeutung zu, da durch die hohen Betriebskosten der Speicherringe, die Differenz zwischen angebotener und abgenommener Ereignisrate möglichst klein sein sollte. Die "vertikale" Datenverarbeitung, in der ein Prozessor alle anfallenden Aufgaben eines Experiments übernimmt, gelangt hier an ihre Grenzen.

Der Prozessor, der in der Regel mit einem Multiuser/Multitasking-Betriebssystem läuft, kann nur eine Aufgabe zur Zeit bearbeiten. Dies macht ein Umschalten zwischen den verschiedenen Programmen notwendig. Durch das Sichern des alten Programmstatus und dem Initialisieren des neuen, entstehen Zeitverluste, die der Datennahme verloren gehen.

An dem Datennahme-System des Crystal-Ball Experiments soll die Einprozessor-Verarbeitung näher erläutert werden.

Grundsätzlich muß beim Datenaufnahme-System zwischen der Hardware- und Software, sowie zwischen dem ONLINE- und OFFLINE-Bereich unterschieden werden. Abbildung 5.1 zeigt das Datenfluß-Schema. Der Offline-Bereich ist mit der IBM-Großrechenanlage verbunden, auf der die Datenanalysen gerechnet und komplexe Eich- und Kontrollprogramme verarbeitet werden. Dieser Bereich wird in dieser Arbeit nicht weiter behandelt.

Das Auslesen der elektrischen Signale während der Datennahme wird durch eine fest verdrahtete Elektronik, der sogenannten Trigger-Logik gestartet. Nur wenn bestimmte Triggerbedingungen erfüllt sind, werden die Signale digitalisiert und vom Online-Rechner registriert und kurzfristig gespeichert (3 - 4 Sek.), um dann über eine Datenverbindung, dem sog. IBM-Link, von der IBM-Großrechenanlage auf einer Magnetplatte gesichert zu werden. Diese Daten werden dann 1 - 2 mal pro Tag endgültig auf Magnetbändern gespeichert und stehen damit der physikalischen Analyse dauerhaft zur Verfügung. Jedes ausgelesene Ereignis wird mit einer sogenannten Event- und Run-Nummer zur eindeutigen Identifikation versehen. Als Run wird die Zeitspanne zwischen zwei Füllungen des Speicherringes bezeichnet.

Die physikalische Motivation für die Triggerbedingungen liegt in der Selektion von physikalisch relevanten Ereignissen vom Untergrund, d.h. Strahlreaktionen und Restgasreaktionen im Strahlrohr. So führt die Impulserhaltung

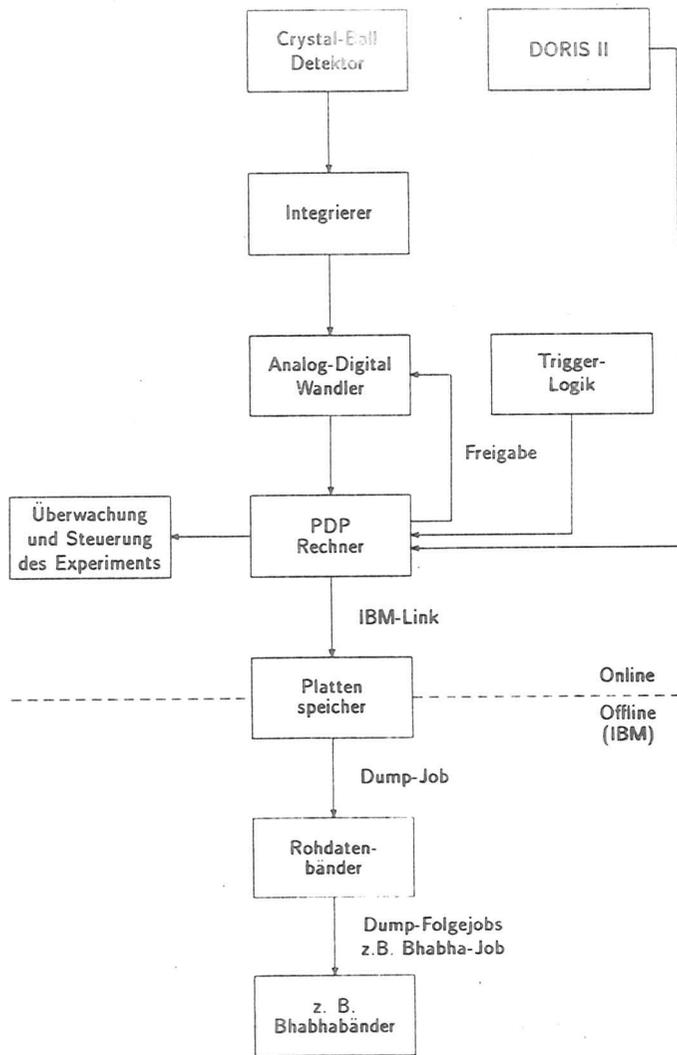


Abbildung 5.1: Der Datenfluß im Crystal-Ball-Experiment

zu einem gewissen räumlichen Gleichgewicht der Energiedeposition d.h. wenn in einem Major-Triangel eine gewisse Energiemenge deponiert worden ist, so fordert man von der gegenüberliegenden Hemisphäre eine ungefähr gleiche Menge.

5.1 Die Hardware des Online-Systems

Nachfolgend sollen die Hauptkomponenten der Hardware des Online-Systems detailliert werden. Die analogen Signale aller 712 Photomultiplier, damit aller NaJ-Kristalle, müssen digitalisiert und ausgelesen werden. Dies erfolgt aus Kostengründen sequentiell, d.h. es existiert nur ein sehr hochwertiger Analog-Digital-Wandler (ADC), der nacheinander die analogen Signale aller Photomultiplier digitalisiert. Dieser Prozeß ist mit etwa 50ms relativ langsam, so daß die analogen Signale in sogenannten "Integrate and Hold"-Modulen (I&H), zwischengespeichert werden müssen [2].

Jedes I&H Modul besitzt Eingänge für neun Photomultiplier, wobei für jeden Eingang zwei Kanäle existieren, der sog. High Energy Channel "HEC" und der Low Energy Channel "LEC" (vgl. Abb. 5.2).

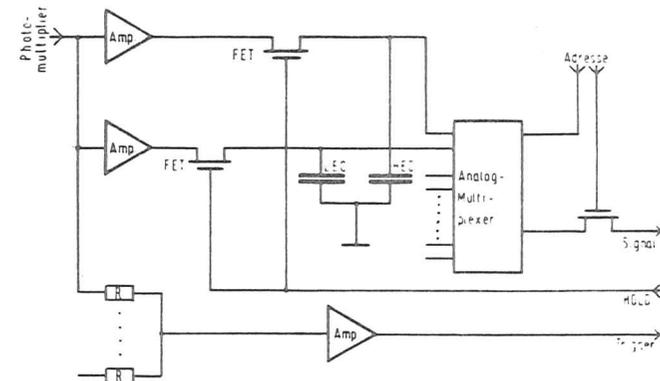


Abbildung 5.2: Das Block-Schaltbild eines 'Integrate and Hold'-Modul

Das Eingangssignal durchläuft in jedem dieser beiden Kanäle einen Verstärker und einen FET-Schalter, der als elektronischer Schalter über ein externes Signal von der Trigger-Logik gesteuert wird. Das Triggersignal entscheidet ob der nachfolgende Kondensator aufgeladen wird oder seine Ladung hält. Das in den Kondensatoren zwischengespeicherte Signal kann über einen Analogmultiplexer, der unter der Kontrolle des NEMO¹ steht, an den Ausgang gelegt werden. Zusätzlich wird das Summensignal aller neun Eingänge auf den Trigger-

¹Adressen-Multiplexer mit Memory

Ausgang des I&H Moduls gelegt und dient damit den Back-to-Back-36-Modulen (BB36) als Eingangssignal.

Die Informationen aus zwei gegenüberliegenden (Back to Back) Minor-Triangles wird von den BB36 analysiert, wobei der Summenausgang der I&H Module für zwei Gruppen von 36 Kristallen angeschlossen ist. Das BB36-Modul bietet die Möglichkeit eine Reihe von internen Bits durch eine Schwellspannung an der Triggerlogik einzustellen. Beim Überschreiten dieser Schwelle werden die entsprechenden Bits gesetzt. In jeden BB36-Modul steht ein 14-Bit breites Wort zur Verfügung, welches an die Triggerlogik zur Analyse weitergeleitet wird. Die Auswahl der Schwellspannung und damit das Setzen der Bits in dem sogenannten Triggerwort, erfolgt aufgrund der gewählten Triggerbedingungen (siehe Abschnitt 5.2 auf Seite 31).

An der Triggerlogik können mit Hilfe eines Schalters verschiedene Triggerbedingungen festgelegt werden. Je nach Einstellung untersucht die Triggerlogik einzelne oder mehrere Bits der von den BB36-Modulen weitergegebenen 14-Bit Worte. Stimmen sie mit dem gewählten Bitmuster der Triggerlogik überein, so wird ein Halte-Impuls für die I&H Module gesetzt. Die Triggerlogik wartet danach auf das Freigabesignal des Online-Computers.

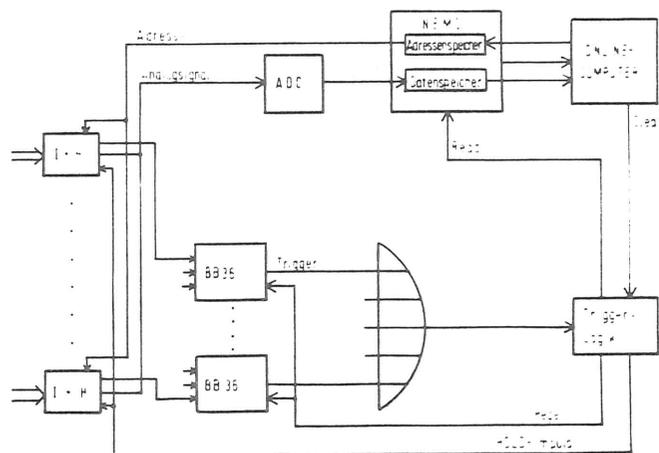


Abbildung 5.3: Die Hardware des Datenaufnahme-Systems

Der gleichzeitig aktivierte NEMO schaltet einzeln für jeden Kristall und Kanal die Signalausgänge der I&H-Module an den ADC (siehe Abb. 5.3). Der digitalisierte Meßwert wird vom NEMO übernommen und in seinem Datenspeicher abgelegt. Sind alle Kanäle digitalisiert, erfolgt ein LAM-Signal ('Look

at me") an den Online-Computer. Dieser liest mittels CAMAC ("Computer Aided Measure and Control System") den NEMO-Datenspeicher aus und speichert seinerseits diese Daten in einem reservierten Bereich seiner Memory. Nach Beendigung des Datentransfers wird über das Freigabesignal an die Triggerlogik das Datennahme-System für das nächste Ereignis freigegeben.

5.2 Die Triggerbedingungen

Für den Speicherring DORIS II ergibt sich an den Wechselwirkungspunkten eine Durchdringungsrates von $\sim 1\text{MHz}$ der Elektronen- und Positronen-Bunche. Neben den erwarteten e^+e^- -Reaktionen treten noch Untergrundreaktionen wie Restgas- oder Strahlwand-Reaktionen auf. Die Aufgabe des Triggers ist die Trennung zwischen erwünschten und unerwünschten Ereignissen. Der Trigger ist als eine Hardware-Logik zu verstehen, in der physikalische Bedingungen fest verschaltet sind.

Es existieren verschiedene Trigger, die je nach Meßprogramm aktiviert werden können. Wenn einer dieser Trigger ausgelöst wird, so wird die Datennahme gestartet. Von den verfügbaren Triggern werden nachfolgend die wichtigsten beschrieben:

1. ETOT-Trigger

Dieser Trigger wird gesetzt, wenn die Summe der deponierten Energie in allen 672 Ball-Kristallen die Gesamtenergie von $E_{TOT} \geq 1800\text{ MeV}$ übersteigt.

2. μ -pair-Trigger

Für diesen Trigger muß die Gesamtenergie die Schwelle von $E_{TOT} \geq 220\text{ MeV}$ erreichen. Zusätzlich müssen in zwei gegenüberliegenden Minor Triangles mindestens 85 MeV gemessen werden. Für diesen Trigger ist ein 'Tunnelveto' möglich. Dieses Veto wird gesetzt, wenn die Energie in den Tunnelkristallen den Wert $2 \times 30\text{ MeV}$ überschreitet. Das Tunnelveto hat die Aufgabe, den Untergrund zu begrenzen.

3. TOPO20-Trigger

Erreicht die Gesamtenergie den Wert $E_{TOT} \geq 770\text{ MeV}$ und wird mindestens in zwei der 20 Ballhemisphären eine Major Triangles-Energie von $E_{Maj} \geq 150\text{ MeV}$ registriert, so wird dieser Trigger ausgelöst.

4. TOPO6-Trigger

Überschreitet die Gesamtenergie den Wert $E_{TOT} \geq 980\text{ MeV}$ und wird in zwei der 6 Ballhemisphären (top, bottom, left, right, front, back) eine Major Triangles-Energie von $E_{Maj} \geq 150\text{ MeV}$ gemessen, wird dieser Trigger gesetzt.

5. DBM-Trigger

Zum Studium des Untergrundes dient der DORIS-Bunch-Mark-Trigger (DBM), der alle 10^7 Wechselwirkungsdurchgänge ($\sim 0.1\text{ Hz}$) gesetzt

wird. Durch dieses 'zufällige' Auslesen der Detektorinformation ist es möglich, sich Kenntnis über den sich zeitlich verändernden Untergrund zu verschaffen.

5.3 Die Software des Online-Systems

Die Aufgabe des Online-Systems läßt sich in drei Bereiche gliedern: der eigentlichen Datennahme, der Funktionskontrolle des Detektors und der Analyse der physikalischen Ereignisse. Entsprechend ihrer Bedeutung haben die einzelnen Programme Prioritäten.

Um eine hohe Eingangsrate zu gewährleisten, muß das hierfür zuständige Programm 'DATIO' auf dem Online-Computer (PDP 11) mit der höchsten Priorität versehen sein. Durch das Vorhandensein mehrerer Detektorkomponenten erfolgt die Datenauslese in mehreren Schritten.

Die in der NEMO-Memory stehenden digitalisierten elektrischen Impulse der Photomultiplier werden über CAMAC in einen reservierten Speicherbereich, den Daten-Buffer, eingelesen. Im zweiten Schritt werden die Daten der Kammern und der Zähler, z.B. die Luminositätsmonitore, sowie die Run-Parameter in den Buffer geschrieben. Da nicht alle Kristalle bei einem Ereignis Signale liefern, werden im nächsten Schritt die Daten in ein "Compressed-Format" sortiert. Anschließend wird der Daten-Buffer in einen als FIFO (First In First Out) organisierten Disc-Buffer kopiert [2].

Um auf die Daten des nächsten Ereignisses schnell genug reagieren zu können, obwohl noch auf den Daten des vorhergehenden Ereignisses Operationen (Compressed, ...) ausgeführt werden, existiert ein zweiter Daten-Buffer. Dieser kann Daten nach dem zweiten Schritt auf den Parallelbuffer aufnehmen, d.h. werden in dem Buffer A die Daten neu organisiert, können im Buffer B die Signale des nächsten Ereignisses eingelesen werden (siehe Abb. 5.5).

Im weiteren Verlauf greift auf den Disc-Buffer, in den die Buffer A oder B kopiert wurden, ein Daten-Transferprogramm zu, das über das PADAC-System (siehe Abschnitt 5.4, Seite 35) ca. alle 5 Sekunden die Rohdaten zur IBM überträgt. Hier werden sie über eine Datensicherungsprozedur auf Magnetbänder abgespeichert. Abbildung 5.4 zeigt den Datenfluß des Datennahme-Systems.

