



KfK 4183 B
Dezember 1986

Ein Multiprozessorsystem zur Meßdaten-Aufnahme bei kernphysikalischen Experimenten

M. Neudold
Institut für Kernphysik

Kernforschungszentrum Karlsruhe

KERNFORSCHUNGSZENTRUM KARLSRUHE

Institut für Kernphysik

KfK 4183 B

**Ein Multiprozessorsystem zur Meßdaten-Aufnahme bei
kernphysikalischen Experimenten**

M. NEUDOLD

Kernforschungszentrum Karlsruhe GmbH, Karlsruhe

Als Manuskript vervielfältigt
Für diesen Bericht behalten wir uns alle Rechte vor

Kernforschungszentrum Karlsruhe GmbH
Postfach 3640, 7500 Karlsruhe 1

ISSN 0303-4003

Kurzbeschreibung

In der vorliegenden Arbeit wird zunächst eine vorläufige Version des Datenaufnahmesystems für den Magnetspektrographen LITTLE JOHN am Karlsruher Isochronenzyklotron beschrieben. Die Grundidee dieses ursprünglichen Konzepts und die aufgetretenen Schwierigkeiten beim Test der Hardware und der Betriebssysteme werden kurz geschildert. Aus den dabei gewonnenen Gesichtspunkten wird eine bezüglich Erweiterungsmöglichkeiten und Busstruktur verbesserte Version des Datenaufnahmesystems erarbeitet.

Im zweiten Teil der Arbeit werden Prototypprogramme vorgestellt, die zum Funktionstest der verwendeten ADC-Systeme und der Datenwege dienen.

A MULTIPROCESSOR SYSTEM FOR EXPERIMENTS IN NUCLEAR PHYSICS

Abstract

In the first part of this report, a provisional version of the data acquisition system for the magnetic spectrometer LITTLE JOHN at the Karlsruhe isochronous cyclotron is described. The conceptual ideas of this first version and the problems appearing during the test of its hardware and software components are discussed. Basing on the experience with this system possible extensions and new busstructures are suggested.

In the second part, some prototype programs are described, which have been developed in order to test the ADC-systems and the dataways.

INHALTSVERZEICHNIS

1. Einleitung	1
2. Beschreibung der ersten Version (Setup 1)	6
2.1 Die ADC-Systeme	9
2.1.1 <i>Das LABEN ADC-System</i>	10
2.1.2 <i>Der LE CROY 12-fach ADC</i>	12
2.2 Die Rechnerkombination zur Datenaufnahme und Parametereinstellung	13
2.3 Die Rechnerkombination zur on-line Datenaufbereitung	14
3. Vorschläge zur Verbesserung der ersten Version	16
3.1 Einbau der beiden Frontend-Rechner in ein gemeinsames CAMAC-Crate	18
3.2 Kopplung der beiden Frontend-Rechner über den systemeigenen Q-Bus	19
3.3 Kopplung der beiden CAMAC-Dataways über Dataway-Koppler	21
3.4 Kopplung der beiden CAMAC-Dataways über VME-Bus	22
3.5 Kopplung der beiden Rechnersysteme über eine gemeinsame ETHERNET- Verbindung	26
4. Realisierung der Rechnerkopplung über ETHERNET (Setup 2)	29

5. Inbetriebnahme eines Starburst-Rechners	33
5.1 Aufbau des CAMAC-Systems	33
5.2 Installation des Starburst-Rechners	35
5.3 Installation des Starburst-Betriebssystems	36
6. Programmbeschreibungen	38
6.1 Programm zum Einstellen des LABEN ADC-Systems	40
6.2 Programm zur Übernahme der LABEN-Listdaten	43
6.3 Programm zur Übernahme der LE CROY-Listdaten	50
6.4 Programm zur Übernahme gemischter Listdaten (LABEN u. LE CROY)	55
7. Schlußbemerkungen	58
Anhang A. Quellenverzeichnis	59
Anhang B. Programmlistings	63

1. EINLEITUNG

Am Karlsruher Isochronzyklotron wurde in den vergangenen Jahren ein Magnetspektrograph ('LITTLE JOHN') als Nachweisgerät für die kernphysikalische Grundlagenforschung aufgebaut /1/. Die Detektorsysteme des Spektrographen liefern elektrische Signale, die von standardisierter, modularer Elektronik (NIM- und/oder CAMAC-Norm) geeignet verstärkt und aufbereitet werden, sodaß die interessierenden Meßgrößen durch die entsprechenden Impulshöhen (0-10 V) charakterisiert sind, die von Analog/Digital-Wandlern (ADCs) digitalisiert werden. Aus den digitalen Werten können nach geeigneter Eichung die gesuchten Eigenschaften wie Impuls, Masse, elektrische Ladung des vom Magnetspektrographen nachgewiesenen nuklearen Teilchens berechnet werden.

Unterschiedliche experimentelle Anforderungen und verschiedene Betriebsmoden des Spektrographen haben zur Folge, daß ein einzelnes nachgewiesenes nukleares 'Ereignis' (eine Kernreaktion) durch eine unterschiedliche Anzahl gleichzeitig zu digitalisierender Signale gekennzeichnet sein kann. Im Minimalfall sind dies 6 Signale, sehr häufig jedoch bis zu 16 und für spezielle Experimente 30 bis 32 Signale. Zur eindeutigen Identifizierung eines Ereignisses müssen daher alle notwendigen ADCs von einem gemeinsamen 'Multiparameter-Datenaufnahmesystem' (MPDS) kontrolliert werden. Derartige Kontroll-

systeme sind nur mit maximal 12 ADCs auf dem Markt erhältlich, sind für die geplanten Anwendungen zu wenig flexibel und die Qualität (Linearität, Auflösung) der ankoppelbaren ADCs ist nicht immer ausreichend. Daher war die Entwicklung eines MPDS notwendig. Die allgemeinen Spezifikationen /2/ für dieses System sollten neben der speziellen Verwendung in Zusammenhang mit dem Magnetspektrographen auch einen möglichst flexiblen Einsatz bei anderen Experimenten sicherstellen.

Daneben mußten aus Platzmangel, finanziellen Gründen sowie Personalmangel folgende Randbedingungen eingehalten werden:

1. ADC- und Kontrollsystem einerseits und Experimentrechner mit Peripherie andererseits müssen in getrennten Räumen untergebracht werden, die ca. 100 m Kabellänge voneinander entfernt sind.
2. Bereits vorhandene ADCs der Firma LABEN mit herausragenden Eigenschaften (impulshöhenunabhängige Konversionszeit, hohe Auflösung und Linearität, hohe Zählratenverarbeitung) sollten weitgehend in das System integriert werden.

3. Eine Minimalversion sollte möglichst schon frühzeitig für den Testbetrieb des Spektrographen einsatzbereit sein.

4. Der Weiterausbau mit mehr als 16 ADCs sollte aufbauend auf den Erfahrungen mit der kleineren Version stufenweise erfolgen, sobald seitens des Experiments die Notwendigkeit dafür besteht. Wenn möglich sollte dieser stufenweise Ausbau ohne grundsätzliche Umstellung der ursprünglich gewählten Konzepte erfolgen.

Für den Aufbau des neuen Datenaufnahmesystems wurden mehrere Rechner der PDP 11-Serie (DEC) angeschafft, welche die weiteste Verbreitung in kernphysikalischen Labors haben. Damit ist eine problemlose Übernahme von Software aus anderen Labors gewährleistet bzw. Gastgruppen in Karlsruhe können eigene Programme auf dem aufzubauenden MPDS verwenden. Im Frühjahr 1986 wurden die neuen Rechner an das gerade installierte lokale Rechnernetzwerk am Zyklotron /3/ (LAN) angeschlossen. Dadurch ist der Zugriff auf die Experimentrechner von verschiedenen, auch räumlich weit entfernt liegenden Terminals im Multiuser-Betrieb möglich, was vor allem bei der Software-Entwicklung und der nachträglichen Datenauswertung vorteilhaft ist.

Das eigentliche Datenaufnahmesystem, eine Sonderanfertigung der Firma LABEN, hat die Aufgabe, die vom Detektorsystem (6 - 32 kernphysikalische Detektoren) angelieferten Analogwerte ('Parameter') zu digitalisieren und, nach mehreren unabhängigen Koinzidenzbedingungen geordnet, zu registrieren. Dabei besteht häufig die Forderung, daß ein Teil der anfallenden Analogdaten mit bestmöglicher Auflösung und höchster Linearität konvertiert werden soll, während für andere Parameter wesentlich geringere Anforderungen genügen. Um diesen verschiedenen Anforderungen zu entsprechen, liegt die Verwendung unterschiedlicher ADCs schon aus kommerziellen Gründen nahe. Schließlich sollen mehrere voneinander unabhängige Rechner auf die konvertierten Daten on-line zugreifen können, etwa um Datenspeicherung und on-line Berechnung ein- und zweidimensionaler Spektren getrennt durchzuführen. Unter diesen Gesichtspunkten war eine erste Version des Datenaufnahmesystems /4/ mit maximal 16 ADCs konzipiert worden. Noch während der Installationsphase dieser ersten Version wurde im Rahmen dieser Arbeit eine hinsichtlich der Datenübertragungswege verbesserte Version entwickelt und realisiert.

Damit wurde die Grundlage für einen Aufbau gelegt, der allen derzeitigen Anforderungen entspricht und erweiterungsfähig ist.

VORHALLE

MESSRAUM 1

LABEN - ADC - System

ER 1 (Datenaufnahme und Steuerung)

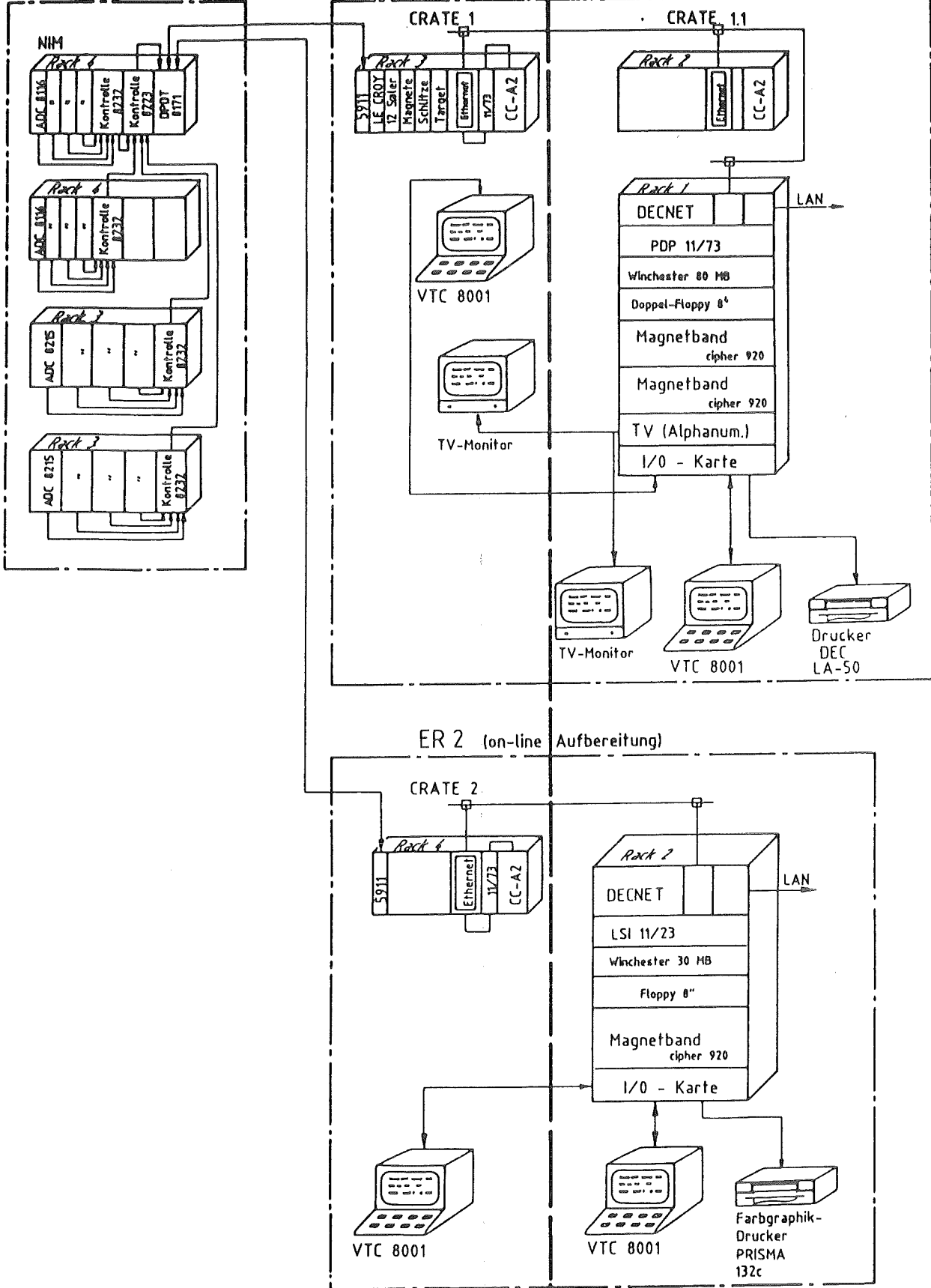


Abb.1: Gesamtübersicht Setup 1 (Erklärung in den Abschnitten 2.1-2.3)

2. BESCHREIBUNG DER ERSTEN VERSION (SETUP 1)

Die ursprüngliche Version des Multiparameter-Datenaufnahmesystems am Magnetspektrographen LITTLE JOHN (Abb.1) baut wesentlich auf CAMAC /5/ als Schnittstelle zwischen Rechner und Experiment auf. Es gliedert sich in:

1. *Die ADC-Systeme* (16 LABEN-ADCs oder ein LE CROY 12-fach ADC);
2. *Rechnerkombination 1*, zur Datenaufnahme sowie zur Einstellung von Magneterregungen und Stellmotoren am Magnetspektrographen;
3. *Rechnerkombination 2*, zur on-line Datenaufbereitung.

Die Aufspaltung in *zwei* weitgehend unabhängige Rechnersysteme ergab sich aus schlechten Erfahrungen der Karlsruher Gruppe mit früheren Generationen einfacherer MPDS, bei denen die Datenaufnahme und Speicherung sowie on-line Datenauswertung über *ein* Rechnersystem häufig zu Problemen führte und entsprechend gute Erfahrungen in anderen Labors (z.B. KVI Groningen), wo der Betrieb zweier paralleler Rechner zu größter Betriebssicherheit führte. Ein

weiterer entscheidender Vorteil eines solchen Aufbaus zeigt sich beim Ausfall eines Rechners, der ohne Hardware-Änderungen abgefangen werden kann.¹

Die beiden Rechnerkombinationen 1 und 2 sind ähnlich aufgebaut. Die Kombination 1 besteht aus einem Frontend-Rechner 1 (PDP 11/73 als CAMAC-Einschub, im folgenden auch gemäß der Firmenbezeichnung 'Starburst' genannt) und einem Host-Rechner ER 1 (PDP 11/73 mit Peripherie), die Kombination 2 aus einem weiteren Frontend-Rechner 2 (ebenfalls PDP 11/73 'Starburst' als CAMAC-Einschub) und einem Host-Rechner ER 2 (LSI 11/23 mit Peripherie).

Die aus der Meßelektronik kommenden Analogdaten werden von den verschiedenen ADCs digitalisiert und gelangen als sogenannte 'Listdaten' (d.h. Folge von Konversionswerten) über die CAMAC-Dataways zu den Frontend-Rechnern, welche die Listdaten sortieren, puffern und teilweise vorverarbeiten. Von den

¹ Es sei an dieser Stelle darauf hingewiesen, daß die genannten Erfahrungen und entsprechenden Konsequenzen bei der Spezifikation neuer Systeme aus Zeiten stammen, in denen keine betriebssicheren, rechner-typ-unabhängigen Rechnerkopplungen im Sinne des LAN-Pilotprojekts am Zyklotron existierten. Die volle Nutzbarkeit eines betriebssicheren LAN-Systems dürfte heute zur Revision einiger älterer Spezifikationen führen.

Frontend-Rechnern werden die vorverarbeiteten Listdaten dann blockweise über separate ETHERNET-Verbindungen, getrennt für die Rechnersysteme 1 und 2, zu den entsprechenden Host-Rechnern übertragen, die dann die Datenarchivierung auf Massenspeicher bzw. die Datenaufbereitung durchführen.

Die drei genannten Funktionsgruppen werden in den folgenden Abschnitten ausführlicher beschrieben.

2.1 DIE ADC-SYSTEME

Aufgrund der in der Einleitung genannten Randbedingungen für den Aufbau des MPDS basiert das System zunächst auf ADCs der Firma LABEN, für die ein spezielles Kontrollsystem für bis zu 16 ADCs von der Firma LABEN entwickelt und gebaut wurde. Parallel dazu wurde ein 12-fach ADC-Modul der Firma LE CROY verwendet, das als Erweiterung des Gesamtsystems in Betracht gezogen wurde, vorerst aber nur allein Verwendung fand.

Sämtliche zur Anwendung kommenden ADCs erfüllen die in der Kernphysik benötigten Spezifikationen:

- * *Pulstrigger-Mode*: Konversion auf einen bestimmten aus dem Experiment kommenden oder vom ADC selbst erzeugten Gate-Impuls hin (letzteres nur bei LABEN-ADCs der Fall);
- * *Linearität*: besser 10^{-4} ;
- * *Einstellmöglichkeit von analogen oder digitalen Schwellen*, ab oder bis zu denen eine Konvertierung erfolgt;
- * *Automatisches Hinzufügen eines Pedestals*: Dies ist ein Wert, den der ADC automatisch zu den Analogwerten hinzuaddiert, um auch für Analogwerte = Null die Funktion des ADC zu bestätigen (Ausfallkontrolle).

