

**KERNFORSCHUNGSZENTRUM**

**KARLSRUHE**

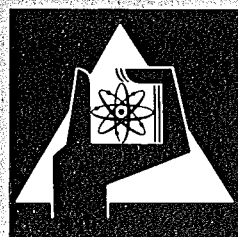
Oktober 1973

KFK 1873

Labor für Elektronik und Meßtechnik

**CAMAC System Controller für Telefunken-Rechner TR86**

P.H. Gruber, J.G. Ottens



**GESELLSCHAFT  
FÜR  
KERNFORSCHUNG M.B.H.**

**KARLSRUHE**

Als Manuskript vervielfältigt

Für diesen Bericht behalten wir uns alle Rechte vor

GESELLSCHAFT FÜR KERNFORSCHUNG M. B. H.  
KARLSRUHE

KERNFORSCHUNGSZENTRUM KARLSRUHE

KFK 1873

Labor für Elektronik und Meßtechnik

CAMAC System Controller für Telefunken-Rechner TR86

P. H. Gruber

J. G. Ottens

GESELLSCHAFT FÜR KERNFORSCHUNG, M. B. H., KARLSRUHE



## CAMAC System Controller für Telefunken-Rechner TR86

### Zusammenfassung

Es wird ein CAMAC System Controller beschrieben, der einen voll ausgebauten Branch bedienen kann und für den Anschluß an einen Rechner vom Typ Telefunken TR86 ausgelegt ist. Mit dem Controller können programmgesteuerte Einzelwort-Transfers, vier verschiedene Arten von Block-Transfers und autonome Transfers nach dem Cycle-Stealing-Verfahren durchgeführt werden. Es gibt insgesamt 8 Interrupt-Ebenen und 16 schnelle autonome Kanäle. Jedes von 7 möglichen Crates belegt eine Interrupt-Ebene; die achte ist für den System Controller selbst reserviert. Der Controller wurde in dem Experiment "Neutrale Resonanzen" am Großbeschleuniger in Serpuchow, UdSSR, zum ersten Mal eingesetzt.

## CAMAC System Controller for Process Computer Type Telefunken TR86

### Abstract

A System Controller capable of driving a full CAMAC Branch as interface to a Telefunken computer type TR86 is described. The device can handle single operations, four different modes of block transfers and fast autonomous transfers using cycle stealing techniques. There are 8 Interrupt and 16 DMA-channels available. Each of up to seven crates is connected to an interrupt channel; the last channel is for use of the System Controller. The device was tested during start up and run time of the experiment 'Neutrale Resonanzen' at the accelerator in Serpuchow, USSR.

Inhalt

Seite

1.	Einleitung	4
2.	Nahtstelle zum Telefunken-Rechner TR 86	6
3.	Hauptfunktionsgruppen des System Controller	10
3.1.	Rechnerperipherie	10
3.2.	Branch Driver	14
3.3.	Interne Logik	16
3.3.1.	Modul "Befehlsregister"	16
3.3.1.1.	Bildung der Crate-Adresse	18
3.3.2.	Modul "Lesebuffer"	19
3.3.3.	Modul "Timing-Generator"	19
3.3.3.1.	Hauptzustands-Generator	24
3.3.3.2.	Enable System Controller	27
3.3.4.	Modul "Control- und Statusregister"	27
3.3.5.	Modul "GL-Register und Interrupt-Behandlung"	29
3.3.6.	Modul "Auto Inkrement N.A."	32
3.3.6.1.	Modul-zu-Modul-Transfer	33
3.3.7.	Modul "DMA-Befehlsregister"	34
3.3.7.1.	Autonomer Transfer in Eingabe-Richtung	34
3.3.7.2.	Autonomer Transfer in Ausgabe-Richtung	36
3.3.8.	Modul "Control Unit"	37
4.	Funktionsabläufe	40
4.1.	Befehls-Operation	40
4.1.1.	Dialog-Rechner - System Controller ohne Operand	43
4.1.2.	Dialog-Rechner - System Controller mit Operand	43
4.1.3.	Einadreß-Lesebefehle	45
4.1.4.	Einadreß-Schreibbefehle	48

	Seite
4.1.5. Befehle ohne Oyerand	48
4.1.6. Blocktransfer IN, IA Mode	48
4.1.7. Blocktransfer Q Mode	52
4.1.7.1. Blocktransfer im Adress Scan Mode	53
4.1.7.2. Blocktransfer im Repeat Mode	53
4.1.7.3. Blocktransfer im Stop Mode	57
4.2. Modul-zu-Modul-Transfer	57
4.3. Alarm-Verarbeitung	61
4.3.1. Interrupt-Behandlung	63
4.3.2. Autonome Transfers (Eingabe)	63
5. Testmöglichkeiten mit dem System Controller	67
5.1. Benutzung des BQ-Signals	67
5.2. Lesen des Statusregisters	67
5.3. Tests über Frontplatten	68
Literatur	68
Anhang: Zusammenstellung benutzter Signalnamen, Subadressen etc.	69
1. Subadressen und Funktionen	69
2. Befehlsliste	70
3. Liste der verwendeten Abkürzungen und Signalnamen	72

## 1. Einleitung

Der im vorliegenden Bericht beschriebene Controller zum direkten Anschluß eines CAMAC Branch Highway ist ein Nachfolger des im KFK-Bericht Nr. 1412 [1] eingehend behandelten Controllers für das CALAS-System. Gegenüber diesem Vorgänger ist der Controller auf Grund von Veränderungen der CAMAC-Spezifikationen (siehe EUR 4100 e, 1972), [2] der Anforderungen vom Experiment und auch wegen der mit dem CALAS-Controller gemachten Erfahrungen an einigen Stellen verbessert worden.

So verfügt er über insgesamt vier verschiedene Methoden des Blocktransfers: die drei mit Address-Scan (ADS), Repeat-Mode (RPM) und Stop Mode (STM) von den neuen CAMAC-Spezifikationen vorgesehenen sowie den IN, IA-Modus, der dem ADS-Betrieb bei vielen gleichartigen Modulen durch die Vermeidung von Leer-Transfers überlegen ist.