Die Funktionskontrolle des Detektors umfasst alle Parameter, deren Kontrolle alle paar Sekunden oder Minuten notwendig ist. In diesem Komplex ist die Prüfung

- der Kristalle und Photomultiplier mit Hilfe des "Flasher-Systems"
- der wichtigsten Versorgungsspannungen.
- die Temperaturen wichtiger Komponenten und
- die Feuchtigkeit- und Drucküberwachung des Crystal-Ball

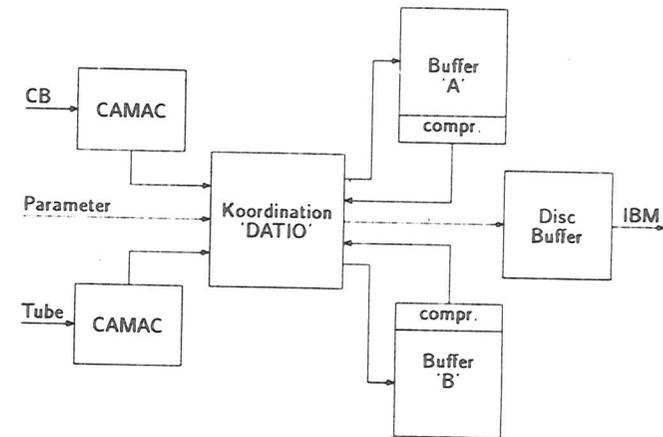


Abbildung 5.4: Der Datenfluß im Datenaufnahme-System

enthalten. Diese Kontrollfunktionen werden von sog. 'Scheduled Tasks' geleistet, die bei auftretenden Fehlern entsprechende Meldungen an der Rechner-Konsole ausgeben.

Um während der Datennahme einen Teil der physikalischen Ereignisse sofort zu analysieren, steht ein Programmsystem, die sog. "Pipeline", zur Verfügung. Jede dieser Pipeline-Tasks ist für eine spezielle Aufgabe programmiert, wobei die Kontrolle und Kommunikation dieser Tasks untereinander von einem Pipeline-Driver organisiert wird (vergl. Abb. 5.5). So überwacht der Driver einen Speicherbereich, den Analyse-Common "PHYS-Common", in den die Rohdaten eines Ereignisses von der Datennahme-Task DATIO kopiert werden. Diese Anforderung an DATIO wird über eine "Flag" gesteuert, d.h. DATIO prüft ein bestimmtes Speicherwort. Ist es gesetzt, so wird der Kopiervorgang gestartet. Über solche Flags findet die gesamte Kommunikation des Online-Systems statt. Der Pipeline-Driver selbst läßt sich extern durch das sog. "Button-Board" steuern, d.h. über die Stellung von Schaltern.

Die wichtigsten Pipeline-Tasks sind die Analyse-, die Histogramm- und die Display-Tasks. Die Analyse-Task prüft den Hardware-Status, berechnet die in den Kristallen deponierte Energie aus den ADC-Werten, bestimmt "Connected Regions" und versucht die Spuren geladener und ungeladener Teilchen zu rekonstruieren. Die von den Histogramm-Tasks erzeugten Graphen und Verteilungen werden als Datenfiles permanent gespeichert und können nach Bedarf ausgedruckt oder über einen Wahlschalter am Button Board auf einen Farb-TV dargestellt werden. Die Display-Tasks gestatten ebenfalls über Wahlschalter am Button Board die Darstellung von verschiedenen Projektionen des Detektors.

Der Pipeline-Bereich des Datennahme-Systems zeigt deutlich den Nachteil

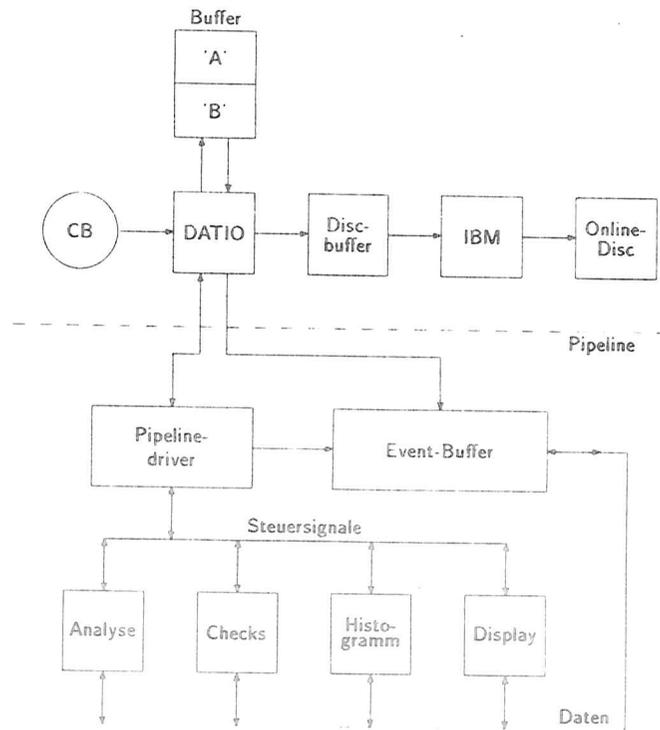


Abbildung 5.5: Die Pipeline des Online-Systems

der "vertikalen" Datenverarbeitung. Die Aufgaben (z.B. Histogramm und Event-Display) werden sequentiell abgearbeitet, obwohl eine parallele Verarbeitung möglich ist. Die notwendigen Umschaltzeiten zwischen den verschiedenen Tasks, führen zu Zeitverlusten und reduzieren damit die Eingangsrate des Datennahme-Systems.

Eine wichtige Größe für ein Datennahme-System ist die sogenannte Totzeit. Sie spiegelt das Verhältnis zwischen der Triggerrate und der Verarbeitungszeit, d.h. dem Auslesen, Speichern und Prüfen der Daten, wieder. Für das Online-System des Crystal-Ball ergeben sich bei einer Triggerrate von ca. 4-6 Hz und einer Verarbeitungszeit von typischerweise 50 - 60 ms bis zu 30% Totzeiten.

5.4 Das PADAC-System

Da bei den heutigen Hochexperimenten das Online-System aufgrund der großen Datenmengen nicht mehr die Sicherung der gewonnenen Daten leisten kann, sind alle Online-Systeme bei DESY an die IBM-Großrechenanlage angeschlossen. Dies setzt ein Kommunikations-System zwischen Großrechner und Online-Rechner voraus. Bei DESY wurde zu diesem Zweck das "Parallel Data Acquisition and Control System" entwickelt.

Dieses System erlaubt den Online-Rechnern über den eigenen Systembus mittels eines PADAC-Interface mit Systemen und Betriebsmitteln, die außerhalb des Rechnerbusses liegen, zu kommunizieren. Dies kann die IBM-Anlage oder das FPSS (siehe Abschnitt 6.1. Seite 37) sein.

Die Transfargeschwindigkeit zwischen den Rechner liegt bei 0.5 Mbyte/sec bei einer Entfernung bis zu 2 km. Zur Zeit sind etwa 44 Prozeßrechner durch dieses sternförmig aufgebaute Online-Netz mit dem IBM-Rechenzentrum verbunden [12]. Ausgehend von einem Prozeßrechner, z.B. einer PDP 11, sieht die Netzverbindung folgendermaßen aus:

Auf dem Systembus des Prozeßrechners wird eine elektronische Steckplatine, das PADAC-Interface, angebracht. Dieses Interface wird durch Breitbandkabel mit dem sogenannten PADAC-Crate-Controller verbunden. Jedes Crate verfügt über einen intern standardisierten Bus, der unabhängig ist vom jeweils angeschlossenen Prozeßrechner. In diesem PADAC-Crate ist Platz für 12 Module, die an beliebiger Stelle positioniert werden können, da die interne Adressierung der Module positionsunabhängig ist. Ein solches Modul kann z.B. ein "EXP-Modul" sein, welches mittels zweier Koaxialleitungen den Datentransfer zur IBM-Rechenanlage ermöglicht. Diese Leitungen enden in zwei sogenannten Multiplexern, die den Datentransfer aller angeschlossenen Prozeßrechner zur IBM-Rechenanlage organisieren und verwalten [5].

Kapitel 6

Das Multi-Prozessor-System

Das im folgenden beschriebene Konzept für eine "horizontale" Datenverarbeitung, stellt eine Alternative zu dem CPU-Nadelöhr der Einprozessor-Verarbeitung dar, das im vorangegangenen Abschnitt beschrieben wurde. Solch ein Multiprozessor-System erfordert neue Methoden der Kommunikation, basierend auf den "Local Area Networks" (LAN). Die Methoden und Techniken sollen nachfolgend näher erläutert werden [11].

Ein Multi-Prozessor-System verfügt über mehr als einen Prozessor, wobei unter Prozessor eine CPU mit entsprechender Peripherie zu verstehen ist. Die Merkmale eines Multi-Prozessor-Systems lassen sich durch vier Punkte charakterisieren :

- die Prozessoren sind miteinander gekoppelt
- die Prozessoren werden von einer zentralen Systemaufsicht überwacht
- die Prozessoren müssen miteinander kommunizieren und kooperieren
- die Prozessoren teilen sich Ein/Ausgabe-Einrichtungen und periphere Geräte

Bei dem für diese Arbeit verwendeten Prozessor-System handelt es sich um ein homogenes System, da alle Prozessoren vom gleichen Typ sind. Die Kopplung und Kommunikation der Prozessoren untereinander wird durch ein lokales Netz, dem "Fast Package Switching System" (FPSS) gewährleistet /F58/. Die Systemaufsicht und die Bereitstellung von Ein/Ausgabe-Einheiten (z.B. Terminals, Drucker, etc.) und peripheren Geräten (z.B. Disc, Floppy, etc.) erfolgt durch einen Mutterrechner, dem HOST. In diesem Fall ist es ein Prozeßrechner vom Typ PDP 11/23. Die Vernetzung des gesamten Systems ist sternförmig, d.h. alle Informationen werden über einen zentralen Knoten verteilt. Die Verteilung ist nicht nach Prioritäten geordnet, sondern sequentiell, nach dem zeitlichen Auftreten der Netzanforderung. Abbildung 6.1 zeigt die sternförmige Verknüpfung mehrerer Empfänger- und Sender-Module. Nachfolgend sollen die Komponenten des Multi-Prozessor-Systems präzisiert werden.

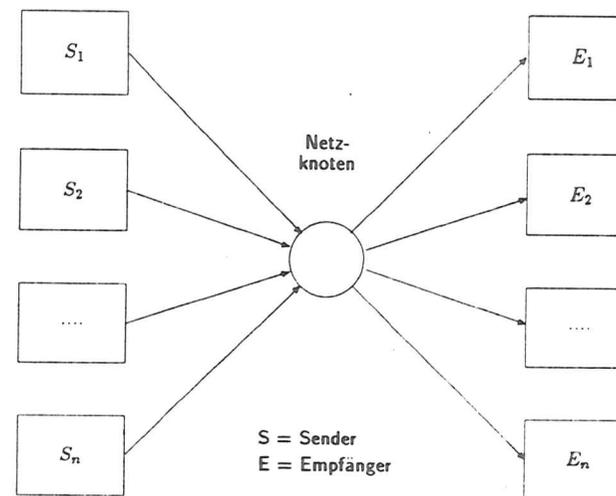


Abbildung 6.1: Ein sternförmiges Netzwerk mit Zentralknoten

6.1 Das FPSS

Beim 'Fast-Package-Switching-System' (FPSS) handelt es sich um eine Hardware-Konfiguration, mit dem ein schneller serieller Datenaustausch zwischen Prozeß-Rechnern, z.B. der PDP-Serie, möglich ist. Dieses System wurde am DESY entwickelt [5].

Der Grundgedanke ist die Umsetzung von paralleler in serielle Information. Dies erfolgt durch einen Parallel/Seriell-Konverter (siehe Abb 6.2). So wird ein Rechnerwort, das aus 16 parallelen elektrischen Signalen besteht, die eine bestimmte Bitfolge repräsentieren, durch den Konverter in 16 seriell zu übertragene Signale umgewandelt. Diese serielle Signalfolge kann somit durch ein einfaches Koaxialkabel übertragen werden. Auf der Empfängerseite wird durch einen Seriell/Parallel-Konverter diese Information wieder decodiert. Die Taktfrequenz beträgt 50 MHz und damit liegt die Übertragungsrate bei 0.4 Mbyte/sec zum/vom Netzknoten und 20 M. yte/sec im Knoten

Die zu übertragenden Rechnerworte werden in Paketen zusammengefasst, die mit einer fixierten Kontrollsequenz beginnen und enden. Solch ein Datenpaket kann bis zu 8 kbyte umfassen. Das durch das FPSS verknüpfte Rechnernetz ist sternförmig, und es können bis zu 256 Rechner angeschlossen werden.

Zu jedem angeschlossenen Rechner gehören zwei Hardware-Module. Diese Module werden in Geräterahmen, sogenannten Crates, eingeschoben. Bei den Crates wird zwischen dem PADAC- und FPSS-Knoten-Crate unterschieden.

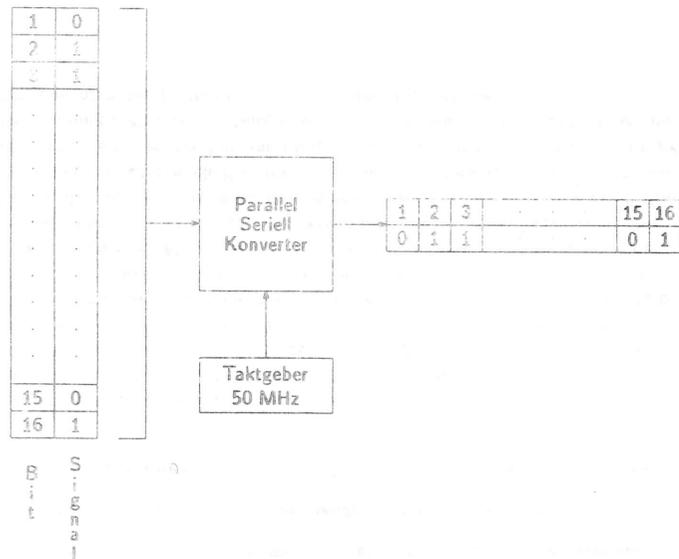


Abbildung 6.2: Parallel/Seriell-Konverter

Das erste Crate ermöglicht über ein Interface die Anbindung an den Systembus des Rechners (siehe Abschnitt 5.4) und das zweite Crate der Vernetzung der Prozessoren. Die Verteilung der Datenpakete im Netzknoten, d.h. welcher Empfänger bekommt welches Senderpaket, erfolgt durch einen Adresscode, der jedes Datenpaket eindeutig kennzeichnet.