2.1.1 Das LABEN ADC-System

Die Module des LABEN ADC-Systems sind zur Spannungsversorgung in NIM-Über-
rahmen untergebracht. Von den maximal 16 anzuschließenden ADCs werden immer
vier von einer Kontrolleinheit (Modell 8232) zusammengefaßt. Die vier not-
wendigen Kontrolleinheiten sind wiederum mit einer übergeordneten Kontroll-
einheit (Modell 8233) verbunden ('Baumstruktur'). Von hier gelangen die
ADC-Daten über ein Splitting-Interface (Modell 8171) und zwei CAMAC-Inter-
facemodule (Modell 5911) auf die beiden CAMAC-Dataways 1 und 2, wo sie von
den Frontend-Rechnern abgeholt werden können. Als ADCs stehen alternativ die
Modelle 8116 und 8215 der Firma LABEN mit einer Auflösung von 13 Bit (ent-
spricht 8K) zur Verfügung. Das System erlaubt, die ADCs beliebig zu grup-
pieren, d.h. Einzel-ADCs bzw. Koinzidenzgruppen aus zwei und mehr ADCs zu
definieren, welche nur dann konvertieren, wenn an allen zu einer Koinzi-
denzgruppe gehörenden ADCs gleichzeitig (im Bereich von ca. 0-32 μ s) ein
Analogwert anliegt. Dabei darf jeder ADC auch in verschiedenen Koinzidenz-
kombinationen vorkommen. Die Zuordnung von ADCs zu einer Koinzidenzgruppe
erfolgt entweder manuell an den Kontrolleinheiten über DIL-Schalter oder per
Software über CAMAC-Befehle (vom Typ 'CNAF', d.h. Crate-Nr. C, Modul-Nr. N,
Subadresse A, Funktions-Code F). Die Kontrolleinheiten fügen den Daten der

einzelnen ADCs noch einige Kennungsbits hinzu, um die Herkunft der einzelnen Daten (ADC-Kennung) bzw. die Zugehörigkeit zu einer bestimmten Koinzidenzgruppe (Koinzidenzgruppenkennung) eindeutig zu kennzeichnen.

Das System kennt folgende Betriebsmoden:

1. *List-Mode*: In dieser Betriebsart werden die anfallend Meßdaten in einem in der Kontrolleinheit 8233 befindlichen FIFO-Speicher (512 Worte a 24 Bit) zwischengespeichert. Ist dieser Speicher gefüllt, erzeugt die Kontrolleinheit einen CAMAC-Interrupt (LAM \equiv Look At Me), woraufhin die angefallenen Daten vom angeschlossenen Frontend-Rechner in einen oder mehrere Wechselpuffer gelesen werden;
2. *Inkrement-Mode*: In dieser Betriebsart werden in der Kontrolleinheit (Modell 8233) einparametrische 8K-Spektren akkumuliert. Das bedeutet, es wird gezählt, wie oft die einzelnen ADC-Werte auftreten;
3. *List- und Inkrement-Mode*: Diese Betriebsart stellt eine Kombination der beiden vorherigen Betriebsmoden dar.

Der Vorteil der Spektrenakkumulation per Hardware in den Betriebsarten 2 und 3 besteht darin, daß sie weit schneller ist als die Akkumulation per Software, sodaß sich dadurch die Totzeit der on-line Datenaufbereitung verringert. Genauere Angaben über die Einstellungen und die Ausleseverfahren in den verschiedenen Betriebsarten sind den Handbüchern zum System /6/ zu entnehmen.

2.1.2 Der LE CROY 12-fach ADC

Bei dem verwendeten 12-fach ADC der Firma LE CROY /7/ handelt es sich um ein einfach breites CAMAC-Modul, das 12 ADCs enthält, die über ein gemeinsames Gate-Signal getriggert werden. Das LE CROY-Modul repräsentiert also eine eigene Koinzidenzgruppe. Die Auflösung beträgt jeweils 11 Bit (entspricht 2K). Nach der Konversion gibt das Modul einen CAMAC-Interrupt an den zugehörigen Frontend-Rechner (hier Starburst 11/73 im Crate 1). Die Daten der 12 A/D-Wandler können dann in beliebiger Reihenfolge über CAMAC gelesen werden. Nach dem Lesen der Daten löscht der dem Crate zugeordnete Rechner mittels CAMAC-Befehl alle 12 ADCs und versetzt das Modul in einen von zwei programmierbaren Zuständen (konversionsbereit oder gesperrt).

2.2 DIE RECHNERKOMBINATION ZUR DATENAUFNAHME UND PARAMETEREINSTELLUNG

Die Rechnerkombination 1 dient sowohl zur Steuerung verschiedener experimenteller Parameter des Magnetspektrographen (Magneterregungen, Targetpositionen, Lafettenwinkel, Schlitzöffnungen usw.) vom Rechner aus, als auch zur Aufnahme und Archivierung (Magnetband) der Meßdaten und Versuchsparameter. Im CAMAC-Crate 1 sind die I/O-Module zur Einstellung der experimentellen Parameter, das CAMAC-Interface des LABEN ADC-Systems, das LE CROY ADC-Modul sowie der Frontend-Rechner 1 und ein Minibackplane mit Q-Bus (DEC-Rechnerbus) Rückwandverdrahtung untergebracht. Der Datenfluß verläuft dabei von den ADCs bzw. den I/O-Einstellmodulen über den Frontend-Rechner 1, der eine Vorverarbeitung der ADC-Daten durchführt. Diese Vorverarbeitung besteht aus dem Sortieren der Daten nach ihrer Koinzidenzgruppenzugehörigkeit, dem Wegstreichen der nach dem Sortieren überflüssigen Kennungsbits und dem Zusammenstellen größerer Datenpakete. Nach Erreichen einer bestimmten Pufferfüllung werden die so entstandenen Datenpakete im Blocktransfer über ETHERNET zum Hostrechner (ER 1) übertragen. Die Kopplung zwischen Starburst und ETHERNET geschieht über eine im Minibackplane untergebrachte ETHERNET-Karte (DEC, DEQNA-Karte). Der Q-Bus im Minibackplane und der Q-Bus des Starburst

sind über zwei 50-polige Flachbandleitungen miteinander verbunden. Der Hostrechner ER 1 schreibt die empfangenen Meßdatenblöcke record-weise auf Magnetband (CIPHER 920). In einem Vor- und einem Endrecord werden die (für die Dauer einer Messung konstanten) Experimentparameter festgehalten. Während der Einstellphase eines Experiments werden die Sollwerte für die Experimentparameter in umgekehrter Richtung ebenfalls über die ETHERNET-Verbindung zum Frontend-Rechner und von da aus über den CAMAC-Dataway 1 zu den I/O-Modulen gesendet.

2.3 DIE RECHNERKOMBINATION ZUR ON-LINE DATENAUFBEREITUNG

Die Rechnerkombination 2 des Datenaufnahmesystems dient zur on-line Datenauswertung, d.h. zur Erzeugung von Spektren schon während des Experiments aus den live anfallenden Meßdaten, sowie als Reservesystem zur Erhöhung der Betriebszuverlässigkeit. Die Listdaten des LABEN-Systems gelangen über das zweite CAMAC-Interfacemodul (5911) auf den CAMAC-Dataway des CAMAC-Crates 2. In diesem Crate befindet sich, analog zum Crate 1, der Frontend-Rechner 2 sowie eine Ankopplung an die ETHERNET-Verbindung 2. Der Frontend-Rechner führt hier eine Datenvorsortierung durch. Die vorsortierten Daten gelangen

dann blockweise über die ETHERNET-Verbindung 2 zum Host-Rechner 2 (LSI 11/23). Im Host-Rechner wird dann die Datenaufbereitung (Spektrenbildung) vorgenommen. Die Spektren werden bei Bedarf auf Farbgraphik-Terminals (VTC 8001) bzw. einen Farbgraphik-Drucker ausgegeben oder zur längerfristigen Archivierung abgespeichert (Diskette, Magnetband oder Festplatte). Außer 'projizierten' ein- und zweidimensionalen Spektren, die durch Projektion der mehrparametrischen Koinzidenzspektren auf eine Koordinatenachse oder eine Koordinatenebene erhalten werden, lassen sich die Listdaten auch per Software einem mathematischen Algorithmus unterwerfen (z.B. linearisieren, limitieren, kombinieren) und anschließend zu ein- oder zweidimensionalen Spektren akkumulieren.

3. VORSCHLÄGE ZUR VERBESSERUNG DER ERSTEN VERSION

Obwohl die Implementierung des LE CROY 12-fach ADCs bei alleiniger Verwendung im MPDS sich als problemlos erwies, ist eine Erweiterung des auf 16 ADCs beschränkten LABEN-Systems etwa durch Hinzunahme von LE CROY-ADCs nicht ohne Einschränkung möglich. Denn die Daten des LE CROY ADCs stehen nur dem zum gleichen CAMAC-Crate gehörendem Frontend-Rechner zur Verfügung, so daß eine on-line Auswertung von dessen Listdaten nicht möglich wäre. Dies bedeutet eine untragbare Einschränkung. Es muß also nach einer Möglichkeit gesucht werden, auch die Daten zusätzlicher ADCs den beiden Rechnerkombinationen 1 und 2 zugänglich zu machen. Zur Lösung des Problems kommen folgende Möglichkeiten in Betracht:

- 1) *Einbau der beiden Frontend-Rechner in ein gemeinsames CAMAC-Crate;*
- 2) *Kopplung der beiden Frontend-Rechner über Q-Bus;*
- 3) *Kopplung der beiden CAMAC-Dataways über Dataway-Koppler;*
- 4) *Kopplung der beiden CAMAC-Dataways über VME-Bus;*
- 5) *Kopplung der beiden ETHERNET-Verbindungen.*

Die Abbildung 2 auf Seite 17 zeigt die Einheiten, welche bei allen Versionen unverändert bleiben.

VORHALLE

MESSRAUM 1

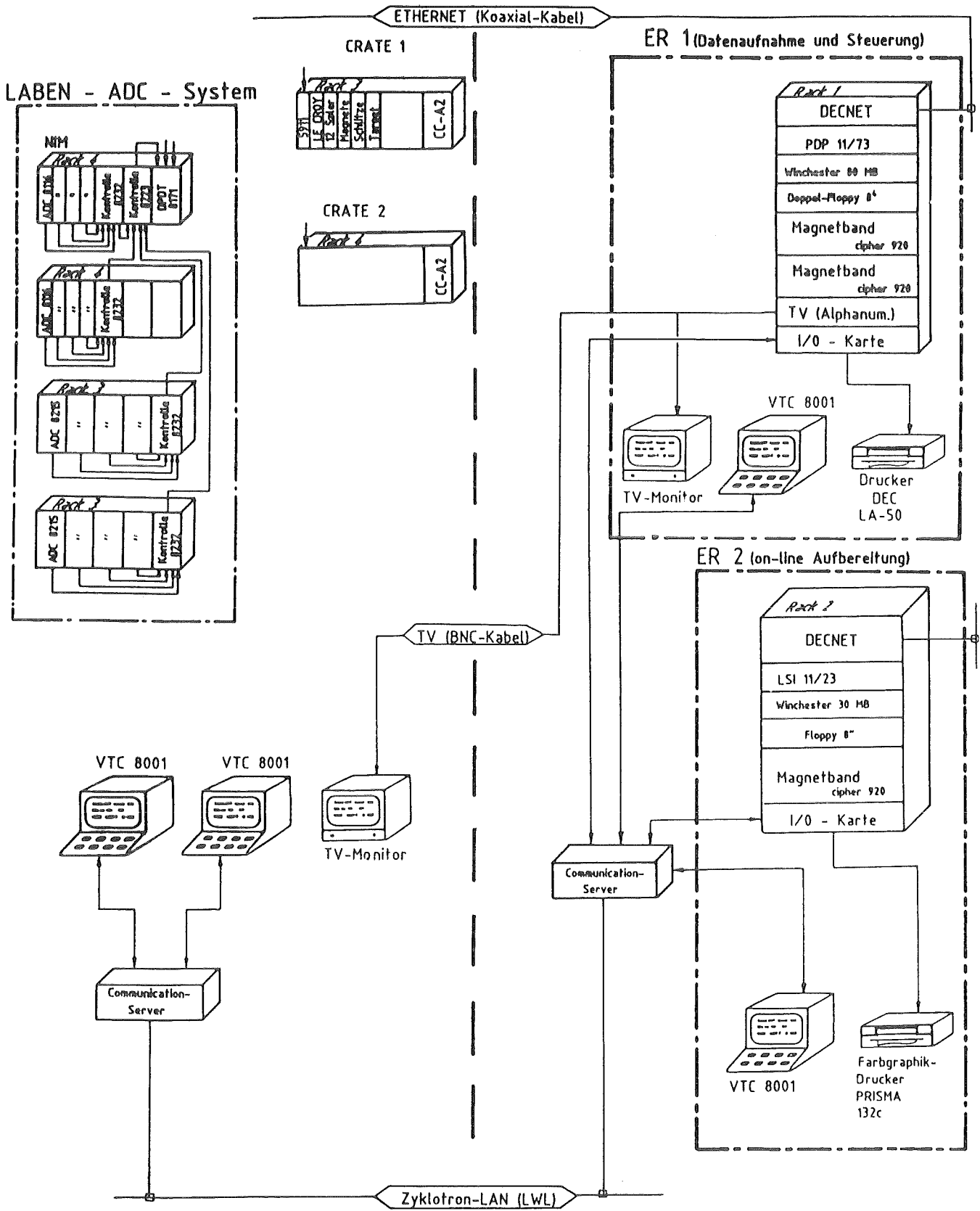


Abb.2: Gesamtübersicht der in allen Versionen unveränderten Einheiten

3.1 EINBAU DER BEIDEN FRONTEND-RECHNER IN EIN GEMEINSAMES

CAMAC-CRATE

Eine einfache Art der Kopplung der beiden Rechnerkombinationen 1 und 2 wäre durch die Unterbringung der beiden Starburst-Rechner im gleichen CAMAC-Crate zu realisieren (Abb.3 Seite 20). In diesem Fall hätten beide Frontend-Rechner die Möglichkeit, auf die Meßdaten zuzugreifen. Die beiden Starburst-Rechner müßten dazu nur untereinander nach Art einer 'Daisy-Chain' verbunden werden, wobei der an erster Stelle in der Kette befindliche Rechner die höchste Priorität hat. Dieser erste der beiden Rechner müßte der Frontend-Rechner 1 sein, da das 'Weiterreichen' der Meßdaten zur Datenarchivierung (ER 1) eine höhere Priorität hat als die on-line Datenauswertung. Diese Variante erfordert allerdings, daß auch alle Nicht-LABEN ADCs im gleichen CAMAC-Crate untergebracht werden müßten, was aus Platzgründen nicht möglich ist.

3.2 KOPPLUNG DER BEIDEN FRONTEND-RECHNER ÜBER DEN SYSTEMEIGENEN Q-BUS

Eine andere naheliegende Lösungsmöglichkeit wäre die Kopplung der beiden Frontend-Rechner (PDP 11/73) über den PDP-Systembus /8/ (Q-Bus), der den Betrieb mehrerer Rechner (Multiprozessorbetrieb) grundsätzlich erlaubt (Abb.4 Seite 20). Die Daten würden in diesem Fall vom Frontend-Rechner an dem CAMAC-Dataway gelesen, der die zusätzlichen Nicht-LABEN ADCs enthält, und dann dem anderen Starburst über den gemeinsamen bidirektionalen Q-Bus zur Verfügung gestellt. Die Kopplung zweier PDP 11 Rechner über Q-Bus ist jedoch unter dem vorhandenen Betriebssystem (RSX-11M) wegen fehlender Unterstützung der Bus-Arbitrierung für mehrere Prozessoren nicht möglich, sondern setzt ein spezielles Multiprozessor-Betriebssystem voraus. Außerdem wäre zur Realisierung der Q-Buskopplung ein normales PDP-Chassis mit vollständiger Q-Busverdrahtung Voraussetzung. Neben der zusätzlichen Kosten (ca. 4000 DM) wurde auch wegen des Platzbedarfs dieses in der äußerst engen 'Vorhalle' zu installierenden Chassis von dieser Möglichkeit Abstand genommen.

