



KERNFORSCHUNGSANLAGE JÜLICH GmbH

Zentrallabor für Elektronik

**PDV-CAMAC-Controller
KFA-ZEL-NE 365**

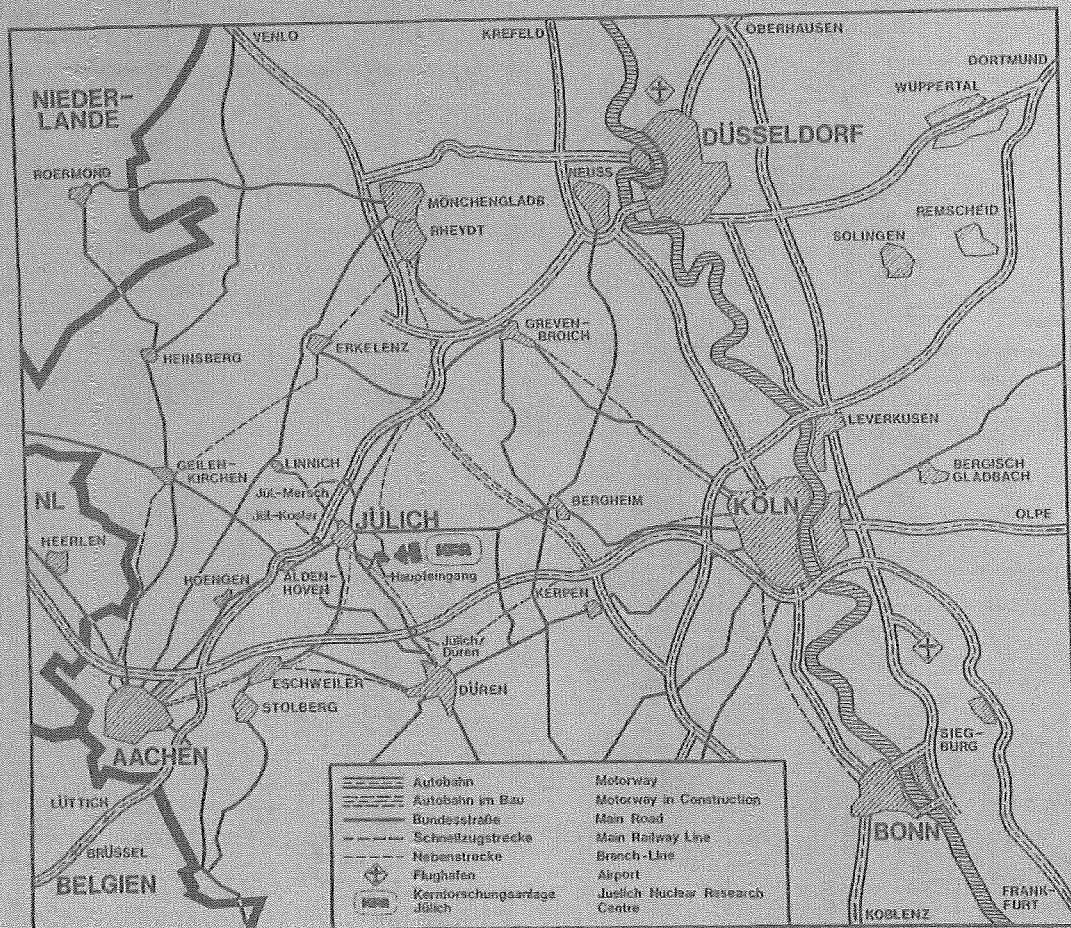
von

M. Sauer, K. Zvoll, R. Becker, B. Becks

Jül - Spez - 27

Januar 1979

ISSN 0343-7639



Als Manuskript gedruckt

Spezielle Berichte der Kernforschungsanlage Jülich – Nr. 27

Zentrallabor für Elektronik Jülich - Spez - 27

Zu beziehen durch: ZENTRALBIBLIOTHEK der Kernforschungsanlage Jülich GmbH,
Jülich, Bundesrepublik Deutschland

PDV-CAMAC-Controller

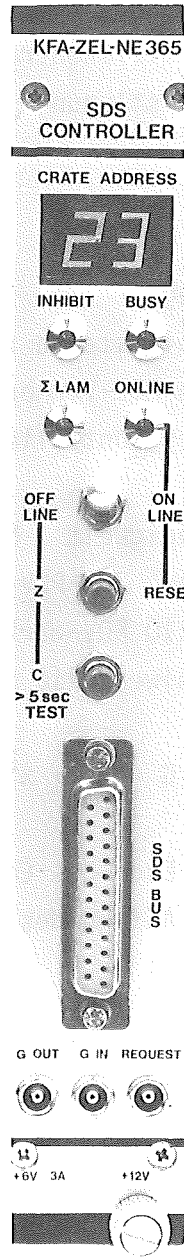
KFA-ZEL-NE 365

von

M. Sauer, K. Zvoll, R. Becker, B. Becks

Dieser Bericht veröffentlicht u. a. Ergebnisse von Forschungsvorhaben, die vom Bundesministerium für Forschung und Technologie im Rahmen des 2. DV-Programms (Projekt Prozeßlenkung mit DV-Anlagen, KFK Karlsruhe) gefördert wurden.

Kurzbeschreibung des SDS Controllers KFA-ZEL-NE 365



Der SDS Controller ermöglicht den Anschluß eines CAMAC Crates an ein PDV-Bus-Übertragungssystem. Das CAMAC Crate übernimmt damit die Funktion einer Übertragungssteuer-einheit. Der Anschluß an den PDV-Bus geschieht über die an der Front-seite des Controllers angeordnete SDS-Anschlußbuchse. Der SDS-Controller kann dazu an beliebiger Steckposition des CAMAC Crates untergebracht sein und gehorcht den ACB-Empfehlungen. Es sind Block-schreib- und Blockleseoperationen möglich, wobei zwischen einer CAMAC-Datenwortbreite von 16 bzw. 24 bit gewählt werden kann. Bei fehlerhaften Übertragungen ermöglicht die im Controller implementierte Software die Wiederholung der gesamten Nach-richt oder die teilweise Übertragung ab erkanntem Fehlerwort. Die ebenfalls implementierte PDV-Bus Prozedur der zyklischen Kurzabfrage ermöglicht eine Alarmabgabe, wobei als Alarmquellen alle gesetzten LAM-Signale herange-zogen werden.

Ein in der Controller-Software ent-haltenes Testprogramm ermöglicht die Überprüfung aller RAM-Speicherbau-steine sowie der Schreib- und Lese-register des CAMAC-Datenweges. Der Controller arbeitet dabei in Verbind-ung mit einem Wortgenerator - Modul als manueller Crate-Controller.

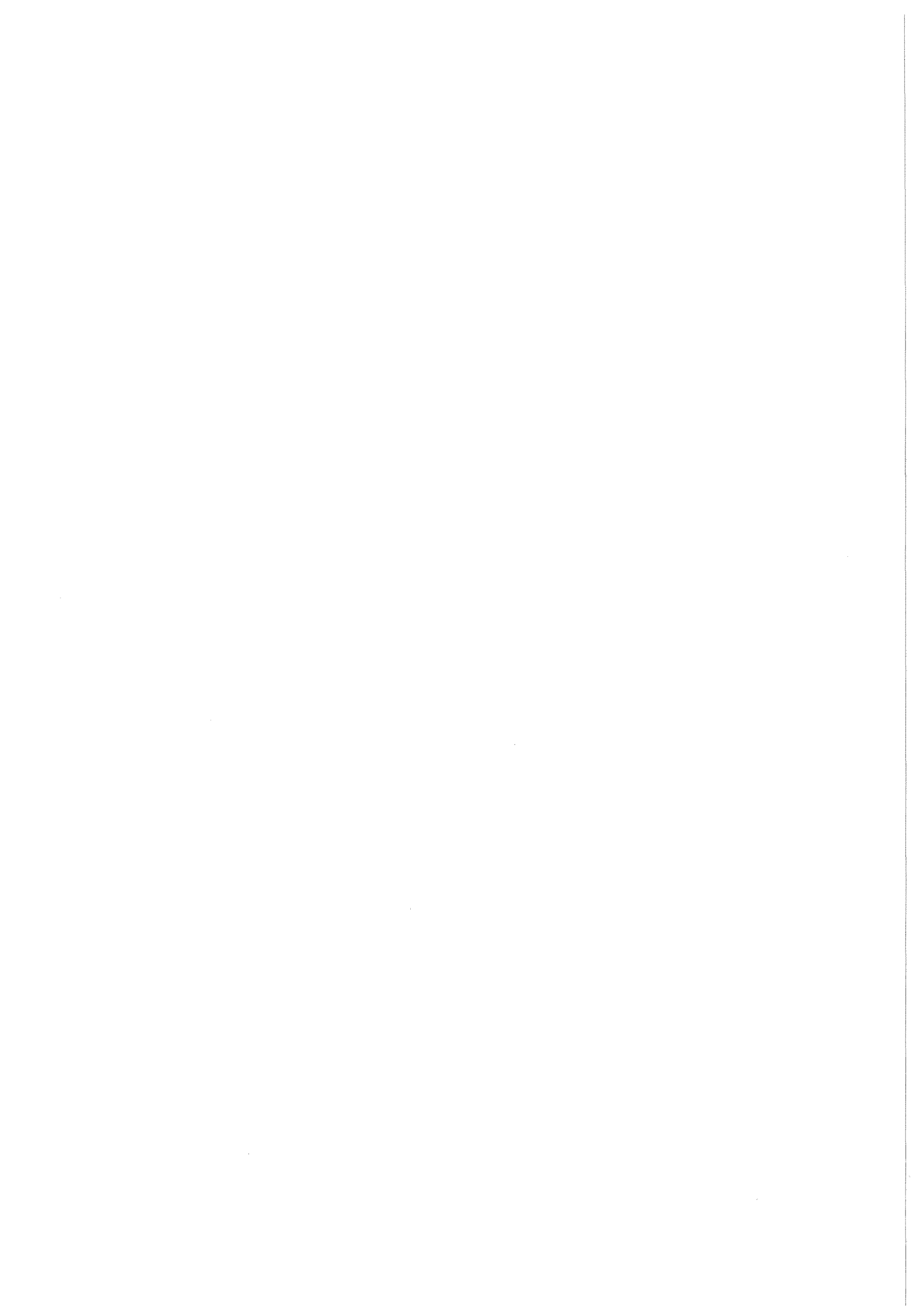
Spezifikationen des SDS Controllers KFA-ZEL-NE 365

Anschluß an:	PDV-Bus Systeme nach Bericht KFK-PDV 70 und CAMAC-Rahmen
Funktion:	Arbeitet als Haupt- oder Hilfskontrolleinheit mit externer Prioritätenfestlegung
Betriebsarten:	ONLINE, OFFLINE, TEST
Schnittstellen:	1) Standard CAMAC-Datenweg 2) Serielle-Digitale-Schnittstelle (SDS) mit Buchsenleiste nach FTZ-Vornorm 386 TV 1 3) ACB Anschluß (<u>A</u> uxiliary <u>C</u> ontroller <u>B</u> us)
Datenübertragung:	Max. Übertragungsgeschwindigkeit 1 Mbit/sec Blocklänge "schreiben" max. 255 PDV-Worte Blocklänge "lesen" max. 110 PDV-Worte CAMAC-Datenwortbreite wahlweise 16 bit oder 24 bit
Datensicherung:	CRC mit Generatorpolynom $G = X^8 + X^2 + X + 1$
Reaktionszeit:	Ca. 20 μ s, gemessen vom Ende des Empfangsrahmen- signales (RS-Signal) bis Anfang des Sendebereit- schaftssignales (SB-Signal)
Programmspeicher:	1 K x 16 bit PROM
Arbeitsspeicher:	256 x 8 bit RAM
Stromversorgung:	+ 6 V, 3,5 A + 24 V, 0,1 A - 24 V, 0,1 A
Modulbreite:	2/25 CAMAC

	Seite
0. Einleitung	1
1. PDV - Bus Konzept	7
2. SDS - Schnittstelle	8
2.1 Physikalische Beschreibung	8
2.2 Signalstruktur	8
3. PDV - Prozeduren	12
3.1 Bedeutung der einzelnen Bitstellen einer Nachricht	13
3.2 Übertragungsoperationen	18
3.3 Übertragungssicherung	21
3.4 Fehlerbehandlung	22
4. Protokoll für CAMAC - Prozeduren	23
4.1 Globaler Crate - Befehl	23
4.2 Datenloser CAMAC - Befehl	25
4.3 CAMAC - Befehl mit Funktion "Daten schreiben"	27
4.4 CAMAC - Befehl mit Funktion "Daten lesen"	29
4.5 Datenloser CAMAC - Befehl mit CAMAC - Funktion "Daten lesen"	31
4.6 CAMAC - Befehl mit Funktion "Daten schreiben und Daten lesen"	33
5. Spezielle PDV - Funktionen	35
5.1 Funktion "Wiederholen Schreiben"	35
5.2 Funktion "Wiederholen Lesen"	36
5.3 Zyklische Kurzabfrage	37
5.4 Rückübertragener Status	39
6. Spezielle CAMAC - Crate - Befehle	41
6.1 Statusabfrage	41
6.2 Read LAM - Pattern	45
6.3 Write Status	47
6.4 Selective Set Status	48
6.5 Selective Clear Status	49
6.6 Test INHIBIT	50

	Seite
7. Hardware - Beschreibung	51
7.1 Aufgaben des Controllers	51
7.2 Lösung mit Mikroprozessor	52
7.3 Auswahlkriterien für 8 X 300	53
7.4 Aufgaben des Mikroprozessors	57
8. Blockschaltbild	58
8.1 Prozessorbaugruppen	60
8.2 Anschluß an SDS - Schnittstelle	62
8.3 Anschluß an CAMAC - Datenweg	64
8.3.1 Read - Register	64
8.3.2 Write - Register	65
8.3.3 LAM - Register	65
8.3.4 N - Register und Decoder	68
8.3.5 A und F Register und Treiber	70
8.3.6 Generator für CAMAC - Zyklus	70
8.3.7 Funktionsbeschreibung des ACB - Anschlußes	75
8.4 Eingangsschaltung der manuellen Bedienung	81
8.5 Crate - Adressregister und Anzeige	83
8.6 Hardware - Statusregister	85
8.7 RAM - Arbeitsspeicher	86
8.8 IV - Byte mit Steuersignaleingang und -ausgang	90
9. Mechanischer Aufbau	96
9.1 Frontplatte und Bedienungselemente	100

	Seite	
10.	Programmbeschreibung	102
10.1	Programmstruktur	102
10.2	Programmsegmentaufteilung	105
10.2.1	Programmsegment ANFANG	107
10.2.2	Programmsegment MAINLP	111
10.2.3	CRC - Kontrolle und CRC - Generierung	117
10.2.4	Programmsegment FALSE	121
10.2.5	Programmsegment WORD0	124
10.2.6	Programmsegment WORD1	127
10.2.7	Programmsegment WORD2	130
10.2.8	Programmsegment WORD3	132
10.2.9	Programmsegment WORDN	136
10.2.10	Programmsegment PRUEF	139
10.2.11	Programmsegment CYCLE	141
10.2.12	Programmsegment LOESCH	144
10.2.13	Programmsegment CRTFKT	147
10.2.14	Programmsegment NIST30	151
10.2.15	Programmsegment CONTRL	154
10.2.16	Programmsegment CHGINH	162
10.2.17	Programmsegment SAMMEL	164
10.2.18	Programmsegment KURZAB	167
10.2.19	Programmsegment STATUS	172
10.2.20	Programmsegment RESTAT	177
10.2.21	Programmsegment OUTPUT	180
10.2.22	Programmsegment NXTWRD	187
10.2.23	Programmsegment TEST	190
11.	Literaturverzeichnis	201



0. Einleitung

Rechner zur Steuerung von industriellen Prozessen und zugehörige geeignete Datenübertragungssysteme sind seit vielen Jahren erfolgreich im Einsatz. In jüngster Zeit kommen zur Steuerung räumlich getrennter und weit auseinanderliegender Prozesse aus wirtschaftlichen Gründen zunehmend Bussysteme mit serieller Datenübertragung zur Anwendung. Im PDV-Arbeitskreis TP 30 wurde ein Prozeßdatenübertragungssystem entworfen, das für alle industriellen Anwendungen geeignet erscheint, bei denen große Entfernungen zu überbrücken sind, Echtzeitbedingungen einzuhalten sind und hohe Betriebssicherheit wichtiger ist als hohe Übertragungsraten /1/.

Bei der Festlegung einer Leitungsstruktur spielen hohe Betriebssicherheit und die Notwendigkeit einer leistungsfähigen Alarmerkennung eine ausschlaggebende Rolle.

Eine hohe Betriebssicherheit wird u. a. erreicht, wenn eine von keinem Signalverstärker unterbrochene Leitung, d. h. eine bidirektionale Leitung, verwendet wird. Wird diese Leitung an der aktiven Steuerstation zu einem Ring geschlossen, besteht die Möglichkeit, Leitungsunterbrechungen zu erkennen und die Nachricht auf Übertragungsfehler zu überprüfen.

Eine mögliche Systemvariante liegt in der Verwendung von zwei getrennten Leitungen (siehe Abb. 0-1). Dabei dient die eine Leitung der Übertragung von Nachrichten (Steuerstation - Unterstation - Steuerstation), die andere der Übertragung von Antworten (Unterstation - Steuerstation) und Alarmmeldungen, wodurch insbesondere eine effektive Alarmbehandlung ermöglicht wird. Darüber hinaus bietet eine Doppelleitung Vorteile für rückwirkungsfreie Ankopplungen über Transformatoren.

Als Systemvariante für größere Entfernungen wird eine unidirektionale Einfachringleitung empfohlen, die durch das Einschalten von Bussignalregeneratoren realisiert werden kann - eine Maßnahme, die für große Entfernungen erforderlich wird (s. Abbildung 0-2, vergl. /1/).

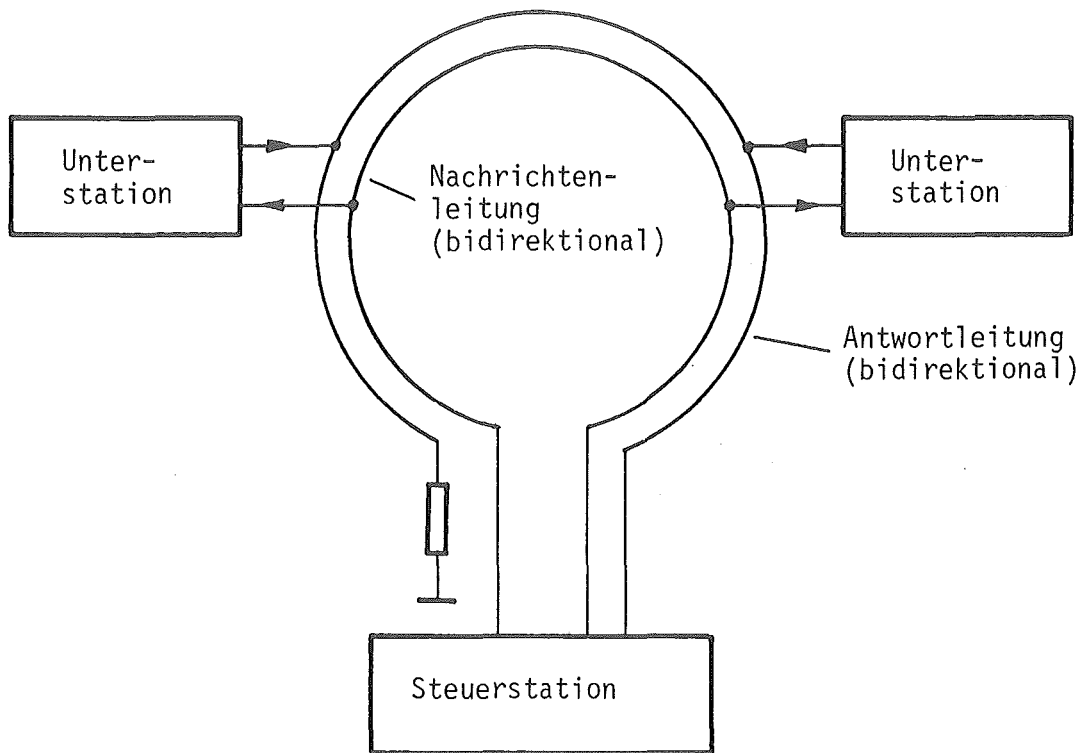


Abb. 0 - 1

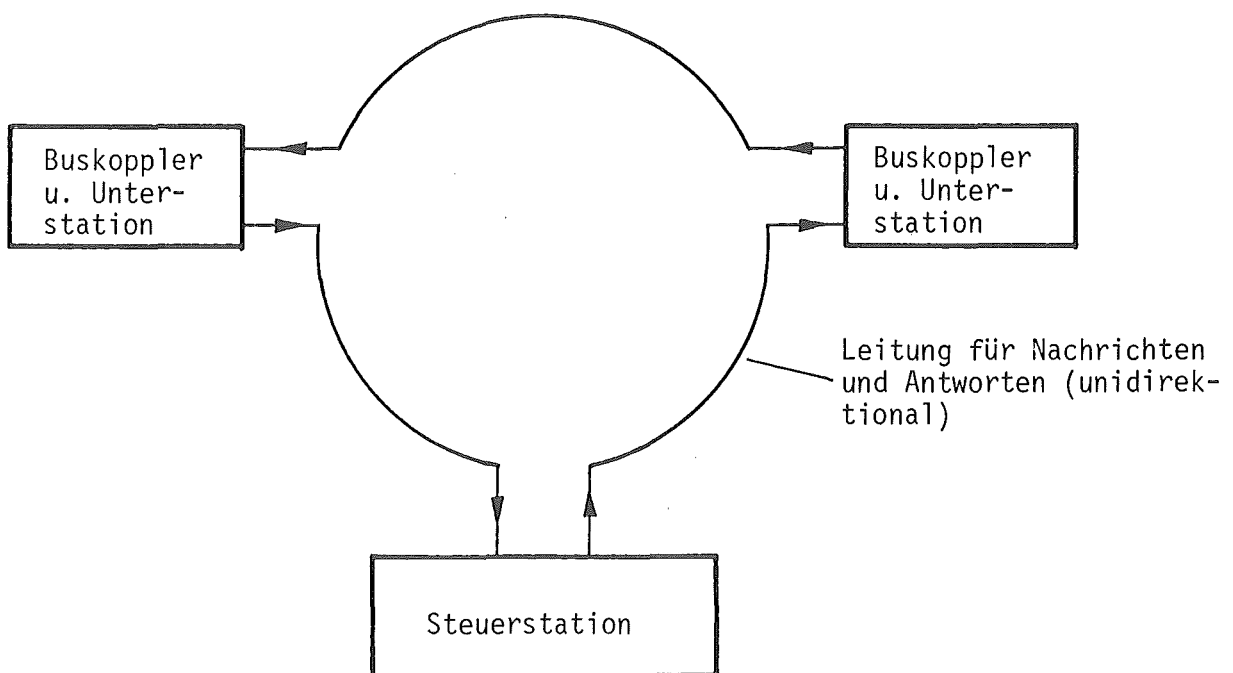


Abb. 0 - 2

Neben der Ringstruktur ist für besondere Anwendungsfälle die offene Doppel- und Einleitungsstruktur denkbar (siehe Abbildungen 0 - 3 und 0 - 4). Dabei muß die Einleitungsvariante immer mit bidirektionalen Leitungen realisiert werden.

Die rückwirkungsfreie Ankopplung der Stationen über Trafos kann bei den offenen Systemen in der gleichen Weise erfolgen wie bei der Doppelringleitung.

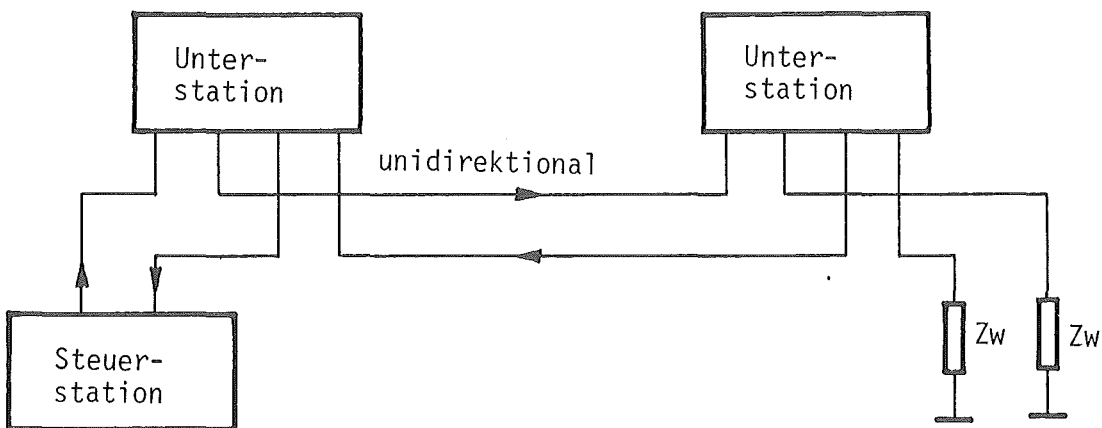
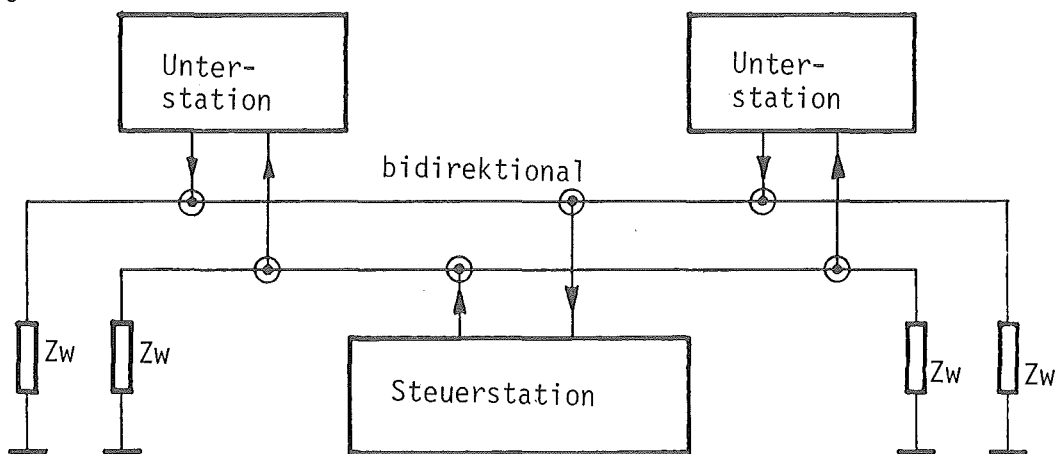


Abb. 0 - 3

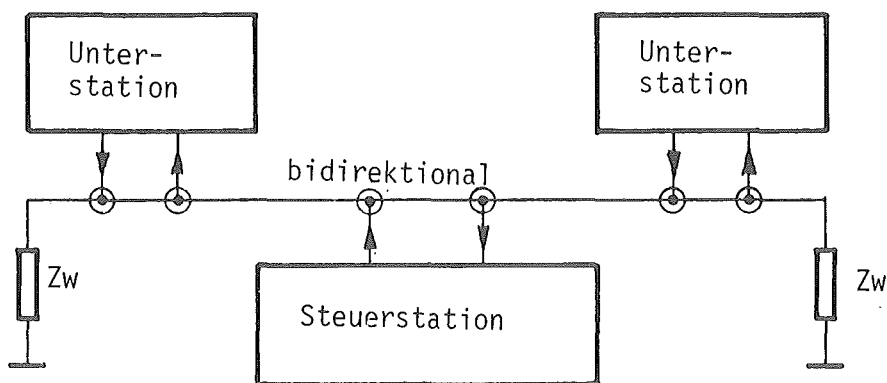


Abb. 0 - 4

Ein bereits weit verbreitetes System der Datenverarbeitung für Anwendungen im "On - Line" Betrieb mit digitalen Steuereinheiten oder Rechnern ist das von europäischen Laboratorien im ESONE - Komitee festgelegte und standardisierte "C A M A C"- System. Als Hauptmerkmal erhält dieses CAMAC - System einen Datenweg und ist bei allen Aufgaben der Datenverarbeitung anwendbar, die eine digitale Steuereinheit oder einen Rechner verwenden /2/3/.

CAMAC ist ein modulares System. Eine Gerätekombination wird durch Montage von passenden Einschubeinheiten in einem Standardchassis oder Rahmen gebildet. Jede Einheit belegt einen oder mehrere Montage-Stationen im Rahmen. Jede Station ist mit einer 86-poligen Buchsenleiste ausgerüstet, die die Verbindung zum CAMAC - Datenweg herstellt, der ein Teil des Rahmens ist. Der Datenweg besteht hauptsächlich aus Sammelleitungen für Daten-, Steuersignale und Stromversorgung. Der Datenverkehr zwischen den Einschubeinheiten wird über den Datenweg abgewickelt. Dieses passive vielfach-Parallelsystem ist in den Rahmen eingebaut und verbindet die Datenwegbuchsenleisten an allen Stationen.

Die äußerste rechte Station, gesehen von der Vorderseite des Rahmens, hat die besondere Aufgabe einer Steuerstation. Die Datenleitungen im Datenweg sind an den übrigen Normalstationen zugänglich, jedoch nicht an der Steuerstation. Eine Steuerung belegt den Platz für die Steuerung und wenigstens den Platz einer benachbarten Normalstation (Abb. 0 - 5). Alle Datenwegoperationen werden von der Steuereinheit generiert und überwacht. Deshalb wird diese Baugruppe auch als "Crate-Controller" bezeichnet.

Über eine gesondert ausgeführte Steckverbindung (z.B. auf der Frontseite der Einschubeinheit oder über ein auf der Rückseite herausgeführtes Kabel) kann der Crate - Controller mit einem Rechner oder einem übergeordnetem Datenübertragungssystem verbunden werden. Damit ist der Controller das Koppellement zwischen dem parallelen Datenweg des CAMAC - Crates und einem peripheren Übertragungssystem. In dieser Betriebsart erfüllt der Crate - Controller nicht nur die Aufgaben der Organisation des Datenverkehrs im CAMAC - Crate, sondern auch die erforderlichen strukturellen Befehls- und Datentransformationen zwischen diesen Übertragungssystemen. Die Überprüfung auf Übertragungsfehler und das Einhalten einer entsprechenden Protokollführung gehören ebenfalls zu den Aufgaben eines Crate - Controllers.

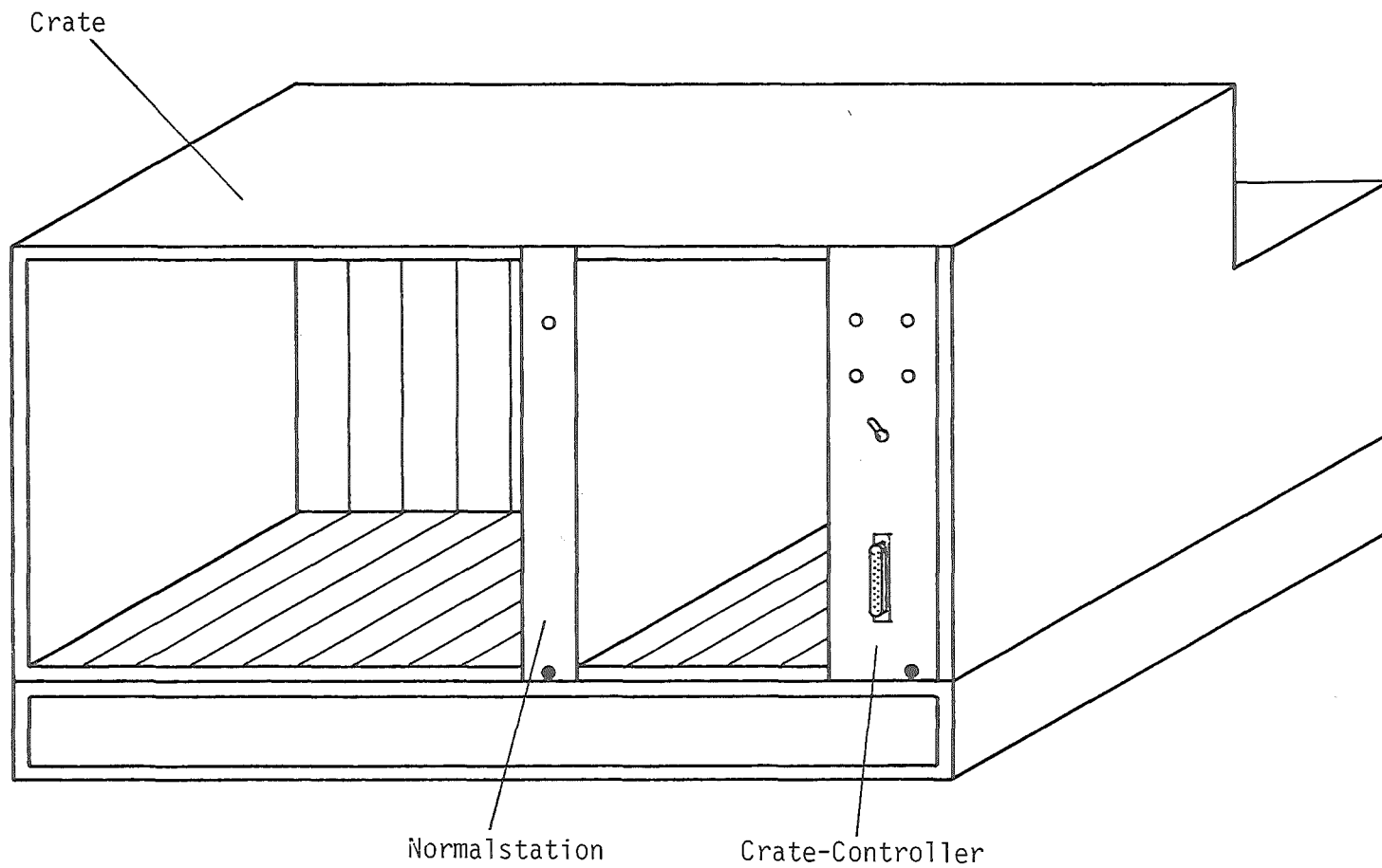


Abb. 0 - 5 CAMAC - Crate mit einer Normalstation und Crate-Controller

Der im vorliegenden Bericht beschriebene PDV - CAMAC Crate - Controller ermöglicht den Anschluß eines CAMAC - Crates an das im PDV - Arbeitskreis entworfene Prozeßdatenübertragungssystem. Damit ist er auf der einen Seite Controller für das CAMAC - Crate und von der SDS - Ankopplungsseite (SDS - serielle Datenschnittstelle zum Buskoppler) eine Unterstation für das PDV - Prozeßdatenübertragungssystem (Abb. 0 - 6). Durch die Anwendung eines Mikroprozessors sowie entsprechender Aufteilung in bestimmte Funktionsbaugruppen kann dieser Controller auf relativ einfache Weise auch für andere Prozeßdatenübertragungssysteme einsatzbereit gemacht werden (siehe auch /3/).

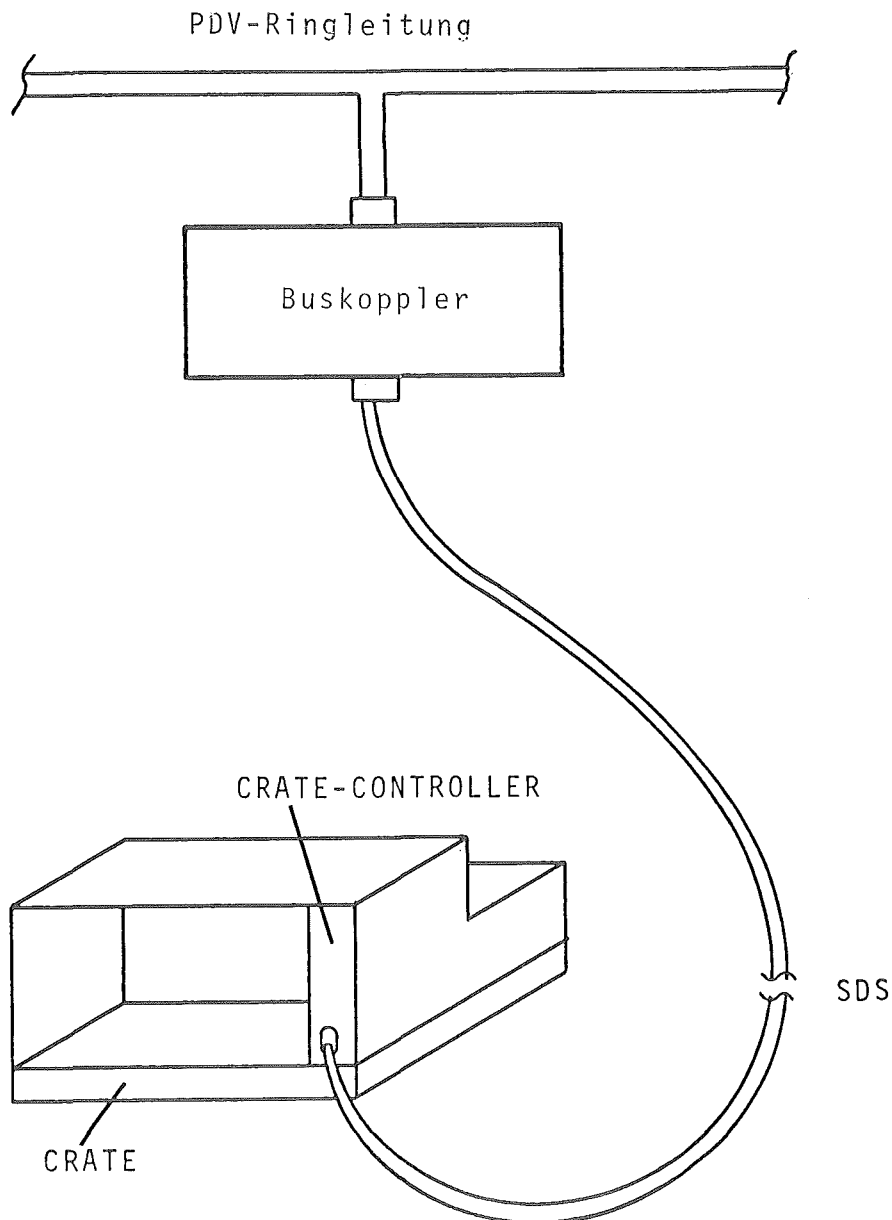


Abb. 0-6

1. PDV - Bus Konzept

Das durch den PDV Arbeitskreis erarbeitete Systemkonzept erlaubt eine bitserielle Datenübertragung bei hoher Betriebssicherheit und Tolerierung des Ausfalls einzelner Stationen (Vermeidung permanenter Rückwirkungen im Fehlerfall). Durch variable Datenblocklängen und prozeßgerechter Datenwortstruktur sowie aufwärtskompatibler Prozeduren ist eine leichte Anwendbarkeit und Anschlußmöglichkeit auch von einfachen Teilnehmern gegeben. Bei der für den PDV - Bus zugrunde gelegten Gesamtstruktur lassen sich bis zu 252 Stationen verschiedener Komplexität anschließen und von einer zentralen Steuerstation bedienen.

Jeder Informationsaustausch geschieht in der Regel nur zwischen der zentralen Steuerstation, die Nachrichten aussendet, und einer oder mehreren Unterstationen. Dabei darf jede Unterstation nur auf Anforderung durch die zentrale Steuerstation, also nach Empfang einer Nachricht, eine Antwort an diese zentrale Steuerstation absetzen. Unterstationen die nicht antworten können gibt es im System nicht.

Zur Ankopplung der Steuereinheit der Unterstation an das PDV - Bus Übertragungssystem ist die bitserielle SDS - Schnittstelle definiert. Die Einführung dieser einheitlichen Nahtstelle ermöglicht die Festlegung der Übertragungsprozeduren unabhängig von der Realisierung der Übertragungsstrecke selbst. Damit erstreckt sich das erarbeitete PDV - Bus Konzept

- a) auf die Festlegung des prozeduralen Teiles von Nachrichten und Antworten, der Übertragungssicherung sowie der Abgabe und Behandlung von Alarmen
- b) auf die Definition der SDS - Schnittstelle in Bezug auf die physikalische Ausführung sowie Signalstruktur und -verteilung.

2. SDS - Schnittstelle

2.1. Physikalische Beschreibung

Die Verbindung zwischen SDS - Schnittstellenkabel und der Steuereinheit der Unterstation geschieht mit einer Steckverbindung der Steckernorm FTZ 386 TV1. Dabei ist das Schnittstellenkabel mit der Steckerleiste und die Steuereinheit mit der Federleiste ausgerüstet. Bei dem CAMAC - SDS Controller ist die Federleiste an der Frontplatte angeordnet und mit den für Schiebeverriegelung notwendigen Verriegelungsbolzen ausgerüstet. Die Zuordnung der einzelnen Signalleitungen ist in der Abb. 2 - 1 dargestellt. Dabei sind die mit Doppelnummern gekennzeichneten Signalleitungen getwistete Doppelleitungen.

2.2. Signalstruktur

Der Abb. 2 - 1 kann die Signalbezeichnung mit zugehöriger Signalflußrichtung entnommen werden. Das Rahmensynchronisationssignal wird mit dem Anfang einer Nachricht gesetzt und nach Erkennen des Endes einer Nachricht wieder zurückgesetzt. Der Empfangstakt gibt mit seiner Rückflanke den Zeitpunkt der Auswertung des Empfangssignales an. Die Vorderflanke des Empfangstaktes gibt den Übergang des Empfangssignales zum nächsten Informationsbit an.

Außerhalb des Rahmensynchronisationssignales sind keine Empfangstakte existent.

Das Sendebereitschaftssignal SB wird vor Übertragung der Antwort von der Steuereinheit gesetzt. Damit wird der Antworttakt angefordert und in Abhängigkeit von diesem das Sendesignal auf der Sendesignalleitung SS bereitgestellt. Nach Übertragung der vollständigen Antwort nimmt die Steuereinheit das SB Signal wieder zurück.

Das Empfangssignal im NRZ - Code und der dazugehörige Empfangstakt werden aus den Bussignalen generiert und auf den Leitungen ES und ET von Buskoppler zur Steuereinheit übertragen.

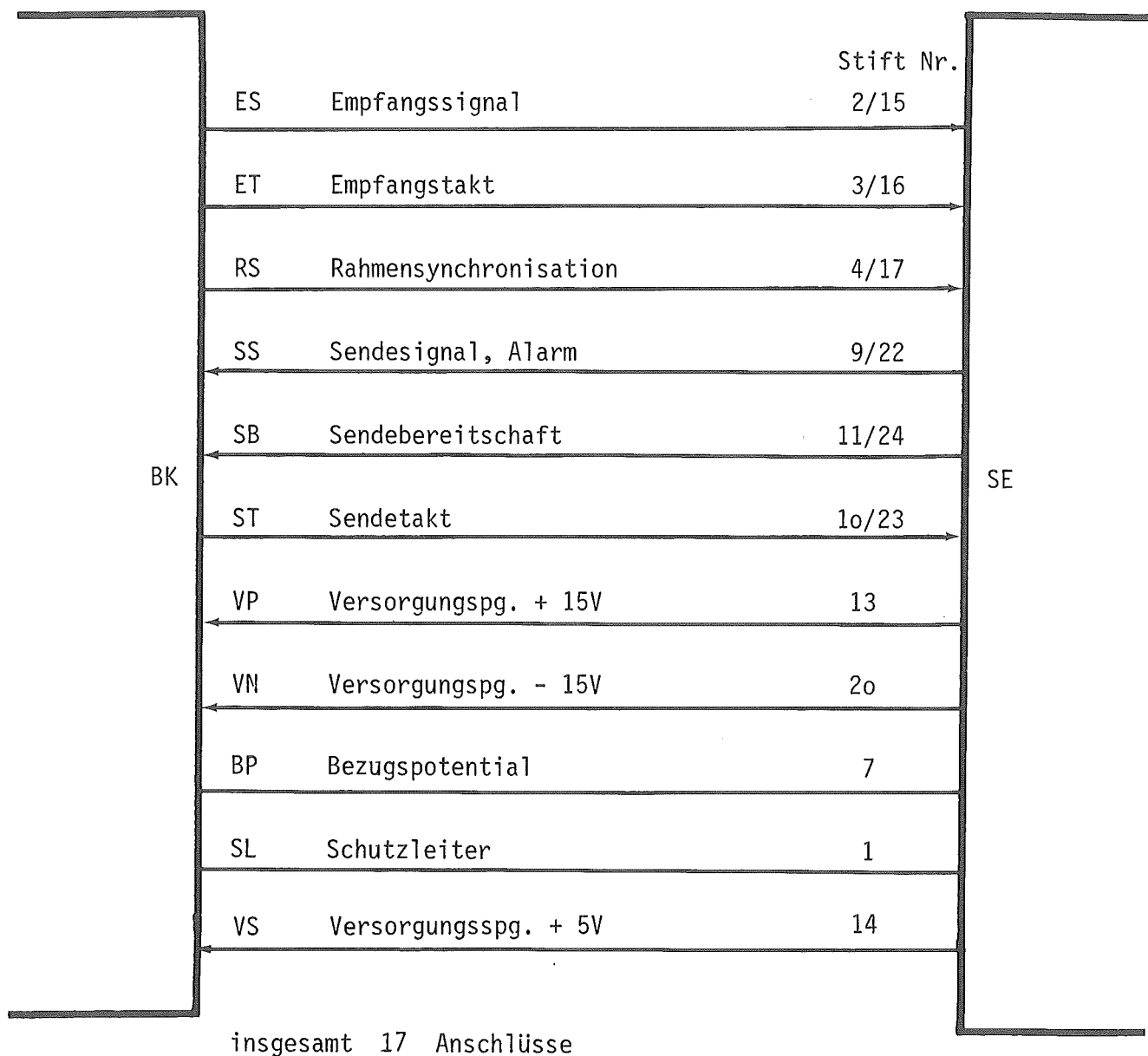


Abbildung 2 - 1

Bezüglich der Zeitabhängigkeiten zwischen den einzelnen Signalen an der SDS - Schnittstelle wird folgendes festgelegt (s. Abb. 2 - 2): (diese Angaben sind gültig für den CAMAC - CRATE - CONTROLLER KFA - ZEL - NE - 365)

- Die maximale Bitfrequenz von 1 M Baud wird bestimmt durch die Angaben:
minimale Pulsbreite des Empfangstaktes ET: $0,75 \mu\text{s}$
minimale Pausenzeit des Empfangstaktes ET: $0,25 \mu\text{s}$
- Die auswertende Flanke des Empfangstaktes ist die Rückflanke.
- Das Rahmensynchronisationssignal RS kommt in der Regel mit der Vorderflanke des 1. Empfangstaktes. Es wird jedoch spätestens $0,5 \mu\text{s}$ vor der Rückflanke des 1. Empfangstaktes gesetzt.
- Frühestens $0,25 \mu\text{s}$ nach der Rückflanke des letzten Empfangstaktes wird das Rahmensynchronisationssignal wieder zurückgenommen.
- Das Sendebereitschaftssignal SB wird von der Steuereinheit gesetzt, wenn gültige Antwortinformationen zur Verfügung stehen. Weil die zeitliche Dauer des SYN - Zeichens implementierungsabhängig ist, wird auch die Zeit zwischen der Rückflanke von RS und der Vorderflanke von SB nicht festgelegt.
- Das Sendebereitschaftssignal SB wird spätestens $0,5 \mu\text{s}$ nach der Vorderflanke des letzten Sendetaktes zurückgenommen.
- Die maximale Verzögerung des Sendesignals SS gegenüber der Vorderflanke des Sendetaktes beträgt $0,5 \mu\text{s}$.

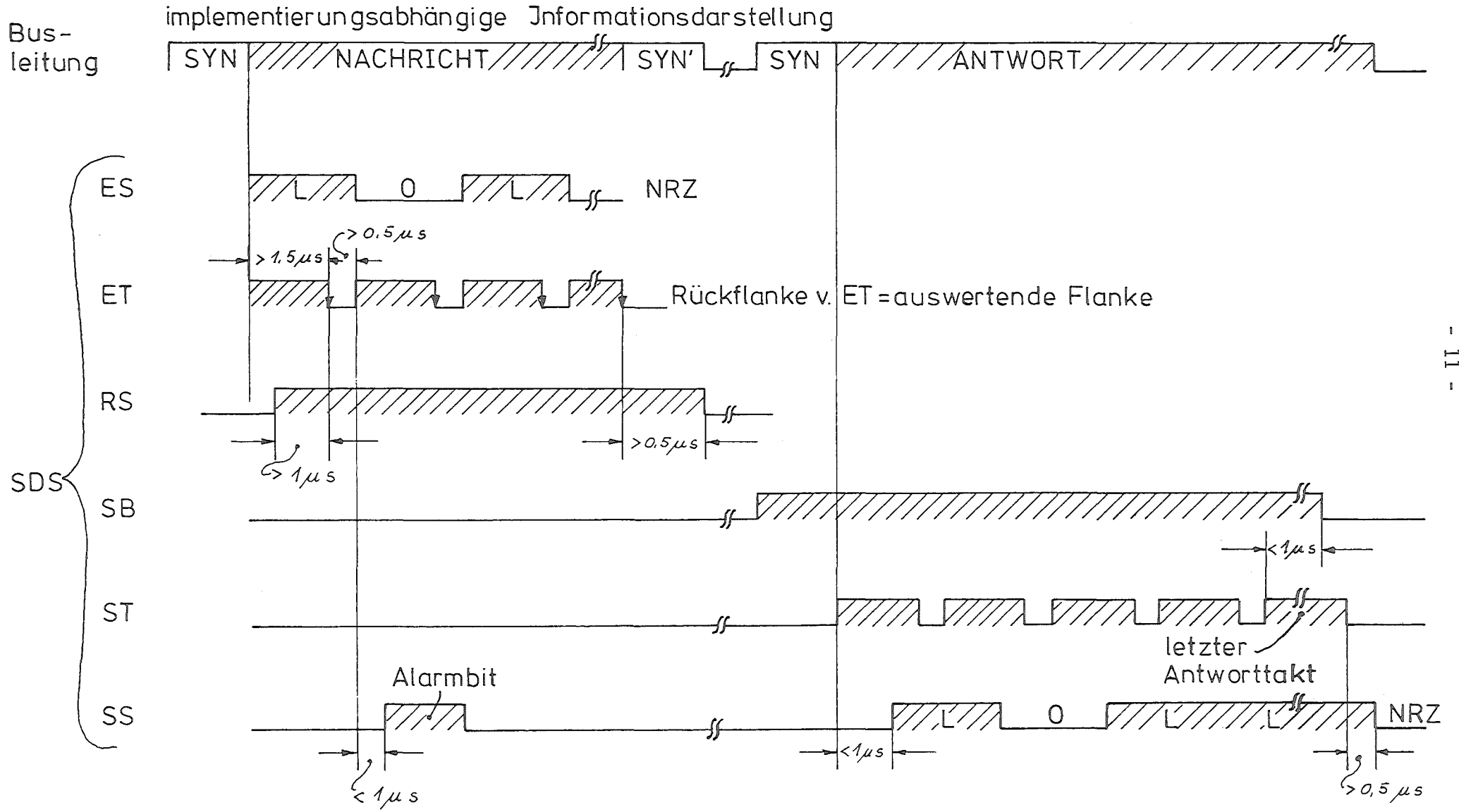


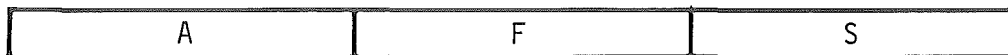
Abb. 2-2

3. PDV - Bus Prozeduren

Das Übertragungselement sowohl einer Nachricht (Informationsquelle ist die aktive Steuerstation) als auch einer Antwort (Informationsquelle ist eine Unterstation) ist ein 8bit Feld (Byte). Diesem 8bit Feld kann abhängig von der Anordnung innerhalb einer Nachricht oder Antwort die Bedeutung eines Adressteiles, eines Funktionenteiles, eines Sicherungsteiles oder eines Datenteiles zukommen. Dabei werden jeweils zwei 8bit Felder durch einen 8bit Sicherungsteil nach den Gesetzen der zyklischen Codierung (CRC) gesichert (siehe 3.3).

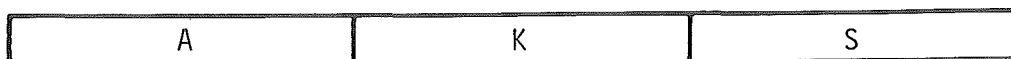
Damit besteht sowohl eine Nachricht als auch eine Antwort aus mindestens 3 Bytes (mit Ausnahme der unter Punkt 5.3 beschriebenen Sonderprozedur) und wird als ein Wort (PDV - Wort) bezeichnet. Jede Nachricht oder Antwort besteht also aus ganzzahligen vielfachen Anordnungen solcher Übertragungswörter. Je nach der Stellung der übertragenen Wörter und der Bytes innerhalb eines Wortes kommen diesen unterschiedliche Bedeutung zu. Das erste Wort einer Nachricht und einer Antwort enthält im 1. Byte die Adresse der angesprochenen Unterstation.

Eine Einzelwortnachricht sieht damit wie folgt aus:



A = Adressbyte F = Funktionsteil S = Sicherungsteil

Bei Einzelantwort haben die einzelnen Bytes folgende Bedeutung:



A = Adressbyte K = Kurzstatus S = Sicherungsteil

Werden in der Nachricht bzw. in der Antwort Daten übertragen, so werden diese in den anschließenden Wörtern (jeweils 16bit Datenteil) untergebracht. Die Bedeutung der Datenteile einer Nachricht kann je nach Stellung innerhalb der Nachricht unterschiedlich sein. So kommt bei einer Blockübertragung dem 1. Byte des 2. Wortes die Bedeutung einer Blocklänge zu (Anzahl der nach dem 1. PDV - Wort übertragenen Datenwörter) und bei einer Leseoperation die Bedeutung als Anzahl der zu lesenden 16bit Datenwörter zu.

3.1 Bedeutung der einzelnen Bitstellen einer Nachricht

A: Adressteil

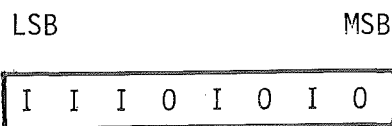
Der gesamte Adressvorrat beträgt 256. Davon müssen jedoch 4 Adressen abgezogen werden.

- Das Adressbit mit der geringsten Wertigkeit (LSB) wird stets zuerst gesendet:

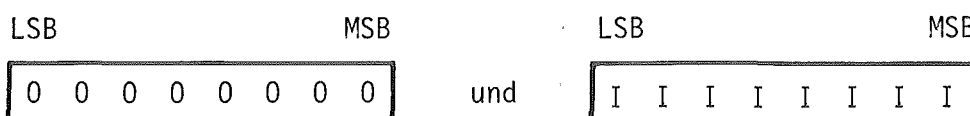
1 2 3 4 5 6 7 8



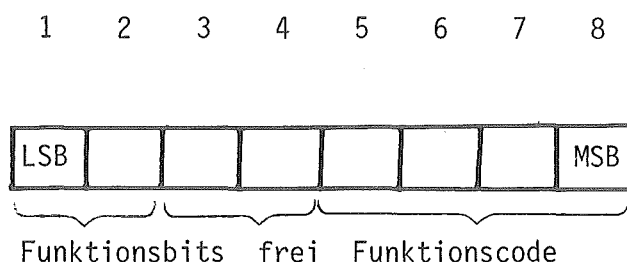
z.B. Adresse 87:



- Aus Gründen der Sicherheit werden die Adressen



im System generell nicht verwendet.



Bit 1

Das 1. Bit im F - Teil gibt an, ob der A - Teil eine Ziel- oder Herkunftadresse enthält. Damit können auch Stationen an Einleitungssystemen zwischen Nachrichten und Antworten unterscheiden, ohne daß verschiedene SYN - Zeichen eingeführt werden müssen.

Es gilt folgende Festlegung für das 1. Bit im F - Teil:

- I = Zieladresse (Nachricht)
- 0 = Herkunftsadresse (Antwort)

Dieses Bit muß also von der aktiven Steuerstation in einer Nachricht auf I gesetzt werden. Entsprechend muß es jede antwortende Unterstation auf 0 setzen.

Bit 2

Das 2. Bit im F - Teil gibt an, ob es sich um eine Einzel- oder Blockübertragung handelt.

Es gilt folgende Festlegung für das 2. Bit im F - Teil:

- I = Blockübertragung
- 0 = Einzelwortübertragung

Ist dieses Bit eine " I ", so enthält der erste Datenteil eine Angabe über die Blocklänge. Die Blocklänge ist immer die Anzahl der zu einem Block gehörenden Datenworte (DD).

Bit 5 - 8

Die mit Bit 5 - 8 codierten Grundfunktionen sind in Tabelle 3.1 - 1 zusammengestellt.

Grundfunktion	Code				
	Bit	5	6	7	8
Normieren		0	0	0	0
Lesen		I	0	0	0
Schreiben		0	I	0	0
Schreiben und Lesen		I	I	0	0
Wiederholung Lesen		I	0	I	0
Wiederholung Schreiben und Lesen		I	I	I	0
Anschalten Unterstation (remote enable)		I	0	0	I
Abschalten Unterstation (remote disable)		0	I	0	I
Statusabfrage (z.B. Alarmanalyse)		I	0	I	I
Beginn zyklischer Kurzabfrage		0	0	I	0
Ende zyklischer Kurzabfrage		I	I	0	I
Beginn Querverkehr		0	0	I	I
Ende Querverkehr		0	I	I	I
Reserve 1		I	I	I	I
Reserve 2		0	0	0	I

Tabelle 3.1 - 1

Die Bits 3 und 4 stehen für anwendungsspezifische Zwecke zur freien Verfügung.

Spezielle Funktionscodes (z.B. für einzelne Geräte einer Unterstation) können in den Datenteilen untergebracht werden.

S: Sicherungsteil

Er dient der Übertragungssicherung von jeweils zwei 8bit Teilen einer Nachricht oder Antwort und wird nach den Gesetzen des zyklischen Codes (CRC) gebildet (siehe hierzu Abschnitt 3.3).

D: Datenteil

Er enthält 8bit transparente Daten. Jeweils 2 Datenteile bilden ein Datenwort (D,D), das als kleinste Dateneinheit in einer Nachricht oder Antwort übertragen wird.

Die Übertragung von mehreren Datenworten (Blockübertragung) innerhalb einer Nachricht oder Antwort ist möglich.

3.2 Übertragungsoperationen

Alle Übertragungsoperationen bauen darauf auf, daß bei Einzeladressoperationen mit Ausnahme der verkürzten Abfrage (s. Abschnitt 5.3) die angesprochenen Unterstationen eine Antwort geben, die aus mindestens zwei 8bit -Teilen und einem zusätzlichen Sicherungsteil besteht. Bei einer Sammeladressierung (paralleles Ansprechen aller Unterstationen) muß auf eine Antwort verzichtet werden.

Weil grundsätzlich sowohl in der Nachricht als auch in der Antwort jeweils zwei 8bit-Felder durch einen S-Teil gesichert werden, wird ein einfacher und gleichmäßiger Aufbau der Übertragungsprozeduren erreicht. Resultierend aus diesen Vorschriften lassen sich die nachstehenden Grundoperationen aufstellen (Tabelle 3.2 - 1):

Übertragungsoperation	Nachricht	Antwort
Sammel-Gruppenadressierung mit Kommandoausgabe	A, F, S	-
Einzeladressierung mit Kommandoausgabe	A, F, S	A, K, S
Einzeladressierung mit Schreiben Einzelwort	A, F, S, D, D, S	A, K, S
Einzeladressierung mit Schreiben Block	A, F, S, D, D, S, D,D,S ...	A, K, S
Einzeladressierung mit Lesen Einzelwort	A, F, S	A, K, S, D, D, S
Einzeladressierung mit Lesen Block	A, F, S, D,D,S	A, K, S D,D,S, D,D,S
Einzeladressierung mit Schreiben und Lesen Einzelwort	A, F, S, D, D, S	A, K, S, D, D, S
Einzeladressierung mit Schreiben und Lesen Block	A, F, S, D,D,S,D,D,S, D,D,S ...	A, K, S D,D,S,D,D,S

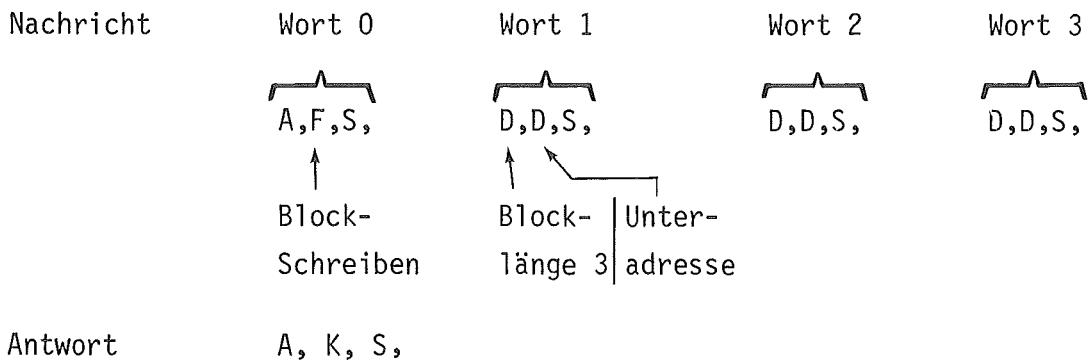
Tabelle 3.2 - 1

Für Blockoperationen gelten folgende Festlegungen:

- Schreiben Block

Die Länge (Anzahl der DDS-Gruppen) des zu schreibenden Blocks wird im 1. D-Byte des 1. Wortes nach A,F,S angegeben. Im 2. D-Byte des 1. Wortes nach A,F,S stehen ggfls. Unteradressen.

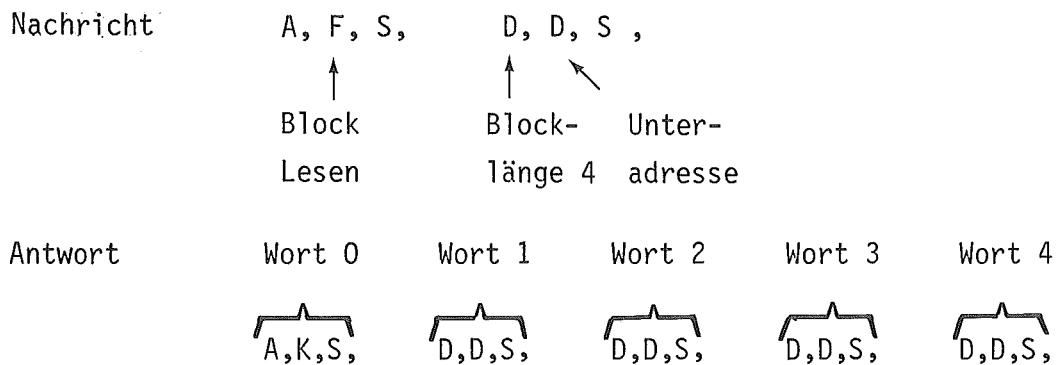
Beispiel:



- Lesen Block

Die Länge des zu lesenden Blocks wird ebenfalls im 1. D-Byte des 1. Wortes nach A,F,S angegeben und im 2. D-Byte stehen ggfls. Unteradressen. Die Antwort enthält keine Blocklängenangabe.