6.2 Das TRON-System

Das TRON-System ('Transparent-Onlining in FPSS-connected Computer Networks') [13] beinhaltet die Software für ein lokales Netzwerk aus DEC PDP-11-Computers, die über das FPSS vernetzt sind. Es bietet

- eine hochentwickelte Kontrollstruktur für alle Computer-Aktivitäten im Netz
- File-Transfer durch das Netz
- Fehler- und Testfunktionen

Die TRON-Software muß auf jedem PDP-Computer im Netz installiert werden und ist lauffähig unter dem 'Real-Time System Executive Operating System

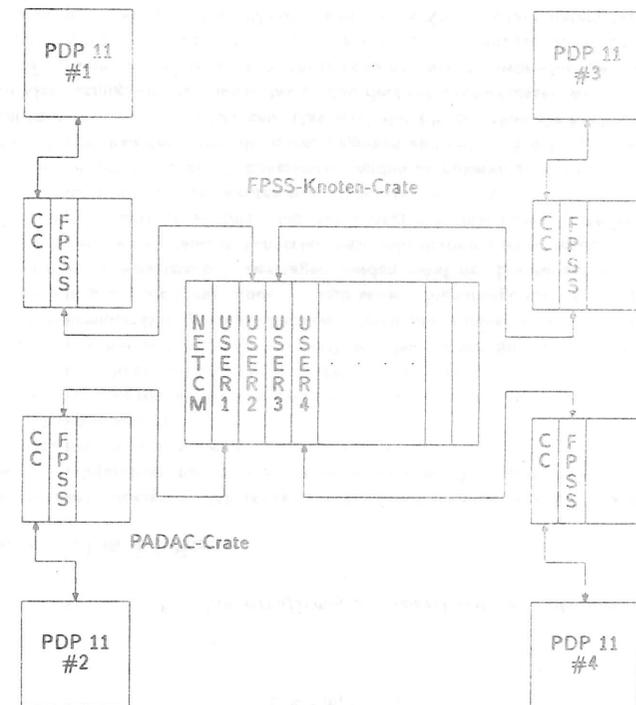


Abbildung 6.3: Die Komponenten des FPSS

(RSX-11) der PDP-Serie.

Das TRON-System gliedert sich in drei Hauptbereiche:

- der Ein-/Ausgabe-Organisation ins/vom Netzwerk (FP-Handler)
- der interaktiven Kommandoausführung und -überwachung (MCX-Monitor)
- Unterprogrammaufrufe aus Benutzerprogrammen (FPX-Subroutinen)

Der sog. FP-Handler organisiert die gesamten Ein-/Ausgabe-Anforderungen zwischen/auf den Rechnern. Diese Anforderungen können kommen:

- aus dem Netzwerk (von anderen Rechnern).
- vom MCX-Monitor oder

- von einem Benutzerprogrammen

Sie werden in einem speziell reserviertem Speicherbereich, dem "FPCOM", durch den FP-Handler verwaltet.

Der MCX-Monitor offeriert dem Benutzer eine Kommandoliste, die den Filetransfer, Jobüberwachung und andere FPSS-Funktionen aktivieren. Der Monitor wird an einem Terminal durch den Befehl

```
> MCX n
```

gestartet. Als Antwort erscheint die Sequenz

```
MCX n >
```

die die Eingabebereitschaft des MCX-Monitors anzeigt. Der Buchstabe 'n' symbolisiert die Computernummer des Empfängers. Der MCX-Mode kann durch die Escape-Taste (ESC) verlassen werden.

Der Datentransfer durch das Netzwerk aus einem Benutzerprogramm heraus, wird durch die Routine "FPX" organisiert. Die beiden Unterprogramme "GETFP" und "PUTFP" dieser Prozedur ermöglichen das Empfangen und Senden von Daten.

Allen ausgetauschten Datenpaketen steht eine fest formatierte Protokollsequenz voran, die die notwendigen Identifikationsparameter dieser Datenpakete enthält. Dadurch ist nur während des Transfers das Netzwerk exklusiv für die beiden kommunizierenden Computer aktiv. Ist die Übertragung abgeschlossen ist das Netz sofort wieder transparent für alle Computer im Netz.

6.3 Der HOST-Rechner

Als HOST-Rechner wurde für diese Arbeit ein PDP 11/23-Rechner, mit einem Speicher von 256 kByte, benutzt. Als Peripherie stehen dem Rechner Magnetplatten, Drucker, Floppies und Terminals zur Verfügung. Die Aufgabe des HOST gliedert sich in drei Teile:

- Programm-Entwicklung für die Prozessoren (sog. Cross-Software)
- Bereitstellung von peripheren Geräten sowie Ein/Ausgabe-Einheiten
- Kontrolle und Modifikation des Prozessor-Systems (Monitoring)

Der bereitgestellten Cross-Software kommt besondere Bedeutung zu. Ihre Aufgabe ist es, aus der Assembler- oder einer anderen Programmiersprache einen für die Prozessoren ausführbaren Maschinencode zu generieren. Dieser Maschinencode basiert auf dem festen Instruktionssatz des Prozessors und ist deshalb auch nur auf ihm exekutierbar. Zur Zeit existiert ein 'Texas-Cross-Compiler' (TCC) auf Basis der Programmiersprache FORTRAN und ein 'Texas-Cross-Assembler' (TCA) [15].

Ist das Prozessor-System im Betriebszustand, so ist die Aufgabe des HOST Command-Management und System-Monitoring. Das Erste umfasst die Interpretation und Ausführung eines Befehls des Prozessor-Betriebssystems, d.h.

ein Betriebs-System-Befehl muß dem Prozessor über den HOST zugänglich sein. Die Kontrolle aller Prozessoren-Aktivitäten im Netz erfolgt über ein Monitor-System. Dadurch ist es möglich, den Zustand eines Prozessors jederzeit zu kontrollieren und zu modifizieren.

Die beschriebenen Aufgaben des HOST kann dieser nur wahrnehmen, wenn der Systembus der PDP 11, der sogenannte UNIBUS, an das FPSS-Netz angebunden ist. Dies erfolgt durch ein PADAC-Interface, das den Zugriff auf das FPSS-Knoten-Modul ermöglicht.

6.4 Der PADAC-Microprozessor

Hierbei handelt es sich um den Microprozessor TMS 9900 der Firma Texas Instruments. Es ist ein 16-Bit frei programmierbarer Prozessor. Er ist mit einem 64 kbyte direkt adressierbaren Speicher ausgestattet, davon sind minimal 2 kbyte, maximal 8 kbyte als PROM ("Programmable Read only Memory") vorgesehen. Der Speicher ist byteadressierbar (8 bit), als auch wortadressierbar (16 bit). Der verfügbare Instruktionssatz umfasst 69 Befehle, einschließlich der Multiplikation und Division. Für die Speicheradressierung stehen 7 Adressierungsarten zur Auswahl, die auf den drei grundlegenden Prinzipien der Adressierung beruhen: der direkten, der indirekten und der relativen Adressierung. Als Arbeitsregister stehen 16 allgemeine Register zur Verfügung, mit denen das "Workspace-Konzept" dieses Prozessors verbunden ist. Weitere Kennzeichen des PADAC-Microprozessors ist die Struktur der Programmunterbrechung ("Interrupt") und der schnelle Datentransfer in/aus dem Speicher mittels DMA ("Direct Memory Access"). Für den Interrupt stehen 16 Leitungen, die nach Prioritäten geordnet sind, zur Verfügung [5].

Zusammenfassend gilt für das Workspace-Konzept, der Interruptstruktur und dem DMA, das sie den Prozessor zu einem leistungsfähigen System machen. Bevor diese drei Merkmale näher beschrieben werden, soll kurz die TMS 9900-Architektur erläutert werden.

6.4.1 Die TMS 9900 Architektur

Alle heute gängigen Microprozessoren basieren auf dem Prinzip der "von Neumann-Maschine". Danach besteht ein Prozessor aus einer zentralen Rechen-einheit (CPU), dem Speicher und einer Ein/Ausgabe-Einheit

In der Regel sind in der CPU sogenannte Register integriert, wobei dem Prozessorstatus-Register (PS) und dem Programmschritt-Zeiger (PC) eine besondere Bedeutung zukommt. Der PC zeigt immer auf die Adresse in dem Speicher, in der der nächste Programmbefehl steht. Im Statusregister werden die Ergebnisse der Vergleichsoperationen (gt, le, overflow, etc.) in einzelnen Bits gespeichert.

Ausgehend von diesem Grundkonzept, lassen sich Prozessoren in der Organisation ihrer arithmetischen und logischen Operationen unterscheiden. Der TMS 9900 gehört danach zu den Prozessoren mit einem universellen Register-

satz (typisch 8 oder 16 Register), wobei seine sogenannte "Speicher zu Speicher Architektur" das Workspace-Konzept ermöglicht [16].

6.4.2 Das Workspace-Konzept

Darunter ist zu verstehen, daß die 16 allgemeinen Register nicht in der CPU des Prozessors als interne Register liegen, sondern ein aus 16 aufeinander folgenden Worten bestehender Teil des Speichers als Registersatz fungiert [10]. In einem speziellem Register, dem "Workspace-Pointer" (WP) der CPU, steht nur noch die Speicheradresse des ersten allgemeinen Registers R0 (siehe Abb. 6.4). Bei

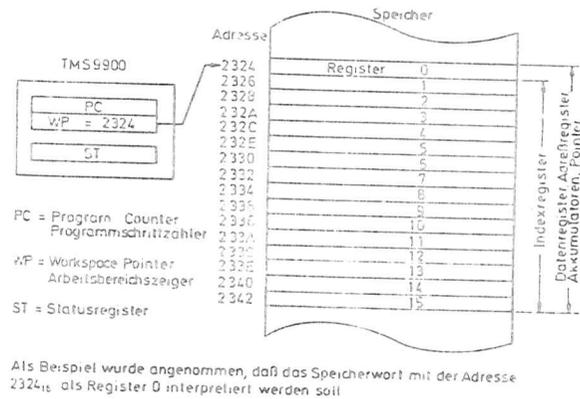


Abbildung 6.4: Der Workspace-Pointer der CPU

Operationen mit einem der Register R0 - R15 wird die absolute Speicheradresse aus den Inhalt des WP's berechnet.

Mittels des Workspace-Pointers ist es möglich die 16 allgemeinen Register in irgendeinen Bereich des Speichers zu legen und somit die komplette Arbeitsumgebung eines Programms umzuschalten. In Abbildung 6.5 wird dieser "Contex-Switch" am Beispiel der Unterprogrammtechnik verdeutlicht. Der Vorteil dieser Umschalttechnik liegt in der kurzen Zeitspanne von ca. 10µsec (je nach Adressierungsart), da im Prinzip nur der Workspacepointer umgesetzt wird. Im Vergleich dazu kann diese Umschaltzeit bei anderen Prozessoren bis zu zehnmal höher liegen, da die internen Register in der CPU erst in einem speziellem Speicherbereich, dem Stack, gerettet werden müssen.

6.4.3 Die Interrupt-Struktur

Um den Anforderungen einer sinnvollen Zusammenarbeit mit der realen Umwelt gerecht zu werden, muß für externe Geräte und Systeme die Möglichkeit

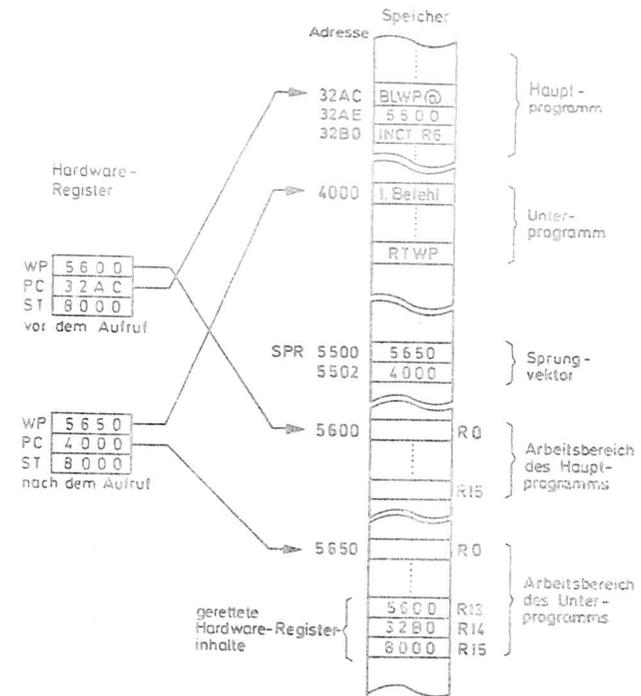


Abbildung 6.5: Der Contex-Switch des TMS 9900

bestehen, das gerade ausführende Programm im Prozessor zu unterbrechen. Bei diesen Interrupts lassen sich zwei Typen unterscheiden:

- der Software-Interrupt und
- der Hardware-Interrupt.

Bei dem Software-Interrupt wird von dem Hauptprogramm, daß gerade im Prozessor abgearbeitet wird, zyklisch ein Unterprogramm aufgerufen, das die Statusleitungen externer Geräte überprüft. Ist eine davon aktiv, so wird das entsprechende Serviceprogramm ("Device-Service-Routine = DSR") für dieses Gerät aufgerufen um die externe Anforderung zu bearbeiten. Ist dies geschehen, so wird mit dem Hauptprogramm fortgefahren (siehe Abb. 6.6).

Bei dem Hardware-Interrupt sendet das Eingabe-Gerät über die sogenannten Interrupt-Leitungen ein Signal an den Prozessor. Die Hardware veranlaßt

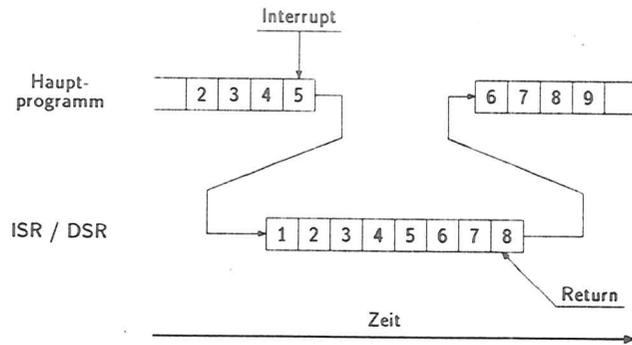
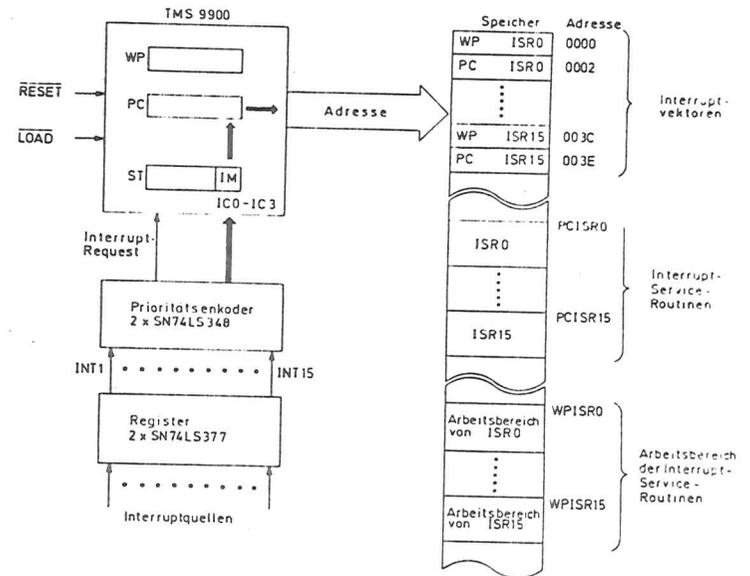


Abbildung 6.6: Der Interrupt-Fluß

die Unterbrechung des Hauptprogramms und startet die entsprechende Service-Routine. Eine optimale Anpassung an die realen Bedingungen wird allerdings durch den Vektor-Interrupt gewährleistet. Hierbei sucht sich die Interruptquelle die entsprechende Service-Routine selbst. Bei dem TMS 9900 wird diese Vektor-Interrupt-Struktur (siehe Abb. 6.7) verwendet und soll aus diesem Grund näher erläutert werden [10].