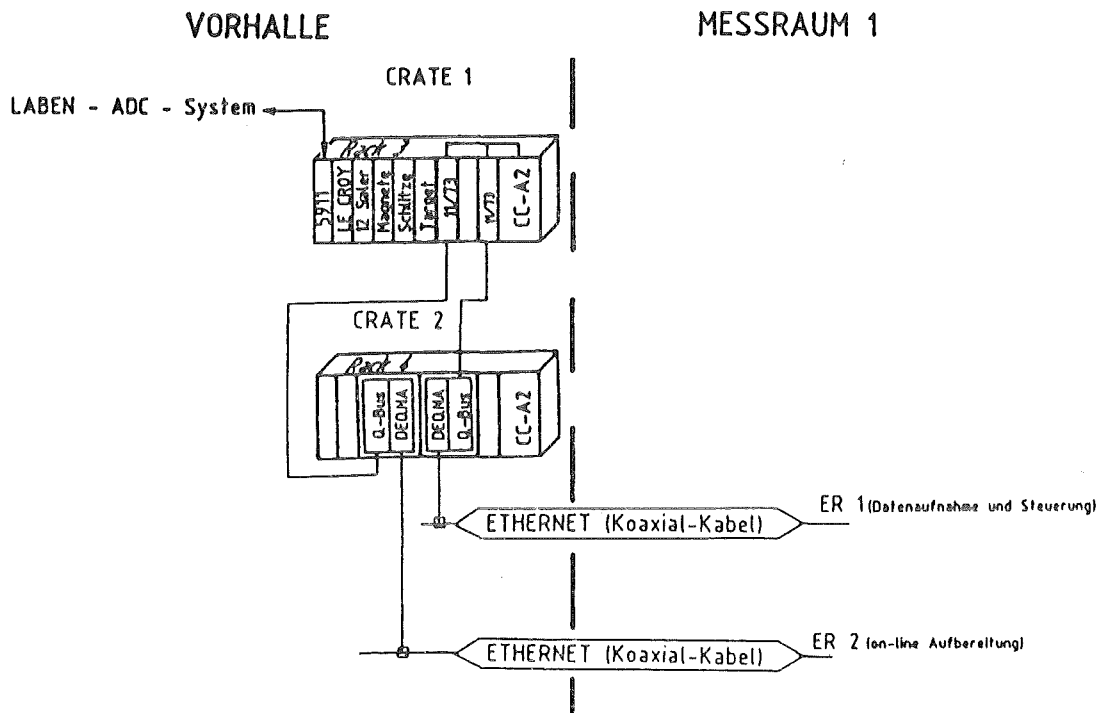


Abb.3: Einbau der beiden Frontend-Rechner in ein gemeinsames CAMAC-Crate

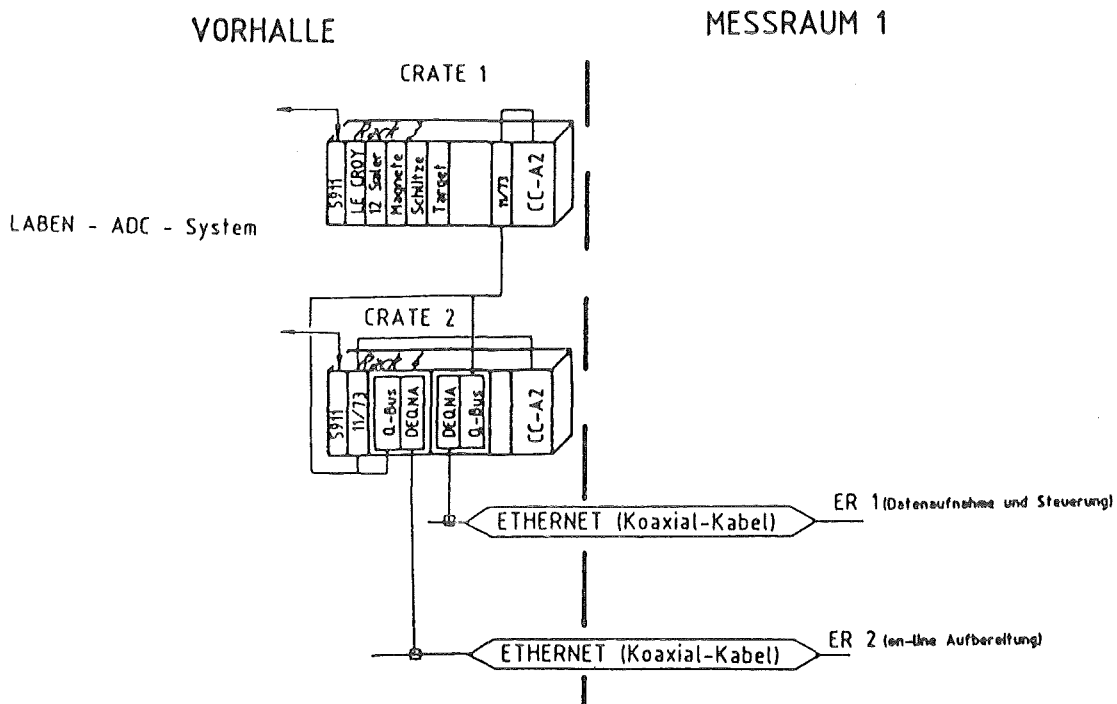


Abb.4: Kopplung der beiden Frontend-Rechner über Q-Bus

3.3 KOPPLUNG DER BEIDEN CAMAC-DATAWAYS ÜBER DATAWAY-KOPPLER

Eine Möglichkeit zur Datenübertragung zwischen zwei CAMAC-Crates bietet die Verwendung eines CAMAC-Moduls zur parallelen Kopplung der Dataways (Branch Highway Transceiver, Abb.5 Seite 25). Ein solches CAMAC-Modul würde, mit dem Crate-Controller des zweiten CAMAC-Crates verbunden, eine unidirektionale Verbindung der Dataways darstellen. Bei dieser Lösungsvariante wäre man darauf beschränkt, die zusätzlichen Nicht-LABEN ADCs in nur einem Crate unterzubringen. Um die Symmetrie der beiden Datenwege zu wahren, sollte man bei der Kopplung der CAMAC-Crates ein solches Parallel-Koppelmodul in jedem der beiden Crates vorsehen, um einen bidirektionalen Datenaustausch zu ermöglichen und somit sämtliche Plätze in beiden Crates für Erweiterungen nutzen zu können. Der Investitionsaufwand würde ca. 12000 DM betragen.

3.4 KOPPLUNG DER BEIDEN CAMAC-DATAWAYS ÜBER VME-BUS

Bei dieser Möglichkeit der Kopplung der beiden CAMAC-Crates werden die beiden CAMAC-Dataways mittels eines übergeordneten Bus-Systems, z.B. über einen VME-Bus, miteinander gekoppelt (Abb.6 Seite 25). Beim VME-Bus /9/ handelt es sich um einen sehr schnellen (maximal 20 MByte/s) 32 Bit breiten Bus, der auf modernste Prozessoren (MOTOROLA 680XX) abgestimmt ist. Auf Grund der umfangreichen Möglichkeiten, auf der Basis des VME-Busses und der speziellen VME-Bus-Peripherie ein komplexes Multiprozessorsystem aufzubauen, welches für die Zukunft alle Anforderungen an ein modernes Prozeßautomatisierungssystem erfüllt, ist dieser Vorschlag, zumindest für eine spätere Modernisierung, attraktiv. Dies ist umsomehr der Fall, als auch andere Gruppen das Konzept des VME-Busses für kernphysikalische Anwendungen einsetzen /10/. Da es auf der Basis des VME-Busses bereits ein sehr umfangreiches Peripherieangebot (I/O-Module, Graphik-Module, Festplatten, Bandeinheiten, Graphik-CPU-Karten, auch echtzeitfähig) sowie eine Vielzahl von CPU-Karten (16-32 Bit, MM68000 - 68020) gibt, eröffnet sich die Möglichkeit, nach und nach ganz auf ein einheitliches Bussystem und neuere Rechner (MM680XX VME-Karte) umzusteigen. Weiterhin läßt sich auf der Basis des VME-Busses sehr einfach ein Multiprozessorsystem aufbauen. Der Umstieg auf dieses Bussystem könnte

schrittweise vonstatten gehen, indem man den VME-Bus zunächst nur zur Kopplung der beiden CAMAC-Crates einsetzt und die übrige Peripherie beibehält. Dazu wäre es notwendig, einen VME-Bus-Überrahmen, ein CPU-Modul (MM68000), eine VME-Wrap-Karte und zwei CAMAC-Kopplungsmodule (AIAC-Module, vom Reaktorlabor der DFVLR-Garching /11/ entwickelt) anzuschaffen. Die Investitionskosten beliefen sich auf ca. 15000 DM. Zur Erstellung der Prozeßsoftware sollten Sprachen und Betriebssysteme verwendet werden, welche die echtzeitorientierte Prozeßlenkung durch die Bereitstellung spezieller Echtzeitoperationen unterstützen. Als Betriebssysteme kämen z.B. UNIX, PDOS oder OS9/68000 in Frage, wobei sich als mögliche Programmiersprache z.B. PEARL, ADA, Realtime-FORTRAN oder Realtime-PASCAL anbieten würden, die ein stark strukturiertes Programmieren ermöglichen. Das MPDS sollte natürlich ebenfalls wieder in das lokale Rechnernetz LAN eingebunden werden.

Nach Rücksprache mit einer Gruppe, die eine Kopplung von CAMAC und VME-Bus realisiert hat (DFVLR-Garching/10/), und nach eigener Marktanalyse stellte sich jedoch heraus, daß es im Augenblick noch keinen Anbieter für die in der Kernphysik verwendeten ADC-Module mit den in Abschnitt 2.1 genannten Spezifikationen auf VME-Bus-Basis gibt. Viele Hersteller von Modulen in CAMAC-Norm entwickeln angeblich jedoch bereits Module in VME-Bus-Norm. Beobachtet man die Entwicklungen in den kernphysikalischen Labors (nicht nur in

Deutschland) und bei den Herstellern von Peripherie-Modulen, so ist abzusehen, daß der VME-Bus das weitverbreitete CAMAC-System vielerorts ersetzen wird. Das Konzept VME-Bus darf also keineswegs für zukünftige Entwicklungen außer acht gelassen werden. Bis geeignete Module in VME-Bus-Norm auf dem Markt sind, müßte zumindest während der Übergangsphase noch ein CAMAC-Crate beibehalten werden. Mit einer solchen Kombination VME-CAMAC (Kopplung mit AIAC-Modul, DFVLR) hätte man ein sehr flexibles Datenaufnahmesystem mit sehr schneller Datenverarbeitung zur Verfügung, welches es erlaubt, vorhandene ADCs (CAMAC) zu verwenden, bis entsprechende VME-Bus Typen erhältlich werden. Bei der Verwirklichung eines solchen Systems könnte auch an die Eigenentwicklung entsprechender ADC-Module gedacht werden, um das doch recht langsame CAMAC-System (maximal 2,5 MByte/s) abzulösen.

Abschließend möchte ich noch darauf hinweisen, daß die Verwendung des an sich eigenständigen LABEN-Systems zusammen mit dem VME-Bus denkbar wäre, da nur an Stelle des CAMAC-Interface-Moduls 5911 ein Modul nach VME-Bus-Norm zu entwickeln ist. Somit könnte das überaus flexible und mit hohen Investitionen (ca. 60000 DM) angeschaffte MPDS weiterverwendet werden.

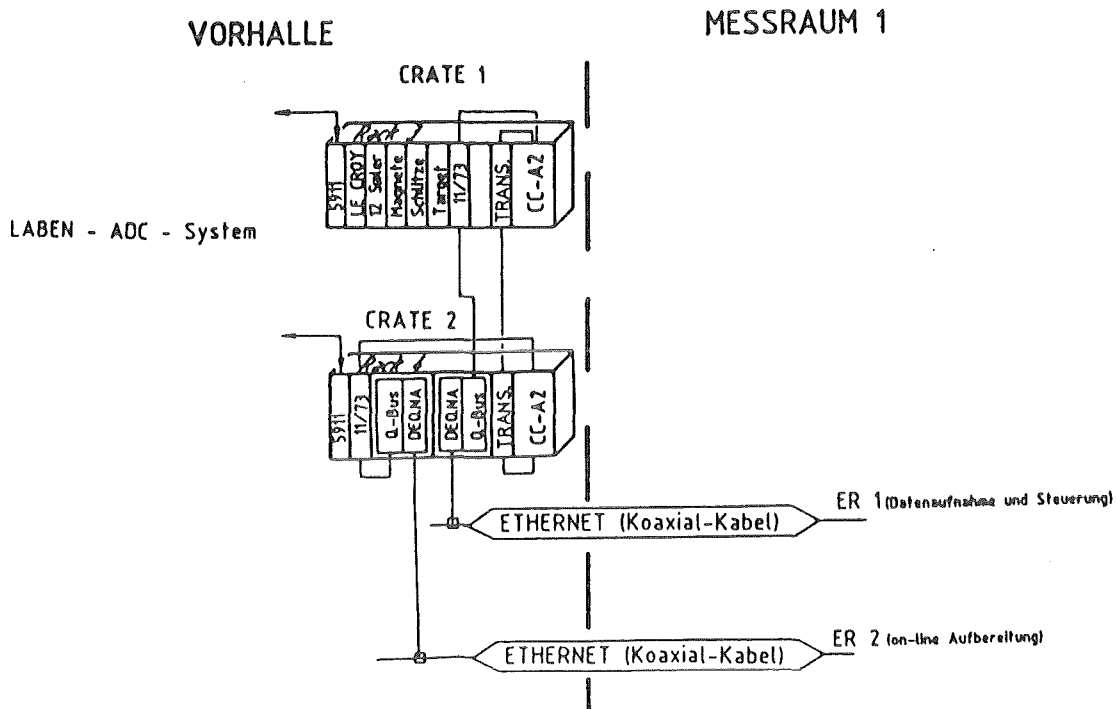


Abb.5: Kopplung der beiden CAMAC-Dataways über Dataway-Koppler

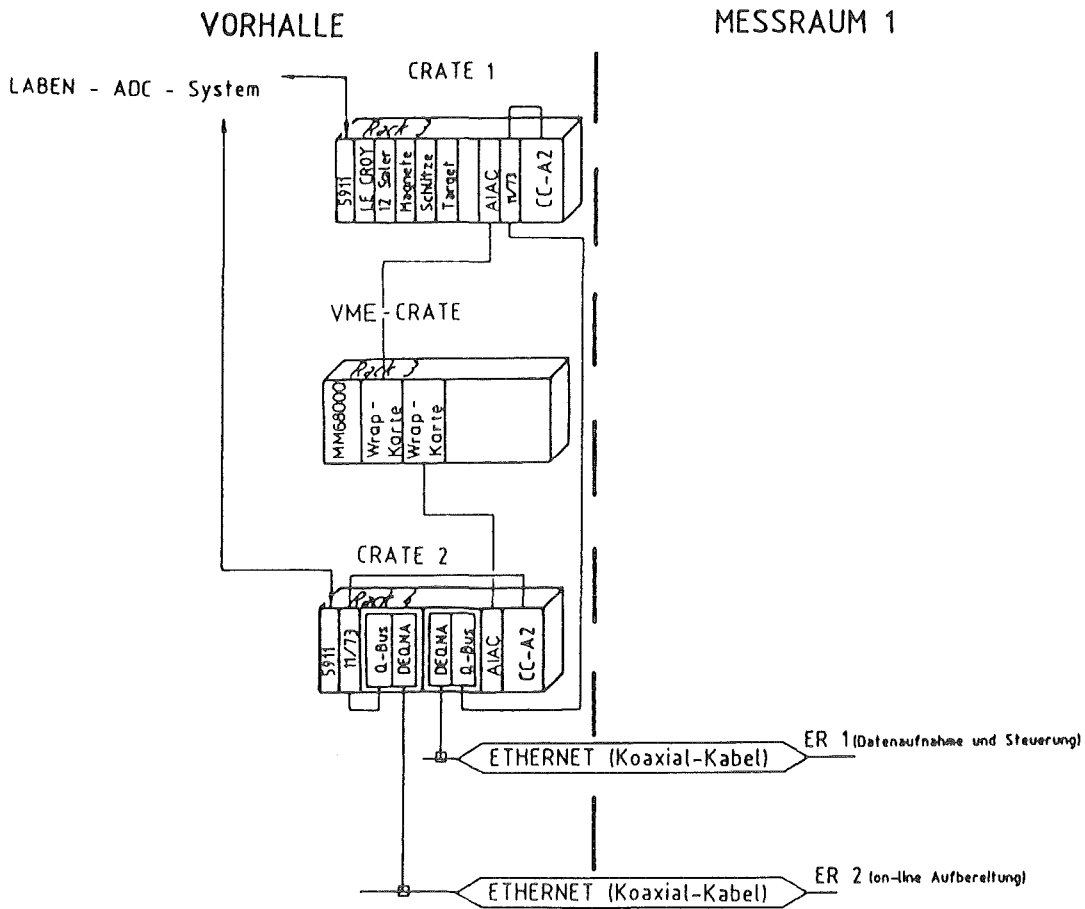


Abb.6: Kopplung der beiden CAMAC-Dataways über VME-Bus

3.5 KOPPLUNG DER BEIDEN RECHNERSYSTEME ÜBER EINE GEMEINSAME ETHERNET-VERBINDUNG

Die beiden Frontend-Rechner werden hierbei nicht mehr über separate ETHERNET-Kabel mit den Hostrechnern verbunden, sondern alle Rechner (Starburst 1 und 2 sowie ER 1 und 2) werden an ein gemeinsames ETHERNET-Kabel angeschlossen. Die Verwendung nur eines Datenübertragungskanal hat den Vorteil, daß sämtliche Rechner miteinander kommunizieren können. Für den diskutierten Anwendungsfall bedeutet dies, daß die Frontend-Rechner die benötigten ADC-Daten von den CAMAC-Crates lesen und sie gegenseitig austauschen. Dazu läuft auf jedem Frontend-Rechner ein Programm, welches die von CAMAC eingelesenen Daten dem anderen Frontend-Rechner zur Verfügung stellt. Diese Art der Rechnerkopplung ermöglicht es, sämtliche freien Steckplätze in den beiden CAMAC-Crates zur Systemerweiterung zu verwenden, wobei man sogar noch auf das Splitting-Interface des LABEN-Systems verzichten könnte, da auch diese Daten über ETHERNET ausgetauscht werden können. Ein weiterer Vorteil dieser Lösungsvariante besteht darin, daß sie mit der vorhandenen Hardware realisiert werden kann und dadurch keinerlei Mehrkosten verursacht. Hätte man

diese Art der Datenwegkopplung, die hauptsächlich auf einem betriebssicheren LAN-System basiert, schon in einem früheren Planungsstadium in Erwägung ziehen können, so hätte man neben dem Splitting-Interface auch auf eines der beiden CAMAC Interface-Module 5911 und die zweite ETHERNET-Verbindung (Meßraum - Vorhalle) verzichten können. Die Realisierung dieses Vorschlags wird im folgenden Kapitel 4 näher beschrieben und ist in der Gesamtübersicht (Abb.7) dargestellt.

VORHALLE

MESSRAUM 1

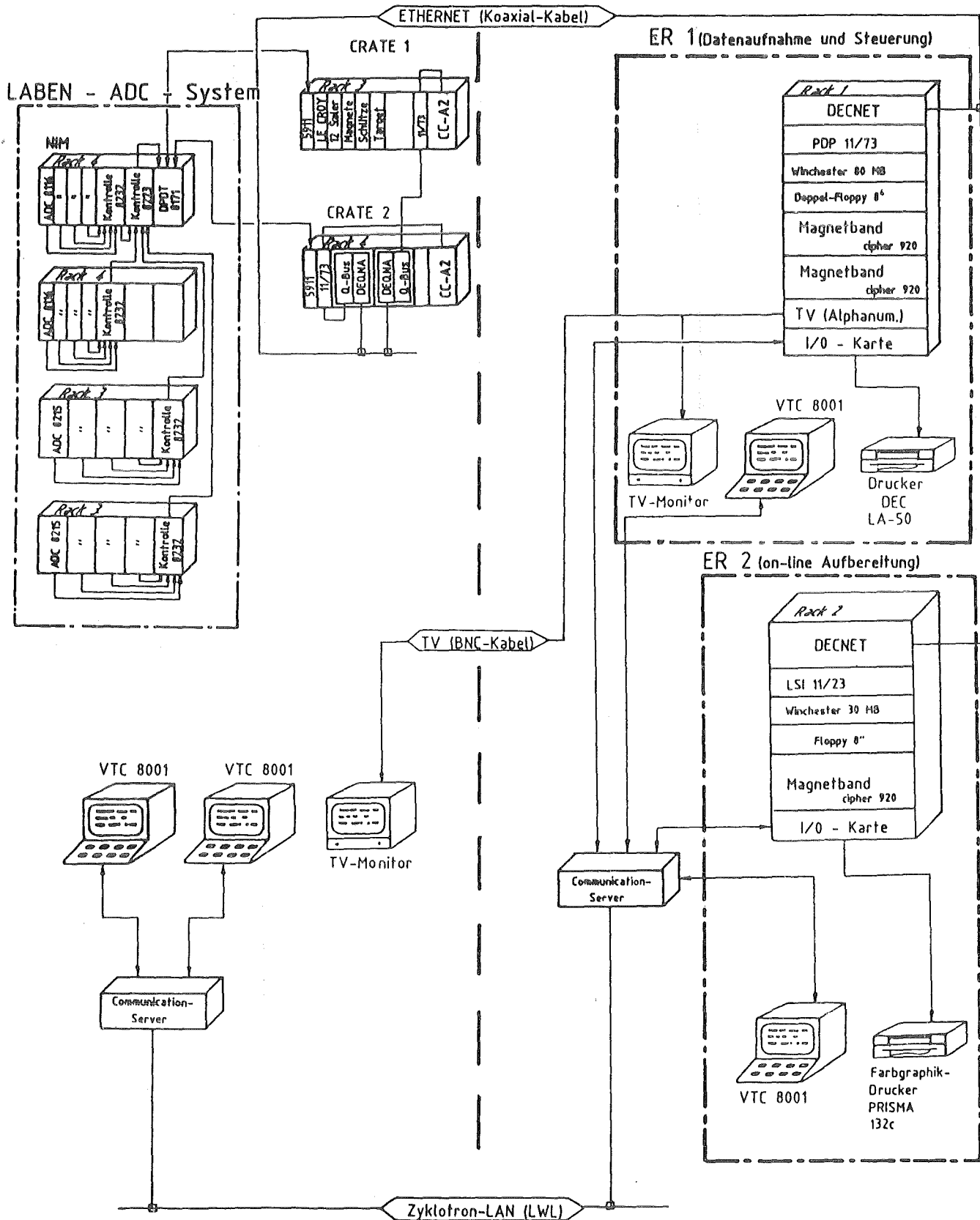


Abb.7: Gesamtübersicht Setup 2, Erklärung im Text Kapitel 4.