Beispiel:

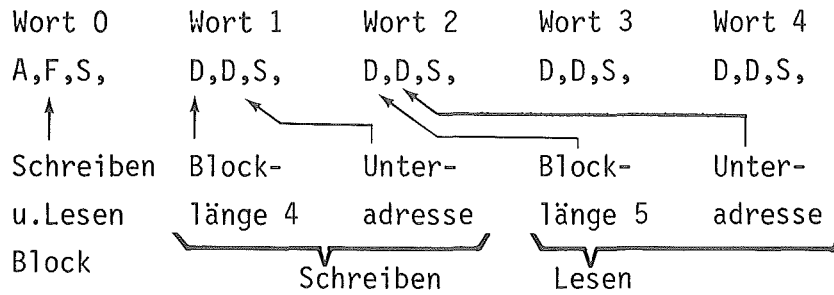


Wie in Kapitel 4. beschrieben werden die für CAMAC - Operationen benötigten Parameter N, A und F in den Datenteilen übertragen. Damit wird eine CAMAC - Leseoperation immer durch eine PDV - Schreiboperation ausgedrückt und die Funktion "Lesen Block" durch "Schreiben und Lesen Block" ersetzt.

- Schreiben und Lesen Block

Für diese Operation gilt die Kombination von "Schreiben Block" "Lesen Block". Auch hier enthält die Antwort keine Blocklängen-angabe.

Beispiel Nachricht



Beispiel Antwort

Wort 0	Wort 1	Wort 2	Wort 3	Wort 4	Wort 5
A,K,S,	D,D,S,	D,D,S,	D,D,S,	D,D,S,	D,D,S,

Prinzipiell sind bei einer Sammel- oder Gruppenadressierung auch die Operationen "Schreiben Einzelwort" oder "Schreiben Block" anwendbar. Da aber für diese Operationen keine Kontrollen durch Antworten möglich sind, sollten sie nur in Ausnahmefällen verwendet werden.

Alarmstatusabfrage

Zur Abfrage des Alarmstatus bei einzelnen Unterstationen durch die aktive Steuerstation ist eine vollständige Nachricht-Antwort-Operation mit einem dafür definierten F-Code (s.Abschnitt 6.1) vorgesehen. Der Alarmstatus wird im Datenteil der Antwort zurückübertragen (s.Tabelle 3.2 - 2)

Übertragungsoperation	Nachricht	Antwort
Alarm-Statusabfrage	A, F, S,	A, K, S, D, D, S,

Tabelle 3.2 - 2

Es kann natürlich auch eine Alarmstatusabfrage ohne vorhergehende Alarm-übertragung durchgeführt werden.

3.3 Übertragungssicherung

Zur Erkennung von Übertragungsfehlern wird eine Sicherung nach dem CRC-Verfahren angewandt.

Ein CRC-Sicherungsteil S besteht aus 8 Bit und sichert jeweils die beiden vorausgehenden 8bit Felder. Der Hauptvorteil dieses Sicherungsverfahrens gegenüber dem von HDLC besteht darin, daß jede Station bereits nach dem 3. Byte einer beliebig langen Nachricht die gesicherten A- und F-Teile empfangen hat und feststellen kann, ob sie angesprochen ist oder nicht. Dadurch kann der Aufwand in "Einfachen Teilnehmern" gering gehalten werden und die lineare Abhängigkeit der Kosten vom Ausbaugrad des Gesamtsystems ist leichter zu erreichen.

Weiterhin läßt sich die CRC-Prüfung für Informationseinheiten konstanter Länge hardwaremäßig einfacher realisieren und die Sicherungseffizienz ist unabhängig von der Blocklänge (konstante Hammingdistanz).

Schließlich bietet dieses Verfahren noch Vorteile bezüglich einer differenzierten Fehlerbehandlung (s. Abschnitt 3.4).

Theoretisch ist sowohl eine Fehlererkennung als auch eine Fehlerkorrektur damit möglich. Da jedoch bereits bei Fehlern niedrigen Grades der Hardware-Aufwand unverhältnismäßig groß ist, wird grundsätzlich auf eine Fehlerkorrektur verzichtet. Der Sicherungsteil wird ausschließlich zur Fehlererkennung benutzt.

Wie aus Untersuchungen hervorgeht, ist die Gesamtzahl der nicht erkannten Fehler bei allen CRC-Generatorpolynomen gleich. Die Verteilung der nicht erkannten Fehler schwankt jedoch in Abhängigkeit vom gewählten Generatorpolynom. Prinzipiell ist somit die Möglichkeit gegeben, die Effizienz der Sicherung an ein gegebenes Störklima, d.h. an eine spezielle Häufigkeitsverteilung der Fehler anzupassen. Damit würde aber auch ein die Prozedur bestimmender Parameter variiert werden, was dem Ziel dieses Vorschlages - eine möglichst einheitliche Lösung zu empfehlen - nicht entspricht. Deshalb wird als Kompromißlösung das Generatorpolynom

$$G = X^8 + X^2 + X + 1$$

gewählt.

Ein großer Vorteil dieses Polynoms besteht darin, daß bereits integrierte Großschaltkreise als Codierer und Decodierer dafür erhältlich sind (VALVO Chip GZF 12o2D).

3.4 Fehlerbehandlung

Unterschiedliche Leitungsstrukturen haben unterschiedliche Identifizierungsmöglichkeiten von Fehlerursachen zur Folge. Unabhängig von den gewählten Leitungsstrukturen sind folgende Fehlerbehandlungsprozeduren möglich:

Gestörte Nachricht

Gemäß der Festlegung im Abschnitt 3.2 quittiert jede Unterstation Empfang einer fehlerfreien Nachricht durch eine Antwort. Ist in einer Nachricht der Adressteil A oder der Funktionsteil F oder der Sicherungsteil S verfälscht, wird grundsätzlich keine Antwort gegeben und die aktive Steuerstation wiederholt die Nachricht. Das Ausbleiben der Antwort stellt die aktive Steuerstation mit Hilfe einer einfachen Zeitüberwachungsschaltung fest. Entsprechende Wiederholungsoperationen sind durch spezielle F-Codes als solche gekennzeichnet (s. Abschnitt 3.1).

Gleiches gilt, wenn ein Fehler in Datenteilen vorliegt. Auch dann antwortet die angesprochene Unterstation nicht und die aktive Steuerstation wiederholt die Nachricht.

Die Fehlerbehandlungsprozeduren bei verfälschten Nachrichten lassen sich wie folgt zusammenfassen (Tabelle 3.4 - 1):

Fehlerhafter Teil in der Nachricht	Reaktion der Unterstation	Reaktion der aktiven Steuerstation
A,F,S D,D,S...	keine Antwort	Wiederholung der Nachricht
A,K,S D,D,S D,D,S...	keine Antwort	Wiederholung der Nachricht

Tabelle 3.4 - 1

Bei den zu bevorzugenden Ringkonfigurationen kann ein Fehler in einer Nachricht auch noch durch Vergleich der ausgesandten mit der zurückerhaltenen Nachricht in der aktiven Steuerstation erkannt werden.

4. Protokoll für CAMAC - Prozeduren

Grundsätzlich bleibt für alle Anwendungen das vorgeschriebene Protokoll der PDV - Bus Beschreibung erhalten. Ist die PDV Übertragungsoperation lediglich der Träger einer anderen Protokollführung, so treten deren Adress- und Funktionenteile sofern sie nicht mit dem PDV - Protokoll übereinstimmen, als PDV - Bus Datenteile auf. Da ein allgemeiner CAMAC - Befehl aus den Parametern C,N,A und F besteht müssen die Parameter N,A und F als Datenteil in das PDV - Protokoll übernommen werden.

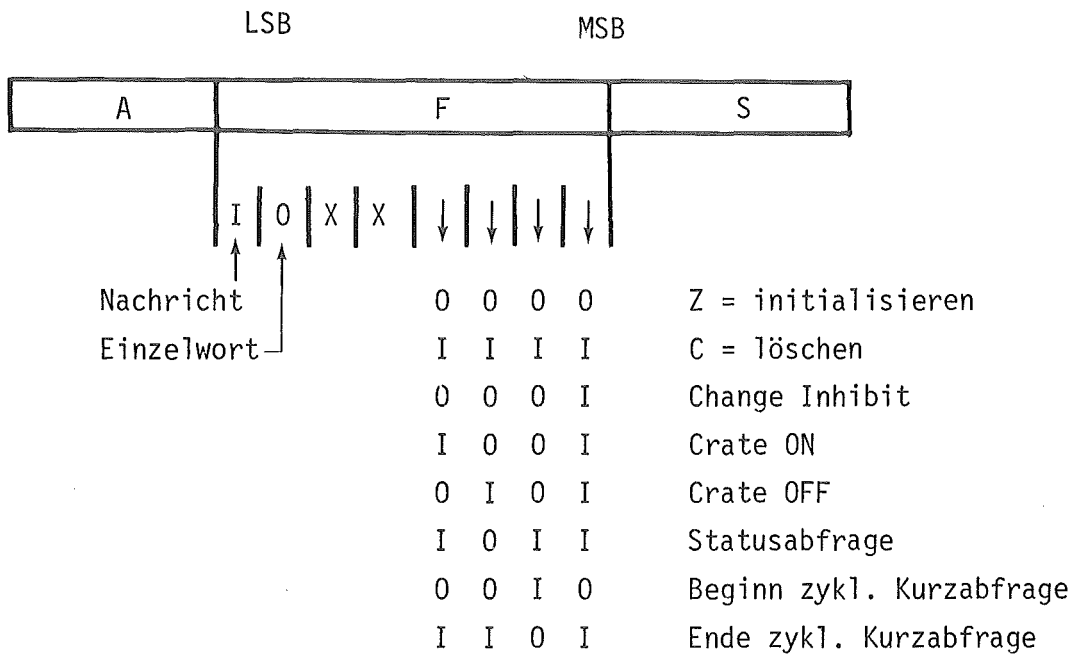
4.1 Globaler Crate - Befehl

Hierbei muß zwischen Befehlen unterschieden werden, die sowohl in der PDV - Bus Beschreibung als auch im standardisierten CAMAC - Protokoll vorkommen und solchen, die nur CAMAC spezifisch sind. Bei Befehlen, die beiden Spezifikationen genügen hat man damit zwei mögliche Nachrichten für die Auslösung der Funktion zur Verfügung. Einmal durch Aussendung eines Einzelwortes wobei im Funktionenteil die gewünschte Funktion angegeben ist; zum Anderen als datenloser CAMAC - Befehl bei dem durch Aufschlüsselung des N,A und F - Teiles die globale CAMAC - Funktion beschrieben wird. Bei den nur CAMAC spezifischen Befehlen ist nur der Aufruf über die N,A und F Funktion möglich.

Globale CAMAC - Cratefunktionen sind:

N 28 A 8 F 26	absetzen von "Z" (initialisieren)
N 28 A 9 F 26	absetzen von "C" (löschen)
N 30 A 0 F 1	read Status
N 30 A 0 F 17	write Status
N 30 A 0 F 19	selective set Status
N 30 A 0 F 23	selective clear Status
N 30 A 9 F 24	Inhibit OFF
N 30 A 9 F 26	Inhibit ON
N 30 A 9 F 27	Test Inhibit
N 30 A12 F 1	read LAM - Pattern

Übertragungsbeispiel einer Crate - Funktion durch PDV - Einzelwort:



X = logischer Zustand ohne Bedeutung

Bei anderen als den aufgeführten Bitkombinationen wird durch die Steuereinheit lediglich eine Einzelantwort generiert, die die Fehlerfreiheit der Übertragung (falls CRC -richtig) signalisiert.

Die Bitkombinationen für "Beginn-" und "Ende zyklische Kurzabfrage" werden nur bei Sammeladressierung (A = I I I I I I I 0) als gültig erkannt!

Die Bitkombinationen für Z,C und Crate ON und Crate OFF werden sowohl bei Einzeladressierung als auch bei Sammeladressierung erkannt und ausgeführt.

Die Funktionen

N 3o A 0 F 17

N 3o A 0 F 19

N 3o A 0 F 23

sind jedoch keine datenlosen CAMAC - Befehle, da die Information zur Änderung des Statusregisters mit übertragen werden muß. Diese Befehle sind deshalb unter den Punkten 6.3,6.4 und 6.5 gesondert beschrieben.

Die Funktionen

N 3o A 0 F 1

N 3o A 12 F 1

sind datenlose CAMAC-Befehle. Da aber Daten in der Antwort übertragen werden, und zwar für das Lesen des Status ein PDV - Datenwort und beim Lesen des LAM - Pattern zwei PDV - Datenworte, sind auch diese im Kapitel 6.1 und 6.2 ausführlich beschrieben.

4.2 Datenloser CAMAC - Befehl

Da jeder CAMAC - Befehl aus den Parametern N,A und F besteht, müssen diese in den Datenteilen des PDV - Protokolls übertragen werden. Damit ist die Übertragung eines CAMAC - Befehles immer mit einer PDV - Blockschreiboperation verbunden. Die Zuordnung der einzelnen CAMAC - Parameter zu den PDV - Datenteilen ist der folgenden Abbildung zu entnehmen (Abb. 4.2 - 1). Im Funktionenteil des ersten PDV - Wortes haben die ersten beiden Bits den logischen Zustand "1". Damit wird die Nachricht als solche charakterisiert und eine Blockübertragung angekündigt, d.h. das erste Byte des zweiten Wortes beinhaltet die Blocklänge (Anzahl der auf das erste übertragene PDV - Wort folgenden Wörter). Das 5. Bit läßt durch seinen logischen Zustand "0" erkennen, daß es sich um keine Leseoperation handelt. Das 6.Bit hat den logischen Zustand "1" womit angezeigt wird, daß die Datenbytes der folgenden PDV - Wörter in die entsprechenden CAMAC - Register (N,A und F) des Controllers eingeschrieben werden müssen. Im ersten Byte des zweiten übertragenen Wortes wird die Blocklänge angegeben. Da für die Übertragung der CAMAC - Parameter N,A und F neben dem ersten übertragenen PDV - Wort noch zwei weitere benötigt werden, muß die Blocklänge (für datenlosen CAMAC - Befehl) mit 2 angegeben werden. Das zweite Byte des zweiten Wortes beinhaltet die binärcodierte Stationsnummer und im dritten Wort werden die CAMAC - Parameter A und F übertragen.

4.3 CAMAC - Befehl mit Funktion "Daten schreiben"

Sollen neben der Übertragung des CAMAC - Befehls auch Daten übergeben werden, so sind diese in den auf die CAMAC - Parameter beinhaltenden PDV - Wörter untergebracht. Am logischen Zustand des Funktionenteiles im ersten Wort ändert sich hierbei gegenüber der Übertragung eines datenlosen CAMAC - Befehles nichts. Die Angabe der Blocklänge im ersten Byte des zweiten Wortes gibt Auskunft über die Anzahl der in dieser Nachricht zur Übertragung gelangenden PDV - Datenwörter.

Bei der vorliegenden Software des Controllers ist wahlweise die Übertragung von 16bit oder 24bit breiten CAMAC - Daten möglich. Zur Unterscheidung dieser Datenformate wird das 6. Bit des 2. Bytes im zweiten Wort (Stationsnummer N) benutzt. Ist dieses Bit logisch "0" gesetzt, so handelt es sich um eine Übertragung 16bit breiter Daten für die jeweils ein PDV - Datenwort benötigt wird. Nach jedem übertragenen und korrekt empfangenen Datenwort wird von der Steuereinheit der Unterstation ein CAMAC - Cyclus mit den empfangenen CAMAC - Parametern generiert. Die innerhalb eines Wortes übertragenen 16 Datenbits werden dabei den CAMAC - Schreibleitungen W1 bis W16 aufgeschaltet. In Abb. 4.3-1 ist ein Übertragungsfall mit drei 16bit breiten Datenwörtern gezeigt.

Ist das 6. Bit des 2. Bytes im zweiten Wort logisch "1" so werden 24bit breite Daten übertragen für die jeweils 2 PDV - Datenwörter benötigt werden. Abb. 4.3 - 2 zeigt die Übertragung zweier 24bit breiten Datenwörter. Tritt während der Übertragung des Datenblockes ein CRC - Fehler auf, so wird die PDV - Wortnummer, bei welcher der CRC - Fehler erkannt wurde, registriert und das weitere Absetzen von Daten mit Hilfe von CAMAC - Cyclen unterdrückt.

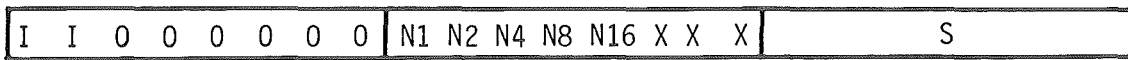
4.4 CAMAC - Befehl mit Funktion "Daten lesen"

Bei der Übertragung einer CAMAC - Funktion zum Datenlesen muß dieses im Funktionsteil des ersten übertragenen PDV - Wortes kenntlich gemacht werden. Gegenüber der Übertragung eines datenlosen CAMAC - Befehles muß hierbei zusätzlich das 5. Bit logisch "1" sein. Damit wird eine Leseoperation angezeigt. Obwohl für die CAMAC - Funktion ausschließlich eine Leseoperation gewünscht wird, muß im Funktionsteil der PDV - Bus Übertragung auch das 6. Bit gleich logisch "1" sein, da von der Sicht der PDV - Übertragung die Parameter N, A und F der CAMAC - Leseoperation als Daten übertragen und in Register des CAMAC - Controllers eingeschrieben werden müssen.

Im zweiten übertragenen PDV - Wort gibt das erste Byte wieder die Länge der Nachricht an, die für eine reine CAMAC - Lesefunktion gleich 3 ist. Das zweite Byte muß für den Fall des reinen Lesens logisch "0" sein. Im dritten Wort gibt das 1. Byte die Länge des Antwortblockes (auf das 0. PDV - Wort der Antwort folgende PDV - Datenwörter) an. Das zweite Byte des dritten Wortes enthält die binärcodierte Stationsnummer aus welcher Daten mit der in Wort 4 angegebenen Subadresse A und Funktion F gelesen werden sollen. Das 6. Bit des 2. Bytes wird zur Kennzeichnung des Datenformates (16bit oder 24bit breite Daten) benutzt. Eine logische "0" kennzeichnet 16bit breite Daten, die die gelesene Information der CAMAC - Leitungen R1 bis R16 wiedergeben. Abb. 4.4 - 1 zeigt den Aufbau einer PDV - Nachricht mit einer reinen CAMAC - Leseoperation für 16bit breite Daten. Eine logische "1" des 6. Bits des 2. Bytes kennzeichnet 24bit breite Daten, die jeweils zwei PDV - Datenwörter der Antwort belegen. Abb. 4.4 - 2 zeigt als Beispiel die Antwort auf eine Nachricht zum Lesen 24bit breiter Daten und deren Aufteilung auf die benötigten PDV - Wörter.

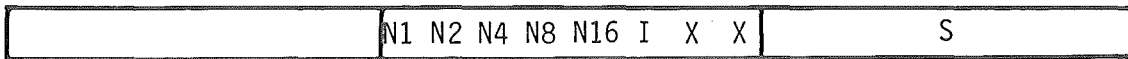
4.5 Datenloser CAMAC - Befehl mit CAMAC - Funktion "Daten lesen"

Aufgrund der PDV - Bus eigenen Nachrichtenstruktur ist hierbei gegenüber den standardisierten CAMAC - Prozeduren eine kombinierte Schreib- und Leseoperation möglich. Mit gewissen Einschränkungen lassen sich auch mit dem hier beschriebenen SDS - CAMAC Controller kombinierte PDV Schreib- Leseoperationen ausführen. Im Gegensatz zur PDV - Nachrichtenstruktur mit CAMAC - Lesebefehl ändert sich lediglich die Angabe des zweiten Bytes im zweiten Wort. Im Falle der reinen CAMAC - Leseoperation muß dieses Byte logisch "0" sein. Damit werden CAMAC - Operationen vor der eigentlichen Leseoperation unterdrückt. Ist dagegen dieses zweite Byte des zweiten Wortes ungleich 0, so wird damit bei der reinen Schreiboperation eine binärcodierte Stationsnummer angegeben, an welche der datenlose CAMAC - Befehl abgesetzt wird. Da die Angabe einer CAMAC - Subadresse A sowie einer CAMAC - Funktion F innerhalb einer Nachricht nur einmal möglich ist, besitzt der abgesetzte datenlose CAMAC- Befehl die gleiche Subadresse wie die anschließende Leseoperation. Für den Parameter F gilt hierbei, daß von wenigen Ausnahmen abgesehen, alle datenlosen CAMAC - Befehle den Parameter F größer gleich 16 haben. Aus diesem Grunde wird bei kombinierten Schreib- Leseoperationen für die Schreiboperation der im zweiten Byte des 4. Wortes angegebenen Parameter F mit zwangsweise logisch "1" gesetzten 5. Bit benutzt ($F_{16} \cong "1"$). Siehe Abb. 4.5 - 1



Blocklänge der
Nachricht

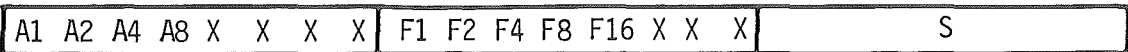
Stationsnummer des
vorabgesetzten
CAMAC-Cycle



Blocklänge der
Antwort
(Anzahl der zu
lesenden Datenworte)

Stationsnummer
aus der gelesen
werden soll

"I" ≙ 24bit Datenformat



Subadresse

Funktion

für Schreiboperation="I"

Abb. 4.5-1

4.6 CAMAC - Befehl mit Funktion "Daten schreiben" und "Daten lesen"

Für den Fall einer kombinierten Daten- Schreib- und Leseoperation kann der Abb. 4.6 - 1 die erforderliche PDV - Nachrichtenstruktur entnommen werden. Sie unterscheidet sich zur angegebenen kombinierten Funktion nach Pkt. 4.5 lediglich dadurch, daß die Blocklänge der Nachricht (erstes Byte des zweiten Wortes) größer als 3 ist. Das zweite Byte des zweiten Wortes gibt wieder die Stationsnummer an, in welche Daten eingeschrieben werden sollen. Das Einschreiben geschieht wieder mit den angegebenen Parametern A und F im 4. übertragenen PDV - Wort. Auch hierbei wird das 5. Bit des F Parameters zwangsweise logisch "1" gesetzt. Die CAMAC - Leseoperation wird erst nach abgeschlossener Schreiboperation eingeleitet.

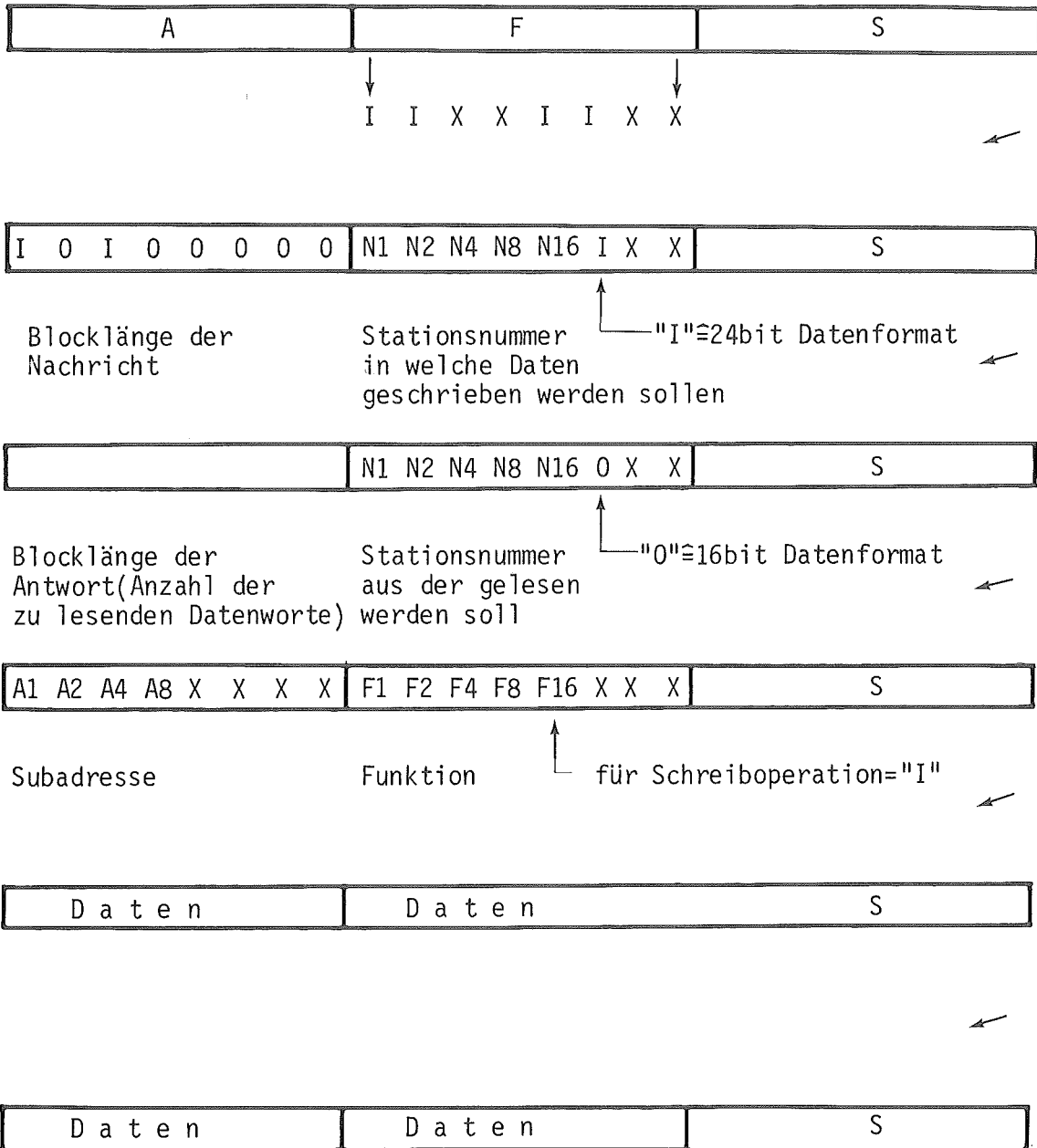


Abb. 4.6-1

5. Spezielle PDV - Funktionen

5.1 Funktion "Wiederholen Schreiben"

Ist bei der Übertragung eines Datenblockes ein CRC - Fehler aufgetreten (siehe Pkt. 4.3) so wird das weitere Absetzen von Daten mit Hilfe von CAMAC - Cyclen unterdrückt. Da ein CRC - Fehler erkannt wurde, gibt der Crate - Controller als Reaktion auf die Nachricht keine Antwort aus. Da die Übertragung nicht fehlerlos war, muß sie vom Treiber neu generiert werden (nach erfolgter Statusabfrage und Fehlererkennung sowie Auswertung). Um den Treiber möglichst wenig zu belasten, hat dieser die Möglichkeit die gesamte Nachricht zu wiederholen. Zur Kenntlichmachung der Nachricht, daß diese eine Wiederholung ist, wird im Funktionenteil der Nachricht (2. Byte des 1. Wortes) das 7. Bit gleich logisch "1" gesetzt. Erkennt der Controller bei einer für ihn abgesetzten Nachricht dieses gesetzte Bit, so wird das Absetzen der Daten mit Hilfe von CAMAC - Cyclen so lange unterdrückt, bis die Wortnummer der wiederholten Nachricht mit der Nummer des gestörten Wortes der abgebrochenen Übertragung übereinstimmt. Es wird also nur der bei der abgebrochenen Übertragung nicht abgesetzte Teil des Datenblockes weitergeleitet und bei fehlerfreier Nachricht eine Antwort abgesetzt. Tritt auch bei wiederholter Block-Schreibfunktion ein CRC - Fehler auf, so wird auch hierbei die Weiterleitung der Daten unterbrochen und die Wortnummer bei der der Fehler aufgetreten ist abgespeichert. Bei mehrfach wiederholten Nachrichten, bei denen ein CRC - Fehler aufgetreten ist wird also die Nummer des ersten erkannten Fehlers der Nachricht überschrieben, wobei das Überschreiben nur dann freigegeben wird, wenn die neue Wortnummer des aufgetretenen CRC - Fehlers größer ist als die bereits abgespeicherte. Auf diese Weise wird verhindert, daß Datenteile doppelt abgesetzt werden und die Gesamtnachricht so oft wiederholt werden kann bis eine fehlerfreie Übertragung stattgefunden hat und der Controller antwortet.

Damit sind zwei Möglichkeiten für die Funktion "Wiederholen Schreiben" möglich:

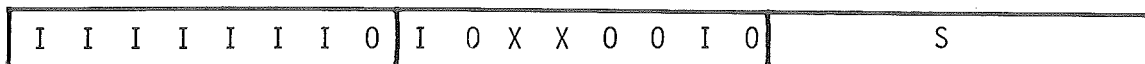
- a) der Treiber wiederholt die gesamte Nachricht und setzt im Funktionenteil des 1. Wortes das 7. Bit gleich "1" oder
- b) der Treiber wertet die im Statuswort übertragene Wortnummer der fehlerhaften Übertragung aus indem er nur den Datenrest überträgt, das 7. Bit im Funktionenteil des 1. Wortes gleich "0" setzt.

5.2 Funktion "Wiederholen Lesen"

Bei der PDV - Funktion "Wiederholen Lesen" ist ebenso wie bei der Funktion "Wiederholen Schreiben" das 7.Bit im Funktionenteil des 1.Wortes gesetzt.Bei der Ausführung einer Lesefunktion werden vom Controller die aus dem CAMAC-crate gelesenen Daten einmal zur Antwortbildung herangezogen zum anderen im RAM Arbeitsspeicher mit aufsteigenden Adressen abgelegt.Erkennt der Controller in der PDV - Nachricht die Funktion "Wiederholen Lesen",so werden keine CAMAC-Cyclen generiert und die zuvor im Arbeitsspeicher abgelegten Daten nochmals in der Antwort übertragen.

5.3 Zyklische Kurzabfrage

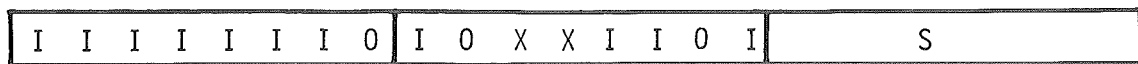
Eine sehr einfache Möglichkeit zur Alarmübertragung ist bei den PDV - Prozeduren die verkürzte zyklische Abfrage einzelner Unterstationen. Um diese zyklische Abfrage mit dem geringsten Zeitaufwand durchzuführen wird bei dieser Sonderprozedur von dem normalen Nachrichten Antworten Protokoll mit Mindestlänge von 24bit (ein PDV - Wort) Abstand genommen und lediglich das Adressbyte als Nachricht abgegeben. Dabei werden alle alarmfähigen Unterstationen nacheinander angesprochen und durch Antworten mit einer Alarmadresse in die Lage versetzt einen Alarm abzugeben. Liegt kein Alarm vor, so wird die eigene Adresse als Antwort zurückgeschickt. Diese Sonderprozedur muß aber durch eine Ankündigung mit Hilfe eines Einzel - PDV - Wortes und Sammeladresse eingeleitet und beendet werden. Die Nachricht zur Ankündigung der Sonderprozedur sieht folgendermaßen aus:



Sammeladresse

Fkt. für Beginn zykl. Kurzabfrage

und die Nachricht zum Beenden der Prozedur:



Sammeladresse

Fkt. für Beenden zykl. Kurzabfrage

Da beide Nachrichten mit Sammeladresse übertragen werden, werden von den Unterstationen auf diese Nachrichten keine Antworten abgegeben. In der Tabelle 5.3 - 1 ist der Ablauf dieser Sonderprozedur beschrieben.

Übertragungsoperation	Nachricht	Antwort
Ankündigung der Sonderprozedur	A, F, S	-
verkürzte zyklische Alarmabfrage einzelner Unterstationen	A . .	A (A ⁺) . .
Beenden der Sonderprozedur	A, F, S	-

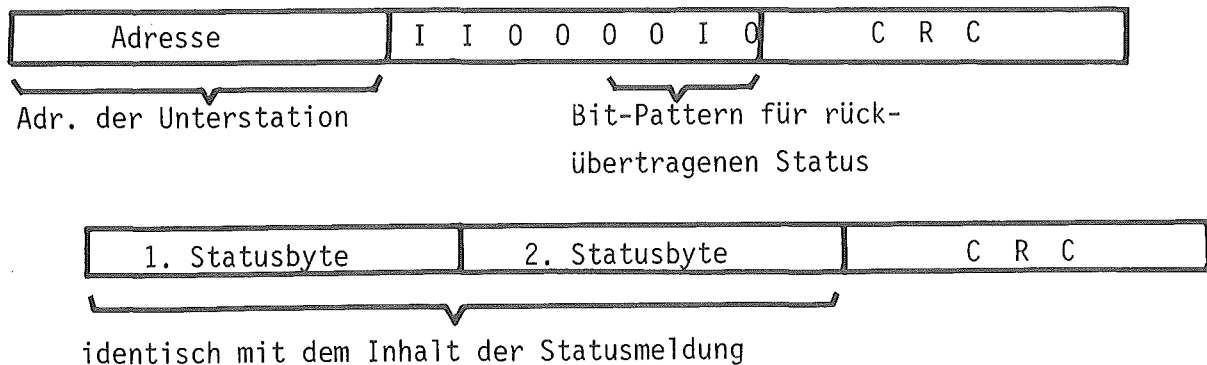
Tabelle 5.3 - 1

A = Adresse der angesprochenen Unterstation
A⁺ = Alarmadresse ("0 I I I I I I I")

5.4 Rückübertragener Status

Um den prozeduralen Ablauf auf dem Übertragungssystem möglichst zeitoptimal zu gestalten, sollte ein von der Unterstation abgesetzter Alarm nur solange aufrechterhalten bleiben (abgesetzt werden), bis die zentrale Steuerstation den Alarm erkannt hat. Die, den Alarm absetzende Unterstation, muß deshalb nach erfolgreicher Übertragung der Statusmeldung die intern gespeicherte Alarmadresse löschen (nicht Zurücksetzen der Alarmquelle).

Um der Unterstation die erfolgreiche Übertragung einer Statusmeldung mitzuteilen, ist in den implementierten PDV - Bus Prozeduren die Rückübertragung einer von der zentralen Steuerstation empfangenen Statusmeldung möglich. Die Struktur dieser PDV - Nachricht sieht folgendermaßen aus:



Im entwickelten Crate - Controller werden die einzelnen LAM - Signale des CAMAC - Crates zwischengespeichert und zur Bildung des Summenalarms herangezogen. Eine von der Unterstation abgesetzte Statusmeldung beinhaltet dabei immer die Adresse des höchstpriorären LAM - Signales (siehe Abb. 6.1-2).

Wird von einer Unterstation die erfolgreiche Übertragung einer Alarmquelle (LAM - Adresse) erkannt, so wird das diesen Alarm kennzeichnende Bit im LAM - Adressenzwischenspeicher des Crate - Controllers gelöscht. Sind im Zwischenspeicher mehrere LAM - Adressen abgelegt, so wird bei Abfrage der Statusinformation die nächsthöhere LAM - Adresse übermittelt (entsprechend der getroffenen Prioritätenfestlegung).

Abb. 5.4-1 zeigt schematisch die logische Verknüpfung der LAM - Adressen zum Summen-LAM Alarmsignal sowie die logischen Abläufe bei Empfang einer rückübertragenen Statusinformation.

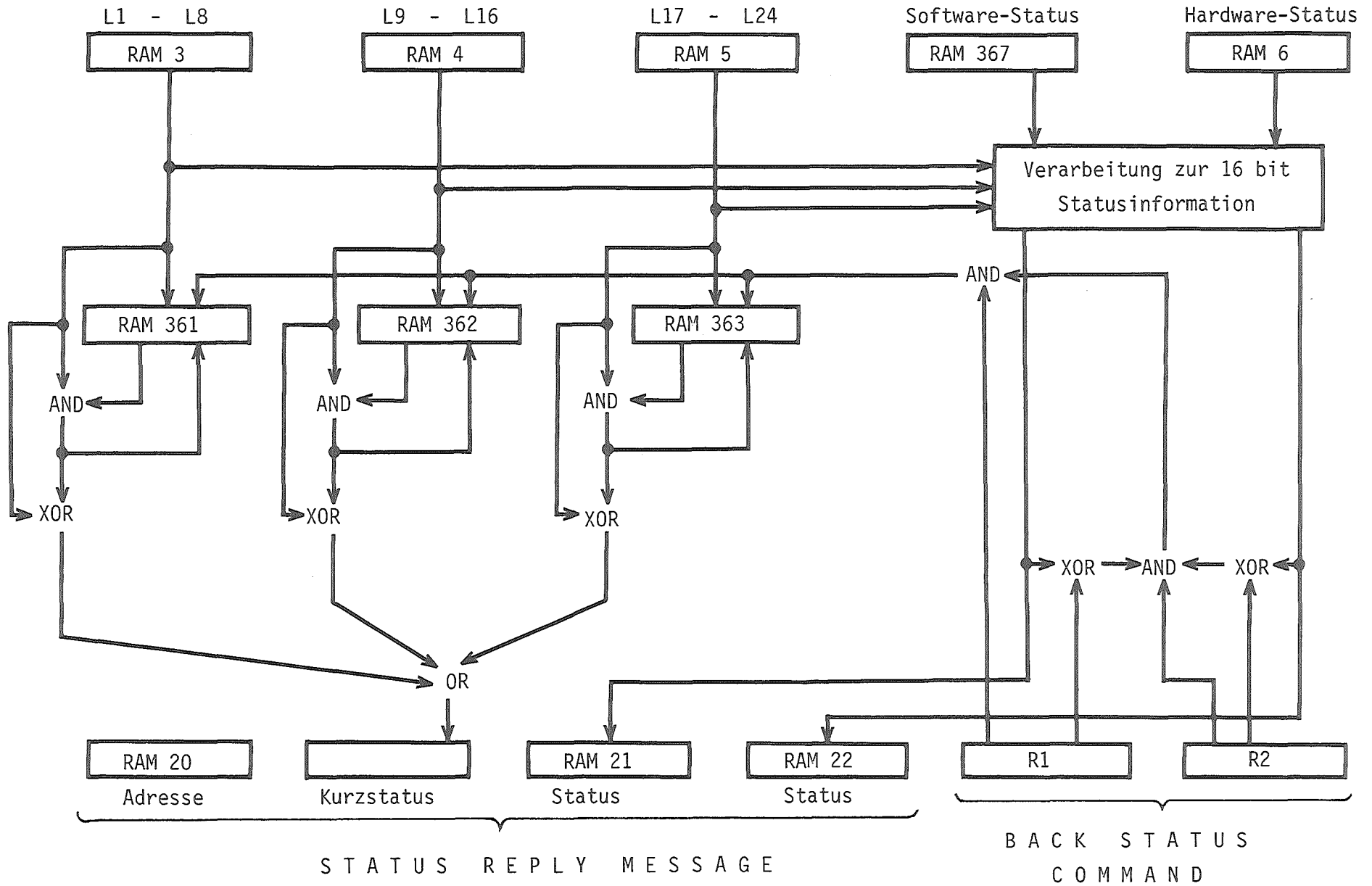


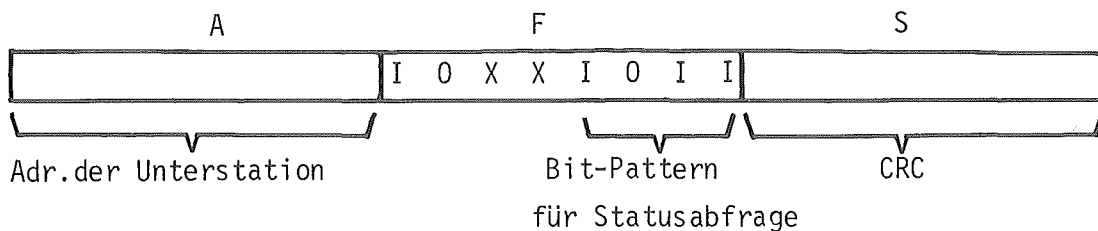
Abb. 5.4-1 Logische Verknüpfung der LAM - Adressen

6. Spezielle CAMAC - Crate - Befehle

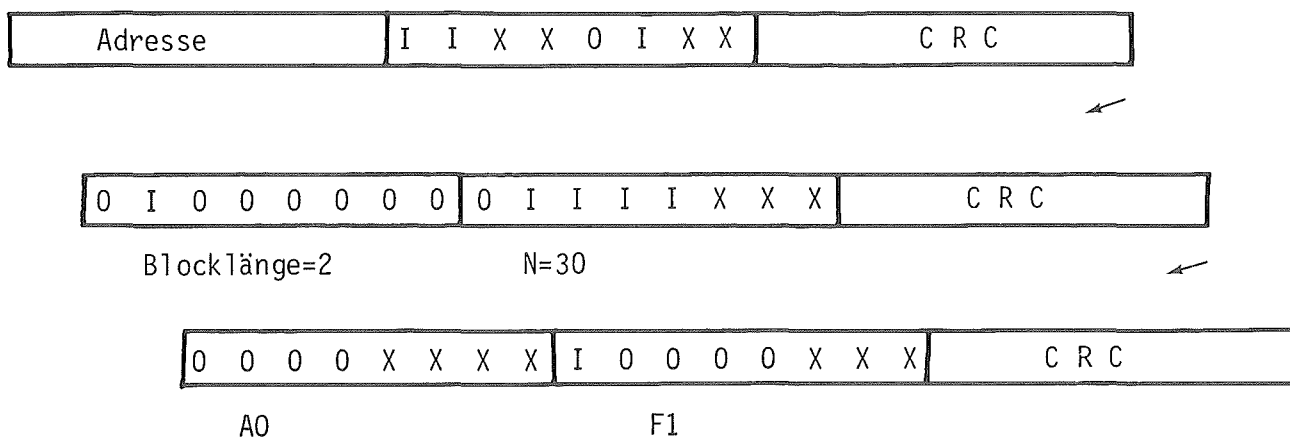
6.1 Statusabfrage

Wie bereits im Kapitel 4.1 gezeigt, kann die Funktion der Statusabfrage durch zwei mögliche Nachrichten ausgelöst werden.

- a) durch eine Einzelwort PDV - Nachricht mit dem für die Statusabfrage entsprechenden Bit - Pattern des Funktionenbytes



- b) durch einen globalen datenlosen CAMAC - Befehl
N 30 A 0 F 1



Die Antwort auf diese Statusabfrage besteht aus zwei PDV - Wörtern. Das 1. Wort enthält im 1. Byte die Crateadresse der entsprechenden Unterstation und im 2. Byte neben den ersten beiden Bits, welche bereits durch das PDV - Protokoll belegt sind, einen Kurzstatus. Abb. 6.1 - 1 zeigt die Bedeutung der einzelnen Bits des 2. Bytes im 1. Wort der Antwort. Das 2. Wort ist ein 16bit Status (Abb. 6.1 - 2), der den Zustand des Controllers, Alarmquellenerkennung und Fehlerlokalisierung ermöglicht. Das 2. Byte des 2. Wortes entspricht dem Inhalt eines Arbeitsspeicherregisters und kann nur durch die Controller - Software verändert werden. Der Zustand des 3. Bits dieses Bytes gibt die Bedeutung des 1. Bytes des 2. Wortes an. Ist der logische Zustand dieses Bits "1", d.h. die vorangegangene Nachricht wurde abgebrochen, so enthält das 1. Byte des 2. Wortes die Wortnummer, bis zu welcher die Übertragung korrekt war.

Möglicher Kurzstatus in der Antwort!

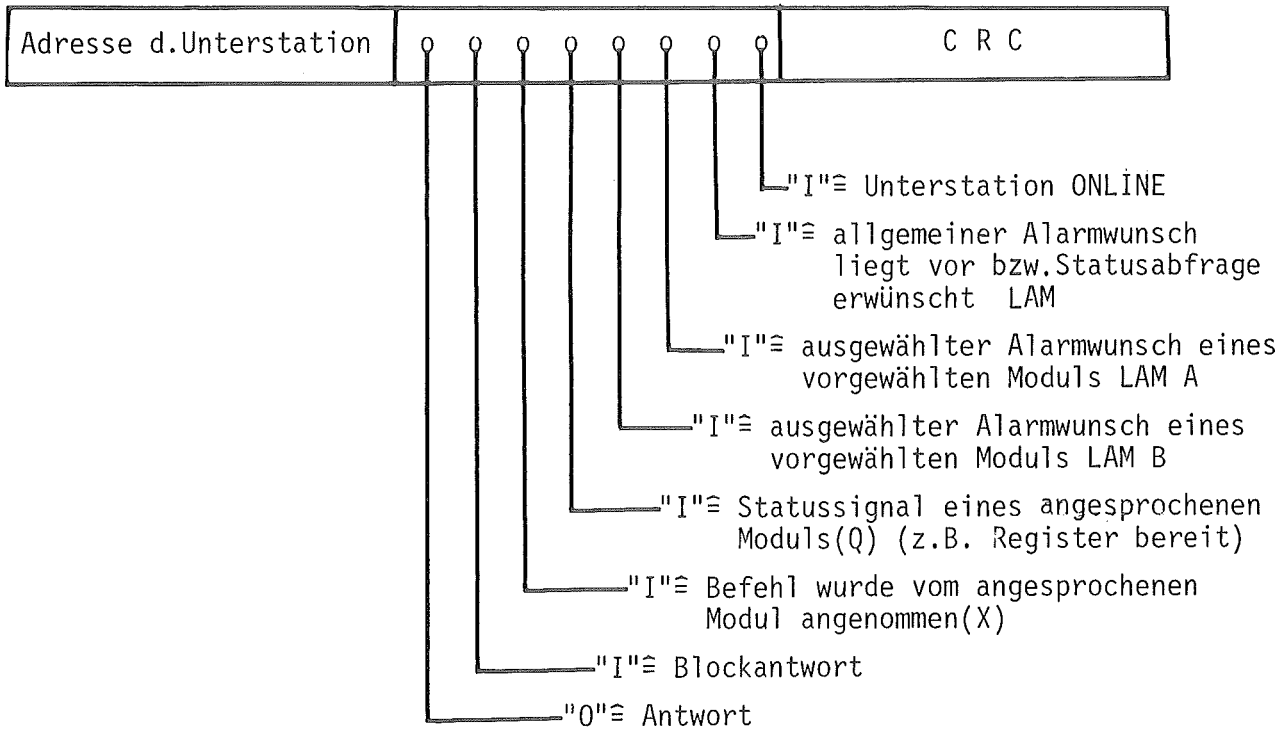


Abb. 6.1-1

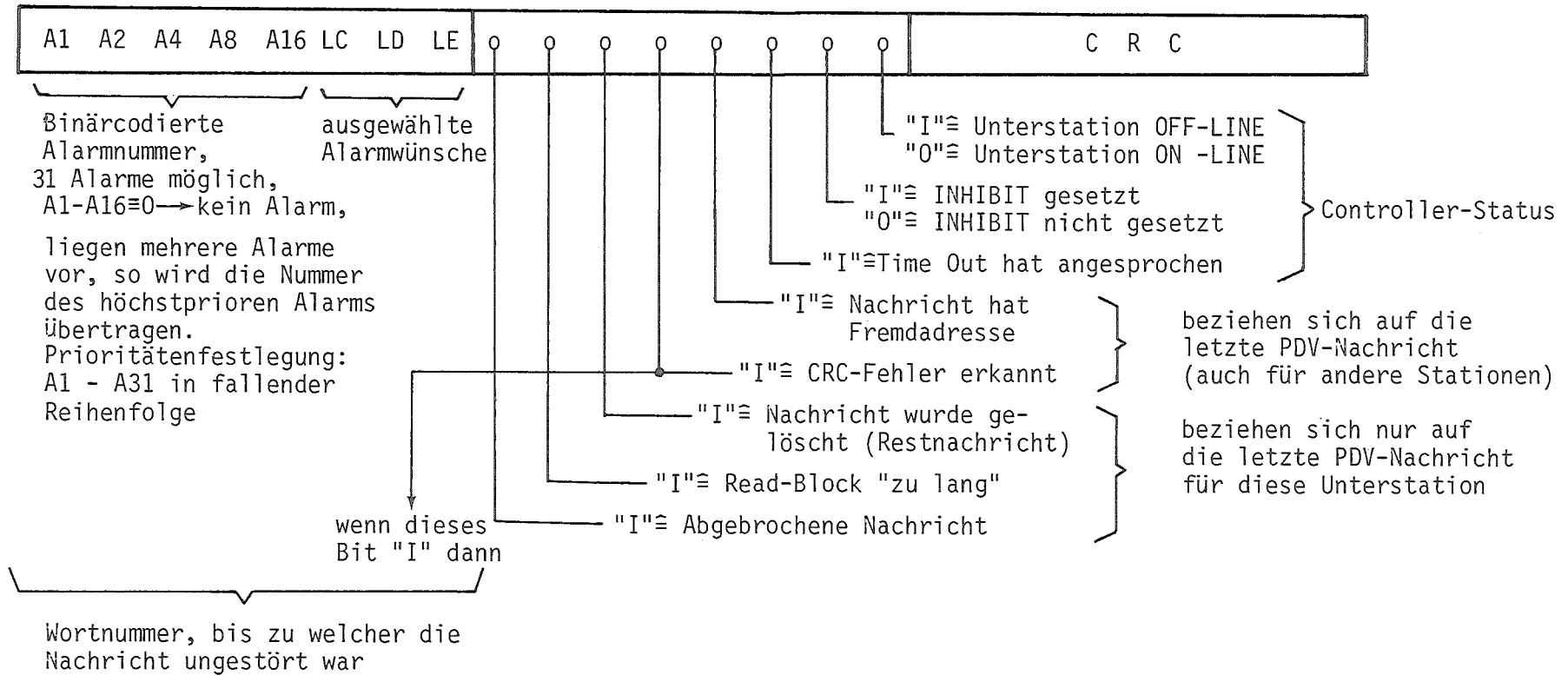


Abb. 6.1-2 Statuswort

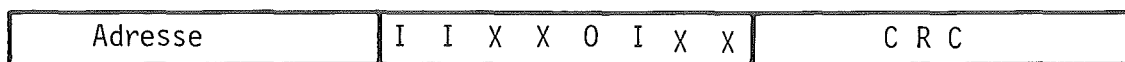
Die Wortnummerierung geschieht dabei in aufsteigender Reihenfolge und beginnt bei der Wortnummer 0 für das 1. PDV - Wort (Crateadress und Funktionenteil als Inhalt). Hat das 3. Bit des 2. Bytes des 2. Wortes den logischen Zustand "0", so enthalten die ersten 5 Bit des 1. Bytes des 2. Wortes die binär codierte Stationsnummer, aus welcher ein Alarm gemeldet wird (LAM - Nummer). Treten mehrere Alarmer gleichzeitig auf, so wird an dieser Stelle die Stationsnummer des höchstpriorären LAM's angezeigt. Die Prioritätenfestlegung ist dabei von L1 bis L24 in fallender Reihenfolge. Mit den restlichen verbleibenden 3 Bits des 2. Wortes werden 3 ausgewählte LAM - Signale (graded LAM's) übermittelt. Die Auswahl der insgesamt 5 graded LAM - Signale geschieht über Jumper auf dem rückwärtig angebrachten ACB - Anschluß. In Abb. 6.1 - 2 ist die Bedeutung der einzelnen Bits der Statusantwort nochmals dargestellt.

6.2 Read LAM - Pattern

Die Funktion zum Lesen des LAM - Patterns ist eine ausschließlich in den CAMAC - Spezifikationen beschriebene Funktion und wird durch den datenlosen CAMAC - Befehl

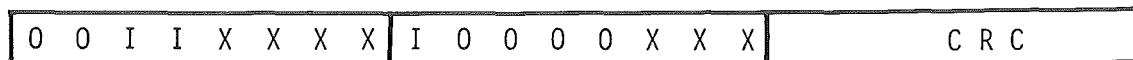
N 3o A 12 F 1

ausgelöst. Die hierfür erforderliche PDV - Nachricht hat folgendes Aussehen:



Blocklänge=2

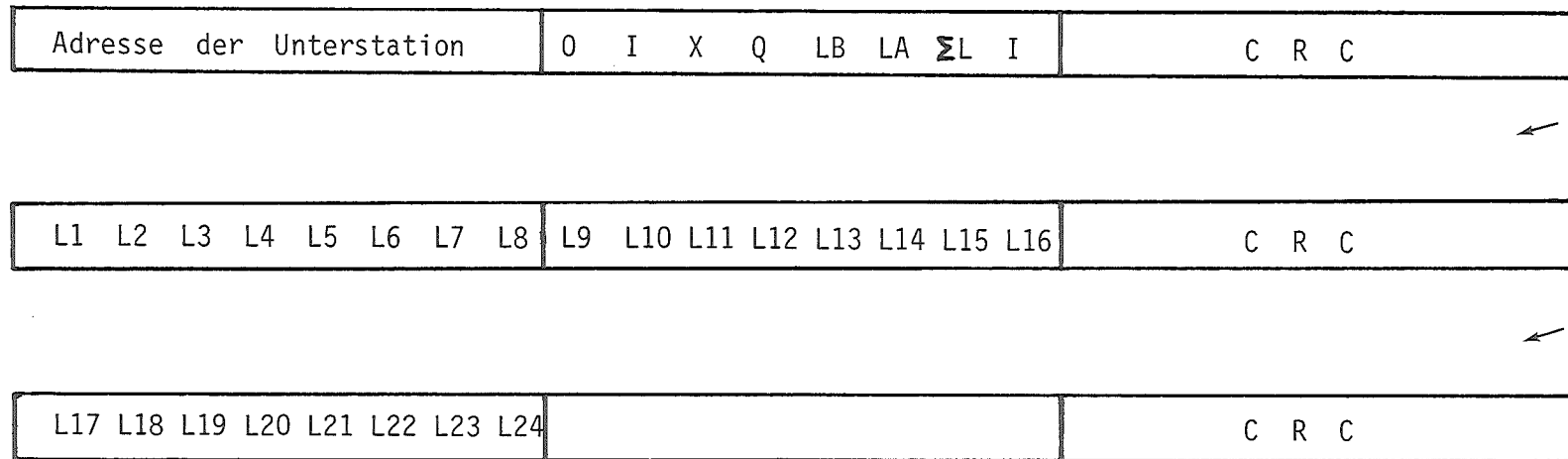
N 30



A 12

F 1

In der Antwort auf diese Nachricht werden die LAM - Signale L1 bis L24 in den ersten 3 Datenbytes eingesetzt. Das 4. Datenbyte, also das 2. Byte des 3. PDV - Wortes der Antwort, entspricht dem 2. Byte des 2. Wortes der Antwort auf die Statusabfrage. Die Bedeutung der einzelnen Bits dieses Bytes kann der Abb. 6.1 - 2 des Kapitels 6.1 entnommen werden. Abb. 6.2 - 1 zeigt die gesamte, auf die Nachricht zum Lesen des LAM - Patterns generierte Antwort.



2. Byte des 2. Wortes der Statusantwort

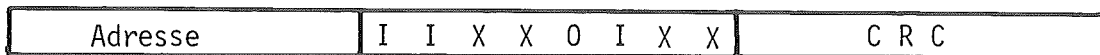
Abb. 6.2-1 Bit-Zuordnung der PDV-Antwort auf der Nachricht "Read LAM Pattern" (N30 A12 F1)

6.3 Write Status

Durch die CAMAC - Funktion

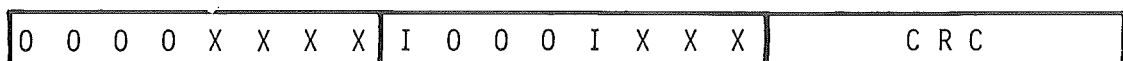
N 30 A 0 F 17

kann der Inhalt des Statusregisters überschrieben werden. Das Überschreiben bzw. Verändern des Statusregisters (6.4 und 6.5) beschränkt sich auf das zweite Byte der gesamten Statusinformation, da nur diese als Register des Arbeitsspeichers realisiert ist und durch die Kommunikationssoftware verändert werden kann. Da die zum Überschreiben des Statusregisters erforderliche Information als Datenbyte in der Nachricht übertragen werden muß, sieht die entsprechende PDV - Nachricht zum Überschreiben des Statusregisters folgendermaßen aus:



Blocklänge=3

N 30



A0

F 17



Datenbyte, welches

2. Statusbyte überschreibt

Damit ist es möglich, einen bestimmten Status des Controllers herzustellen (z.B. Crate ON bzw. Crate OFF). Als Antwort auf diese Nachricht wird lediglich die nach der PDV - Beschreibung erforderliche Minimum - Antwort, bestehend aus Adressbyte, Kurzstatusbyte und Sicherungsteil (A,K,S), generiert.

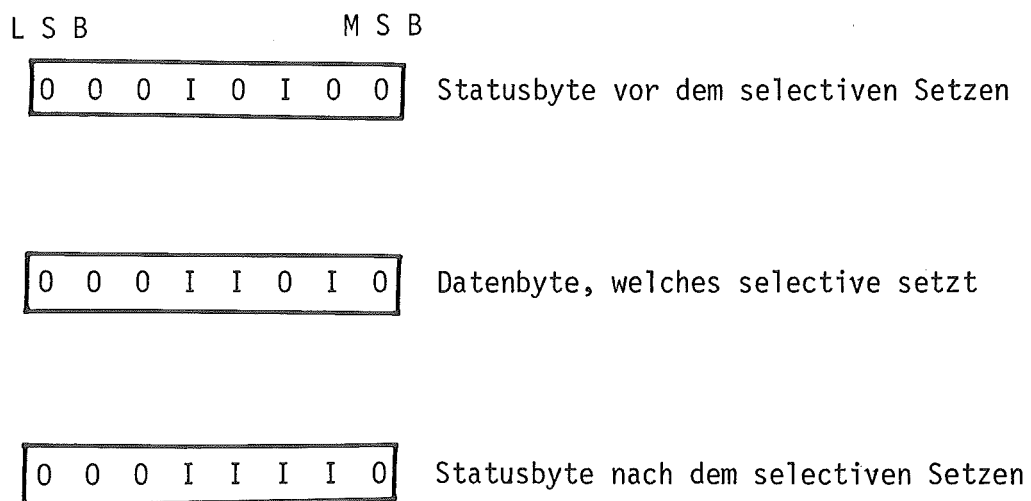
6.4 Selective Set Status

Die CAMAC - Funktion

N 30 A 0 F 19

erlaubt ein selectives Setzen einzelner Bits des Statusregisters (nur innerhalb des 2. Bytes). Die erforderliche PDV - Nachricht hat die gleiche Struktur wie im Pkt. 6.3 zum Überschreiben des Statusregisters. Im Gegensatz zum gesamten Überschreiben werden beim selectiven Setzen nur die Bits des Statusregisters zwingenderweise logisch "1" gesetzt, deren korrespondierende Bits im übertragenen Datenbyte ebenfalls gleich logisch "1" sind. Die anderen Bits des Statusregisters werden in ihrem logischen Zustand nicht verändert.

Beispiel:



Auch hierbei generiert der Controller als Antwort nur ein PDV - Einzelwort (A, K, S).

6.5 Selective Clear Status

Wie bei der CAMAC - Funktion zum selectiven Setzen des Statusregisters werden durch die Funktion

N 30 A 0 F 23

einzelne Bits des Statusregisters verändert. Die entsprechende PDV - Nachricht hat die in Pkt. 6.3 beschriebene Struktur. Im Gegensatz zur Funktion zum selective Setzen des Statusregisters werden hierbei die Bits des Statusregisters gleich logisch "0" gesetzt, deren korrespondierende Bits im übertragenen Datenbyte gleich "1" sind. Alle anderen Bitstellen bleiben unverändert.

Beispiel:

0	0	I	I	0	I	I	0
---	---	---	---	---	---	---	---

 Statusbyte vor dem selectiven clearen

0	0	0	I	I	0	I	0
---	---	---	---	---	---	---	---

 Datenbyte, welches selectiv cleared

0	0	I	0	0	I	0	0
---	---	---	---	---	---	---	---

 Statusbyte nach dem selectiven clearen

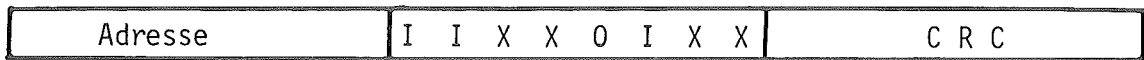
Die Antwort hierauf ist wiederum ein PDV - Einzelwort (A, K, S).

6.6 Test Inhibit

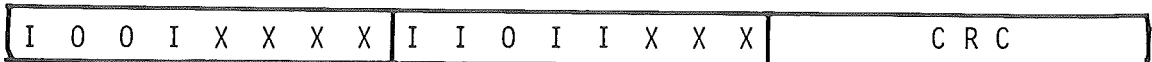
Mit der datenlosen CAMAC - Funktion

N 30 A 9 F 27

kann der logische Zustand des Inhibits überprüft bzw. gelesen werden. Die Struktur der PDV - Nachricht zur Auslösung dieser CAMAC - Funktion hat folgendes Aussehen:

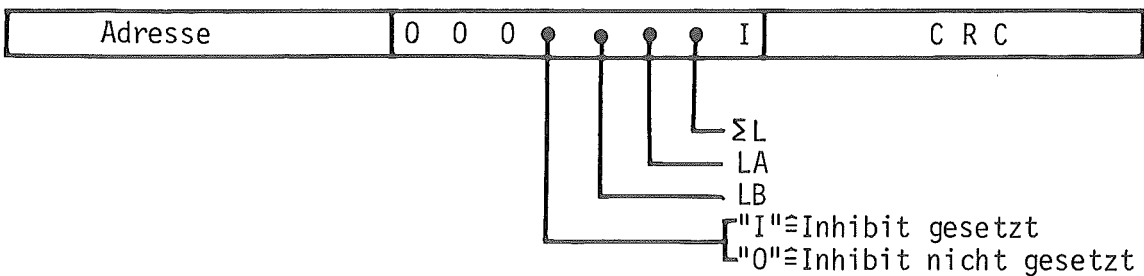


Blocklänge=2 N 30



A 9 F 27

Auf diese Nachricht wird vom Controller eine Antwort generiert, die aus einem PDV - Einzelwort besteht. Im Kurzstatusbyte dieses Einzelwortes gibt das 4. Bit den Zustand des Inhibits an. Das 3. Bit in diesem Kurzstatusbyte wird logisch "0" gesetzt. D.h. das normalerweise im Kurzstatus enthaltene Echosignal Q wird zum Testen des Inhibits herangezogen.