Der Prozessor verfügt über vier Interrupt-Request-Eingänge (IC0 - IC3) die eine Unterscheidung von $2^4 = 16$ Interruptquellen ermöglichen. Mit Hilfe des Prioritätsdecoders wird ein 4-Bit-Wort erzeugt, das die Priorität des Interrupts widerspiegelt. In dem Prozessor-Statuswort sind ebenfalls 4 Bit als Interrupt-Maske reserviert. Durch Vergleich dieser beiden 4-Bit-Worte kann der Prozessor feststellen, ob dieser Interrupt zugelassen wird oder nicht. Passiert der Interrupt diese Maske, so wird das gerade ausführende Programm unterbrochen. Dies geschieht nach folgendem Schema:

Aus dem Interrupt-Vektor (d.h. aus seiner Priorität) wird die Adresse des Sprungvektors berechnet. Zu jedem Sprungvektor gehören zwei 16-Bit-Worte. Im ersten steht der Workspace-Pointer und im zweiten die Startadresse der entsprechenden Interrupt-Service-Routine (ISR). Die Sprungvektoren aller Service-Routinen müssen im Speicher in den ersten 32 16-Bit-Worten stehen. Die Adresse des berechneten Sprungvektors wird zwischengespeichert. Die notwendige Programmumschaltung erfolgt nun durch den "Context-Switch". Im ersten Schritt werden die momentanen Inhalte der drei Prozessor-Register, d.h. der Workspacepointer (WP), der Programmcounter (PC) und das Statusregister (ST), in die Register R13 - R15 des neuen Arbeitsbereichs gerettet. Anschließend werden die zwischengespeicherten Adressen der ISR in das Workspace- und Programmcounter-Register geladen. In die Interrupt-Maske des Statuswortes wird zuletzt der neue Interruptvektor (alt minus eins) geladen. Damit wird gewährleistet, das die gerade aktivierte ISR nur von einem



- WP = Workspace Pointer
- PC = Program Counter
- ST = Statusregister
- IM = Interruptmaske
- ISR = Interrupt-Service-Routine (IRS)
- PCISR_n = Startadresse der ISR mit der Priorität n
- WPISR_n = Adresse des Workspace mit der Priorität n

Abbildung 6.7: Der Vektor-Interrupt des TMS 9900

Interrupt höherer Priorität unterbrochen werden kann. Ist der Interrupt abgearbeitet, wird der vor der Unterbrechung existierende Prozessorzustand durch das Zurückladen der drei Register R13 - R15 in die entsprechenden Prozessorregister erreicht.

Durch die Verknüpfung des Vektor-Interrupts mit dem Context-Switch, ergeben sich sehr kurze Reaktionszeiten für den Prozessor auf externe Anforderungen. Da Online-Systeme in der Regel Real-Time-Anforderungen unterliegen, eignet sich diese Technik hervorragend für ein Datennahme-System.

6.4.4 Der Direct Memory Access

Mit dem Direkt-Memory-Access (DMA) ist es möglich, einen schnellen Datentransfer zwischen Ein/Ausgabe-Einheiten und der Memory ohne Kontrolle der CPU zu erreichen. Normalerweise kontrolliert die CPU die Datenwege (Busse), so daß jeder Datentransport mit Operationen der CPU verbunden ist, d.h. mit Instruktionen, die interpretiert werden müssen und damit zeitintensiv sind. Der

der Anfangsadresse, der Anzahl der zu transferierenden Worte und der Übertragungsrichtung initialisiert und die Übertragung der Daten beginnt.

Ist sie beendet, bekommt die CPU über das Zurücksetzen des HOLD-Signals die Kontrolle über den Bus zurück.[1].

Zusammenfassend bilden der DMA und die Interrupt-Struktur die hardwaremäßige Basis für einen schnellen Datentransfer zwischen einzelnen Prozessoren, die über das FPSS verknüpft sind.

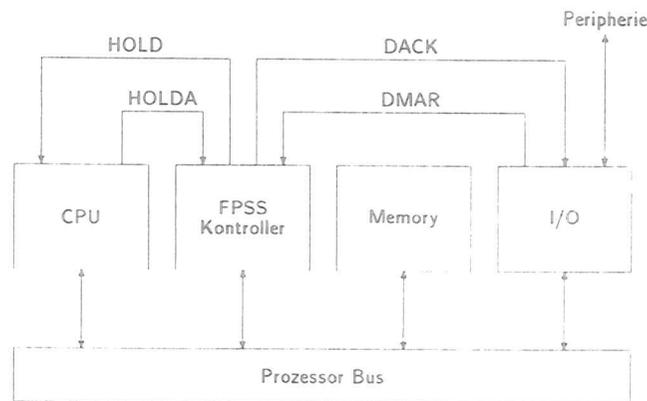


Abbildung 6.8: Der DMA

durch die Software kontrollierte Datentransfer zwischen Ein/Ausgabe-Geräten und der Memory wird beim DMA durch eine spezielle Hardware-Einheit übernommen, dem DMA-Kontroller. Der Aufbau eines DMA geschieht in mehreren Schritten:

Zuerst meldet z.B. ein Eingabegerät über den Ein/Ausgabeprozessor einen DMA an. Dieses Signal gelangt über die DMA-Request-Leitung (DMAR) in den DMA-Kontroller. Dieser fordert seinerseits von der CPU über die "HOLD"-Leitung die Buskontrolle an. Diese wird ihm über die HOLDA-Leitung (HOLD-Acknowledge) gewährleistet. Ist dies geschehen, so wird der Kontroller mit

Kapitel 7

Das Prozessor-Betriebs-System

Für die Anpassung der Benutzerwelt an die Maschinenwelt und die Organisation, Koordination und Steuerung der Betriebsabläufe des Prozessors ist ein Betriebssystem erforderlich. Im Rahmen dieser Netz-Konfiguration muß das System die Belange der Prozessor-Vernetzung berücksichtigen. Zu diesem Zweck wurde ein Real Time-/ Multitasking-System entwickelt [15]. Es basiert auf der Architektur des Prozessors Texas TMS 9900 und ermöglicht über einen FPSS-Kanal die Anbindung an das lokale Datennetz. Die einzelnen Aufgaben des "Micro Computer Executive" (MCX) -Systems stellen sich wie folgt dar:

1. Befehlsmanagement vom HOST-Rechner
2. Task-/Time-Verwaltung
3. Speicherorganisation und -verwaltung
4. Ein-/Ausgabe-Organisation über die Kanäle
 - (a) FPSS
 - (b) CAMAC
 - (c) EXPU (Link zum Zentral-Rechner)

Das auf dem HOST laufende MCX-Monitorprogramm stellt dem Benutzer eine Reihe von Kommandos (siehe Anhang) zur Verfügung, die es ihm erlauben, den Systemzustand zu modifizieren, zu testen oder zu kontrollieren. Wird ein MCX-Kommando an einem der HOST-Terminals eingegeben, so gelangt das Kommandowort über das Netz zum MCX-System, wird interpretiert und wenn möglich ausgeführt. Auftretende Fehler oder auszugebende Meldungen werden über den gleichen Weg am Terminal angezeigt. Weiterhin kann das MCX-System die Ausführung von bis zu 16 Tasken im Real-Time Betrieb verwalten, mit allen Ein-/Ausgabe-Aktivitäten über die drei möglichen I/O-Kanäle FPSS, CAMAC und EXPU.

Die Speicherverwaltung ermöglicht das automatische Umorganisieren nach dem Laden/Löschen einer Task. Die Aufteilung des Speichers ist in Abbildung 7.1 dargestellt. Danach belegt das System ca. 10 kByte und vier festgelegte

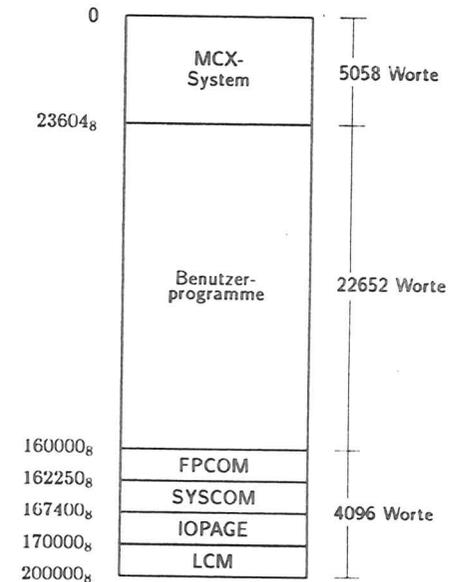


Abbildung 7.1: Die Aufteilung des TMS 9900 - Speichers

Systemsegmente insgesamt ca. 8 kByte, so daß für die Benutzerprogramme noch 46 kByte zur Verfügung stehen.

Die vier Systemsegmente (siehe Abb. 7.1) haben folgende Funktion :

- FPCOM
In diesem Bereich wird die gesamte Kommunikation mit dem FPSS verwaltet, d. h. hier werden die ausgetauschten Protokollpakete abgelegt und der Datentransfer kontrolliert.
- SYSCOM
Dieses Segment ist für alle im Prozessor installierten Tasken zugänglich, wodurch das Anbinden (Link) von Programmbibliotheken und Commonblöcken an die einzelnen Tasken möglich ist (siehe Abschnitt 8.2.2. Seite 65).
- IOPAGE
In diesem Block liegen die Speicheradressen für die vorhandenen I/O--

Kanäle.

• LCM

Der in diesem Adressbereich gespeicherte Programmcode liegt im PROM des Speichers und ist damit permanent. Der "Loader and Control Monitor" (LCM) ist ein Teil des MCX-Systems. Seine wichtigste Funktion ist das Laden des MCX-Betriebssystems. Dieser Vorgang wird durch einen MCX-Monitor-Befehl eingeleitet und führt zu dem sogenannten LOAD-Interrupt. Dieser Interruptvektor steht auf den letzten beiden Speicherstellen im PROM. Der LCM wird gestartet und initialisiert den Prozessor. Nachfolgend wird das MCX-System geladen, der LCM verlassen und das eigentliche System gestartet.

Neben dieser Hauptfunktion stehen im LCM wichtige Systemunterprogramme, sowie nützliche Hilfsprogramme für den Benutzer. Weiterhin wird die Speicheraufteilung ("Memory-Mapping") und ein Teil der sogenannten Systempointer verwaltet.

Neben der Kommando-Ebene des MCX-Systems gibt es noch die Direktiven-Ebene (siehe Abschnitt 7.4, Seite 58). Sie erlaubt dem Benutzer in seinen Programmen immer wiederkehrende Funktionen oder Aufgaben durch das System direkt erledigen zu lassen. Dies sind im allgemeinen Ein-/Ausgabe-Operationen, Warteschleifen oder Fehlermeldungen. Eine komplette Liste aller Direktiven befindet sich im Anhang. Ein Beispiel (im Assembler-Code) soll den Gebrauch verdeutlichen.

```
DATUM BLKW 8.  
.  
.  
DIR .DATE, DATUM  
.  
.
```

In den reservierten acht Worten "DATUM" wird mit dem Aufruf

```
DIR .DATE, DATUM
```

das aktuelle Datum in der Reihenfolge Jahr, Monat, Tag, Stunde, ... bis Ticks pro Sekunde eingetragen.

7.1 Der Aufbau des MCX-Systems

Um den Erfordernissen eines Real Time-/Multitasking-Systems gerecht zu werden, erfolgt die Verwaltung der erteilten Aufträge in Form von verknüpften Listen. So wird jedes Programm, das mittels des MCX-Monitor-Kommandos 'INS taskname' in den Speicher des Prozessors geladen wird, in die sog. "Taskliste" eingetragen (siehe Abb. 7.2). Diese Liste ist in Blöcken zu je 18 Worten aufgeteilt, wobei das erste Wort den Zeiger auf den nächsten Block enthält

und somit die Taskliste der Struktur eines FIFO-Buffers (First In First Out) entspricht. In den verbleibenden Worten des Blocks werden die notwendigen Task-Parameter und Adressen verwaltet, unter anderem der Workspace-Pointer. Jeder Task wird vom MCX-System ein eigener Workspace zugewiesen.

Wird die Task durch das Kommando 'RUN taskname' aktiviert, so wird einzig der Zeiger auf diese Task in der Taskliste in die verkettete Liste der aktivierten Tasken eingetragen und deren Zeiger auf den nächsten freien Eintrag dieser Liste umgesetzt (siehe Abb. 7.3). Stehen mehrere Tasken in der Aktivliste, so werden sie nach Prioritäten geordnet. Der Prioritätenbereich ist in 16 Klassen unterteilt und legt die Bearbeitungsreihenfolge der Tasken fest.

Bei jeder Task mit einer Priorität größer Null, wird der Programmcode so lange abgearbeitet, bis ein sogenannter System-Wait (z.B. die MCX-Direktiven .WS oder .WF) oder das Programmende erreicht wird. Bei einer Priorität gleich Null erfolgt die CPU-Zuteilung für die konkurrierenden Tasken nach dem sogenannten Time-Slicing. Hierbei wird die CPU-Zuteilung vom MCX-System überwacht und kontrolliert (siehe Abschnitt 7.2.2, Seite 53). Ist eine Task aktiv und will z.B. über den EXPU-Kanal Daten von der IBM lesen, so wird diese Anforderung ebenfalls in eine Liste, der I/O-Liste (Input/Output-Queue) eingetragen. Die Task selber kann dann in einen Wartezustand gehen oder aber im Programmfluß fortfahren, je nach der gewählten Direktive. In einer weiteren Liste, der Clock-Queue, werden die Tasken verwaltet, die eine Zeitverzögerung benötigen, da sie in festen Zeitintervallen ihre Aufgabe wiederholen. Mit diesen vier Listen werden die Real Time-/Multitasking-Anforderungen verwaltet und organisiert.

7.2 Die Arbeits-Schleife des MCX-Systems

Das MCX-System selbst durchläuft eine Arbeitsschleife, die sog. "IDLE-Loop" (siehe Abb. 7.4), die die Einträge in den Auftragslisten prüft und bei Vorhandensein von Einträgen diese durch Aufrufen der entsprechenden System-Routinen erledigt. Für die IDLE-Loop lassen sich der inaktive- und aktive Zustand unterscheiden.

7.2.1 Der inaktive IDLE-Loop Zustand

In diesem Zustand existieren keine Einträge in den Arbeitslisten, oder nur solche, die auf externe Signale warten. In diesem Leerlauf erfolgt der Schleifendurchgang wie folgt:

Durch die Idle-Instruktion befindet sich die CPU in Wartestellung. Die interne Prozessoruhr erzeugt alle 200ms einen Interrupt. Dieser sog. Clock-Interrupt führt zu einer Timer-Service-Routine, die die interne Uhr und den Kalender aktualisiert. Als nächstes erfolgt ein Umorganisieren des Speichers, falls es erforderlich ist. Dies ist nur möglich, wenn keine Benutzer-Task aktiv ist, da sich durch das "Memory-Compress" die absoluten Adressen der Programme ändern. Der Compress-Schritt erfolgt durch ein neues Laden ("Re-install") der Tasken vom HOST.

Adresse	Inhalt		
11132	11176	Listen-Pointer	Task 1
11140	00002	Task-Status	
11150	21450	WP-Pointer	
11174	22372	Stack-Pointer	Task 2
11176	11242		
11240	00000		
11242	11306		

Abbildung 7.2: Struktur der permanenten Task-Liste

	Adresse	Inhalt	
Task #1	10766	10770	Listen-Pointer
	10770	10776	nachste Task
	10772	11140	Taskstatus-Pointer
	10774	00005	Priorität
Task #2	10776	11004	
	11004	11017	

Abbildung 7.3: Struktur der Aktiv-Task-Liste

In den nachfolgenden Schritten wird der Status der drei Auftragslisten der Aktiv-Taskliste, der I/O-Queue und der Clock-Queue überprüft. Für jede Liste existiert ein "Auftragszähler", der bei Einträgen in diese Liste einen Wert ungleich Null anzeigt. Damit entfällt ein Durchsuchen der ganzen Liste, da nur der Wert des Zählers geprüft werden muß. Am Ende der Schleife springt das System wieder zu der Idle-Anweisung zurück (siehe Abb. 7.4).