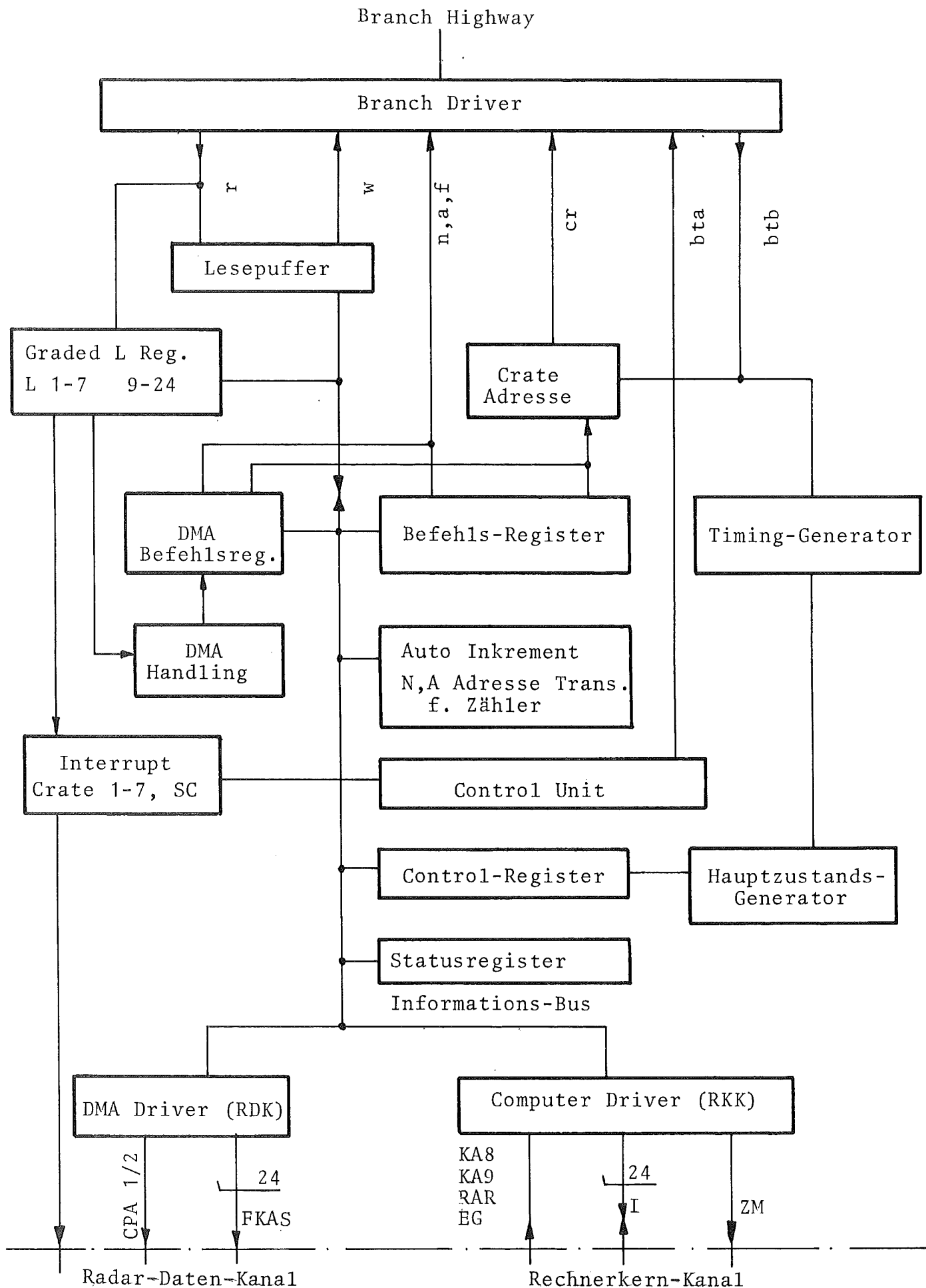
Neu ist auch der Anschluß des von der Firma Telefunken entwickelten sog. Radar-Daten-Kanales (RDK). Auf Wunsch können per Lesebefehl herangeschaffte Daten auf diesen Kanal geschaltet und ohne Programm-Assistenz im Kernspeicher an Stellen abgelegt werden, die im Kanal selbst bestimmt werden. Bei dieser Betriebsart können die Zähler für die Anzahl der Transfers in N und A gekettet und bei DMA-Block-Übertragungen daher Blöcke bis zu einer maximalen Länge von 512 ( $2^9$ ) Wörtern bewältigt werden. Der Controller macht im übrigen von dem Konzept der Bus-Leitungen intern Gebrauch zur Verbindung seiner verschiedenen Funktionsblöcke. Der äußere Aufbau erfolgt in einzelnen modularen Einheiten, die für sich prüfbar sind und in ihrer Logik eine gewisse Geschlossenheit zeigen (Funktionseinheiten), siehe Fig. 1.

Der System Controller besteht grob gesehen aus drei Funktionseinheiten (s. Fig. 1). Die erste ist der sogenannte Branch Driver, also der Ausgang zum Branch Highway. Der Branch Driver ist zum größten Teil festgelegt durch die Spezifikationen des Euratomberichtes EUR 4600 e [3], daher soll hier nichts mehr darüber gesagt werden. Die zweite Funktionseinheit ist die "Interne Logik",

---

eingegangen am 24. 9. 1973





TR 86

Fig. 1 CAMAC System Controller für Telefonken-Rechner TR86 mit Radar-Daten-Kanal, Blockschaltbild

in der die einzelnen Fähigkeiten des Gerätes verwirklicht werden. Im Idealfalle ist auch dieser wichtige Teil noch unabhängig von dem verwendeten Rechnertyp. Die folgenden Ausführungen beschäftigen sich hauptsächlich mit der "Internen Logik". Die letzte Funktionseinheit heißt Rechnerperipherie. In diesem Teil werden die Eigenschaften des jeweiligen Rechners berücksichtigt.

Die Darstellung dieses Berichtes lehnt sich soweit wie möglich und sinnvoll an KFK 1412 an. Dieses Vorgehen erzeugt zwar eine gewisse Redundanz, die für den potentiellen Anwender nicht unbedingt erforderlich wäre, andererseits jedoch ist es eine sehr geeignete Form für Reparatur, Service und die Erstellung dieser notwendigen Unterlage.

## 2. Nahtstelle zum Telefunken-Rechner TR 86

In dem Schema nach Fig. 2 (rechts) und in Tabelle 1 sind alle Verbindungen vom Rechner zum System Controller eingetragen. Die Bedeutung der einzelnen Leitungen sei kurz besprochen.

### a) für den Rechnerkern-Kanal RKK

#### IB (Informations-Bus)

Über die insgesamt 24 bidirektionalen Informations-Busleitungen IB werden sowohl CAMAC-Befehle (CR N A F) als auch Daten geschickt. Daraus ergibt sich, daß im System Controller auch in der kleinsten Ausbaustufe ein Befehlsregister vorhanden sein muß, denn bei Schreib-Operationen müssen der CAMAC-Befehl CRNAF und die zu schreibenden Daten gleichzeitig auf dem Branch Highway erscheinen. Daher muß der CAMAC-Befehl zwischengespeichert werden.