4. REALISIERUNG DER RECHNERKOPPLUNG ÜBER ETHERNET (SETUP 2)

Wie bei einem Vergleich der Übersichten Abb.1 und Abb.7 zu erkennen ist, unterscheidet sich der zweite Setup vom ersten im wesentlichen in der Vernetzungsstruktur der Rechner untereinander. Desweiteren zeigt sich auch die Verbindungsstruktur der peripheren Komponenten (Terminals, Drucker, TV-Monitore, etc.) überarbeitet. In Abb.7 sind sämtliche Hardwareverbindungen vollständig wiedergegeben.

Beginnt man, die beiden Aufbauvarianten von der ADC-Seite her miteinander zu vergleichen, so ist festzustellen, daß die vom LABEN ADC-System gelieferten Daten nach wie vor über das (vorhandene) Splitting-Interface (8171) und zwei CAMAC-Interfaces (5911) auf die beiden CAMAC-Crates übertragen werden. Auf diese Weise erspart man das in Abschnitt 3.5 erwähnte Transferprogramm zwischen den beiden Frontend-Rechnern zumindest für die LABEN-Listdaten.

Im CAMAC-Crate 1 sind ein CAMAC-Interface (5911), sämtliche I/O-Einstellmodule sowie der Frontend-Rechner 1 untergebracht. Die nach Koinzidenzgruppen sortierten Listdaten werden vom Frontend-Rechner 1 blockweise über die ETHERNET-Verbindung zum Host-Rechner ER 1 geschickt. Die Datenübertragung der I/O-Module erfolgt vom Rechner ER 1 zum Frontend-Rechner 1 und umgekehrt

ebenfalls über ETHERNET. In CAMAC-Crate 2 befinden sich ebenfalls ein CAMAC-Interface (5911), ferner (aus Platzgründen) der Frontend-Rechner 2 sowie die beiden Minibackplanes zur Aufnahme der Karten der ETHERNET-Kopplung. Jedes der Minibackplanes ist mit einem der beiden Frontend-Rechner verbunden. Zur Erweiterung des bestehenden Datenaufnahmesystems können sämtliche freien Slots in den beiden CAMAC-Crates (ca.20 Stück) verwendet werden, was für zukünftige Entwicklungen als ausreichend anzusehen ist.

An dem zur Datenarchivierung und Parametereinstellung verwendeten Rechner ER 1 sind eine 80 MB-Platte, eine Doppel-Disketteneinheit (2 x 8" DS DD), zwei Magnetbandgeräte (CIPHER 920) und zwei TV-Monitore (Rasterbildschirme) direkt angeschlossen. Der Rechner enthält eine Schnittstelle zur gruppeneigenen ETHERNET-Verbindung, die zum Datenaustausch sowie zum Downloading der Betriebsprogramme auf die beiden Frontend-Rechner dient. Entwickelt und gespeichert werden diese Betriebsprogramme auf dem Host-Rechner ER 1. Um von mehreren Terminals aus mit dem Rechner ER 1 kommunizieren zu können (Einstellung der Versuchsparameter, Steuerung der Datenaufnahme, off-line Datenauswertung, Softwareentwicklung), ist der Rechner über einen Communication-Server mit dem Zyklotron-LAN verbunden. Der zur on-line Auswertung verwendete Rechner ER 2 besitzt als periphere Geräte ebenfalls ein Magnetbandgerät, ferner ein 8" Disketten-Laufwerk und ein 30 MB Festplattenlauf-

werk. Auch dieser Rechner ist zur Softwareentwicklung und zur Ausgabe der on-line akkumulierten Spektren an das Zyklotron-LAN angeschlossen. Weiterhin ist er, um Meßdaten vom Starburst 1 zu empfangen, noch mit dem gruppeneigenen ETHERNET verbunden.

An jedem der beiden Rechner ist ein Drucker angeschlossen. Der Rechner ER 2 verfügt über einen Farbgraphik-Drucker (PRISMA 132c) zum Ausdrucken der on-line erzeugten Spektren, der aber während der off-line Datenauswertung auch am Rechner ER 1 betrieben werden kann. Bei dem an den Rechner ER 1 angeschlossenem Drucker handelt es sich um einen einfachen Matrix-Drucker (DEC PA 50) zur Erstellung von Programmlistings und Protokollierung des Experimentablaufes.

Zugang zu den beiden Rechnern ER 1 und ER 2 haben die Experimentatoren bzw. Programmentwickler über zwei gruppeneigene Farbgraphik-Terminals (VTC 8001) in der Vorhalle und zwei weitere gruppeneigene Farbgraphik-Terminals (VTC 8001) im Meßraum 1 sowie über beliebige andere am LAN befindliche Terminals. Wie in Abb.7 erkennbar, sind die Terminals wie auch die Rechner über Communication-Server (zur Protokollanpassung bzw. LAN-Steuerung) ans LAN angeschlossen.

Während des Experiments dient je ein Farbterminal in der Vorhalle und im Meßraum 1 dazu, die Experimentparameter einzustellen sowie den Programmab-

lauf über den Rechner ER 1 zu steuern, während die beiden anderen Farbterminals zur Bedienung des Rechners ER 2 und zur Darstellung der on-line Spektren dienen. Zusätzlich werden Statusinformationen bezüglich der Experimentparameter vom Rechner ER 1 über je einen TV-Monitor in der Vorhalle bzw. im Meßraum 1 ausgegeben.

5. INBETRIEBNAHME EINES STARBURST-RECHNERS

Bevor die in der Aufgabenstellung dieser Arbeit geforderten Programmmodule der Auslesesoftware entwickelt werden konnten, war es notwendig, ein CAMAC-Crate und zumindest einen der beiden Starburst-Rechner in Betrieb zu nehmen. Eine Reihe von Anfangsschwierigkeiten ergaben sich dadurch, daß auf keinerlei Erfahrungen mit dem zu verwendenden Starburst-Rechner und mit dem für diesen Rechner vorgesehenen Betriebssystem RSX-11M zurückgegriffen werden konnte.

Die folgenden beiden Kapitel beschreiben den Aufbau des verwendeten CAMAC-Systems mit Starburst-Rechner sowie die Installation des Betriebssystems mit der CAMAC-Library.

5.1 AUFBAU DES CAMAC-SYSTEMS

Als Schnittstelle zwischen Experimentelektronik (ADCs) und Rechner dient das in den vorhergehenden Kapiteln mehrfach erwähnte CAMAC-System (Computer Aided Measurement And Control), das im folgenden näher beschrieben werden soll.

Das CAMAC-System wurde in den europäischen Kernforschungszentren als Datenübertragungssystem entwickelt. Es handelt sich dabei um ein standardisiertes Interface zwischen einem technischen Prozeß (Experiment) und einem beliebigen Rechner. Neben den Übertragungsprozeduren sind die mechanischen Abmessungen, die elektrischen und die logischen Funktionen festgelegt. Bei der Verwendung eines Starburst-Rechners, als dem CAMAC-System zugeordneten Rechner, besteht das System aus einem CAMAC-Crate (19" Überrahmen) mit zugehörigem Crate-Kontroller und dem ebenfalls im CAMAC-Crate untergebrachten Starburst-Rechner. Von den 22 noch freien Steckplätzen wurden zwei weitere durch das CAMAC-Interface des LABEN-Systems (Modell 5911) sowie durch den LE CROY 12-fach ADC belegt. Der Starburst-Rechner kann jedes einzelne Modul im CAMAC-Crate ansprechen (z.B. Register lesen oder löschen) sowie CAMAC-Interrupts (LAM \equiv Look At Me), lokalisieren. Erzeugt ein Modul einen Interrupt, so meldet es diesen über eine Stichleitung (LAM-Leitung) dem Starburst, welcher den Interrupt lokalisieren und die entsprechende Interruptroutine starten kann.

5.2 INSTALLATION DES STARBURST-RECHNERS

Bei dem verwendeten Starburst-Rechner der Firma CES (Creative Electronic Systems) /12/ handelt es sich um ein einfach breites CAMAC-Modul, in welchem ein vollständiger Rechner mit PDP 11/73-CPU und einem Speicher von 512 KByte untergebracht ist. Seine Spannungsversorgung bezieht der Starburst von der CAMAC-Verdrahtung über einen Direktstecker an der Geräterückseite. Der rechnerinterne Q-Bus ist auf zwei 50-polige Steckerleisten an der Frontseite herausgeführt. Zum Betrieb des Rechners ist das Modul in einen CAMAC-Überahmen zu stecken, und folgende Verbindungen sind herzustellen:

1. Verbindung des Starburst mit dem Crate-Controller über die jeweils an der Rückseite der Geräte befindlichen 40-poligen Steckerleisten.
2. Verbinden des Starburst mit einem Terminal (hier mit dem Zyklotron-LAN) über den 10-poligen Stecker an der Vorderseite.
3. Die Anschlüsse RQT (Request) und Grant IN an der Vorderseite des Starburst sind mit einem kurzen LEMO 01-Kabel zu brücken.

4. Der Anschluß Grant OUT an der Frontseite des Starburst und der Anschluß Grant IN an der Frontseite des Crate-Controllers sind ebenfalls mit einem LEMO 01-Kabel zu verbinden.

5. An der LEMO 01-Buchse RSTRT (Restart) des Starburst sollte ein Taster zum Resetten des Rechners angebracht werden.

5.3 INSTALLATION DES STARBURST-BETRIEBSSYSTEMS

Bei dem Betriebssystem RSX-11M von DEC /14/ handelt es sich um ein echtzeitfähiges Multiuser-, Multitasking-Betriebssystem. Es ist daher besonders gut geeignet, komplexe Prozeßsteueraufgaben zu übernehmen und quasi gleichzeitig Nebenaufgaben, z.B. den Datenaustausch mit anderen Rechnern, zu erledigen. Um das Betriebssystem auf den Starburst zu spielen, wurde er über seine Q-Bus-Verbindung mit einem normalen Q-Bus-Crate verbunden, in welchem sich der Kontroller eines 30 MB Festplattenlaufwerks, der Kontroller eines Diskettenlaufwerks sowie eine Karte mit Druckerschnittstelle befand. Zuvor war das Betriebssystem RSX-11M (für das eine Kopierlizenz vorhanden ist), mit Hilfe eines normalen PDP 11/73-Rechners von dessen Festplatte auf die

mit dem Starburst Q-Bus verbundene Festplatte kopiert worden. Nachdem das Betriebssystem und ein FORTRAN77-Compiler auf dem Starburst installiert waren, wurde noch die vom Hersteller des Starburst vertriebene, auf den FORTRAN77-Compiler von DEC zugeschnittene CAMAC-Library /13/ von Diskette auf das Festplattenlaufwerk des Starburst kopiert und zum Betriebssystem dazugebunden.

Im späteren Ausbau des Datenaufnahmesystems (Setup 2) wird die Programmentwicklung nicht auf dem Starburst, sondern auf dem Rechner ER 1 (PDP 11/73) durchgeführt werden. Die lauffähigen Programmodule werden dann über die ETHERNET-Vernetzung der Rechner in den Hauptspeicher des Starburst geladen und gegebenenfalls gestartet. Dazu ist es jedoch notwendig, die Betriebssystemergänzungen zum Datenaustausch über ETHERNET (ETHERNET-Treiber) als Erweiterungen des Betriebssystems anzuschaffen und zu installieren.

6. PROGRAMMBESCHREIBUNGEN

In den folgenden Abschnitten werden die Prototypprogramme zum Test der ADC-Systeme beschrieben. Vorausgehend soll aber noch auf einige Besonderheiten bei der Entwicklung von CAMAC-Software für einen Starburst-Rechner eingegangen werden.

Bei der Erstellung von FORTRAN-Programmen mit CAMAC-Zugriffen unter RSX-11M für einen Starburst-Rechner sind folgende Punkte zu beachten:

1. Das Betriebssystem muß die CAMAC-Library enthalten (die Files CAMLIB.OLB und CACOM.OLB unter dem Account [1,1])
2. Beim Linken eines Programms muß die CAMAC-Library mit angegeben werden und bei der Verarbeitung von CAMAC-Interrupts muß zusätzlich noch das File RSX11M.STB (Symbol Table, Account [1,1]) mit zum compilierten FORTRAN-Programm dazugebunden werden. Desweiteren muß durch die Angabe PR:0 eine privilegierte Task erzeugt werden, die hauptspeicherresident direkt über dem Betriebssystem (Pool) abgelegt wird. Die Eingabezeile zum Linken des Programms hat demnach folgendes Aussehen:

LINK/PR:0/opt 'File'/CODE:FPP,[1,1]RSX11M.STB,[1,1]CAMLIB/LIB,

F77FCS/LIB

3. Bei der Abfrage der Link-Options durch den Rechner muß noch die Option:
COMMON=CACOM:RW eingegeben werden. Diese Eingabe ist notwendig, damit
im Hauptspeicher ein residenter COMMON-Bereich allokiert wird, über den
Daten zwischen CAMAC und dem FORTRAN-Programm ausgetauscht werden.

Beim Testen der im folgenden beschriebenen Programme wurde ein Hardware-Mangel des Starburst festgestellt, welcher als Seiteneffekt zu einigen Befehlen aus der CAMAC-Library zutage tritt. Dieser Seiteneffekt macht sich als ungewollte Aktivität des Starburst auf dem CAMAC-Bus bemerkbar, die zur Folge hat, daß sich Register von anderen CAMAC-Modulen verändern. Dieser Hardware-Fehler wurde mittlerweile auch von anderen Benutzern eines anderen Starburst-Rechners (der Firma CES) festgestellt und ist dem Hersteller mitgeteilt worden. Der Fehler soll beseitigt werden.

Das Starburst-Betriebssystem und die in den Abschnitten 6.1 bis 6.4 beschriebenen Programme befinden sich auf der logischen Platte DL2 des Festplattenlaufwerks am Rechner ER 2, wobei die Testprogramme unter dem Account [3,3] zugänglich sind.

6.1 PROGRAMM ZUM EINSTELLEN DES LABEN ADC-SYSTEMS

Programmname: STELL

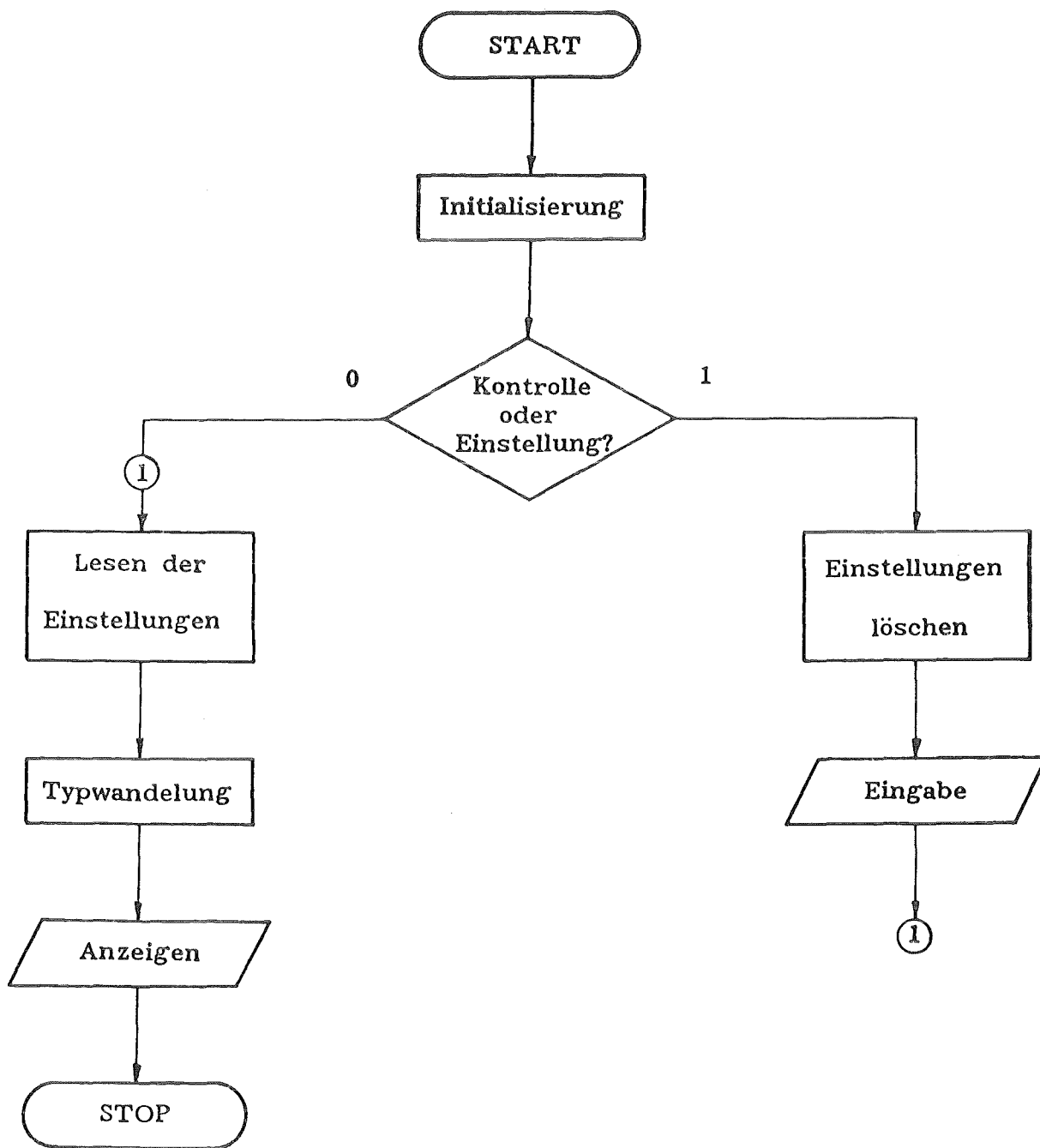
Programmbeschreibung:

Das Hauptprogramm STELL dient zur Verifizierung bzw. Einstellung der Koinzidenzkombinationen des LABEN Multiparameter-Systems.