7.2.2 Der aktive IDLE-Loop Zustand

Existieren Einträge in den Auftragslisten, so werden sie bearbeitet. Dies soll an einem Beispiel verdeutlicht werden. In der Aktiv-Taskliste seien die Tasken T_1 und T_2 mit der Priorität Null eingetragen. Beim Auffinden dieser Einträge vollzieht das System eine komplette Programmumschaltung mittels des Context-Switch. Das heißt, aus der Taskliste werden die Adressen des Workspace und Programmcounters sowie das Statusregister z. B. der Task T_1 geladen und durch die "RTWP"-Instruktion die Umschaltung erzwungen. Von diesem Moment an hat die Task T_1 die Kontrolle über die CPU, d. h. ihr Programmcode wird nun abgearbeitet.

Durch das Time-Slicing unterbricht der Clock-Interrupt nach 200 ms die Task T_1 . Es werden das Statusregister und die momentane Programmadresse in den Parameterblock der Task T_1 in der Taskliste gerettet. Der Workspace braucht nicht gerettet zu werden, da jeder Task ein eigener Workspace zugewiesen worden ist. Nach der "Rettung" wird die interne Uhr weitergezählt und der nächste Eintrag in der Aktiv-Taskliste geprüft.

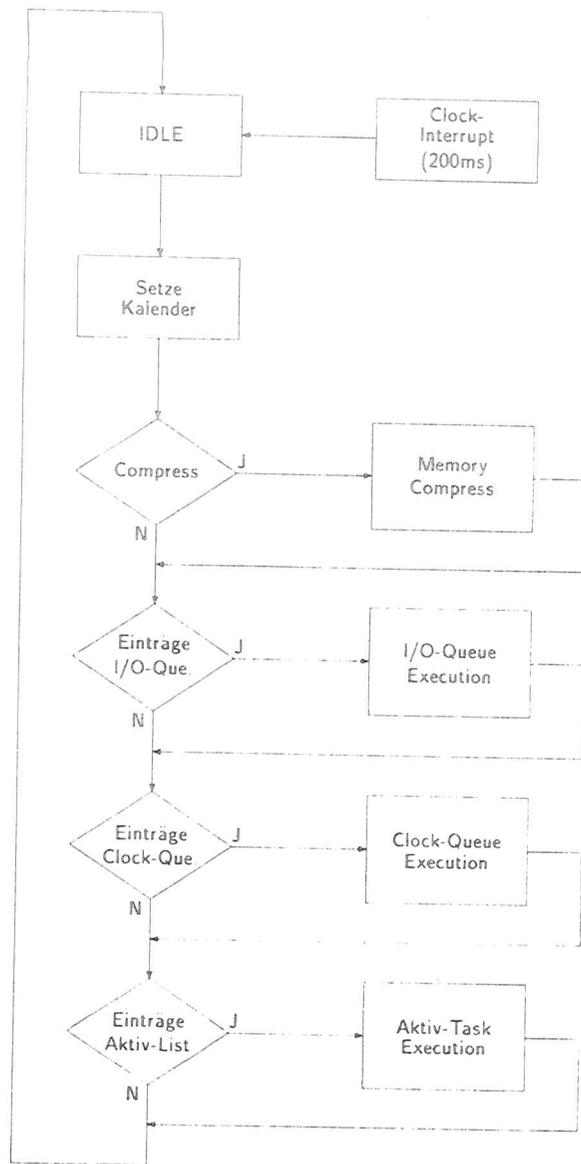


Abbildung 7.4: Die IDLE-Loop des MCX-Systems

In diesem Fall ist es der Eintrag für die Task T_2 . Das System vollzieht ebenfalls eine Programmumschaltung und T_2 übernimmt für die nächsten 200ms die Kontrolle über die CPU. Nach Ablauf dieser Zeit wird wiederum vom Clock-Interrupt die Umschaltung auf das System erzwungen. Da jetzt in der Aktiv-Taskliste das Ende der Einträge erreicht ist, beginnt der Systemzyklus wieder von vorne. Die nun folgende IDLE-Instruktion in der Arbeitsschleife des System wird übersprungen, da sich in der Aktiv-Liste noch Einträge befinden und es damit nutzlos wäre den Prozessor in einen Wartezustand zu bringen.

Das Flußdiagramm in Abbildung 7.5 soll den beschriebenen Systemablauf erläutern. Als Umschaltzeiten zwischen den Betriebszuständen ergeben sich die folgenden gemessenen Werte [14]:

Task A → MCX : 0,65 ms
 Task A → Task B : 1,30 ms

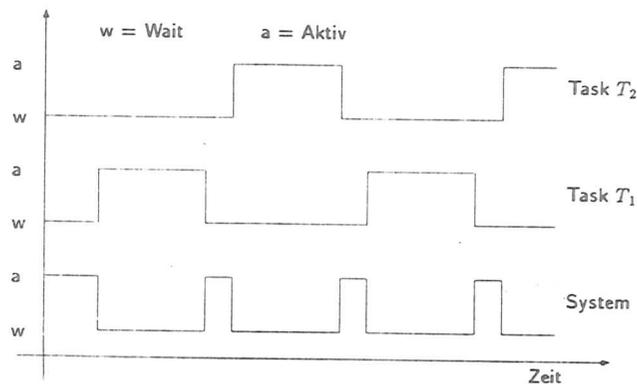
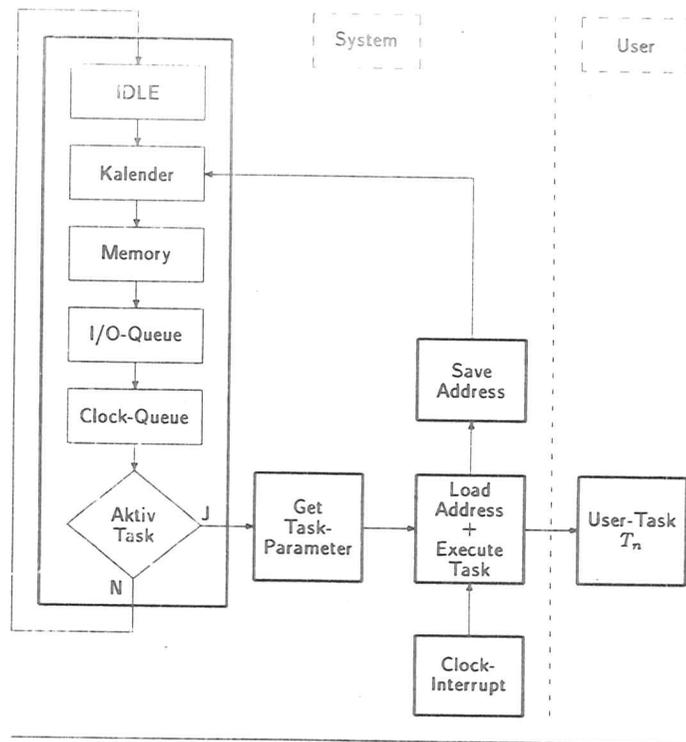


Abbildung 7.5: Der Systemfluß bei Listeneinträgen

7.3 Externe Anforderungen an das MCX-System

Der oben beschriebene zyklische Ablauf der System-Arbeitsschleife muß verlassen werden, wenn externe Anforderungen an das System gestellt werden, so z.B. über den MCX-Monitor auf dem HOST oder von einem anderen Prozessor aus dem Netz. Solch eine Anforderung führt zu einem Hardware-Vektor-Interrupt, d.h. der momentane Prozessorzustand wird über die drei Register (WP, PC, PS) in den Workspace der Interrupt-Routine gesichert und die Interrupt-Service-Routine (ISR) geladen. Durch diese Umschaltung wird eine Anforderung aus dem FPSS-Netz prompt bedient. Die ISR prüft den Interrupt und ruft dann seinerseits die eigentliche Service-Routine auf, die den FPSS-Interrupt bedient. Diese Routine liest mittels DMA einen Datensatz, ein sog. Protokollpaket, ein. Dieses Protokoll enthält neben Kontrollinformationen den Kommandocode der externen Anforderung. Dieser Code wird mit einer Kommandotabelle verglichen. Ist er fehlerfrei, so wird er durch Aufrufen der entsprechenden Routine ausgeführt. Sind alle Schritte erfolgreich abgeschlossen, erfolgt wieder die Umschaltung über den Context-Switch auf den Prozessorzustand vor dem Interrupt.

Im Gegensatz zu diesem "zufälligen" Interrupt als Folge eines MCX-Monitorkommandos, zeigt ein "geplanter" Interrupt als Resultat ein Ein/Ausgabe-Sequenz durch eine Benutzertask komplexere Bearbeitungsstrukturen. Ausgehend von einem Programm, das Daten über den FPSS-Kanal von einem anderen Prozessor lesen will, wird durch eine Systemdirektive dieser Bedarf in die I/O-Queue eingetragen. Das Benutzerprogramm kann danach in einen Wartezustand gehen oder aber in der Programmausführung fortfahren und später auf die Erfüllung der Anforderung warten. Erwartungsgemäß findet das MCX-System bei der Prüfung des I/O-Queuestatus diese Anforderung. Zur Durchführung wird ein Protokollpaket aufgesetzt und über das FPSS-Netz dem anderen Prozessor übermittelt und gleichzeitig im FPCOM vermerkt. Dies erfolgt aber nur für I/O-Anforderungen über den FPSS-Kanal. Dieses Protokollpaket enthält alle Parameter zur eindeutigen Identifikation dieser Anforderung, d.h. die Empfänger/Sender-Prozessornummern und Tasken.

Ist diese Anforderung hinterlegt, fährt das System mit der Bearbeitung weiterer Listeneinträge, wenn vorhanden, fort. Erfolgt während einer dieser Bearbeitungsschritte die Antwort des anderen Prozessors, so wird über das FPSS-Netz ein Hardware-Interrupt initialisiert. Durch die Prioritätsklasse wird er als FPSS-Interrupt identifiziert und die entsprechende Interrupt-Service-Routine wird aktiviert um diese Unterbrechung zu bedienen. Dazu wird die Antwortsequenz gelesen, in der die gleichen Identifikationsparameter stehen wie im Auftragsprotokoll. Mit dieser Information kann das MCX-System der eventuell in einem Wartezustand stehen Benutzertask die Antwort direkt zuweisen. Dies ist möglich, da im FPCOM bei der Anforderung die Adresse des Speicherbereichs hinterlegt worden ist, in den die Daten übertragen werden sollen (siehe Abschnitt 8.3.1, Seite 71). Der nun eingeleitete eigentliche Datentransfer wird über einen DMA abgewickelt. Ist dieser Transfer beendet, springt das System zu dem Zustand zurück, der vor der Unterbrechung existierte.

7.4 Die Unterprogramm-Technik

Zur Unterstützung der Benutzerprogramme existieren im MCX-Betriebssystem diverse Direktiven und im LCM einige Hilfsprogramme. Die Verknüpfung zwischen diesen Dienstprogrammen und einem Benutzerprogramm soll nachfolgend kurz erläutert werden.

Für die Syntax des Direktiven-Aufrufs gilt:

DIR .name,parameterblock

In dem Parameterblock wird dem MCX-System ein Feld mit den entsprechenden Werten übergeben, bzw. bei Beendigung der Direktive dem Benutzerprogramm die angeforderten Daten. Der Vorteil der Direktiven wird deutlich, bei der Benutzung der I/O-Kanäle, da hier die Verwaltung und Kontrolle vom MCX-System übernommen wird. Eine komplette Liste der vorhandenen Direktiven befindet sich im Anhang (Seite 77).

Da beim Aufruf einer Direktive das Benutzerprogramm verlassen wird, muß die Rücksprungadresse gesichert werden. Dies erfolgt durch einen Context-Switch, der vom MCX-System ausgeführt wird. Dadurch bleibt die Programmumgebung des Benutzer unverändert und wird beim Rücksprung aus dem MCX-System wieder aktiviert.

Der Zugriff auf die Hilfsprogramme im LCM erfolgt über den Aufruf

BL @\$name

Die Parameter werden in Registern übergeben. Die Hilfsprogramme selbst sind nützliche Konvertierungsprogramme, so z.B. "Convert Binary to ASCII Dezimal Sign" (CBDSG).

Neben der Anwendung der Dienstprogramme ist die Verwendung eigener Unterprogramme ein wichtiges Hilfsmittel, um komplexe Programme über die Unterprogrammtechnik zu modularisieren. Im Instruktionssatz des TMS 9900 wird diese Technik durch zwei Anweisungen unterstützt [17]:

der 'BL' (branch and link) und 'RT' (return) -Anweisung.

Wird aus dem Hauptprogramm das Unterprogramm UP1 durch die BL-Anweisung aufgerufen, so wird die Rücksprungadresse ins Hauptprogramm in dem Arbeitsregister R11 gesichert. Werden Unterprogramme verschachtelt, zeigt sich der Nachteil der BL-Anweisung. Ein weiterer Aufruf eines zweiten Unterprogramms aus dem Unterprogramm UP1 führt zum Überschreiben der Rücksprungadresse für UP1 ins Hauptprogramm. Der Benutzer kann dies umgehen, indem er die Rücksprungadresse in R11 in ein anderes Register rettet. Bei einer tiefen Unterprogrammverschachtelung führt dies zu einer gewissen Unübersichtlichkeit und zur Einschränkung des Registergebrauchs.

Durch die Einführung eines Stacks wird die Verwaltung der Rücksprungadressen vom Stack übernommen. Der Stack selbst ist ein LIFO-Speicher (Last In First Out), der sicherstellt, dass die letzte Rücksprungadresse immer "oben" steht.

Angelegt wird der Stack bei der Übersetzung (TCA, TCC) der Programme auf dem HOST durch die Cross-Software. Dadurch verfügt jedes Programm, indem die Anweisung "CALL" steht, über einen eigenen Stack. Mit der Anweisung

CALL name

werden die eigenen Benutzer-Unterprogramme aufgerufen. Wird der Stack bei der Ausführung des Programms über diese Anweisungen aktiviert, so wird die Rücksprungadresse im Stack eingetragen und der sogenannte Stackpointer auf diese Eintragung umgesetzt. Damit ist es für den Benutzer unproblematisch, Unterprogramme und Direktiven ineinander zu verschachteln.

Erfolgt der Rücksprung ins aufrufende Programm, wird die "oberste" Rücksprungadresse in den Programmcounter (PC) geladen und der Stackpointer zurückgesetzt.

Im Vergleich zur BL-Anweisung, sind durch die notwendigen Stackoperationen die Ausführungszeiten etwas länger.

Kapitel 8

Das Crystal-Ball-Micro-System

Das im Abschnitt 5 beschriebene Problem der Leerzeiten bei der Datennahme des Crystal-Ball-Online-Systems, bildet die Motivation für das "Crystal-Ball-Micro-System" (CBMS).

Um diese Leerzeiten zu reduzieren, werden die Display-Tasken der Pipeline nur noch selten aktiviert. Zwar haben alle Pipeline-Tasken gegenüber der Datennahme-Task eine geringere Priorität, aber bei jeder Programmumschaltung muß vom Rechnersystem erst der ganze Programmstatus gerettet, und dann der neue Status geladen werden. Diese Umschaltzeiten sind für die Datennahme Leerzeiten und erhöhen damit die Totzeit des Online-Systems.