7. Hardware - Beschreibung

Neben der reinen Beschreibung des konstruktiven mechanischen und elektronischen Aufbaus werden in diesem Kapitel auch die Voraussetzungen und Kriterien aufgeführt, die zur vorliegenden Lösung geführt haben. Von besonderer Bedeutung für die peripheren Anschlußmöglichkeiten ist hierbei der Einsatz eines Mikroprozessors als zentrales Steuerungselement und FIFO - Bausteine (First in - first out Register) als Dateninterface zur SDS - Schnittstelle.

7.1 Aufgaben des Controllers

Als Koppellement zwischen dem parallelen Datenweg des CAMAC - Crates und der SDS - Schnittstelle des PDV - Bussystems muß der Controller neben der physikalischen Anpassung und der Datenübertragung auch den prozeduralen Ablauf und die Funktionen beider Übertragungssysteme beherrschen. Dabei ist der von den Übertragungssystemen abhängige prozedurale Teil der für den Controller Aufgabenintensivste. Für den PDV - CAMAC Controller ergeben sich im Einzelnen folgende Aufgabenbereiche:

- a) Wandlung der seriellen SDS - Daten in parallele CAMAC - Daten
- b) Wandlung der parallelen CAMAC - Daten in serielle SDS - Daten
- c) CRC - Kontrolle und CRC - Generierung
- d) Vergleich des ersten Bytes einer Nachricht mit der eigenen Adresse (Crate - Adresse)
- e) Interpretation des Funktionenteils (2. Byte einer Nachricht)
- f) Verteilung der empfangenen Informationen entsprechend ihren Bedeutungen
- g) Ausblenden fehlerhafter Übertragungen
- h) Generierung von CAMAC - Zyklen
- i) Zusammenstellung einer Antwort
- k) Absetzen einer Antwort
- l) Übertragungsfehlerbearbeitung
- m) Auslösen globaler Crate - Funktionen
- n) Statusbehandlung
- o) Sammeladresserkennung
- p) Einleitung einer Sonderprozedur
- q) Beenden der Sonderprozedur
- r) Alarmabgabe

- s) Zwischenspeichern gelesener Daten (notwendig bei wiederholender Lesefunktion)
- t) ACB - Funktionen
- u) manuelles Eingreifen (OFFLINE, Z, C)

7.2 Lösung mit Mikroprozessor

Zur Realisierung der in 7.1 genannten Aufgaben e,f,l,p,q und s des Controllers wird bei einer reinen Hardware - Lösung eine große Anzahl von integrierten Schaltkreisen benötigt. Dadurch entstehen

- a) räumliche Probleme für einen doppelt breiten Controller
- b) Stromversorgungsprobleme, da bei zu hoher Schaltkreisdichte die zulässige Strombelastung der Steckerleiste überschritten wird
- c) zu aufwendiges und unübersichtliches Timing der Einzelschaltkreise
- d) sehr spezielle und unflexible Interface - Schaltung, die auch kleine Änderungen nur mit sehr großem Arbeitsaufwand ermöglicht

Zur Vermeidung aller dieser Nachteile bietet sich der Einsatz eines Mikroprozessors an. Er dient hierbei in erster Linie als Hardwareersatz, ermöglicht aber durch die Software - Bearbeitung der anfallenden Aufgaben ein sehr flexibles Verhalten des Crate - Controllers. Gleichzeitig wird durch die Bus - Struktur des Mikroprozessors ein gut strukturierter Aufbau erreicht, sodaß bei entsprechender Aufteilung der Baugruppen ein Controller entsteht, der durch Auswechseln einer Interfacekarte auch für andere Datenübertragungssysteme (z.B. serielle CAMAC - Ringleitung) einsatzbereit gemacht werden kann.

7.3 Auswahlkriterien für 8 X 300

Durch die Verlagerung vieler Aufgaben in die Software wird mit der unvermeidbaren Zykluszeit eines Mikroprozessors der Zeitraum zwischen empfangener Nachricht und Ausführung stark vergrößert. Damit können bei hohen Übertragungsraten und zeitkritischen Prozeduranteilen Verzögerungen entstehen, die entweder von der zentralen Steuerstation zugelassen oder durch prozedurale- bzw. Hardwareänderungen aufgefangen werden.

Weiterhin ist es im Allgemeinen für einen Prozessor mit fester Datenlänge schwierig bzw. zeitaufwendig einzelne Bitstellen eines Datenwortes zu überprüfen und entsprechende Reaktionen davon abzuleiten.

Bei der Wahl eines geeigneten Mikroprozessors für den vorgenannten Aufgabenbereich kommt es also weniger auf datenverarbeitungstechnische Grundinstruktionen an als in erster Linie auf kurze Zykluszeiten und einen für Hardwareersatz geeigneten Mikrobefehlsatz. Aufgrund dieser Auswahlkriterien wurde für den realisierten PDV - CAMAC Controller der von der Firma VALVO vertriebene Mikroprozessor 8 X 300 eingesetzt /4/. Dieser Mikroprozessor gehört mit einer Zykluszeit von 250ns, die auch für die peripheren Schaltkreise bzw. Signale zur Geltung kommen, zu den schnellsten zur Zeit verfügbaren Mikroprozessoren. Mit seinem in Tabelle 7.3-1 dargestellten Instruktionssatz läßt sich auch eine sehr effektive Einzelbitprüfung und -manipulation durchführen. Durch die Trennung von Programmspeicher und Arbeitsspeicher sowie die leichte Zugänglichkeit über den IV - Bus (Interface - Vektor - Bus) und die entsprechenden Kontrolleitungen läßt er sich sehr leicht mit anderen externen Bausteinen und Baugruppen verbinden. Außerdem stehen durch die sogenannten IV - Bytes (Interface Vektor Chips) spezielle für diesen Prozessortyp zugeschnittene Bausteine zur Verfügung, mit denen sich ohne weitere externe Schaltelemente Ein- bzw. Ausgabeports realisieren lassen.

Abb. 7.3-2 und 7.3-3 zeigen das Blockschaltbild des Prozessors und eine prinzipielle Zusammenschaltung zwischen Programmspeicher und Ein- Ausgabeports.

- 54 -
Instruction Set Summary

INSTRUCTION MNEMONIC	OP CODE	FORMATS	DESCRIPTION	I/O CONTROL SIGNALS	INSTRUCTION CYCLE											
					INSTRUCTION INPUT AND DATA PROCESSING	ADDRESS/IV BUS OUTPUT										
MOVE	0	Register to Register <table border="1"> <tr><td>0</td><td>2 3</td><td>7 8</td><td>10 11</td><td>15</td></tr> <tr><td>0</td><td>S</td><td>R/L</td><td>D</td><td></td></tr> </table> S=07,17,20-37 _g D=10,20-37 _g	0	2 3	7 8	10 11	15	0	S	R/L	D		(S) - D Move contents of register specified by S to register specified by D. Right rotate contents of register S by R/L places before operation.	SC=0 WC=0 LB/RB=1 if D=17	0 0 1 if D=17	1 if D=07,17 0 1 if D=17
		0	2 3	7 8	10 11	15										
		0	S	R/L	D											
		IV Bus to Register: <table border="1"> <tr><td>0</td><td>2 3</td><td>7 8</td><td>10 11</td><td>15</td></tr> <tr><td>0</td><td>S</td><td>R/L</td><td>D</td><td></td></tr> </table> S=20-37 _g D=10,20-37 _g	0	2 3	7 8	10 11	15	0	S	R/L	D		Move right rotated IV bus (source) data specified by S to register specified by D. R/L specifies the length of source data with most significant bits set to zero	SC=0 WC=0 LB/RB=0 if S=20-27 1 if S=30-37	0 0 0 if S=20-27 1 if S=30-37	1 if D=07,17 0 1 if D=17 1 if D=17
0	2 3	7 8	10 11	15												
0	S	R/L	D													
Register to IV Bus: <table border="1"> <tr><td>0</td><td>2 3</td><td>7 8</td><td>10 11</td><td>15</td></tr> <tr><td>0</td><td>S</td><td>R/L</td><td>D</td><td></td></tr> </table> S=07,17,20-37 _g D=20-37 _g	0	2 3	7 8	10 11	15	0	S	R/L	D		Move contents of register specified by S to the IV bus. Before placement on IV bus, data is shifted as specified by D, and R/L bits merged with existing IV bus data.	SC=0 WC=0 LB/RB=X LB/RB=X	0 1 X X	0 1 0 if D=20-27 1 if D=30-37		
0	2 3	7 8	10 11	15												
0	S	R/L	D													
IV Bus to IV Bus: <table border="1"> <tr><td>0</td><td>2 3</td><td>7 8</td><td>10 11</td><td>15</td></tr> <tr><td>0</td><td>S</td><td>R/L</td><td>D</td><td></td></tr> </table> S=20-37 _g D=20-37 _g	0	2 3	7 8	10 11	15	0	S	R/L	D		Move right rotated IV bus data (source) specified by S to the IV bus. Before placement on IV bus, data is shifted or specified by D and R/L specifies the length of source data and of destination data merged with existing IV bus data.	SC=0 WC=0 LB/RB=0 if S=20-27 1 if S=30-37	0 1 0 if S=20-27 1 if S=30-37	0 1 0 if D=30-37 1 if D=30-37		
0	2 3	7 8	10 11	15												
0	S	R/L	D													
ADD	1	SAME AS MOVE	(S) plus (AUX) - D Same as MOVE but contents of AUX ADDED to the source data. If carry from most significant bit then OVF=1 otherwise OVF=0.		SAME AS MOVE											
AND	2	SAME AS MOVE	(S) & (AUX) - D Same as MOVE but contents of AUX ADDED with source data.		SAME AS MOVE											
XOR	3	SAME AS MOVE	(S) + (AUX) - D Same as MOVE but contents of AUX exclusive Ored with source data.		SAME AS MOVE											
XEC	4	Register Immediate: <table border="1"> <tr><td>0</td><td>2 3</td><td>7 8</td><td>15</td></tr> <tr><td>4</td><td>S</td><td>I</td><td></td></tr> </table> S=07,17,20-37 _g I=000-377 _g	0	2 3	7 8	15	4	S	I		Execute instruction at current page address offset by I + (S).	SC=0 WC=0 LB/RB=X	0 0 X	0 0 X		
		0	2 3	7 8	15											
4	S	I														
IV Bus Immediate: <table border="1"> <tr><td>0</td><td>2 3</td><td>7 8</td><td>10 11</td><td>15</td></tr> <tr><td>4</td><td>S</td><td>R/L</td><td>I</td><td></td></tr> </table> S=20-37 _g I=00-37 _g	0	2 3	7 8	10 11	15	4	S	R/L	I		EXECUTE the instruction at the address determined by concatenating 8 high order bits of PC with the 5 bit sum of I and rotated IV bus data (source) specified by S. R/L specifies length of source data with most significant bits set to zero. PC is not incremented.	SC=0 WC=0 LB/RB=0 if S=20-27 1 if S=30-37	0 0 0 if S=20-27 1 if S=30-37	0 0 X X		
0	2 3	7 8	10 11	15												
4	S	R/L	I													
NZT	5	Register Immediate: <table border="1"> <tr><td>0</td><td>2 3</td><td>7 8</td><td>15</td></tr> <tr><td>5</td><td>S</td><td>I</td><td></td></tr> </table> S=07,17,20-37 _g I=000-377 _g	0	2 3	7 8	15	5	S	I		If (S) ≠ 0, jump to current page address offset by I; otherwise PC = I + PC	SC=0 WC=0 LB/RB=X	0 0 X	0 0 Y		
		0	2 3	7 8	15											
5	S	I														
IV Bus Immediate: <table border="1"> <tr><td>0</td><td>2 3</td><td>7 8</td><td>10 11</td><td>15</td></tr> <tr><td>5</td><td>S</td><td>R/L</td><td>I</td><td></td></tr> </table> S=20-37 _g I=00-37 _g	0	2 3	7 8	10 11	15	5	S	R/L	I		If contents of register specified by S is Non Zero then Transfer to address determined by concatenating 5 high order bits of PC with I; otherwise increment PC. If right rotated IV bus data (source) is Non Zero then Transfer to address determined by concatenating 8 high order bits of PC with I; otherwise increment PC.	SC=0 WC=0 LB/RB=0 if S=20-27 1 if S=30-37	0 0 0 if S=20-27 1 if S=30-37	0 0 X X		
0	2 3	7 8	10 11	15												
5	S	R/L	I													
XMIT	6	Register Immediate: <table border="1"> <tr><td>0</td><td>2 3</td><td>7 8</td><td>15</td></tr> <tr><td>6</td><td>D</td><td>I</td><td></td></tr> </table> S=20-37 _g I=000-377 _g	0	2 3	7 8	15	6	D	I		Transmit I - S TRANSMIT and store 8 bit binary pattern I to register specified by D.	SC=0 WC=0 LB/RB=X	0 0 X	1 if D=07,17 0 1 if D=17		
		0	2 3	7 8	15											
6	D	I														
IV Bus Immediate: <table border="1"> <tr><td>0</td><td>2 3</td><td>7 8</td><td>10 11</td><td>15</td></tr> <tr><td>6</td><td>D</td><td>R/L</td><td>I</td><td></td></tr> </table> S=20-37 _g I=00-37 _g	0	2 3	7 8	10 11	15	6	D	R/L	I		TRANSMIT binary pattern I to IV bus. Before placement on IV bus, literal I is shifted as specified by D and R/L bits merged with existing IV bus data.	SC=0 WC=0 LB/RB=X LB/RB=X	0 0 X X	0 1 0 if S=20-27 1 if S=30-37		
0	2 3	7 8	10 11	15												
6	D	R/L	I													
JMP	7	Address Immediate: <table border="1"> <tr><td>0</td><td>2 3</td><td>15</td></tr> <tr><td>7</td><td>A</td><td></td></tr> </table> A=00000-17777 _g	0	2 3	15	7	A		Jump to Program Address A. JUMP to program storage address A. A is stored in the address register (AR).	SC=0 WC=0 LB/RB=X	0 0 X	0 0 X				
0	2 3	15														
7	A															

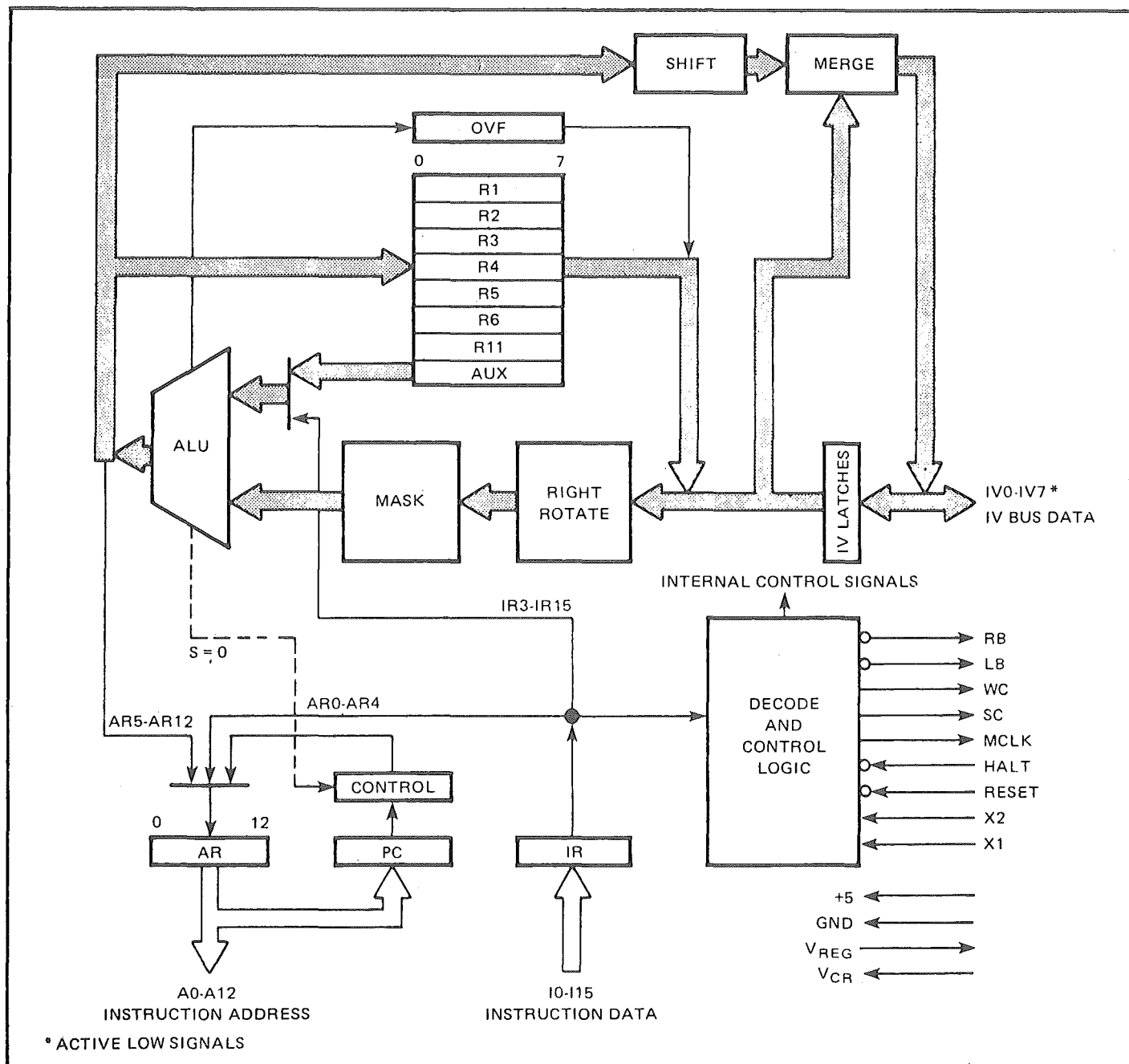


Abb. 7.3-2

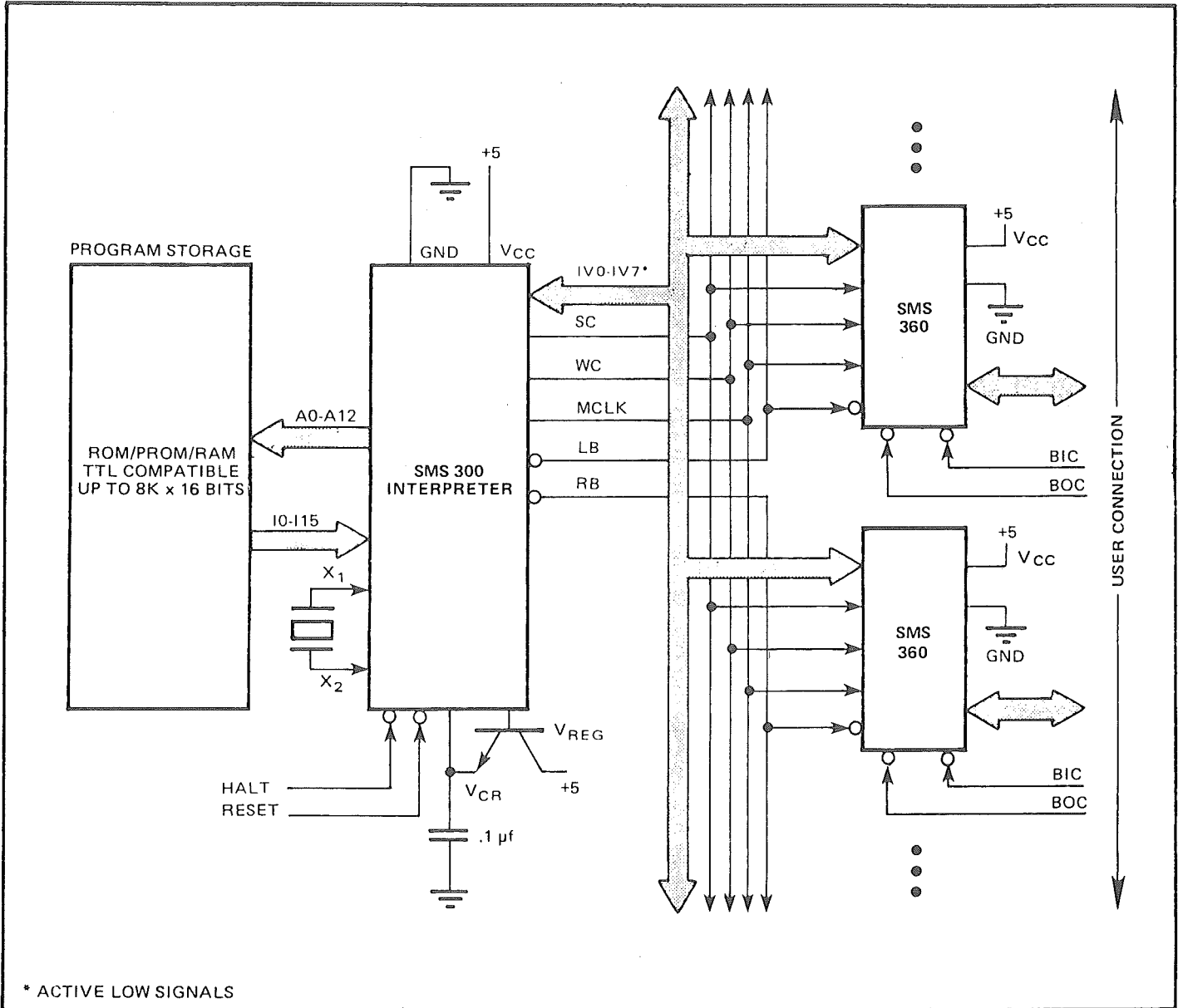


Abb. 7.3-3

7.4 Aufgaben des Mikroprozessors

Durch den im PDV - CAMAC Controller eingesetzten Mikroprozessor 8 x 300 werden hauptsächlich jene Aufgaben durchgeführt, die durch Hardware - Schaltungen nicht oder nur mit sehr großem Aufwand hätten erfolgen können. Gleichzeitig werden auch solche Aufgaben vom Prozessor übernommen, die nicht zeitkritisch sind und die den Prozessor in seinen anderen Aufgabengebieten nicht beeinträchtigen. Insgesamt übernimmt der Mikroprozessor folgende Aufgaben:

- a) Übertragen von SDS nach CAMAC und umgekehrt
- b) CRC - Kontrolle und CRC - Generierung
- c) Zählen und Verteilen der empfangenen Wörter
- d) Ausblenden fehlerhafter Übertragungen
- e) Adressenvergleich
- f) Auslösen globaler Crate - Funktionen
- g) Statusbehandlung
- h) LAM - Auslesen und Behandlung
- i) Bei Blockübertragung mit Fehler Ausblenden aller Daten ab Fehlerwort und bei Wiederholung der Nachricht Fortsetzen der Übertragung ab Fehlerwort der 1. Nachricht
- k) Bei Blocklesen aus CAMAC - Modulen zwischenspeichern der Daten im Arbeitsspeicher zur Wiederholung der Antwort wenn Übertragungsfehler vorlagen
- l) Testroutinen

8. Blockschaltbild

Abb. 8 - 1 zeigt das vereinfachte Blockschaltbild des Mikroprozessor gesteuerten Crate - Controllers. Anhand dieser Blockdarstellung lassen sich die wesentlichsten Baugruppen des Controllers erkennen, die mit Hilfe des Mikroprozessor - Bussystems untereinander verbunden sind, und eine klare Strukturierung des Controllers erkennen lassen. Der mit PROM bezeichnete Programmspeicher des Mikroprozessors hat eine Kapazität von 1 K Programminstruktionen und ist aus schnellen PROM - Bausteinen mit 45ns accesstime aufgebaut. Er ist über die Addresslines (10 Leitungen) und die Instructionslines (16 Leitungen) mit dem Mikroprozessor 8 X 300 verbunden. Der mit RAM bezeichnete Arbeitsspeicher besitzt eine Kapazität von 256 Bytes und ist als selbstständige Baugruppe an den Prozessorbus angeschlossen. Die mit S/P bezeichneten Blöcke zur Ankopplung an die SDS - Schnittstelle sind die mit FIFO - Bausteinen realisierten Seriell-Parallel - bzw. Parallel - Seriellwandler und erfüllen außerdem die Funktion eines Datenzwischenpuffers. Durch die, aus dem Mikroprozessor ausgelagerte Parallel - Seriell - bzw. Seriell - Parallelwandlung und der Zwischenspeicherfunktion wird der Prozessor entlastet und größere zusammenhängende Programmsegmente ermöglicht. Die Ankopplung des CAMAC - Datenweges an den Mikroprozessorbus geschieht über Lese- bzw. Schreibregister, die als Register des RAM - Arbeitsspeichers angesehen werden und einen Teil des Adressvorrates des Arbeitsspeichers überdecken. Durch den Aufbau des Arbeitsspeichers mit RAM - Bausteinen der Organisation 256 mal 1 Bit sind diese Adressen im Arbeitsspeicher nochmals vorhanden und müssen durch entsprechende Schaltungsauslegung bei Adressierung der Lese- bzw. Schreibregister zum CAMAC - Datenweg abgeschaltet werden (lediglich bei der Lesefunktion erforderlich). Die für den gesamten Arbeitsablauf des Controllers erforderlichen Kontroll- und Steuersignale werden über entsprechende Hardware- Schaltungsgruppen und ein Interfacevektorbyte - Chip mit dem Mikroprozessorbus verbunden und so der Logik des Mikroprozessors zugänglich gemacht.

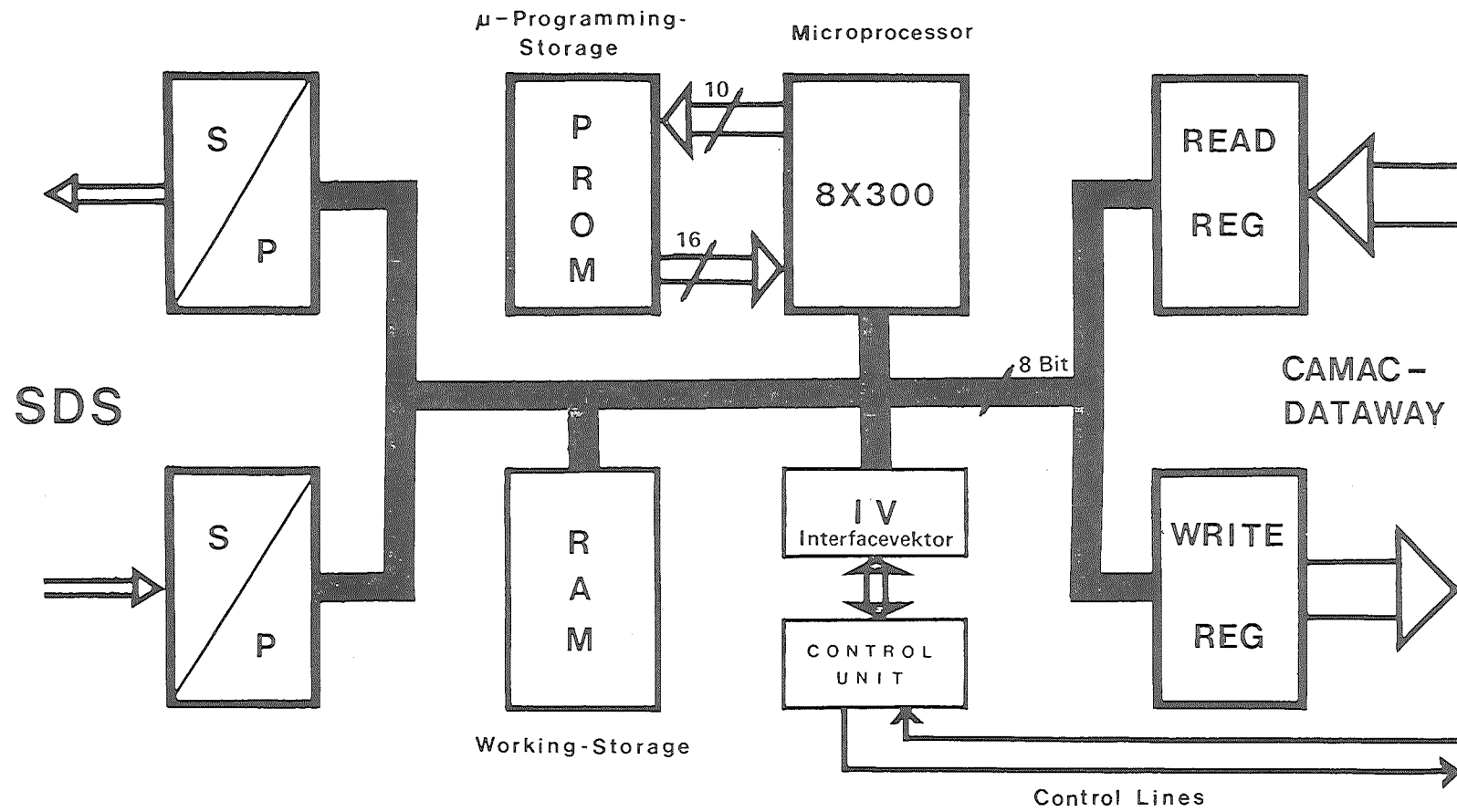


Abb. 8-1 Vereinfachtes Blockschaltbild

8.1 Prozessorbaugruppe

Das Kernstück des PDV - CAMAC Controllers ist die Prozessorbaugruppe mit dem bipolaren Prozessor 8 X 300 und entsprechenden PROM - Bausteinen. Zur Bildung von 1 K Programminstruktionskapazität werden 4 Bausteine des Typs 82S115 eingesetzt. Da der direkte Programmadressumfang des 8 X 300 Prozessors 8 K beträgt stehen entsprechend dieser Programmspeicherkapazität 13 Adressleitungen mit der Bezeichnung A0 bis A12 zur Verfügung. A12 ist hierbei die Adressleitung mit der niedrigsten Wertigkeit (LSB) und die Adressleitung mit der Bezeichnung A0 diejenige mit höchster Wertigkeit (MSB). Da der für den Controller realisierte Programmspeicher lediglich 1 K beträgt werden von den 13 vorhandenen Adressleitungen nur 10 benutzt. Die Adressausgänge des Prozessors mit der Bezeichnung A0, A1 und A2 werden nicht benutzt. Die Instruktionslänge des Mikroprozessors beträgt 16 bit. Aus diesem Grunde sind 16 Instruktionsleitungen erforderlich, die mit der Bezeichnung I15 bis I0 versehen sind und bei welchen die Leitung mit der Bezeichnung I0 die Leitung mit der höchsten Wertigkeit (MSB) darstellt.

In der Stromversorgung für den Prozessor ist ein Längsregeltransistor Typ 2N5320 eingesetzt. Die für die Steuerung dieses Regeltransistors benötigte Referenzspannung wird dem Prozessor am Anschlußpin 50 entnommen. Das für die Funktion des Prozessors erforderliche Taktsignal wird als Gegentaktsignal mit der Bezeichnung X1 und X2 den Anschlußpins 10 und 11 zugeführt. Das zugeführte Clocksignal besitzt eine Frequenz von 5 MHz woraus sich für den Prozessor eine Zykluszeit von 400ns ergibt. Damit läßt sich mit der realisierten Hard- und Software eine Verarbeitung und Generierung von seriellen Daten an der SDS - Schnittstelle bis zu 1 MHz realisieren. Der Prozessorbus besteht aus den 8 Signalleitungen mit der Bezeichnung IV7 bis IV0 wobei das Signal mit der Bezeichnung IV0 das höchstwertigste Bussignal darstellt (MSB). Der Prozessorbus dient zur Übertragung von Adressen und Daten welche in logisch LOW aktiver Form allen angeschlossenen Baugruppen zur Verfügung stehen. Am Anschlußpin 42 des Mikroprozessors steht als Ausgangssignal ein Clocksignal (MCLK = Masterclock) zur Verfügung, welches von den Clocksignalen X1 und X2 abgeleitet ist und mit seiner Phasenlage die Funktion des Prozessorbus-systems (Input - bzw. Output) angezeigt.

Die vier weiteren Steuerausgangssignale des Prozessors WC, SC, RB und LB stehen ebenfalls in festem Phasenbezug zur Masterclock. Damit sind alle externen Funktionen und Prozessorbusoperationen synchrone Funktionen und erlauben damit das unkomplizierte Ansteuern externer Baugruppen.

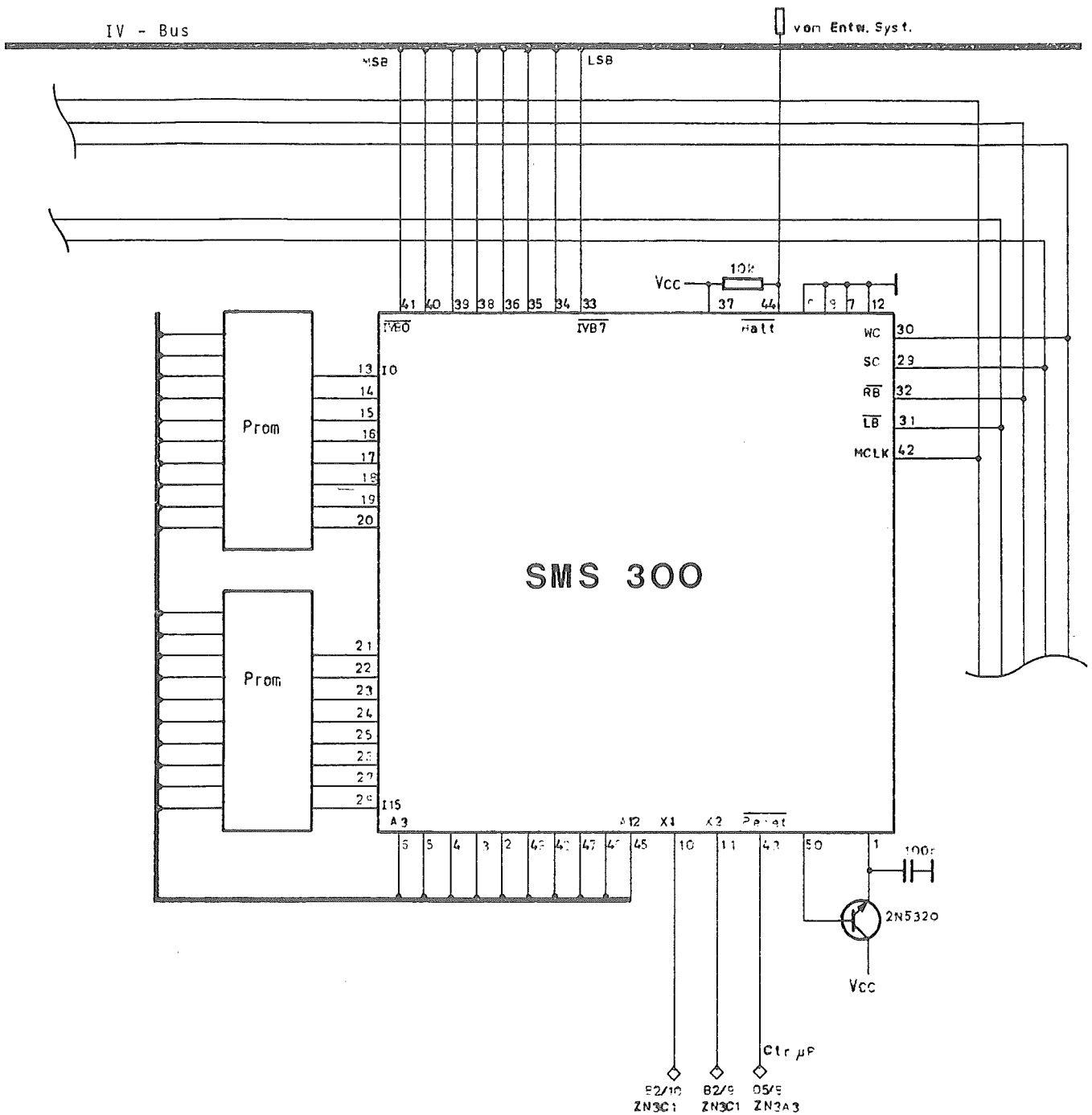


Abb. 8.1-1 Prozessorbaugruppe

8.2 Anschluß an SDS - Schnittstelle

Die physikalische Ankopplung zur SDS - Schnittstelle geschieht mit Bausteinen des Typs SN75115 (S4 und S5) bzw. SN75113 (S7), die das Umsetzen des Logikpegels und der Signalstruktur (Gegentaktsignal an der SDS - Schnittstelle zu TTL - Signal) bewerkstelligen. Die für den Dateneingang erforderliche Seriell - Parallelwandlung wird mit Hilfe der FIFO - Bausteine 9403 (P3 und R3) vorgenommen, die mit ihrem parallelen Ausgang direkt am Prozessorbus angeschlossen sind und eine Zwischenspeicherung von 16 Bytes erlauben (Abb. 8.2 - 1). Mit Hilfe der Bausteine P7, H2, Q7, M7, O6 und I7 werden die für den Prozessor benötigten Hilfssignale (Eingangsdaten stehen an und Input - FIFO - Bausteine sind zurückgesetzt) erzeugt. Diese Bausteine bewerkstelligen auch ein automatisches Zurücksetzen der FIFO - Bausteine (Masterreset). Für die Verarbeitung der Ausgangssignale (Antworten) stehen ebenfalls 2 FIFO - Bausteine P4 und R4 zur Verfügung, die vom Prozessorbus mit 8 bit parallelen Daten versorgt werden und die Parallel - Seriellwandlung vornehmen. Die nach den PDV - Vorschlägen erforderliche Signalstruktur für Antworten macht einen 1 bit Zwischenspeicher erforderlich. Dieser Zwischenspeicher wird durch das Flipflop R6 dargestellt, welches in seiner Funktion ein 1 bit Schieberegister darstellt. Mit Hilfe des zweiten Flipflop Bausteines R6 wird das benötigte Sendebereitschaftssignal SB aufgesetzt. Sobald alle in die FIFO - Bausteine geladenen Daten durch empfangene Sendetakte ausgelesen sind wird über den Baustein Q6 (SN49703) das Sendebereitschaftssignal zurückgesetzt. Damit werden solange Sendetakte vom Buskoppler generiert bis vom Mikroprozessor keine weiteren Daten in die FIFO - Bausteine geladen werden bzw. eine längere Pause (abhängig von der Datenübertragungsrate an der SDS - Schnittstelle) auftritt. Bei gefülltem FIFO - Baustein und weiteren anstehenden Daten zur Antwort wartet der Prozessor mit dem Ladevorgang solange bis das Eingangsregister der FIFO - Bausteine durch empfangene Sendetakte leergeäumt worden ist und wieder neue Daten aufnehmen kann.

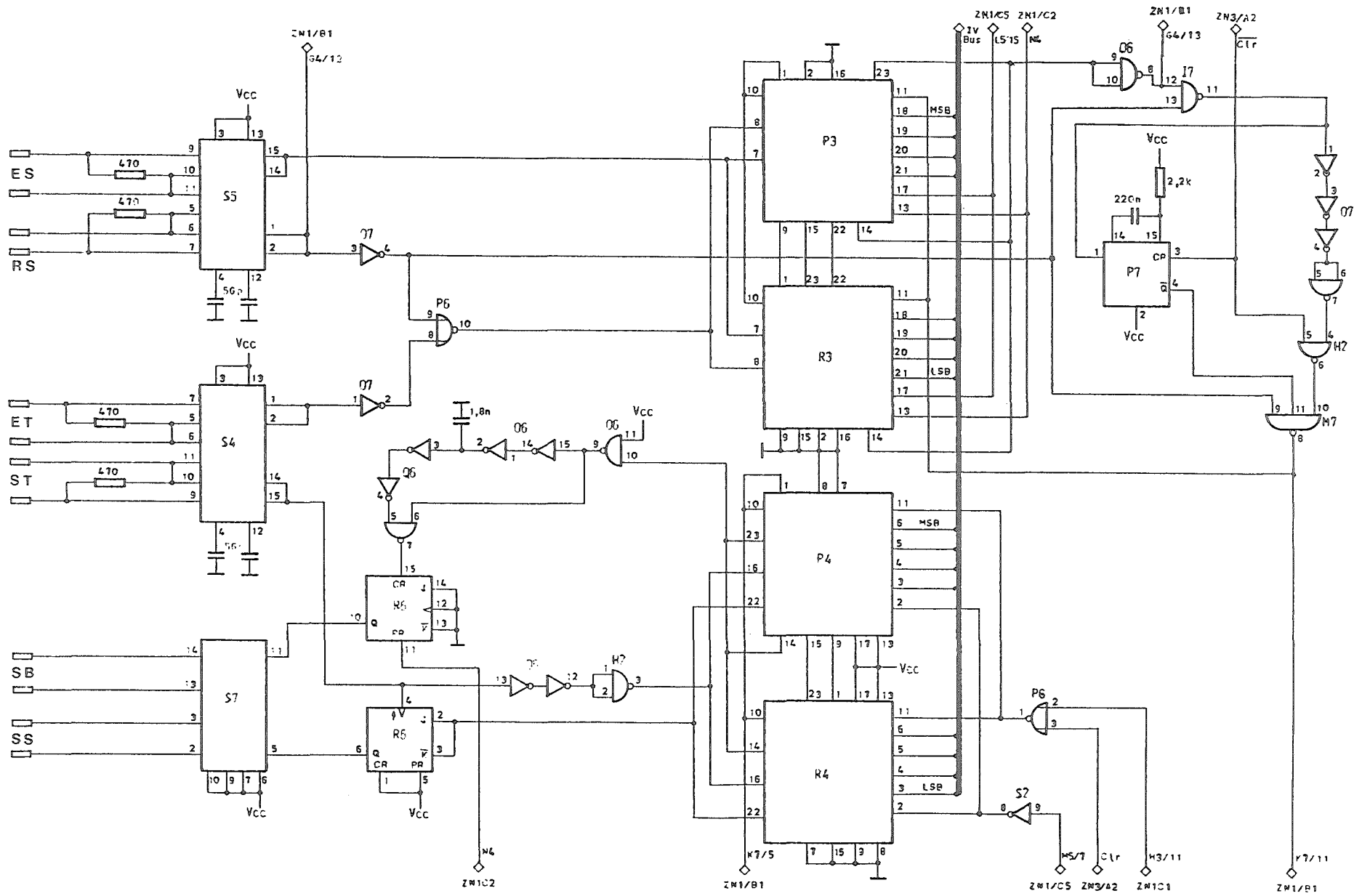


Abb. 8.2 - 1

8.3 Anschluß an CAMAC - Datenweg

Die Ankopplung des CAMAC - Datenweges an den Mikroprozessorbus geschieht wie bereits im Kapitel 8 beschrieben und wie in Abb. 8 - 1 zu sehen mit Hilfe von zwischengeschalteten Registern. Diese Register arbeiten unidirektional und sind entsprechend ihrer Funktion als Lese- bzw. Schreibregister mit ihren Aus- bzw. Eingängen mit dem Prozessorbus verbunden. Für die Aufschaltung der CAMAC - Parallelbus spezifischen Steuersignale (Busy, Strobe 1 und Strobe 2 sowie C = Löschen und Z = initialisieren) sind lediglich Bustreiber erforderlich, die vom CAMAC - Zyklus Timing-generator angesteuert werden.

8.3.1 Read - Register

Entsprechend ihrer Funktion als Datenzwischenspeicher zur Übertragung von parallelen CAMAC - Daten zum Mikroprozessorbus sind hierfür 3 Bausteine des Typs SN74LS373 eingesetzt. Die Eingänge dieser Bausteine sind mit den Read - Leitungen R1 bis R24 sowie den für die Funktion als Hauptcontroller erforderlichen Pull - Up Widerständen verbunden. Als gemeinsames Clocksignal dieser Registerbausteine wird das durch das Controllermodul selbst erzeugte CAMAC - Strobosignal S1 benutzt. Die Tristate - Ausgänge dieser Register sind mit dem Prozessorbus verbunden und können als adressierbare RAM - Zelle angesprochen werden. Wie auch aus Tabelle 8.7 -3 zu entnehmen werden die Register zum Lesen der Readleitungen R1 bis R8 mit der Adresse "0" der rechten Bank angesprochen, die Register zum Lesen der Readleitungen R9 bis R16 mit der Adresse "1" der rechten Bank und die Register zum Lesen der Readleitungen R17 bis R24 mit der RAM - Adresse "2".

8.3.2 Write - Register

Für die Datenübertragung vom Mikroprozessor zum 24bit breiten Datenweg des CAMAC - Crates werden drei 8bit Zwischenregister vom Typ SN74LS273 verwendet, die über Adressierung mit RAM - Adresse "12" (W1 bis W8), "13" (W9 bis W16) und "14" (W17 bis W24) vom Prozessor mit 8bit breiten Datenteilen geladen werden können. Da der von den Writeleitungen nach den CAMAC - Spezifikationen aufzunehmende Mindeststrom von diesen Registerbausteinen nicht aufgenommen werden kann sind Leitungstreiber vom Typ SN74S241 nachgeschaltet. Diese Leitungstreiber haben Tristate - Ausgänge und werden durch das mit der Schreibfunktion von CAMAC - Daten verknüpfte eigene Busy - Signal bzw. $\overline{\text{Busy}}$ - Signal (B' und \overline{B} ') durchgeschaltet. Die für die Writeleitungen erforderlichen Pull - Up Widerstände sind durch Widerstandsarrays (jeweils 16 Pull - Up Widerstände) realisiert. Abb. 8.3.2 - 1 zeigt den Schaltungsauszug mit den für die Kopplung der Writeleitungen des CAMAC - Crates mit dem Mikroprozessorbus (IV - Bus) erforderlichen Elemente.

8.3.3 LAM - Register

Zum Lesen des 24bit breiten LAM - Pattern (Leitungen L1 bis L24) sind wie zum Lesen der CAMAC - Daten 3 Registerbausteine vom Typ SN74LS373 eingesetzt (vergleiche Kap. 8.3.1 Read - Register). Ebenso werden entsprechende Pull - Up Widerstände benötigt. Im Gegensatz zu den Read - Registern wird für die Kontrollfunktion aller LAM - Signale noch ein zusätzliches Summen - LAM Signal erzeugt, welches die logische "ODER" - Verknüpfung aller 24 LAM - Signale darstellt. Diese logische "ODER" - Verknüpfung wird mit Hilfe der Bausteine H6 und H7 vorgenommen und über ein auf der Frontplatte angeordnetes LED angezeigt. Das den drei Registerbausteinen gemeinsame Strobesignal wird durch das Ausschreiben des Bitpatterns "00000000" auf der linken Bank generiert. Durch Anwählen der Adressen 3, 4 und 5 auf der rechten Bank werden die Register zum Auslesen der LAM - Signale L1 bis L8, L9 bis L16 und L17 bis 24 zum IV - Bus durchgeschaltet. Abb. 8.3.3 - 1 zeigt den Schaltungsauszug der LAM - Register.

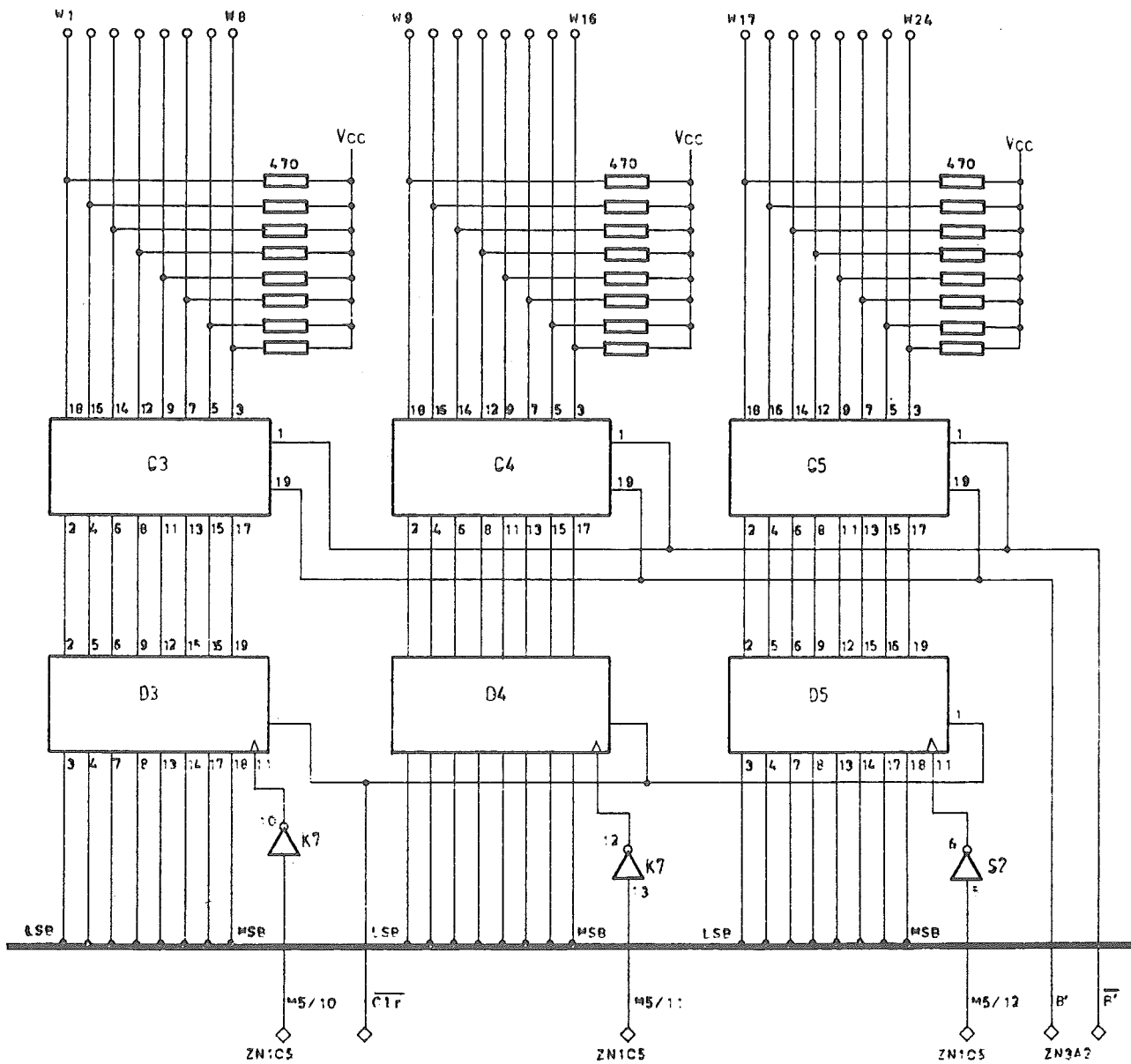


Abb. 8.3.2 - 1 Write - Register

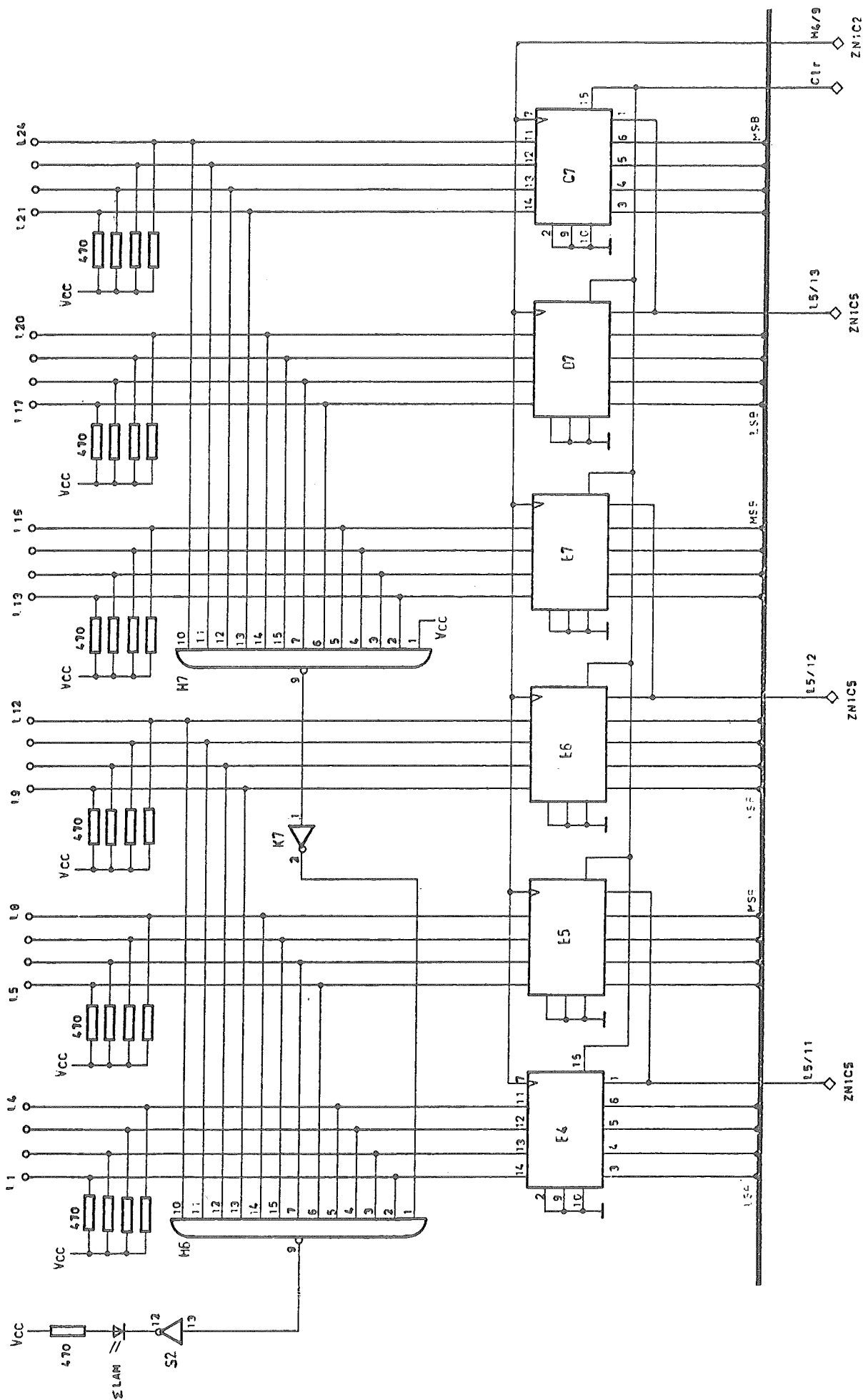


Abb. 8.3.3 - 1 LAM - Register

8.3.4.N - Register und Decoder

Das Anwählen einer bestimmten Stationsnummer geschieht durch Aktivierung einer der 24 N - Stichleitungen N1 bis N24. Da jeweils zu einem vorgegebenen Zeitpunkt nur eine Station (Modul) aktiviert sein darf genügt die binär codierte Darstellung der Stationsnummer an welche ein CAMAC - Zyklus abgesetzt werden soll. Damit sind 5bit zur Adressierung einer Stationsnummer erforderlich. Diese 5bit Stationsadresse wird zusammen mit dem Unteradressbit A1 in den 8bit Registerbaustein SN74LS373 auf Steckplatz G3 und unter der Adresse 16 der rechten Bank eingeschrieben. In Abb.8.3.4 - 1 ist dieser Registerbaustein mit nachgeschaltetem Stationsnummerdecoder auszugsweise dargestellt. Für die Stichleitungen sind wiederum Pull - Up Widerstände erforderlich.

Um den Betrieb mehrerer Controller in einem Crate zu ermöglichen (mit verteilten Prioritäten und mindestens einem Controller auf rechter Steckposition) ist ein zusätzliches Bussystem erforderlich, welches die eingesetzten Controller untereinander verbindet und eine Prioritätenzuordnung zuläßt. Dieses Bussystem wird als ACB (Auxiliary Controller Bus) bezeichnet und weist unter anderem einen 5bit Parallelbus zur Übertragung der binär codierten Stationsnummer auf. Aus diesem Grund muß die 5bit Stationsadresse auch dem auf der Controllerrückseite angeordnetem ACB - Steckanschluß (siehe auch Kap. 8.3.7) zugeführt werden. Da der Controllerbetrieb auf unterschiedlichen Steckplätzen und mit unterschiedlicher Priorität möglich sein muß, hat das Stationsnummernregister einen Tristate Ausgang und ist mit dem Stationsnummerdecoder sowie dem ACB - Anschluß direkt verbunden. Abhängig von der gewünschten Funktion wird der Registerausgang bzw. der Decoderausgang aktiviert. Bei Betrieb des Controllers auf rechter Steckposition sind deshalb noch Pull - Up Widerstände erforderlich.

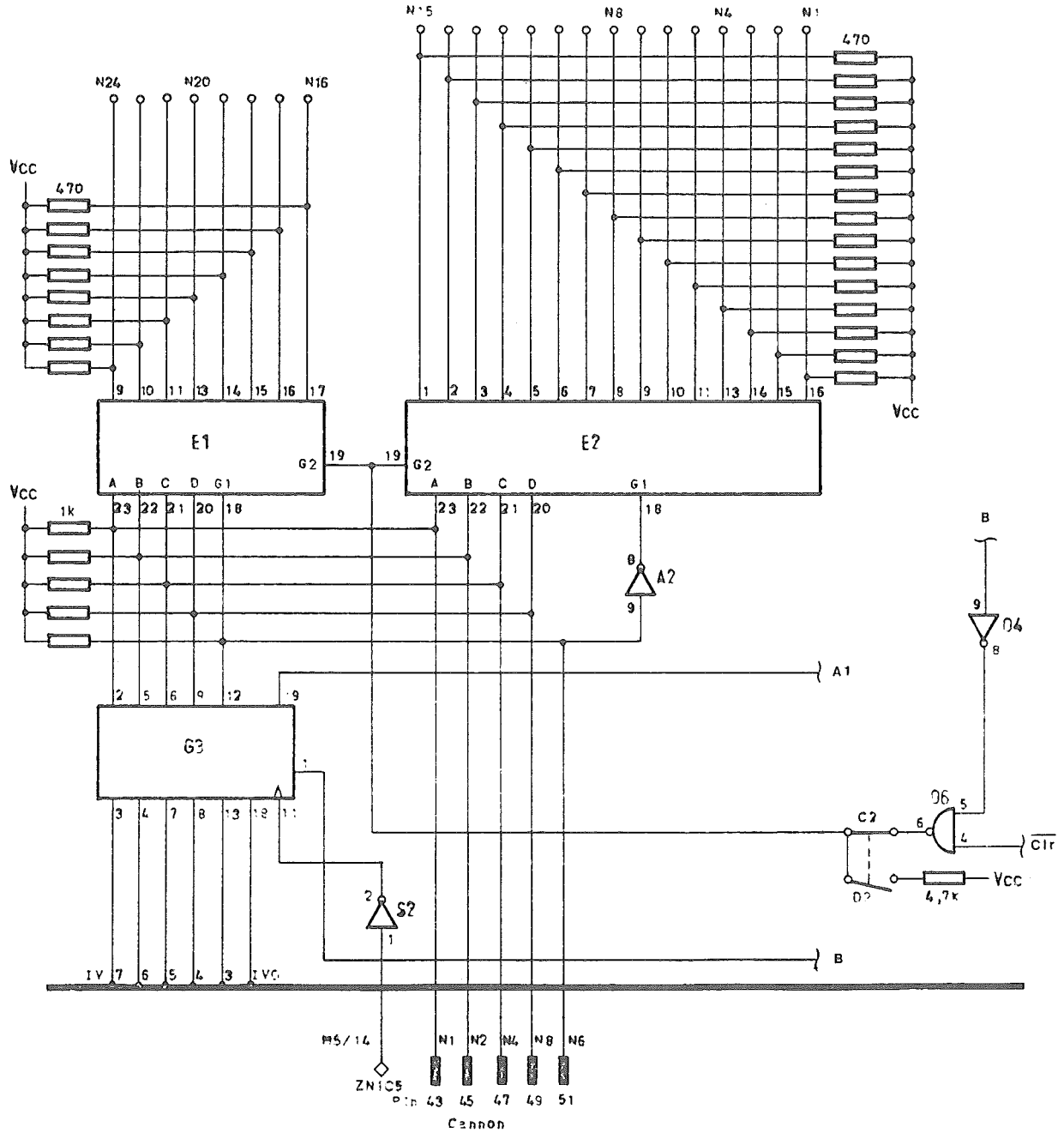


Abb. 8.3.4-1 N - Register und Decoder

8.3.5 A u. F - Register und Treiber

Die für einen kompletten CAMAC - Befehl benötigte Subadresse bestehend aus den Sammelleitungen A1, A2, A4 und A8 und Funktionsleitungen bestehend aus F1, F2, F4, F8 und F16 müssen ebenso wie die Stationsnummeradresse N1, N2, N4, N8 und N16 bei Auslösung eines CAMAC - Befehles aktiviert werden. Der Subadress- sowie Funktionenteil muß damit ebenso wie die Stationsnummer vom Prozessor in ein Register eingebracht werden. Als Register wurde ein SN 74 LS373 verwendet (Steckplatz B5). Da dieses Register 8bit groß ist, wurde die Subadresse A1 im Stationsnummernregister untergebracht. Vergleiche Kapitel 8.3.4. Die Ausgänge des A und F Zwischenregisters werden zur Stromverstärkung über den Leitungstreiber SN74S241 (B3) dem parallelen CAMAC - Datenweg zugeführt. Das Output - Strobosignal für diesen Leitungstreiber wird aus dem eigenen Busysignal gewonnen jedoch bei Auslösen der Lösch- bzw. Initialisierungsfunktion unterdrückt. Mit Hilfe der Gatter P6 und O5 wird das Strobosignal für die Leitungstreiber der Writeregister (siehe Kapitel 8.3.2) nur im Falle eines CAMAC - Befehles mit Daten (F8 = "0" und F16 = "1") durchgeschaltet.

Abb. 8.3.5 - 1 zeigt den Schaltungsauszug mit Funktions- und Subadressenregister sowie Leitungstreibern.