Zu Beginn des Hauptprogramms werden zunächst die benötigten Nichtstandard-Variablen deklariert und das CAMAC-System mit dem CAMAC-Interface 5911 initialisiert. Danach verzweigt das Programm je nach Benutzereingabe zum Programmteil, der die Einstellungen verifiziert (Eingabe: 0) oder zum Programmteil, der die Koinzidenzgruppen neu festlegt (Eingabe: 1). Zur Anzeige der Einstellungen werden diese zuerst über CAMAC von den LABEN-Kontrolleinheiten als Integer-Zahl gelesen und zur Darstellung binär gewandelt. Nach der Typwandelung werden die Einstellungen mit der entsprechenden Adresskennung auf den Bildschirm ausgegeben und das Programm beendet.

Zur Neudefinition der Koinzidenzgruppen werden zunächst sämtliche möglichen Einstellungen der Kontrolleinheiten gelöscht (65536 Möglichkeiten). Danach erfragt das Programm die neuen Einstellungen mit der Angabe der Adresse. Wurden alle Eingaben durch den Benutzer korrekt ausgeführt, so werden die Register der LABEN-Kontrolleinheiten mit den neuen Einstellungen über CAMAC

beschrieben. Zur Kontrolle der Einstellungen werden die Register anschließend wieder gelesen und die Einstellungen am Bildschirm ausgegeben.



Programmablaufplan: Des Programms STELL

6.2 PROGRAMM ZUR ÜBERNAHME DER LABEN-LISTDATEN

Programmname: LAB

Programmbeschreibung:

Das durch CAMAC-Interrupts von der LABEN-Kontrolleinheit 8233 gesteuerte Hauptprogramm LAB liest auf jedes LAM hin 1 mal den Eventpuffer, das ist der in 2.1.1 beschriebene FIFO-Speicher des LABEN-Systems und bricht nach mehreren LAMs selbständig ab. Danach gibt es die Listdaten mit ADC- und Koinzidenzgruppenkennung am Bildschirm aus (vgl. Abschnitt 2.1.1).

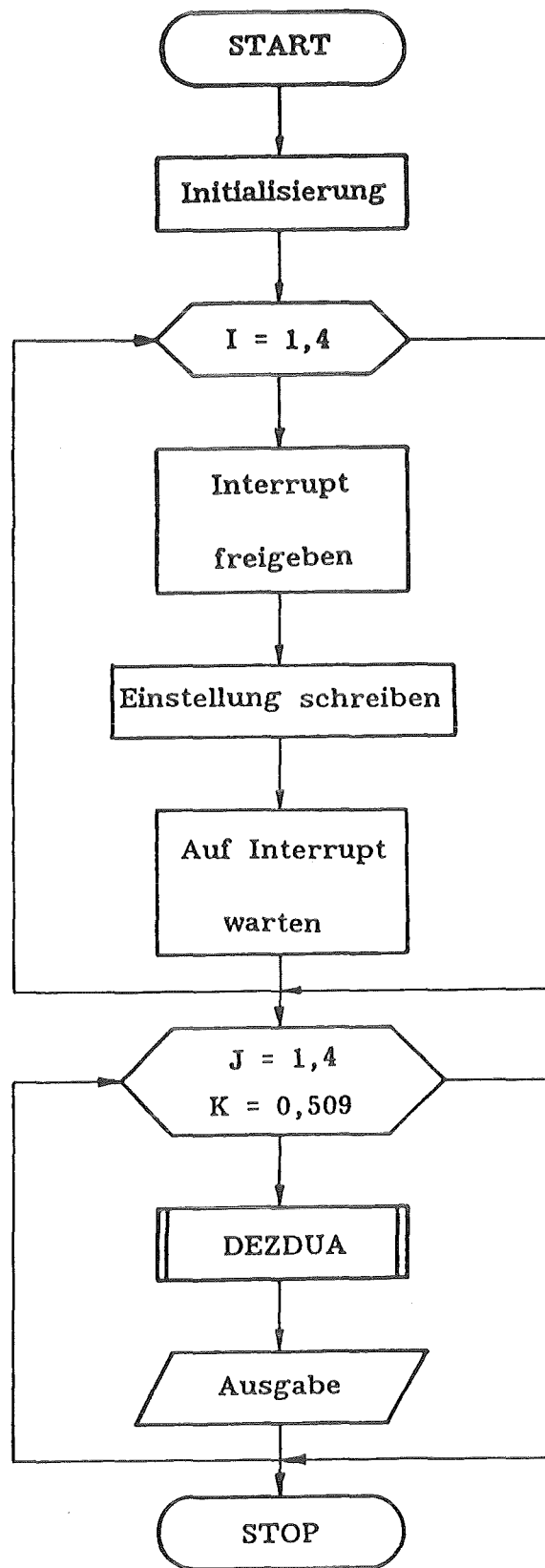
Das Hauptprogramm beginnt mit der Deklaration der benötigten Variablen und Felder. Danach wird der zur Parameterübergabe an das Unterprogramm LABELS notwendige COMMON-Bereich definiert. Nachdem das CAMAC-Interface 5911 initialisiert (N-Adresse des Moduls = Wert der Variablen NOFLAM) und die Freigabe des Interrupts gegeben wurde, wartet das Programm auf einen von der Kontrolleinheit 8233 herrührenden Interrupt und ruft beim Eintreffen eines solchen die Subroutiene LABELS (s.u.) auf, die den Listdatenpuffer liest. Die Anzahl der Lesevorgänge wird durch eine DO-Schleife bestimmt, welche nach jeder Rückkehr aus dem Unterprogramm durchlaufen werden muß. Zum Abfangen des in Kapitel 6.0 besprochenen Hardwarefehlers des Starburst werden, bei jedem Schleifendurchlauf die Koinzidenzgruppen des LABEN-Systems neu defi-

niert. Wurde die Maximalzahl der Schleifendurchläufe erreicht, so werden die Listdaten mit den genannten Kennungen am Bildschirm ausgegeben, wobei zur Wandelung der rechnerinternen Integerdarstellung der Daten jeweils das Unterprogramm DEZDUA (s.u.) aufgerufen wird. Nach der Ausgabe der Meßwerte bricht das Programm ab.

Neben dem Programm LAB wurden noch zwei weitere Versionen zur Übernahme von LABEN-Listdaten entwickelt, die sich nur geringfügig vom Programm LAB unterscheiden und von denen auch nur die Unterschiede zu LAB erläutert werden sollen:

1. *Das Programm LAB2* schreibt die gelesenen LABEN-Listdaten in ein File (LAB2.DAT).
2. *Das Programm LAB3* liest die LABEN-Listdaten in einer Endlosschleife und gibt sie am Bildschirm aus.

Programmabbruch durch Eingabe von: ABO/TASK taskname

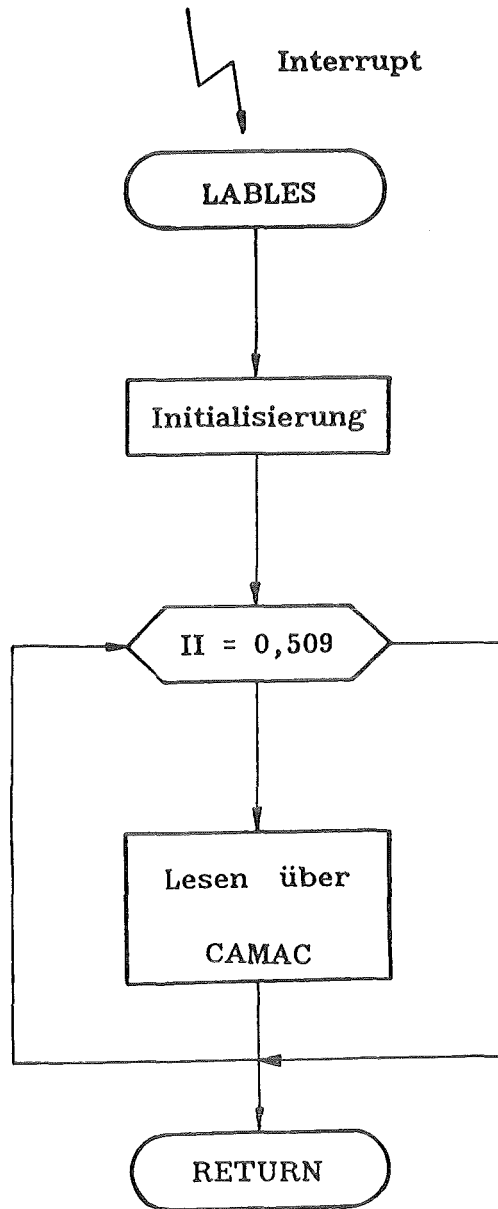


Programmablaufplan: Des Programms LAB

Unterprogrammname: LABELS

Programmbeschreibung:

Das Unterprogramm LABELS liest den Eventpuffer des LABEN-Systems über CAMAC, wobei die ADC-Daten in einem Feld abgelegt werden. Es startet sofort nach dem Eintreffen eines CAMAC-Interrupts des LABEN-Systems. Das Unterprogramm LABELS und das Hauptprogramm tauschen die beiderseitig benötigten Parameter über einen gemeinsamen COMMON-Bereich aus.

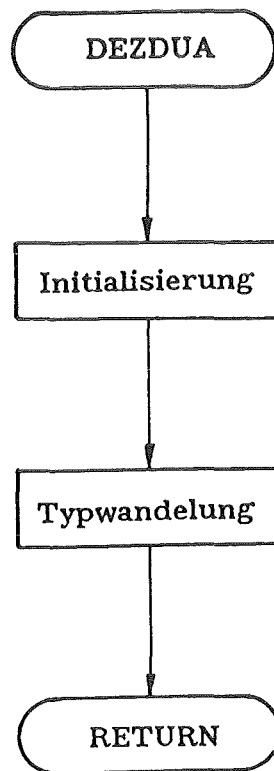


Programmablaufplan: Des Unterprogramms LABELS

Unterprogrammname: DEZDUA

Programmbeschreibung:

Das Unterprogramm DEZDUA wandelt rechnerinterne Zahlen in Integer-Darstellung, in Zahlen, die in Binär-Darstellung ausgegeben werden können. Die Parameterübergabe geschieht über Argumente.



Programmablaufplan: Des Unterprogramms DEZDUA

6.3 PROGRAMM ZUR ÜBERNAHME DER LE CROY-LISTDATEN

Programmname: LEC

Programmbeschreibung:

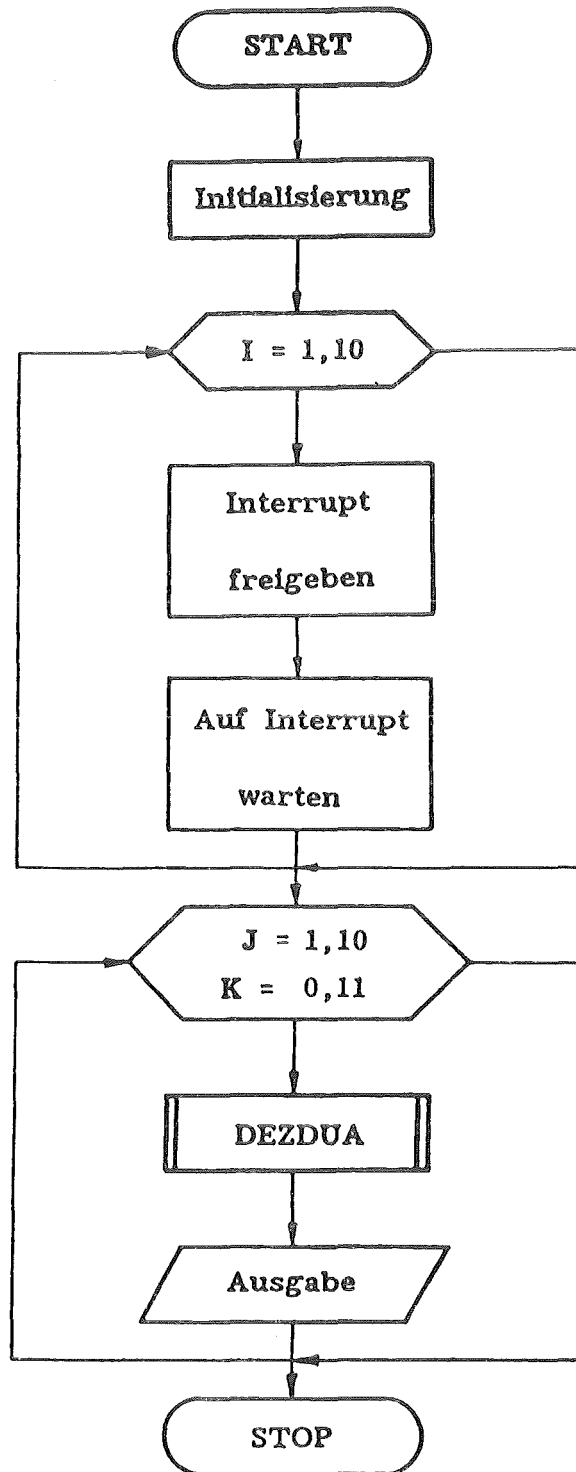
Das Programm LEC liest auf einen Interrupt des LE CROY ADC-Moduls hin die 12 Listdaten in der Reihenfolge ADC 0 - ADC 11 und gibt sie mit ADC- und Eventkennung am Bildschirm aus.

Das Hauptprogramm beginnt mit der Deklaration der benötigten Variablen und Felder, und dem Festlegen des zur Parameterübergabe an das Unterprogramm LECLES notwendigen COMMON-Bereichs. Nachdem das ADC-Modul initialisiert (N-Adresse des Moduls = Wert der Variablen NOFLAM) und die Freigabe des Interrupts gegeben wurde, liest das Programm das Modul (ADC 0-11), indem beim Eintreffen eines Interrupts des LE CROY-Moduls das Unterprogramm LECLES aufgerufen wird. Die Anzahl der Lesevorgänge wird durch eine DO-Schleife bestimmt, die nach jeder Rückkehr aus dem Unterprogramm durchlaufen werden muß. Wurde die Maximalzahl der Schleifendurchläufe erreicht, so werden die ADC-Daten mit den notwendigen Kennungen am Bildschirm ausgegeben, wobei zur Wandelung der rechnerinternen Integerdarstellung der Daten jeweils das Unterprogramm DEZDUA (siehe Kapitel 6.2) aufgerufen wird. Nach der Ausgabe der Meßwerte bricht das Programm ab.

Neben dem Programm LEC wurden auch hier noch zwei weitere Versionen zur Übernahme von LE CROY-Listdaten entwickelt, die sich nur geringfügig vom Programm LEC unterscheiden und von denen auch nur die Unterschiede zu LEC erläutert werden sollen:

1. *Das Programm LEC2* schreibt die gelesenen LE CROY-Listdaten in ein File (LEC2.DAT).
2. *Das Programm LEC3* liest die LE CROY-Listdaten in einer Endlosschleife und gibt sie am Bildschirm aus.

Programmabbruch durch Eingabe von: ABO/TASK taskname



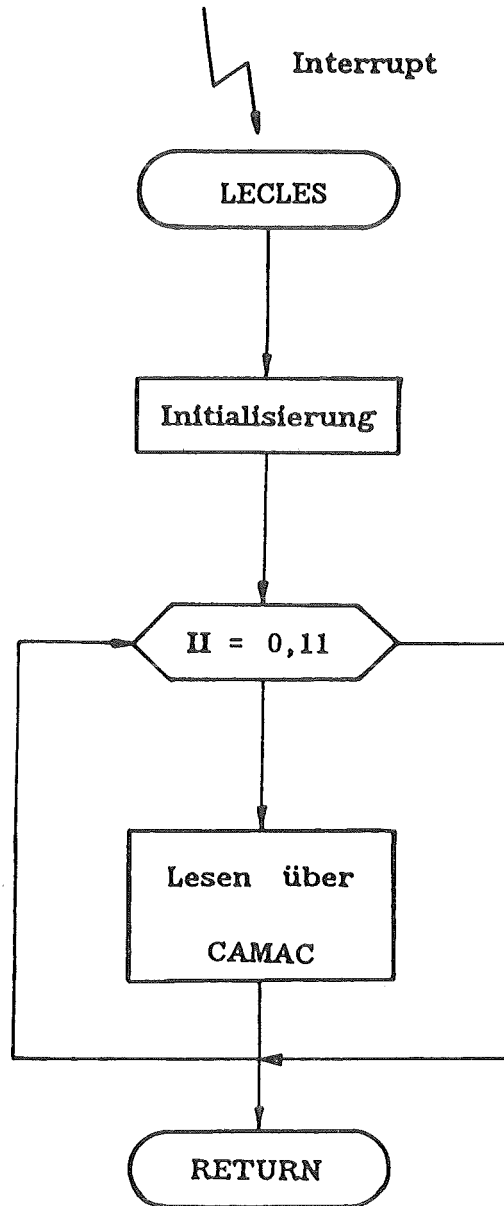
Programmablaufplan: Des Programms LEC

Unterprogrammname: LECLES

Programmbeschreibung:

Das Unterprogramm LECLES liest in einer Schleife die 12 ADCs des LE CROY ADC-Moduls über CAMAC und legt die 12 LE CROY-Listdaten in einem Feld ab.

Die Parameterübergabe zwischen dem Hauptprogramm und dem Unterprogramm LECLES geschieht über einen gemeinsamen COMMON-Bereich.