Die Aufgabe des CBMS ergab sich aus dem Verzicht der kontinuierlichen Display-Darstellungen durch die Pipeline-Tasken. Das Ziel des CBMS war es, diese Kontinuität zu ermöglichen.

Die Bewältigung dieser Aufgabe sollte durch eine "horizontale" Datenverarbeitung in Form eines Multi-Prozessor-Systems erfolgen.

Dazu mußte die Hardware, die in Abschnitt 8.1 beschrieben wird, aus den Komponenten zusammengestellt werden, die in den vorangegangenen Abschnitten erläutert wurden. Die aus der oben gestellten Anforderung entwickelte Software, wird in Abschnitt 8.2 dargestellt.

8.1 Die CBMS-Hardware

Die Hauptkomponenten des CBMS sind zwei PADAC-Microprozessoren und ein DEC PDP 11/23 Computer als HOST. Als Netzwerk dient das FPSS. Zur Darstellung der Detektorprojektionen werden vier PADAC-Video-Module und ein RGB-Monitor (Rot-Grün-Blau) mit 512×256 Bildpunkten benutzt. Da es sich um drei Netzwerkkomponenten handelt (HOST und zwei Prozessoren) werden drei PADAC-Crates und für das Netzwerk ein FPSS-Knoten-Crate benötigt. Als Verbindungen zwischen den PADAC-Crates und dem FPSS-Knoten-Crate werden einfache Koaxialkabel benötigt. Dagegen ist ein breitbandiges Flachkabel als Verbindung zwischen dem HOST-Rechner und dem PADAC-Crate

notwendig. Den kompletten Aufbau des CBMS zeigt die Abbildung 8.1.

8.2 Die CBMS-Software

Die oben beschriebene Aufgabe des CBMS läßt sich in zwei Teile zerlegen, die je einem Prozessor zufallen, der über das FPSS-Netz mit dem anderen Prozessor verbunden ist.

- Prozessor I
 - Lesen der Rohdaten von der IBM-CB-Online-Disc
 - die physikalische Analyse der Ereignisse
 - Monitoring des CBMS
 - Statistik des Datendurchsatzes
- Prozessor II
 - Aufbereitung der Ereignisse zur graphischen Darstellung
 - Erzeugung der verschiedenen Detektor-Projektionen
 - Darstellung der Ereignisse auf einem Farb-TV

Durch diese Trennung der Aufgabe in einen Analyse- und Display-Bereich, ist es möglich, alle gelesenen Ereignisse zu analysieren, aber z.B. nur alle 20 Sekunden ein Ereignis graphisch darzustellen. Die Anforderungsprofile der Prozessoren werden nachfolgend näher erläutert.

8.2.1 Prozessor I

Die Aufteilung der Memory des Prozessor I zeigt Abbildung 8.3. In der Task "PIB" ist als Leitprozeß das Kopieren der Rohdaten über den EXPU-Kanal in den Speicherbereich "RING" und die physikalische Analyse der Ereignisse integriert. Das Kopieren der Rohdaten erfolgt in mehreren Schritten. Wird die Task PIB gestartet über das MCX-Kommando 'RUN PIB', wird zuerst geprüft, ob der Datenbuffer RING existiert, der Commonblock "MICCOM" und die Library "MATLIB", in der einige mathematischen Funktionen definiert sind. Tritt bei dieser Prüfung ein Fehler auf, bricht die Task mit einer Fehlermeldung ab.

Im nächsten Schritt wird der "Link" zur IBM-Großrechenanlage initialisiert und auf der IBM ein Online-Job, ein sogenanntes EXP-Job, gestartet. Über einen festen Befehlssatz können mittels dieses EXP-Jobs Folgefunktionen eingeleitet werden [4]. Eine besteht in der Möglichkeit, ein sogenanntes User-Modul, in diesem Fall das Programm "ONLIN" auf der IBM, zu starten. Dieses Programm liest die Rohdaten von der CB-IBM-Online-Disc und überträgt sie mittels des EXPU-Kanals in den Datenbuffer RING. Dieser Datentransfer erfolgt über einen DMA, d.h. die Task PIB initialisiert mit Hilfe einer MCX-Direktive den Transfer, indem es einen Befehlsblock von 128 Worten für das

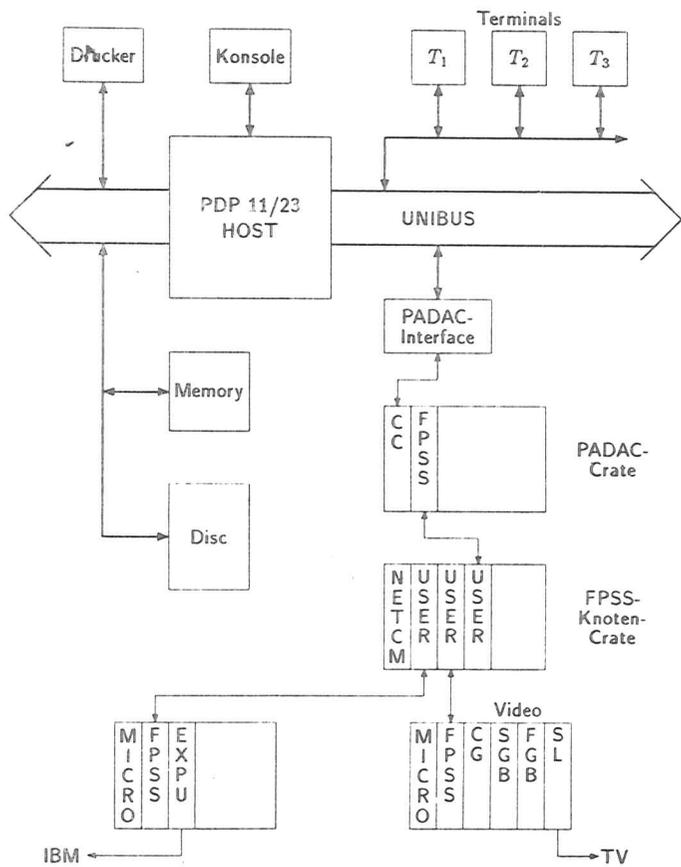


Abbildung 8.1: Der CBMS-Hardware-Aufbau

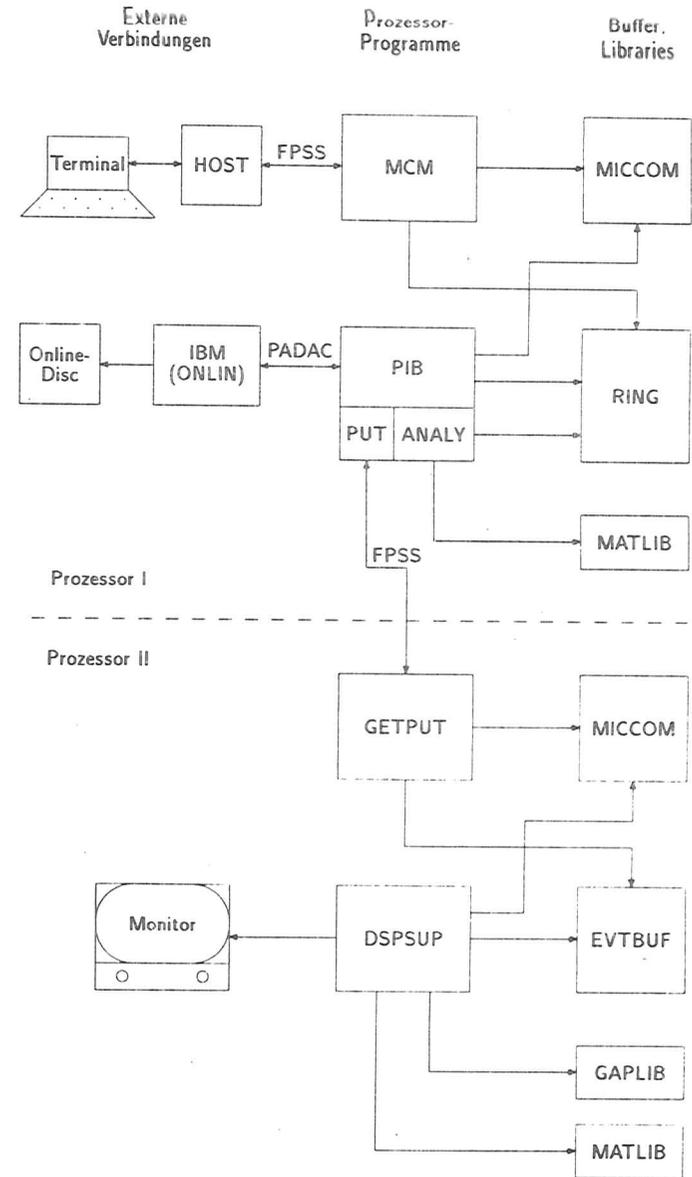


Abbildung 8.2: Die Vernetzung der CBMS-Programme

```

MCX1>PAR
MCXSYS 000000 024234 MAIN TASK      16-01-86 12:53 00023
GEN    024234 133544 MAIN SYS
      024234 017674 SUB (MCM ) 000003 033214 00023
      044130 022056 SUB (PIB ) 000002 050252 00001
      066206 024704 SUB (RING ) 000000 113060 02023
FPCOM 160000 002000 MAIN COM
SYSCOM 162250 005130 MAIN COM
MATLIB 162250 000146 SUB LIBRARY
MICCOM 162416 000240 SUB DYNAMIC
IOPAGE 167400 000400 MAIN COM
LCM    170000 007270 MAIN TASK
MCX1>

```

Abbildung 8.3: Das Memory-Layout des Prozessor I

IBM-User-Modul sendet. Das MCX-System geht danach in eine Warteschleife. Über einen EXP-Befehl leitet das User-Modul seinerseits den Datentransfer von 3072 Worten über den EXPU-Kanal ein. Erst wenn dieser Transfer abgeschlossen ist, gibt das MCX-System die Kontrolle an die Task PIB zurück. Die Transferwortlänge von 3072 Worten ergibt sich aus der Tatsache, daß eine Spur (Track) auf der IBM-Online-Disc 3*3072 Worte lang ist. In einem Block von 3072 Worten befinden sich ca. 3-5 physikalische Ereignisse.

Nach Prüfung der Programmumgebung und der Initialisierung des IBM-Link liest die Task PIB als erstes den Statusblock auf der IBM-Online-Disc. In diesem Block werden die Zeiger (Pointer) auf die Ereignis-Spuren (Tracks), die Runparameter und der Online-Dump verwaltet. Aus der Pointerinformation werden die Startwerte für das User-Modul ONLIN berechnet, d.h. es wird die Anzahl der 3072-Wort-Blöcke bestimmt, die übertragen werden sollen. Diese Werte werden dem User-Modul übermittelt. Jeder dieser Datenblöcke wird über einen Code angefordert und als Antwort wird der Datenblock in den Buffer RING übertragen.

Ist der Transfer fehlerfrei abgeschlossen, wird im nächsten Schritt die physikalische Analyse der Ereignisse, die in diesem Datenblock stehen, eingeleitet. Diese Analyse (siehe Abschnitt 8.4, Seite 73) umfasst die Umrechnung der elektrischen Impulse der Photomultiplier von jedem Kristall in Energiewerte, die Auswertung der Kammerinformation und eine einfache Selektion der Ereignisse in die Kategorien Bhabha, Hadron, Cosmics, Beam Gas und Unknown. Die Resultate dieser Berechnungen werden in einem 2000 Worte umfassenden "Eventblock" gespeichert. Ist die Analyse abgeschlossen, wird geprüft, ob der Prozessor II auf den Transfer eines Eventblocks wartet. Ist dies der Fall, so wird dieser Datenblock durch das Netz mittels eines DMA zum Prozessor II übertragen. Sind noch weitere Ereignisse in dem Rohdatenbuffer, so werden sie nacheinander analysiert und bei Bedarf zum Prozessor II transferiert.

Ist die Analyse abgeschlossen, beginnt der Zyklus von vorne, d.h. es wird wieder ein Rohdatenblock von der IBM angefordert. Diese Schleife wird so lange

durchlaufen, bis die vorher berechnete Anzahl von Rohdatenblöcke übertragen ist. Anschließend wird der Statusblock der IBM-Online-Disc erneut gelesen und geprüft, ob sich neue Ereignisse auf der Disc befinden. Trifft dies zu, werden die neuen Parameter für den nächsten Übertragungszyklus berechnet und der Transfer eingeleitet, wie oben dargestellt. Befinden sich keine neuen Daten auf der Online-Disc, so geht die Task PIB in eine Warteschleife und prüft alle 3 Sekunden den Statusblock. Ändern sich die Pointer für die Rohdatenblöcke, wird der Übertragungszyklus wieder initialisiert. In Abbildung 8.4 ist dieser Verarbeitungsfluß dargestellt.

Das Memory-Layout des Prozessors I zeigt noch die Task "MCM". Dieses "Micro-Computer-Menue" ist ein Serviceprogramm für die Benutzer des CB-Processor-Systems. Es ermöglicht, den Zustand der Prozessoren zu kontrollieren und zu modifizieren. Um dies zu erreichen, wird auf dem Terminal, von dem aus die Task MCM gestartet worden ist, eine Eingabe-Maske angeboten. Die zulässigen Antworten werden aufgezeigt und der Benutzer kann durch seine Wahl die entsprechenden Funktionen einleiten. Abbildung 8.5 zeigt die erste Seite des Menues.

Neben der Zustandsinformation des Datenbuffers RING und des Menues selbst, können das Display-Programm auf dem Prozessor II und die Task PIB modifiziert werden. So ist es möglich, zwischen verschiedenen Projektionen und Darstellungen des Crystal-Balls auf einem Farb-TV zu wählen. Ebenso kann PIB umschalten, um die Rohdaten des letzten und nicht die des derzeitigen Runs zu lesen.

Der Datenbuffer RING ist in vier Bereiche unterteilt. Der größte Bereich ist der Rohdatenblock mit 3072 Worten. In den beiden nächsten Bereichen stehen der Statusblock der IBM-Online-Disc und deren "Booking-Buffer". In diesem Buffer werden die Pointer der letzten 10-20 Runs verwaltet. Dieser Bereich umfasst insgesamt 428 Worte. Die verbleibenden 2000 Worte stehen dem Event-Buffer zur Verfügung, in dem die Resultate der physikalischen Analyse gespeichert werden. Die ersten vier Worte von RING zeigen jeweils auf die Startadresse der entsprechenden Bereiche (siehe Abb. 8.6). Während der Startphase des CBMS werden alle Speicherorte von RING auf Null gesetzt.