Fig. 2: Übersicht über die Nahtstellen des System Controller

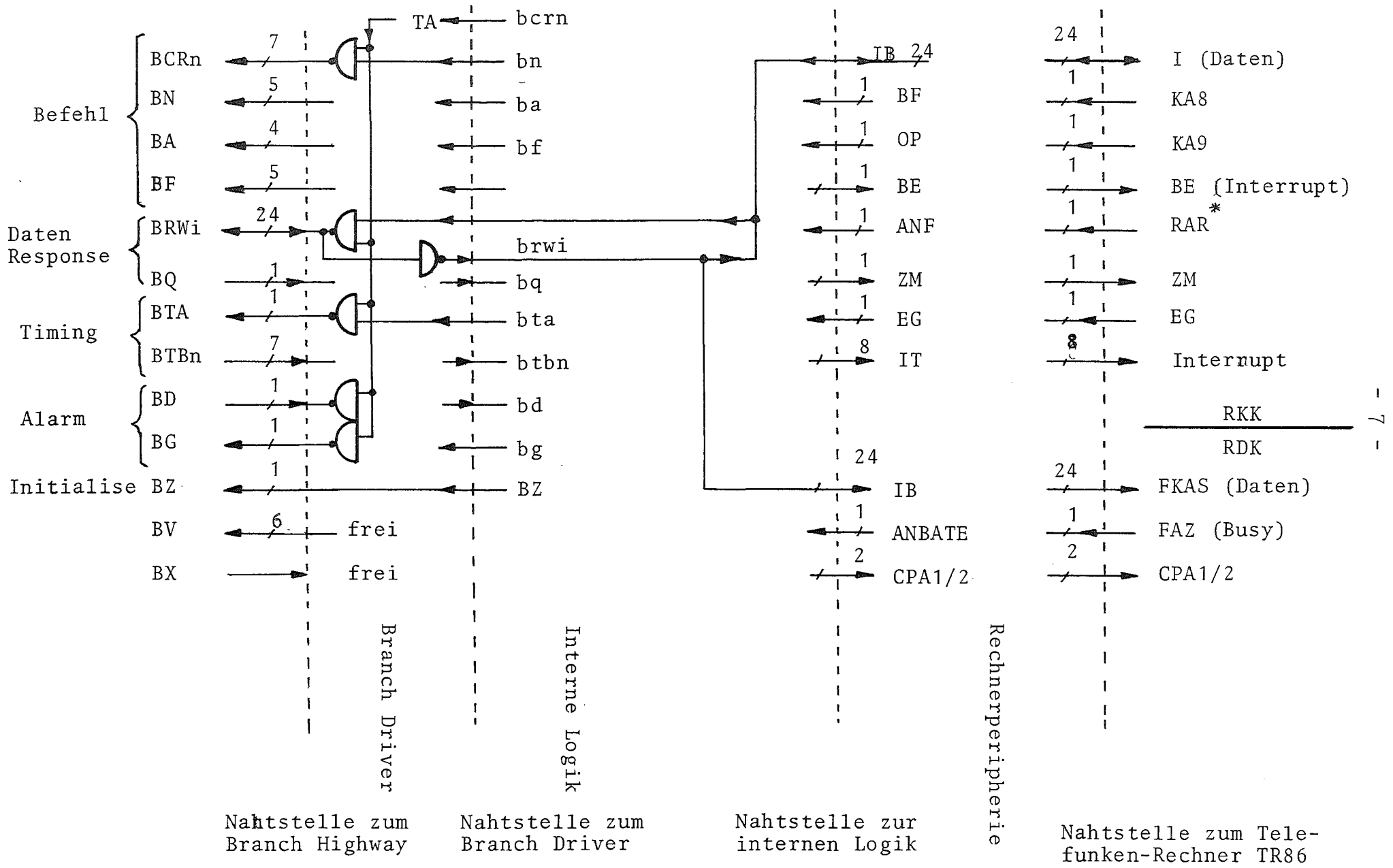


Tabelle 1: Leitungen der Nahtstelle zum Telefunken-Rechner TR 86

Benennung	Bedeutung	Anzahl der Leitungen
IB	Informations-Bus (bidirektional, für CAMAC-Befehle und Daten vom/zum Rechnerkern-Kanal RKK [I]) (nur Dateneingabe [FKAS] Radardatenkanal RDK)	24
BF	CAMAC-Befehl (Kanal-Adresse 8, KA8)	1
OP	Operand, CAMAC-Daten (Kanal Adresse 9, KA9)	1
BE	Blockende	1
ANF	Anforderung (Rechneraufruf RAR*)	1
ZM	Zeichenmeldung	1
EG	Eingabe in den Rechner ( $\overline{EG}$ = Ausgabe)	1
ANBATE	Anforderung für Block autonomer Transfers in Eingabe-Richtung (FAZ)	1
CPA1/2	Clock Puls für RDK-Datenübernahme	2
IT	Interrupt	max. 9

BF (Befehl)

Die Leitung steht auf 0 V (das entspricht logisch "1"), wenn es sich bei der Information auf den IB-Leitungen um einen CAMAC-Befehl handelt.

OP (Operand)

Die Leitung führt "1", wenn es sich bei der Information auf den IB-Leitungen um Daten handelt. Die beiden Leitungen BF und OP sind zueinander antivalent.

BE (Blockende)

Das Signal BE wird vom System Controller gebildet und bedeutet das Ende eines Blocktransfers. Es darf frühestens zu Beginn des letzten Transfers des Blockes erscheinen, spätestens mit der Rückmeldung des letzten Transfers.

ANF (Anforderung) RAR\*

Der 0 → 1-Übergang dieses vom Rechner kommenden Signals bedeutet den Beginn einer programmgesteuerten Operation (Command Operation CO).

EG (Eingabe)

Diese Leitung führt "1", wenn eine Eingabe in den Rechner erfolgt.

ZM (Zeichenmeldung)

Das Signal meldet dem Rechner den Vollzug der Operation. Das Signalarpaar ANF (bzw. RAR\*) und ZM verwirklicht ein Ping-Pong- oder Hand-Shake-Timing-System.

IT Programmunterbrechung des Rechners

b) Für den Radar-Daten-Kanal RDK

Auf den 24 IB- bzw. FKAS-) Leitungen findet der Datentransport im Cycle-Stealing-Verfahren direkt zum Kernspeicher statt. Die IB-Leitungen werden hier also nur zur Eingabe verwendet.

CPA1/2 (Clockpulse zur Informationsübernahme)

ANBATE (Anforderung Block autonomer Transfers Eingabe)

Wenn der Rechner autonome Transfers zulassen will, so setzt er diese Leitung auf "1". Den wirklichen Zeitpunkt des Transfers kann er nicht bestimmen, sondern er wird ihm durch CPA1/2 mitgeteilt.

### 3. Hauptfunktionsgruppen des System Controller

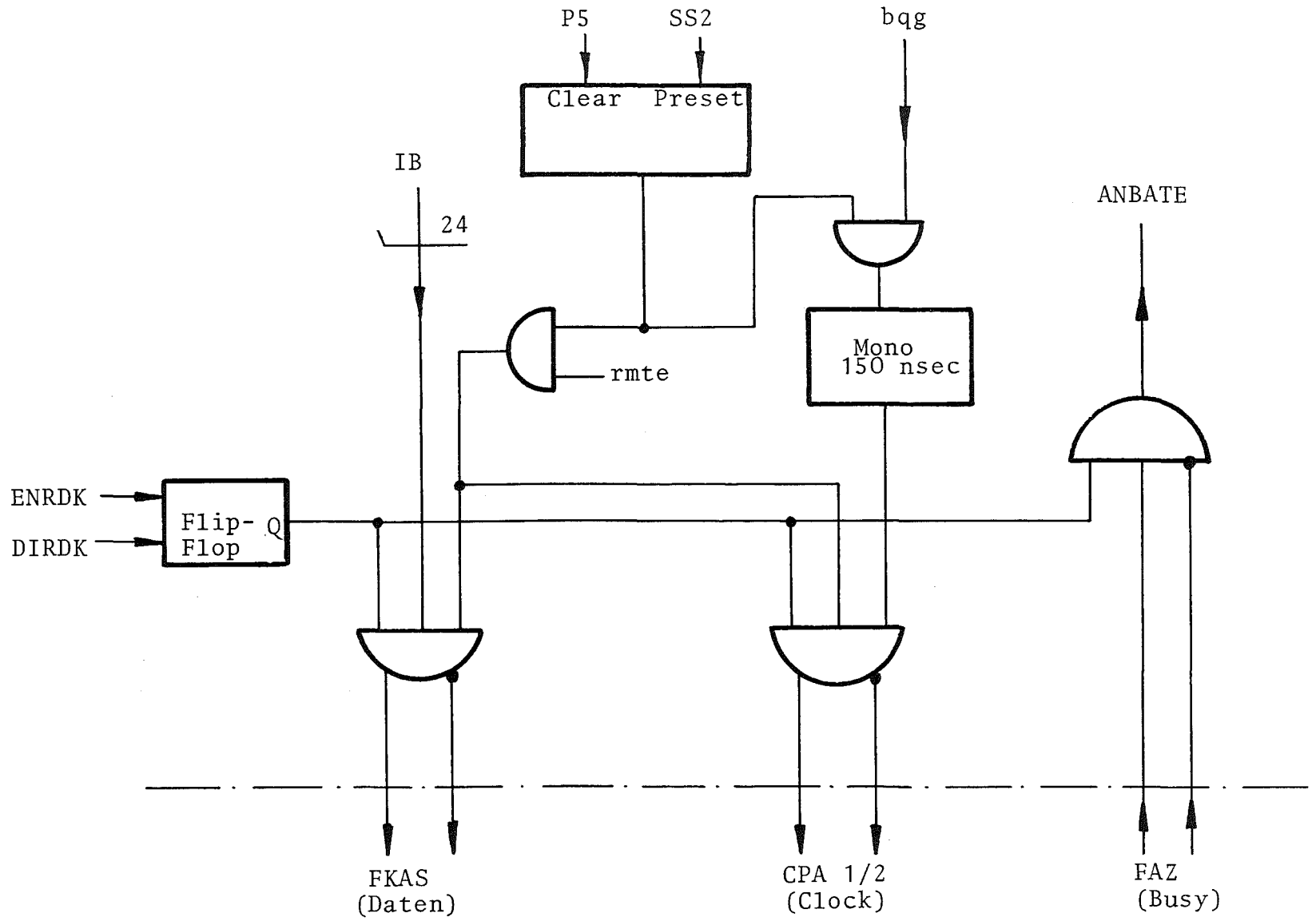
Wie schon eingangs erwähnt, besteht der System Controller aus drei logisch klar abgrenzbaren Funktionsgruppen, es sind

- a) Rechnerperipherie
- b) Branch Driver
- c) Interne Logik

#### 3.1. Rechnerperipherie

Die Funktionsgruppe "Rechnerperipherie" ist im vorliegenden Fall natürlich eine Peripherie zu den TR86-Kanalwerken. Hier werden nur die Signale auf den in Kapitel 2 aufgezählten Leitungen geschaltet und verstärkt, sonst enthält diese Funktionsgruppe keine weitere Elektronik (siehe Fig. 3 und 4).