Programmablaufplan: Des Unterprogramms LECLES

6.4 PROGRAMM ZUR ÜBERNAHME GEMISCHTER LISTDATEN (LABEN U. LE CROY)

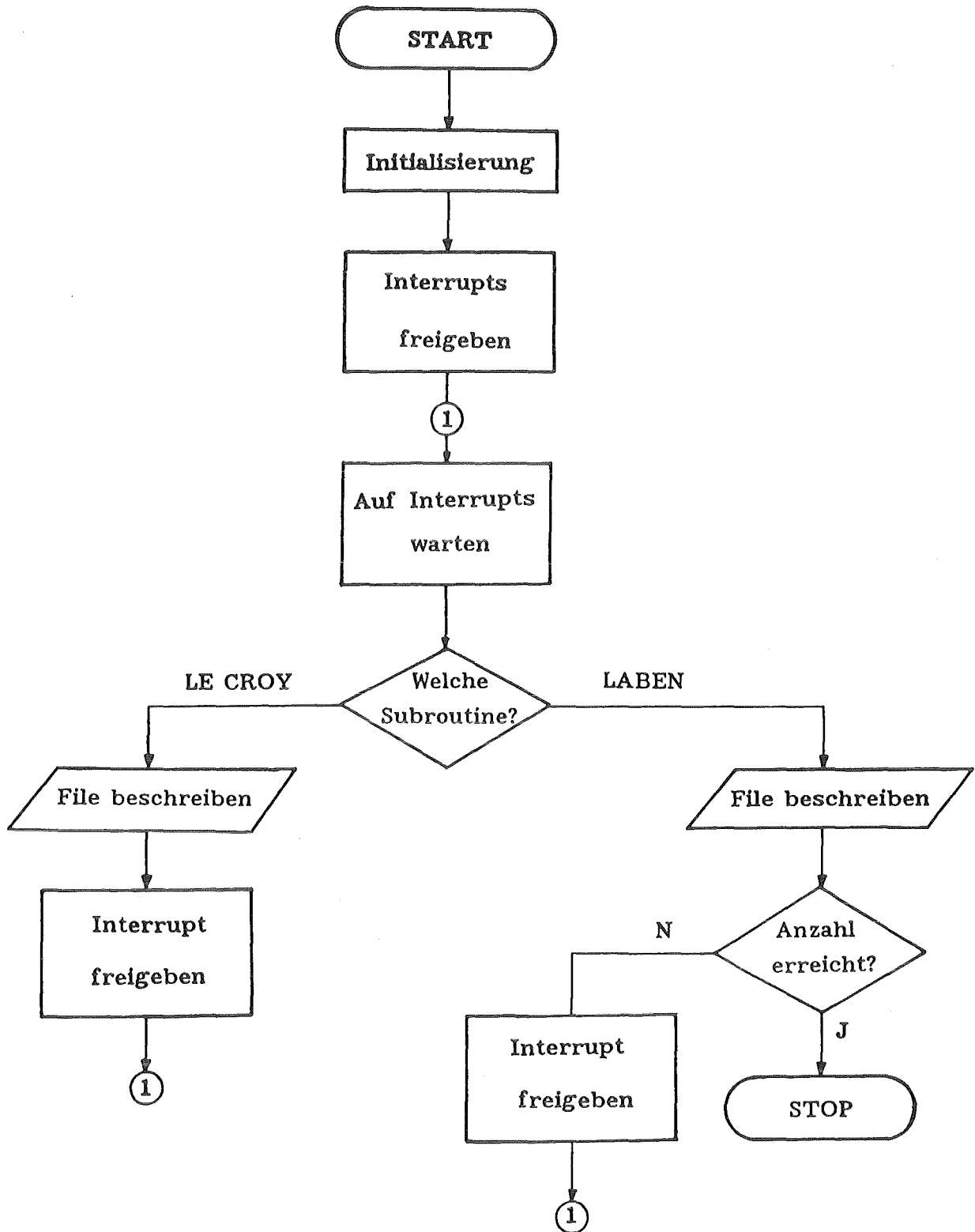
Programmname: LABLEC

Programmbeschreibung:

Das nachfolgend beschriebene Programm zur Übernahme von gemischten LABEN- und LE CROY-Listdaten konnte aufgrund des in Kapitel 6.0 beschriebenen Hardwarefehlers des Starburst noch nicht im simulierten Meßbetrieb wie die vorangegangenen Programme getestet werden, da durch das Vorhandensein mehrerer Interrupts das Überschreiben der Einstellungen des LABEN-Systems nicht mehr durch softwareseitiges Neuschreiben der Einstellungen abgefangen werden konnte. Jedes Neuschreiben der Koinzidenzeinstellungen des LABEN-Systems hat nämlich ein Löschen des Puffers in der Kontrolleinheit 8233 zur Folge, was bedingt, daß sich der Puffer nicht ganz füllt und somit kein Interrupt zum Auslesen des LABEN-Systems erscheint.

Die Idee bei der Programmerstellung war, daß bei abwechselndem Lesen des LE CROY-Moduls bzw. des LABEN-Eventpuffers koinzidente Listdaten aufgenommen werden, solange die Datenrate nicht in den Bereich der Frequenz kommt, die sich aus der Summe aus Konversionszeit und der Zeit zum Lesen des ADC-Moduls ergibt.

Das Programm beginnt mit der Deklaration der benötigten Variablen, dem Anlegen der COMMON-Bereiche sowie der Initialisierung der CAMAC-Module. Danach werden die CAMAC-Interrupts freigegeben. Das Hauptprogramm wartet dann auf die Beendigung des durch einen CAMAC-Interrupt aufgerufenen entsprechenden Unterprogramms (LABELS oder LECLES, siehe oben) und verzweigt nach der Rückkehr aus dem jeweiligen Unterprogramm zum Programmteil, durch welchen die Listdaten in ein File (COMB.DAT) weggeschrieben werden. Danach wird der CAMAC-Interrupt wieder freigegeben und das Programm wartet wieder auf das Eintreffen eines Interrupts bis ein Schleifenzähler einen bestimmten Wert erreicht hat (Anzahl der Lesevorgänge für den Eventpuffer). Danach werden alle CAMAC-Interrupts gesperrt und das Programm endet.



Programmablaufplan: Des Programms LABLEC

7. SCHLUßBEMERKUNGEN

Die vorliegende Arbeit wurde im Institut für Kernphysik III des Kernforschungszentrums Karlsruhe angefertigt. Herrn Professor Dr. G. Schatz und Herrn Professor Dr. H. Rebel möchte ich meinen Dank für die Ermöglichung und Förderung dieser Arbeit aussprechen. Für die Betreuung der Arbeit bedanke ich mich bei Herrn Dr. H.J. Gils, KfK, und Herrn Dipl.Ing. M. Haas, Berufsakademie Karlsruhe. Ferner möchte ich mich bei Herrn Dipl.Inf. H. Heinzmann, Herrn Dipl.Phys. J. Kiener und Herrn Dipl.Ing. S. Zagromski, alle KfK, für ihre selbstlose Mithilfe bei der Erstinstallation des Starburst-Betriebssystems und des LABEN ADC-Systems sowie beim Aufbau der Testelektronik bedanken.

ANHANG A. QUELLENVERZEICHNIS

/1/ Gils H.J. :

Design of a Simple Magnetic Spectrograph

for the Karlsruhe Isochronous Cyclotron

KfK Report 2972 (1980)

/2/ Gils H.J. :

Specification for the ADC-System

Unveröffentlichte Systemspezifikation

/3/ Rudolph, Flach, Roller :

NIK-Bericht Nr.3

KfK Report to be published

/4/ Gils H.J., Bialy J., Schlösser H., Zagromski S. :

The new data acquisition system for charged particle

experiments at the cyclotron

Annual Report IK, KfK Report 3969 (1985), Seite 176

/5/ Ottes J.G. :

CAMAC - Ein System rechnergeführter Elektronik

KfK Report 1402 (1971)

/6/ LABEN :

Via Bassini 15

20133 Milano / Italy

Manuals zum LABEN ADC-System 5911, 8232, 8233, DPDT 8171

Manuals zu den LABEN ADCs 8116, 8215

/7/ LE CROY RESEARCH SYSTEMS CORPORATION :

700 South Main Street

Spring Valley, N.Y. 10977 / USA

CAMAC MODEL 2259B, Analog to Digital Converter

LE CROY-Manual

/8/ Färber G. :

Bussysteme

Oldenbourg-Verlag (1982)

- Koch, Rembold, Ehlers :

Einführung in die Informatik Teil 1 u. 2

Carl Hanser Verlag München Wien (1977)

/9/ Furthermore, Mostec, Motorola, Signetics/Philips

VME-Bus Specification Manual

Rev.B. Aug. 1982

- Computer & Elektronik :

Datenspiele mit dem Host (VME-Bus Anwendungen)

VME-Bus contra Multibus (Board Markt)

Verlag Moderne Industrie München (Juni 1986)

/10/ Schöffel K., Liebl W. :

Der AIAC-Prozessor

Jahresbericht Tandem-Labor, TU und LMU München (1985), Seite 172

/11/ Schöffel K. :

MLLE-Ein datenflußkontrolliertes Multiprozessorsystem

Jahresbericht Tandem-Labor, TU und LMU München (1985), Seite 170

/12/ Creative Electronic Systems SA :

Route du Pont-Batin 70

CH - 1213 Petit-Lancy / Schweiz

- The User Manual for the CES 2180 Starburst Fast Processor

ACC 2180 family.ma Rev.1.00

/13/ Creative Electronic Systems SA :

CAMLIB, The F77 CAMAC-Library for the C E S ACC 2180

F77camlib Rev.1.10

/14/ Digital Equipment Corporation :

P.O.Box CS 2008

Nashua, NH 03061 / USA

- PDP-11 FORTRAN77

USER's Guide, Language Reference, Object Time System,

Instalation Guide

- RSX-11M (Version 4.1)

Volume 1 bis Volume 7

DEC-Manuals

ANHANG B. PROGRAMMLISTINGS

```
C*****
C*
C*      PROGRAMM ZUM EINSTELLEN DES LABEN SYSTEMS UEBER CAMAC      *
C*      UNTER RSX11 AUF EINEM STARBURST // MARIO NEUDOLD // SEP-86 // *
C*****
```

```
PROGRAMM STELL      !Programm zum Einstellen des LABEN-Sys.

INTEGER*4 IHILF,I2  !INT.*4 fuer 24 BIT CAMAC-DATEN
INTEGER INTA(2)     !Feld fuer CCLNK
INTEGER IEXT(2)     !Feld fuer CDREG
DIMENSION IARR(24)  !CHAR. fuer 24 Bit Dualzahlen
```

c- - - - INITIALISIERUNG DES STARBURST UND DES LABEN CAMAC-INTERFACE

```
NOFACC=23           !Position des Starburst
call CDSET(IX,NOFACC) !Initialisieren des Starburst
NOFLAM=11           !Position des LABEN-Interface

call CDREG(IEXT,0,0,NOFLAM,0) !Setzen von IEXT
call CCCC(IEXT)         !Clear Dataway
call CCCZ(IEXT)        !Initialize Dataway
INTA(1)=NOFLAM        !LAM Quelle (=Slotnummer)
INTA(2)=0

call CDLAM(LLAM,0,0,NOFLAM,0,INTA) !Setzen von LLAM
call CCLM(LLAM,.false.) !Disable LAM
```

c- - - - AUSWAHL ZWISCHEN DEM LESEN ODER DEM SCHREIBEN DER EINSTELLUNGEN

```
IN1=0
write(6,20)         !Auswahl zwischen
accept *,IN1       !Einstellungen ansehen bzw.
if(IN1.LT.0)goto 10 !Einstellungen aendern
if(IN1.GT.1)goto 10 !Abfangen von Falscheingaben
if(IN1.EQ.1)goto 70
```

c- - - - EINSTELLUNGEN LESEN UND ANZEIGEN

```
200 write(6,30)
do 40 I=1,5        !Alte Einstellungen ueber
call CDREG(IEXT,0,0,NOFLAM,I)
call CFSA(1,IEXT,IHILF,IQ) !CAMAC einlesen

do 50 J=1,24      !Wandelung INTEGER -> 24 BIT binaer
IARR(J)=JMOD(IHILF,2)
IHILF=IHILF/2
50 continue

write(6,60)I,(IARR(J),J=1,24) !Anzeige der Einstellungen
40 continue

goto 190          !Sprung zum Programmende
```


c- - - - ALTE EINSTELLUNGEN LOESCHEN UND NEUE EINLESEN

```
70      call CDREG(IEXT,0,0,NOFLAM,0)
        write(6,*)
        write(6,*)'Bitte warten, alte Einstellungen werden gecleart'

        do 80 IHILF=0,(2**16-1)
        call CFSA(17,IEXT,IHILF,IQ)      !Clearen der alten Einstellungen
80      continue

        write(6,90)                      !Eingabe
        write(6,100)

        do 140 J=0,5                      !der neuen
170     write(6,110)J
        read(5,130)(IARR(I),I=1,24)     !Einstellungen
        IHILF=0

        do 150 I=1,24                    !Wandelung der Eingaben
        if(IARR(I).LT.0) goto 160
        if(IARR(I).GT.1)goto 160        !24 Bit binaer -> INTEGER und
        if(IARR(I).EQ.0) goto 180
        IHILF=IHILF+I2**(I-1)          !abfangen von Fehleingaben

180     continue
150     continue
```

c- - - - SCHREIBEN DER NEUEN EINSTELLUNGEN UND ANZEIGE DIESER

```
        call CDREG(IEXT,0,0,NOFLAM,J)  !Schreiben der neuen
        call CFSA(17,IEXT,IHILF,IQ)    !Einstellungen ueber CAMAC

140     continue
        goto 200                        !Sprung zur Anzeige der Einstellungen
160     write(5,120)                    !Bei Eingabefehlern die
        goto 170                        !Meldung 'Falscheingabe' ausgeben

20      format(/('+', 'Einstellen oder Einstellung ansehen? 1/0 : '))
30      format(/(' Die Einstellungen des LABEN-System sind: ')/)
60      format((3X, ' A(', I1, ')      ', 24I1)/)
90      format(/(1X, 'Bitte Einstellungen im Format '))
100     format(' 00000000000000000000000000000000 eingeben: ')
110     format(/(' Eingaben fuer A(', I1, ') : ')/)
120     format((' Falscheingabe ! ')/)
130     format(24I1)

190     STOP

        END
```

```
c*****
c*
c*      TESTPROGRAMM ZUM AUSLESEN DES LABEN ADC-SYSTEMS UEBER CAMAC      *
c*      UNTER RSX11 AUF EINEM STARBURST // MARIO NEUDOLD // SEP-86 //    *
c*
c*      PROGRAMM LIEST DEN PUFFER DES LABEN-SYSTEMS (85 EVENTS, BEI      *
c*      5 ADCs) 4 MAL AUS UND SCHREIBT DIE DATEN AUF DEN BILDSCHIRM.    *
c*      (ADC-DATEN MIT LAM- UND ADC-KENNUNG)                             *
c*
c*****
```

```
PROGRAM LAB                                !Auslesen des LABEN ADC-SYSTEMS

external LABELS                            !Subroutine lesen des Puffers nach LAM
external DEZDUA                            !Subroutine zum Typwandeln

INTEGER*4 M, IDATA(0:4, 0:509)             !INT. fuer ADC - Daten
INTEGER INTA(2)                            !Feld fuer CCLNK
INTEGER IEXT(2)                             !Feld fuer CDREG
DIMENSION IARR(24)                         !Fuer 24 Bit ADC-Wert

COMMON /SUB/ IDATA, IEXT, I, NOFLAM, ILAM
```

c- - - - INITIALISIERUNG DES STARBURST UND DES LABEN-CAMACMODUL

```
NOFACC=23                                  !Position des Starburst
call CDSET(IX, NOFACC)                     !Initialisieren des Starburst
NOFLAM=11                                  !Position des ADC

call CDREG(IEXT, 0, 0, NOFLAM, 0)          !Setzen von IEXT
call CCCC(IEXT)                            !Clear Dataway
call CCCZ(IEXT)                            !Initialize Dataway

INTA(1)=NOFLAM                             !LAM Quelle (= Slotnummer)
INTA(2)=0
call CDLAM(ILAM, 0, 0, NOFLAM, 0, INTA)    !Setzen von ILAM
call CCLNK(ILAM, LABELS)                  !Zuweisung LAM-Subroutine
call CLREF(9)                              !Clear Flagge (9)
```

c- - - - BEGINN DER AUSLESESCHLEIFE

```
do 10 I=1, 4                               !Feld fuer 4 Pufferfuellungen
```

c- - - - EINSTELLEN DES LABEN SYSTEMS (entfaellt normalerweise!)

```
M=83888639
call CDREG(IEXT, 0, 0, NOFLAM, 0)          !Schreiben
call CFSA(17, IEXT, M, IQ)
M=31
call CDREG(IEXT, 0, 0, NOFLAM, 1)          !der
call CFSA(17, IEXT, M, IQ)
M=1028
call CDREG(IEXT, 0, 0, NOFLAM, 2)          !Einstellungen
call CFSA(17, IEXT, M, IQ)
```

```
M=62
call CDREG(IEXT,0,0,NOFLAM,3)  !des
call CFSA(17,IEXT,M,IQ)
M=574
call CDREG(IEXT,0,0,NOFLAM,4)  !LABEN
call CFSA(17,IEXT,M,IQ)
M=1086
call CDREG(IEXT,0,0,NOFLAM,5)  !ADC-Systems
call CFSA(17,IEXT,M,IQ)

call CDREG(IEXT,0,0,NOFLAM,1)  !LAM enable
call CFSA(26,IEXT,0,IQ)

call WAITFR(9)                  !Warten auf die Subroutine (Flagge (9))
call CLREF(9)                   !Clear Flagge (9)
call CCRGL(ILAM)                !Reconnect LAM

10  continue

c- - - - ENDE DER AUSLESESCHLEIFE / AUSGABE DER MESSWERTE

call CLREF(9)                   !Clear Flagge (9)
call CDZEL                      !Loeschen aller LAM-Eintraege

write(6,70)

do 20 J=1,4                     !Ausgabe
do 30 NN=0,504,6
JJ=JJ+1
do 90 N=0,5                     !der
K=N+NN
call DEZDUA (IDATA,J,K,IARR)    !Messwerte
call WAITFR(9)                 !Warten auf die Subroutine (Flagge (9))
write(6,40)J,(IARR(I),I=1,13),(IARR(I),I=14,20),(IARR(I),I=21,24),JJ,K

call CLREF(9)                   !Clear Flagge (9)
90  continue
write(6,*)
30  continue
write(6,*)
20  continue

40  format(1X,'LAM Nr.',I2,5X,13I1,2X,7I1,2X,4I1,4X,'EVENT Nr.',I3,4X,I3)
70  format(/(' ADC - WERTE ')/)

call CDREG(IEXT,0,0,NOFLAM,1)  !Disable LAM
call CFSA(24,IEXT,0,IQ)

write(6,*)'PROGRAMMENDE ERREICHT !'

STOP

END
```

C*****

subroutine LABLES

COMMON /SUB/ IDATA,IEXT,I,NOFLAM,ILAM

INTEGER*4 IDATA(0:4,0:509) !INT. fuer ADC - Daten
INTEGER IEXT(2) !Feld fuer CDREG

call CDREG(IEXT,0,0,NOFLAM,1) !Disable LAM
call CFSA(24,IEXT,0,IQ)

do 50 II=0,509 !Lesen
call CDREG(IEXT,0,0,NOFLAM,1)
call CFSA(2,IEXT,IDATA(I,II),IQ) !der