8.3.6 Generator für CAMAC - Zyklen

Um das nach den CAMAC - Spezifikationen zeitoptimale CAMAC - Zyklustiming zu gewährleisten wird dieses im vorliegenden Crate - Controller von einem 10 MHz Grundtakt abgeleitet. Durch Zwischenschalten eines einzelnen Flipflops wird auch der für den Mikroprozessor erforderliche Gegenphasentakt X1 und X2 erzeugt. Die Oszillatorschaltung besteht aus dem 10 MHz Quarz und zwei invertierenden Gates, die zur Arbeitspunkteinstellung mit diskreten Widerständen überbrückt sind. Die Funktion der CAMAC - Zyklusgenerierung ist in Abb. 8.3.6 dargestellt. Zum Starten des CAMAC - Zyklus wird das Flipflop B2 durch Ansteuerung am J - Eingang mit der nächsten positiven Flanke des 10 MHz Taktsignals gesetzt. Damit steht am Ausgang das Busysignal zur Verfügung welches gleichzeitig das 8bit Schieberegister B1 freigibt. Durch die externe Beschaltung dieses Schieberegisters wird erreicht, daß mit dem am Clockeingang anstehenden 10 MHz Takt eine Doppeleins ("11") durchgeschoben wird. Damit stehen an den Ausgängen D und G des Schieberegisters bereits die für den CAMAC - Zyklus benötigten Strobosignale S1 und S2 zur Verfügung.

Nach Ablauf von 10 Einzeltakten wird der K - Eingang des Flipflops aktiviert und damit das Busysignal zurückgenommen.

Abb. 8.3.6-2 zeigt das für die CAMAC - Zykluserzeugung zugehörige Impulsdiagramm. Schaltet sich vor Beginn des Strobsignals und für die Funktion des Controllers als Hilfscontroller (untergeordnete Priorität) der Hauptcontroller mit höchster Priorität durch Setzen des ACL - Signales (Auxiliary Controller Lockout) in die CAMAC - Datenwegoperation ein, so wird über den Clear - Eingang des Flipflop B2 zurückgesetzt und damit der bereits begonnene CAMAC - Zyklus abgebrochen. Wird das ACL - Signal jedoch nach Beginn vom Strobsignal S1 gesetzt, so wird der begonnene CAMAC - Zyklus korrekt durchgeführt.

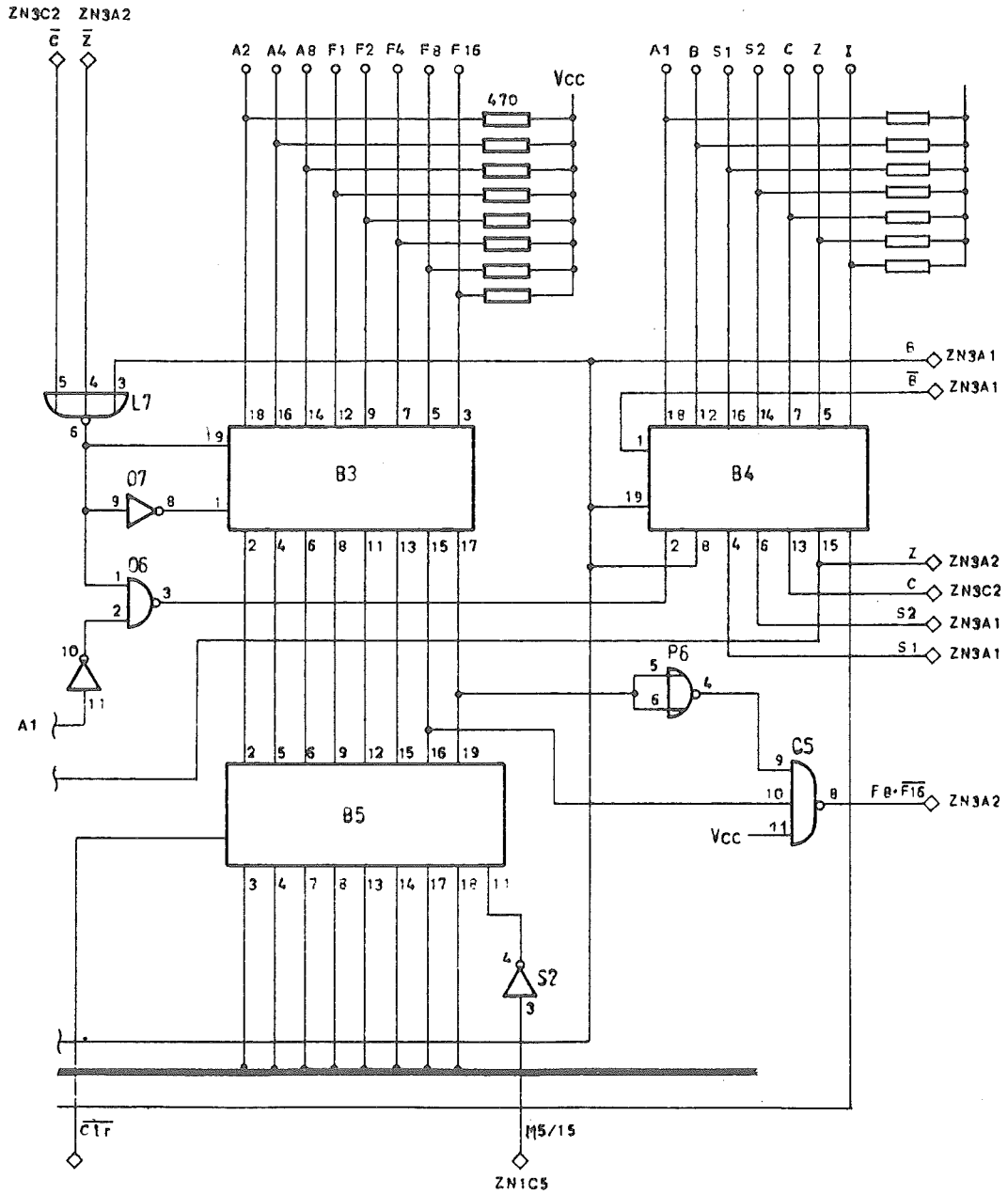


Abb. 8.3.5-1 A u. F - Register und Treiber

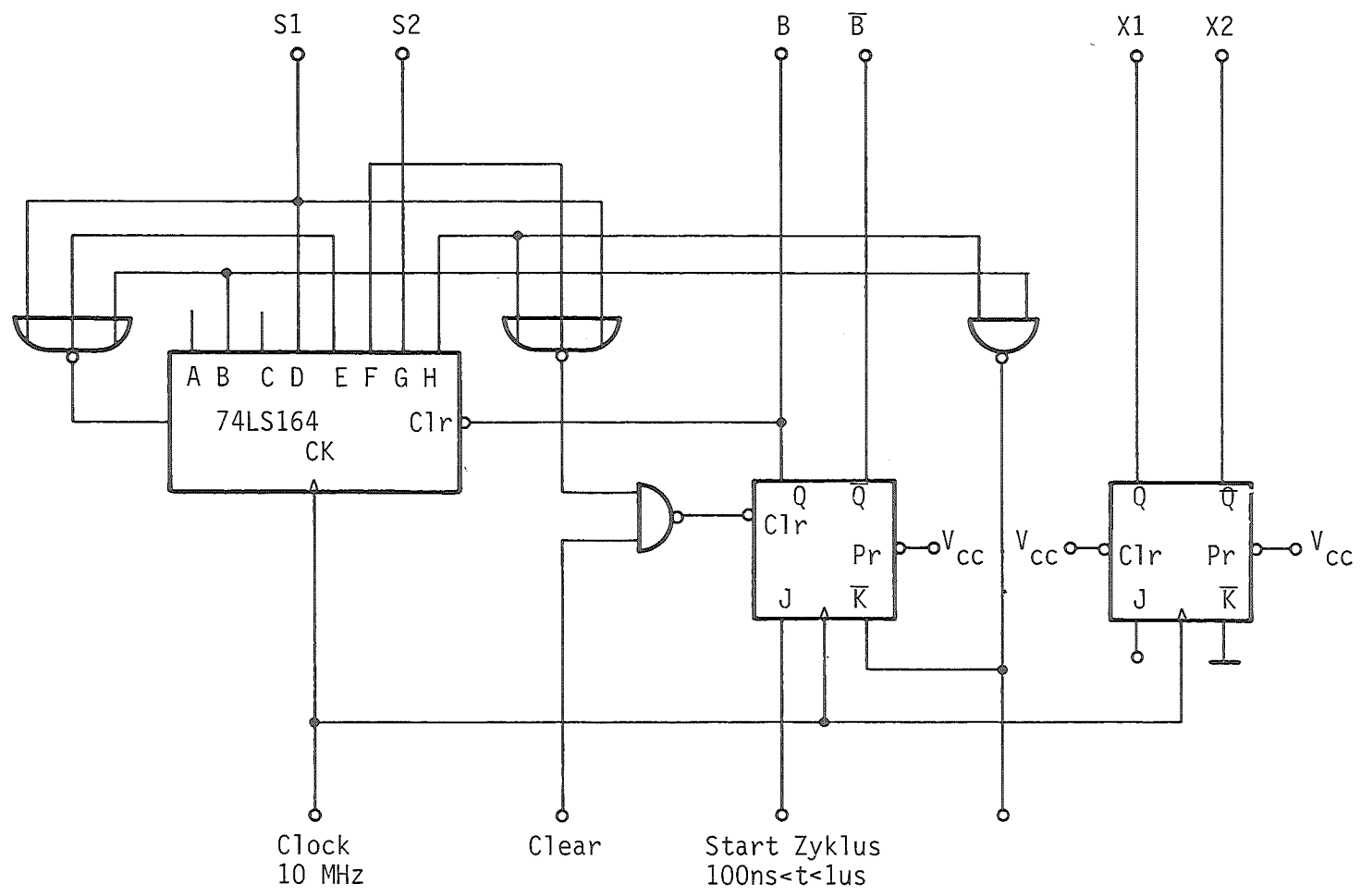


Abb. 8.3.6-1 CAMAC-Zyklusgenerator

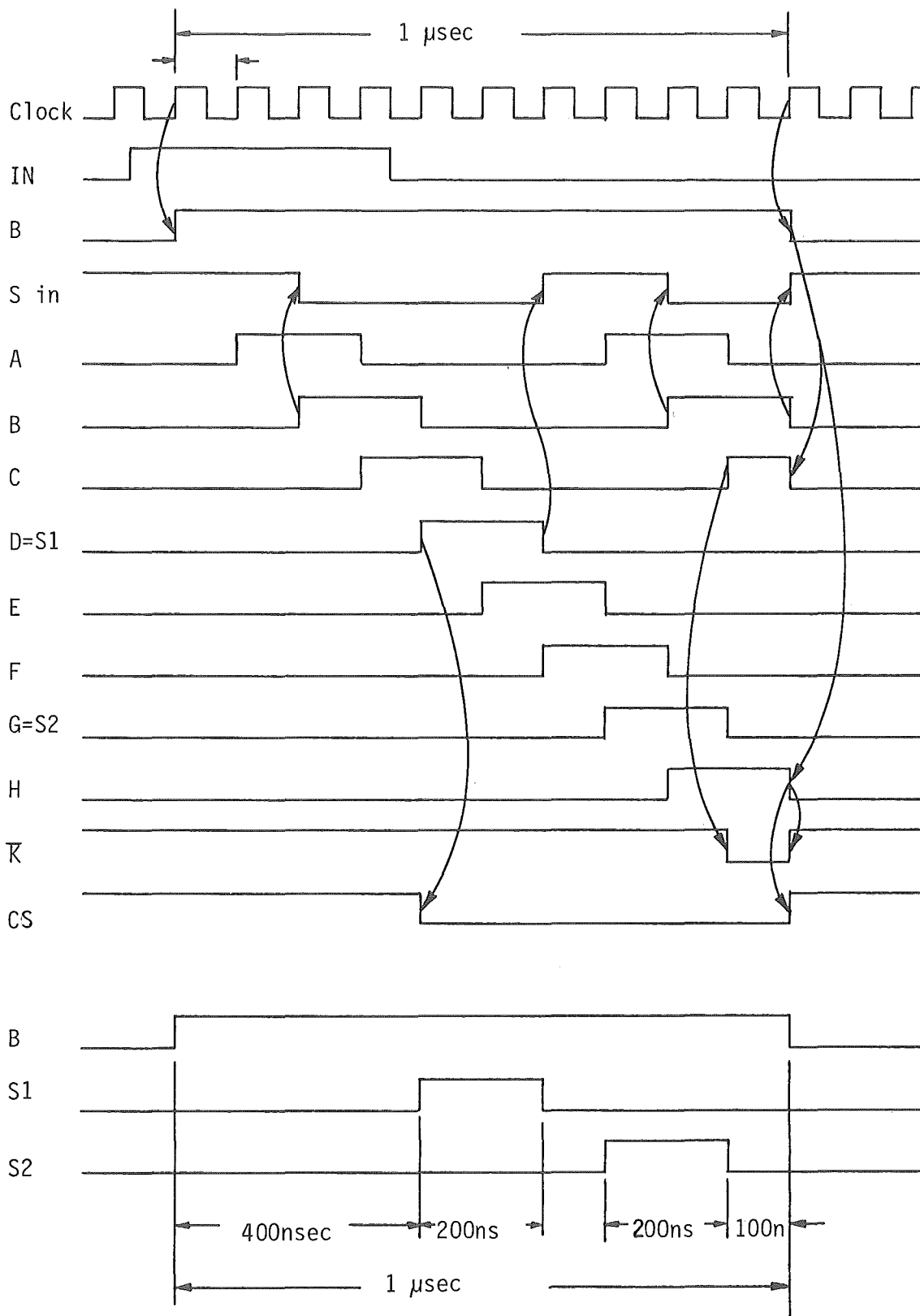


Abb. 8.3.6-2 Impulssdiagramm für CAMAC-Zykluserzeugung aus 10MHz Clock

8.3.7 Funktionsbeschreibung des ACB - Anschlusses

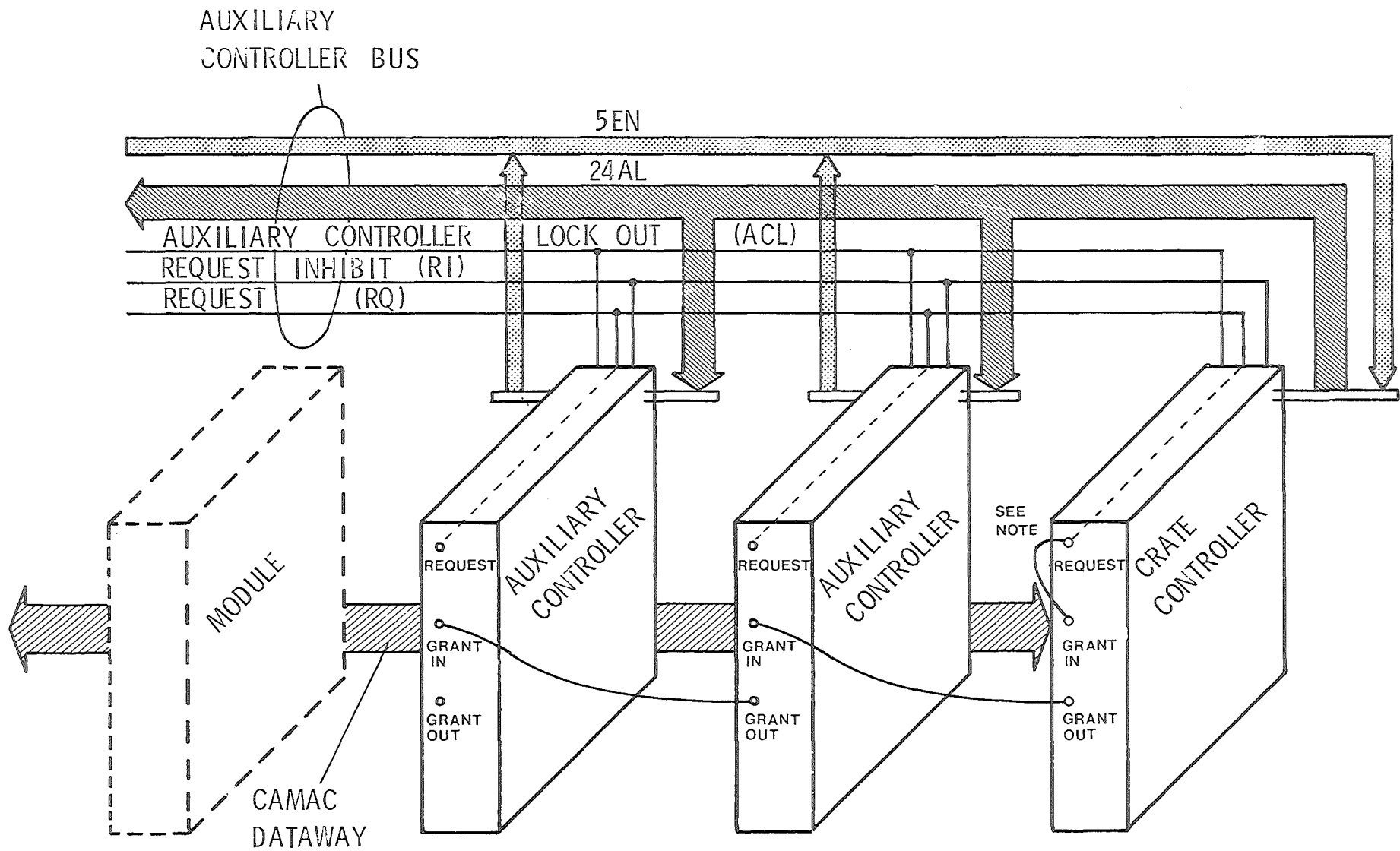
Um den Betrieb mehrerer Controller in einem Crate zu ermöglichen ist ein zusätzliches Bussystem erforderlich, welches die eingesetzten Controller untereinander verbindet und eine Prioritätenordnung zuläßt. Dieses Bussystem wird als ACB (Auxiliary Controller Bus) bezeichnet und besteht

- a) aus einem 40 poligen Parallelbus und verbindet entsprechende Steckanschlüsse, die auf den Controllerrückseiten angeordnet sind
- b) aus einer seriellen Verschaltung der eingesetzten Controller (Daisy Chain), die über auf den Controllerfrontseiten angeordneten LEMO - Buchsen vorgenommen wird und mit der die gewünschte Prioritätenverteilung eingestellt wird.

Abb. 8.3.7-1 zeigt die prinzipielle Verschaltung mehrerer Controller mit dem ACB - Bus. Dieser Parallelbus besteht aus 5 Leitungen für eine binär-codierte Stationsnummer, 24 Busleitungen für die LAM - Signale und 3 weiteren Signalleitungen zur Ablauf- und Prioritätensteuerung.

Abb. 8.3.7-2 zeigt die prinzipielle Verschaltung der 40 poligen ACB - Anschlußbuchse mit den controllerinternen Signalleitungen. Da der im vorliegenden Controller realisierte ACB - Anschluß auch die Spezifikationen des SGL - Encoder Connectors (Crate Controller L2) erfüllen soll und dieser eine 52 polige Anschlußbuchse mit Freikontakten besitzt, wurde für die Kombination des ACB- und SGL - Anschlusses diese 52 polige SGL - Buchse gewählt. Die für den ACB - Anschluß notwendigen Signalleitungen REQUEST und REQUEST INHIBIT wurden auf die Freikontakte des SGL - Connectors verteilt. Die daraus resultierende Pinbelegung dieser ACB - SGL - Anschlußbuchse kann der Abbildung 9.-3 entnommen werden.

Abb. 8.3.7-3 zeigt den Schaltungsauszug mit den für die Prioritätensteuerung erforderlichen Schaltelementen. Die damit eng verbundene Schaltungsgruppe zur CAMAC - Zyklusgenerierung ist hierbei nur als Block wiedergegeben. Tritt in einem Controller der interne Wunsch zum Absetzen eines CAMAC - Zyklus auf, so wird über das Signal "Internal Request" das entsprechende Flipflop gesetzt und die Requestleitung des ACB - Bussystems aktiviert (logisch "0" gesetzt). Im Controller mit der höchsten Priorität (dies muß nicht zwangsläufig der Controller auf der rechten Steckposition sein) wird mit Hilfe eines Jumperkabels der auf der Frontseite des Controllers herausgeführte Requestanschluß mit dem Grant In Anschluß verbunden. Liegt kein Wunsch zur Auslösung eines CAMAC - Zyklus vor, so erscheint dieses Signal nach einer



Note: This connection made at highest priority controller using R/G protocol

Abb.8.3.7-1 MULTIPLE CONTROLLERS IN A CAMAC CRATE

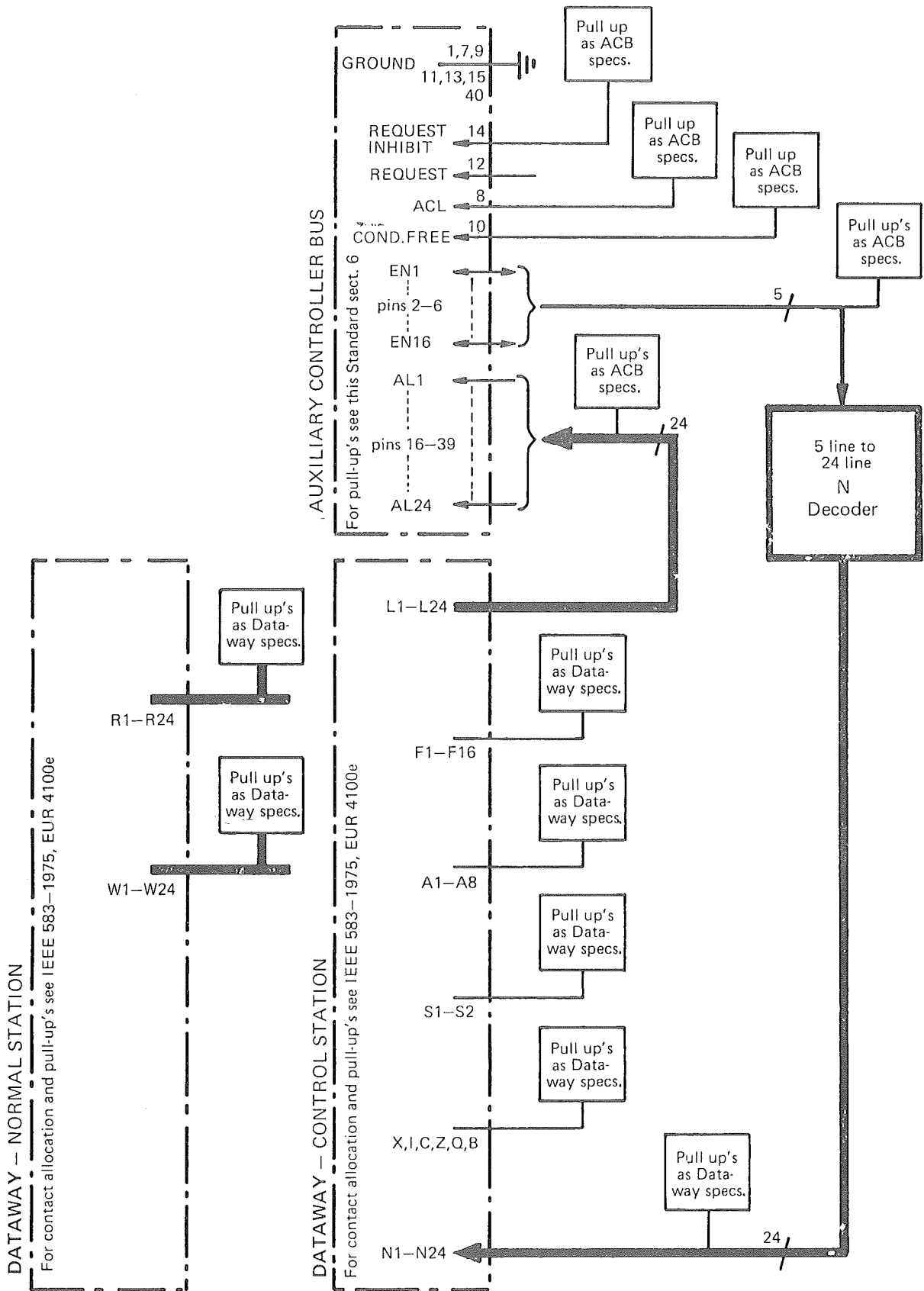


Abb. 8.3.7-2

CRATE CONTROLLER
MINIMUM CONFIGURATION

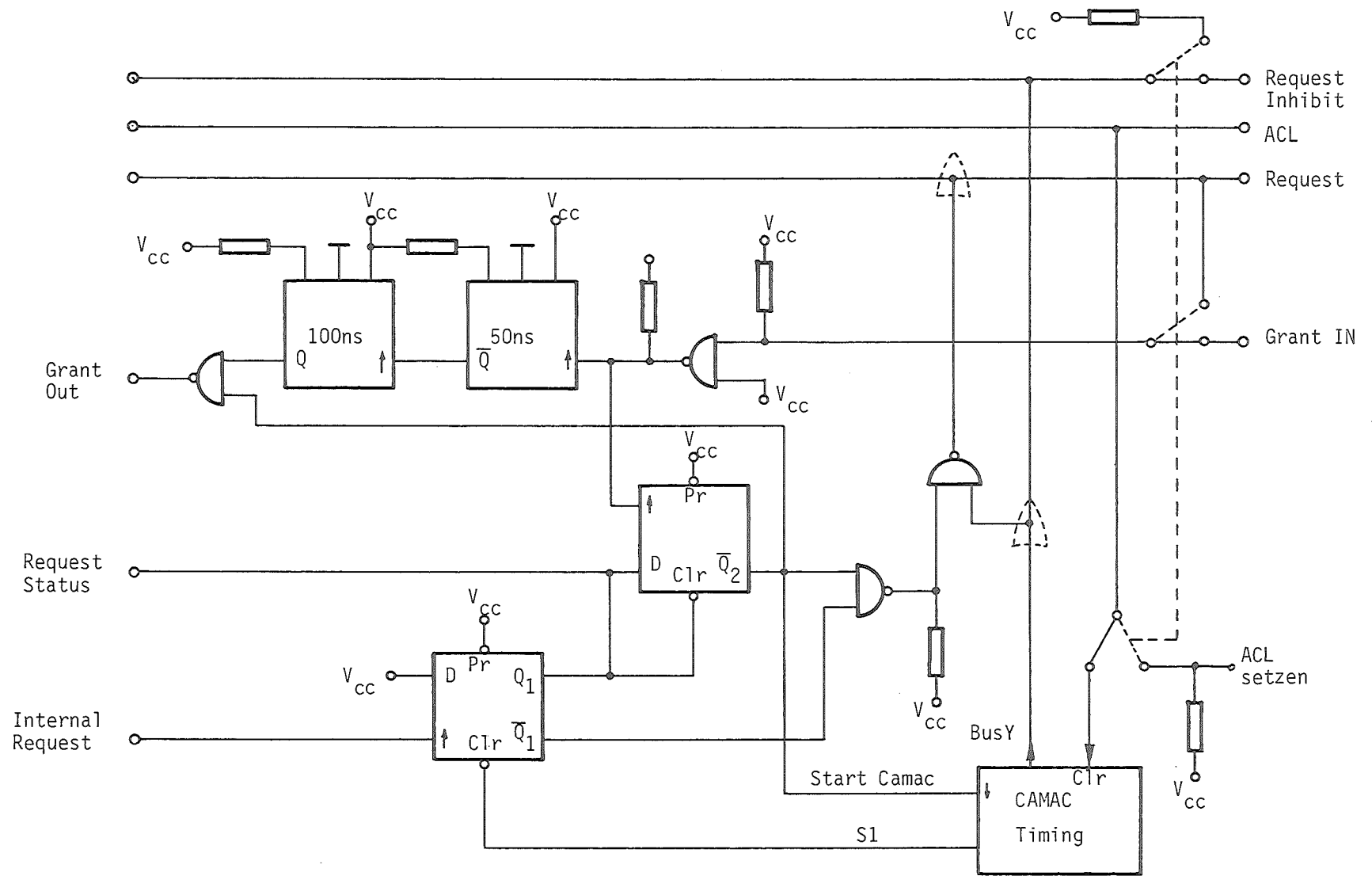


Abb. 8.3.7-3

Verzögerung von 50 ns und mit einer Dauer von 100 ns am Grant Out Anschluß. Entsprechend der Prioritätenverteilung durch die externe Verkabelung der Grant In und Grant Out Anschlüsse durchläuft dieses Signal in fallender Prioritätenfolge alle Controller bis zu jenem, der durch den internen CAMAC - Zykluswunsch die ACB -Requestleitung aktiviert hat. In diesem Controller veranlaßt das Grant Signal das Sperren des Grant Out Anschlusses und das Starten des CAMAC - Zyklus. Wird vor Beginn des Strobsignales S1 der CAMAC - Zyklusgenerator durch das ACL - Signal angehalten, so bleibt der interne Zykluswunsch solange aufrechterhalten, bis eine Freigabe erfolgt und ein kompletter CAMAC - Zyklus abgesetzt werden konnte. Wird der begonnene CAMAC - Zyklus nicht durch ein ACL - Signal abgebrochen, so wird mit Beginn von S1 das Internal Request Flipflop gelöscht und nach Ablauf des kompletten CAMAC - Zyklus der Ausgangszustand wieder erreicht. Durch das vom CAMAC - Zyklusgenerator erzeugte Busy - Signal wird die ACB - Request Inhibitleitung aktiviert und damit die Ausgabe eines Request-signalles durch einen anderen Controller verhindert. Damit ist sichergestellt, daß das gleichzeitige Belegen des CAMAC - Datenweges durch mehrere Controller nicht erfolgen kann und beim Auftreten mehrerer gleichzeitiger Anforderungen zur Durchführung eines CAMAC - Zyklus die Reihenfolge entsprechend der vorgenommenen Prioritätenverteilung eingehalten wird und keine Anforderung unbearbeitet bleibt.

In Abb. 8.3.7-4 ist das Impulssdiagramm für die Durchführung eines CAMAC - Zyklus nach den ACB - Vorschriften dargestellt.



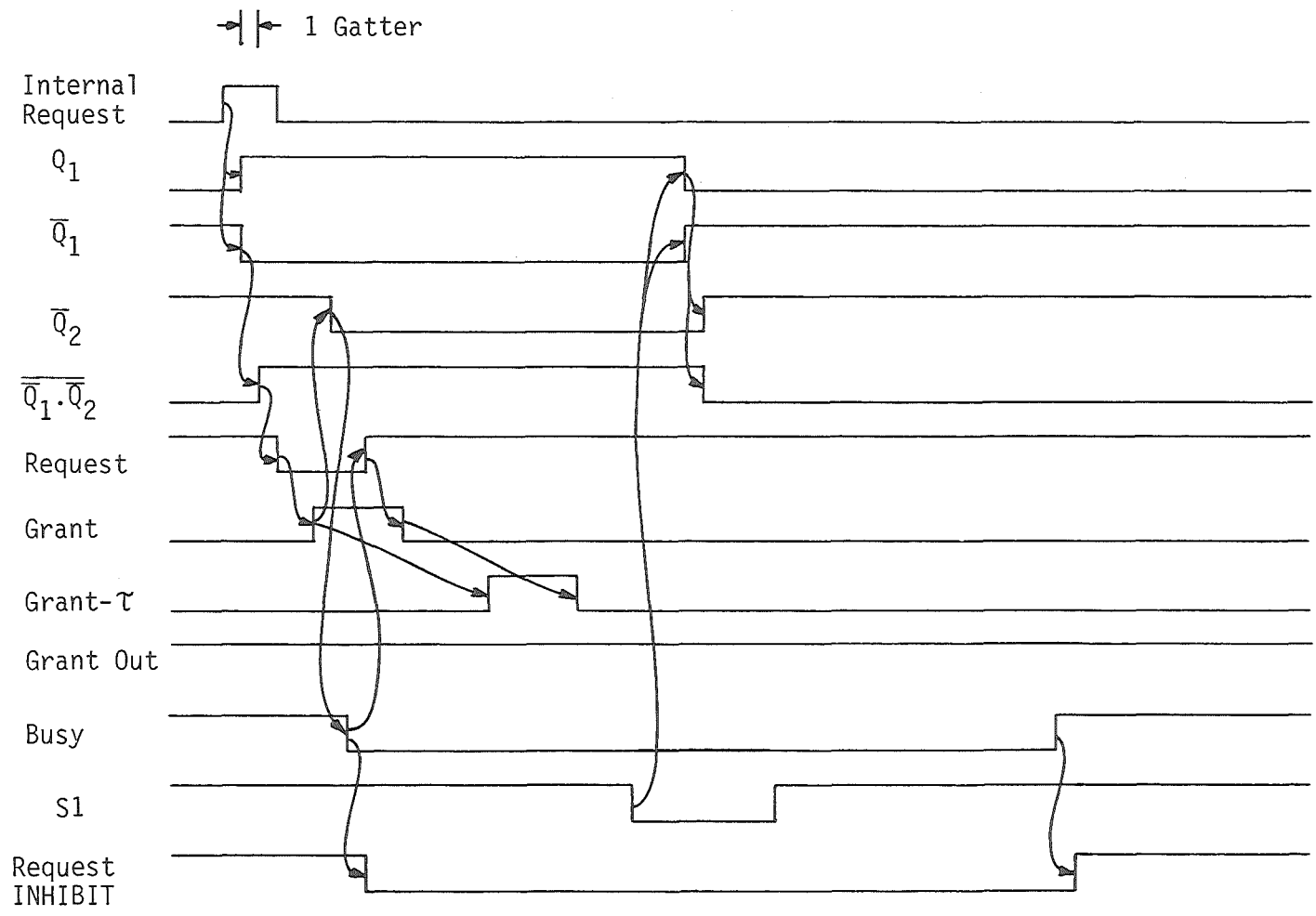


Abb. 8.3.7-4

8.4 Eingangsschaltung der manuellen Bedienung

Zur manuellen Bedienung stehen auf der Frontplatte des Controllers ein Kippschalter und zwei Tastschalter zur Verfügung. Mit Hilfe des Kippschalters kann zwischen "ONLINE" und "OFFLINE" gewählt werden. In Abhängigkeit von der Stellung des Kippschalters werden den beiden Tastschaltern unterschiedliche Funktionen zugeordnet. In Stellung "OFFLINE" können damit die CAMAC - Funktionen "C" (löschen) und "Z" (initialisieren) ausgelöst werden. Wird die Taste "C" ununterbrochen länger als 5 sec gedrückt so wird vom Prozessor ein Testprogramm ausgeführt. In Stellung "ONLINE" wird durch Betätigung der Z - Taste zusätzlich der Mikroprozessor zurückgesetzt, d.h. der weitere Programmablauf beginnt ab Adresse 0.

Die folgende Tabelle 8.4 -1 zeigt alle Schaltstellungen mit zugehörigen Funktionen.

KIPPSCHALTER	ON		OFF
	ON	OFF	
CONTROLLER			
TASTE "C"	X	C >5sec → Test	C >5sec → Test
TASTE "Z"	μP Reset	Z μP Reset	Z

Tabelle 8.4 - 1

In der Abb. 8.4 - 2 ist die Eingangsschaltung der manuellen Bedienung zu sehen. Die genannten Schaltfunktionen werden zur Erstellung eines sauberen Schaltimpulses mit Hilfe von RS - Flipflops entprellt. Das RS - Flipflop zur Entprellung der ON- bzw. OFFLINE - Funktion hat einen weiteren Eingang, über welchen der Prozessor die ONLINE - Funktion verriegeln kann. Zu diesem Zweck ist ein weiteres RS - Flipflop vorhanden, welches vom Prozessor gesetzt bzw. zurückgesetzt werden kann. Wie der Schaltung zu entnehmen, wird nur dann der ONLINE - Zustand erreicht, wenn der manuelle Schalter in Position ONLINE steht und der Prozessor die Funktion zum ONLINE schalten ausgelöst hat. Ist der ONLINE - Zustand erreicht, so werden die Funktionseingänge der Tastschalter zum Auslösen der Löscho- und Initialisierungsfunktion abgeschaltet. Da die Signale Z und C mit einer CAMAC - Datenwegoperation verbunden sein müssen und nur während dieser Operation auf den Datenweg aufgeschaltet werden dürfen, werden die von den Tastschaltern generierten Signale in Flipflops statuiert, während des Bussignals durchgeschaltet und nach Ablauf der Datenwegoperation wieder gelöscht. Zum Auslösen der internen Controller - Clearfunktionen werden die in den Flipflops N5 statuierten Signale Z und C durch logische ODER - Verknüpfung zu den Signalen CLEAR und $\overline{\text{CLEAR}}$ weiterverarbeitet.

8.5 Crate - Address Register und Anzeige

Um den Betrieb mehrerer CAMAC - Crates an einem gemeinsamen Datenübertragungssystem (Ringleitung) zu betreiben muß jedes einzelne Crate mit einer Adresse versehen werden unter der es vom Leitreechner anzusprechen ist. Diese dem Crate eigene Adresse wird im Controller mit Hilfe entsprechender Schalter im Bereich von "00" bis "99" festgelegt. Die eingestellte Crateadresse muß vom Prozessor ausgelesen werden können und dem weiteren Programmablauf zur Verfügung stehen. Da die neben dem RAM - Arbeitsspeicher möglichen 8 Adressen zum Auslesen externer Register belegt sind, mußte zum Lesen der Crateadresse ein anderer Weg gewählt werden. Wie aus Abb. 8.5-1 zu entnehmen wurde zum Lesen der Crateadresse die Schreibfunktion eines externen Registers benutzt. Unter der Adresse 11 auf der rechten Bank ist das Register auf dem Steckplatz P2 (SN74LS273) zu erreichen. Das vom Prozessor in dieses Register eingeschriebene Bitpattern entspricht einer BCD - codierten Crateadresse, die durch die eingesetzten Anzeigen vom Typ HP 5082 auf der Frontplatte des Cratecontrollers angezeigt wird. Die Ausgänge dieses externen Crateadressregisters werden zusammen mit den Schaltleitungen der manuellen Schalter zwei Vergleicherausteinen P1 und Q2 zugeführt. Das Ausgangssignal der Vergleicherausteine welches Identität zwischen eingestellter und angezeigter Adresse angibt wird als Input - Steuersignal dem Prozessor zugeführt. Bei Programmbeginn durchläuft der Prozessor eine Softwareschleife, bei welcher eine inkrementierte Crateadresse ausgegeben und verglichen wird. (bei Adresse "0" beginnend). Stellt der Prozessor Identität zwischen eingestellter und ausgeschriebener BCD - codierter Crateadresse fest verläßt er diese Softwareschleife, wandelt den BCD - Code in einen Binärcode um und speichert die gefundene Crateadresse in der RAM - Arbeitsspeicherzelle mit der Adresse 20 ab. Durch die Trennung von eingestellter BCD - Adresse und angezeigter Adresse sowie der Ausschreibefunktion durch den Prozessor kann das Display der Crateadressenanzeige auch für eine andere Ausgabefunktion des Controllers benutzt werden. Z.B. kann in einer vom Prozessor durchgeführten Testroutine ein festgestellter Fehler als Fehlernummer ausgegeben werden.

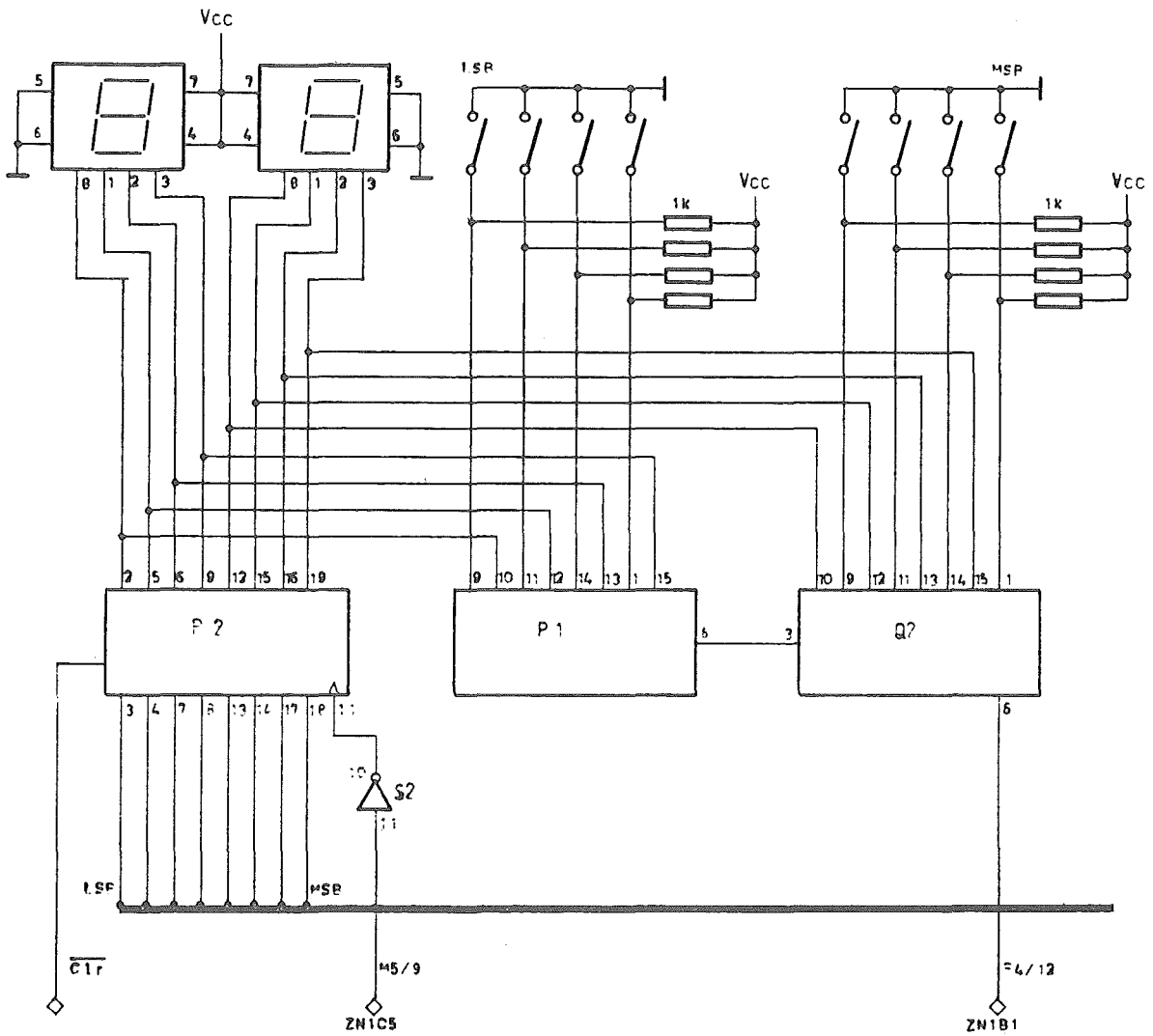


Abb. 8.5-1 Crate - Adress Register und Anzeige

8.6 Hardware-Statusregister

Neben dem in RAM Adresse 367 realisierten Statusbyte existiert zur Bildung des Gesamtstatus ein Hardware - Statusregister, welches durch Adressierung unter der Adresse 6 auf der rechten Bank zu erreichen ist und das Auslesen bestimmter durch Hardware realisierter Signale ermöglicht. Diese Signale sind einmal die durch Hardwarejumper ausgewählten LAM - Signale (graded LAM's) sowie die den CAMAC - Spezifikationen entsprechenden Signale Q, X und Inhibit. Das für dieses Register zum Statuieren dieser Signale verwendete Strobesignal geht aus der logischen "ODER" Verknüpfung des Strobesignals S1 und dem Strobesignal der LAM - Register hervor.

Abb. 8.6-1 zeigt das mit Hilfe des Bausteins SN74LS373 realisierte Hardware-Statusregister und gibt die Signalbedeutung der einzelnen Bitstellen an.

Im Gegensatz zum Softwarestatusregister (RAM 367), welches die Schreib- und Lesefunktion erlaubt, kann dieser durch Hardware realisierte Statusteil nur gelesen werden.

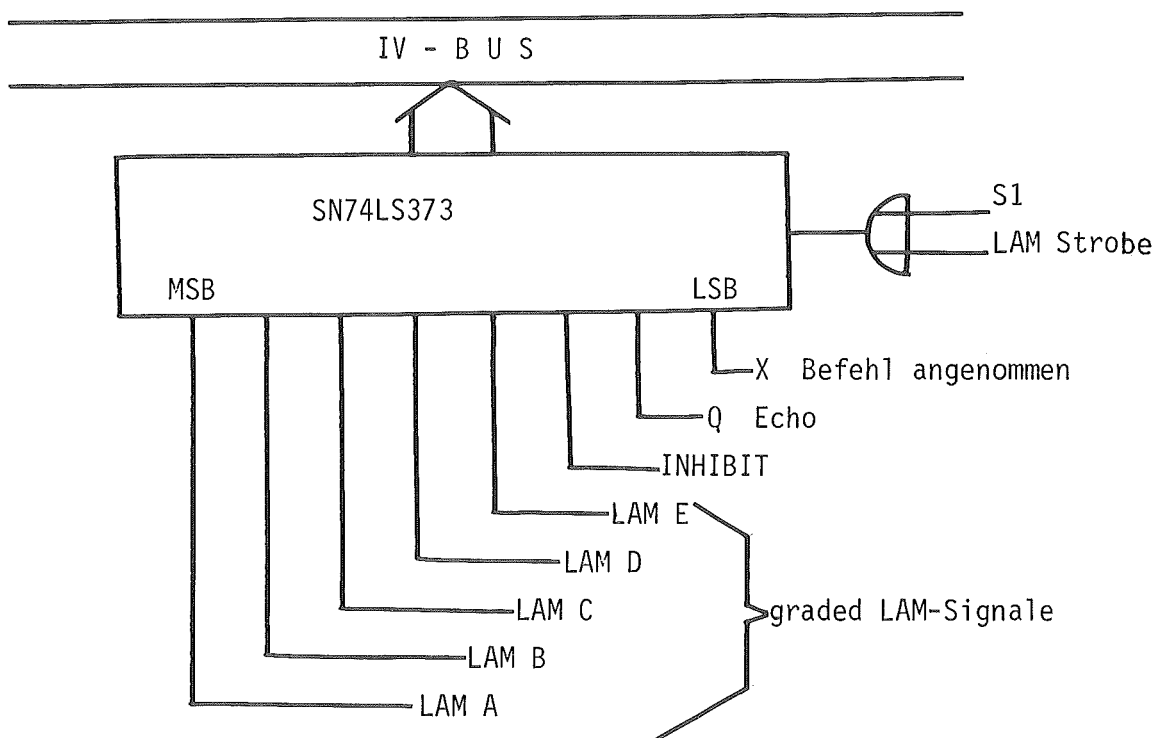


Abb. 8.6-1

8.7 RAM - Arbeitsspeicher

Die Abb. 8.7-1 zeigt das Teilblockschaltbild der im vorliegenden Controller realisierten Arbeitsspeicherfunktion. Da über den IV -Bus des Prozessors auf jeder Bank 256 angeschlossene Adressregister angesprochen werden können und auf der linken Bank aus Hard- und Softwaregründen lediglich ein IV-Byte angeschlossen ist, sind ohne weitere Zwischenadressierungen lediglich 256 Adressen auf der rechten Bank zur Verfügung. Zur Realisierung des Arbeitsspeichers wurden 8 RAM - Bausteine vom Typ SN74S201 und der Organisation 256 x 1 bit eingesetzt. Dieser Bausteintyp ist einer der wenigen zur Verfügung stehenden schnellen Typen und erfordert neben der Vorschaltung eines 8 bit Adressregisters (SN74193) am Ausgang einen Tristate - Bustreiber (SN74LS240), um einen IV - Bus kompatiblen Anschluß herzustellen. Da neben dem Arbeitsspeicher weitere 16 Adressen für externe Registerbausteine (Lese- und Schreibregister für CAMAC - Datenweg sowie Kontrollregister) benötigt werden und der Umweg über eine Zwischenadressierung vermieden werden sollte, wurde durch Anschluß eines Adressdecoders (SN74LS138) und entsprechende Logikbausteine zum Abschalten des Bustreibers ein externer Adressraum von 16 Adressen geschaffen. Diese dadurch zur Verfügung stehenden externen Adressen überdecken einen Teil des RAM - Arbeitsspeichers (16 Adressen) und werden wie eine Zelle des Arbeitsspeichers angesprochen, haben jedoch im Gegensatz zu dieser lediglich eine Schreib- oder Lesefunktion. In der Tabelle 8.7-2 sind alle für bestimmte Funktionen benutzten Arbeitsspeicheradressen aufgeführt.

Für das in der RAM-Zelle 367 realisierte Softwarestatusbyte wird in der Tabelle 8.7-3 die Signalbedeutung der einzelnen Bitstellen angegeben.

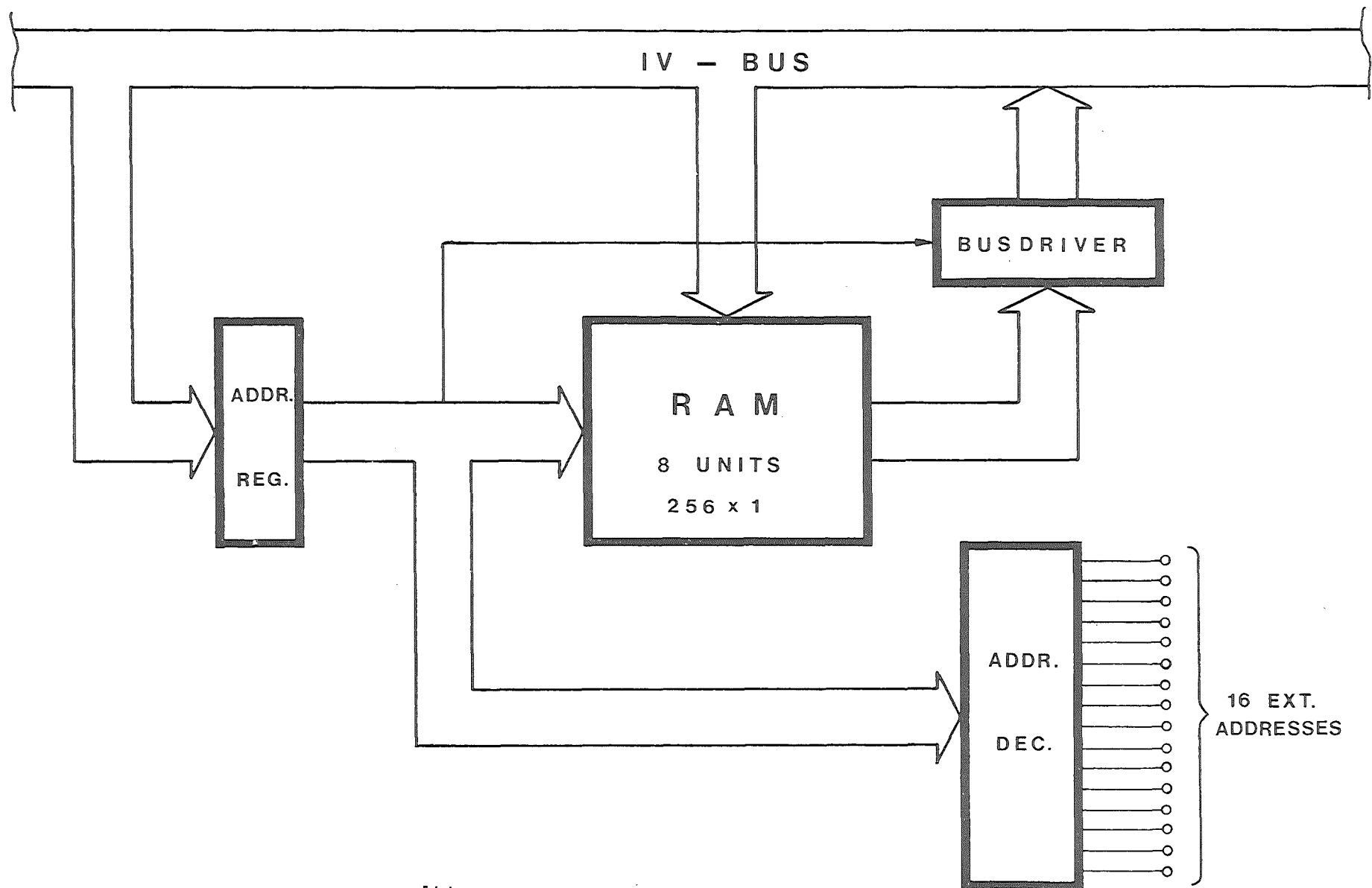


Abb.8.7-1 Arbeitsspeicher

Liste spezieller RAM - Register

Adresse	Funktion	
0	R1 - R8	} Read - Register vom CAMAC - Dataway
1	R9 - R16	
2	R17 - R24	
3	L1 - L8	} LAM - Register vom CAMAC - Dataway
4	L9 - L16	
5	L17 - L24	
6	Status- Register	
7	FIFO-IN-Register	
10	FIFO-OUT-Register	
11	Crate-Adress-Register	
12	W1 - W8	} Write - Register vom CAMAC - Dataway
13	W9 - W16	
14	W17 - W24	
15	frei	
16	NA-Register	N,A,F - Register vom
17	AF-Register	CAMAC - Dataway
20	Crate-Adresse	
360	Status der Vorsendung	
361	L1 - L8	} noch nicht verarbeitete LAM - Signale
362	L9 - L16	
363	L17 - L24	
364	Blocklänge lesen	1er - Komplement
366	Blocklänge schreiben	
367	Status-Register	
370	Restblocklänge	
372	A,F - Register	für Schreib- und Lesefunktion
373	N,A - Register	für Lesefunktion
374	N,A - Register	für Schreibfunktion
375	Wortnummernspeicher	für das letzte richtige PDV -Wort
377	Prüfregister	PDV - Funktionenteil

Adresse: 0 - 7 nur lesen } Hardware-Zugriff
 10 - 17 nur schreiben }
 20 - 377 lesen und schreiben — Software-Register mit
 speziellen Funktionen

Tabelle 8.7 - 2

Status-Register RAM 367

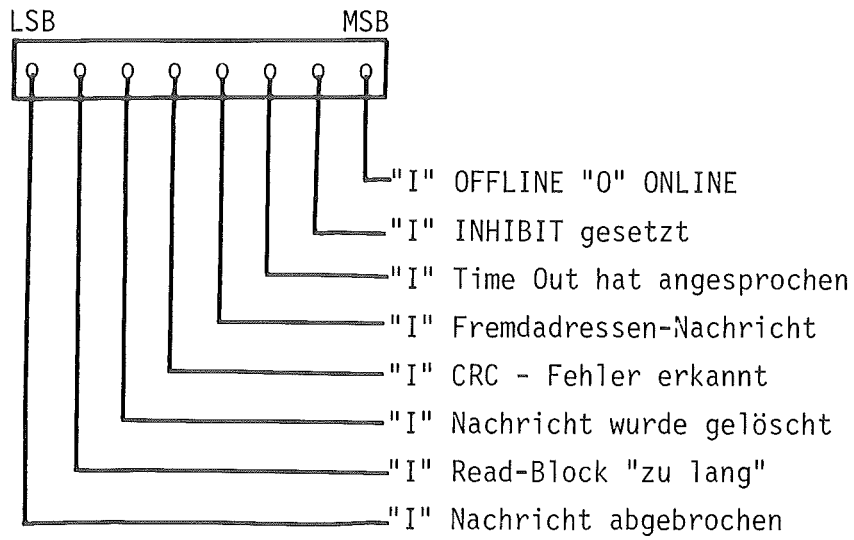


Abb. 8.7-3

8.8 IV - Byte mit Steuersignaleingang und -ausgang

Zur Steuersignal Ein- und Ausgabe wird im wesentlichen ein Interface - Vektor Chip (Typ 8 T 32 mit Tristate - Ausgang auf der Benutzerseite) eingesetzt. Dieser Chip hat die fest eingebrannte Adresse 1 und kann vom Prozessor mit entsprechenden Instruktionen ein- bzw. ausgeschaltet werden. Durch die Aufteilung des möglichen Adressumfangs des Prozessorbussystems in zwei Gruppen (rechte Bank und linke Bank) und der Verschaltung des IV - Byte Chips als einzigstes adressierbares Element auf der linken Bank braucht dieses Chip bei Programmbeginn lediglich einmal eingeschaltet zu werden. Da während des weiteren Programmablaufs dieses Chip immer eingeschaltet bleibt und der Prozessor ohne Zwischenadressierungsfunktion darauf zugreifen kann ist die Ein- oder Ausgabe von Signalen über dieses IV - Byte ohne Verzögerung möglich. Da dieses IV - Byte als 8bit Ein- und Ausgabeport benutzt wird, muß einmal auf der Benutzerseite eine Trennung von Eingangs- und Ausgangssignalen realisiert werden und zum anderen ein Umschaltsignal erzeugt werden, welches über den BIC - Anschluß des IV - Bytes das Benutzerport (User - Port) von Eingabe auf Ausgabe umschaltet.

Der Schaltungsauszug in Abb. 8.8 - 1 zeigt die prinzipielle Verschaltung der Inputs und Outputs zum IV - Byte und die zugehörigen Signalabläufe. Der Schaltzustand des IV - Bytes ist in der Regel für Inputsignale von der Peripherie geöffnet. Da von der Peripherie nur 8 Inputsignale benötigt werden, können diese ohne Codierung über die Zwischenspeicher G4 und F4 (SN74173) dem User - Port des IV - Bytes zugeführt werden. Damit beschränkt sich das Abtesten eines Inputsignales (Zustand der Peripherie) für den Prozessor auf das Abtesten eines einzelnen Bitsignales. (Tabelle 8.8 - 2 zeigt die Eingangssignale mit ihrer Zuordnung). Da dieses IV - Byte das einzige adressierbare Register der linken Bank ist kann das Umschaltsignal von Input- nach Outputverkehr des IV - Bytes (BIC - Signal) von den Signalen LB und WC abgeleitet werden. Damit kann der Prozessor ohne zwischengeschaltete Umadressierung oder Umschaltung Inputs bzw. Outputs durchführen. Wird das IV - Byte von Input nach Output umgeschaltet so wird mit Hilfe des Signales BIC der Ausgang der Zwischenspeicher G4 und F4 auf Tristate - Zustand umgeschaltet und die Gates für den Signal- ausgang durchgeschaltet.

Um das Durchschalten der Ausgangssignale mit Sicherheit nur während des Zeitraumes des Anstehens valider Ausgangsdaten zu gewährleisten wird neben dem Umschaltsignal für das IV - Byte BIC auch noch die verzögerte Masterclock benutzt. Da insgesamt mehr Einzelausgangssignale benötigt werden als die 8 vorhandenen Einzelbits, müssen die Ausgangssignale zumindest zum Teil durch ein binärcodiertes Ausgangsbitpattern im IV - Byte realisiert werden. Wie aus Abb. 8.8-3 zu ersehen werden die 3 höchstwertigsten Ausgangsbit zu 8 Ausgangssignalen decodiert. Damit stehen mit den restlichen 5 Ausgangsbits insgesamt 13 Ausgangssignale zur Verfügung von denen aus der Gruppe der ersten 8 decodierten Ausgangssignale jedoch keine 2 gleichzeitig aktiviert werden können.

In Tabelle 8.8-4 sind alle Outputfunktionen mit den für ihre Generierung erforderlichen und auf das IV - Byte aufzubringenden Bitpattern aufgeführt. Da die Outputfunktionen während der vom Prozessor generierten Writephase für die linke Bank aktiviert werden, sind alle Outputsignale Einzelimpulse deren Dauer von der Clockfrequenz des Mikroprozessors abhängt und in ihrer Phasenlänge verkürzte Masterclockimpulse darstellen.