Fig. 3: DMA-Driver



Radar-Daten-Kanal

Rechnerkern-Kanal

Fig. 4: Computer Driver

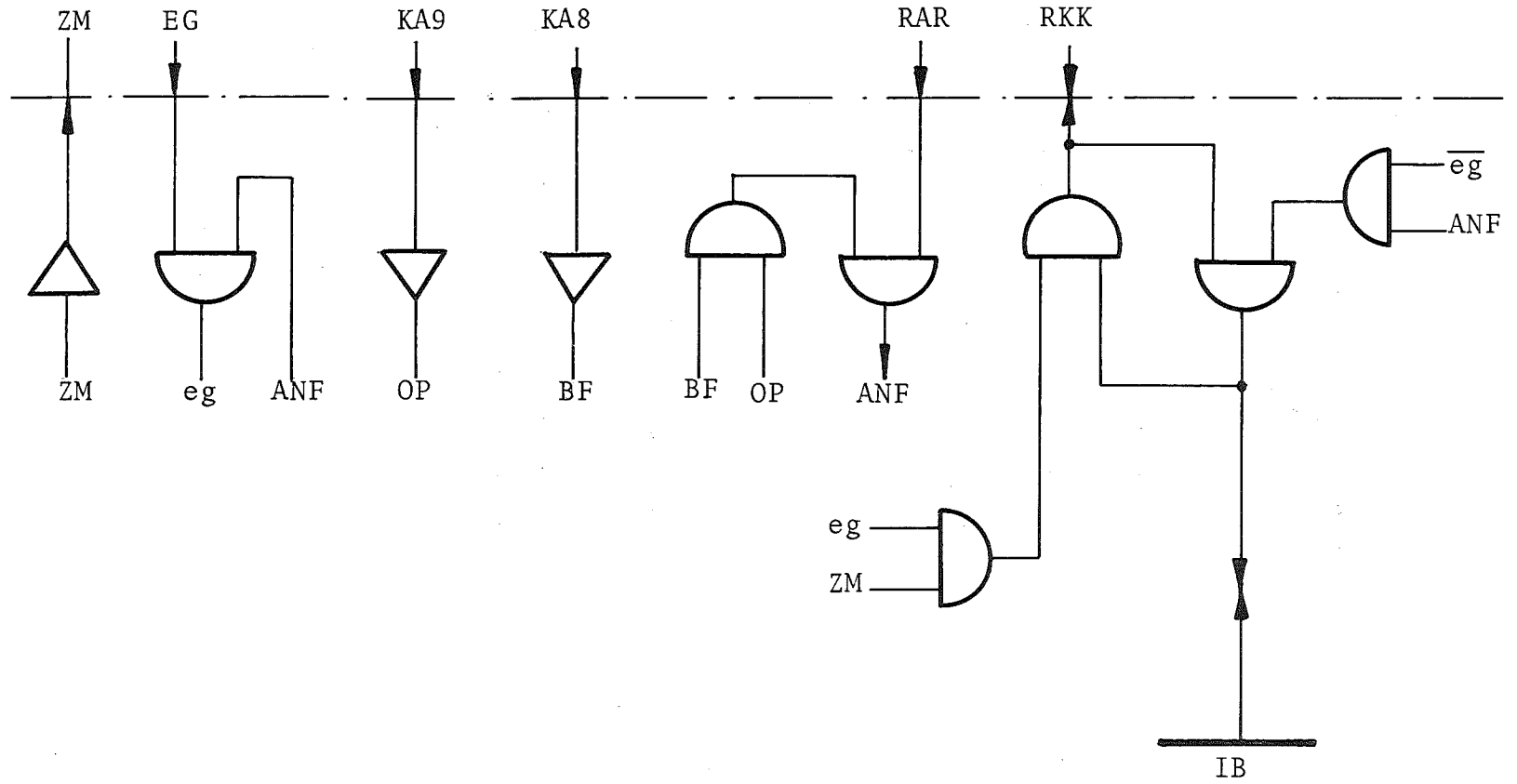




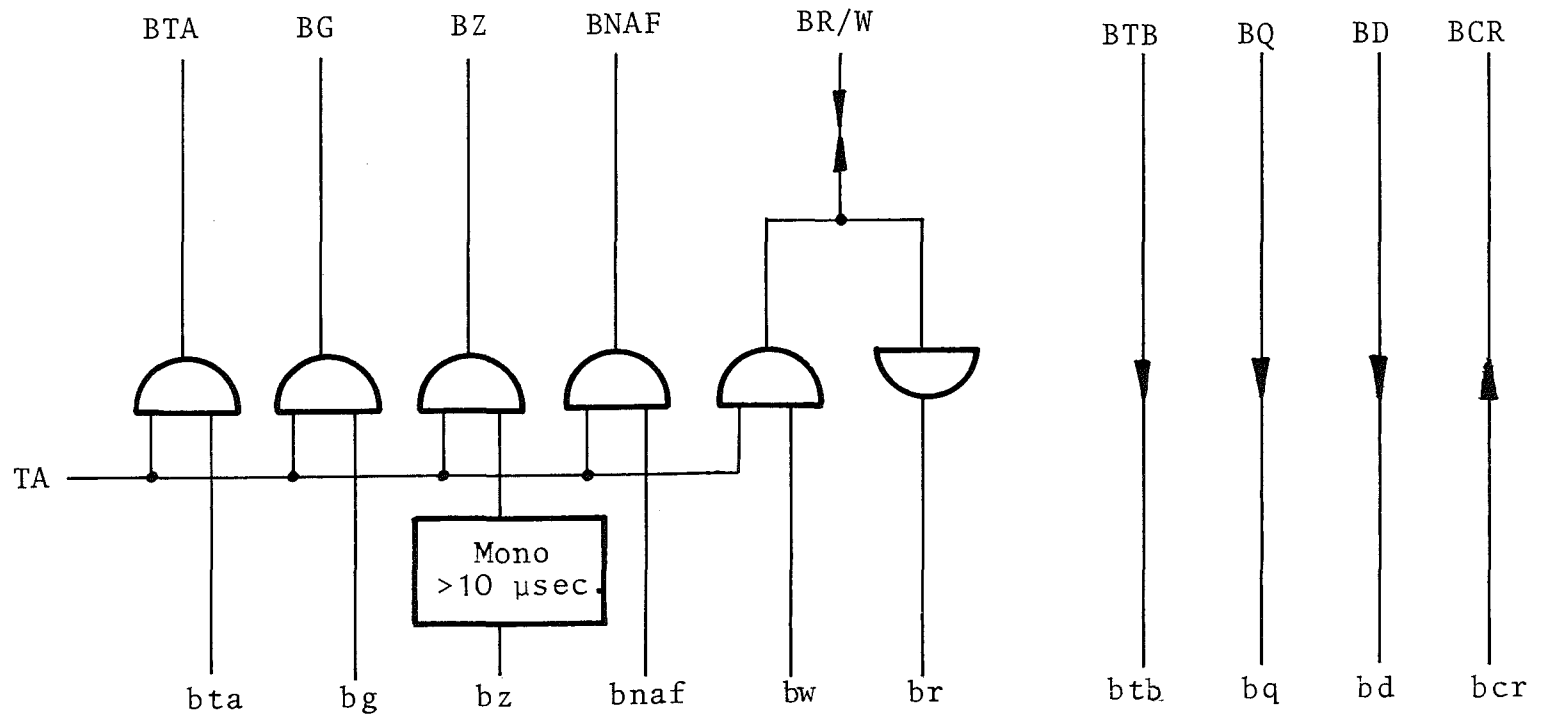
Tabelle 2: Leitungen der Nahtstelle zum Branch Highway