50 continue !ADCs

call SETEF(9) !Setzen der Flagge (9)

return
end

C*****

subroutine DEZDUA (IDATA,J,K,IARR)

INTEGER*4 IDATA(0:4,0:509),IHILF !INT. fuer ADC - Daten

DIMENSION IARR(24) !Fuer 24 Bit ADC - Daten

IHILF=IDATA(J,K)

do 60 L=1,24 !Wandelung
IARR(L)=JMOD(IHILF,2)
IHILF=IHILF/2

60 continue !Dezimal -> Dual

call SETEF(9) !Setzen der Flagge (9)
return
end

```
c*****
c*
c* TESTPROGRAMM ZUM AUSLESEN DES LABEN ADC-SYSTEMS UEBER CAMAC *
c*
c* UNTER RSX11 AUF EINEM STARBURST // MARIO NEUDOLD // SEP-86 // *
c*
c* PROGRAMM LIEST DEN PUFFER (85 EVENTS, BEI 5 ADCs) 2 MAL AUS *
c* UND SCHREIBT DIE DATEN IN EIN FILE(LAB2.DAT). *
c* (ADC-DATEN MIT LAM- UND ADC-KENNUNG) *
c* ANSCHAUEN DER MESSDATEN UEBER: TYPE LAB2.DAT *
c*
c*****
```

```
PROGRAM LAB2 !Auslesen des LABEN ADC-SYSTEMS

external LABLES !Subroutine lesen des Puffers nach LAM
external DEZDUA !Subroutine zum Typwandeln

INTEGER*4 M, IDATA(0:2,0:509) !INT. fuer ADC - Daten
INTEGER INTA(2) !Feld fuer CCLNK
INTEGER IEXT(2) !Feld fuer CDREG
DIMENSION IARR(24) !Fuer 24 Bit ADC-Wert

COMMON /SUB/ IDATA, IEXT, I, NOFLAM, ILAM
```

c- - - - INITIALISIERUNG DES STARBURST UND DES LABEN-CAMACMODUL

```
NOFACC=23 !Position des Starburst
call CDSET(IX,NOFACC) !Initialisieren des Starburst
NOFLAM=11 !Position des ADC

call CDREG(IEXT,0,0,NOFLAM,0) !Setzen von IEXT
call CCCC(IEXT) !Clear Dataway
call CCCZ(IEXT) !Initialize Dataway

INTA(1)=NOFLAM !LAM Quelle (= Slotnummer)
INTA(2)=0
call CDLAM(ILAM,0,0,NOFLAM,0,INTA) !Setzen von ILAM
call CCLNK(ILAM,LABLES) !Zuweisung LAM -> Subroutine
call CLREF(9) !Clear Flagge (9)
```

c- - - - BEGINN DER AUSLESESCHLEIFE

```
do 10 I=1,2 !Feld fuer 2 Pufferfuellungen
```

c- - - - EINSTELLEN DES LABEN-SYSTEMS (entfaellt normalerweise!)

```
M=83888639
call CDREG(IEXT,0,0,NOFLAM,0) !Schreiben
call CFSA(17,IEXT,M,IQ)
M=31
call CDREG(IEXT,0,0,NOFLAM,1) !der
call CFSA(17,IEXT,M,IQ)
M=1028
```

```
call CDREG(IEXT,0,0,NOFLAM,2) !Einstellungen
call CFSA(17,IEXT,M,IQ)
M=62
call CDREG(IEXT,0,0,NOFLAM,3) !des
call CFSA(17,IEXT,M,IQ)
M=574
call CDREG(IEXT,0,0,NOFLAM,4) !LABEN
call CFSA(17,IEXT,M,IQ)
M=1086
call CDREG(IEXT,0,0,NOFLAM,5) !ADC-Systems
call CFSA(17,IEXT,M,IQ)

call CDREG(IEXT,0,0,NOFLAM,1) !LAM enable
call CFSA(26,IEXT,0,IQ)

call WAITFR(9) !Warten auf die Subroutine (Flagge (9))
call CLREF(9) !Clear Flagge (9)
call CCRGL(ILAM) !Reconnect LAM
10 continue

c- - - ENDE DER AUSLESESCHLEIFE / WEGSCHREIBEN DER MESSDATEN INS FILE

call CLREF(9) !Clear Flagge (9)
call CDZEL !Loeschen aller LAM-Eintraege

open(UNIT=1,FILE='DL2:c3,3!LAB2.DAT',STATUS='UNKNOWN') !Oeffnen des
c !Datenfiles
write(1,70)

do 20 J=1,2 !Ausgabe
do 30 NN=0,504,6
JJ=JJ+1 !der
do 90 N=0,5
K=N+NN !Messwerte
call DEZDUA (IDATA,J,K,IARR)
call WAITFR(9) !Warten auf die Subroutine (Flag)
write(1,40)J,(IARR(I),I=1,13),(IARR(I),I=14,20),(IARR(I),I=21,24),JJ,K
call CLREF(9) !Clear Flagge (9)
90 continue
write(1,*)
30 continue
write(1,*)
20 continue

40 format(1X,'LAM Nr.',I2,5X,13I1,2X,7I1,2X,4I1,4X,'EVENT Nr.',I3,4X,I3)
70 format(/(1X,' ADC - WERTE (LABEN ADC-System)')/)

close(1) !Schliessen des Datenfiles

call CDREG(IEXT,0,0,NOFLAM,1) !Disable LAM
call CFSA(24,IEXT,0,QI)

STOP
END
```

C*****

subroutine LABLES

COMMON /SUB/ IDATA,IEXT,I,NOFLAM,ILAM

INTEGER*4 IDATA(0:2,0:509) !INT. fuer ADC - Daten
INTEGER IEXT(2) !Feld fuer CDREG

call CDREG(IEXT,0,0,NOFLAM,1) !Disable LAM
call CFSA(24,IEXT,0,QI)

do 50 II=0,509 !Lesen

call CDREG(IEXT,0,0,NOFLAM,1)
call CFSA(2,IEXT,IDATA(I,II),IQ) !der

50 continue !ADCs

call SETEF(9) !Setzen der Flagge (9)
return

end

C*****

subroutine DEZDUA (IDATA,J,K,IARR)

INTEGER*4 IDATA(0:2,0:509),IHILF !INT. fuer ADC - Daten

DIMENSION IARR(24) !Fuer 24 Bit ADC - Daten

IHILF=IDATA(J,K)
do 60 L=1,24 !Wandelung

IARR(L)=JMOD(IHILF,2)

IHILF=IHILF/2

60 continue !Dezimal -> Dual

call SETEF(9) !Setzen der Flagge (9)
return

end

ADC - WERTE (LABEN ADC-System)

LAM Nr. 1	1111100000000	0000000	1101	EVENT Nr. 1	0
LAM Nr. 1	0101011001010	0000000	1000	EVENT Nr. 1	1
LAM Nr. 1	1001001001010	1000000	1000	EVENT Nr. 1	2
LAM Nr. 1	0001001001010	0100000	1000	EVENT Nr. 1	3
LAM Nr. 1	0000000000000	1100000	1000	EVENT Nr. 1	4
LAM Nr. 1	00000010111010	0010000	1001	EVENT Nr. 1	5
LAM Nr. 1	1111100000000	0000000	1111	EVENT Nr. 2	6
LAM Nr. 1	1101011001010	0000000	1010	EVENT Nr. 2	7
LAM Nr. 1	0101001001010	1000000	1010	EVENT Nr. 2	8
LAM Nr. 1	0101001001010	0100000	1010	EVENT Nr. 2	9
LAM Nr. 1	0000000000000	1100000	1010	EVENT Nr. 2	10
LAM Nr. 1	00000010111010	0010000	1011	EVENT Nr. 2	11
LAM Nr. 1	1111100000000	0000000	1101	EVENT Nr. 3	12
LAM Nr. 1	1001011001010	0000000	1000	EVENT Nr. 3	13
LAM Nr. 1	1001001001010	1000000	1000	EVENT Nr. 3	14
LAM Nr. 1	0001001001010	0100000	1000	EVENT Nr. 3	15
LAM Nr. 1	0000000000000	1100000	1000	EVENT Nr. 3	16
LAM Nr. 1	00000010111010	0010000	1001	EVENT Nr. 3	17
LAM Nr. 1	1111100000000	0000000	1111	EVENT Nr. 4	18
LAM Nr. 1	0101011001010	0000000	1010	EVENT Nr. 4	19
LAM Nr. 1	0101001001010	1000000	1010	EVENT Nr. 4	20
LAM Nr. 1	0101001001010	0100000	1010	EVENT Nr. 4	21
LAM Nr. 1	0000000000000	1100000	1010	EVENT Nr. 4	22
LAM Nr. 1	00000010111010	0010000	1011	EVENT Nr. 4	23
.... bis					
LAM Nr. 2	1111100000000	0000000	1101	EVENT Nr. 168	492
LAM Nr. 2	0101011001010	0000000	1000	EVENT Nr. 168	493
LAM Nr. 2	0101001001010	1000000	1000	EVENT Nr. 168	494
LAM Nr. 2	1001001001010	0100000	1000	EVENT Nr. 168	495
LAM Nr. 2	0000000000000	1100000	1000	EVENT Nr. 168	496
LAM Nr. 2	10000010111010	0010000	1001	EVENT Nr. 168	497
LAM Nr. 2	1111100000000	0000000	1111	EVENT Nr. 169	498
LAM Nr. 2	0001011001010	0000000	1010	EVENT Nr. 169	499
LAM Nr. 2	1001001001010	1000000	1010	EVENT Nr. 169	500
LAM Nr. 2	0001001001010	0100000	1010	EVENT Nr. 169	501
LAM Nr. 2	0000000000000	1100000	1010	EVENT Nr. 169	502
LAM Nr. 2	00000010111010	0010000	1011	EVENT Nr. 169	503
LAM Nr. 2	1111100000000	0000000	1101	EVENT Nr. 170	504
LAM Nr. 2	0001011001010	0000000	1000	EVENT Nr. 170	505
LAM Nr. 2	0001001001010	1000000	1000	EVENT Nr. 170	506
LAM Nr. 2	1110001001010	0100000	1000	EVENT Nr. 170	507
LAM Nr. 2	0000000000000	1100000	1000	EVENT Nr. 170	508
LAM Nr. 2	11111100111010	0010000	1001	EVENT Nr. 170	509


```
c*****
c*
c* TESTPROGRAMM ZUM AUSLESEN DES LABEN ADC-SYSTEMS UEBER CAMAC *
c* UNTER RSX11 AUF EINEM STARBURST // MARIO NEUDOLD // SEP-86 // *
c* *
c* PROGRAMM LIEST DEN PUFFER DES LABEN-SYSTEMS (85 EVENTS, BEI *
c* 5 ADCs) UND SCHREIBT DIE DATEN AUF DEN BILDSCHIRM (ENDLOS - *
c* SCHLEIFE). (ADC-DATEN MIT EVENT- UND ADC-KENNUNG) *
c* PROGRAMMSTOP MIT: ABO/TASK taskname *
c* *
c*****
```

```
PROGRAM LAB3 !Auslesen des LABEN ADC-SYSTEMS

external LABLES !Subroutine lesen des Puffers nach LAM
external DEZDUA !Subroutine zum Typwandeln

INTEGER*4 M, IDATA(0:509) !INT. fuer ADC - Daten
INTEGER INTA(2) !Feld fuer CCLNK
INTEGER IEXT(2) !Feld fuer CDREG
DIMENSION IARR(24) !Fuer 24 Bit ADC-Wert

COMMON /SUB/ IDATA, IEXT, I, NOFLAM, ILAM

J=0 !LAM-Zaehler
JJ=0 !EVENT-Zaehler
```

c- - - - INITIAISIERUNG DES STARBURST, SOWIE DES LABEN SYSTEMS

```
NOFACC=23 !Position des Starburst
call CDSET(IX,NOFACC) !Initialisieren des Starburst
NOFLAM=11 !Position des LABEN-Interface

call CDREG(IEXT,0,0,NOFLAM,0) !Setzen von IEXT
call CCCC(IEXT) !Clear Dataway
call CCCZ(IEXT) !Initialize Dataway

INTA(1)=NOFLAM !LAM Quelle (= Slotnummer)
INTA(2)=0
call CDLAM(ILAM,0,0,NOFLAM,0,INTA) !Setzen von ILAM
call CCLNK(ILAM,LABLES) !Zuweisung LAM -> Subroutine
call CLREF(9) !Clear Flagge (9)
```

c- - - - BEGINN DER AUSLESESCHLEIFE

```
10 continue !Anfang der Ausleseschleife
```

c- - - - EINSTELLUNG DES LABEN-SYSTEMS (normalerweise nicht notwendig!)

```
M=83888639
call CDREG(IEXT,0,0,NOFLAM,0) !Schreiben
call CFSA(17,IEXT,M,IQ)
M=31
call CDREG(IEXT,0,0,NOFLAM,1) !der
```

```
call CFSA(17,IEXT,M,IQ)
M=1028
call CDREG(IEXT,0,0,NOFLAM,2) !Einstellungen
call CFSA(17,IEXT,M,IQ)
M=62
call CDREG(IEXT,0,0,NOFLAM,3) !des
call CFSA(17,IEXT,M,IQ)
M=574
call CDREG(IEXT,0,0,NOFLAM,4) !LABEN
call CFSA(17,IEXT,M,IQ)
M=1086
call CDREG(IEXT,0,0,NOFLAM,5) !ADC-Systems
call CFSA(17,IEXT,M,IQ)

call CDREG(IEXT,0,0,NOFLAM,1) !LAM enable
call CFSA(26,IEXT,0,IQ)

call WAITFR(9)           !Warten auf die Subroutine (Flagge (9))
call CLREF(9)           !Clear Flagge (9)

J=J+1
write(6,70)

do 20 NN=0,504,6          !Ausgabe
JJ=JJ+1
do 30 N=0,5              !der
K=N+NN
call DEZDUA (IDATA,K,IARR) !ADC-Werte
call WAITFR(9)          !Warten auf die Subroutine (Flagge (9))
write(6,40)J, (IARR(I),I=1,13), (IARR(I),I=14,20), (IARR(I),I=21,24),JJ,K
call CLREF(9)           !Clear Flagge (9)
30 continue
write(6,*)
20 continue

40 format(1X,'LAM Nr.',I2,5X,13I1,2X,7I1,2X,4I1,4X,'EVENT Nr.',I3,4X,I3)
70 format(/(1X,' ADC - WERTE')/)

call CCRGL(ILAM)        !Reconnect LAM

goto 10

END
```

C*****

subroutine LABLES

COMMON /SUB/ IDATA,IEXT,I,NOFLAM,ILAM

INTEGER*4 IDATA(0:509) !INT. fuer ADC - Daten
INTEGER IEXT(2) !Feld fuer CDREG

call CDREG(IEXT,0,0,NOFLAM,1) !Disable LAM
call CFSA(24,IEXT,0,IQ)

do 50 II=0,509 !Lesen

call CDREG(IEXT,0,0,NOFLAM,1)
call CFSA(2,IEXT,IDATA(II),IQ) !der

50 continue !ADCs

call SETEF(9) !Setzen der Flagge (9)
return

end

C*****

subroutine DEZDUA (IDATA,K,IARR)

INTEGER*4 IDATA(0:509),IHILF !INT. fuer ADC - Daten

DIMENSION IARR(24) !Fuer 24 Bit ADC - Daten

IHILF=IDATA(K)

do 60 L=1,24 !Wandelung
IARR(L)=JMOD(IHILF,2)
IHILF=IHILF/2

60 continue !Dezimal -> Dual

call SETEF(9) !Setzen der Flagge (9)
return

end

```
c*****
c*
c* TESTPROGRAMM ZUM AUSLESEN DES LE CROY 12 - FACH ADC UEBER CAMAC *
c* UNTER RSX11 AUF EINEM STARBURST. // MARIO NEUDOLD // SEP-86 // *
c* PROGRAMM LIEST DAS MODUL 10 MAL (10 LAMs) AUS UND GIBT DIE *
c* DATEN AUF DEM BILDSCHIRM AUS. *
c* (ADC-DATEN DEZIMAL bzw. BINAER, MIT LAM- UND ADC-KENNUNG) *
c*
c*****
```

```
PROGRAM LEC                                !Auslesen des LE CROY 12-fach ADC

external LECLES                             !Subroutine lesen nach LAM
external DEZDUA                             !Subroutine zum Typwandeln

INTEGER IDATA(0:10,0:11)                   !INT. fuer ADC - Daten
INTEGER INTA(2)                            !Feld fuer CCLNK
INTEGER IEXT(2)                             !Feld fuer CDREG
DIMENSION IARR(11)                         !Fuer 11 Bit ADC-Wert

COMMON /SUB/ IDATA,IEXT,I,NOFLAM
```

c- - - - INITIALISIERUNG DES STARBURST UND DES ADC-MODULS

```
NOFACC=23                                  !Position des Starburst
call CDSET(IX,NOFACC)                      !Initialisierung des Starburst
NOFLAM=3                                   !Position des ADC-Moduls

call CDREG(IEXT,0,0,NOFLAM,0)             !Setzen von IEXT
call CCCC(IEXT)                            !Clear Dataway
call CCCZ(IEXT)                            !Initialize Dataway

INTA(1)=NOFLAM                             !LAM Quelle (= Slotnummer)
INTA(2)=0
call CDLAM(ILAM,0,0,NOFLAM,0,INTA)        !Setzen von ILAM
call CCLNK(ILAM,LECLES)                   !Zuweisung LAM -> Subroutine
call CLREF(9)                              !Clear Flagge (9)
```

c- - - - SCHLEIFE FUER 10 LAMs

```
do 10 I=1,10                               !Feld fuer 10 LAMs...

call CCLM(ILAM,.true.)                    !LAM enable
call WAITFR(9)                             !Warten auf die Subroutine (Flagge (9))
call CLREF(9)                              !Clear Flagge (9)
call CCRGL(ILAM)                          !Reconnect LAM
```

10 continue

c- - - - ENDE DER SCHLEIFE UND AUSGABE DER MESSWERTE

```
call CLREF(9)                              !Clear Flagge (9)
call CDZEL                                 !Loeschen aller LAM-Eintraege

write(6,1010)
```

```
do 20 J=1,10                !Ausgabe
do 30 K=0,11
call DEZDUA (IDATA,J,K,IARR) !der Messwerte

call WAITFR(9)              !Warten auf die Subroutine (Flagge (9))

write(6,1020)J,(IARR(I),I=1,11),IDATA(J,K),K

call CLREF(9)                !Clear Flagge (9)

30  continue
write(6,*)
20  continue

1010 format(/(' ADC - WERTE (BINAER/DEZIMAL)'))/
1020 format(1X,'LAM Nr.',I2,5X,11I1,3X,I5,4X,'ADC(',I2,')')
```