8.2.2 Prozessor II

Die Aufgabe der Programme, die im Prozessor II installiert sind, liegt in der Aufbereitung der physikalischen Ereignisse zur Darstellung auf einem Farb-TV. Die Darstellung umfasst drei Detektorprojektionen:

- Die Kristalle in der FLATTY-Projektion
- Die Kammern in der X-Y-Ebene
- Die Kammern in der Z-Ebene

In der FLATTY-Projektion werden in Abhängigkeit für die in einem Kristall deponierte Energie Bildpunkte berechnet, maximal 67, deren Darstellung die Energieverteilung im Crystal-Ball widerspiegelt. In der Kammer-Darstellung

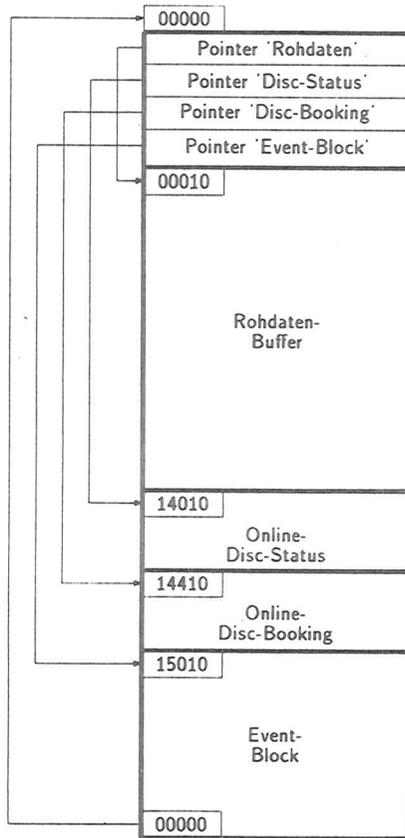


Abbildung 8.6: Der RING-Buffer

MCX2>PAR							
MCXSYS	000000	024234	MAIN	TASK	16-01-86	12:53	00023
GEN	024234	133544	MAIN	SYS			
	024234	0C112C	SUB	(GETPUT)	0C0002	024462	02023
	025354	007124	SUB	(DPSUP)	000003	026242	00002
	03450C	0G7674	SUB	(EVIBUF)	0C000C	04434C	02023
FPCOM	160000	002000	MAIN	COM			
SYSCOM	16225C	0C513C	MAIN	COM			
MATLIB	162250	000146	SUB	LIBRARY			
GAPLIB	16241E	004132	SUB	LIBRARY			
MICCOM	166550	000240	SUB	DYNAMIC			
IOPAGE	167400	000400	MAIN	COM			
LCM	170000	007270	MAIN	TASK			

Abbildung 8.7: Das Memory-Layout des Prozessors II

- ein Full Graphic Buffer (FGB)
- ein Semi Graphic Buffer (SGB)
- ein Clock Generator (CG)
- ein Selector (SL)

Der FGB und SGB dienen als Bildspeicher. Im FGB lassen sich vier Bilder mit je 512 in der Horizontalen und 256 Bildpunkten in der Vertikalen speichern. Jeder Bildpunkt läßt sich einzeln ansteuern, und es kann zwischen sieben Farben gewählt werden, sowie dem Blinkmode. Der SGB ermöglicht es, eigene Muster zu entwickeln, z.B. den ASCII-Zeichensatz zur Textdarstellung. Dieser Buffer zerlegt den TV-Schirm in "Kacheln" zu je 8 x 8 Bildpunkten, so daß horizontal 64 und vertikal 32 Kacheln zur Verfügung stehen. Jede dieser Kacheln kann mit den vorher definierten Mustern belegt werden.

Mit dem Selektor ist es möglich, die gespeicherten Bilder in den Graphik-Buffern auf einem Farb-TV, einem RGB-Monitor (Rot-Grün-Blau), darzustellen. Dies erfolgt durch einen Code, der durch die Software aufgesetzt und in einem speziellen Register abgelegt wird. Die Ansteuerung der Graphik-Buffer geschieht ebenfalls durch Register. Diese Register sind Speicherorte, die in der IOPAGE des MCX-Systems liegen (siehe Abb. 8.3 und 8.7). Wird in diesem Adressbereich ein Code abgelegt, so gelangt er über den internen PADAC-Bus zu dem entsprechendem Modul, das diese Adresse belegt hat. So umfaßt der Registersatz des FGB vier Speicherorte. Die ersten beiden dienen zur Angabe der X/Y-Koordinaten des Monitors und die beiden anderen sind zur Bildsteuerung (Farbe, Bildspeichernummer, etc.) notwendig.

Die entwickelte Graphiksoftware dient als Interface zwischen einem Benutzerprogramm und der Graphikhardware. In ihr sind graphische Basisfunktionen definiert, die eine Vereinfachung der Benutzerprogramme zur Folge haben, da sie nicht den Code für die Graphikregister in der IOPAGE erzeugen müssen. So ist zur Darstellung eines Vektors nur die Angabe der Koordinaten und die Farbe

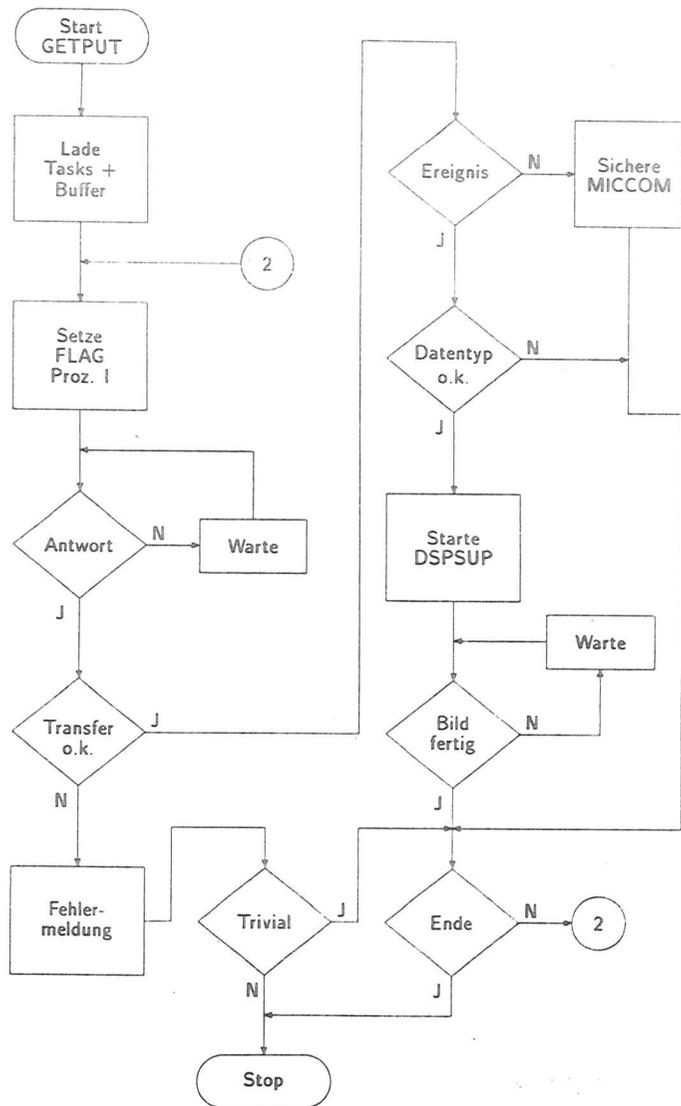


Abbildung 8.8: Der Verarbeitungsfluß auf Prozessor II

notwendig. Über zusätzliche Parameter können die Hintergrundfarbe und der interne Bildspeicher gewählt werden. Der Aufruf erfolgt über

BL @\$name

In Register R1 wird die Startadresse des Parameterblocks übergeben. Eine Liste der Graphikfunktionen befindet sich im Anhang B.

Um zu gewährleisten, daß mehrere Benutzerprogramme auf die Graphikfunktionen zugreifen können, ist die Graphik-Library "GAPLIB" in den SYSCOM der Prozessormemory abgelegt (siehe Abb. 8.7). Dieser Speicherbereich wird vom MCX-System kontrolliert und ist einem Benutzerprogramm über eine MCX-Direktive zugänglich. Diese Direktive verknüpft im Fall der Graphik-Library die Funktionsaufrufe in dem Benutzerprogramm mit dem auszuführenden Code dieser Funktionen, der in der Library steht. Durch diese Organisation ist im Multitasking-Betrieb der Zugriff mehrerer Aufgaben auf Bibliotheksfunktionen, die im SYSCOM stehen, gesichert.

8.3 Synchronisation und Kommunikation

Die Kommunikation des Prozessor-Systems gliedert sich in zwei Ebenen. Zum einen in den Nachrichtenaustausch der MCX-Betriebssysteme untereinander, bzw. MCX-System und HOST. Zum anderen in den Bereich der Aufgaben des CB-Micro-Systems untereinander, d.h. innerhalb des Prozessors und übergreifend zu anderen Prozessoren.

8.3.1 Die externe Prozessor-Kommunikation

Die Ebene der MCX-System-Kommunikation wird in dem reserviertem Memorybereich "FPCOM" (siehe Abb. 8.3 und 8.7) verwaltet. Wird in dem MCX-Monitor auf dem HOST ein Befehl eingegeben, so wird dieser mit weiteren Parametern in den FPCOM auf dem HOST abgelegt. Für den Transfer durch das Netz wird ein Protokollpaket aufgesetzt. Neben der Befehlsinformation und weiteren Kontrollparametern enthält das Paket die Prozessornummer. Der Netzkontroller ist dadurch in der Lage das in das Netz vom HOST abgesetzte Protokollpaket an den richtigen Prozessor weiterzuleiten.

Die Prozessornummerierung ist willkürlich und erfolgt durch sogenannte Adressen-Stecker in den Modulen des Netzkontrollers und der Prozessoren selbst. Das Paket wird in den FPCOM des empfangenden Prozessors übertragen und vom MCX-System interpretiert. Während dieser Zeit befindet sich der sendende Prozeß in einer Zeitschleife. Erfolgt in diesem Zeitraum (ca. 2 Sekunden) keine Antwort, so wird die exklusiv für diese Übertragung aufgebaute Netzverbindung gelöscht und auf dem HOST-Terminal erscheint eine Fehlermeldung.

Sind die Befehle komplexer, wie z.B. 'INS', so wird nach dem Austausch der beiden Protokolle des sendenden und empfangenden Prozesses der Programmcode der zu installierenden Task blockweise, mittels DMA, übertragen.

In dieser impliziten Synchronisation von sequentiellen Prozessen sind zwei Mechanismen für die Prozeß-Kommunikation enthalten. Zum einen die Klasse der 'RENDEVOUS MECHANISM' [6], wonach der sendende Prozeß in einem Wartestatus verbleibt, bis durch eine entsprechende Antwort der Empfänger seine Kommunikationsbereitschaft anzeigt. Nach dieser hergestellten Synchronisation beginnt die eigentliche Übertragung der Daten, z.B. beim Befehl 'INS'.

Die andere Klasse ist das 'REMOTE INVOCATION' [6]. Hierbei geht der sendende Prozeß in einen Wartezustand, bis der Vollzug des in der übertragenden Information enthaltende Befehl durch den Empfänger gemeldet wird. Dies ist der Fall bei den MCX-Befehlen 'RUN' oder 'ABO'.

Um die Vorteile des FPSS-Netzes, schneller Datentransfer und minimales Protokoll-Handling, auch für die Prozessor-Prozessor-Kommunikation nutzbar zu machen, ist es notwendig, die Sender- und Empfänger-Task in dem Prozessor permanent zu definieren. Über einen Buchungs-Vorgang, das vor dem eigentlichen Datentransfer einmal erfolgt, wird die Sender/Empfänger-Definition in einem "Read-Write-Booking-Buffer" des MCX-Systems determiniert. Ist diese Festlegung einmalig vollzogen, so kann die Empfänger-Task eine Anforderung durch die MCX-Direktive 'GET' bei der Sender-Task in Form einer gesetzten Flag hinterlegen.

Die Empfänger-Task geht danach in eine Warteschleife, die die eigene Empfänger-Flag auf eine Antwort der Sender-Task prüft. Da die Sender-Task ihre Flag gesetzt vorfindet, führt sie den Datentransfer durch die Direktive 'PUT' aus. Ist der Transfer beendet, wird auf der Empfängerseite die Flag gesetzt und die Empfänger-Task kann die Warteschleife verlassen.

Im Fall des Datentransfers zwischen dem Prozessor I und II des CBMS setzt die Empfänger-Task GETPUT in dem Prozessor II eine Flag für die Sender-Task PIB auf Prozessor I. Damit wird der Bedarf eines neuen analysierten Ereignisses für die graphische Darstellung angefordert.

8.3.2 Die interne Prozessor-Kommunikation

Die Synchronisation und Kommunikation der Tasken des CB-Micro-Systems erfolgt durch den Common-Block MICCOM, der in dem Prozessor-Speicherbereich SYSCOM liegt. Dieser Bereich ist jeder Task in einem Prozessor durch die MCX-Direktive 'CDC' zugänglich. Existiert dieser Bereich noch nicht, so kann er damit reserviert werden.

Erzeugt wird MICCOM von der MonitorTask MCM, die gleichzeitig notwendige Startwerte- und Adressen hinterlegt. Für jede Task des CBMS ist in dem Common ein Bereich von 16 Worten vorgesehen, in dem die speziellen Parameter dieser Task verwaltet werden. Jede Task organisiert ihren Bereich, so daß sich der aktuelle Zustand des CBMS in dem ganzen Common-Bereich widerspiegelt. Dies gilt für beide Prozessoren (vergl. das Memory-Layout des Prozessors I und II in Abb. 8.3 und 8.7).

Der Informationsfluß zwischen den Tasken ist durch die Transparenz des Common MICCOM gewährleistet, da jede Task die Steuerparameter jeder anderen Task lesen und modifizieren kann. Soll z.B. der Display-Mode auf dem

Prozessor II geändert werden, so wird die Änderung vom Monitor MCM in den Commonbereich der Display-Task DSPSUP eingetragen. Anschließend wird der gesamte Commonblock durch das FPSS-Netz zum Prozessor II übertragen und die Modifikation steht 'DSPSUP' zur Verfügung.

Diese Übertragung ist aus konzeptionellen Gründen einseitig, da der Leitprozeß an die Task PIB auf Prozessor I gebunden ist. Diese "Master"-Funktion des Prozessors I wird noch durch die Monitor-Task MCM unterstützt. Damit zeigt das System die Struktur einer zentralen Kontrolle, die als "Master-Slave"-Prinzips [6] bezeichnet wird. Dieses Prinzip ist effektiv und Störungen der Basis gelangen nicht zur zentralen Kontrollinstanz. Für das CBMS bedeutet dies, das auftretende Fehler in den Tasken des Prozessors II nicht an den Prozessor I übertragen werden.

8.4 Die physikalische Analyse der Daten

Um die Rohdaten sinnvoll darzustellen, ist es notwendig, sie in physikalische Größen umzurechnen. Diese Analyse wird auf dem Prozessor I geleistet und entspricht der Pipeline-Analyse. Dies gilt aber nur mit Einschränkungen, da der PADAC-Prozessor über keinen hardwaremäßigen "Floating-Point-Prozessor" verfügt. Die Entwicklung einer "Floating-Point-Software" hätte den Rahmen dieser Arbeit verlassen.

Das Analyseprogramm ist in der Leittask 'PIB' auf dem Prozessor I als Unterprogramm "ANALY" integriert. Die Resultate der einzelnen Analyseschritte werden in dem Eventblock des Speicherbereichs RING (siehe Abb. 8.6) abgelegt. Dieser Bereich hat eine feste Wortlänge, der aber dynamisch aufgeteilt wird. Die ersten 9 Worte beinhalten Ereignisparameter (Run-/Eventnummer, Strahlenergie, Triggerworte, etc.) zur Identifikation des Ereignisses. In den nächsten 10 Worten werden fünf dynamische Bereiche verwaltet (Pointer und Wortlänge). In diese Abschnitte werden die Resultate der aus fünf Schritten bestehenden Analyse abgelegt.

1. Auswertung der Kammerinformation

In diesem Schritt, der immer durchlaufen wird, werden einige Triggerworte überprüft. So kann es sich z.B. um ein Flasher-Ereignis¹ oder einen End-Run-Record handeln. Liegt ein physikalisches Ereignis vor, so werden aus den Rohdaten der Kammern die "Hits" bestimmt, d.h. die Pulshöhe und die Raumkoordinaten (Φ, Z) berechnet.

2. Berechnung der Energie in den Kristallen

Hier wird den ADC-Werten der Kristalle durch Multiplikation mit Kalibrationskonstanten Energien zugeordnet. Dies erfolgt für den "High"- und "Low"-Channel getrennt, wobei der Low-Channel nicht berücksichtigt wird, wenn der High-Channel gesetzt ist.