Benennung	Bedeutung	Anzahl der Leitungen
BCRn	Crate-Adresse	7
BN	Stationsnummer	5
BA	Subadresse	4
BF	Funktionscode	5
BRWi	Lese-Schreib-Leitungen	24
BTA	Timing-Signal	1
BTBn	Antwort-Signale von den Rahmensteuerungen	7
BQ	Response	1
BD	Demand (Sammel-Alarm von den Crates)	1
BG	Anforderung des Alarm-Musters	1
BZ	Grundzustand herstellen	1
BX	Command Accepted	1
BV	Reserve	7
BSC	Cable Screen	<u>1</u>
		66
		==

### 3.2. Branch Driver

Die Funktionsgruppe Branch Driver enthält ausgangsseitig die Terminationen und die bereits durch die Beschreibung des Branch Highway [3] bekannten und in Tabelle 2 noch einmal zusammengestellten Leitungen. Die Leitungen des Branch Highway werden aus Gründen der Störsicherheit auf Kosten eines in Kauf genommenen höheren Energieverbrauches an beiden Enden nahezu angepaßt abgeschlossen. Die Signale auf den Leitungen des Branch Highway unterscheiden sich von den in Fig. 2 (S. 7, links) erkennbaren gleichnamigen Leitungen mit kleinen Buchstaben dadurch, daß sie verstärkt sind und im Pegel umgekehrt wurden. Außerdem werden zum Ausgleich differentieller Laufzeitunterschiede auf den einzelnen Leitungen bis auf wenige Ausnahmen fast alle ausgehenden Signale etwa 400 ns vor dem Timing-Signal BTA auf den Branch Highway gegeben. Das wird durch Gaten mit einem Vorläufer von BTA, nämlich TA, erreicht. Auf diese Weise wird sichergestellt, daß alle Leitungen eingeschwungen sind, wenn die Rahmensteuerungen durch die Ankunft von BTA aktiviert werden (s. Fig. 5).

Fig. 5: Branch Driver



### 3.3. Interne Logik

#### 3.3.1. Modul Befehlsregister

Ein Pufferspeicher für Befehle ist primär dadurch nötig, daß es nur einen Satz von 24 bidirektionalen Leitungen gibt, über die Befehle und Daten geschickt werden. Befehle werden zuerst geschickt und zwischengespeichert. In Fig. 6 sind die einzelnen Bitgruppen des CAMAC-Befehls CRNAF zu erkennen. Das Befehlsregister (BR) hat eine Kapazität von 20 Bit. Der eigentliche Befehl besteht nur aus 18 Bit, nämlich

CR	4
N	5
A	4
F	<u>5</u>
	18

Die übrigen 2 Bit werden für Steuerzwecke gebraucht. So muß das Bit 19, MOP (mit Operand), gesetzt werden bei jedem Befehl, der von einem Operanden begleitet wird, d. h.

MOP = 1 bei jedem Schreib- oder Lesebefehl

MOP = 0 bei jedem Steuerbefehl

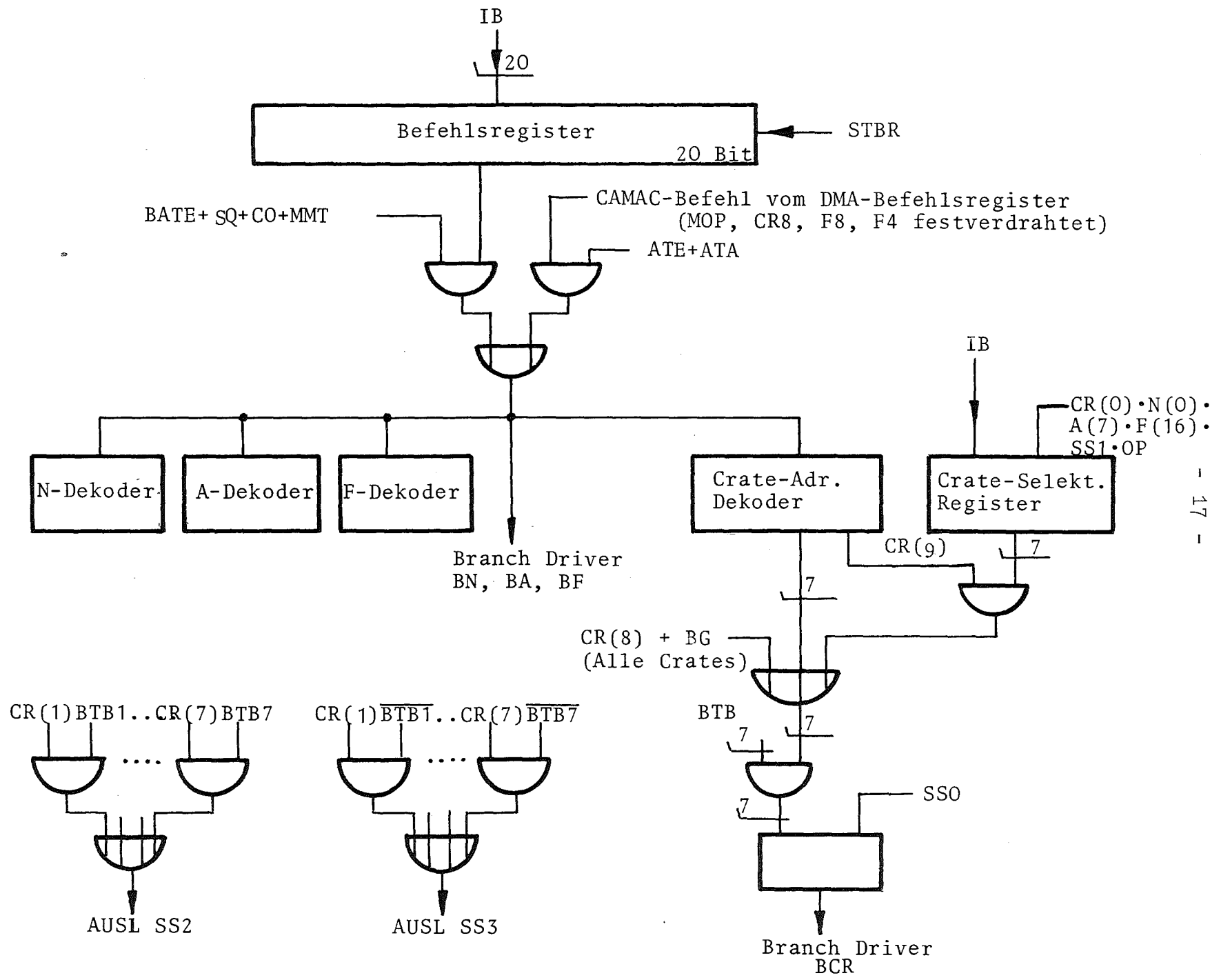
Das Bit 20, E (Eingabe), entscheidet, ob eine Eingabe vom CAMAC Branch zum Rechner oder eine Ausgabe erfolgen soll.