STOP
END

C*****

```
subroutine LECLES

COMMON /SUB/ IDATA,IEXT,I,NOFLAM

INTEGER IDATA(0:10,0:11)    !INT. fuer ADC - Daten
INTEGER IEXT(2)             !Feld fuer CDREG

do 50 II=0,11              !Lesen

call CDREG(IEXT,0,0,NOFLAM,II)
call CFSA(2,IEXT,IDATA(I,II),IQ) !der

50  continue                !ADCs

call SETEF(9)              !Setzen der Flagge (9)
return

end
```

C*****

```
subroutine DEZDUA (IDATA,J,K,IARR)

INTEGER IDATA(0:10,0:11)    !INT. fuer ADC - Daten
DIMENSION IARR(11)         !Fuer 11 Bit ADC - Daten

IHILF=IDATA(J,K)

do 60 L=1,11              !Wandelung
IARR(L)=IMOD(IHILF,2)
IHILF=IHILF/2

60  continue                !Dezimal -> Dual

call SETEF(9)              !Setzen der Flagge (9)
return

end
```

```
C*****  
C*  
C* TESTPROGRAMM ZUM AUSLESEN DES LE CROY 12 - FACH ADC UEBER CAMAC *  
C* UNTER RSX11 AUF EINEM STARBURST // MARIO NEUDOLD // SEP-86 // *  
C* PROGRAMM LIEST DAS MODUL 10 MAL (10LAMs) AUS UND SCHREIBT DIE *  
C* DATEN IN EIN FILE (LEC.DAT). *  
C* ANZEIGEN DER DATEN DURCH DIE EINGABE: TYPE LEC.DAT *  
C*  
C*****
```

```
PROGRAM LEC2 !Auslesen des LE CROY - ADC  
  
external LECLES !Subroutine lesen nach LAM  
external DEZDUA !Subroutine zum Typwandeln  
  
INTEGER IDATA(0:10,0:11) !INT. fuer ADC - Daten  
INTEGER INTA(2) !Feld fuer CCLNK  
INTEGER IEXT(2) !Feld fuer CDREG  
DIMENSION IARR(11) !Fuer 11 Bit ADC-Wert  
  
COMMON /SUB/ IDATA,IEXT,I,NOFLAM  
  
open(UNIT=1,FILE='DL2:¢3,3!LEC2.DAT',STATUS='UNKNOWN') !Oeffnen des  
c !Datenfiles  
c- - - - INITIALISIERUNG VON STARBURST UND LE CROY-CAMAC-MODUL  
  
NOFACC=23 !Position des Starburst  
call CDSET(IX,NOFACC) !Initialisieren des Starburst  
NOFLAM=3 !Position des ADC  
  
call CDREG(IEXT,0,0,NOFLAM,0) !Setzen von IEXT  
call CCCC(IEXT) !Clear Dataway  
call CCCZ(IEXT) !Initialize Dataway  
  
INTA(1)=NOFLAM !LAM Quelle (= Slotnummer)  
INTA(2)=0  
call CDLAM(ILAM,0,0,NOFLAM,0,INTA) !Setzen von ILAM  
call CCLNK(ILAM,LECLES) !Zuweisung LAM->Subroutine  
call CLREF(9) !Clear Flagge (9)  
  
c- - - - BEGINN DER AUSLESESCHLEIFE  
  
do 10 I=1,10 !Feld fuer 10 LAMs...  
  
call CCLM(ILAM,.true.) !LAM enable  
  
call WAITFR(9) !Warten auf die Subroutine (Flagge (9))  
call CLREF(9) !Clear Flagge (9)  
call CCRGL(ILAM) !Reconnect LAM  
  
10 continue
```

c- - - - ENDE DER AUSLESESCHLEIFE / SCHREIBEN DER DATEN INS DATENFILE

```
call CLREF(9)           !Clear Flagge (9)
call CDZEL              !Loeschen aller LAM-Eintraege

write(1,1010)

do 20 J=1,10           !Ausgabe
do 30 K=0,11           !der
call DEZDUA (IDATA,J,K,IARR) !Messwerte
call WAITFR(9)         !Warten auf die Subroutine (Flagge (9))
write(1,1020)J,(IARR(I),I=1,11),IDATA(J,K),K
call CLREF(9)         !Clear Flagge (9)
30 continue
write(1,*)
20 continue

1010 format(/(1X,' ADC - WERTE (BINAER/DEZIMAL)')/)
1020 format(1X,' LAM Nr. ',I2,5X,11I1,3X,I5,4X,' ADC(',I2,')')

close(1)

STOP

END
```

C*****

subroutine LECLES

COMMON /SUB/IDATA,IEXT,I,NOFLAM

INTEGER IDATA(0:10,0:11) !INT. fuer ADC - Daten
INTEGER IEXT(2) !Feld fuer CDREG

do 50 II=0,11 !Lesen

call CDREG(IEXT,0,0,NOFLAM,II)
call CFSA(2,IEXT,IDATA(I,II),IQ) !der

50 continue !ADCs

call SETEF(9) !Setzen der Flagge (9)
return

end

C*****

subroutine DEZDUA (IDATA,J,K,IARR)

INTEGER IDATA(0:10,0:11) !INT. fuer ADC - Daten

DIMENSION IARR(11) !Fuer 11 Bit ADC - Daten

IHILF=IDATA(J,K)

do 60 L=1,11 !Wandelung

IARR(L)=IMOD(IHILF,2)

IHILF=IHILF/2

60 continue !Dezimal -> Dual

call SETEF(9) !Setzen der Flagge (9)
return

end

ADC - WERTE (BINAER/DEZIMAL)

LAM Nr. 1	01000001001	1154	ADC(0)
LAM Nr. 1	11101000000	23	ADC(1)
LAM Nr. 1	00011000000	24	ADC(2)
LAM Nr. 1	11010001001	1163	ADC(3)
LAM Nr. 1	00011000000	24	ADC(4)
LAM Nr. 1	00111000000	28	ADC(5)
LAM Nr. 1	11001001001	1171	ADC(6)
LAM Nr. 1	01011000000	26	ADC(7)
LAM Nr. 1	01001000000	18	ADC(8)
LAM Nr. 1	00101000000	20	ADC(9)
LAM Nr. 1	00000001001	1152	ADC(10)
LAM Nr. 1	11001000000	19	ADC(11)
LAM Nr. 2	11111110001	1151	ADC(0)
LAM Nr. 2	11101000000	23	ADC(1)
LAM Nr. 2	11101000000	23	ADC(2)
LAM Nr. 2	10010001001	1161	ADC(3)
LAM Nr. 2	00011000000	24	ADC(4)
LAM Nr. 2	00111000000	28	ADC(5)
LAM Nr. 2	00001001001	1168	ADC(6)
LAM Nr. 2	01011000000	26	ADC(7)
LAM Nr. 2	01001000000	18	ADC(8)
LAM Nr. 2	00101000000	20	ADC(9)
LAM Nr. 2	01111110001	1150	ADC(10)
LAM Nr. 2	11001000000	19	ADC(11)
LAM Nr. 3	11000001001	1155	ADC(0)
LAM Nr. 3	00011000000	24	ADC(1)
LAM Nr. 3	00011000000	24	ADC(2)
LAM Nr. 3	00110001001	1164	ADC(3)
LAM Nr. 3	00011000000	24	ADC(4)
LAM Nr. 3	00111000000	28	ADC(5)
LAM Nr. 3	00101001001	1172	ADC(6)
LAM Nr. 3	01011000000	26	ADC(7)
LAM Nr. 3	11001000000	19	ADC(8)
LAM Nr. 3	10101000000	21	ADC(9)
LAM Nr. 3	10000001001	1153	ADC(10)
LAM Nr. 3	11001000000	19	ADC(11)

.... bis

LAM Nr.10	00000001001	1152	ADC(0)
LAM Nr.10	11101000000	23	ADC(1)
LAM Nr.10	11101000000	23	ADC(2)
LAM Nr.10	10010001001	1161	ADC(3)
LAM Nr.10	00011000000	24	ADC(4)
LAM Nr.10	00111000000	28	ADC(5)
LAM Nr.10	00001001001	1168	ADC(6)
LAM Nr.10	01011000000	26	ADC(7)
LAM Nr.10	01001000000	18	ADC(8)
LAM Nr.10	00101000000	20	ADC(9)
LAM Nr.10	01111110001	1150	ADC(10)
LAM Nr.10	11001000000	19	ADC(11)

```
C*****
C*
C* TESTPROGRAMM ZUM AUSLESEN DES LE CROY 12 - FACH ADC UEBER CAMAC *
C* UNTER RSX11 AUF EINEM STARBURST // MARIO NEUDOLD // SEP-86 // *
C* NACH DEN STARTEN DER TASK BEFINDET SICH DAS PROGRAMM IN EINER *
C* ENDLOSSCHLEIFE UND SCHREIBT DIE ADC-DATEN AUF DEN BILDSCHIRM *
C* PROGRAMMSTOP : ABO/TASK taskname *
C*
```

```
PROGRAM LEC3 !Auslesen des LE CROY - ADC und
c !Wandelung der Messdaten

external LECLES !Subroutine lesen nach LAM

INTEGER IDATA !INT. fuer ADC - Daten
INTEGER INTA(2) !Feld fuer CCLNK
INTEGER IEXT(2) !Feld fuer CDREG

COMMON /SUB/ IEXT,NOFLAM
```

c- - - - INITIALISIEREN DES STARBURST UND DES LE CROY-CAMAC-MODULS

```
NOFACC=23 !Position des Starburst

call CDSET(IX,NOFACC) !Initialisieren des Starburst
NOFLAM=3 !Position des ADC

call CDREG(IEXT,0,0,NOFLAM,0) !Setzen von IEXT
call CCCC(IEXT) !Clear Dataway
call CCCZ(IEXT) !Initialize Dataway

INTA(1)=NOFLAM !LAM Quelle (= Slotnummer)
INTA(2)=0
call CDLAM(ILAM,0,0,NOFLAM,0,INTA) !Setzen von ILAM
call CCLNK(ILAM,LECLES) !Zuweisung LAM-Subroutine
call CLREF(9) !Clear Flagge (9)
```

c- - - - BEGINN DER ENDLOSSCHLEIFE ZUM AUS LESEN DER ADCs

```
10 call CCLM(ILAM,.true.) !LAM enable
call WAITFR(9) !Warten auf die Subroutine (Flagge (9))
call CLREF(9) !Clear Flagge (9)
call CCRGL(ILAM) !Reconnect LAM
goto 10

END
```

C*****

subroutine LECLES

COMMON /SUB/ IEXT,NOFLAM

INTEGER IDATA

!INT. fuer ADC - Daten

INTEGR IEXT(2)

!Feld fuer CDREG

DIMENSION IARR(11)

!Fuer 11 Bit ADC - Daten

do 20 II=0,11

!Lesen

call CDREG(IEXT,0,0,NOFLAM,II)

!der

call CFSA(2,IEXT,IDATA,IQ)

!Messdaten und

IHILF=IDATA

do 30 L=1,11

!Wandelung

IARR(L)=IMOD(IHILF,2)

IHILF=IHILF/2

30

continue

!Dezimal -> Dual

write(6,40)(IARR(I),I=1,11),IDATA,II

20

continue

write(6,50)

call SETEF(9)

!Setzen der Flagge (9)

40

format(11X,11I1,3X,I5,4X,'ADC(',I2,')')

50

format(6X,'-----')

return

end

```
C*****
C*
C* PROGRAMM ZUM AUSLESEN DES LABEN ADC-SYSTEMS UND DES LE CROY ADCs UEBER *
C* CAMAC, UNTER RSX11 AUF EINEM STARBURST // MARIO NEUDOLD // SEP-86 // *
C* *
C* PROGRAMM LIEST KOINZIDENTE EVENTS AUS DEN ADCs AUS, UNTER VERWENDUNG *
C* DER BEIDEN UNTERSCHIEDLICHEN ADC-TYPEN. *
C* IN EINER SPEZIELLEN KONFIGURATION BEI VERWENDUNG VON 5 LABEN-ADCs. *
C* DAS BEDEUTET 85 MALIGES AUSLESEN DES LE CROY ADCs, BEI EINMALIGEM *
C* AUSLESEN DES EVENT-PUFFERS IN LABEN-SYSTEM (512 SPEICHERPLAETZE => *
C* 85 EVENTS BEI 5 ADCs) UND SCHREIBT DIE DATEN IN EIN FILE. *
C* *
C*****
```

```
c PROGRAM LABLEC !Auslesen des LABEN ADC-SYSTEMS und
!des LE CROY 12-fach ADCs

external LABLES !Subroutine, lesen des Puffers nach LAM
external LECLES !Subroutine, lesen des LE CROY ADCs

INTEGER*4 IDLAB(0:509) !INT.*4 fuer ADC-Daten (LABEN, 24 BIT)
INTEGER INTA(2) !Feld fuer CCLNK
INTEGER IEXA(2) !Feld fuer CDREG

INTEGER IDLEC(0:11) !Feld fuer ADC-Daten (LE CROY, 11 BIT)
INTEGER INTB(2) !Feld fuer CCLNK
INTEGER IEXB(2) !Feld fuer CDREG

COMMON /LAB/ IDLAB,NEND,NOFLAB,IFLAG
COMMON /LEC/ IDLEC,NOFLEC

NEND=2 !Anzahl der Auslesungen des Eventpuffers
IFLAG=0 !Loeschen der Verzweigungs-Flagge
```

c- - - INITIALISIERUNGEN ZUM ARBEITEN MIT DEM STARBURST (+ CAMAC-LIBRARY)

```
NOFACC=23 !Position des Starburst
call CDSET(IX,NOFACC) !Initialisierung des Starburst
```

c- - - - INITIALISIERUNGEN FUER DAS LABEN INTERFACE 5911

```
NOFLAB=11 !Position des LABEN-Interface
call CDREG(IEXA,0,0,NOFLAB,0) !Setzen von IEXA (LABEN)
call CCCC(IEXA) !Clear Dataway
call CCCZ(IEXA) !Initialize Dataway
```

```
INTA(1)=NOFLAB !LAM Quelle (= Slotnummer)
INTA(2)=0
call CDLAM(ILAB,0,0,NOFLAB,0,INTB) !Setzen von ILAB
call CCLNK(ILAB,LABLES) !Zuweisung LAM -> Subroutine
```

c- - - - INITIALISIERUNGEN FUER DAS LECROY ADC-MODUL

```
NOFLEC=3                                !Position des LE CROY ADC-Moduls
call CDREG(IEXB,0,0,NOFLEC,0)           !Setzen von IEXB (LE CROY)
call CCCC(IEXB)                          !Clear Dataway
call CCCZ(IEXB)                          !Initialize Dataway

INTB(1)=NOFLEC                          !LAM Quelle (= Slotnummer)
INTB(2)=0
call CDLAM(ILEC,0,0,NOFLEC,0,INTB)      !Setzen von ILEC
call CCLNK(ILEC,LECLCS)                 !Zuweisung LAM -> Subroutine

call CLREF(9)                            !Clear Flagge (9)

c- - - - EIGENTLICHE AUSLESESCHLEIFE

open(UNIT=1,FILE='DL2:c3,3!COMB.DAT',STATUS='UNKNOWN') !Oeffnen des
c                                          !Daten-Files

10    call CDREG(IEXA,0,0,NOFLAB,1)      !LAM enable LABEN
      call CFSA(26,IEXA,0,IQ)
      if(IFLAG.EQ.1) goto 40

30    call CCLM(ILEC,.true.)             !LAM enable LE CROY

40    IFLAG=0                            !Loeschen der Verzweigungs-Flag
      call WAITFR(9)                     !Warten auf die Subroutine (Flag(9))
      call CLREF(9)                       !Clear Flagge (9)

      if(IEEND.EQ.NEND) goto 50           !Abbruchbedingung

      if(IFLAG.EQ.1) goto 20

      IEV1=IEV1+1                         !Wegschreiben
      write(1,1010)IEV1
      do 60 I=0,11                         !der
      write(1,1020)IDLEC(I)
60    continue                           !Messdaten (LE CROY)

      call CCRGL(ILEC)                    !Reconnect LAM LE CROY
      if(IEEND.NE.NEND) goto 30

      IEV2=IEV2+1                         !Wegschreiben

      write(1,1030)IEV2
      do 70 I=0,509                         !der
      write(1,1040)IDLAB(I)
70    continue                           !Messdaten (LABEN)

20    call CCRGL(ILAB)                    !Reconnect LAM LABEN
      if(IEEND.NE.NEND) goto 10

c- - - - ENDE DER SCHLEIFE UND PROGRAMMENDE

50    close(1)                            !Schliessen des Daten-Files

      call CDREG(IEXA,0,0,NOFLAB,1)      !Disable LAM LABEN
```

```
call CFSA(24, IEXA, 0, IQ)  
call CDZEL
```

```
!Loeschen aller LAM-Eintraege
```

```
1010 format(1X, 'LE CROY EVENT Nr.', I3)  
1020 format(1X, I5)  
1030 format(1X, ' LABEN EVENT Nr.', I3)  
1040 format(1X, I8)
```

```
STOP  
END
```

C*****

```
subroutine LECLES

COMMON /LEC/ IDLEC,NOFLEC

INTEGER IDLEC(0:11)           !INT. fuer ADC - Daten
INTEGER IEXB(2)              !Feld fuer CDREG

do 11 N=0,11                 !Lesen

call CDREG(IEXB,0,0,NOFLEC,N)
call CFSA(2,IEXB,IDLEC(N),IQ) !der ADCs

110 continue                 !ueber CAMAC

call SETEF(9)                !Setzen der Flagge (9)
return
end
```

C*****

```
subroutine LABELS

COMMON /LAB/ IDLAB,NEND,NOFLAB,IFLAG

INTEGER*4 IDLAB(0:509)       !INT.*4 fuer ADC - Daten
INTEGER IEXA(2)              !Feld fuer CDREG

call CCLM(ILEC,.true.)      !LAM enable LE CROY

call CDREG(IEXA,0,0,NOFLAB,1) !Disable LAM LABEN zum vollstaendigen

call CFSA(24,IEXA,0,IQ)     !Auslesen des Puffers

do 210 N=0,509              !Lesen

call CDREG(IEXB,0,0,NOFLAB,1)
call CFSA(2,IEXB,IDLAB(N),IQ) !des Puffers

210 continue                 !ueber CAMAC
IFLAG=1                      !Setzen der Verzweigungs-Flagge
NEND=NEND+1                  !Zaehler fuer Abbruchbedingung

call SETEF(9)                !Setzen der Flagge (9)
return
end
```