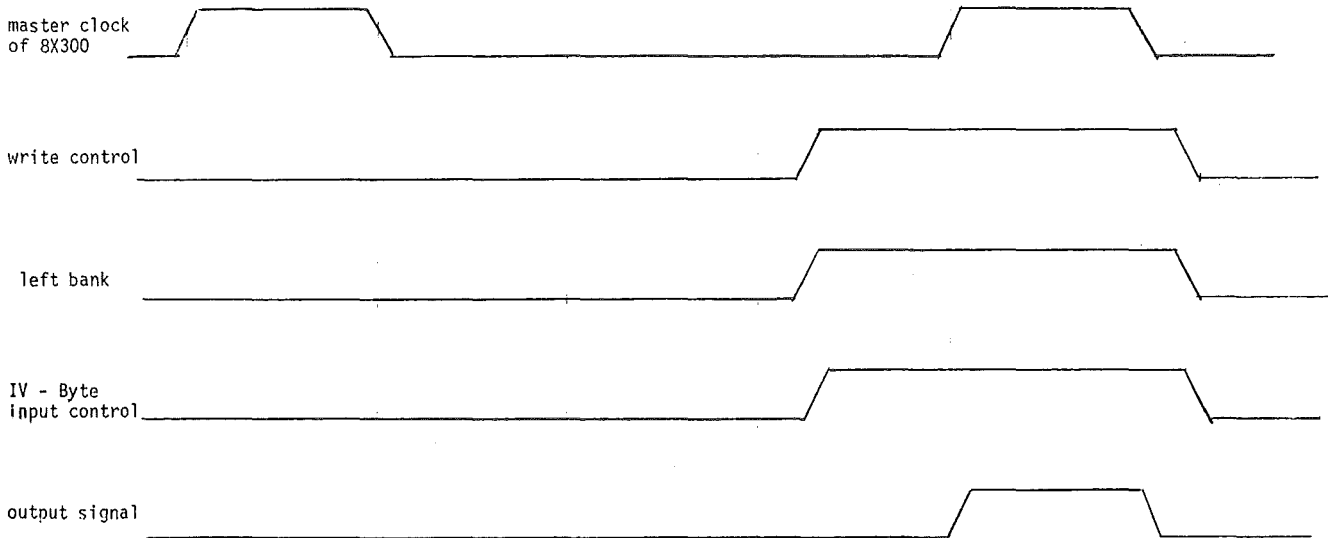
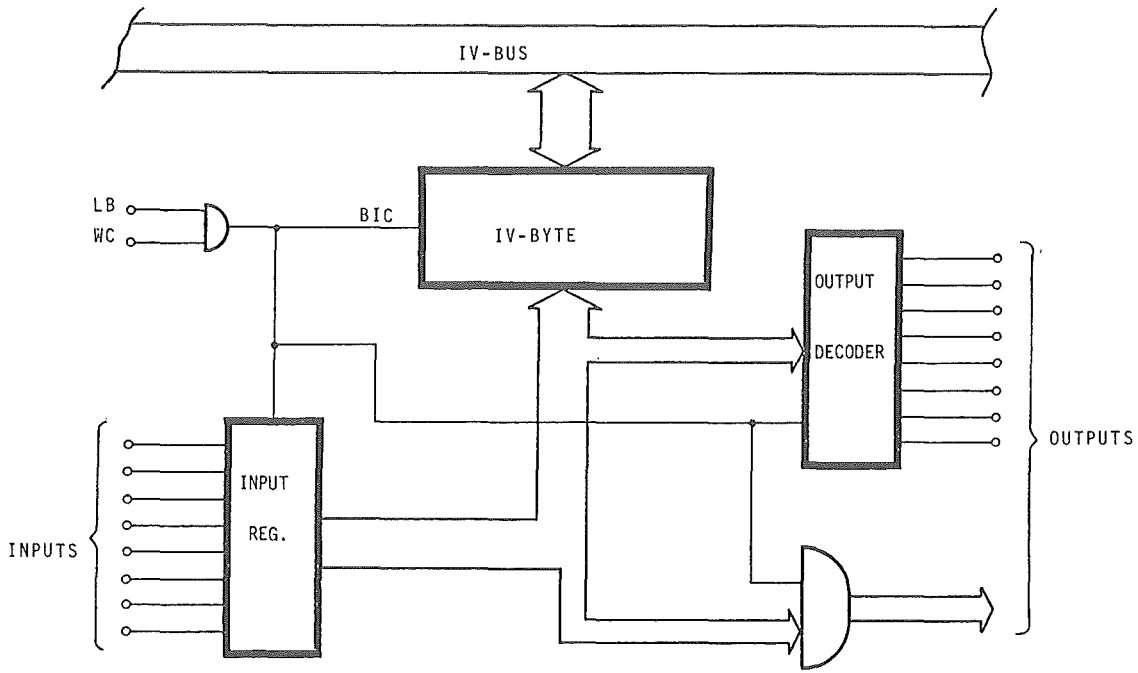


Abb. 8.8-1

Signaldarstellung für Output - Funktion über das IV - Byte

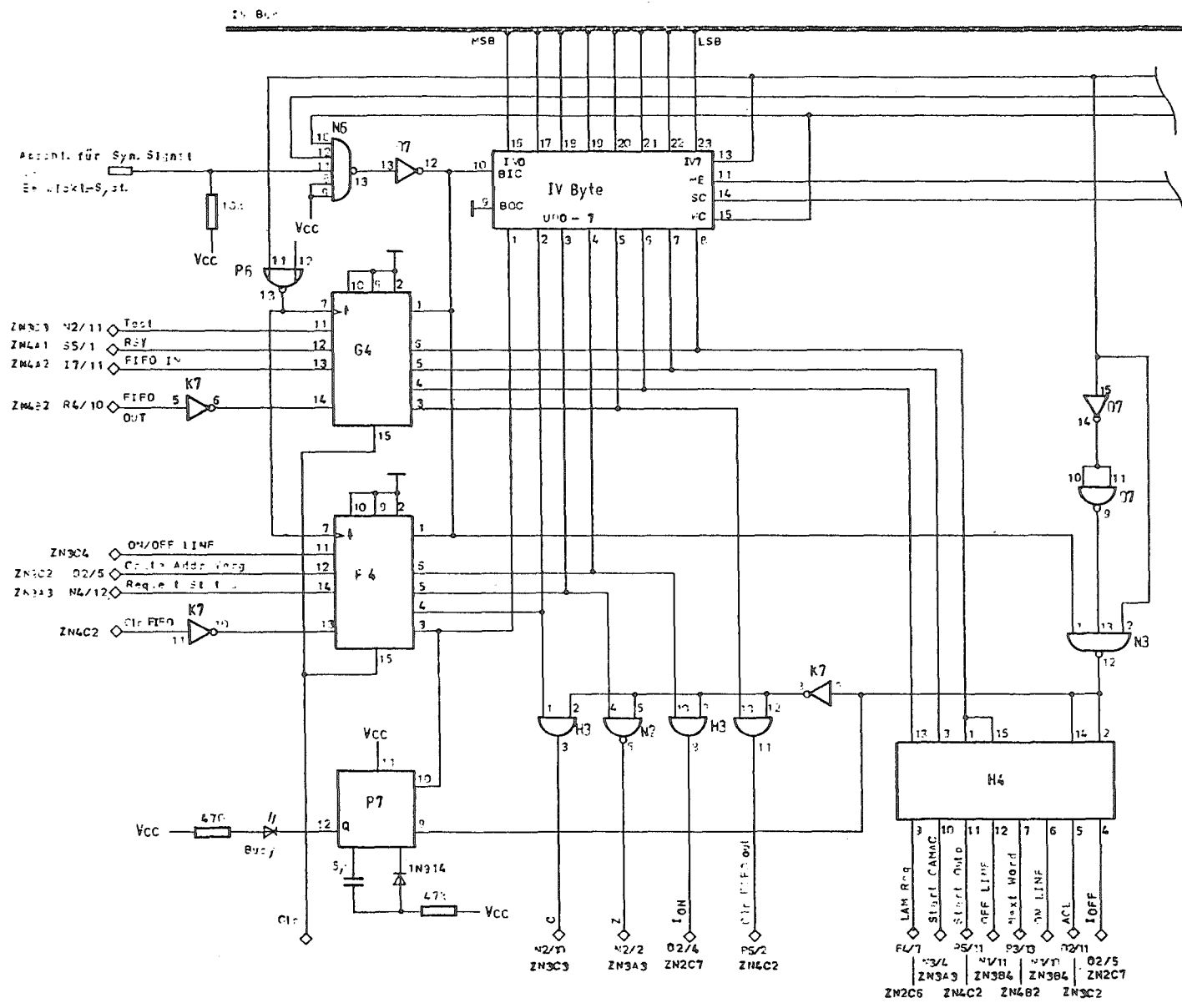


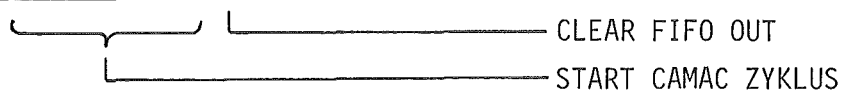
Abb. 8.8-3 IV - Byte mit Steuersignaleingang und -ausgang

Abb. 8.8 - 4 Tabelle der Output - Funktionen vom IV - Byte

MSB								LSB	Funktion	Code	
UD0	UD1	UD2	UD3	UD4	UD5	UD6	UD7	bin.		okt.	
0	0	0	0	0	0	0	0	Strobe LAM's	0	0	
I	0	0	0	0	0	0	0	FIFO - IN NEXT WORD	128	200	
0	I	0	0	0	0	0	0	aufsetzen A C L	64	100	
I	I	0	0	0	0	0	0	START OUTPUT	192	300	
0	0	I	0	0	0	0	0	START CAMAC ZYKLUS	32	40	
I	0	I	0	0	0	0	0	ONLINE	160	240	
0	I	I	0	0	0	0	0	OFFLINE	96	140	
I	I	I	0	0	0	0	0	INHIBIT OFF	224	340	
0	0	0	I	0	0	0	0	CLEAR FIFO OUT	16	20	
0	0	0	0	I	0	0	0	INHIBIT ON	8	10	
0	0	0	0	0	I	0	0	Absetzen "Z"	4	4	
0	0	0	0	0	0	I	0	Absetzen "C"	2	2	
0	0	0	0	0	0	0	I	Anzeige für BUSY LED	1	1	

Absetzen mehrerer Funktionen gleichzeitig möglich, jedoch darunter nur eine Funktion aus Gruppe I!

Beispiel:



9. Mechanischer Aufbau

Die mechanischen Abmessungen des entwickelten PDV - CAMAC - Controllers entsprechen einem doppelt breitem CAMAC - Einschub. Die doppelt breite Ausführung ist für einen CAMAC - Controller die minimal mögliche Breite, da die Signalleitungen für Stationsnummern N und LAM - Signale L des CAMAC - Datenweges auf die Buchsenleisten vom Einschubplatz 24 und 25 verteilt sind. Die Anschlußbuchse für den ACB - Bus ist vereinbarungsgemäß an der Rückseite des Controllers angeordnet. Die Anschlußbuchse für die SDS - Schnittstelle sowie die zur Realisierung einer Daisy - Chain - Leitung erforderlichen LEMO Buchsen sind auf der Frontseite des Controllers untergebracht. Die Stiftbelegung und Signalverteilung des SDS - Steckers kann der Abb. 2 - 1 entnommen werden. Zur Vervollständigung sind die Stecker- bzw. Buchsenbelegungen der CAMAC - Steckplätze 24 und 25 sowie der ACB - Anschlußbuchse in Abb. 9. - 1, 9. - 2 und 9. - 3 wiedergegeben.

STIFTZUORDNUNG BEI EINER NORMALEN STATION
von der Vorderseite des Rahmens gesehen

Freier Sonderstift		P1	B	Belegt	} Sammelleitungen
„		P2	F16	Funktion	
„		P3	F8	„	
„		P4	F4	„	
„		P5	F2	„	
Sammelleitung	Reserviert	X	F1	„	
Sammelleitungen	Sperrn	I	A8	Subadresse	
m. Anschlußstiften	Löschen	C	A4	„	
Stichleitungen	Stationsnummer	N	A2	„	
m. Anschlußstiften	Anforderung	L	A1	„	
Sammelleitung	Strobe 1	S1	Z	Initialisieren	
„	Strobe 2	S2	Q	Echo	
		W24	W23		
		W22	W21		
		W20	W19		
24 Schreibleitungen (Sammelleitungen)		W18	W17		
		W16	W15		
W1 = Bit niedrigster Wertigkeit		W14	W13		
		W12	W11		
W24 = Bit höchster Wertigkeit		W10	W9		
		W8	W7		
		W6	W5		
		W4	W3		
		W2	W1		
		R24	R23		
		R22	R21		
		R20	R19		
24 Leseleitungen (Sammelleitungen)		R18	R17		
		R16	R15		
		R14	R13		
R1 = Bit niedrigster Wertigkeit		R12	R11		
		R10	R9		
R24 = Bit höchster Wertigkeit		R8	R7		
		R6	R5		
		R4	R3		
		R2	R1		
Reserviert für -12 V.		-12	-24	-24V _±	
Reserviert für +200V.		+200	-6	-6V _±	
Reserviert für 117V ~ (Phase)		ACL	ACN	Reserviert für 117V ~ (Null)	
Reserviert		Y1	E	Reserviert für störungsfreie Erde	
Reserviert für +12V		+12	+24	+24V _±	
Reserviert		Y2	+6	+6V _±	
0 Volt (Stromrückführung)		0	0	0 Volt (Stromrückführung)	

Abb. 9. - 1 Buchsenbelegung des CAMAC - Steckplatzes 24

Contact	Signal	Direction	Contact	Signal	Direction
1	Demand Busy	Out	2	L1	Out
3	SGLE1	In	4	L2	Out
5	SGLE2	In	6	L3	Out
7	SGLE3	In	8	L4	Out
9	SGLE4	In	10	L5	Out
11	SGLE5	In	12	L6	Out
13	External Repeat	In	14	L7	Out
15	---		16	L8	Out
17	---		18	L9	Out
19	Time-out	Out	20	L10	Out
21	Demand Message Initiate	In	22	L11	Out
23	Start timer	In	24	L12	Out
25	Selected L's present	In	26	L13	Out
27	---		28	L14	Out
29	Auxiliary Controller Lockout	Out	30	L15	Out
31	Byte Clock	Out	32	L16	Out
33	GND		34	L17	Out
35	Request		36	L18	Out
37	Request Inhibit		38	L19	Out
39	GND		40	L20	Out
41	SCC Busy	Out	42	L21	Out
43	N1	In	44	L22	Out
45	N2	In	46	L23	Out
47	N4	In	48	L24	In/Out
49	N8	In	50	L-SUM	Out
51	N16	In	52	Gnd	---

Abb. 9. - 3 Buchsenbelegung der ACB - Anschlußbuchse

9.1 Frontplatte und Bedienungselemente

Abbildung 9.1 - 1 zeigt die Frontplatte des PDV - CAMAC - Controllers und läßt die Anordnung der Bedienungs- und Kontrollelemente erkennen. An oberster Stelle über allen anderen Bedienungselementen ist die zweistellige Crate-Adressanzeige angeordnet. Das Einstellen der gewünschten Crate Adresse geschieht mit Hilfe von Dual - In - Line Schaltern, die innerhalb des Controllers angeordnet sind. Dabei ist zu beachten, daß die Codierung der Schalter dem BCD - Code entspricht. Unterhalb der Crate - Adress - Anzeige sind die Kontroll - LEDs zur Anzeige des eingeschalteten INHIBIT - Signales und zur " Busy " - Anzeige zu sehen. Die Anzeige des " Busy " Zustandes wird bei erkannter Eigenadresse oder Sammeladressierung eingeschaltet und die Anzeigeweitenspanne durch ein Monoflop verlängert. Zur Anzeige eines vorhandenen LAM - Signales (Summen - LAM) sowie zur Anzeige des ONLINE Zustandes sind zwei weitere LEDs vorhanden. Dabei wird der ONLINE Zustand jedoch nur dann erreicht wenn mit Hilfe des Betriebsartenschalters der ONLINE Zustand vorgeählt ist und über einen entsprechenden Befehl der Mikroprozessor diesen Zustand übernimmt. Die Zuordnung der Mehrfachfunktionen der unterhalb des Betriebsartenschalters angeordneten Tastenschalter kann der Beschriftung und den Sinnlinien entnommen werden. Zur Ankopplung der SDS - Schnittstelle ist der Controller mit der Federleiste einer Steckverbindung entsprechend der Steckernorm FTZ 386TV1 ausgerüstet. Am unteren Ende der Frontplatte sind die zur Realisierung eines Multiprozessorsystems mit zugehöriger Prioritätenverteilung erforderlichen LEMO - Buchsen angeordnet. Die Buchse mit der Bezeichnung " Request " ist direkt mit der entsprechenden Signalleitung vom ACB - Bus verbunden. Durch externe Verkabelung (Daisy Chain) zwischen den Grant Out (G Out) und Grant IN (G IN) Anschlußbuchsen wird eine entsprechende Prioritätenfolge eingestellt. Dabei muß für den Controller bzw. Einschub mit höchster Priorität die Buchsen Request und Grant IN verbunden werden. Das Modul mit höchster Priorität muß dabei nicht zwangsweise auf den Steckplätzen 24 und 25 des Crate Controllers untergebracht sein. Eine von der Daisy-Chain unabhängige Priorität kann auch durch das Setzen des ACL - Signales (Auxilary Controller Lookout) vorgenommen werden. Hierzu sind innerhalb des Controllers ebenfalls zwei Mikroschalter angeordnet.

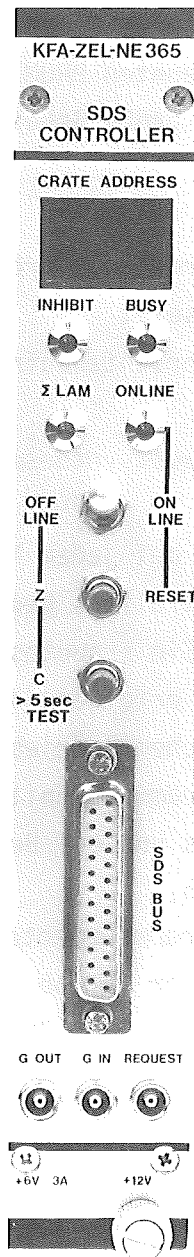


Abb. 9.1 - 1

10. Programmbeschreibung

10.1. Programmstruktur

Das für den PDV - Controller implementierte Programm umfaßt 1K Instruktionen die in einem vom Arbeitsspeicher getrennten PROM gespeichert sind. Der gesamte Programmablaufplan kann in drei große Bereiche untergliedert werden (siehe Abbildung 10 - 1):

- a) Programmeinstieg, hier werden die Arbeitsspeicher gelöscht und spezielle Parameter vorge setzt
- b) Programmabschnitt zur Nachrichtenaufnahme, CRC - Prüfung und Ausführung
- c) Programmabschnitt zur Antwortsynthese und Antwortabgabe

Die Programmabschnitte b und c werden bei jedem Nachrichten- Antwortenzyklus durchlaufen. Beim Einschalten des Controllers beginnt dieser automatisch bei Adresse 0 des Programmspeichers. Ein erneutes Starten bei Adresse 0 kann durch Auslösen der Resetfunktion erfolgen, d.h. Betätigung der "Z" Taste bei in ONLINE Position befindlichem Betriebsschalter. Ein in diesem Blockdiagramm mit aufgeführter Programmabschnitt beinhaltet Testroutinen, die im OFFLINE Zustand des Controllers und bei Betätigung der "C" Taste (länger als 5 sec) ausgeführt werden.

Um den Einstieg in die einzelnen Programmsegmente zu erleichtern, ist in Abb. 10-2 eine weitere Aufgliederung der in Abb. 10-1 gezeigten Programmblöcke aufgeführt. In dieser Abbildung ist wegen der Übersichtlichkeit die Testroutine nicht mehr enthalten.

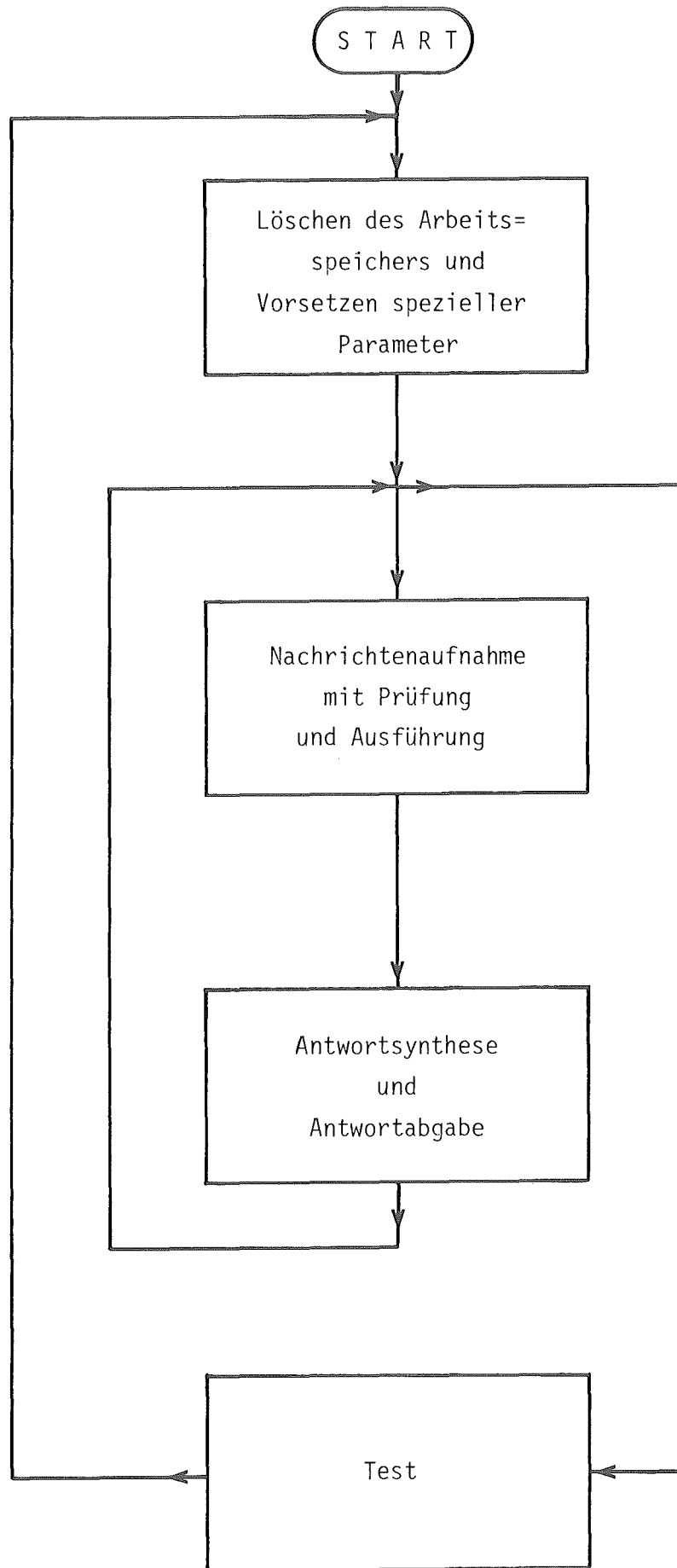


Abb. 10 - 1 Programmstruktur

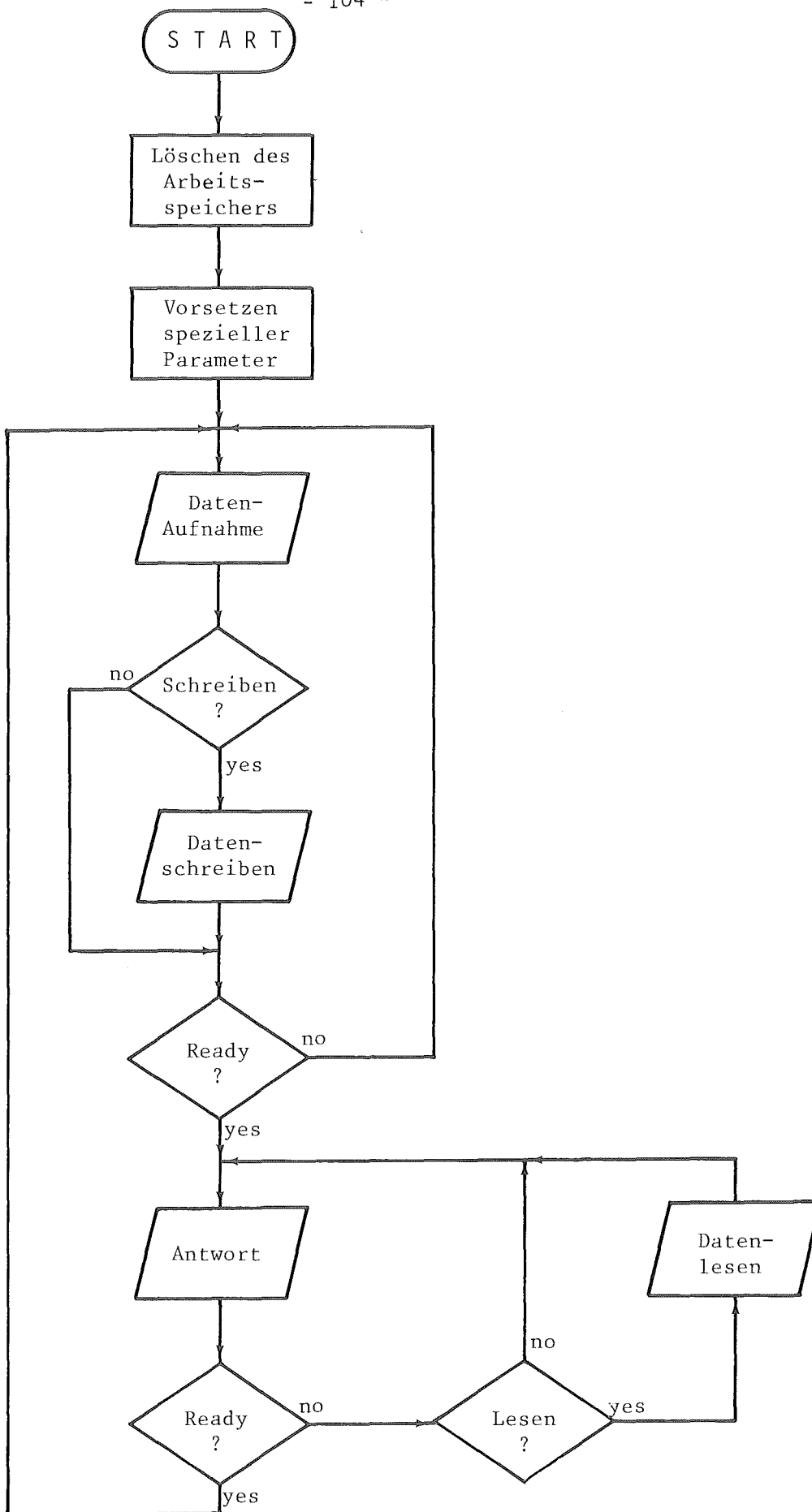


Abb. 10 - 2 Programmstruktur

10.2. Programmsegment - Aufteilung

Um das Programm möglichst verständlich beschreiben zu können ist das Programm in kleine Abschnitte zerlegt und beschrieben. Zu jedem Programmabschnitt ist die entsprechende Programmliste und das dazugehörige Flußdiagramm aufgeführt. Abbildung 10.2. - 1 zeigt alle in den Flußdiagrammen verwendeten Zeichen und gibt deren Bedeutung an.

Zeichenerklärung

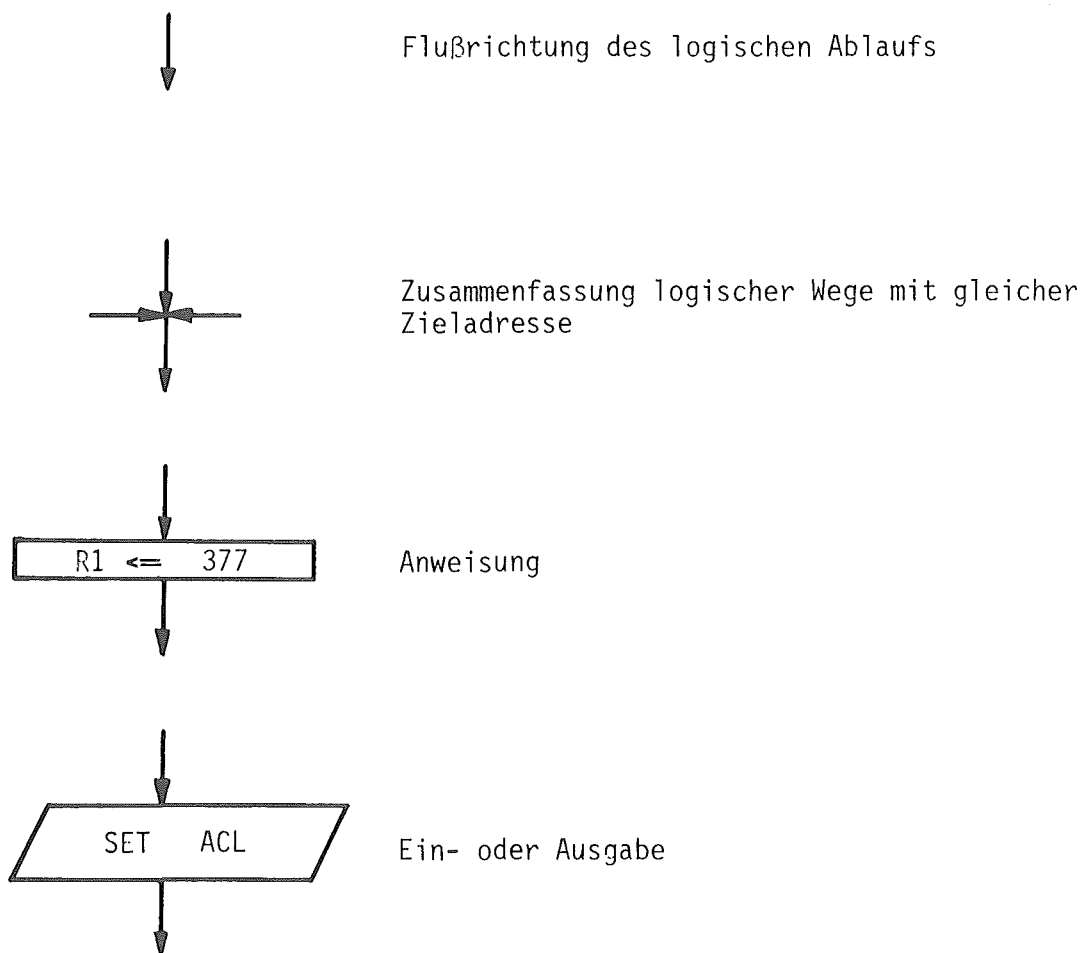
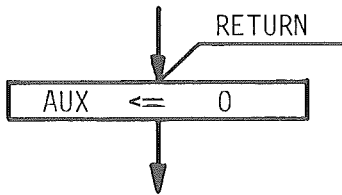
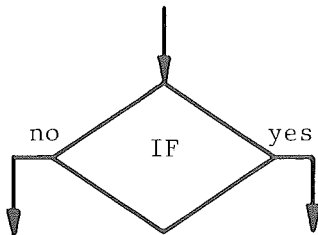


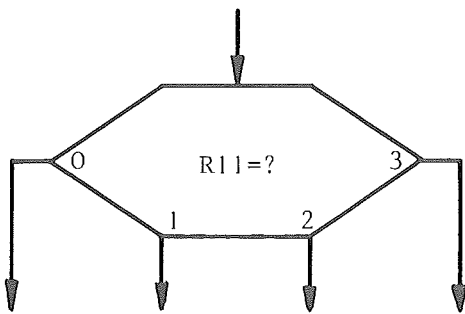
Abb. 10.2.- 1



Anweisung mit Statement "RETURN" (Sprungziel)



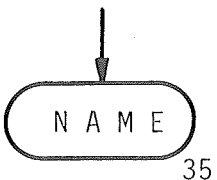
logische Verzweigung (bedingte Sprunganweisung)



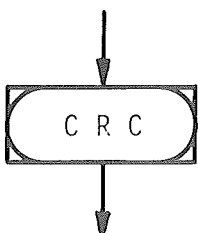
logische Mehrfachverzweigung (Computed GO TO)
(Sprungziel abhängig vom Inhalt einer
Variablen z.B. R11)



Hauptstatement (Programmsegment "N A M E")



Absprung zu Programmsegment "N A M E"
(auf Programmlisting Seite 35)



Aufruf des Unterprogramms C R C

10.2.1 Programmsegment ANFANG

Dieses Programmsegment steht ab Adresse 0 am Beginn des Programmspeichers und stellt die Betriebsbereitschaft des Controllers auf Hard- und Softwarebasis her. Dieses Programmsegment wird automatisch beim Einschalten des Controllers (Powerclear) und beim Auslösen der Resetfunktion aktiviert. Im Einzelnen übernimmt dieses Programmsegment folgende in logischer Reihenfolge aufgeführten Funktionen wahr:

- a) Einschalten des IV - Bytes auf der linken Bank und Vorsetzen zweier Parameter
- b) Löschen des RAM - Arbeitsspeichers von RAM - Adresse 10 bis RAM - Adresse 377₈.
- c) Aufsetzen des internen Request - Signales zum Auslösen einer CAMAC - Funktion
- d) Vorsetzen weiterer Parameter
- e) Auslösen der CAMAC - Initialisierungsfunktion "Z"
- f) Auslesen der eingestellten Crate Adresse durch hochzählen und ausschreiben einer Zahlenrampe
- g) Umwandlung der im BCD - Code gelesenen Crate Adresse in den Binär-code und Ablegen unter der RAM - Adresse 20
- h) Signalüberprüfung an der SDS - Schnittstelle zum Auffinden einer Nachrichtenlücke und Einsynchronisierung in das PDV Nachrichten- und Antwortenprotokoll
- i) OFFLINE schalten des Crate Controllers und Vorsetzen entsprechender Statusbits im RAM Statusregister 367

Abbildung 10.2.1-1 zeigt das Flußdiagramm des Programmsegmentes ANFANG und die Abbildung 10.2.1-2 das zu diesem Flußdiagramm gehörende Programmlisting.

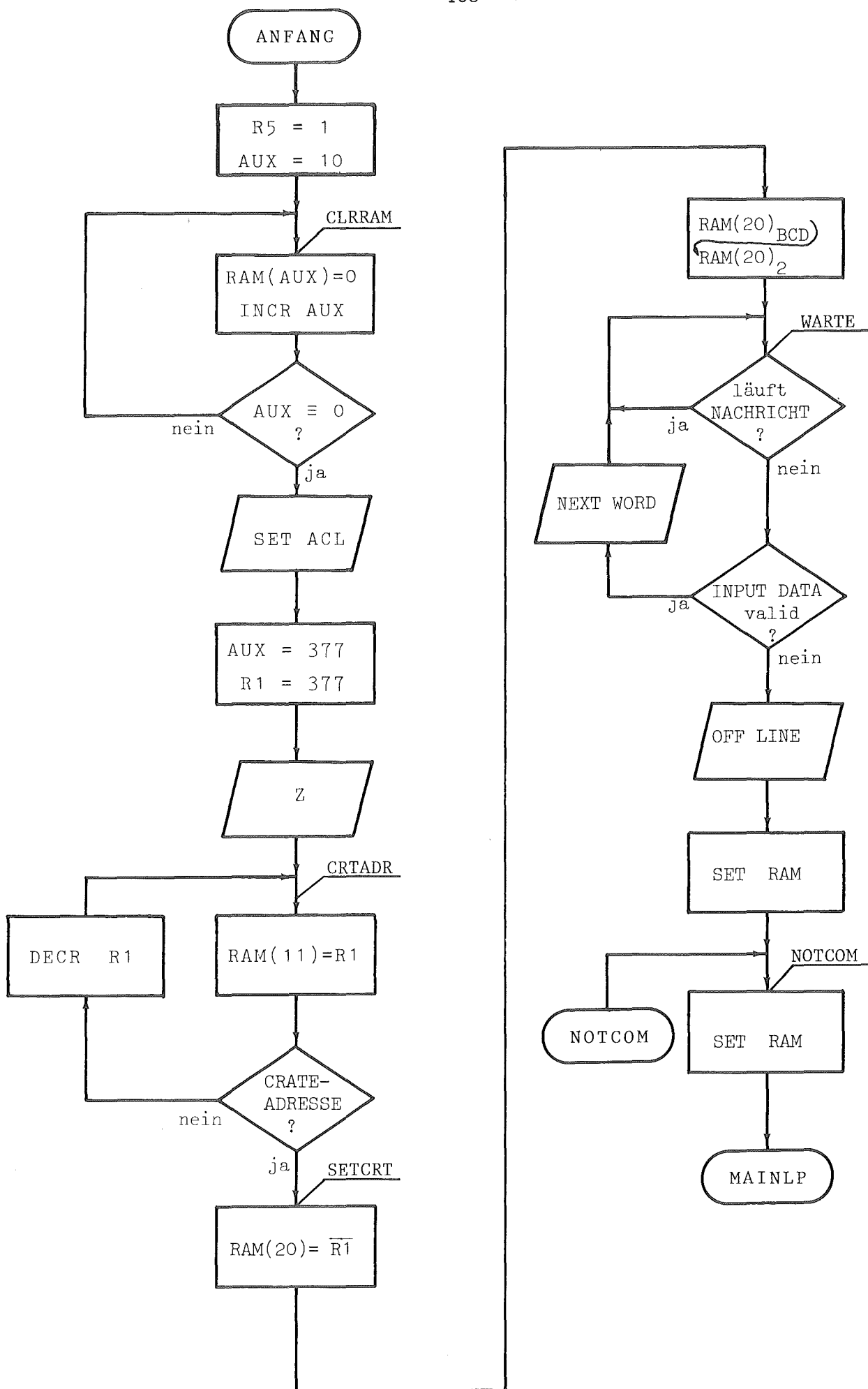


Abb. 10.2.1-1

1	000000		,ASECT	
2	000000		PARAM	
3	000000	ANFANG:	XMIT	IVL,1
	000000	143401		
4	000002		XMIT	R5,1
	000002	142401		
5	000004		XMIT	AUX,10
	000004	140010		
6	000006	CLRRAM:	MOVE	AUX,0,IVR
	000006	000017		
7	000010		XOR	AUX,0,RIV7
	000010	060037		
8	000012		ADD	R5,0,AUX
	000012	022400		
9	000014		NZT	AUX,CLRRAM
	000014	120003		
10	00016		XMIT	AUX,100
	00016	140100		
11	00020		MOVE	AUX,0,LIV7
	00020	000027		
12	00022		XMIT	AUX,377
	00022	140377		
13	00024		XMIT	R1,377
	00024	140777		
14	00026		XMIT	R4,4
	00026	142004		
15	00030		MOVE	R4,0,LIV7
	00030	002027		
16	00032		XMIT	IVR,11
	00032	147411		
17	00034	CRTADR:	MOVE	R1,0,RIV7
	00034	000437		
18	00036		NZT	LIV5,1,SETCRT
	00036	132462		
19	00040		ADD	R1,0,R1
	00040	020401		
20	00042		JMP	CRTADR
	00042	160016		
21	00044	SETCRT:	XMIT	IVR,20
	00044	147420		
22	00046		XOR	R1,0,RIV7
	00046	060437		
23	00050		MOVE	RIV7,4,AUX
	00050	017600		
24	00052		MOVE	RIV3,4,R1
	00052	015601		
25	00054		ADD	R1,5,AUX
	00054	020640		
26	00056		ADD	R1,7,R1
	00056	020741		
27	00060		MOVE	R1,0,RIV7
	00060	000437		
28	00062		XMIT	R6,200
	00062	143200		
29	00064	WARTE:	NZT	LIV1,1,WARTE
	00064	130472		
30	00066		NZT	LIV2,1,SETRAM

```
00066 131076
31 00070          MOVE      R6,0,LIV7
00070 003027
32 00072          JMP      WARTE
00072 160032
33 00074          SETRAM: XMIT    AUX,140
00074 140140
34 00076          MOVE      AUX,0,LIV7
00076 000027
35 00100          XMIT      IVR,367
00100 147767
36 00102          XMIT      RIV1,2,3
00102 154503
37 00104          NOTCOM: XMIT    RIV7,1,1
00104 157441
```

Abb. 10.2.1-2 Fortsetzung

10.2.2 Programmsegment MAINLP

Dieses Programmsegment wird nach jeder empfangenen und ausgewerteten Nachricht angesprungen und stellt den Programmbeginn für das Nachrichten- Antwortenprotokoll dar. Zu Beginn dieses Programmsegments wird der Wortzähler (R4) auf 0 gesetzt und das Register R6 mit dem Bitpattern zum Stack - Transfer der FIFO - Daten geladen.

Die mit dem Statement LOOP beginnende und bis zum Statement IN reichende Programmschleife wird in Ruhe bzw. Wartezustand des Controllers kontinuierlich durchlaufen. Innerhalb dieser Softwareschleife wird der ON- bzw. OFF-LINE Zustand des Controllers, die CLEAR - Tastenfunktion und das Signal, welches das Anstehen valider Eingangsdaten signalisiert, geprüft. In Abhängigkeit der abgetesteten Bedingungen wird das ONLINE-OFFLINE-Bit gesetzt bzw. die Schleife zum Daten einlesen verlassen oder der Sprung zum Programmsegment TEST ausgeführt. Mit dem Statement DATAIN beginnt die Einleseroutine valider Daten. Durch die auf die Statements INBYT1, INBYT2, INBYT3 folgenden Instruktionen werden die empfangenen Bytes aus den FIFO's ausgelesen und durch anschließenden Aufruf des Unterprogramms CRC auf Übertragungsfehler kontrolliert. Im Falle einer Übertragungsfehlererkennung wird ins Unterprogramm FALSE abgesprungen. Liegt kein Übertragungsfehler vor, so wird das Programm in Abhängigkeit vom Inhalt des Wortzählers R4 bei den Programmsegmenten WORD0, WORD1, WORD2, WORD3, WORDN, WORDNL, bzw. WRDEND fortgesetzt.

Das Flußdiagramm dieses Programmabschnittes ist aufgrund des Umfanges in den 3 Abbildungen 10.2.2-1, 10.2.2-2 und 10.2.2-3 dargestellt. Das Programmlisting ist den Abbildungen 10.2.2-4 und 10.2.2-5 zu entnehmen.

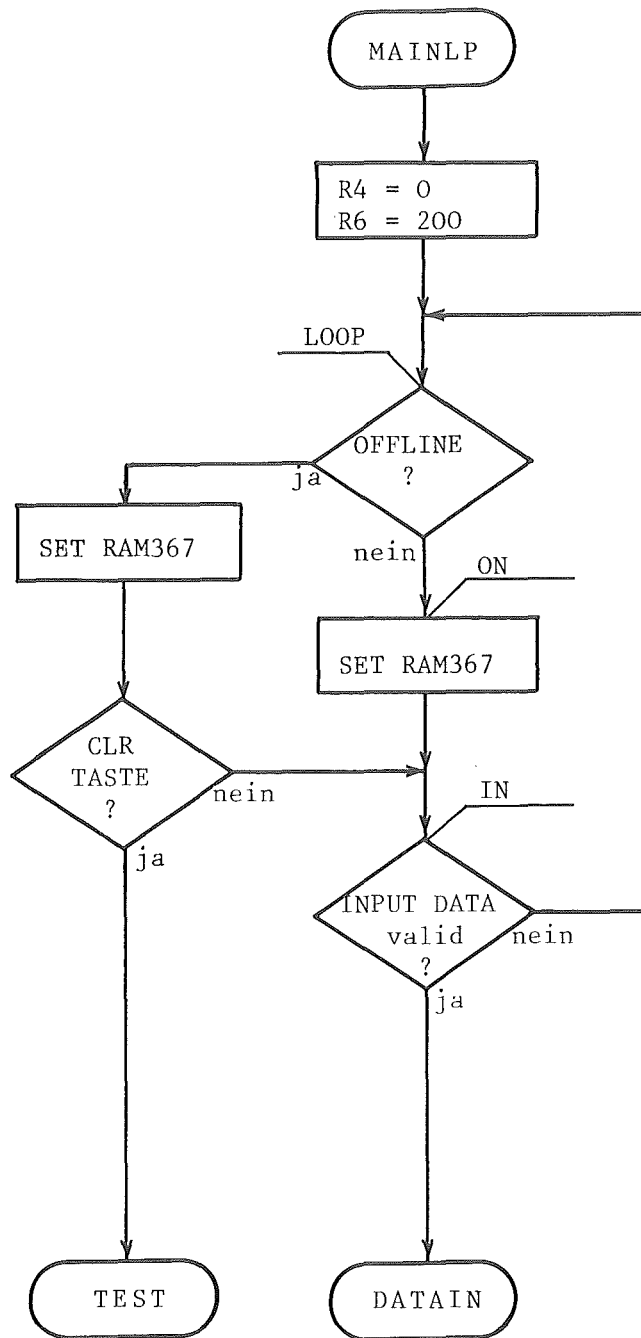


Abb. 10.2.2-1

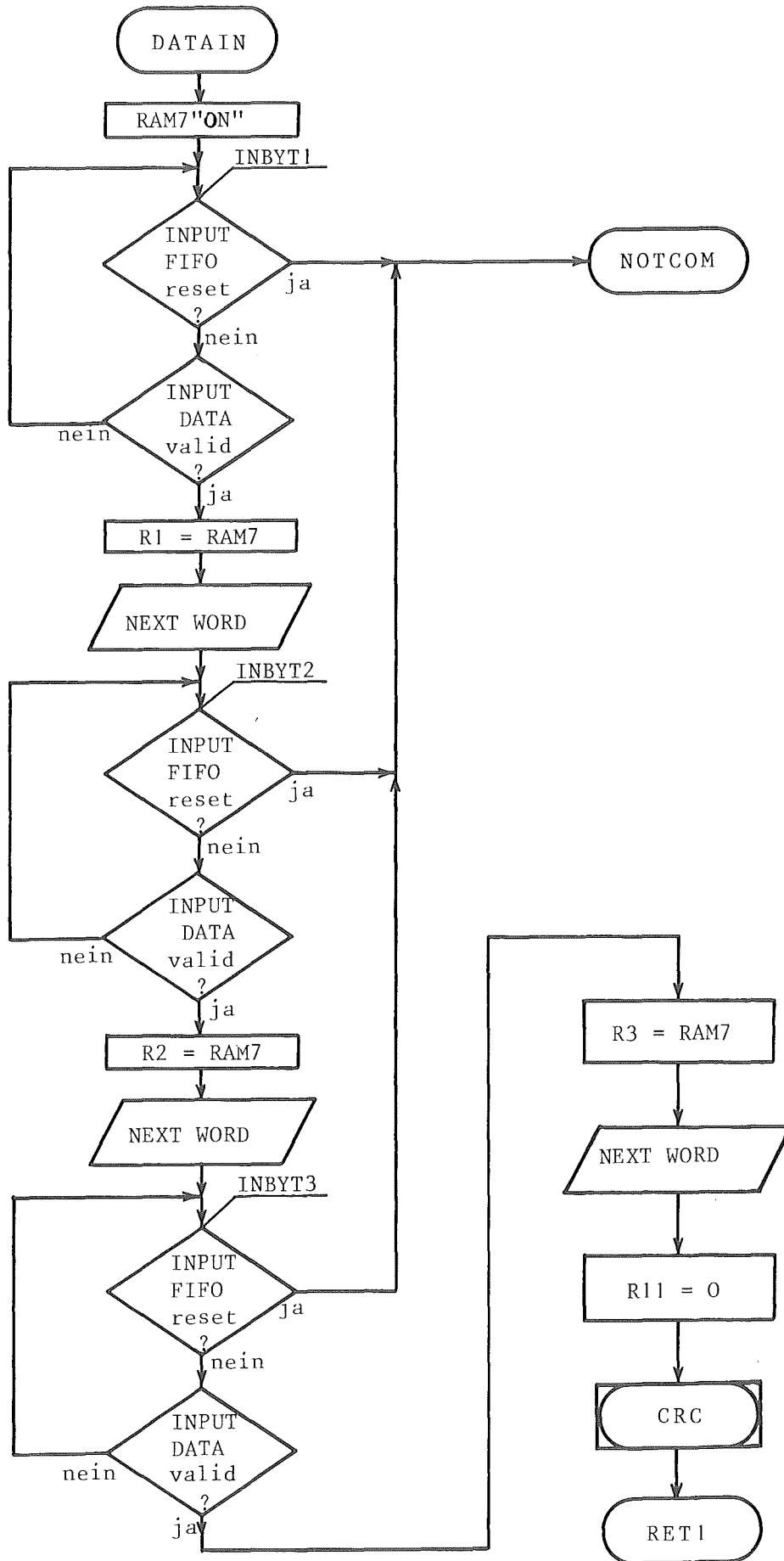


Abb. 10.2.2-2

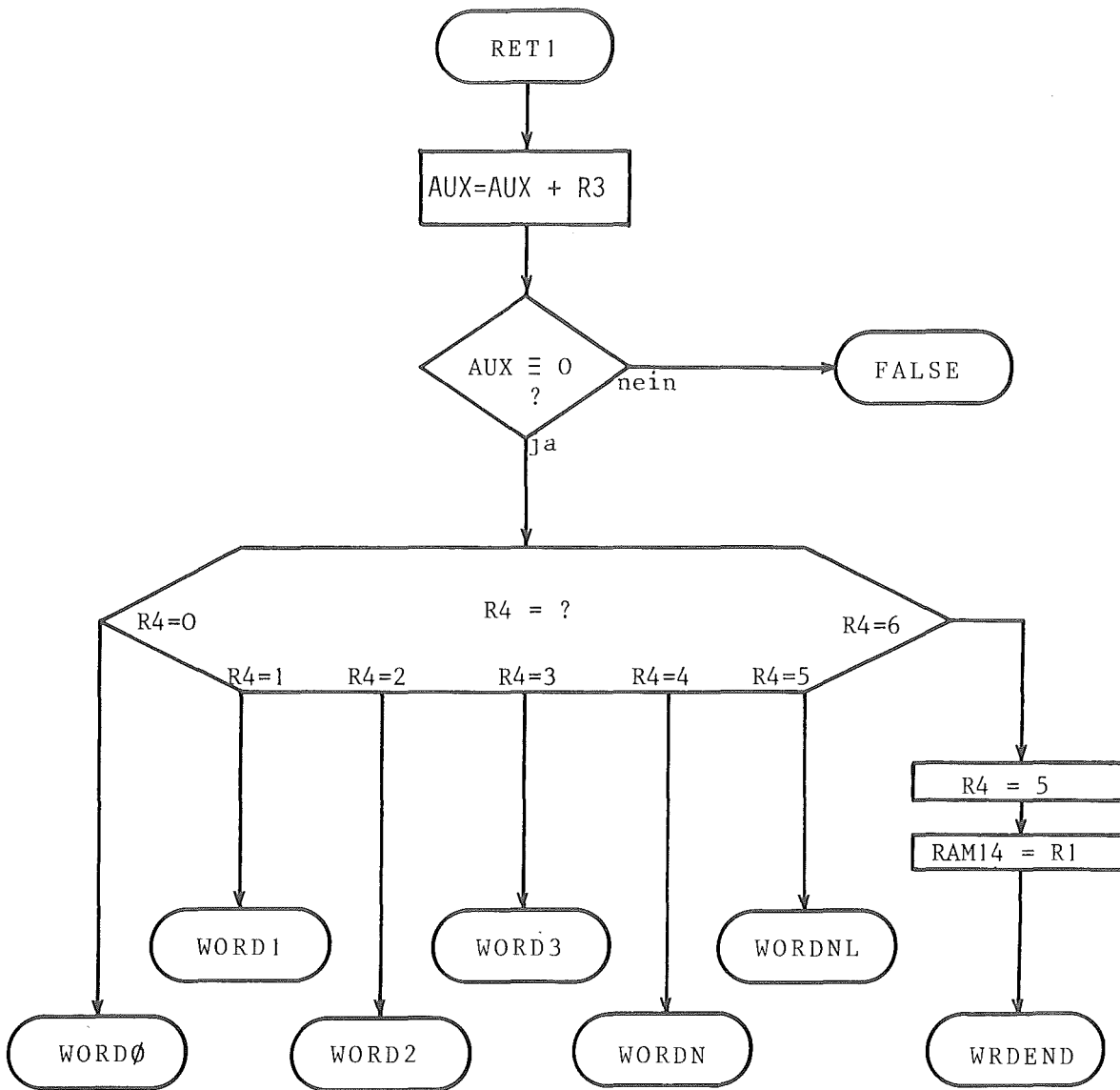


Abb. 10.2.2-3

38	00106	MAINLP:	XMIT	R4,0
	00106	142000		
39	00110		XMIT	R6,200
	00110	143200		
40	00112		XMIT	IVR,367
	00112	147767		
41	00114	LOOP:	NZT	LIV4,1,ON
	00114	132052		
42	00116		XMIT	RIV0,1,1
	00116	154041		
43	00120		NZT	LIV0,1,IN
	00120	130053		
44	00122		JMF	TEST
	00122	160770		
45	00124	ON:	XMIT	RIV0,1,0
	00124	154040		
46	00126	IN:	NZT	LIV2,1,LOOP
	00126	131046		
47	00130	DATAIN:	XMIT	IVR,7
	00130	147407		
48	00132	INBYT1:	NZT	LIV6,1,NOTCOM
	00132	133042		
49	00134		NZT	LIV2,1,INBYT1
	00134	131055		
50	00136		MOVE	RIV7,0,R1
	00136	017401		
51	00140		MOVE	R6,0,LIV7
	00140	003027		
52	00142	INBYT2:	NZT	LIV6,1,NOTCOM
	00142	133042		
53	00144		NZT	LIV2,1,INBYT2
	00144	131061		
54	00146		MOVE	RIV7,0,R2
	00146	017402		
55	00150		MOVE	R6,0,LIV7
	00150	003027		
56	00152	INBYT3:	NZT	LIV6,1,NOTCOM
	00152	133042		
57	00154		NZT	LIV2,1,INBYT3
	00154	131065		
58	00156		MOVE	RIV7,0,R3
	00156	017403		

Abb. 10.2.2-4

59	00160		MOVE	R6,0,LIV7
	00160	003027		
60	00162		XMIT	R11,0
	00162	144400		
61	00164		JMP	CRC
	00164	160774		
62	00166	RET1:	XOR	R3,0,AUX
	00166	061400		
63	00170		NZT	AUX,FALSE
	00170	120110		
64	00172		XEC	R4,TABLE1
	00172	102101		
65	00174		XMIT	IUR,14
	00174	147414		
66	00176		MOVE	R1,0,RIU7
	00176	000437		
67	00200		JMP	WRDEND
	00200	160322		
68	00202	TABLE1:	JMP	WORD0
	00202	160127		
69	00204		JMP	WORD1
	00204	160175		
70	00206		JMP	WORD2
	00206	160202		
71	00210		JMP	WORD3
	00210	160211		
72	00212		JMP	WORDN
	00212	160315		
73	00214		JMP	WORDNL
	00214	160314		
74	00216		XMIT	R4,5
	00216	142005		

Abb. 10.2.2-5

10.2.3 CRC - Kontrolle und CRC - Generierung

Sowohl bei Nachrichten als auch bei Antworten werden jeweils 2 Bytes durch einen CRC (Cyclic Redundancy Check) gesichert. Da für den dem PDV-Bus zugrunde gelegten CRC ein Generatorpolynom 8ten Grades benutzt wird können die zur Kontrolle und Bildung des CRC erforderlichen Aufgaben ohne großen Aufwand in die Software verlagert werden. Ein Aufgabenpunkt des eingesetzten Mikroprozessors ist die CRC - Kontrolle und CRC - Generierung. Dadurch kann der Aufwand an Hardwareelementen reduziert werden und bei Bedarf beliebige Generatorpolynome 8ten Grades eingesetzt werden.

Das für den entwickelten Controller eingesetzte Verfahren zur softwaremäßigen CRC - Kontrolle und CRC Generierung beruht auf einem Tabellenverfahren. Da das Generatorpolynom 8ten Grades ist, beschränkt sich die dafür erforderliche Tabelle auf 256 Worte zu 8 bit. Um den CRC über 16 Datenbits entsprechend 2 Bytes zu bilden, werden die folgenden nacheinander aufgeführten Softwareschritte benutzt.

1. das 1. Byte der 16bit breiten Dateninformation wird als Adresse der CRC - Tabelle gewertet und der unter dieser Adresse stehende " Teil - CRC " gelesen
2. zwischen dem gelesenen " Teil - CRC " und dem 2. Byte des 16bit breiten Datenwortes wird eine Exklusiv - Oder Verknüpfung durchgeführt
3. das Ergebnis dieser Exklusiv - Oder Verknüpfung wird wiederum als Adresse der CRC - Tabelle gewertet. Das aus der CRC - Tabelle gelesene Ergebnis entspricht dem gesuchten 8bit CRC über die 16bit breite Dateninformation.

Das Flußdiagramm dieses Softwareabschnittes ist in Abb. 10.2.3 - 1 zu sehen. Da aus softwaretechnischen Gründen mit dem Instruktionssatz des 8X300 Prozessors eine direkte Verzweigung aufgrund eines Registerinhaltes lediglich über 255 Adressen möglich ist wird der Registerinhalt "0" in einer vorgeschalteten Abfrage detektiert, bearbeitet (in diesem Falle der CRC-Tabelle ist keine weitere Verarbeitung erforderlich) und zum Ausgang der bedingten Verzweigung gesprungen.

Da zur Bildung des CRC's die Tabelle zweimal benutzt werden muß und der CRC - Softwareabschnitt als Unterprogramm aufgebaut ist, wurde die dafür erforderliche logische Verzweigung mit der Rücksprungtabelle kombiniert. Durch den Instruktionssatz des eingesetzten Mikroprozessors 8X300 werden zur Bildung des CRC's maximal 14 Cyclen benötigt. In diesen 14 Cyclen sind die für den Rücksprung aus der als Unterprogramm aufgebauten CRC - Routine benötigten Instruktionen bereits enthalten. Bei der zugrunde gelegten Zykluszeit des Mikroprozessors von 250ns ist damit die Bildung des CRC's in 3,5µs abgeschlossen.

Zur CRC - Kontrolle wird die gleiche Softwareroutine benutzt, d.h. über das empfangene 16bit Datenwort wird ein CRC gebildet und dieser generierte CRC mit dem empfangenen CRC verglichen.

Das Programmlisting für das Unterprogramm zur CRC - Bildung kann der Abbildung 10.2.3-2 entnommen werden.

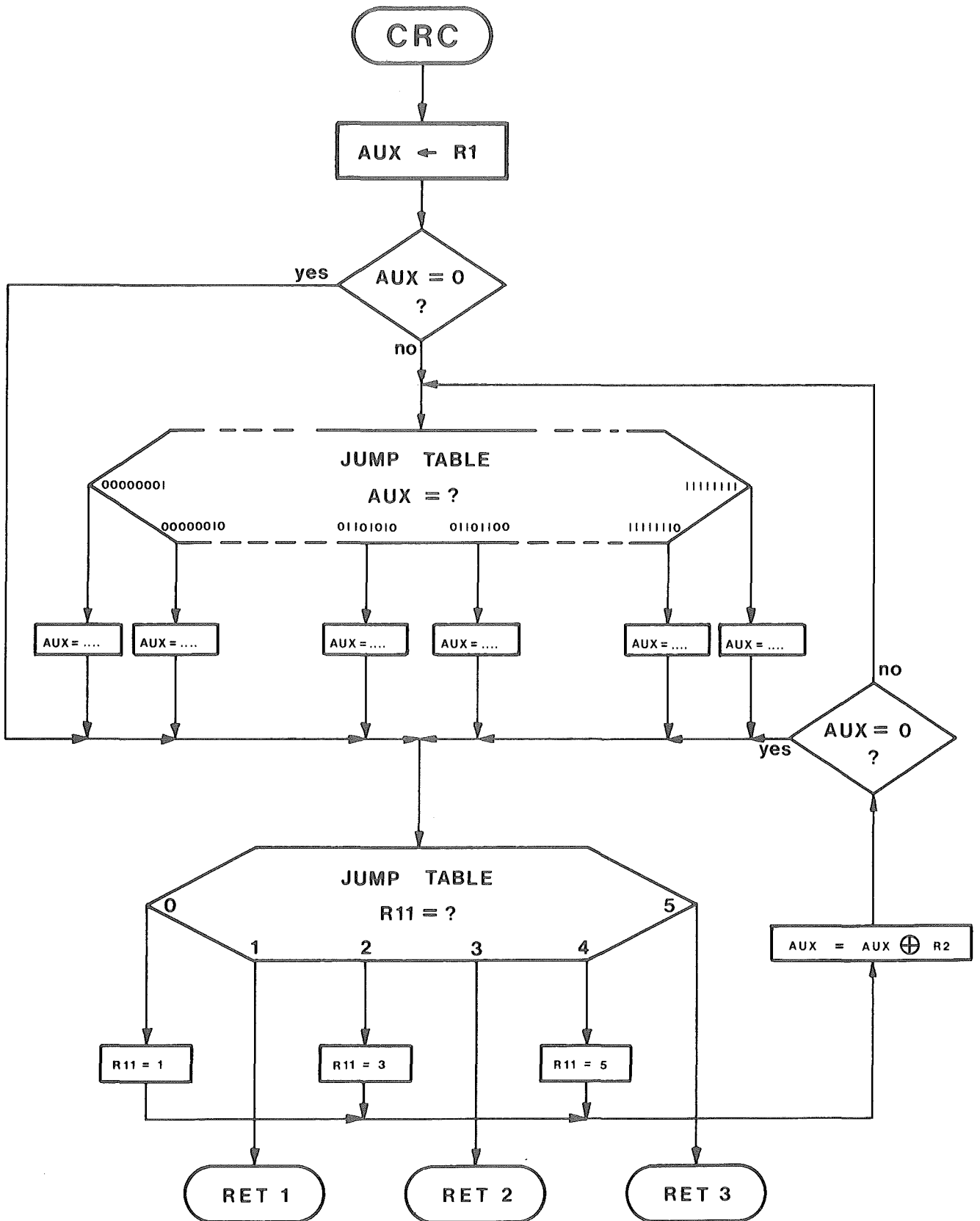


Abb. 10.2.3 - 1

511	1770	CRC:	MOVE	R1,0,AUX
	1770	000400		
512	1772		NZT	AUX,PAGE
	1772	120377		
513	1774		JMP	NEXT
	1774	161400		
514	1776	PAGE:	JMP	CRCTAB
	1776	161377		
515	2000		XMIT	AUX, 145,
	2000	140221		
516	2002		XMIT	AUX, 227,
	2002	140343		
517	2004		XMIT	AUX, 114,
	2004	140162		
	2770	140054		
770	2772		XMIT	AUX, 94,
	2772	140136		
771	2774		XMIT	AUX, 207,
	2774	140317		
772	2776	CRCTAB:	XEC	AUX,CRCTAB
	2776	100377		
773	3000	NEXT:	XEC	R11,RETURN
	3000	104405		
774	3002		XOR	R2,0,AUX
	3002	061000		
775	3004		NZT	AUX, TABLE
	3004	120004		
776	3006		JMP	NEXT
	3006	161400		
777	3010	TABLE:	JMP	CRCTAB
	3010	161377		
778	3012	RETURN:	XMIT	R11,1
	3012	144401		
779	3014		JMP	RET1
	3014	160073		
780	3016		XMIT	R11,3
	3016	144403		
781	3020		JMP	RET2
	3020	161462		
782	3022		XMIT	R11,5
	3022	144405		
783	3024		JMP	RET3
	3024	160762		
784	3026		MOVE	AUX,0,AUX
	3026	000000		
785	3030		MOVE	AUX,0,AUX
	3030	000000		

10.2.4 Programmsegment FALSE

Wird bei einer Nachricht ein CRC - Fehler erkannt so wird in diesem Programmsegment in Statusregister (RAM - Zelle 367) das zur Kennzeichnung eines CRC - Fehlers zugeordnete Bit gesetzt und die Wortnummer, bis zu der die Nachricht korrekt empfangen wurde, abgespeichert (RAM375). Der Absprung von diesem Programmsegment erfolgt in das Unterprogramm LOESCH wo der Rest der einlaufenden Nachricht gelöscht wird.

Abbildung 10.2.4-1 zeigt das Flußdiagramm dieses Programmsegmentes und Abbildung 10.2.4-2 das zugehörige Programmlisting.

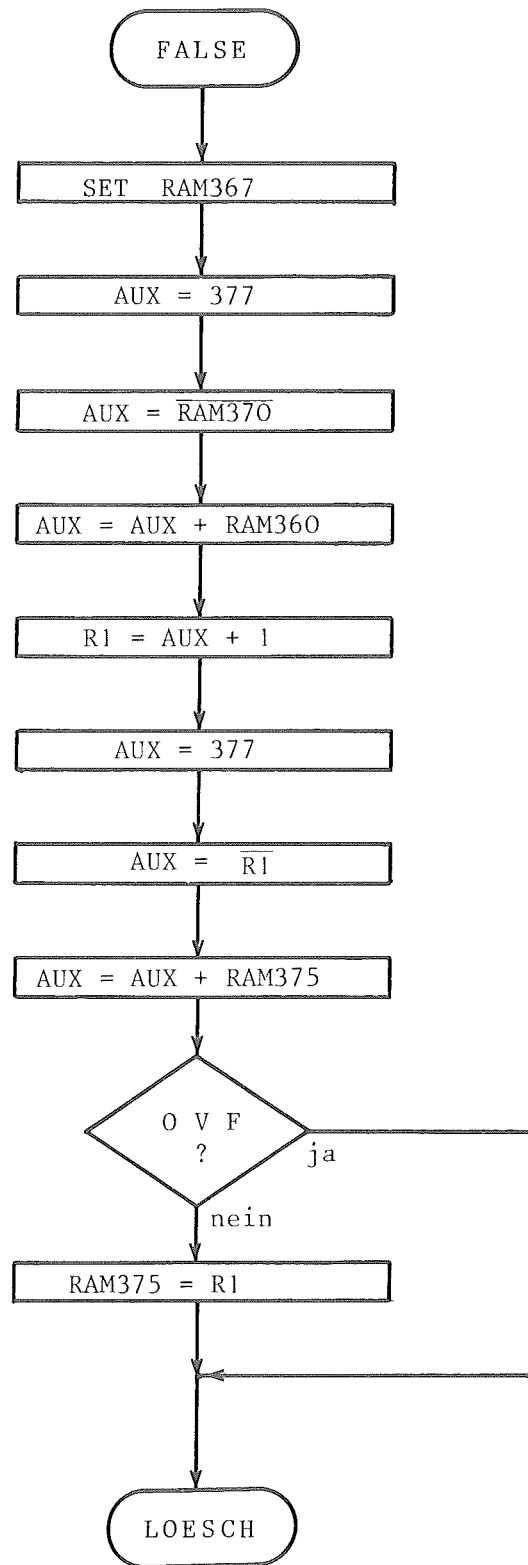


Abb. 10.2.4-1

75	00220	FALSE:	XMIT	IVR,367
	00220	147767		
76	00222		XMIT	RIV4,1,1
	00222	156041		
77	00224		XMIT	AUX,377
	00224	140377		
78	00226		XMIT	IVR,370
	00226	147770		
79	00230		XOR	RIV7,0,AUX
	00230	077400		
80	00232		XMIT	IVR,366
	00232	147766		
81	00234		ADD	RIV7,0,AUX
	00234	037400		
82	00236		ADD	R5,0,R1
	00236	022401		
83	00240		XMIT	AUX,377
	00240	140377		
84	00242		XOR	R1,0,AUX
	00242	060400		
85	00244		XMIT	IVR,375
	00244	147775		
86	00246		ADD	RIV7,0,AUX
	00246	037400		
87	00250		NZT	OVF,LOESCH
	00250	124263		
88	00252		MOVE	R1,0,RIV7
	00252	000437		
89	00254		JMP	LOESCH
	00254	160263		

Abb. 10.2.4-2

10.2.5 Programmsegment WORDO

Unter diesem Programmsegment wird das erste empfangene Wort überprüft und ausgewertet.