E = Eingabe      $\bar{E}$  = Ausgabe

Dieser Befehlsspeicher dient bei allen Betriebsarten des System Controller außer bei autonomen Transfers zur Bereitstellung des CAMAC-Befehles. Bei autonomen Transfers wird der CAMAC-Befehl aus dem Modul "DMA Befehlsregister" abgerufen.

Die für den System Controller benötigten Adressen und Funktionscodes werden ebenfalls in diesem Modul dekodiert.

Fig. 6: Befehlsregister



### 3.3.1.1. Bildung der Crate-Adresse

In einem CAMAC-Branch können maximal 7 Rahmen sein, die mit einzelnen Adreß-Stichleitungen angesprochen werden, so daß zur Bildung der Crate-Adresse eigentlich 3 Bit genügen würden. In den 3 Bits bcr1, bcr2 und bcr4 des Befehlsregisters lassen sich 8 Adressen unterbringen, 7 für die CAMAC-Crates und Cr(0) für den System Controller (vgl. Fig. 6, S. 17). Das vierte Crate-Adreß-Bit wird für Multi-Crate-Adressierung gebraucht, ähnlich wie bei der Vielfach-Adressierung der Stationen durch die Rahmensteuerung [4]. So werden durch bcr(8) alle vorhandenen Crates adressiert und durch bcr(9) alle vorgewählten. Das Muster der vorgewählten Crates wird in einem eigenen vom Rechner setzbaren Register, dem Select Crate Register (SCR), aufbewahrt. Außerdem gibt es noch eine Einrichtung, die zu Beginn einer Operation feststellt, ob ein angewähltes Crate augenblicklich dialogfähig ist. Das geschieht mit Hilfe der BTB-Leitungen, denn nur ein Crate, welches vorhanden ist, seine Stromversorgung eingeschaltet hat und on-line ist, kann seine BTB-Leitung auf 0 V ziehen.

Bei Alarm-Operationen muß der System Controller durch das Signal BG den Alarmzustand des gesamten Systems erfragen. Die BG-Operation wirkt - ohne daß jetzt auf Einzelheiten eingegangen werden soll - wie eine Lese-Operation an allen on-line Crates. Das bedeutet, daß der System Controller in diesem Falle die Crate-Adressen aller on-line-Crates selbständig zu bilden hat.

Aus Fig. 6 ist zu sehen, daß sowohl bei rechnerkontrollierten Operationen als auch bei Alarm-Verarbeitung (BG-Signal) der Zustand der BTB-Leitungen zur Taktzeit SSO des Timing-Generators getestet und festgehalten wird. Beim erfolglosen Dialogversuch mit einem Crate wird ein Signal "Time Out" (TO) erzeugt und im Statusregister des System Controller angezeigt. Dies erleichtert dem Rechner die Fehlerdiagnose.

### 3.3.2. Modul "Lesebuffer"

Im einfachsten Falle können die Informations-Leitungen vom Rechner zum Transport von Daten auf die Treiber der BRW-Leitungen gegeben werden. Da der System Controller so weit ausgebaut ist, daß er Modul-zu-Modul-Transfer bewältigen kann, wird nach Fig. 7 ein Lesebuffer-Register als Zwischenspeicher für Lesedaten benötigt. Aus diesem Register heraus werden in dieser Betriebsart die zuvor eingelesenen Daten an den empfangenden Modul gesendet.

### 3.3.3. Modul "Timing-Generator"

In diesem Modul werden die für den ordnungsgemäßen Betrieb des System Controller benötigten Zeitsignale erzeugt (s. Fig. 8). Es handelt sich um vier Uhrimpulse (SS0 bis SS3), die sich nicht überlappen. Der Generator kann durch einen Betriebsarten-Wahlschalter in verschiedene Moden gezwungen werden: "Run", "OP" (Einzeloperationen) und "Step". In der Betriebsart "Run" hat der Rechner bzw. das CAMAC-System (bei Alarm-Operationen) die volle Kontrolle über den zeitlichen Ablauf der Operationen. Eingriffe von Hand sind in den Stellungen "OP" und "Step" möglich. Bei "OP" läuft nach Druck auf eine Taste eine Operation komplett ab. Bei "Step" wird ein Ablauf in seine einzelnen Phasen zerlegt; zur vollständigen Abwicklung der Operation muß die Taste 4mal gedrückt werden. Bei beiden Betriebsarten bleibt der Rechner mit dem System verbunden, der zeitliche Ablauf jedoch dem Operator überlassen, der dadurch in Zusammenhang mit dem Anzeigefeld eine gute Testmöglichkeit erhält. Über die Verwendung der einzelnen Uhrimpulse des Generators gibt Tabelle 3 Auskunft. Die Zuordnung der internen Uhrimpulse des System Controller zu dem extern im Branch Highway verwendeten Timing-Paar BTA/BTBn ist in Fig. 9 dargestellt.155