¹Hierbei handelt es sich um ein Ereignis, das vom Online-System initialisiert wird, um die Photomultiplier zu überprüfen.

3. Bestimmung von "Connected Regions"

Wenn die Energie der Kristalle, die mindestens an einer Ecke aneinanderstoßen, 10 MeV übersteigt, so werden sie zu einer Connected Region zusammengefasst. Zusätzlich wird für die Ereignis-Selektion (Schritt 5) eine "Energimatrix" berechnet. In ihr steht die Summe aller Kristallenergien, gewichtet mit der Raumrichtung. Weiterhin wird die Asymmetrie der deponierten Energie in den drei Raumbereichen (top-bottom, left-right, front-back) bestimmt.

4. Bestimmung der Teilchenspuren (Tagging)

Es wird vom Z-Vertex = 0 ausgegangen und eine Gerade zu dem energiereichsten Kristall gebildet. Ist die Anzahl der Hits in einem bestimmten Raumbereich um diese Gerade größer als drei, wird diese Spur einem geladenen Teilchen zugeordnet.

5. Klassifikation der Ereignisse

In diesem Schritt wird der Typ der Teilchenreaktion bestimmt. Es wird unterschieden zwischen Hadronen, Bhabhas, Beam-Gas, Cosmic und Unknown. Die Identifizierung erfolgt aus der Wertung der Energimatrix, dem erzeugten Ladungsquerschnitt und bevorzugten Raumwinkeln.

Auftretende Fehler in einem der Schritte führen zum Abbruch der Analyse. Die Pointer des Eventblocks werden neu initialisiert und es wird mit dem nächsten Ereignis fortgefahren. Fehlerursachen liegen zum einen in der Verwaltung des Eventblocks (überschreiten der vorgesehenen Wortlänge) und zum anderen in der physikalischen Struktur des Ereignisses (z.B. fehlende Strahlenergie).

Ist die Analyse erfolgreich abgeschlossen, kann der Eventblock zum Prozessor II transferiert werden. Vor der Analyse des nächsten Ereignisses wird der Block gelöscht und neu initialisiert.

Kapitel 9

Schlußbetrachtung

In der Meßperiode im Frühjahr 1986 wurde das CBMS parallel zu der Online-Pipeline getestet. Während dieser Zeit war es möglich, die aufbereiteten Ereignisse kontinuierlich auf einem Farb-TV darzustellen. Somit konnten während eines Runs bestimmte Ereignisgruppen (z.B. Bhabha's) kontrolliert werden.

Als Vergleichsdatensatz dienen ca. 70 Runs, mit insgesamt 638713 Ereignissen, die von der IBM-CB-Online-Disc gelesen wurden. Davon waren 83.7% physikalische Ereignisse, der Rest sogenannte "Kontroll-Ereignisse", die die Detektorfunktionen prüfen (z.B. die Flasher-Ereignisse).

Von den physikalischen Ereignissen wurden 96.7% analysiert und eine Selektion der Ereignisse durchgeführt. Die verbleibenden 3.3% führten zu Fehlern in der Analyse, so daß sie abgebrochen wurde. Die Ursache dieser Analysefehler waren, soweit es zu erkennen war, fehlerhafte Rohdaten (z.B. das Fehlen der Strahlenergie). In Tabelle 9 sind die Ergebnisse gegenübergestellt.

	CBMS	Pipeline
phy. Ereignisse	83.7%	84.7%
davon analysiert	96.7%	29.0%
Hadron	7.9%	8.1%
Bhabha	16.7%	17.0%
Beam Gas	21.4%	22.1%
Cosmic	24.5%	26.3%
Unknown	25.1%	19.6%

Tabelle 9.1: Der Datendurchsatz CBMS - Pipeline

Auffallend ist der dreimal so hohe Anteil der analysierten Ereignisse in Bezug zur Pipeline. Hier zeigt sich der Vorteil der "horizontalen" Datenverarbeitung durch ein Multi-Prozessorsystem, da in dieser konkreten Anwendung es möglich ist, auf einem Prozessor die Daten zu analysieren und auf dem anderem Prozessor parallel die Ereignisse graphisch darzustellen.

Der Grund für die im Mittel niedrigeren Werte des CBMS bei der Selektion der Ereignisse, liegt in der fehlenden Gleitkomma-Arithmetik. Die Bestimmung der physikalischen Größen wird mit jeder Operation ungenauer und am Ende

fällt das Ereignis in die Kategorie "Unknown". Der Wert dieser Kategorie liegt auch deutlich über dem der Pipeline.

Das System erwies sich als stabil, d.h. auftretende Fehler in der Analyse oder der Aufbereitung zur graphischen Darstellung, führten zu keinem Abbruch des CBMS. Der Zusammenbruch des ganzen Systems, so daß das MCX-Betriebs-System und das CBMS neu geladen werden mußten, ereignete sich im Mittel pro Woche einmal. Diese Fehler ließen sich nicht vollständig analysieren, da teilweise der Programmcode zerstört worden ist.

Als "Verzögerung" zwischen dem Auslesen der Rohdaten und der Darstellung auf dem Farb-TV durch das CBMS, ergab sich ein Mittelwert von sechs Sekunden. Der Grund liegt in der Pufferung der Daten auf dem Online-Disc-Buffer, der ca. alle 5 Sekunden zur IBM übertragen wird.

Für den Transfer zwischen der IBM-Online-Disc und dem CBMS sowie der Analyse der Rohdaten, wurden Zeiten zwischen 35 und 50 ms gemessen, je nach Ereignis-Selektion. Die Darstellung auf dem Farb-TV ist durch das Video-System beschränkt und liegt minimal bei 20 ms. Das ist die Zeit, die notwendig ist, um ein Bild auf dem Farb-TV zu erzeugen.

Anhang A

MCX- Kommandos/Direktiven

Kommando	Beschreibung
BOO image(.ext)	Boot MCX-System-Image (.ext=Version)
SAV	Save System-Image(.ext)
INS task	Install Task
REM task	Remove Task
RUN task	Run Task
ABO task	Abort Task
MEM	Memory-Layout
PAR	Partition-Layout
DEV 06 LP	Device - LUN Assigment
REA LP 07	Reassign LUN - Device
OPE loc	Open Location
DUM A S	Dump von 'A' dresse bis A + 'S'ize
CAM	CAMAC-Test
ABC (ASCII-string)	User Defined MCX-Funktion
FIN	File Input from Connected Computer
FOU	File Output to Connected Computer
DIR	Directory Listing from linked Computer
TES	Test of FPSS Connection

Tabelle A.1: MCX-Kommando-Ebene

Direktive	Beschreibung
.EXIT	Exit Task
.WF	Wait for Event
.WS	Wait for Significant Event
.DS	Declare Significant Event
.DATE	Get Date and Time
.SDEL	Delay (Start)
.EDEL	Delay (End)
.GTP	Get Task Parameter
.INS	Install Task from HOST
.RUN	Run Task
.ABO	Abort Task
.IOH	Common HOST I/O
.IOHW	.IOH with Wait for Completion
.IOHC	Cancel I/O-Packet
.FP	Read/Write Block Data via FPSS
.FPW	.FP with Wait for Completion
.FPB	Book Read/Write Info on Reading Side
.CA	Read/Write CAMAC
.CAW	.CA with Wait for Completion
.CAP	Pack CAMAC Word 'FFFFFFNNNNAAAA'
.PA	Read/Write to PADAC
.PAW	.PA with Wait for Completion
.CDC	Create/Map to Dym. Common or Library
.ERR	Print Error on HOST
.GMA	Get MCX-Function (User Defined)

Tabelle A.2: MCX-Direktiven-Ebene

Anhang B

Graphische Basisfunktionen in der GAPLIB

Funktion	Beschreibung
TVSEL	Set Picture Buffer to TV-Channel
PLOTG	Clear Full-Graphic Buffer
PLOTS	Set Internal Selector to Output Channel
PLOTW	Set Write Mode at Picture Buffer
CLRSB	Clear Semi-Graphic Buffer
LOADCH	Load ASCII Pattern into Semi-Buffer
TVTXT	Draw a String on TV
VECTOR	Draw a Vector
DOT	Draw a Dot
CIRCLE	Draw a Circle
CIRPNT	Draw a Point of Circle
BOX	Draw a Box
KOSYS	Draw a Koordinaten System
RPKOR	Transform R and Phi to X and Y

Abbildungsverzeichnis

2.1	Die Strahllebensdauer	8
2.2	Der Speicherring DORIS II	10
3.1	Die Detektorkomponenten des Crystal-Ball	12
3.2	Der NaJ-Schauerzähler	12
3.3	Die Geometrie des Zentral-Detektors	14
3.4	Der Innen-Detektor	15
3.5	Die Ausdehnung eines Schauers	18
4.1	Die Bhabha-Streuung	21
4.2	Die Leptonen-Erzeugung	22
4.3	Die Quark-Erzeugung	23
4.4	Energieabhängigkeit des R-Wertes	24
4.5	Resonanzkurve des $\Upsilon(1S)$ -Zustands	24
4.6	Die Υ -Erzeugung	25
4.7	Niveauschema der Υ -Familie	26
5.1	Der Datenfluß im Crystal-Ball-Experiment	28
5.2	Das Blockschaltbild der I&H-Module	29
5.3	Die Hardware des Datenaufnahme-Systems	30
5.4	Der Datenfluß im Datenaufnahme-System	33
5.5	Die Pipeline des Online-Systems	34
6.1	Ein sternförmiges Netzwerk mit Zentralknoten	37
6.2	Parallel/Seriell-Konverter	38
6.3	Die Komponenten des FPSS	39
6.4	Der Workspace-Pointer der CPU	42
6.5	Der Context-Switch des TMS 9900	43
6.6	Der Interrupt-Fluß	44
6.7	Der Vektor-Interrupt des TMS 9900	45
6.8	Der DMA	46
7.1	Die Aufteilung des TMS 9900 - Speichers	49
7.2	Struktur der permanenten Task-Liste	52
7.3	Struktur der Aktiv-Task-Liste	53
7.4	Die System-IDLE-LOOP	54
7.5	Der Systemfluß bei Listeneinträgen	56

8.1	Der CBMS-Hardware-Aufbau	62
8.2	Die Vernetzung der CBMS-Programme	63
8.3	Das Memory-Layout des Prozessors I	64
8.4	Verarbeitungsfluß auf Prozessor I	66
8.5	Menue-Maske	67
8.6	Der RING-Buffer	68
8.7	Das Memory-Layout des Prozessors II	69
8.8	Verarbeitungsfluß auf Prozessor II	70

Literatur

- [1] CERN, School of Computing, 1984
- [2] Crystal-Ball, Summer School 1980
- [3] Dodd, J.E., The Ideas of Particle Physics
- [4] EXP-Manual, Interner Bericht, DESY R1-76/4
- [5] DESY, Gruppe F58, private Mitteilungen zum FPSS und PADAC-System
- [6] Giloi, W.K., Rechnerarchitekturen, Springer-Verlag 1981
- [7] Kloiber, T., Diplomarbeit, Universität Erlangen-Nürnberg 1984
- [8] Lohrmann, E., Hochenergiephysik, Stuttgart 1981
- [9] Mayer-Kuckuck, T., Kernphysik, Stuttgart 1979
- [10] Mikroprozessor-Praxis, Sonderausgabe, Okt. 1978
- [11] Mußtopf, G., Mikroprozessor-Systeme, Stuttgart 1982
- [12] Notz, D., Rechnernetze bei DESY, Bericht, Dez. 1985
- [13] Selonke, F. u. Drews, G., TRON, Arbeitspapier, Aug. 1985
- [14] Selonke, F., persönliche Mitteilungen
- [15] Selonke, F., Arbeitspapier zum TCA, TCC und MCX-System, 1986
- [16] Stuckenberg, H.-J., Das Innenleben moderner Mikroprozessoren
Vortrag, Sept. 1985
- [17] Texas Instruments, Handbuch zum TMS 9900, 1979
- [18] Wachs, K., Crystal Ball Plenary-Meeting, Mai 1985
- [19] Wille, K., The Physics of e^+e^- -Storage Rings
Vorlesungen auf der Herbstschule Maria Laach 1983.
Interner Bericht, DESY M-83-16

Danksagung

Ich möchte allen danken, die zum Gelingen dieser Arbeit beigetragen haben.

Mein besonderer Dank gilt Herrn Prof. Dr. J.K. Bienlein für die Ermöglichung dieser Arbeit, sowie Herrn Dr. U. Selonke und Herrn Dr. G. Drews für ihre umfangreiche Unterstützung bei der Überwindung der großen und kleinen Probleme, die sich mir bei dieser Arbeit stellten.

Herrn Prof. Dr. E. Lohrmann danke ich dafür, daß er sich als Gutachter zur Verfügung stellte.

Mein Dank gilt auch den anwesenden Mitglieder der Kollaboration, die durch zahlreiche Diskussionen die kleinen Probleme beseitigten.

Diese Arbeit ist Heike Wetzstein gewidmet.

Im folgenden sind die Mitglieder der Crystal-Ball Kollaboration aufgeführt:

C. Peck, F. Porter, P. Ratoff (California Institute of Technology, Pasadena, USA); I. Brock, A. Engler, B. Kraemer, D. Marlow, D. Prindle, B. Renger, C. Rippich, H. Vogel (Carnegie-Mellon University, Pittsburgh, USA); Z. Jakubowski, G. Nowak (Cracow Institute of Nuclear Physics, Cracow, Poland); H.-W. Bartels, J.K. Bienlein, K. Brockmüller, T. Kloiber, W. Koch, H. Marsiske, H. Meyer, T. Skwarnicki, H.-J. Trost, A. Voigt, K. Wachs (DESY, Hamburg, Germany); D. Antreasyan, J. Irion, K. Strauch, D. Williams (Harvard University, Cambridge, USA); D. Besset, R. Cabenda, M. Cavalli-Sforza, R. Cowan, D. Coyne, C. Newman-Holmes, (Princeton University, Princeton, USA); E. D. Bloom, R. Clare, S. Cooper, J. Gaiser, G. Godfrey, S. Leffler, W. Lockman, S. Lowe, B. Niczyporuk, A. Schwarz, K. Wacker, J. Yeager (SLAC, Stanford University, Stanford); D. Gelpman, R. Hofstadter, I. Kirkbride, R. Lee, A. M. Litke, B. Pollock, J. Tompkins (Stanford University, Department of Physics and HEPL, Stanford, USA); G. Folger, G. Glaser, M. Kobel, B. Lurz, J. Schütte, U. Volland, H. Wegener (Universität Erlangen-Nürnberg, Erlangen, Germany); A. Bizzeti, A. Cartacci, A. Companuggi, G. Conforto, B. Monteleoni, P. G. Pelfer (INFN and University of Firenze, Italy); A. Fridman, F. Heimlich, R. Lekebusch, P. Lezoch, W. Maschman, R. Nernst, D. Sievers, U. Strobusch (Universität Hamburg, I. Institut für Experimentalphysik, Hamburg, Germany); A. König, W. Metzger, J. Schotanus, R. T. Van de Walle, W. Walk (University of Nijmegen, The Netherlands); K.-H. Karch, S. Keh, H. Kilian, K. Königsmann, M. Scheer, P. Schmitt (Universität Würzburg, Germany); D. Aschman (University of Cape Town, South Africa).

Erlärung

Ich versichere, daß ich die vorliegende Arbeit selbständig und ausschließlich unter Benutzung der angegebenen Hilfsmittel und Quellen angefertigt habe.

Hamburg, November 1986

(Hermann Meyer)