An erster Stelle wird überprüft ob die empfangene Sendung eine Nachricht oder Antwort ist (Wird als PDV - Bus - Übertragungssystem ein Einleitungssystem verwendet so werden von jeder angeschlossenen Unterstation auch die Antworten anderer Unterstationen empfangen). Wird die empfangene Sendung als Antwort erkannt so wird ins Unterprogramm LOESCH gesprungen und der Rest der empfangenen Sendung gelöscht. Trägt die empfangene Sendung das Kennzeichen einer Nachricht so wird ab Statement WORK die Adresse der empfangenen Nachricht mit der eigenen Adresse verglichen. Stimmt die Adresse der empfangenen Nachricht nicht mit der eigenen Adresse überein so wird ab Statement NOCRAD nach Überprüfung auf ONLINE - Zustand des Controllers die Adresse der Nachricht mit der Sammeladresse verglichen und im Falle der Übereinstimmung in das Unterprogramm SAMMEL gesprungen. Wird auch mit der Sammeladresse keine Übereinstimmung festgestellt so wird im Statusregister das Bit für Fremdadressierung gesetzt und der Rest der Nachricht gelöscht. Fällt die Überprüfung mit der eigenen Adresse positiv aus so wird in Abhängigkeit vom 2. Bit des 2. Bytes in das Unterprogramm zur Auswertung von Einzelwortübertragungen (CRTFKT) gesprungen bzw. nach Setzen des Wortzählers R4 zum Statement DATAIN gesprungen und die folgenden PDV - Wörter empfangen und ausgewertet.

Das Flußdiagramm des Programmsegmentes WORDO ist in Abbildung 10.2.5-1 zu sehen, das entsprechende Programmlisting zeigt Abbildung 10.2.5-2.

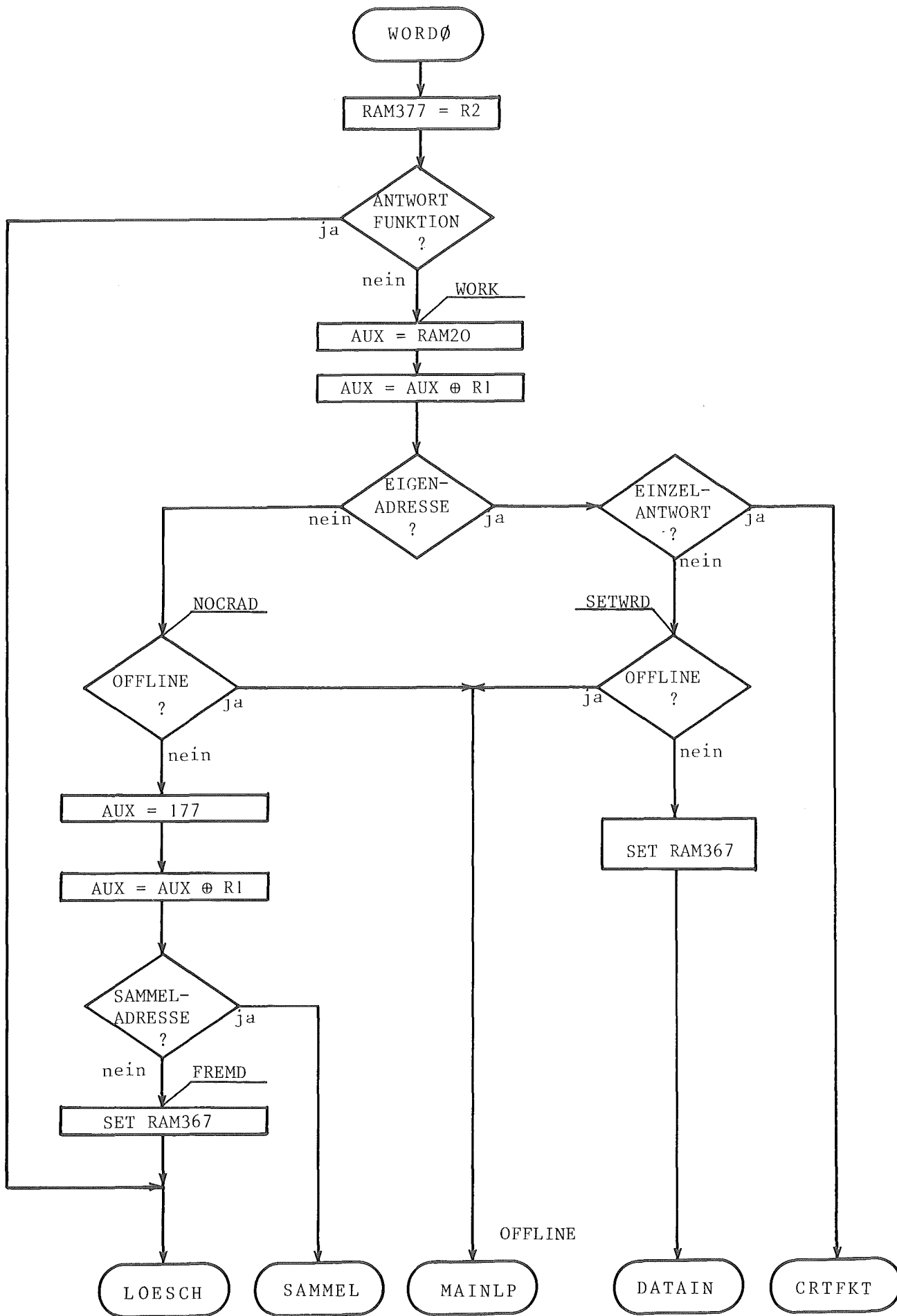


Abb. 10.2.5-1

90	00256	WORDO:	XMIT	IVR,377
	00256	147777		
91	00260		MOVE	R2,0,RIV7
	00260	001037		
92	00262		NZT	RIV7,1,WORK
	00262	137473		
93	00264		JMP	LOESCH
	00264	160263		
94	00266	WORK:	XMIT	IVR,20
	00266	147420		
95	00270		MOVE	RIV7,0,AUX
	00270	017400		
96	00272		XOR	R1,0,AUX
	00272	060400		
97	00274		NZT	AUX,NOCRAD
	00274	120143		
98	00276		MOVE	R5,0,LIV7
	00276	002427		
99	00300		XMIT	IVR,377
	00300	147777		
100	0302		NZT	RIV6,1,SETWRD
	0302	137054		
101	0304		JMP	CRTFKT
	0304	160551		
102	0306	NOCRAD:	XMIT	IVR,367
	0306	147767		
103	0310		NZT	RIV0,1,OFFLNE
	0310	134060		
104	0312		XMIT	AUX,177
	0312	140177		
105	0314		XOR	R1,0,AUX
	0314	060400		
106	0316		NZT	AUX,FREMD
	0316	120152		
107	0320		MOVE	R5,0,LIV7
	0320	002427		
108	0322		JMP	SAMMEL
	0322	160670		
109	0324	FREMD:	XMIT	RIV3,1,1
	0324	155441		
110	0326		JMP	LOESCH
	0326	160263		
111	0330	SETWRD:	XMIT	IVR,367
	0330	147767		
112	0332		NZT	RIV0,1,OFFLNE
	0332	134060		
113	0334		XMIT	R4,1
	0334	142001		
114	0336		JMP	DATAIN
	0336	160054		
115	0340	OFFLNE:	JMP	MAINLP
	0340	160043		

10.2.6 Programmsegment WORD1

Zu Beginn des Programmsegmentes WORD1 wird der Funktionenteil des ersten Wortes daraufhin überprüft ob die empfangene Nachricht der Rücksendung einer zuvor abgegebenen Statusübertragung entspricht. Ist dies der Fall so wird ins Unterprogramm RESTAT gesprungen und einzelne Alarmschwellen gezielt zurückgesetzt (nicht bearbeitet). Ist die empfangene Nachricht keine rückübertragene Statusmeldung so wird die in diesem Wort empfangene Blocklänge der Nachricht in der RAM - Zelle 370 und 366 abgelegt, die CAMAC - Stationsnummer in RAM - Zelle 374 gespeichert und in Abhängigkeit von der Funktion Schreiben und Lesen der Wortzähler R4 auf 2 bzw. 3 gesetzt. Daraufhin wird im Programmabschnitt INCWRD überprüft ob das Ende der Nachricht erreicht ist und zum Statement DATAIN zurückgesprungen.

Abbildung 10.2.6-1 zeigt das Flußdiagramm und Abbildung 10.2.6-2 das Programmlisting dieses Programmsegmentes.

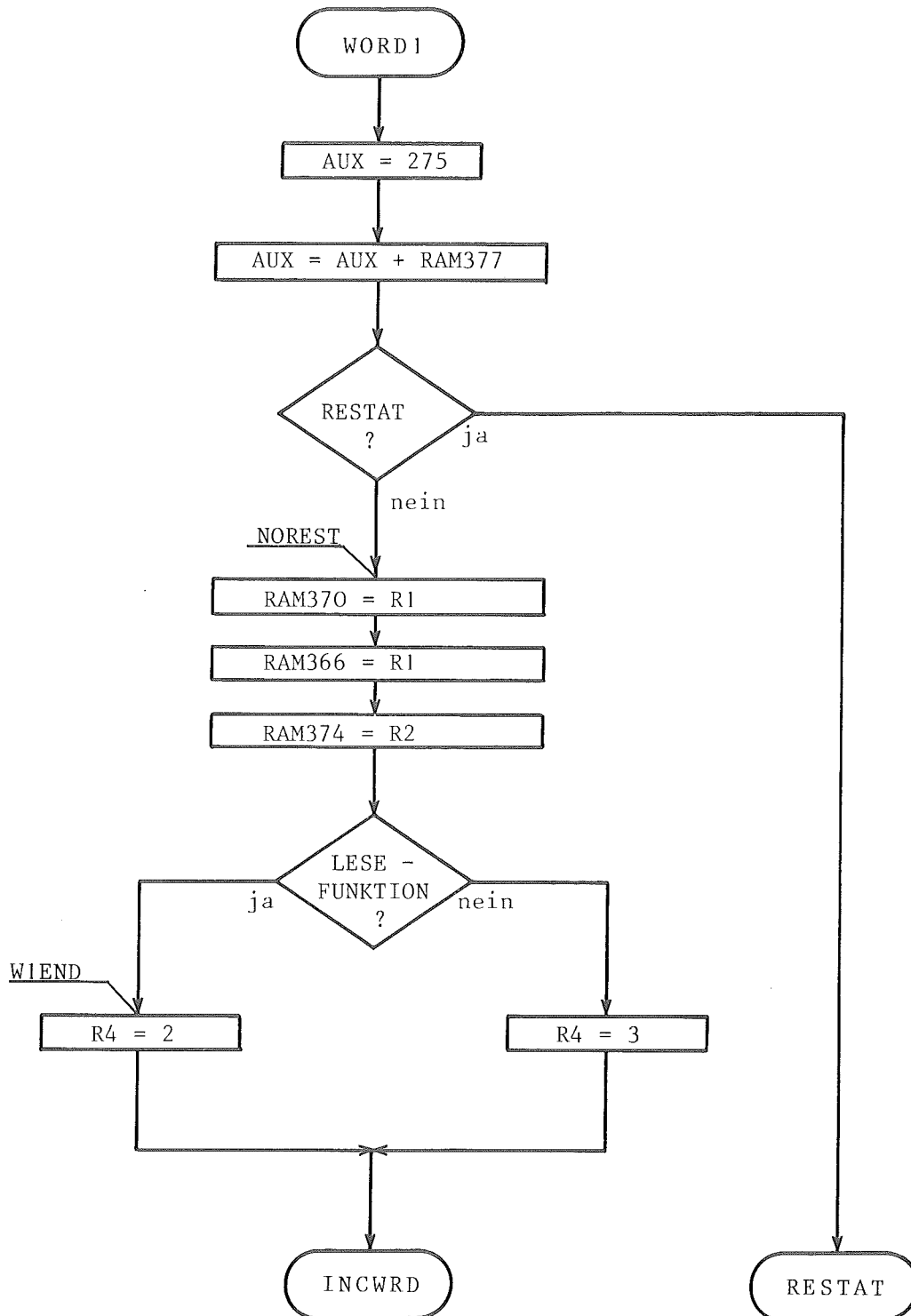


Abb. 10.2.6-1

116	0342		NOREST:	XMIT	IVR,370
	0342	147770			
117	0344			MOVE	R1,0,RIV7
	0344	000437			
118	0346			XMIT	IVR,366
	0346	147766			
119	0350			MOVE	R1,0,RIV7
	0350	000437			
120	0352			XMIT	IVR,374
	0352	147774			
121	0354			MOVE	R2,6,RIV7
	0354	001337			
122	0356			XMIT	IVR,377
	0356	147777			
123	0360			NZT	RIV3,1,WIEND
	0360	135473			
124	0362			XMIT	R4,3
	0362	142003			
125	0364			JMP	INCWRD
	0364	160343			
126	0366		WIEND:	XMIT	R4,2
	0366	142002			
127	0370			JMP	INCWRD
	0370	160343			
128	0372		WORD1:	XMIT	AUX,275
	0372	140275			
129	0374			XMIT	IVR,377
	0374	147777			
130	0376			ADD	RIV7,0,AUX
	0376	037400			
131	0400			NZT	AUX,NOREST
	0400	120161			
132	0402			JMP	RESTAT
	0402	160635			

Abb. 10.2.6-2

10.2.7 Programmsegment WORD2

Dieses Programmsegment wird nur dann angesprungen wenn die empfangene Nachricht eine CAMAC - Leseoperation beinhaltet und die für diese Leseoperation erforderlichen Angaben über Leseblocklänge und CAMAC - Stationsnummer abgespeichert werden. Dabei wird die zu lesende Blocklänge als 1er Komplement in der RAM - Speicherzelle 364 abgelegt und die CAMAC - Stationsnummer unter RAM - Zelle 373. Nach Setzen des Wortzählers R4 auf 3 und Überprüfung auf Nachrichtenende in Programmabschnitt INCWRD wird zum Programmsegment DATAIN zurückgesprungen.

In der Abbildung 10.2.7-1 ist für dieses Programmsegment das Flußdiagramm und das dazugehörige Programmlisting gezeigt.

133	0404	WORD2:	XMIT	IVR,364
	0404	147764		
134	0406		XMIT	AUX,377
	0406	140377		
135	0410		XOR	R1,0,RIV7
	0410	060437		
136	0412		XMIT	IVR,373
	0412	147773		
137	0414		MOVE	R2,6,RIV7
	0414	001337		
138	0416		XMIT	R4,3
	0416	142003		
139	0420		JMP	INCWRD
	0420	160343		

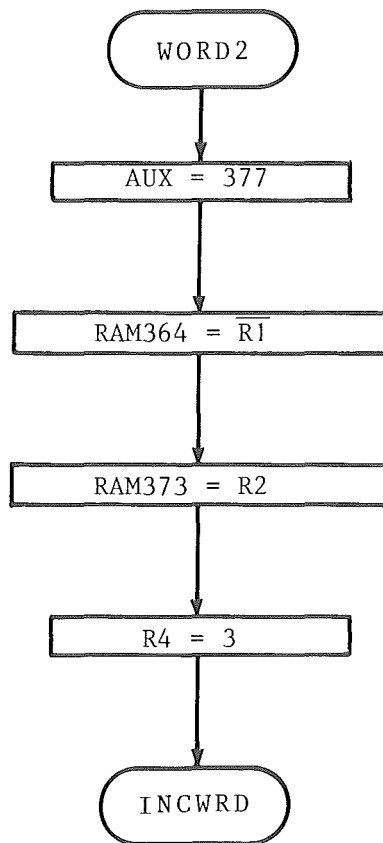


Abb. 10.2.7-1

10.2.8 Programmsegment WORD3

Im Programmsegment WORD3 werden die empfangenen CAMAC - Parameter A und F in die RAM - Zellen 374 und 372 abgelegt (siehe Abb. 10.2.8 - 2). In Abhängigkeit vom gewünschten Schreibformat (16bit und 24bit Datenbreite) wird der Wortzähler R4 auf den Wert 4 bzw. 5 gesetzt. Unter dem Statement FRMT24 wird der interne Requeststatus überprüft und im Falle eines angeforderten und bis zu diesem Zeitpunkt noch nicht abgesetzten CAMAC - Zyklus zum Timeout Statement TMEOUT gesprungen. Sind alle früher aufgesetzten internen Requests befriedigt worden, d.h. der Requeststatus ist nicht gesetzt, so kann die neue Stationsnummer N in das entsprechende Register übernommen werden. Unter dem Statement NOREAD werden die CAMAC - Parameter A und F in die entsprechenden Register eingeschrieben wobei bei Vorliegen einer Leseoperation unter dem Statement SETF16 die Binärstelle F16 des CAMAC - Parameters F gleich "1" gesetzt wird. Das Blocklängenregister in RAM - Zelle 370 wird um 1 dekrementiert und zum Datenaufnahmesegment DATAIN zurückgesprungen falls das Ende der Nachricht noch nicht erkannt wird. Ist der empfangene CAMAC - Parameter N für die Schreibfunktion identisch 0 so wird zum Statement RDYIN gesprungen, da kein CAMAC - Schreibzyklus abgesetzt werden soll. Ist N ungleich 0 so wird zum Statement READY gesprungen.

Der zum Statement READY gehörende Programmteil ist aufgrund der engen Verknüpfung mit dem Programmsegment WORD3 diesem zugeordnet. Das Flußdiagramm des Programmsegmentes WORD3 ist den Abbildungen 10.2.8-1 und Abbildung 10.2.8-3 (READY) zu entnehmen. Die Abbildungen 10.2.8-2 und 10.2.8-3 zeigen das Programmlisting für dieses Programmsegment.

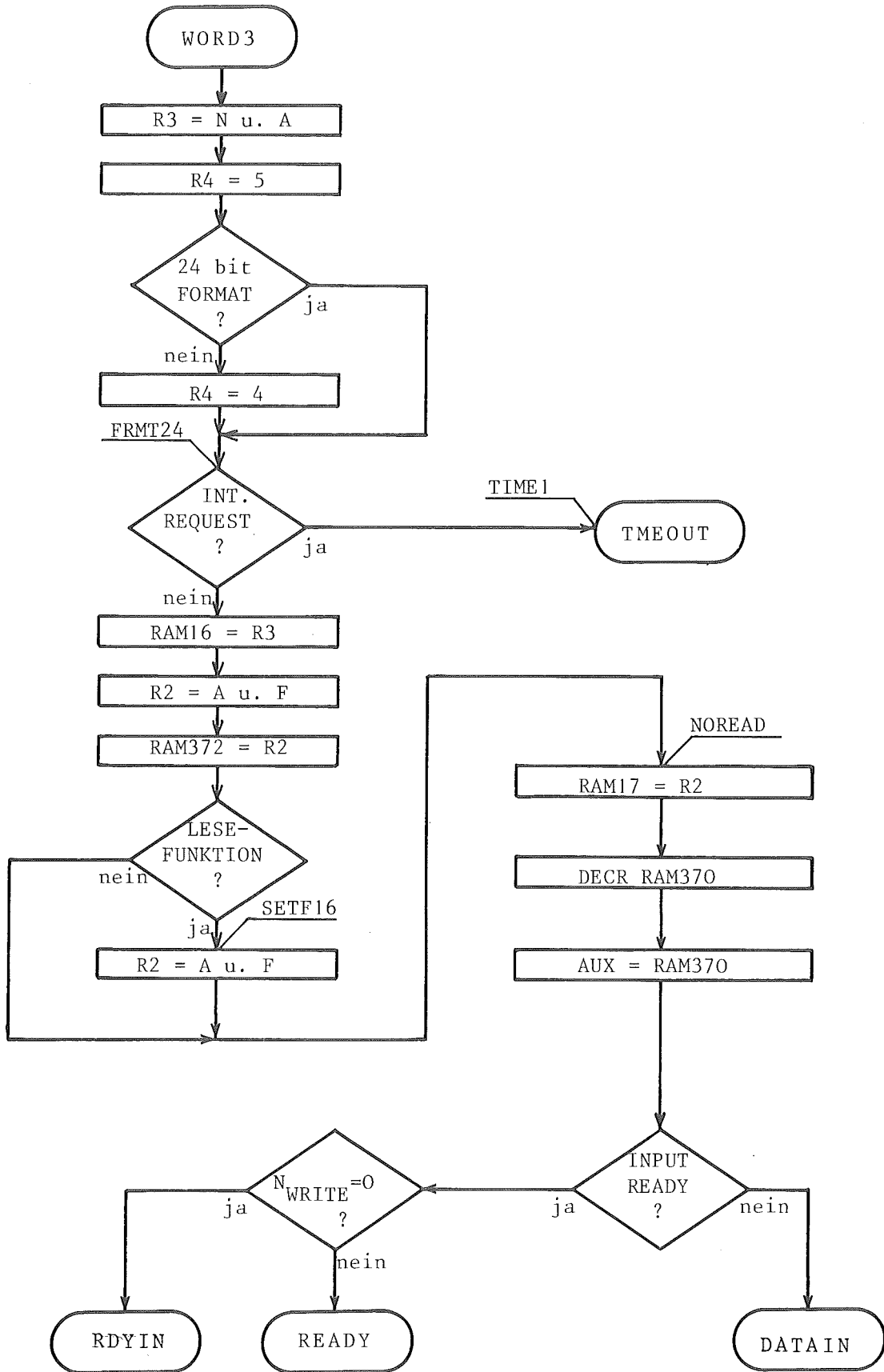


Abb. 10.2.8-1

140	0422	WORD3:	XMIT	IUR,374
	0422			147774
141	0424		MOVE	R1,1,RIV0
	0424			000470
142	0426		MOVE	RIV7,0,R3
	0426			017403
143	0430		XMIT	R4,5
	0430			142005
144	0432		NZY	RIV2,1,FRMT24
	0432			135057
145	0434		XMIT	R4,4
	0434			142004
146	0436	FRMT24:	NZY	LIV7,1,TIME1
	0436			133472
147	0440		XMIT	IUR,16
	0440			147416
148	0442		MOVE	R3,0,RIV7
	0442			001437
149	0444		XMIT	IUR,372
	0444			147772
150	0446		MOVE	R1,4,RIV7
	0446			000637
151	0450		MOVE	RIV6,3,RIV7
	0450			017177
152	0452		MOVE	R2,5,RIV4
	0452			001274
153	0454		MOVE	RIV7,0,R2
	0454			017402
154	0456		XMIT	IUR,377
	0456			147777
155	0460		NZY	RIV3,1,SETF16
	0460			135473
156	0462		JMP	NOREAD
	0462			160240
157	0464	TIME1:	JMP	TMEOUT
	0464			160632
158	0466	SETF16:	XMIT	IUR,373
	0466			147773
159	0470		MOVE	R1,1,RIV0
	0470			000470
160	0472		XMIT	AUX,177
	0472			140177
161	0474		AND	R2,0,AUX
	0474			041000
162	0476		ADD	R5,1,R2
	0476			022442
163	0500	NOREAD:	XMIT	IUR,17
	0500			147417
164	0502		MOVE	R2,0,RIV7
	0502			001037
165	0504		XMIT	IUR,370
	0504			147770
166	0506		XMIT	AUX,377
	0506			140377
167	0510		ADD	RIV7,0,AUX
	0510			037400

168	0512		MOVE	AUX,0,RIV7
	0512	000037		
169	0514		NZT	AUX,DATAIN
	0514	120054		
170	0516		XMIT	IUR,374
	0516	147774		
171	0520		NZT	RIV7,5,READY
	0520	137652		
172	0522		JMP	RDYIN
	0522	160350		
173	0524	READY:	XMIT	AUX,347
	0524	140347		
174	0526		ADD	RIV7,5,AUX
	0526	037640		
175	0530		NZT	OVF,CONTRL
	0530	124373		
176	0532		XMIT	AUX,40
	0532	140040		
177	0534		ADD	AUX,0,LIV7
	0534	020027		
178	0536		MOVE	AUX,0,AUX
	0536	000000		
179	0540		MOVE	AUX,0,AUX
	0540	000000		
180	0542		MOVE	AUX,0,LIV7
	0542	000027		
181	0544		JMP	RDYIN
	0544	160350		

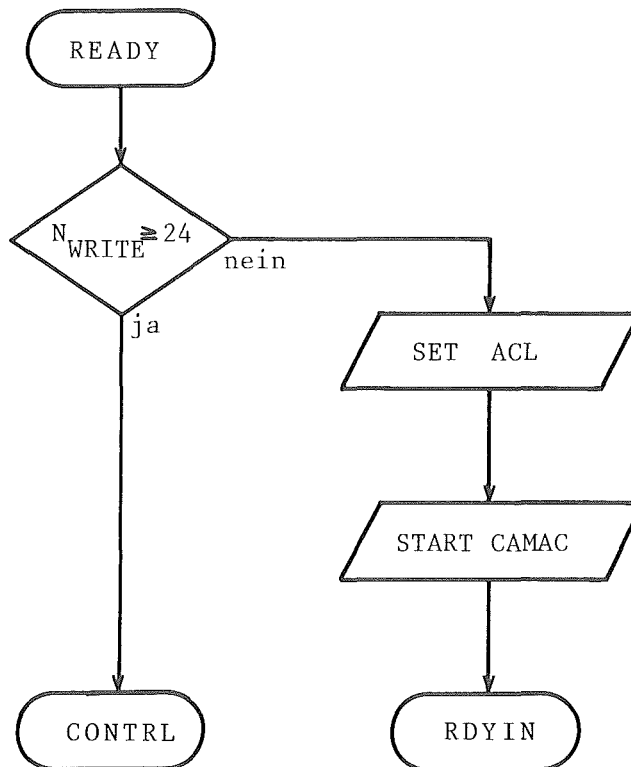


Abb. 10.2.8-3

10.2.9 Programmsegment WORDN

Dieses Programmsegment wird spätestens nach Einlesen des 5. empfangenen PDV - Wortes angesprungen und hat die Aufgabe die empfangenen Daten in die CAMAC - Schreibregister (Leitungen W1 bis W16) einzuschreiben. Bevor ein Überschreiben dieser Register durchgeführt wird, wird durch überprüfen des internen Requeststatus die Zulässigkeit der Registeränderung festgestellt. Ist der interne Request nicht zurückgesetzt so wird zum Programmsegment TMEOUT gesprungen. Eine Überprüfung auf gültig gesetztes Schreibfunktionsbit (6.Bit im Funktionenteil des 1. empfangenen Wortes) entscheidet ob lediglich das Blocklängenregister RAM 37o dekrementiert wird (Statement INCWRD) oder zum Programmstatement WRITE gesprungen wird. In Abhängigkeit vom gewünschten Datenformat (16bit bzw. 24bit breite Datenwörter) wurde in Programmstatement WORD3 der Wortzähler R4 entsprechend vorgesetzt. Werden 24bit breite Datenwörter angezeigt so wird dem Wortzähler R4 alternierend der Wert 5 bzw. 6 zugewiesen, wodurch die unterschiedliche Behandlung der Lower Bytes und des High Bytes ermöglicht wird. Unter dem Programmstatement WRITE wird die Nachricht auf Wiederholungsfunktion geprüft und im Falle der angezeigten Wiederholungsfunktion zum Programmstatement PRUEF abgesprungen. Im weiteren Programmablauf wird im Falle des Erkennens des zuletzt übertragenen Datenwortes und der gesetzten CAMAC - Parameter $N = 30$ und $A1 = 0$ zum Programmstatement NIST30 gesprungen. Ist dies nicht der Fall so wird der Programmablauf ab Statement CYCLE fortgeführt.

Die Abbildung 10.2.9-1 zeigt das Flußdiagramm des Programmsegmentes WORDN und die Abbildung 10.2.9-2 das Programmlisting.

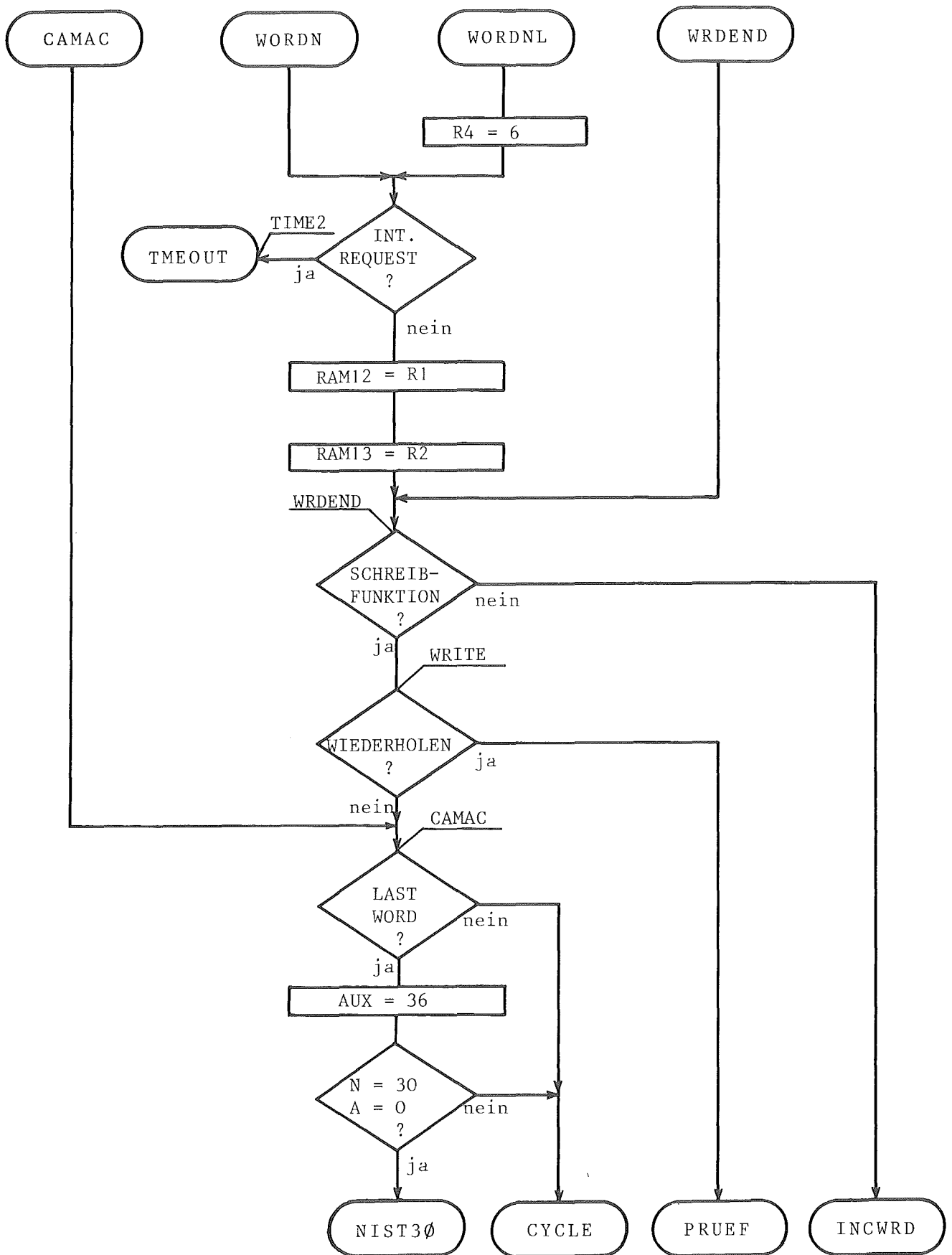


Abb. 10.2.9-1

207	0630	WORDNL:	XMIT	R4,6
	0630	142006		
208	0632	WORDN:	NZT	LIV7,1,TIME2
	0632	133465		
209	0634		XMIT	IVR,12
	0634	147412		
210	0636		MOVE	R1,0,RIV7
	0636	000437		
211	0640		XMIT	IVR,13
	0640	147413		
212	0642		MOVE	R2,0,RIV7
	0642	001037		
213	0644	WRDEND:	XMIT	IVR,377
	0644	147777		
214	0646		NZT	RIV2,1,WRITE
	0646	135066		
215	0650		JMP	INCWRD
	0650	160343		
216	0652	TIME2:	JMP	TMEOUT
	0652	160632		
217	0654	WRITE:	NZT	RIV1,1,PRUEF
	0654	134441		
218	0656	CAMAC:	XMIT	IVR,370
	0656	147770		
219	0660		NZT	RIV6,7,CYCLE
	0660	137376		
220	0662		XMIT	IVR,374
	0662	147774		
221	0664		XMIT	AUX,36
	0664	140036		
222	0666		XOR	RIV7,0,AUX
	0666	077400		
223	0670		NZT	AUX,CYCLE
	0670	120336		
224	0672		JMP	NIST30
	0672	160510		

Abb. 10.2.9-2

10.2.10 Programmsegment PRUEF

In diesem Programmabschnitt wird im Falle einer Nachricht mit Wiederholungsfunktion (7. Bit des Funktionenbytes des 1. übertragenen Wortes = 1) die laufende Wortnummer mit jener verglichen, die bei der vorangegangenen Nachricht beim Auftreten des Übertragungsfehlers als noch gültig abgespeichert wurde. Ist die laufende Wortnummer größer als die abgespeicherte, so wird zum Statement CAMAC gesprungen und die folgende Ausführung eingeleitet. Ist die laufende Wortnummer kleiner gleich der abgespeicherten so wird unter den Statement INCWRD der Blocklängenzähler dekrementiert und zur Datenaufnahme zurückgesprungen.

Die Abbildung 10.2.10-1 zeigt das Flußdiagramm und das Programmlisting für dieses Programmsegment.

196	0602	PRUEF:	XMIT	AUX,377
	0602	140377		
197	0604		XMIT	IVR,370
	0604	147770		
198	0606		XOR	RIV7,0,AUX
	0606	077400		
199	0610		XMIT	IVR,366
	0610	147766		
200	0612		ADD	RIV7,0,AUX
	0612	037400		
201	0614		XMIT	IVR,375
	0614	147775		
202	0616		XOR	RIV7,0,AUX
	0616	077400		
203	0620		NZT	AUX,INCWRD
	0620	120343		
204	0622		XMIT	IVR,377
	0622	147777		
205	0624		XMIT	RIV1,1,0
	0624	154440		
206	0626		JMP	CAMAC
	0626	160327		

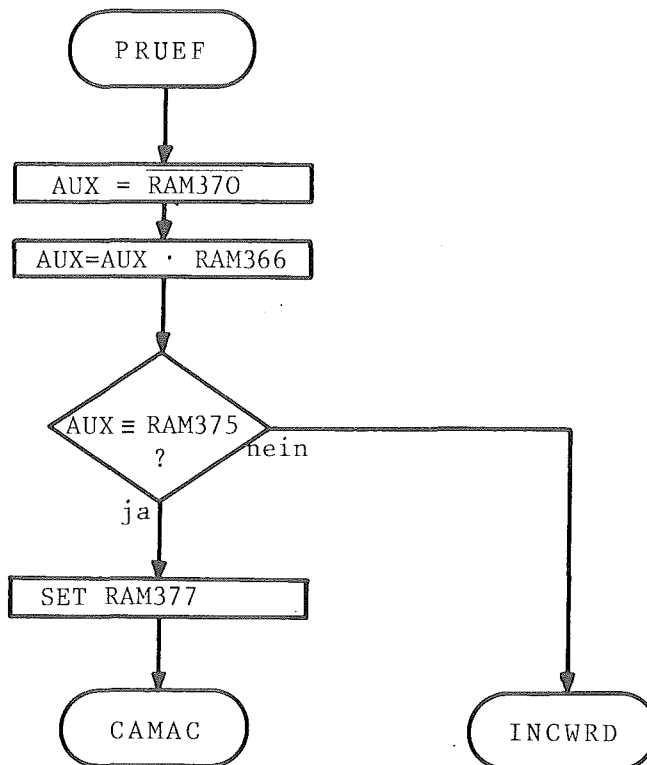


Abb. 10.2.10-1

10.2.11 Programmsegment CYCLE

Unter diesem Programmsegment wird in Abhängigkeit vom Inhalt des Wortzählers R4 ein CAMAC - Zyklus ausgelöst. Zum Auslösen eines CAMAC - Zyklus wird zuerst das ACL Signal aufgesetzt und nach Ablauf von $1,2 \mu\text{s}$ der interne Request gesetzt. Sollen 24bit breite Daten geschrieben werden und wurden beim vorangegangenen empfangenen PDV - Wort die Lowerbytes empfangen, so wird kein CAMAC - Zyklus ausgelöst ($R4 = 6$) und der Ablauf des Programms bei dem Statement INCWRD fortgeführt. Durch Dekrementieren und Abfragen des Blocklängenzählers wird das Ende der Nachricht erkannt, der Rücksprung zum Dateneinlesesegment DATAIN unterdrückt und das Programm mit dem Statement RDYIN fortgesetzt. Soll keine Leseoperation ausgeführt werden, so wird auf Grund der folgenden logischen Abfrage zum Programmabschnitt der Antwortbildung (OUTPUT) gesprungen. Sollen Daten gelesen werden, so wird ab dem Programmstatement LESEN das 1er Komplement der Blocklänge der Antwort in Register 4 geladen und überprüft ob die gewünschte Blocklänge der Antwort kleiner gleich 110 ist. Ist dies der Fall wird im RAM - Statusregister (RAM - Zelle 367) das zweite Bit, welches die Angabe eines zu langen Antwortblockes kennzeichnet, gesetzt und über das Programmsegment GOBACK zur Datenaufnahme im Programmsegment MAINLP zurückgesprungen. Bei dem Erkennen einer noch zulässigen Antwortblocklänge wird das Programmsegment OUTPUT aktiviert. In diesem Programmsegment wird die Antwort zusammengestellt sowie CAMAC - Daten ausgelesen.

Das Flußdiagramm dieses Programmsegmentes ist in Abbildung 10.2.11-1 zu sehen. Abbildung 10.2.11-2 zeigt das dazugehörige Programmlisting.

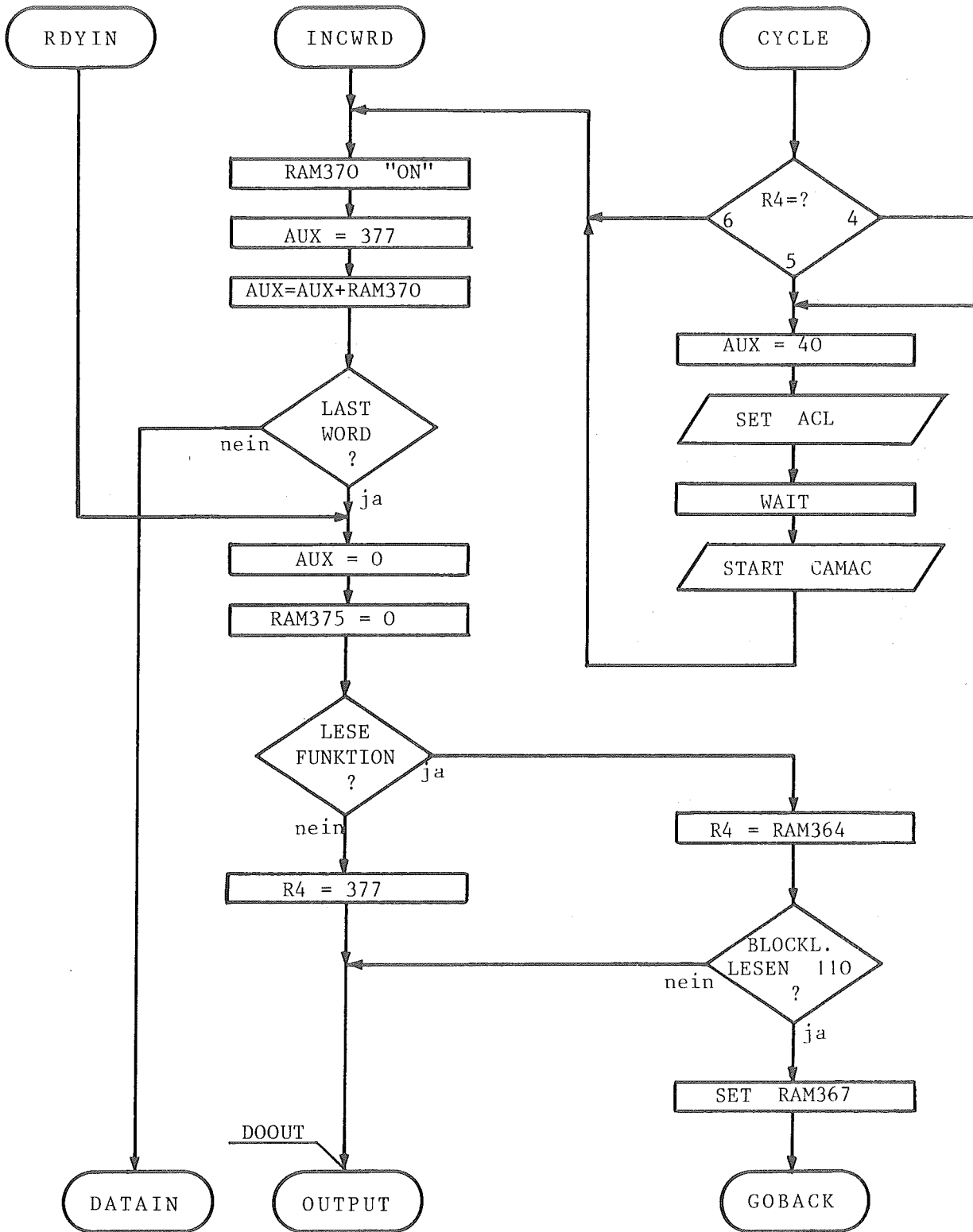


Abb. 10.2.11-1

225	0674		CYCLE:	XEC	R4, TABLE4
	0674	102363			
226	0676			ADD	AUX, 0, LIV7
	0676	020027			
227	0700			MOVE	AUX, 0, AUX
	0700	000000			
228	0702			MOVE	AUX, 0, AUX
	0702	000000			
229	0704			MOVE	AUX, 0, LIV7
	0704	000027			
230	0706		INCWRD:	XMIT	IUR, 370
	0706	147770			
231	0710			XMIT	AUX, 377
	0710	140377			
232	0712			ADD	RIV7, 0, AUX
	0712	037400			
233	0714			MOVE	AUX, 0, RIV7
	0714	000037			
234	0716			NZT	AUX, DATAIN
	0716	120054			
235	0720		RDYIN:	XMIT	AUX, 0
	0720	140000			
236	0722			XMIT	IUR, 375
	0722	147775			
237	0724			MOVE	AUX, 0, RIV7
	0724	000037			
238	0726			XMIT	IUR, 377
	0726	147777			
239	0730			NZT	RIV3, 1, LESEN
	0730	135457			
240	0732			XMIT	R4, 377
	0732	142377			
241	0734		DOOUT:	JMP	OUTPUT
	0734	161415			
242	0736		LESEN:	XMIT	IUR, 364
	0736	147764			
243	0740			MOVE	RIV7, 0, R4
	0740	017404			
244	0742			XMIT	AUX, 157
	0742	140157			
245	0744			ADD	R4, 0, AUX
	0744	022000			
246	0746		TABLE4:	NZT	OVF, DOOUT
	0746	124356			
247	0750			XMIT	IUR, 367
	0750	147767			
248	0752			XMIT	RIV6, 1, 1
	0752	157041			
249	0754			JMP	GOBACK
	0754	160272			
250	0756			XMIT	AUX, 40
	0756	140040			
251	0760			XMIT	AUX, 40
	0760	140040			
252	0762			JMP	INCWRD
	0762	160343			

10.2.12 Programmsegment LOESCH

In einigen Fällen ist es notwendig, die über die SDS - Schnittstelle empfangenen Daten nicht weiter auszuwerten und ohne Antwortabgabe für die nächste Nachrichtenaufnahme zum Programmsegment MAINLP zurückzuspringen. Im vorliegenden Programmabschnitt LOESCH werden die empfangenen Daten gelöscht und im RAM - Statusregister 367 das 3. Bit (Kennzeichnung einer gelöschten Nachricht) gesetzt. Zum Löschen wird in einer Warteschleife, bei welcher das Empfangsrahmensignal RS überprüft wird, das Ende der laufenden Sendung abgewartet. Anschließend wird durch Dekrementierung des FIFO's (Stack - Transfer) und Überprüfung des \overline{ORE} - Signales (Output Register Empty) das FIFO leergeäumt. Das Löschen empfangener Daten wird in folgenden Fällen vorgenommen:

- a) empfangene Übertragungssequenz hat das Kennzeichen einer Antwort
- b) CRC - Fehler
- c) Adresse der empfangenen Nachricht ist nicht identisch mit eigener Adresse oder Sammeladresse
- d) Timeout Überprüfung hat angesprochen

Für das Programmsegment LOESCH kann der Abbildung 10.2.12-1 das Flußdiagramm entnommen werden während Abbildung 10.2.12-2 das Programmlisting zeigt.

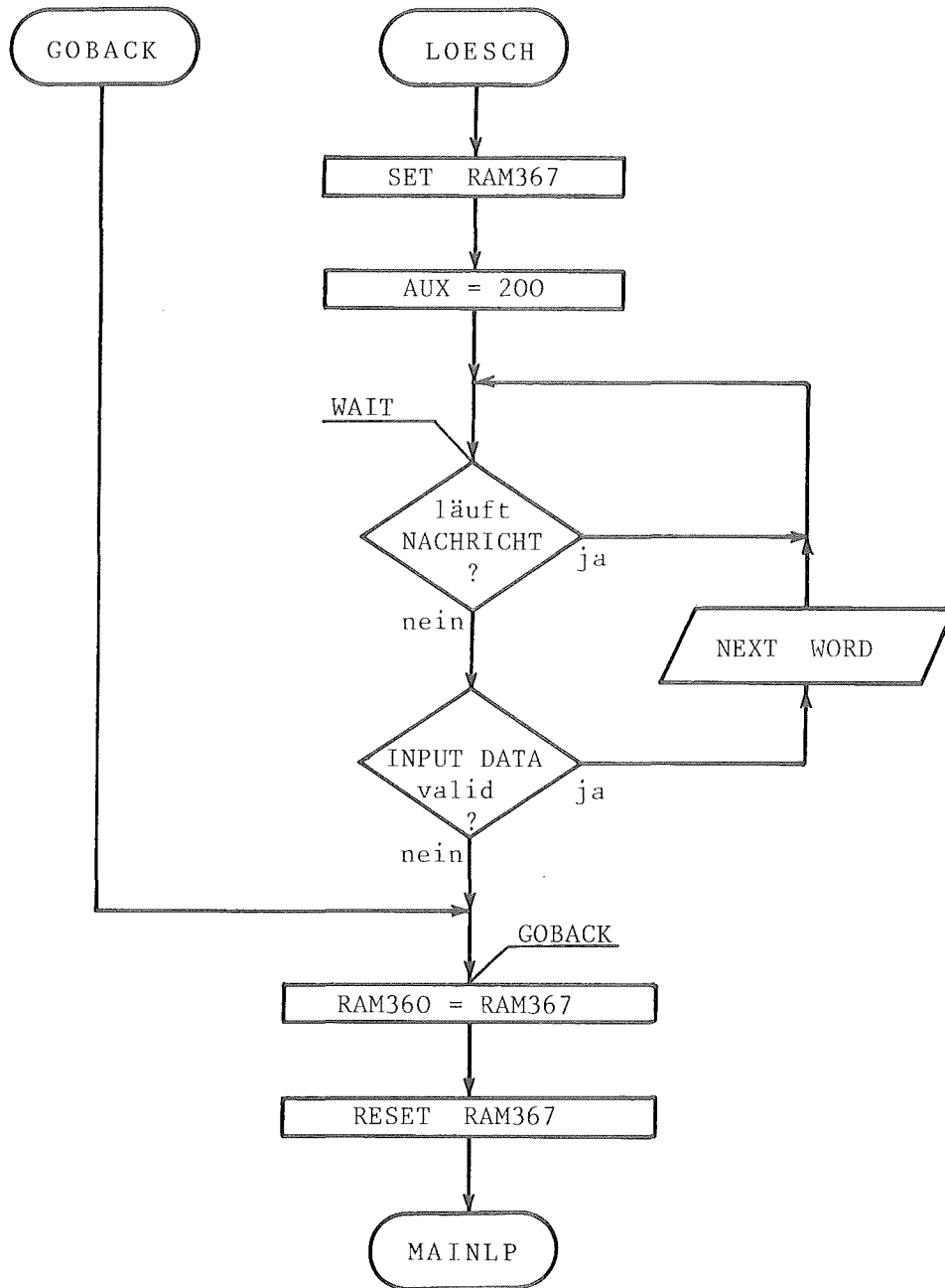


Abb. 10.2.12-1

182	0546	LOESCH:	XMIT	IUR,367
	0546	147767		
183	0550		XMIT	RIV5,1,1
	0550	156441		
184	0552		XMIT	AUX,200
	0552	140200		
185	0554	WAIT:	NZT	LIV1,1,WAIT
	0554	130466		
186	0556		NZT	LIV2,1,GOBACK
	0556	131072		
187	0560		MOVE	AUX,0,LIV7
	0560	000027		
188	0562		JMP	WAIT
	0562	160266		
189	0564	GOBACK:	XMIT	IUR,367
	0564	147767		
190	0566		MOVE	RIV7,0,R11
	0566	017411		
191	0570		XMIT	RIV7,5,0
	0570	157640		
192	0572		XMIT	RIV2,1,0
	0572	155040		
193	0574		XMIT	IUR,360
	0574	147760		
194	0576		MOVE	R11,0,RIV7
	0576	004437		
195	0600		JMP	MAINLF
	0600	160043		

Abb. 10.2.12-2

10.2.13 Programmsegment CRTFKT

Dieses Programmsegment wird aktiviert wenn bei einer gültig empfangenen Nachricht erkannt wird, daß es sich um ein Einzel - PDV - Wort handelt. In diesem Falle kann es sich nur um einen globalen Crate-Controllerbefehl (PDV - Funktion) handeln. Die dabei gewünschte Funktion wird durch die Codierung der 4 höchstwertigsten Bits des 2. übertragenen Bytes angegeben. Dieses für die Funktion relevante Bitpattern wird in das Register R3 übertragen und nach der Softwaretabelle TABLE2 zum entsprechenden Programmsegment verzweigt. Zum Auslösen der Funktion "Z" (initialisieren) und "C" (löschen) wird unter dem Programmstatement DOFKT der interne Requeststatus abgefragt, das ACL - Signal gesetzt, das zu den vorgenannten Funktionen entsprechende Bitpattern auf das IV - Byte aufgebracht und zum Antwortsegment ANSWER gesprungen.

Das für dieses Programmsegment entworfene Flußdiagramm zeigt Abbildung 10.2.13-1. Das Programmlisting hierfür ist den Abbildungen 10.2.13-2 und 10.2.13-3 zu entnehmen.

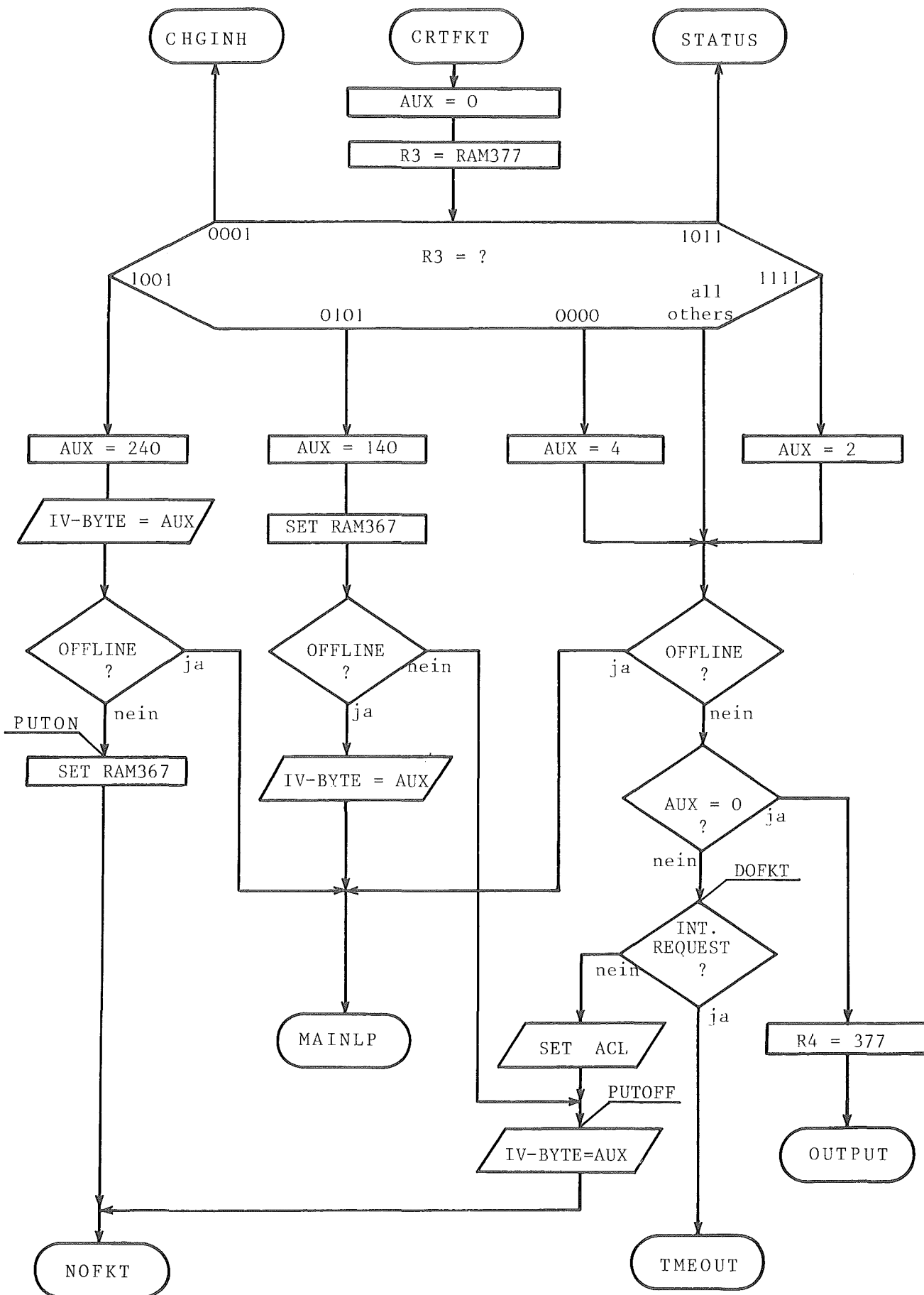


Abb. 10.2.13-1

364	1322		CRTFKT:	XMIT	AUX,0
	1322	140000			
365	1324			MOVE	RIV3,4,R3
	1324	015603			
366	1326			XMIT	IVR,367
	1326	147767			
367	1330			XEC	R3, TABLE2
	1330	101562			
368	1332			NZT	RIV0,1,OFF
	1332	134061			
369	1334			NZT	AUX,DOFKT
	1334	120202			
370	1336			XMIT	R4,377
	1336	142377			
371	1340			JMP	OUTPUT
	1340	161415			
372	1342		OFF:	JMP	MAINLF
	1342	160043			
373	1344		TABLE2:	XMIT	AUX,4
	1344	140004			
374	1346			MOVE	AUX,0,AUX
	1346	000000			
375	1350			MOVE	AUX,0,AUX
	1350	000000			
376	1352			MOVE	AUX,0,AUX
	1352	000000			
377	1354			MOVE	AUX,0,AUX
	1354	000000			
378	1356			MOVE	AUX,0,AUX
	1356	000000			
379	1360			MOVE	AUX,0,AUX
	1360	000000			
380	1362			MOVE	AUX,0,AUX
	1362	000000			
381	1364			JMP	CHGINH
	1364	160624			
382	1366			JMP	CRTON
	1366	160616			
383	1370			JMP	CRTOFF
	1370	160611			
384	1372			MOVE	AUX,0,AUX
	1372	000000			
385	1374			MOVE	AUX,0,AUX
	1374	000000			
386	1376			JMP	STATUS
	1376	161551			
387	1400			MOVE	AUX,0,AUX
	1400	000000			
388	1402			XMIT	AUX,2
	1402	140002			

389	1404		DOFKT:	NZT	LIV7,1,TMEOUT
	1404	133472			
390	1406			XMIT	R3,100
	1406	141500			
391	1410			MOVE	R3,0,LIV7
	1410	001427			
392	1412			MOVE	AUX,0,AUX
	1412	000000			
393	1414		PUTOFF:	MOVE	AUX,0,AUX
	1414	000000			
394	1416			MOVE	AUX,0,LIV7
	1416	000027			
395	1420			JMP	NOFKT
	1420	160655			
396	1422		CRTOFF:	XMIT	AUX,140
	1422	140140			
397	1424			XMIT	RIV0,1,1
	1424	154041			
398	1426			NZT	LIV4,1,PUTOFF
	1426	132046			
399	1430			MOVE	AUX,0,LIV7
	1430	000027			
400	1432			JMP	MAINLP
	1432	160043			
401	1434		CRTON:	XMIT	AUX,240
	1434	140240			
402	1436			MOVE	AUX,0,LIV7
	1436	000027			
403	1440			NZT	LIV4,1,PUTON
	1440	132062			
404	1442		OFF1:	JMP	MAINLP
	1442	160043			
405	1444		PUTON:	XMIT	RIV0,1,0
	1444	154040			
406	1446			JMP	NOFKT
	1446	160655			
413	1464		TMEOUT:	XMIT	IVR,367
	1464	147767			
414	1466			XMIT	RIV2,1,1
	1466	155041			
415	1470			JMP	LOESCH
	1470	160263			

Abb. 10.2.13-3 (Fortsetzung von 10.2.13-2)

10.2.14 Programmsegment NIST30

Wie unter Kapitel 4.1 beschrieben kann auf Grund globaler CAMAC - Funktionen das Statusregister geändert werden. Dabei ist in Abhängigkeit vom Funktionenteil des CAMAC - Befehles ein Überschreiben, selectives Setzen oder selectives Löschen des Statusregisters möglich. Die dafür vorgesehenen CAMAC - Funktionen:

N30 A0 F17

N30 A0 F19

N30 A0 F23

sind jedoch keine datenlosen CAMAC - Befehle, da die Information zur Änderung des Statusregisters mit übertragen werden muß. Der vorliegende Programmabschnitt NIST30 hat die Aufgabe diese Statusmodifikationen vorzunehmen. Bei Beginn dieses Programmsegments ist bereits das Vorliegen der Stationsnummer N30 festgestellt worden. Auf Grund der folgenden Abfragen auf A0 F17, A0 F19 und A0 F23 wird zu den Programmsegmenten NOTWRS und NSLSET verzweigt und die entsprechende Statusänderung vorgenommen. Wird keine Übereinstimmung mit diesen 3 Unteradress- und Funktionsteilen festgestellt wird dieses Programmsegment mit einem Sprung nach NOFKT verlassen. Nach erfolgter Statusänderung, die bis jetzt lediglich in der RAM - Zelle 367 vorgenommen wurde, muß diese auf den Hardwarezustand des Controllers übertragen werden. Dabei kann jedoch der ONLINE Zustand nur dann erhalten bleiben wenn auf Grund der manuellen Bedienung der ONLINE Zustand zulässig ist. Im anderen Fall kann trotz Vorgabe des selectiven Setzens des ONLINE Zustandes dieser nicht erreicht werden. Unter dem Statement CRATE erfolgt ein Testen der Statusinhibit - Information und ein entsprechendes Verzweigen zum Programmabschnitt INHON bzw. INHOFF. Das Flußdiagramm und das Programmlisting des Programmsegmentes NIST30 sind den Abbildungen 10.2.14-1 und 10.2.14-2 zu entnehmen.