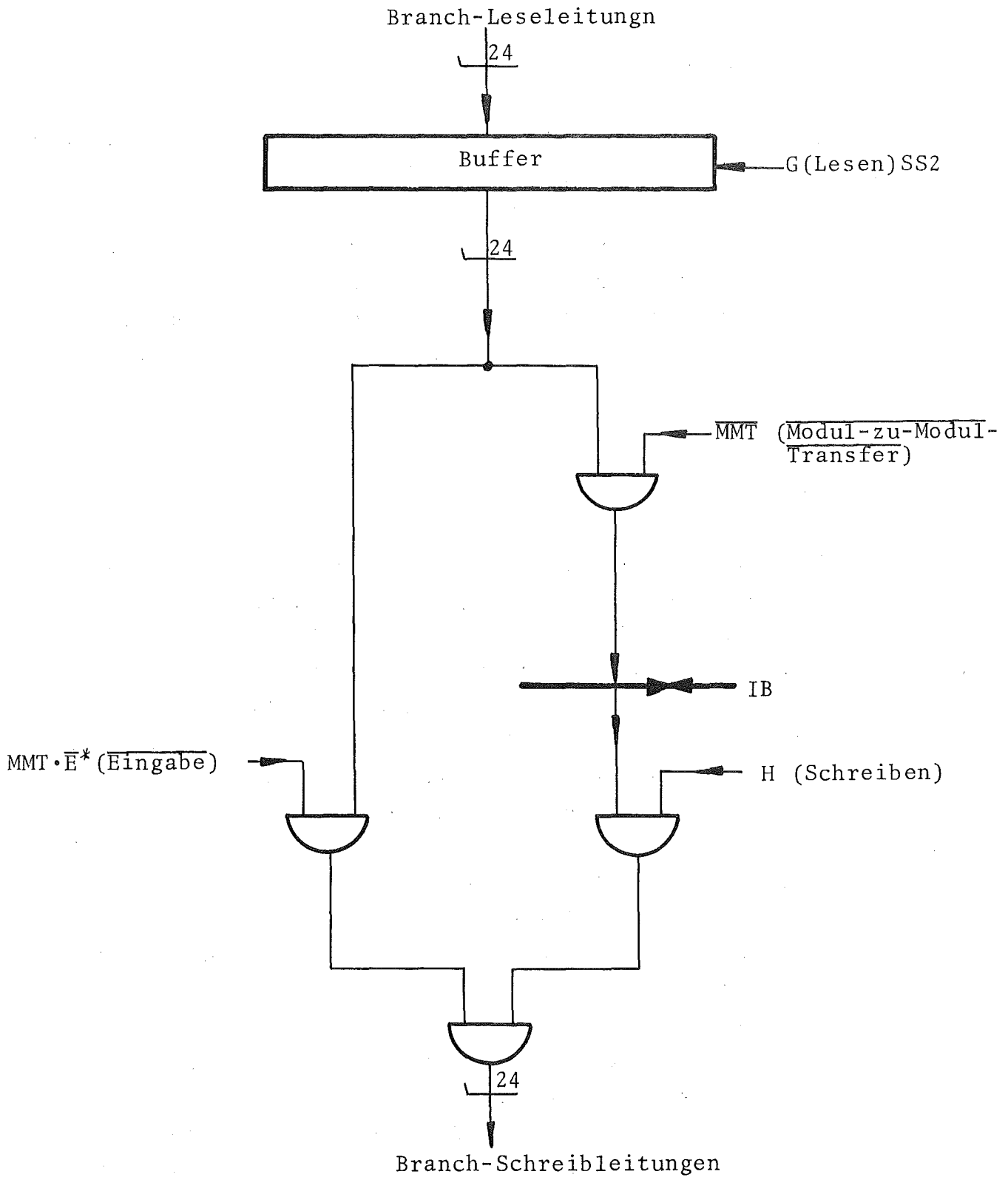


Fig. 7: Lesebuffer



Fig. 8: Timing-Generator

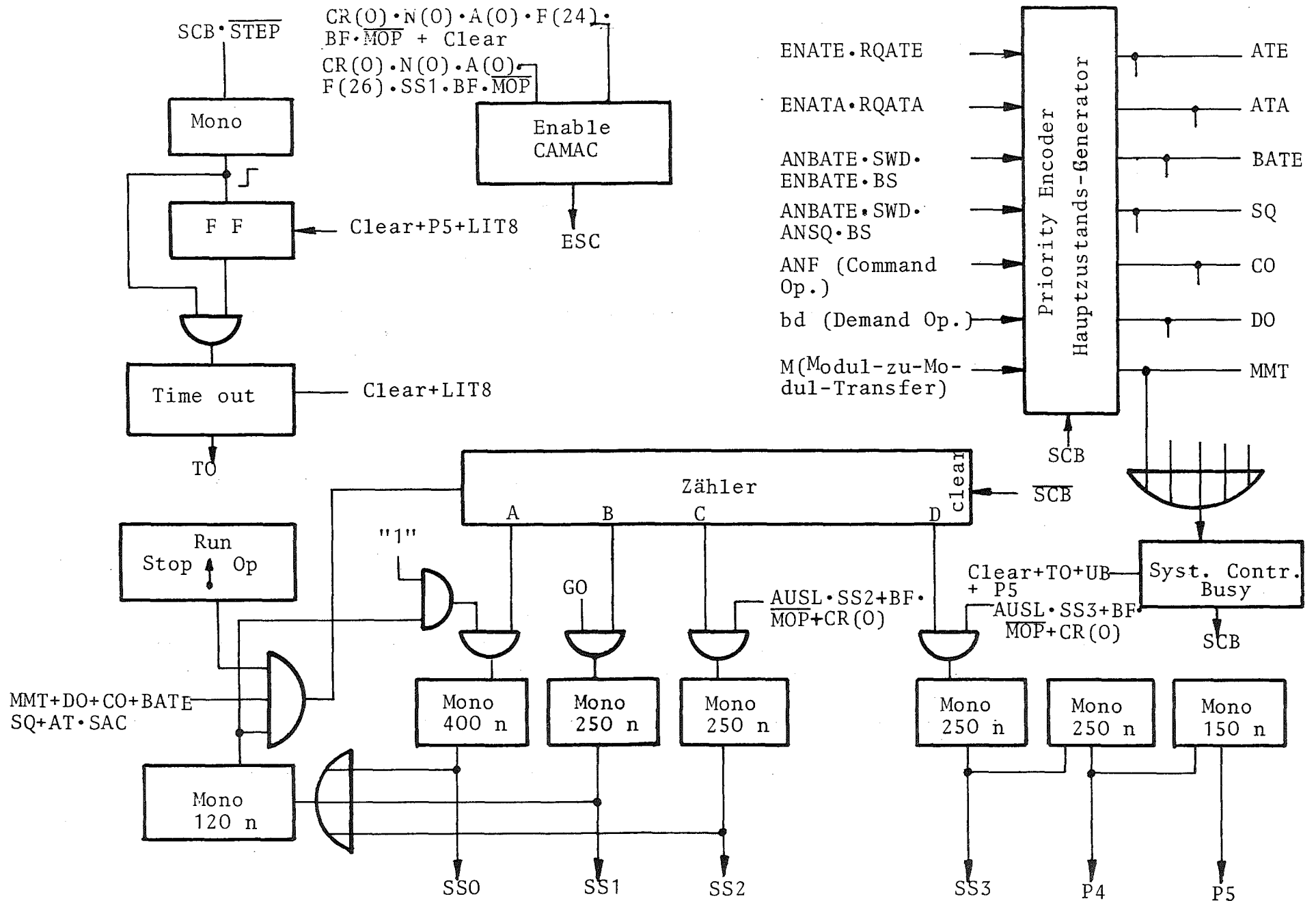
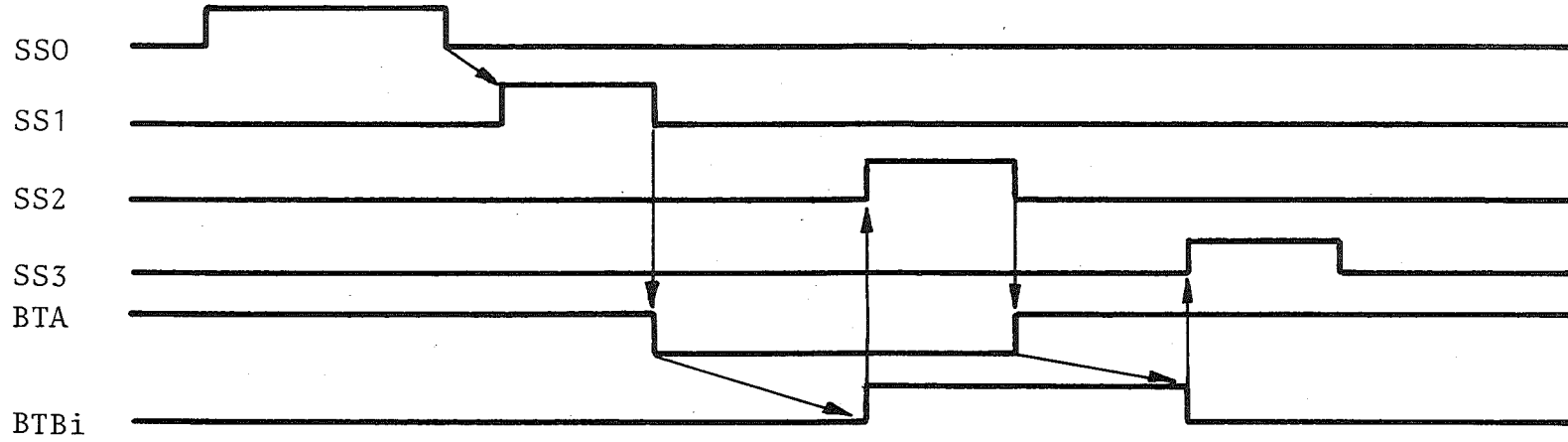
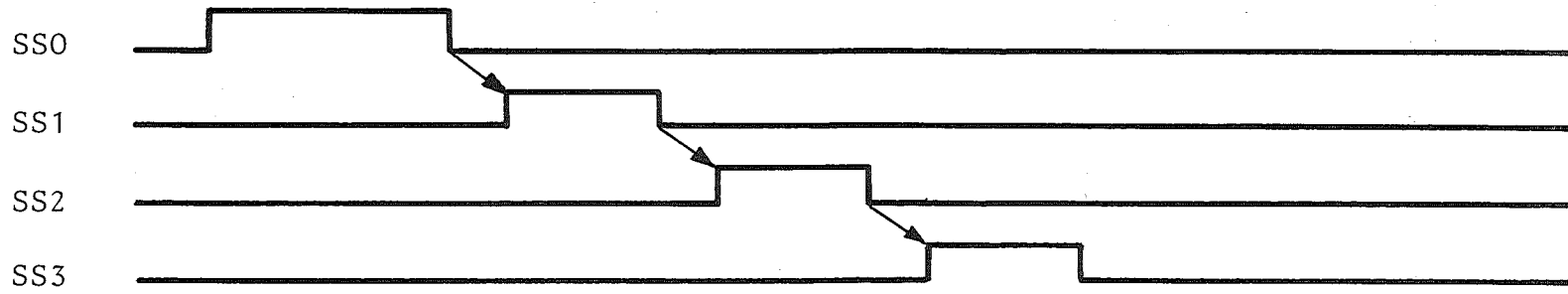