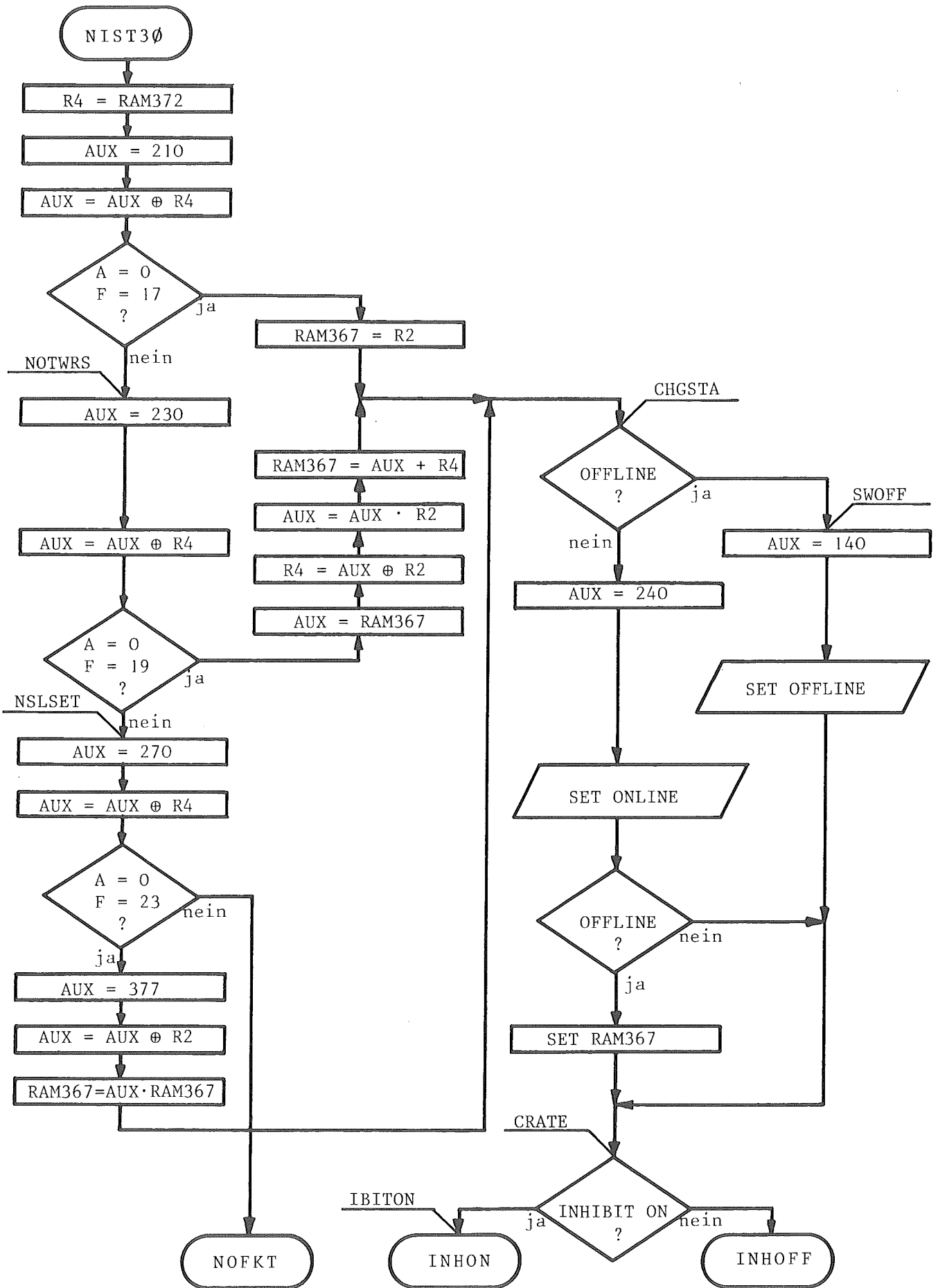


Abb. 10.2.14-1

331	1220	NIST30:	XMIT	IUR,372
	1220	147772		
332	1222		MOVE	RIV7,0,R4
	1222	017404		
333	1224		XMIT	IUR,367
	1224	147767		
334	1226		XMIT	AUX,210
	1226	140210		
335	1230		XOR	R4,0,AUX
	1230	062000		
336	1232		NZT	AUX,NOTWRS
	1232	120141		
337	1234		MOVE	R2,0,RIV7
	1234	001037		
338	1236		JMP	CHGSTA
	1236	160526		
339	1240	NSLSET:	XMIT	AUX,270
	1240	140270		
340	1242		XOR	R4,0,AUX
	1242	062000		
341	1244		NZT	AUX,NOFKT
	1244	120255		
342	1246		XMIT	AUX,377
	1246	140377		
343	1250		XOR	R2,0,AUX
	1250	061000		
344	1252		AND	RIV7,0,RIV7
	1252	057437		
345	1254	CHGSTA:	NZT	RIV0,1,SWOFF
	1254	134075		
346	1256		XMIT	AUX,240
	1256	140240		
347	1260		MOVE	AUX,0,LIV7
	1260	000027		
348	1262		NZT	LIV4,1,CRATE
	1262	132077		
349	1264		XMIT	RIV0,1,1
	1264	154041		
350	1266		JMP	CRATE
	1266	160537		
351	1270	IBITON:	JMP	INHON
	1270	160461		
352	1272	SWOFF:	XMIT	AUX,140
	1272	140140		
353	1274		MOVE	AUX,0,LIV7
	1274	000027		
354	1276	CRATE:	NZT	RIV1,1,IBITON
	1276	134474		
355	1300		JMP	INHOFF
	1300	160474		
356	1302	NOTWRS:	XMIT	AUX,230
	1302	140230		
357	1304		XOR	R4,0,AUX
	1304	062000		
358	1306		NZT	AUX,NSLSET
	1306	120120		
359	1310		MOVE	RIV7,0,AUX
	1310	017400		
360	1312		XOR	R2,0,R4
	1312	061004		
361	1314		AND	R2,0,AUX
	1314	041000		
362	1316		ADD	R4,0,RIV7
	1316	022037		
363	1320		JMP	CHGSTA
	1320	160526		

10.2.15 Programmsegment CONTRL

In diesem Programmsegment werden vom Controller alle datenlosen CAMAC - Befehle ausgeführt die nicht an spezielle Module mit entsprechender Stationsnummer weitergeleitet werden sollen sondern die eine globale Crate- bzw. Controllerfunktion bewerkstelligen. Solche Funktionen sind:

N28	F26	A8	initialisieren "Z"
N28	F26	A9	löschen "C"
N30	A9	F24	Inhibit OFF
N30	A9	F26	Inhibit ON
N30	A9	F27	Test Inhibit
N30	A0	F1	Read Status
N30	A12	F1	Read LAM - Pattern

In Abb. 10.2.15 - 1 ist in Form einer Baumstruktur die Aufteilung dieser CAMAC - Funktionen zu den entsprechenden Programmabschnitten gezeigt. Der Vollständigkeit halber sind in dieser Baumstruktur auch die Funktionen zur Modifikation des Status aufgeführt die bereits in Kapitel 10.2.13 beschrieben worden sind. Da das gesamte Programmsegment CONTRL 76 Einzelinstruktionen umfaßt, ist sowohl die Befehlsliste als auch das Flußdiagramm in 3 Einzelabschnitte aufgeteilt. Im ersten Abschnitt findet das Auswählen aller Funktionen mit N ungleich 30 (also für die Funktionen "Z" und "C") sowie der Funktion READ Status statt. Zu Beginn dieses Programmabschnittes und somit zu Beginn des gesamten Programmsegmentes CONTRL wird die CAMAC - Unteradresse A auf das Vorhandensein der Wertigkeit 2 (A2) überprüft. Da in keinem der zulässigen globalen CAMAC - Funktionen die Wertigkeit 2 der Unteradresse existent ist können solche Funktionen ausgeschieden werden und eine weitere Überprüfung der CAMAC - Parameter kann entfallen. Die restlichen 8 CAMAC - Einzelparameter A1, A4, A8, F1, F2, F4, F8 und F16 sind im Register R2 zusammengefaßt und können mit einem vorgegebenen Zahlenwert verglichen werden. Nach Überprüfung auf die CAMAC - Stationsnummer N gleich 28 und der Parameter A gleich 8 und F gleich 26 bzw. A gleich 9 und F gleich 26 werden nach Sprung zum Programmstatement DOFKT die globalen CAMAC - Funktionen "Z" und "C" ausgelöst. Wird als Stationsnummer N = 30 festgestellt so wird entsprechend den nachfolgenden Abfragen die entsprechenden Instruktionen ausgeführt.

Aufgrund der umfangreichen Struktur dieses Programmsegmentes ist das Flußdiagramm auf die Abbildungen 10.2.15-1, 10.2.15-3 und 10.2.15-5 verteilt. Das Programmlisting ist den Abbildungen 10.2.15-2, 10.2.15-4 und 10.2.15-6 zu entnehmen.

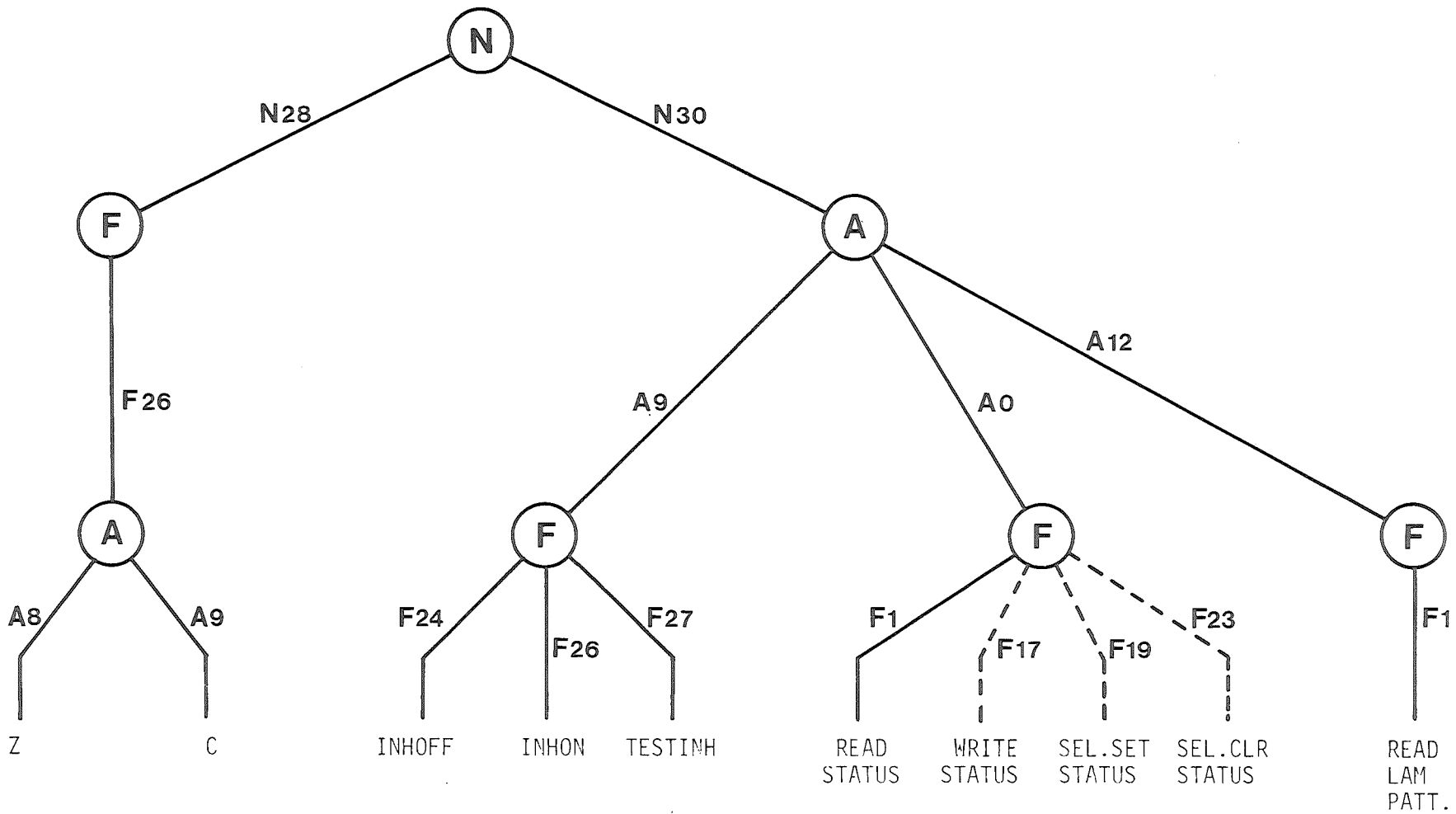


Abb. 10.2.15 - 1

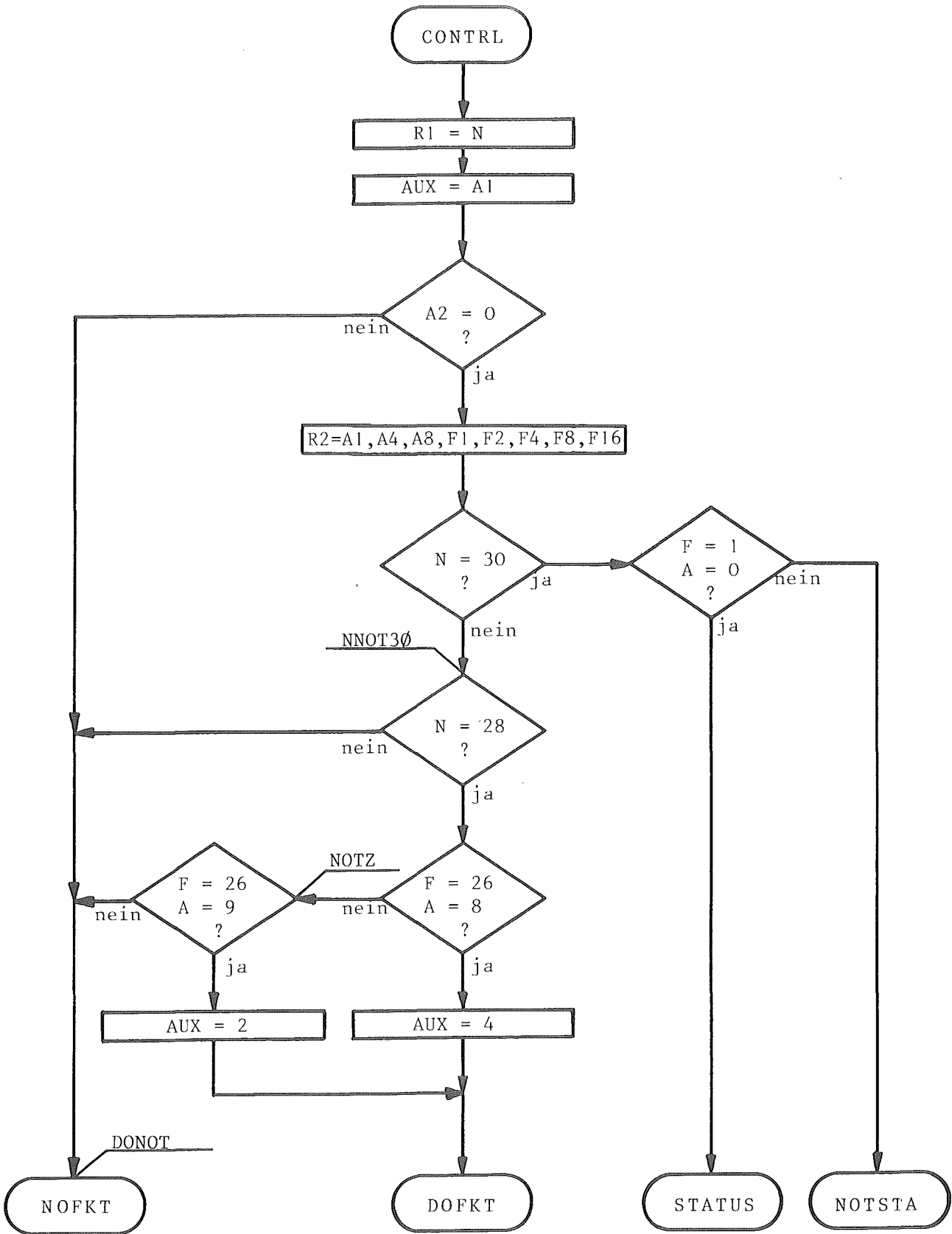


Abb. 10.2.15-1

253	0764		DONOT:	JMP	NOFKT
	0764	160655			
254	0766		CONTRL:	MOVE	RIV7,5,R1
	0766	017641			
255	0770			MOVE	RIV0,1,AUX
	0770	014040			
256	0772			XMIT	IVR,372
	0772	147772			
257	0774			NZT	RIV7,1,DONOT
	0774	137472			
258	0776			MOVE	RIV6,7,R2
	0776	017342			
259	1000			ADD	R2,7,R2
	1000	021342			
260	1002			XMIT	AUX,36
	1002	140036			
261	1004			XMIT	IVR,367
	1004	147767			
262	1006			XOR	R1,0,AUX
	1006	060400			
263	1010			NZT	AUX,NNOT30
	1010	120011			
264	1012			XMIT	AUX,10
	1012	140010			
265	1014			XOR	R2,0,AUX
	1014	061000			
266	1016			NZT	AUX,NOTSTA
	1016	120026			
267	1020			JMP	STATUS
	1020	161551			
268	1022		NNOT30:	XMIT	AUX,34
	1022	140034			
269	1024			XOR	R1,0,AUX
	1024	060400			
270	1026			NZT	AUX,NOFKT
	1026	120255			
271	1030			XMIT	AUX,324
	1030	140324			
272	1032			XOR	R2,0,AUX
	1032	061000			
273	1034			NZT	AUX,NOTZ
	1034	120021			
274	1036			XMIT	AUX,4
	1036	140004			
275	1040			JMP	DOFKT
	1040	160602			
276	1042		NOTZ:	XMIT	AUX,325
	1042	140325			
277	1044			XOR	R2,0,AUX
	1044	061000			
278	1046			NZT	AUX,NOFKT
	1046	120255			
279	1050			XMIT	AUX,2
	1050	140002			
280	1052			JMP	DOFKT
	1052	160602			

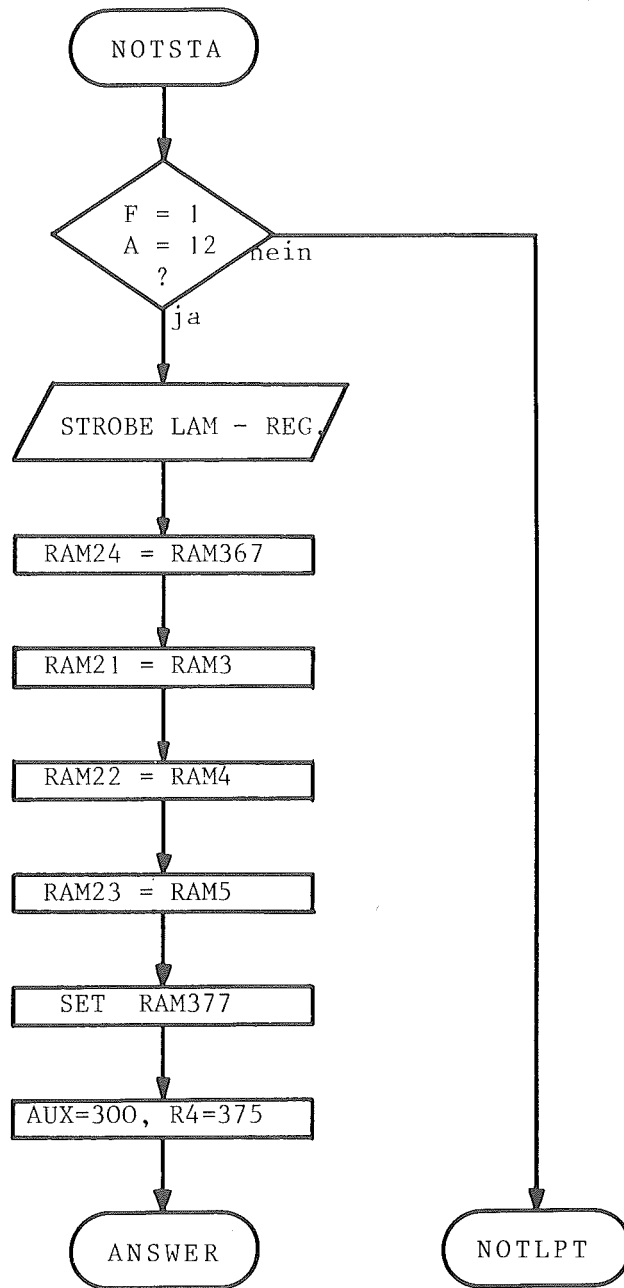


Abb. 10.2.15-3

281	1054	NOTSTA:	XMIT	AUX,16
	1054	140016		
282	1056		XOR	R2,0,AUX
	1056	061000		
283	1060		NZT	AUX,NOTLPT
	1060	120056		
284	1062		MOVE	AUX,0,LIV7
	1062	000027		
285	1064		MOVE	RIV7,0,R1
	1064	017401		
286	1066		XMIT	IUR,24
	1066	147424		
287	1070		MOVE	R1,0,RIV7
	1070	000437		
288	1072		XMIT	IUR,3
	1072	147403		
289	1074		MOVE	RIV7,0,R1
	1074	017401		
290	1076		XMIT	IUR,21
	1076	147421		
291	1100		MOVE	R1,0,RIV7
	1100	000437		
292	1102		XMIT	IUR,4
	1102	147404		
293	1104		MOVE	RIV7,0,R1
	1104	017401		
294	1106		XMIT	IUR,22
	1106	147422		
295	1110		MOVE	R1,0,RIV7
	1110	000437		
296	1112		XMIT	IUR,5
	1112	147405		
297	1114		MOVE	RIV7,0,R1
	1114	017401		
298	1116		XMIT	IUR,23
	1116	147423		
299	1120		MOVE	R1,0,RIV7
	1120	000437		
300	1122		XMIT	IUR,377
	1122	147777		
301	1124		XMIT	RIV1,1,1
	1124	154441		
302	1126		XMIT	R4,375
	1126	142375		
303	1130		XMIT	AUX,300
	1130	140300		
304	1132		JMP	ANSWER
	1132	161420		

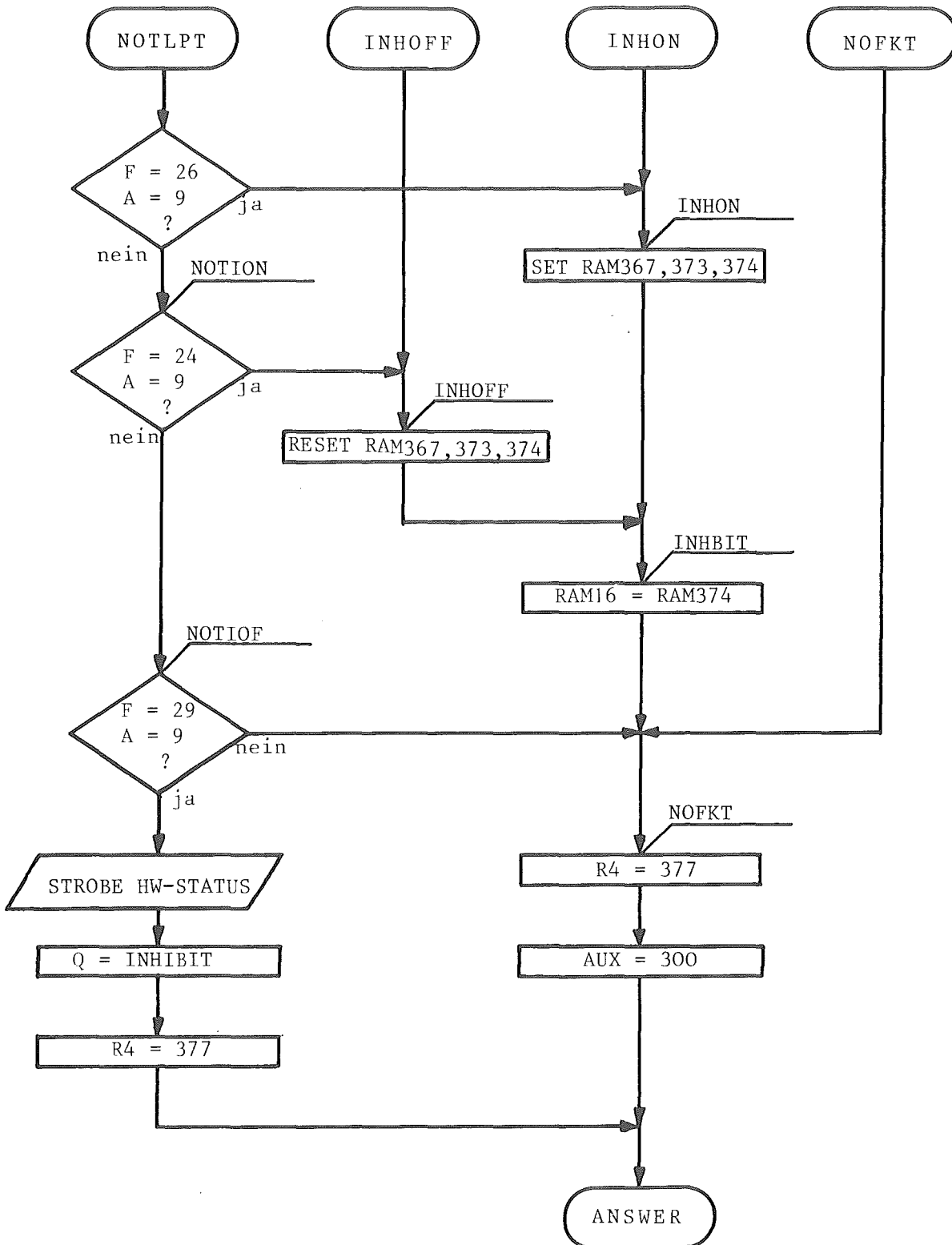


Abb. 10.2.15-5

305	1134		NOTLEFT: XMIT	AUX,325
	1134	140325		
306	1136		XOR	R2,0,AUX
	1136	061000		
307	1140		NZT	AUX,NOTION
	1140	120071		
308	1142		INHON: XMIT	AUX,10
	1142	140010		
309	1144		MOVE	AUX,0,LIV7
	1144	000027		
310	1146		INHBIT: XOR	AUX,0,LIV7
	1146	060027		
311	1150		XMIT	IVR,6
	1150	147406		
312	1152		MOVE	RIV5,1,AUX
	1152	016440		
313	1154		XMIT	IVR,367
	1154	147767		
314	1156		MOVE	AUX,1,RIV1
	1156	000071		
315	1160		JMP	NOFKT
	1160	160655		
316	1162		NOTION: XMIT	AUX,305
	1162	140305		
317	1164		XOR	R2,0,AUX
	1164	061000		
318	1166		NZT	AUX,NOTIOF
	1166	120077		
319	1170		INHOFF: XMIT	AUX,340
	1170	140340		
320	1172		MOVE	AUX,0,LIV7
	1172	000027		
321	1174		JMP	INHBIT
	1174	160463		
322	1176		NOTIOF: XMIT	AUX,335
	1176	140335		
323	1200		XOR	R2,0,AUX
	1200	061000		
324	1202		NZT	AUX,NOFKT
	1202	120255		
325	1204		XOR	AUX,0,LIV7
	1204	060027		
326	1206		XMIT	IVR,6
	1206	147406		
327	1210		MOVE	RIV5,1,AUX
	1210	016440		
328	1212		MOVE	AUX,1,AUX
	1212	000040		
329	1214		XMIT	R4,377
	1214	142377		
330	1216		JMP	ANSWER
	1216	161420		

10.2.16 Programmsegment CHGINH

Um den Zustand des Inhibit Signales auch ohne CAMAC - Funktion zu ändern, kann eine globale PDV - Funktion (vergleiche Kapitel 4.1) benutzt werden. Da in PDV - Bericht 70 lediglich zwei Reservefunktionen offen sind und eine bereits für die CAMAC - Löschfunktion eingesetzt wurde kann zur Inhibit Änderung nur die letzte verbleibende Reservefunktion eingesetzt werden. Um trotzdem das Inhibit ein- bzw. ausschalten zu können wurde diese Funktion als Inhibit Änderungsfunktion benutzt. Um das Inhibit gezielt in einen Zustand zu versetzen ist es daher erforderlich den Ist - Zustand des Inhibits zuvor abzufragen. Dies kann durch Statusabfrage ebenfalls mit Hilfe einer PDV - Funktion geschehen. Im Programmsegment CHGINH wird der eingestellte Inhibit Zustand abgetestet und in Anhängigkeit vom erkannten Zustand zu den Programmstatements INHON bzw. INHOFF gesprungen, unter welchen der gewünschte Inhibit Zustand eingeschaltet wird.

Das Programmlisting sowie das Flußdiagramm dieses Programmsegmentes ist der Abbildung 10.2.16-1 zu entnehmen.

407	1450	CHGINH:	NZT	RIV0,1,OFF1
	1450			134061
408	1452		XOR	AUX,0,LIV7
	1452			060027
409	1454		XMIT	IUR,6
	1454			147406
410	1456		NZT	RIV5,1,DOIOFF
	1456			136471
411	1460		JMP	INHON
	1460			160461
412	1462	DOIOFF:	JMP	INHOFF
	1462			160474

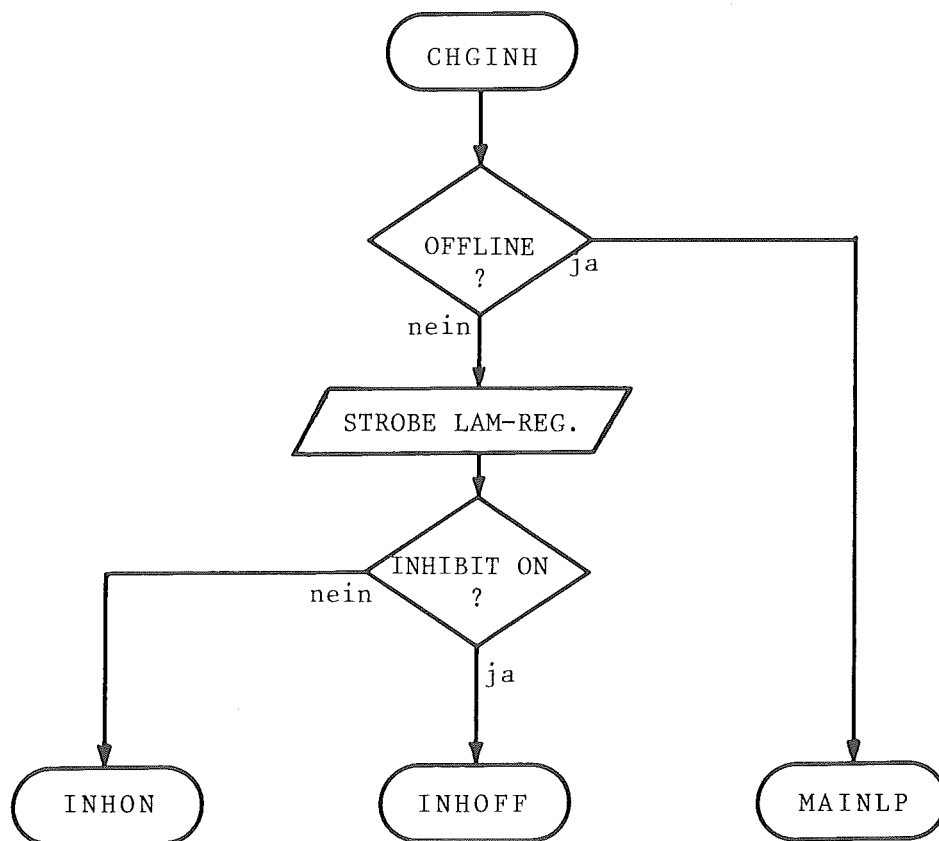


Abb. 10.2.16-1

10.2.17 Programmsegment SAMMEL

Dieses Programmsegment wird immer dann aktiviert wenn die aufgenommene Nachricht die im PDV - Bericht 70 festgelegte Sammeladresse hat. Im vorliegenden Controllerprogramm sind 4 mögliche und ausführbare Sammeladressierungen vorgesehen:

- a) Beginn zyklische Kurzabfrage
- b) Ende zyklische Kurzabfrage
- c) Initialisieren
- d) Löschen

Das Programmsegment SAMMEL kann also von 3 möglichen Sammeladressierungen angesprochen werden (einer Nachricht mit Sammeladresse zum Beenden der zyklischen Kurzabfrage muß immer eine Nachricht zum Beginn zyklische Kurzabfrage vorausgegangen sein). Zu Beginn dieses Programmabschnittes wird der PDV - Funktionenteil zuerst auf die Funktion " Beginn zyklische Kurzabfrage " abgefragt. Ist dies der Fall so wird zum Programmstatement KURZAB verzweigt. Im anderen Fall wird die codierte PDV - Funktion auf Codierung zum Absetzen der Initialisierungsfunktion "Z" bzw. LösCHFunktion "C" überprüft. Wird eine Übereinstimmung festgestellt so wird durch Aufbringen des entsprechenden Bitpatterns auf das IV - Byte die gewünschte Funktion ausgelöst (Programmstatement DOC). Wird keine gültige Funktion unter der Sammeladresse erkannt so wird zum Beginn der Datenaufnahme zurückgesprungen. Die Abbildung 10.2.17-1 zeigt das Flußdiagramm und Abbildung 10.2.17-2 das programmlisting.

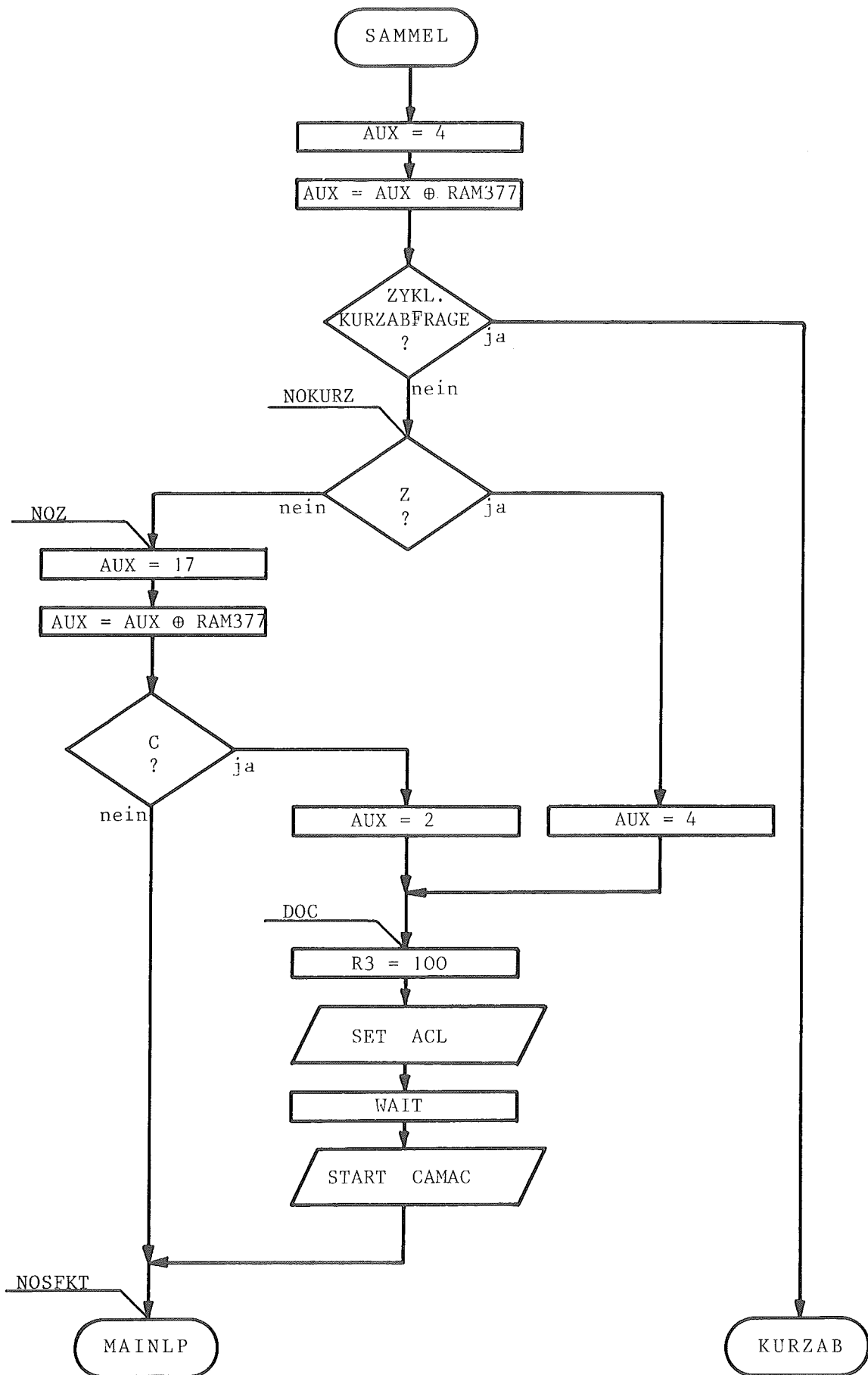


Abb. 10.2.17-1

443	1560	SAMMEL:	XMIT	IUR,377
	1560	147777		
444	1562		XMIT	AUX,4
	1562	140004		
445	1564		XOR	RIV3,4,AUX
	1564	075600		
446	1566		NZT	AUX,NOKURZ
	1566	120304		
447	1570		JMP	KURZAB
	1570	160721		
455	1610	NOKURZ:	NZT	RIV3,4,NOZ
	1610	135614		
456	1612		XMIT	AUX,4
	1612	140004		
457	1614	DOC:	XMIT	R3,100
	1614	141500		
458	1616		MOVE	R3,0,LIV7
	1616	001427		
459	1620		MOVE	AUX,0,AUX
	1620	000000		
460	1622		MOVE	AUX,0,AUX
	1622	000000		
461	1624		MOVE	AUX,0,LIV7
	1624	000027		
462	1626	NOSFKT:	JMP	MAINLP
	1626	160043		
463	1630	NOZ:	XMIT	AUX,17
	1630	140017		
464	1632		XOR	RIV3,4,AUX
	1632	075600		
465	1634		NZT	AUX,NOSFKT
	1634	120313		
466	1636		XMIT	AUX,2
	1636	140002		
467	1640		JMP	DOC
	1640	160706		

10.2.18 Programmsegment KURZAB

In diesem Programmabschnitt wird die verkürzte zyklische Alarmabfrage behandelt. Alle erkannten PDV - Bus - Nachrichten werden als Einzelbytes mit dem Inhalt einer Stationsadresse gewertet und mit der eigenen Adresse verglichen. Bei Übereinstimmung wird durch Auslesen der LAM - Register und Bildung des Summen - LAM's ein Alarmwunsch festgestellt. In Anhängigkeit von diesem wird das Ausgabe-FIFO mit der eigenen Adresse (kein Summen-LAM vorhanden) bzw. mit der Alarmadresse geladen. Nach dem Starten der Antwort wird zur Datenaufnahme unter dem Statement KURZAB zurückgesprungen. Wird keine Übereinstimmung der empfangenen Adresse mit der eigenen festgestellt so überprüft eine weitere Abfrage ob die empfangene Adresse identisch mit der Sammeladresse ist. Ist dies nicht der Fall, so wird ebenfalls zur Datenaufnahme unter dem Statement KURZAB zurückgesprungen. Ist die empfangene Adresse die Sammeladresse so werden die nächsten zwei empfangenen Bytes gelesen und mit der Sammeladresse als PDV - Einzelwort gewertet. Das zuletzt empfangene Byte wird als CRC gewertet und eine entsprechende Überprüfung auf Richtigkeit vorgenommen. Fällt die CRC - Überprüfung positiv aus, wird die Funktion des empfangenen PDV - Wortes daraufhin untersucht, ob es sich um die Funktion " Ende zyklische Kurzabfrage " handelt. Ist dies der Fall, so wird zum Programmstatement MAINLP zurückgesprungen und der normale Programmablauf eingeleitet. Fällt die CRC - Überprüfung negativ aus bzw. handelt es sich nicht um den Befehl " Ende zyklische Kurzabfrage " so verbleibt das Programm innerhalb der zyklischen Kurzabfrage.

Das für dieses Programmsegment entwickelte Flußdiagramm zeigt Abbildung 10.2.18-1. Die Abbildung 10.2.18-2 gibt das entsprechende Programmlisting wieder. Unter dem Statement NOADR ist in der Abbildung 10.2.18-3 der weitere Verlauf des Flußdiagramms zu sehen und in Abbildung 10.2.18-4 das dazugehörige Programmlisting.

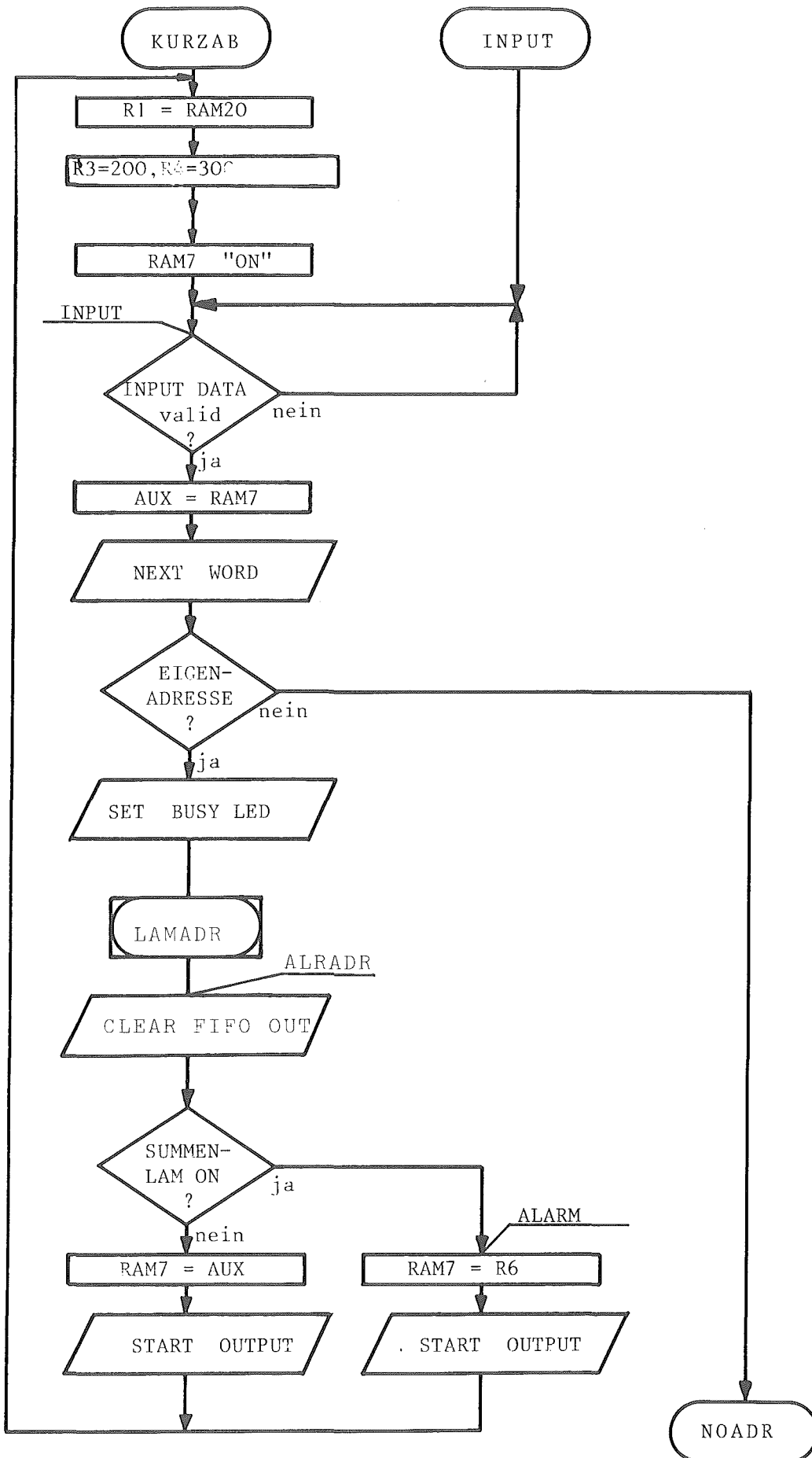


Abb. 10.2.18-1

448	1572	ALRADR:	XMIT	IVR,10
	1572	147410		
449	1574		XMIT	R11,20
	1574	144420		
450	1576		MOVE	R11,0,LIV7
	1576	004427		
451	1600		NZT	R3,ALARM
	1600	121736		
452	1602		MOVE	R2,0,RIV7
	1602	001037		
453	1604		MOVE	R4,0,LIV7
	1604	002027		
454	1606		JMP	KURZAB
	1606	160721		
468	1642	KURZAB:	XMIT	IVR,20
	1642	147420		
469	1644		MOVE	RIV7,0,R2
	1644	017402		
470	1646		XMIT	R3,200
	1646	141600		
471	1650		XMIT	R4,300
	1650	142300		
472	1652		XMIT	IVR,7
	1652	147407		
473	1654	INPUT:	NZT	LIV2,1,INPUT
	1654	131066		
474	1656		MOVE	RIV7,0,AUX
	1656	017400		
475	1660		MOVE	R3,0,LIV7
	1660	001427		
476	1662		XOR	R2,0,R11
	1662	061011		
477	1664		NZT	R11,NOADR
	1664	124742		
478	1666		MOVE	R5,0,LIV7
	1666	002427		
479	1670		XMIT	R11,2
	1670	144402		
480	1672		JMP	LAMADR
	1672	161565		
481	1674	ALARM:	XMIT	R6,376
	1674	143376		
482	1676		MOVE	R6,0,RIV7
	1676	003037		
483	1700		MOVE	R4,0,LIV7
	1700	002027		
484	1702		JMP	KURZAB
	1702	160721		

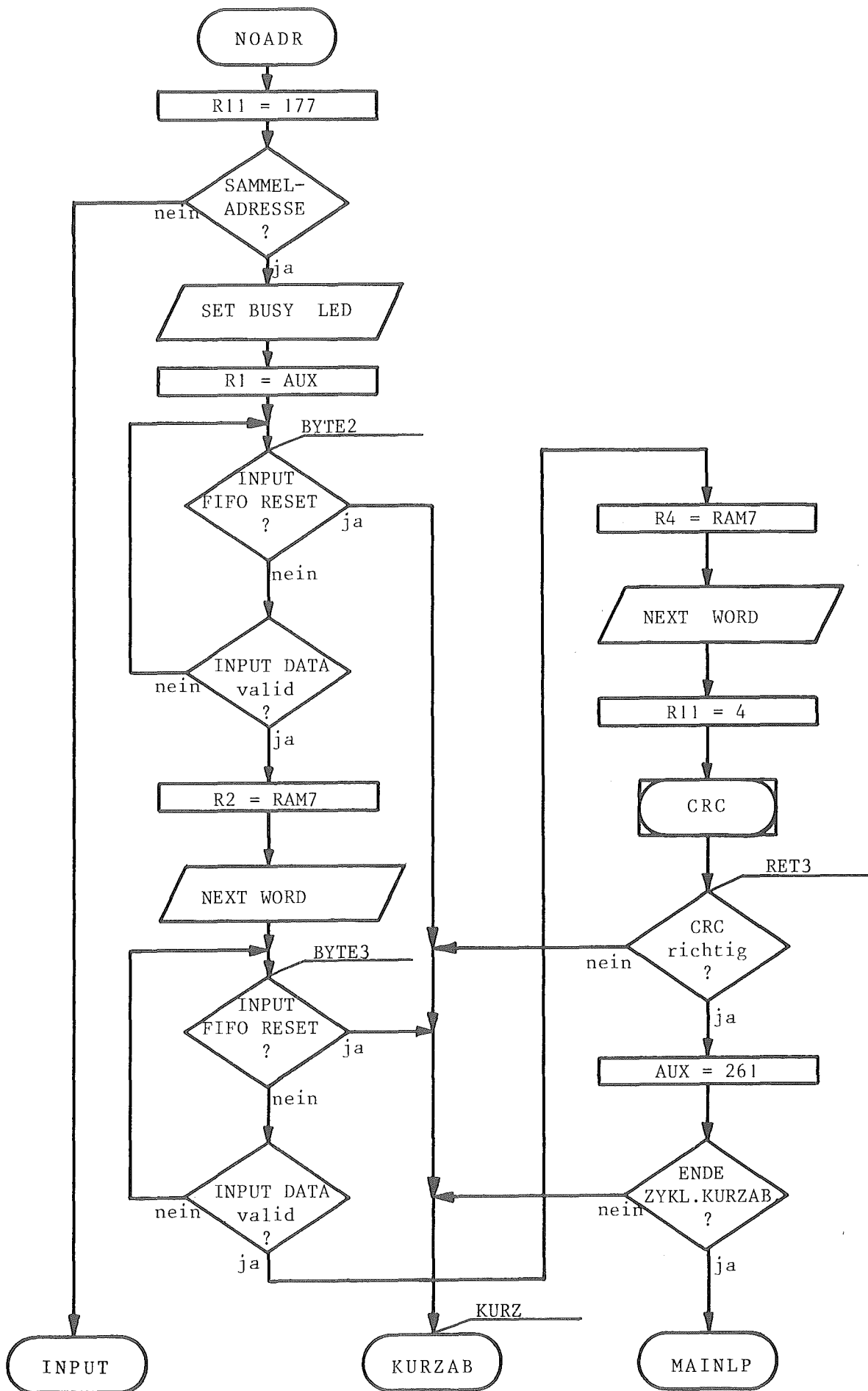


Abb. 10.2.18-3

485	1704		NOADR:	XMIT	R11,177
	1704	144577			
486	1706			XOR	R11,0,R11
	1706	064411			
487	1710			NZT	R11,INPUT
	1710	124726			
488	1712			MOVE	R5,0,LIV7
	1712	002427			
489	1714			MOVE	AUX,0,R1
	1714	000001			
490	1716		BYTE2:	NZT	LIV6,1,KURZ
	1716	133061			
491	1720			NZT	LIV2,1,BYTE2
	1720	131047			
492	1722			MOVE	RIV7,0,R2
	1722	017402			
493	1724			MOVE	R3,0,LIV7
	1724	001427			
494	1726		BYTE3:	NZT	LIV6,1,KURZ
	1726	133061			
495	1730			NZT	LIV2,1,BYTE3
	1730	131053			
496	1732			MOVE	RIV7,0,R4
	1732	017404			
497	1734			MOVE	R3,0,LIV7
	1734	001427			
498	1736			XMIT	R11,4
	1736	144404			
499	1740			JMP	CRC
	1740	160774			
500	1742		KURZ:	JMP	KURZAB
	1742	160721			
501	1744		RET3:	XOR	R4,0,AUX
	1744	062000			
502	1746			NZT	AUX,KURZAB
	1746	120321			
503	1750			XMIT	AUX,261
	1750	140261			
504	1752			XOR	R2,0,AUX
	1752	061000			
505	1754			NZT	AUX,KURZAB
	1754	120321			
506	1756			JMP	MAINLF
	1756	160043			

10.2.19 Programmsegment STATUS

Dieses Programmsegment organisiert die Datenstruktur zum Auslesen des Status. Eine Statusmeldung besteht aus zwei PDV - Wörtern wobei das 1. PDV - Wort mit Crate Adress- und Kurzstatusteil identisch mit dem 1. PDV - Wort aller möglichen Antworten ist. Das 2. PDV - Wort enthält einen 16bit Status der aus einzelnen Zustandsmeldungen und Alarmsignalen (LAM - Signale) besteht. Unter Kapitel 6.1 ist die genaue Bedeutung aller Einzelbitstellen der Statusinformation beschrieben. Das 2. Statusbyte entspricht dabei dem in RAM - Zelle 367 abgelegten Softwarestatus. Dabei ist zu beachten, daß der angezeigte Status zur vorangegangenen PDV - Nachricht gehört. Lediglich die Statusbits für ONLINE und INHIBIT entsprechen dem augenblicklichen Stand. Das 4. Bit dieses Software- Statusbyte zeigt bei Zustand logisch "1" einen CRC - Fehler an und entscheidet damit über die Funktion des 1. Bytes der Statusinformation. Wurde ein CRC - Fehler erkannt (4. Bit gleich 1) so entspricht der Inhalt des 1. Bytes der binärcodierten Wortnummer bis zu welcher die PDV - Nachricht korrekt übertragen wurde. Wird keine CRC - Fehlermeldung abgegeben, so entsprechen die 8 Bits des 1. Statusbytes einer LAM - Meldung. In den ersten 5 Bits dieses Bytes wird die binärcodierte LAM - Nummer des höchstpriorären LAM's wiedergegeben während die drei höchstwertigsten Bits durch Jumper ausgewählte Einzel - LAM - Meldungen sind. Im Programmsegment STATUS wird nach dem Zusammenstellen des zweiten Software - Statusbytes und der Überprüfung auf CRC - Fehlermeldung ab dem Statement LAMADR das LAM - Pattern gelesen und die binärcodierte LAM - Nummer des höchstpriorären LAM s gebildet. Da das Feststellen einer LAM - Meldung in mehreren Programmabschnitten benötigt wird, ist diese Routine als Unterprogramm aufgebaut. Der Inhalt des Registers R11 gibt das Rücksprungziel aus dieser Unterprogrammroutine an. Im Register R3 steht dabei die binärcodierte höchstprioräre LAM - Nummer. Dabei werden jedoch nur solche LAM - Signale angezeigt die noch nicht durch eine rückübertragene Statusmeldung im Controller - Zwischenspeicher gelöscht wurden. Das Flußdiagramm des Programmsegmentes STATUS ist in zwei Abschnitte unterteilt. Abbildung 10.2.19-1 zeigt den ersten Abschnitt und Abbildung 10.2.19-3 den zweiten Teil. Die dazugehörenden Programmlistingabschnitte sind den Abbildungen 10.2.19-2 und 10.2.19-4 zu entnehmen.

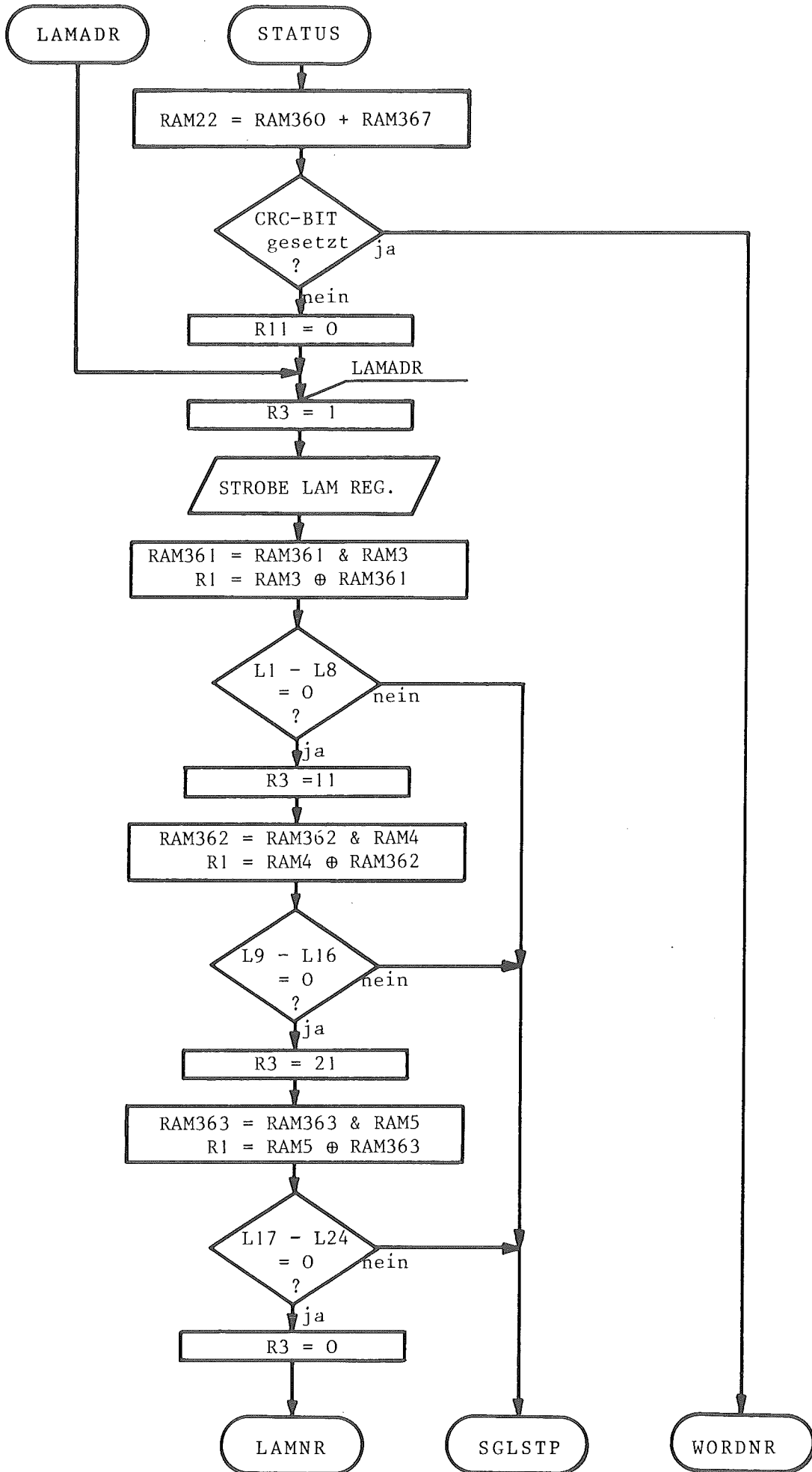


Abb. 10.2.19-1

878	3322		STATUS: MOVE	RIV0,1,AUX
	3322	014040		
879	3324		XMIT	IVR,360
	3324	147760		
880	3326		MOVE	RIV7,6,R1
	3326	017701		
881	3330		XOR	AUX,0,LIV7
	3330	060027		
882	3332		XMIT	IVR,6
	3332	147406		
883	3334		MOVE	RIV5,1,R3
	3334	016443		
884	3336		XMIT	IVR,22
	3336	147422		
885	3340		MOVE	R1,6,RIV7
	3340	000737		
886	3342		MOVE	R3,1,RIV1
	3342	001471		
887	3344		MOVE	AUX,1,RIV0
	3344	000070		
888	3346		NZT	RIV4,1,WORDNR
	3346	136042		
889	3350		XMIT	R11,0
	3350	144400		
890	3352		LAMADR: XMIT	R3,1
	3352	141401		
891	3354		XOR	AUX,0,LIV7
	3354	060027		
892	3356		XMIT	IVR,3
	3356	147403		
893	3360		MOVE	RIV7,0,AUX
	3360	017400		
894	3362		XMIT	IVR,361
	3362	147761		
895	3364		AND	RIV7,0,RIV7
	3364	057437		
896	3366		XOR	RIV7,0,R1
	3366	077401		
897	3370		NZT	R1,SGLSTP
	3370	120622		
898	3372		XMIT	R3,11
	3372	141411		
899	3374		XMIT	IVR,4
	3374	147404		
900	3376		MOVE	RIV7,0,AUX
	3376	017400		
901	3400		XMIT	IVR,362
	3400	147762		
902	3402		AND	RIV7,0,RIV7
	3402	057437		
903	3404		XOR	RIV7,0,R1
	3404	077401		
904	3406		NZT	R1,SGLSTP
	3406	120622		
905	3410		XMIT	R3,21
	3410	141421		
906	3412		XMIT	IVR,5
	3412	147405		
907	3414		MOVE	RIV7,0,AUX
	3414	017400		
908	3416		XMIT	IVR,363
	3416	147763		
909	3420		AND	RIV7,0,RIV7
	3420	057437		
910	3422		XOR	RIV7,0,R1
	3422	077401		
911	3424		NZT	R1,SGLSTP
	3424	120622		
912	3426		XMIT	R3,0
	3426	141400		

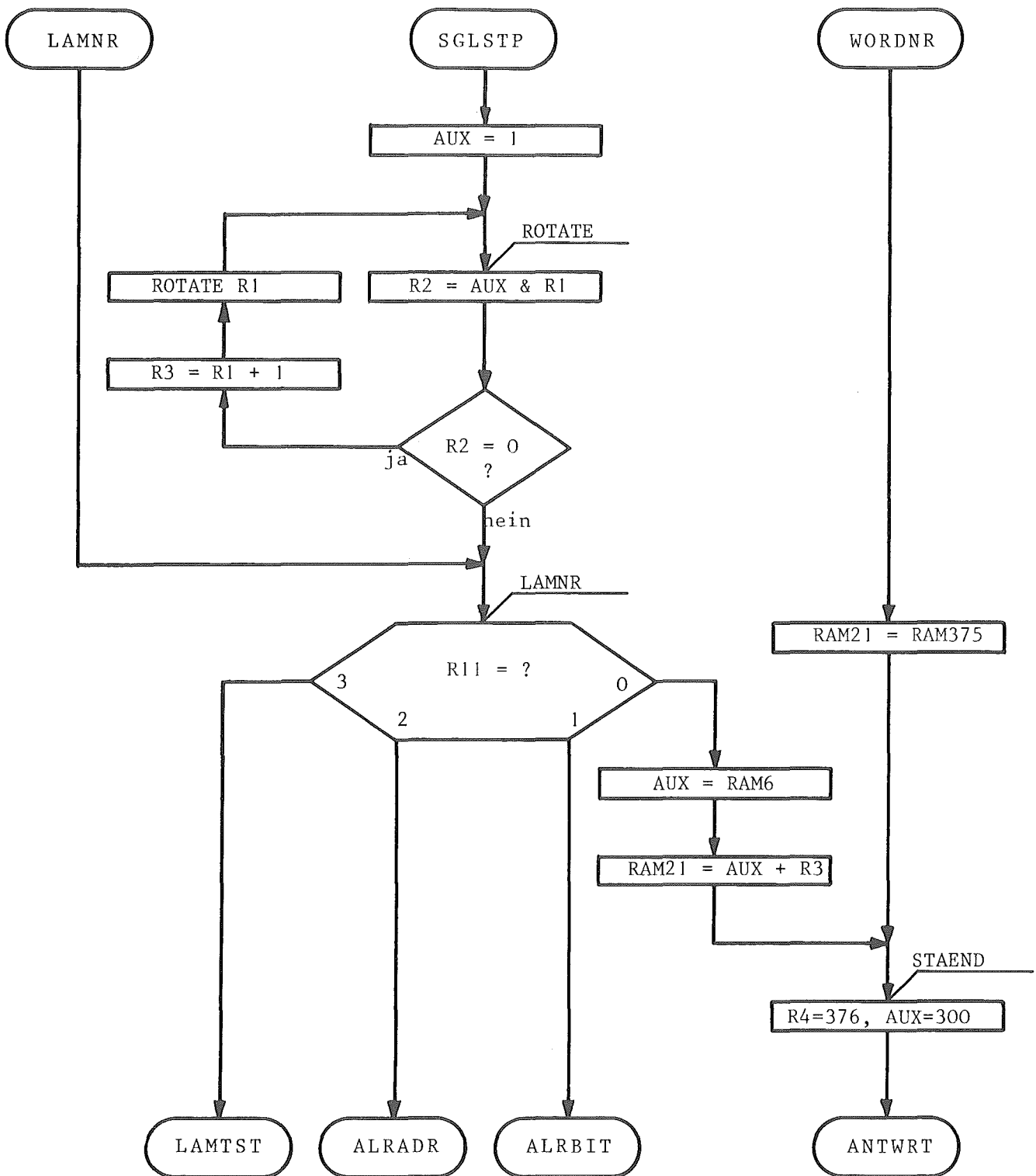


Abb. 10.2.19-3

913	3430	LAMNR:	XEC	R11, TABLE5
	3430	104630		
914	3432		MOVE	RIV4, 3, AUX
	3432	016140		
915	3434		XMIT	IVR, 21
	3434	147421		
916	3436		MOVE	R3, 5, RIV7
	3436	001677		
917	3440		MOVE	AUX, 3, RIV2
	3440	000172		
918	3442		JMP	STAEND
	3442	161546		
919	3444	SGLSTP:	XMIT	AUX, 1
	3444	140001		
920	3446	ROTATE:	AND	R1, 0, R6
	3446	040406		
921	3450		NZT	R6, LAMNR
	3450	123214		
922	3452		ADD	R3, 0, R3
	3452	021403		
923	3454		MOVE	R1, 1, R1
	3454	000441		
924	3456		JMP	ROTATE
	3456	161623		
925	3460	TABLE5:	XMIT	IVR, 6
	3460	147406		
926	3462		JMP	ALRBIT
	3462	161425		
927	3464		JMP	ALRADR
	3464	160675		
928	3466		JMP	LAMTST
	3466	161752		
871	3304	WORDNR:	XMIT	IVR, 375
	3304	147775		
872	3306		MOVE	RIV7, 0, AUX
	3306	017400		
873	3310		XMIT	IVR, 21
	3310	147421		
874	3312		MOVE	AUX, 0, RIV7
	3312	000037		
875	3314	STAEND:	XMIT	R4, 376
	3314	142376		
876	3316		XMIT	AUX, 300
	3316	140300		
877	3320		JMP	ANTWRT
	3320	161421		

10.2.20 Programmsegment RESTAT

Um das Datenübertragungssystem möglichst zeitoptimal zu nutzen, sollten einmal übertragene Alarmmeldungen (LAM - Signale) bei wiederholter Statusabfrage nicht mit übertragen werden. Um dem Controller die korrekte Alarrmeldungsaufnahme vom Treiber mitzuteilen, kann eine vom Treiber empfangene Statusmeldung als Nachricht zur Unterstation zurückgeschickt werden. Durch Vergleich dieser rückübertragenen Statusmeldung mit der im Controller Zwischenspeicher vorliegenden können die als korrekt übertragen erkannten LAM - Signale gelöscht werden. Dieses Zurücksetzen im Alarmzwischenspeicher wird durch das Programm RESTAT bewerkstelligt. Da in der Statusmeldung die binärcodierte LAM - Nummer des höchstprioreren LAM's übertragen wurde kann auch nur dieses höchstpriorere LAM im Zwischenspeicher zurückgesetzt werden.

Das Flußdiagramm und das Programmlisting sind den Abbildungen 10.2.20-1 und 10.2.20-2 zu entnehmen.

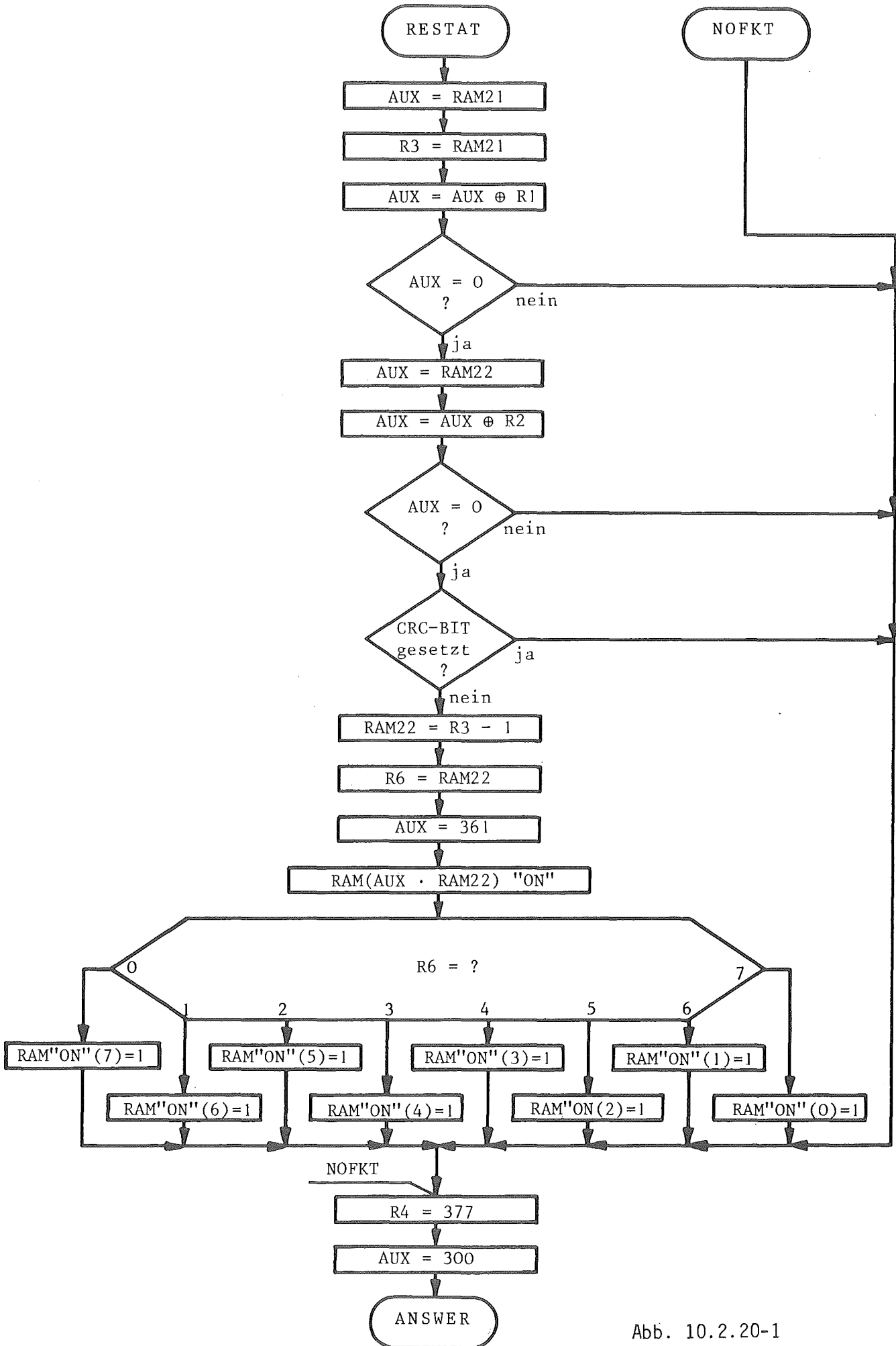


Abb. 10.2.20-1

416	1472		RESTAT:	XMIT	IUR, 21
	1472	147421			
417	1474			MOVE	RIV7, 0, AUX
	1474	017400			
418	1476			MOVE	RIV7, 5, R3
	1476	017643			
419	1500			XOR	R1, 0, AUX
	1500	060400			
420	1502			NZT	AUX, NOFKT
	1502	120255			
421	1504			XMIT	IUR, 22
	1504	147422			
422	1506			MOVE	RIV7, 0, AUX
	1506	017400			
423	1510			XOR	R2, 0, AUX
	1510	061000			
424	1512			NZT	AUX, NOFKT
	1512	120255			
425	1514			NZT	RIV4, 1, NOFKT
	1514	136055			
426	1516			XMIT	AUX, 377
	1516	140377			
427	1520			ADD	R3, 0, RIV7
	1520	021437			
428	1522			MOVE	RIV7, 3, R6
	1522	017546			
429	1524			XMIT	AUX, 361
	1524	140361			
430	1526			ADD	RIV4, 2, IUR
	1526	036117			
431	1530			XEC	R6, TABLE3
	1530	103260			
432	1532		NOFKT:	XMIT	R4, 377
	1532	142377			
433	1534			XMIT	AUX, 300
	1534	140300			
434	1536			JMP	ANSWER
	1536	161420			
435	1540		TABLE3:	XMIT	RIV7, 1, 1
	1540	157441			
436	1542			XMIT	RIV6, 1, 1
	1542	157041			
437	1544			XMIT	RIV5, 1, 1
	1544	156441			
438	1546			XMIT	RIV4, 1, 1
	1546	156041			
439	1550			XMIT	RIV3, 1, 1
	1550	155441			
440	1552			XMIT	RIV2, 1, 1
	1552	155041			
441	1554			XMIT	RIV1, 1, 1
	1554	154441			
442	1556			XMIT	RIV0, 1, 1
	1556	154041			

10.2.21 Programmsegment OUTPUT

In diesem Programmsegment geschieht die Synthese des 1. PDV - Wortes der Antwort. Gleichzeitig werden vorbereitende Arbeiten (Vorsetzen einiger Parameter) für die Funktion " Daten lesen " durchgeführt. Als erstes wird der im 2. Byte der Antwort zu sendende Kurzstatus (in R2) zusammengestellt. Bei einigen globalen Befehlen, bei denen kein Modul gezielt angesprochen wird, müssen entsprechend den CAMAC - Spezifikationen die CAMAC - Variablen X und Q gleich "1" gesetzt werden. Aus diesem Grund besitzt das Programmsegment zur Antwortsynthese zwei verschiedene Start- Statements. Bei Programmablauf vom Statement OUTPUT wird X und Q gelesen und in den Kurzstatus eingesetzt. Beim Ablauf ab Statement ANSWER sind die Variablen X und Q bereits gleich "1" gesetzt. Zur Feststellung noch nicht abgesetzter Alarmmeldungen (LAM - Signale) wir ins Unterprogramm mit dem Statement LAMADR gesprungen. Im weiteren Programmablauf wird das Summen - Alarmsignal (aufgesetzt bei Vorliegen eines noch nicht gemeldeten LAM - Signales) dem Kurzstatus hinzugefügt. Ebenso wird das aus dem IV - Byte ausgelesene ONLINE Bit dem Register R2 zuaddiert und nach Prüfung des Registerinhaltes R4 (Antwortblocklänge) bei Bedarf das eine Blockantwort kennzeichnende 2. Bit hinzuaddiert. Anschließend wird das OUTPUT FIFO zurückgesetzt und ins Unterprogramm zur CRC - Bildung gesprungen. Ab dem Rücksprungstatement RET2 werden die Registerinhalte R1 und R2 sowie der im Hilfsregister stehende CRC in das OUTPUT FIFO geladen. Dabei wird jeweils vor Ausführen der Ladefunktion überprüft ob das Eingangsregister des FIFO's frei ist. Sind alle drei Bytes des 1. PDV - Wortes in das OUTPUT FIFO geladen so wird durch Aufbringen des entsprechenden Bit - Patterns auf das IV - Byte das Sendebereitschaftssignal gesetzt. Durch inkrementieren des Komplements der Antwortblocklänge in Register R4 wird festgestellt ob die Antwort aus mehr als einem PDV - Wort besteht. Ist dies der Fall so sollen Daten gelesen werden und das Programm verzweigt zum Statement NXTWRD. Besteht die Antwort aus nur einem PDV - Wort so wird der Programmablauf bei dem Statement GOBACK fortgesetzt.

Das für das Programmsegment entwickelte Flußdiagramm ist abschnittsweise auf die Abbildungen 10.2.21-1, 10.2.21-3 und 10.2.21-5 verteilt. Die entsprechenden Programmlistingabschnitte sind den Abbildungen 10.2.21-2, 10.2.21-4 und 10.2.21-6 zu entnehmen.

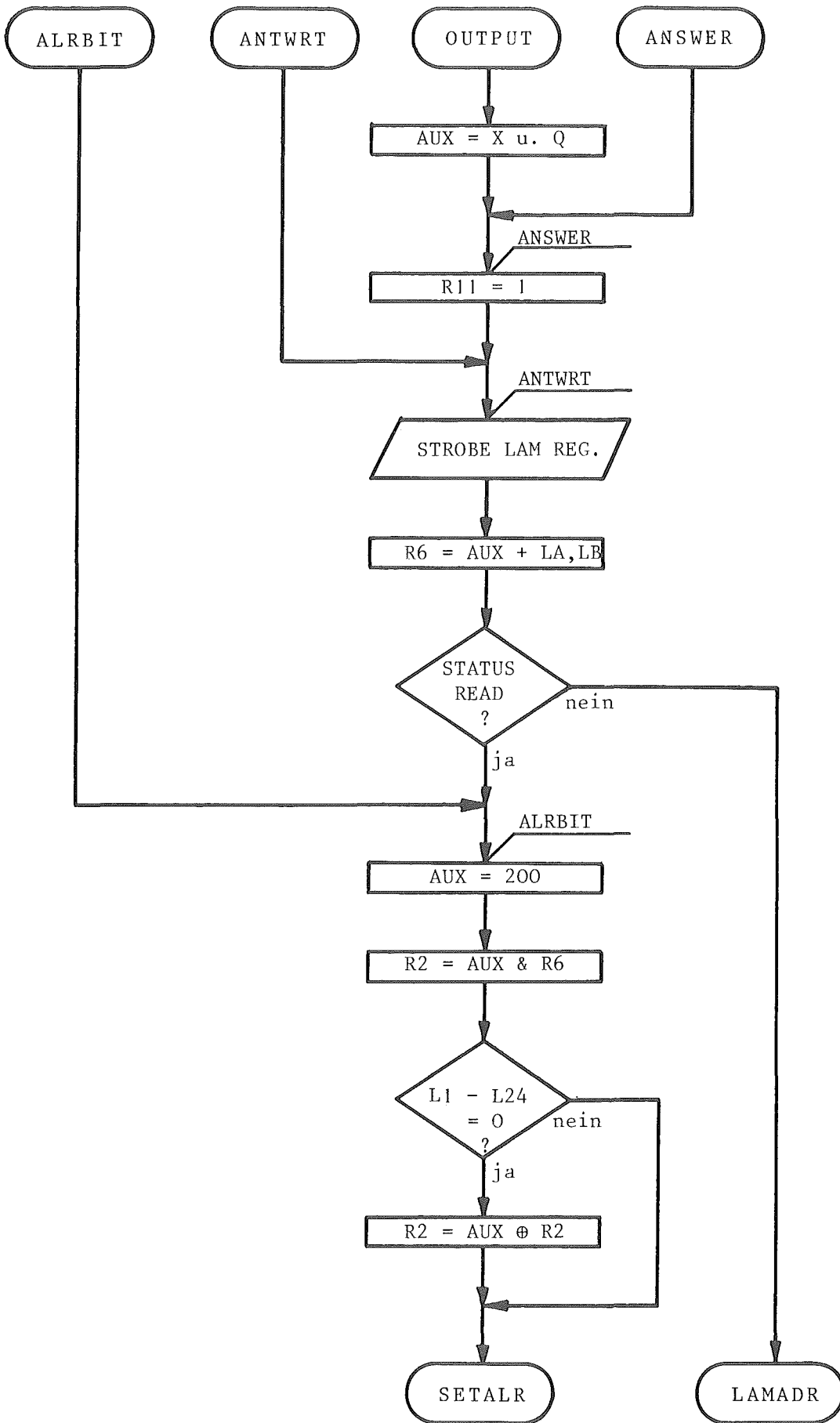


Abb. 10.2.21-1

786	3032	OUTPUT:	XMIT	IUR,6
	3032	147406		
787	3034		MOVE	RIV7,2,AUX
	3034	017500		
788	3036		MOVE	AUX,2,AUX
	3036	000100		
789	3040	ANSWER:	XMIT	R11,1
	3040	144401		
790	3042	ANTWRT:	XOR	AUX,0,LIV7
	3042	060027		
791	3044		XMIT	IUR,6
	3044	147406		
792	3046		ADD	RIV1,2,R2
	3046	034502		
793	3050		NZT	R11,LAMADR
	3050	124565		
794	3052	ALRBIT:	XMIT	AUX,200
	3052	140200		
795	3054		ADD	R2,3,R2
	3054	021142		
796	3056		NZT	R3,SETALR
	3056	121431		
797	3060		XOR	R2,0,R2
	3060	061002		

Abb. 10.2.21-2

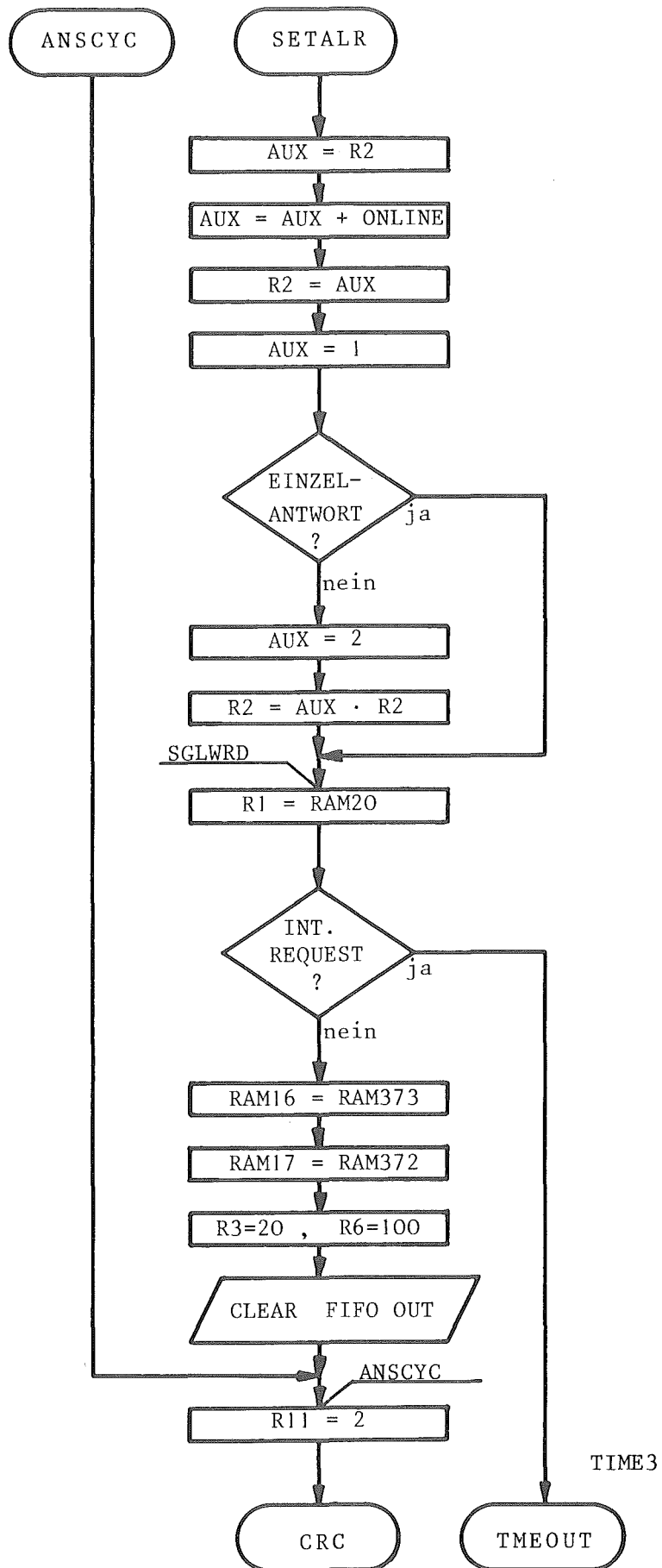


Abb. 10.2.21-3

798	3062	SETALR:	MOVE	R2,0,AUX
	3062	001000		
799	3064		ADD	LIV4,1,AUX
	3064	032040		
800	3066		MOVE	AUX,1,R2
	3066	000042		
801	3070		XMIT	AUX,1
	3070	140001		
802	3072		ADD	R4,0,R11
	3072	022011		
803	3074		NZT	OVF,SGLWRD
	3074	124041		
804	3076		XMIT	AUX,2
	3076	140002		
805	3100		ADD	R2,0,R2
	3100	021002		
806	3102	SGLWRD:	XMIT	IVR,20
	3102	147420		
807	3104		MOVE	RIV7,0,R1
	3104	017401		
808	3106		XMIT	IVR,373
	3106	147773		
809	3110		MOVE	RIV7,0,R3
	3110	017403		
810	3112		NZT	LIV7,1,TIME3
	3112	133461		
811	3114		XMIT	IVR,16
	3114	147416		
812	3116		MOVE	R3,0,RIV7
	3116	001437		
813	3120		XMIT	IVR,372
	3120	147772		
814	3122		MOVE	RIV7,0,R3
	3122	017403		
815	3124		XMIT	IVR,17
	3124	147417		
816	3126		MOVE	R3,0,RIV7
	3126	001437		
817	3130		XMIT	R3,20
	3130	141420		
818	3132		XMIT	R6,300
	3132	143300		
819	3134		MOVE	R3,0,LIV7
	3134	001427		
820	3136	ANSCYC:	XMIT	R11,2
	3136	144402		
821	3140		JMP	CRC
	3140	160774		
822	3142	TIME3:	JMP	TMEOUT
	3142	160632		

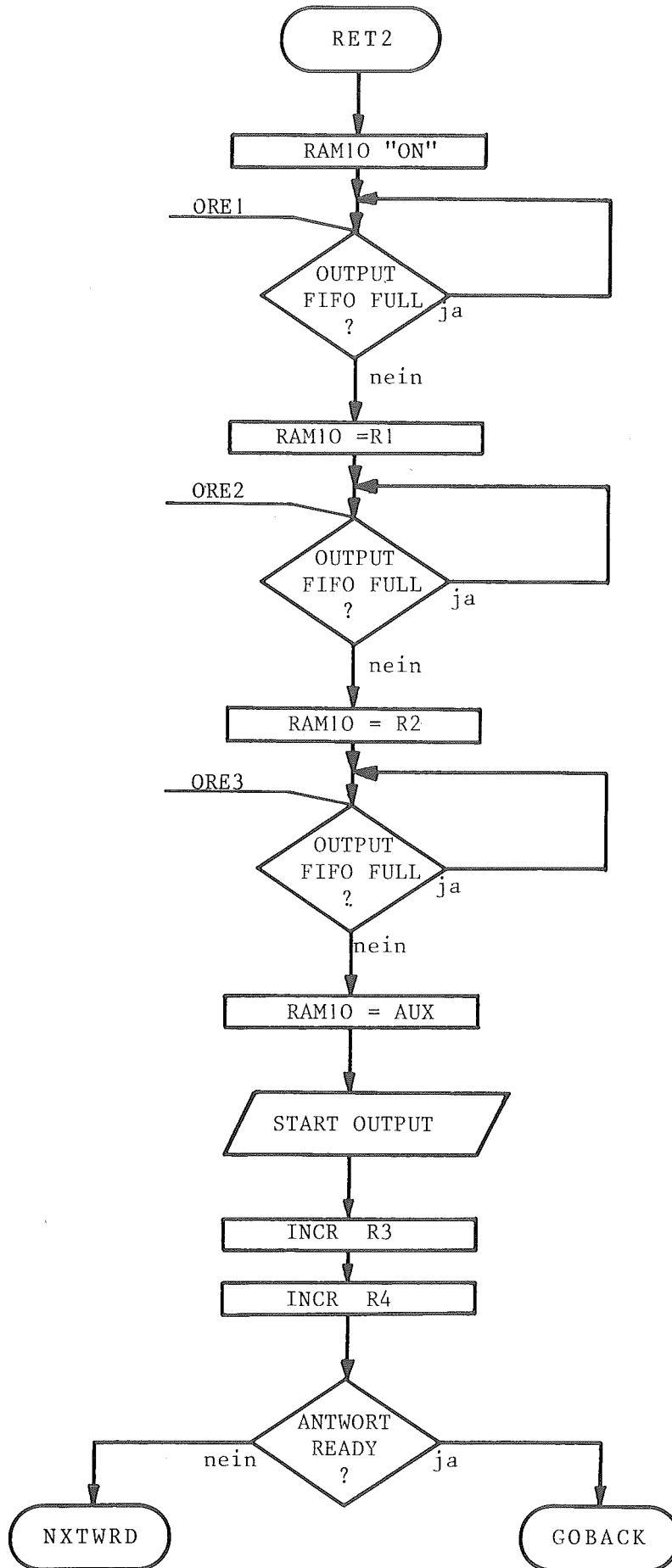


Abb. 10.2.21-5

823	3144		RET2:	XMIT	IVR,10
	3144	147410			
824	3146		ORE1:	NZT	LIV3,1,ORE1
	3146	131463			
825	3150			MOVE	R1,0,RIV7
	3150	000437			
826	3152		ORE2:	NZT	LIV3,1,ORE2
	3152	131465			
827	3154			MOVE	R2,0,RIV7
	3154	001037			
828	3156		ORE3:	NZT	LIV3,1,ORE3
	3156	131467			
829	3160			MOVE	AUX,0,RIV7
	3160	000037			
830	3162			MOVE	R6,0,LIV7
	3162	003027			
831	3164			XMIT	AUX,1
	3164	140001			
832	3166			ADD	R3,0,R3
	3166	021403			
833	3170			ADD	R4,0,R4
	3170	022004			
834	3172			NZT	R4,NXTWRD
	3172	122077			
835	3174			JMP	GOBACK
	3174	160272			

Abb. 10.2.21-6

10.2.22 Programmsegment NXTWRD

Dieses Programmsegment wird immer dann aktiviert wenn eine PDV - Antwort aus mehr als einem PDV - Wort besteht (Blockantwort). Alle PDV - Worte, die nach dem 1. PDV - Wort als Antwort abgegeben werden, beinhalten Datenteile wobei auch der Status und der Inhalt der CAMAC - LAM - Register als Daten angesehen werden. Für den Fall das eine Blockantwort fehlerhaft zum Treiber übermittelt wurde, muß dieser die Daten erneut abfragen können (gleiche Nachricht mit gesetztem Bit für Wiederholungsfunktion). Da im Falle der Wiederholung keine erneuten CAMAC - Lesezyklen generiert werden dürfen (z.B. bei selbstdekrementierenden FIFO - Bufferspeichern) müssen die zuvor gelesenen Daten im Controller abgespeichert werden. Aus diesem Grund werden bei einer PDV - Blockantwort alle zu übertragenden Datenteile vor absenden der Antwort im RAM - Speicher des Controllers abgelegt. Jede Blockantwort des Controllers besteht also aus dem zusammengesetzten 1. PDV-Wort und den ab RAM - Zelle 21 in aufsteigender Reihenfolge abgelegten Datenteilen. Unter dem Statement NXTWRD wird also zuerst überprüft ob die angesprochene Funktion eine Wiederholungsfunktion ist. Nur für den Fall das keine Wiederholung vorliegt aber Daten gelesen werden sollen wird zum Programmstatement READ verzweigt. Unter diesem Statement erfolgt eine Überprüfung des gewünschten Leseformats und im Falle für 24bit breite Lesedaten zum Statement RFRM24 weitersprungen. Hier erfolgt die Überprüfung und Entscheidung ob die zur Bildung des PDV - Wortes herangezogenen Datenbytes die Lowerbytes (Daten der CAMAC - Readleitungen R1 - R16) oder das Highbyte (CAMAC - Readleitungen R17 - R24) sind. Für den Fall der Übertragung des Highbytes darf kein CAMAC - READ - Zyklus erzeugt werden, da diese Daten durch den vorangegangenen CAMAC - Zyklus bereits in das entsprechende Register aufgenommen wurden. Sollen die Lowerbytes gelesen und übertragen werden so muß ein CAMAC- Lesezyklus generiert werden. Unter dem Statement RDCYCL wird dafür zunächst der interne Request überprüft, anschließend das ACL - Signal gesetzt und der CAMAC - Zyklus angestoßen. Die gelesenen Lowerbytes werden in die Register R1 und R2 übernommen und ab dem Statement STORE in den entsprechenden RAM - Zellen abgespeichert. Zur Antwortabgabe (einschreiben in das OUTPUT - FIFO) wird zum Statement ANSCYC zurückgesprungen. Die Abbildung 10.2.22-1 zeigt das Flußdiagramm und die Abbildung 10.2.22-2 das Programmlisting.

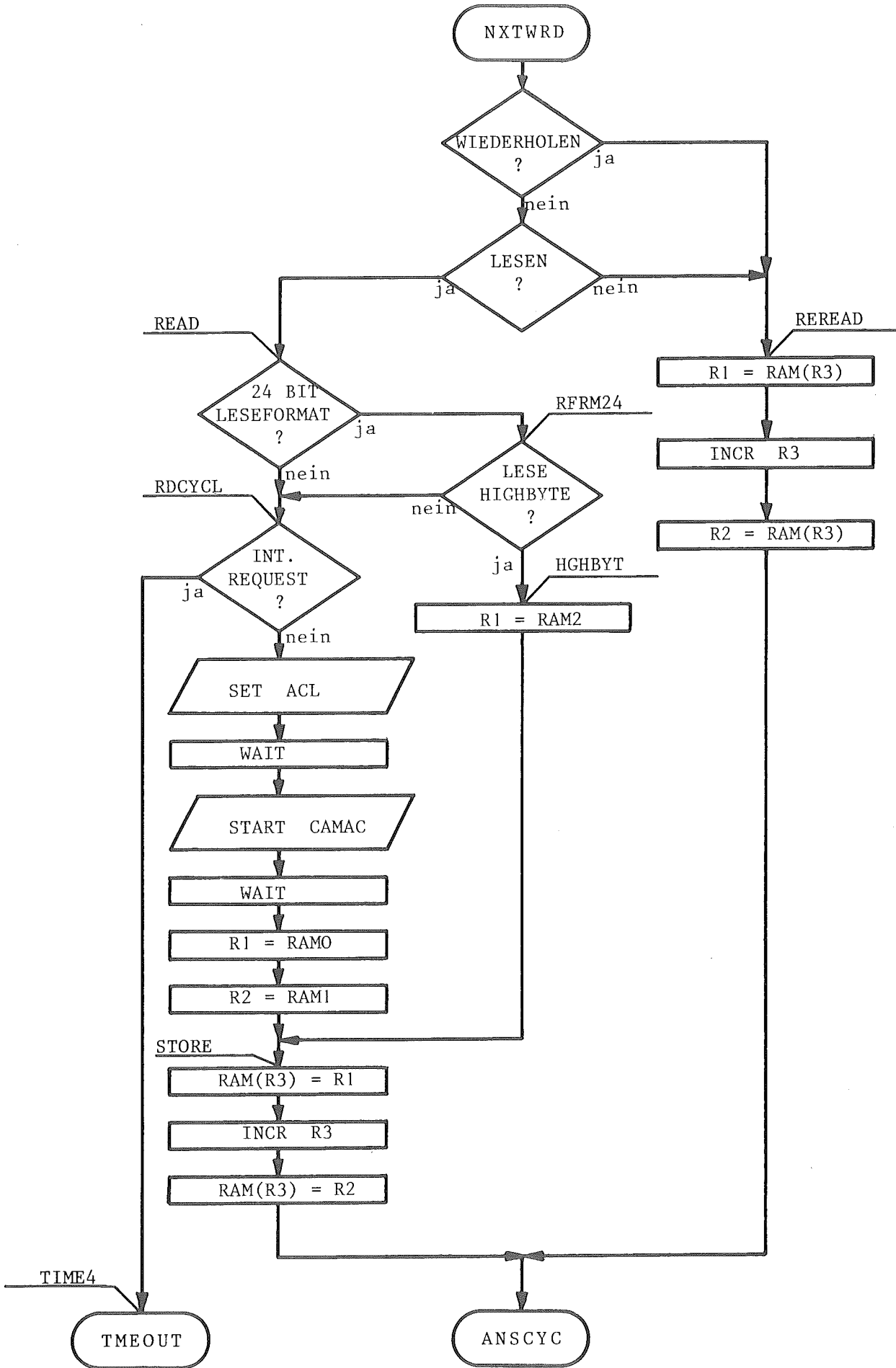


Abb. 10.2.22-1

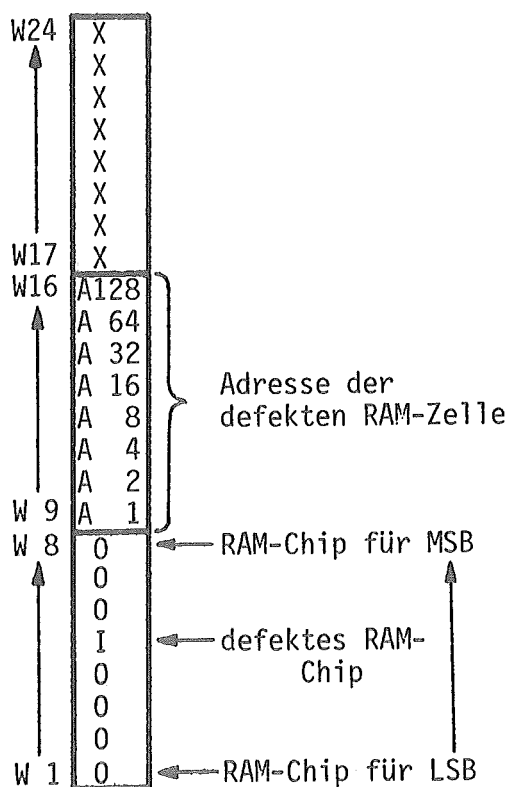
836	3176		NXTWRD:	XMIT	IVR,377
	3176	147777			
837	3200			NZT	RIV1,1,REREAD
	3200	134442			
838	3202			NZT	RIV3,1,READ
	3202	135457			
839	3204		REREAD:	MOVE	R3,0,IVR
	3204	001417			
840	3206			MOVE	RIV7,0,R1
	3206	017401			
841	3210			ADD	R3,0,R3
	3210	021403			
842	3212			MOVE	R3,0,IVR
	3212	001417			
843	3214			MOVE	RIV7,0,R2
	3214	017402			
844	3216			JMP	ANSCYC
	3216	161457			
845	3220		RFRM24:	AND	R3,1,R1
	3220	041441			
846	3222			NZT	R1,HGHBYT
	3222	120513			
847	3224			JMP	RDCYCL
	3224	161521			
848	3226		HGHBYT:	XMIT	IVR,2
	3226	147402			
849	3230			MOVE	RIV7,0,R1
	3230	017401			
850	3232			JMP	STORE
	3232	161534			
851	3234		TIME4:	JMP	TMEOUT
	3234	160632			
852	3236		READ:	XMIT	IVR,373
	3236	147773			
853	3240			NZT	RIV2,1,RFRM24
	3240	135050			
854	3242		RDCYCL:	NZT	LIV7,1,TIME4
	3242	133456			
855	3244			XMIT	R1,100
	3244	140500			
856	3246			MOVE	R1,0,LIV7
	3246	000427			
857	3250			MOVE	AUX,0,AUX
	3250	000000			
858	3252			XMIT	R1,40
	3252	140440			
859	3254			MOVE	R1,0,LIV7
	3254	000427			
860	3256			MOVE	AUX,0,AUX
	3256	000000			
861	3260			XMIT	IVR,0
	3260	147400			
862	3262			MOVE	RIV7,0,R1
	3262	017401			
863	3264			XMIT	IVR,1
	3264	147401			
864	3266			MOVE	RIV7,0,R2
	3266	017402			
865	3270		STORE:	MOVE	R3,0,IVR
	3270	001417			
866	3272			MOVE	R1,0,RIV7
	3272	000437			
867	3274			ADD	R3,0,R3
	3274	021403			
868	3276			MOVE	R3,0,IVR
	3276	001417			
869	3300			MOVE	R2,0,RIV7
	3300	001037			
870	3302			JMP	ANSCYC
	3302	161457			

10.2.23 Programmsegment TEST

Dieses Programmsegment ermöglicht eine Funktionskontrolle des Controllers und erlaubt die Überprüfung aller Prozessorfunktionen, Schreib- und Leseoperationen des RAM Speichers sowie aller CAMAC Schreib- und Leseregister. Die Aktivierung dieses TEST- Programmabschnittes geschieht durch Betätigen der Taste "C" (CAMAC - Löschroutine) wobei die Taste länger als 5 sec gedrückt bleiben muß und der Controller sich im OFFLINE - Zustand befindet. Insgesamt werden drei verschiedene Testroutinen nacheinander durchgeführt. Die augenblicklich vorgenommene Testfunktion wird über das Crateadressen - Anzeigedisplay durch Ausgabe der Ziffern 1, 2 und 3 angezeigt. Die einzelnen Testroutinen haben folgende Funktionen:

1) Test des Arbeitsspeichers:

Hierbei wird in jede RAM - Speicherzelle beginnend bei RAM - Zelle 20_8 bis 377_8 alle binären Werte von 0 bis 255 eingeschrieben, wieder ausgelesen und verglichen. Wird bei der Überprüfung des RAM - Arbeitsspeichers eine fehlerhaft arbeitende RAM - Zelle entdeckt so wird der Weiterlauf des Programms gestoppt und über das Dataway Display-Modul die Nummer des fehlerhaften RAM - Chips und die Adresse der fehlerhaften Zelle angezeigt. Für die Ausgabe über das Display Modul gelten folgende Zuordnungen:

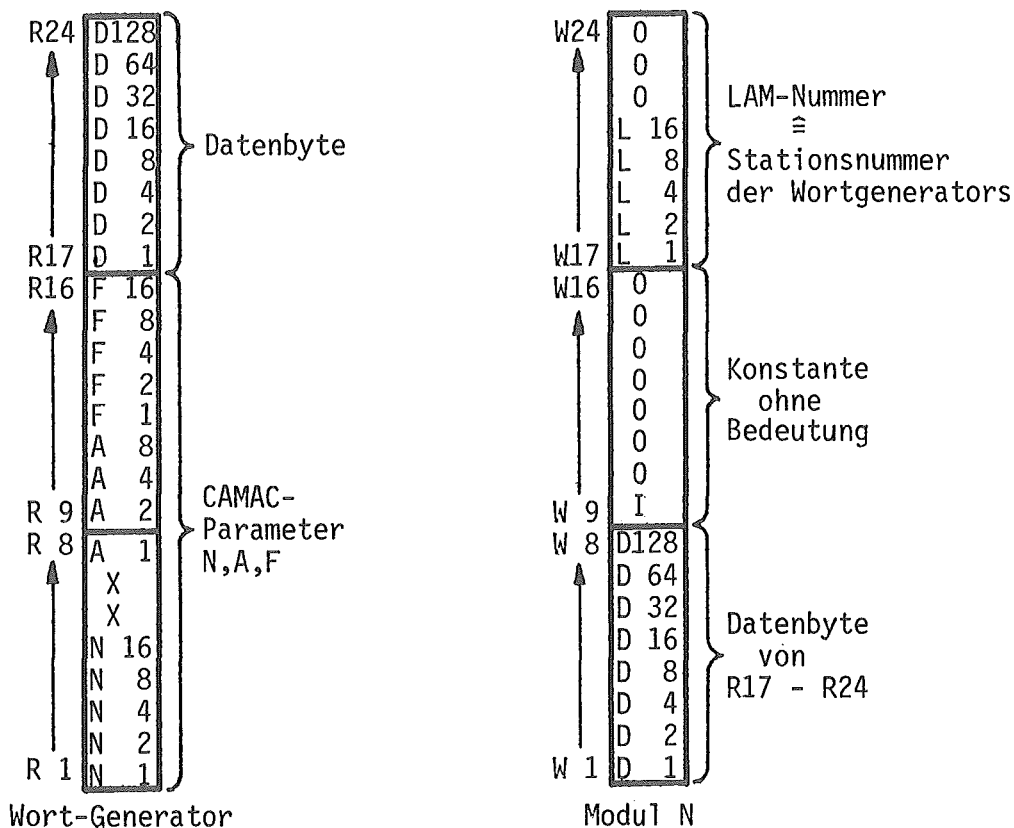


2) Testen der CAMAC - Schreibregister

Um diese Register zu testen wird ein CAMAC - Dataway Display Modul benötigt welches im gleichen Crate eingesteckt sein muß. Da keine direkte Reaktion von einem Modul erwartet werden kann ist für diesen Test die optische Überprüfung am Dataway Display erforderlich. Dazu wird auf dem 24bit breiten CAMAC - Datenweg eine "1" ausgeschrieben und um eine Bitposition rotiert. Nach jeweils 25 Rotationen und Ausschreiben des Bitpatterns wird eine "1" addiert. Dies wird solange fortgesetzt bis alle Write - Leitungen den Zustand logisch "1" besitzen.

3) Test des Leseregisters (LAM - Readregister)

Um diese Register zu überprüfen ist ein CAMAC - Wortgenerator erforderlich der an beliebiger Position aber im gleichen Crate sitzen muß. Bei dieser dritten Testfunktion arbeitet der Controller als manuelles Gerät, d.h. mit Hilfe des Wortgenerators können beliebige CAMAC - Funktionen abgesetzt werden. Mit Hilfe der Schalter des CAMAC - Wortgenerators werden dabei die CAMAC - Parameter N, A und F gewählt. Mit Hilfe der 8 höchstwertigsten Schalter kann der Wert eines Datenbytes eingestellt werden. Durch Absetzen eines LAM - Signales des Wortgenerators wird die Einstellung der Schalter automatisch ausgelesen und ein entsprechender CAMAC - Zyklus generiert. Die Zuordnung der einzelnen Schalter sowie die durch die Testfunktion auf die Write - Leitungen abgesetzten Daten sind der folgenden Skizze zu entnehmen.



Die dritte Testfunktion (manueller Controller) bleibt solange erhalten bis der Prozessor zurückgesetzt wird (ONLINE schalten und betätigen der "Z" Taste).

Das Flußdiagramm des Programmsegmentes TEST kann den Abbildungen 10.2.23-1, 10.2.23-3, 10.2.23-5 und 10.2.23-7 entnommen werden. Die Abbildungen 10.2.23-2, 10.2.23-4, 10.2.23-6 und 10.2.23-8 zeigen das Programmlisting.

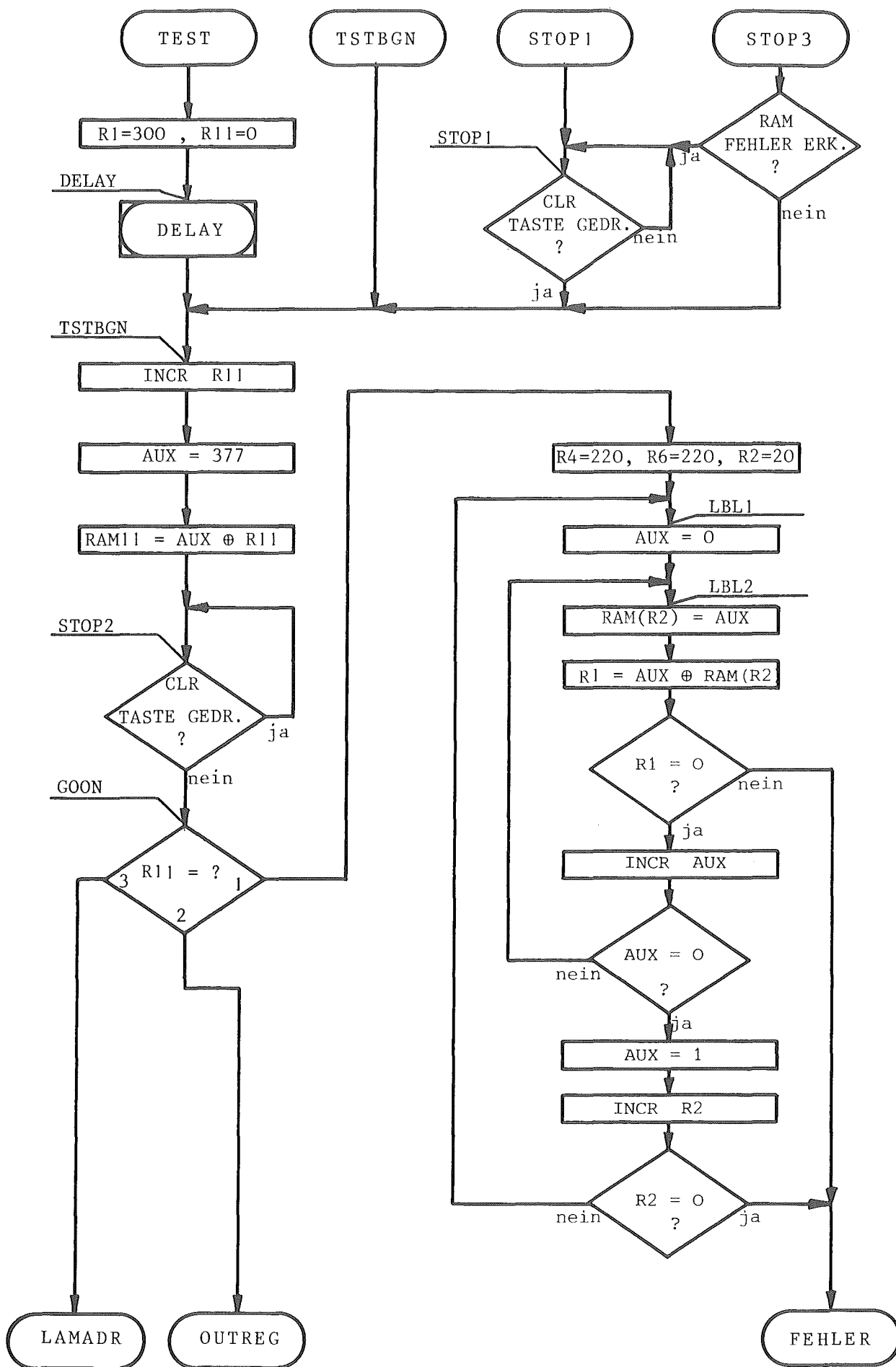


Abb. 10.2.23-1

507	1760	TEST:	XMIT	R1,300
	1760	140700		
508	1762		XMIT	R11,0
	1762	144400		
509	1764		JMP	DELAY
	1764	161721		
510	1766		MOVE	AUX,0,AUX
	1766	000000		
929	3470	STOP3:	NZT	R1,STOP1
	3470	120636		
930	3472		JMP	TSTBGN
	3472	161637		
931	3474	STOP1:	NZT	LIV0,1,STOP1
	3474	130076		
932	3476	TSTBGN:	XMIT	AUX,1
	3476	140001		
933	3500		ADD	R11,0,R11
	3500	024411		
934	3502		XMIT	IVR,11
	3502	147411		
935	3504		XMIT	AUX,377
	3504	140377		
936	3506		XOR	R11,0,RIV7
	3506	064437		
937	3510	STOP2:	NZT	LIV0,1,GOON
	3510	130046		
938	3512		JMP	STOP2
	3512	161644		
939	3514	GOON:	XEC	R11,TABLE6
	3514	104663		
940	3516		XMIT	R6,220
	3516	143220		
941	3520		XMIT	R2,20
	3520	141020		
942	3522	LBL1:	XMIT	AUX,0
	3522	140000		
943	3524		MOVE	R2,0,IVR
	3524	001017		
944	3526	LBL2:	MOVE	AUX,0,RIV7
	3526	000037		
945	3530		XOR	RIV7,0,R1
	3530	077401		
946	3532		NZT	R1,FEHLER
	3532	120772		
947	3534		ADD	R5,0,AUX
	3534	022400		
948	3536		NZT	AUX,LBL2
	3536	120253		
949	3540		XMIT	AUX,1
	3540	140001		
950	3542		ADD	R2,0,R2
	3542	021002		
951	3544		NZT	R2,LBL1
	3544	121251		
952	3546	TABLE6:	JMP	FEHLER
	3546	161772		
953	3550		XMIT	R4,220
	3550	142220		
954	3552		JMP	OUTREG
	3552	161667		
955	3554		JMP	LAMADR

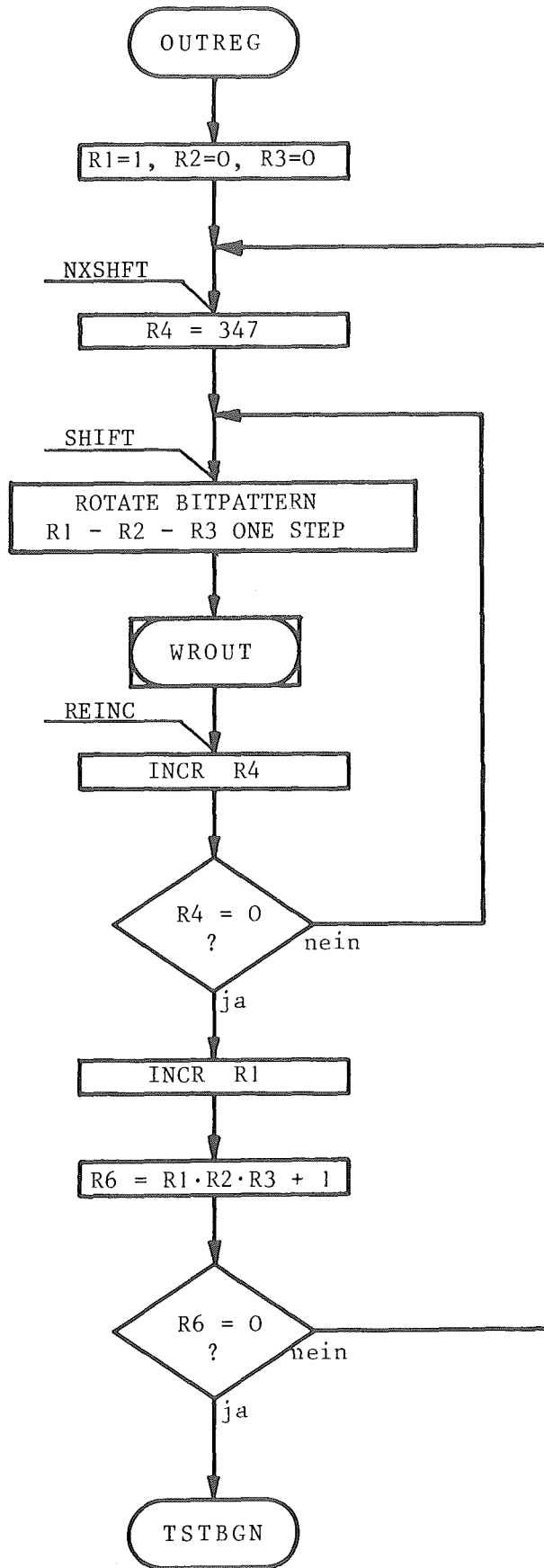


Abb. 10.2.23-3

956	3556	OUTREG:	XMIT	R3,0
	3556 141400			
957	3560		XMIT	R2,0
	3560 141000			
958	3562		XMIT	R1,1
	3562 140401			
959	3564	NXSHFT:	XMIT	R4,347
	3564 142347			
960	3566	SHIFT:	MOVE	R3,7,AUX
	3566 001740			
961	3570		AND	R5,0,AUX
	3570 042400			
962	3572		XOR	R3,7,R3
	3572 061743			
963	3574		MOVE	AUX,0,R6
	3574 000006			
964	3576		MOVE	R2,7,AUX
	3576 001340			
965	3600		AND	R5,0,AUX
	3600 042400			
966	3602		XOR	R2,7,R2
	3602 061342			
967	3604		ADD	R3,0,R3
	3604 021403			
968	3606		MOVE	R1,7,AUX
	3606 000740			
969	3610		AND	R5,0,AUX
	3610 042400			
970	3612		ADD	R2,0,R2
	3612 021002			
971	3614		XOR	R1,7,AUX
	3614 060740			
972	3616		ADD	R6,0,R1
	3616 023001			
998	3702	REINC:	ADD	R4,0,R4
	3702 022004			
999	3704		NZT	R4,SHIFT
	3704 122273			
1000	706		ADD	R1,0,R1
	706 020401			
1001	710		MOVE	R1,0,AUX
	710 000400			
1002	712		AND	R2,0,AUX
	712 041000			
1003	714		AND	R3,0,AUX
	714 041400			
1004	716		ADD	R5,0,R6
	716 022406			
1005	720		NZT	R6,NXSHFT
	720 123272			
1006	722		JMP	TSTBGN
	722 161637			

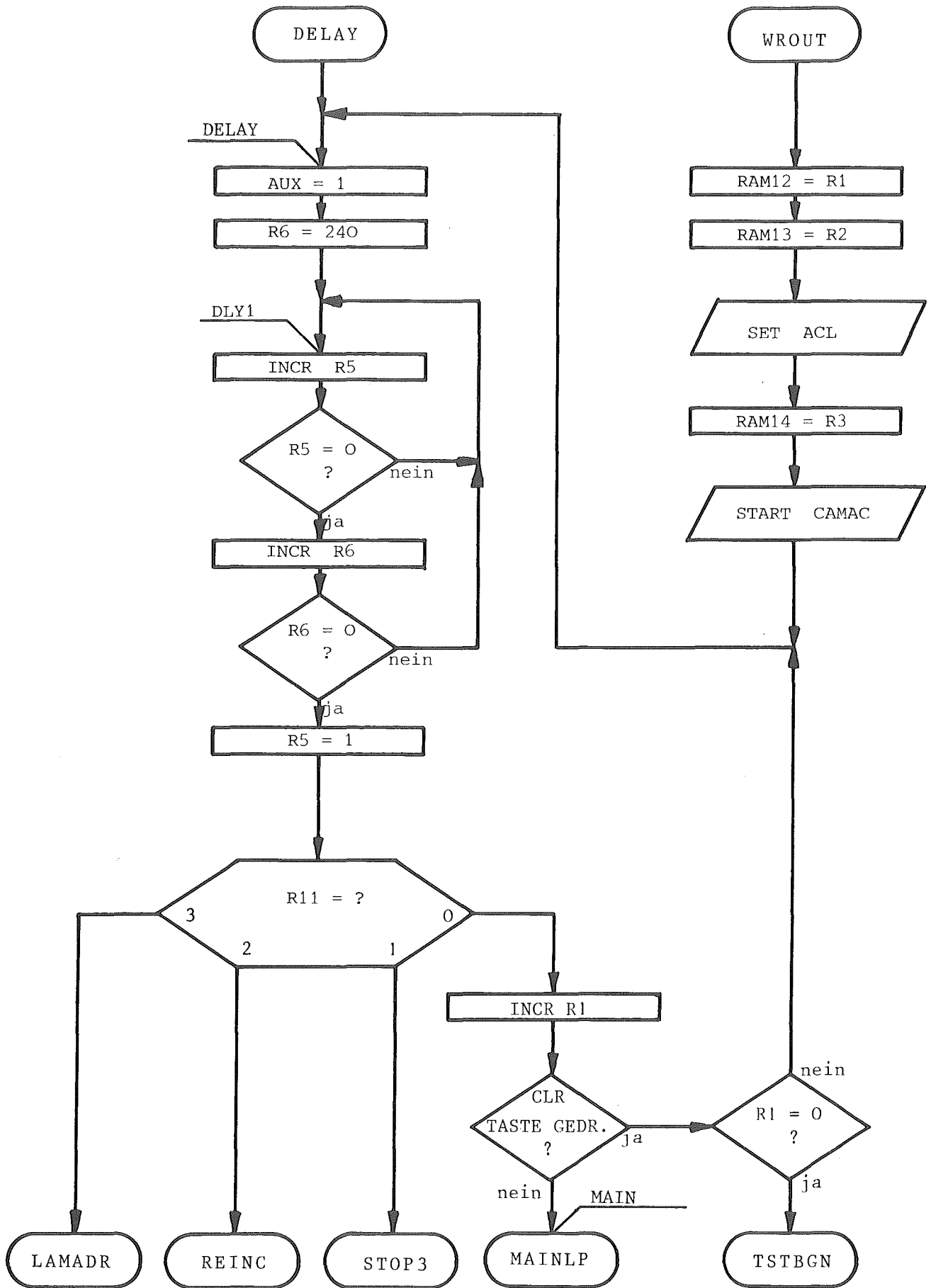


Abb. 10.2.23-5

973	3620	WROUT:	XMIT	IVR,12
	3620	147412		
974	3622		MOVE	R1,0,RIV7
	3622	000437		
975	3624		XMIT	IVR,13
	3624	147413		
976	3626		MOVE	R2,0,RIV7
	3626	001037		
977	3630		XMIT	AUX,40
	3630	140040		
978	3632		ADD	AUX,0,LIV7
	3632	020027		
979	3634		XMIT	IVR,14
	3634	147414		
980	3636		MOVE	R3,0,RIV7
	3636	001437		
981	3640		MOVE	AUX,0,LIV7
	3640	000027		
982	3642	DELAY:	XMIT	AUX,1
	3642	140001		
983	3644		XMIT	R6,240
	3644	143240		
984	3646	DLY1:	ADD	R5,0,R5
	3646	022405		
985	3650		NZT	R5,DLY1
	3650	122723		
986	3652		ADD	R6,0,R6
	3652	023006		
987	3654		NZT	R6,DLY1
	3654	123323		
988	3656		XMIT	R5,1
	3656	142401		
989	3660		XEC	R11, TABLE7
	3660	104735		
990	3662		NZT	LIV0,1,MAIN
	3662	130074		
991	3664		NZT	R1,DELAY
	3664	120721		
992	3666		JMP	TSTBGN
	3666	161637		
993	3670	MAIN:	JMP	MAINLP
	3670	160043		
994	3672	TABLE7:	ADD	R1,0,R1
	3672	020401		
995	3674		JMP	STOP3
	3674	161634		
996	3676		JMP	REINC
	3676	161741		
997	3700		JMP	LAMADR
	3700	161565		

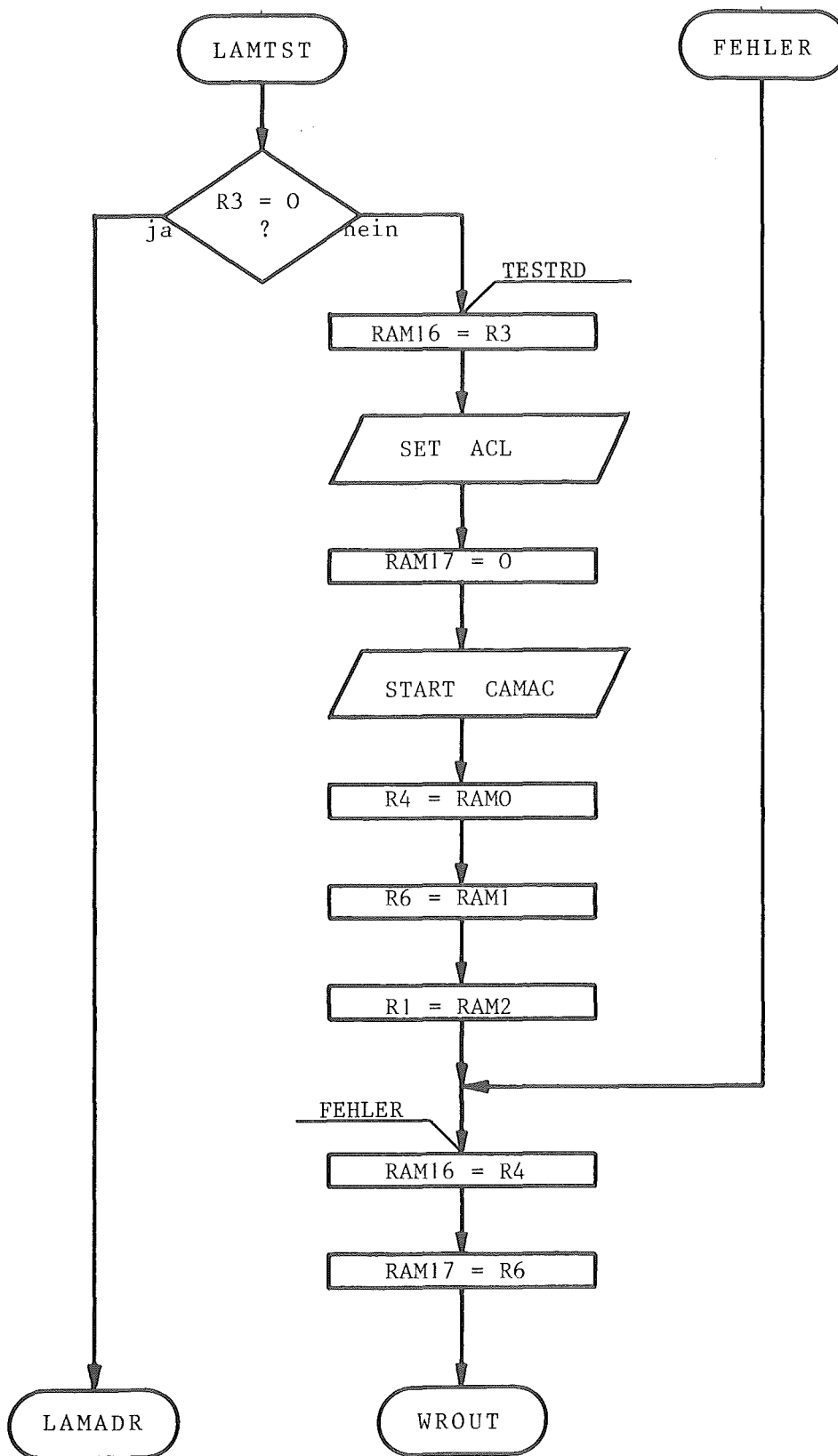


Abb. 10.2.23-7


```
1007 724          LAMTST: NZT      R3,TESTRD
      724 121754
1008 726          JMP      LAMADR
      726 161565
1009 730          TESTRD: XMIT    IVR,16
      730 147416
1010 732          MOVE     R3,0,RIV7
      732 001437
1011 734          XMIT     AUX,40
      734 140040
1012 736          ADD      AUX,0,LIV7
      736 020027
1013 740          XMIT     IVR,17
      740 147417
1014 742          XOR      AUX,0,RIV7
      742 060037
1015 744          MOVE     AUX,0,LIV7
      744 000027
1016 746          MOVE     AUX,0,AUX
      746 000000
1017 750          XMIT     IVR,0
      750 147400
1018 752          MOVE     RIV7,0,R4
      752 017404
1019 754          XMIT     IVR,1
      754 147401
1020 756          MOVE     RIV7,0,R6
      756 017406
1021 760          XMIT     IVR,2
      760 147402
1022 762          MOVE     RIV7,0,R1
      762 017401
1023 764          FEHLER: XMIT    IVR,16
      764 147416
1024 766          MOVE     R4,0,RIV7
      766 002037
1025 770          XMIT     IVR,17
      770 147417
1026 772          MOVE     R6,0,RIV7
      772 003037
1027 774          JMP      WROUT
      774 161710
1028 776          MOVE     AUX,0,AUX
      776 000000
1029          000001'      .END
```

Abb. 10.2.23-8

11. Literaturverzeichnis

1. PDV-Arbeitskreis TP 30, KFK-PDV 70, Serielles Bussystem für industrielle Anwendungen unter Echtzeitbedingungen (PDV - BUS), Gesellschaft für Kernforschung mbH, 7500 Karlsruhe 1, Postfach 3640, Mai 1976; neue Ausgabe in Vorbereitung (KFK-PDV -150).

2. Commission of the European Communities, CAMAC - A MODULAR INSTRUMENTATION SYSTEM FOR DATA HANDLING, EUR 4100e Luxembourg, August 1972; IEC Recommendation 516

3. M. Sauer, K. Zwoil
Microprocessor Based Flexible Interface for Communication Protocol Handling between the CAMAC - System and Serial PDV - Data Link, EUROMICRO Symposium, München, Okt. 1978

4. A.H.J. Schatorjé, 8X300 INSTRUCTION SET, PHILIPS Laboratory report Nr. EDP7703, Feb. 1977