Fig. 9: Prinzipielles Timing-Diagramm



a) Verkehr des System Controller mit dem CAMAC-System



b) Schneller Dialogverkehr des Rechners mit dem System Controller

Tabelle 3: Verwendung der Uhrimpulse des Timing-Generators

	Verwendung
SS0	Übernahme der Befehle ins Befehlsregister; Test der Dialogfähigkeit des (der) ange- wählten Crate(s); automatische Bildung der Crate-Adressen bei Alarm-Operationen;
Rückflanke	Einspeichern der Crate-Adressen;
Rückflanke	Absenden von Befehlen (und Schreibdaten)
SS1	Laden von internen Registern
Rückflanke	Erzeugung des Timing-Signals BTA
SS2	Adreßfortschaltung A, N;
Rückflanke	Rücknahme des Timing-Signals BTA;
Rückflanke	Erzeugung der Zeichenmeldung
SS3	Rücknahme der Befehle (nach Rücknahme von BTB durch die Crate Controllers); Beendigung der Operation

Nun betrifft nicht jede Operation des Rechners den Branch, daher wird nicht bei jedem Rechnerzugriff BTA erzeugt, und folglich kann auch kein BTB erwartet werden. Dieser Umstand wird ausgenutzt und in diesen Fällen der interne Zyklus des System Controller mit maximaler Geschwindigkeit gefahren. Der Dialog des Rechners mit dem System Controller ist also sehr schnell (siehe dazu das Diagramm in Fig. 10).

Bei Operationen mit dem Branch wird BTA erzeugt und entsprechend wartet der System Controller auf BTB, ehe er seinen Zyklus fortsetzt. Beim gleichzeitigen Verkehr mit mehreren Crates muß der System Controller abwarten, bis alle Crates mit BTB geantwortet haben. Es gilt allgemein Gleichung (1)

$$(\overline{BTB1} + \overline{BCR1}) \cdot (\overline{BCR2} + \overline{BTB2}) \dots (\overline{BCR7} + \overline{BTB7}) = 1 \quad \text{Gl. (1)}$$

Wenn die Bedingung nach Gl. (1) erfüllt ist, erzeugt der System Controller seinen Uhrimpuls SS2 und wartet dann wieder mit der Erzeugung von SS3, bis die Bedingung nach Gl. (2) erfüllt ist, die bedeutet, daß alle Crates ihre Meldung BTB wieder zurückgenommen haben.

$$(\overline{BCR1} + BTB1) \cdot (\overline{BCR2} + BTB2) \dots (\overline{BCR7} + BTB7) = 1 \quad \text{Gl. (2)}$$

Mit Hilfe der Gleichungen (1) und (2) adaptiert sich das System automatisch an variierende Längen des Branch Highway, wobei nie mehr Zeit verbraucht wird als unbedingt nötig.

### 3.3.3.1. Hauptzustands-Generator

Der wichtigste Teil der Steuerlogik des System Controller ist der sogenannte Hauptzustands-Generator (siehe Fig. 8, S. 21), der die Zyklen des System Controller nach einem fest eingebauten

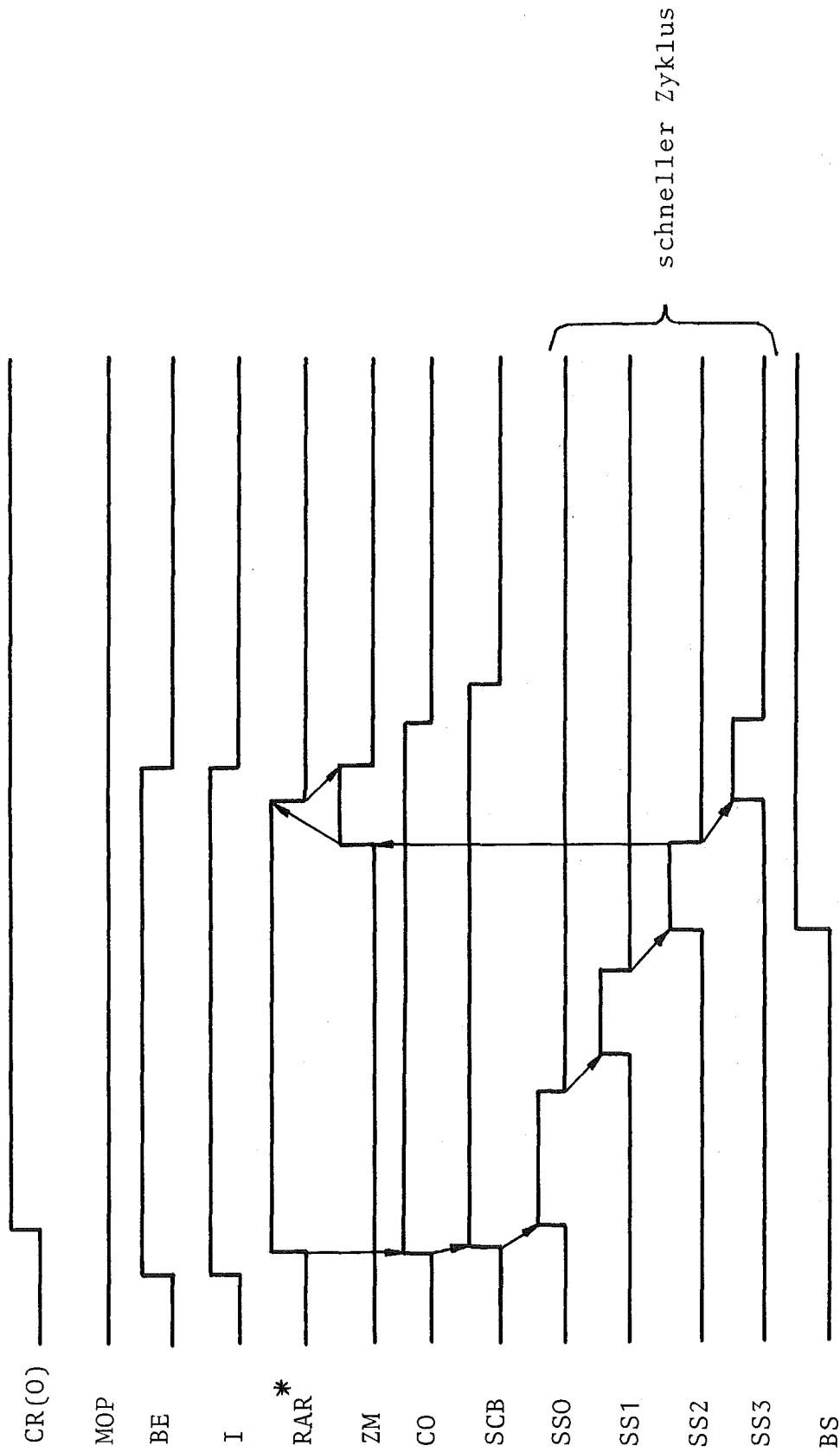


Fig. 10: Timing-Diagramm für Dialog Rechner System Controller ohne Daten

Prioritätssystem verteilt. Der Generator kann sich in Ruhe befinden oder in einem (und nur in einem) der sieben Hauptzustände. Für die Hauptzustände des System Controller wurden folgende Prioritäten festgelegt:

- 1 Autonomer Transfer in Eingabe-Richtung (ATE)
- 2 Autonomer Transfer in Ausgabe-Richtung (ATA)
- 3 Block Autonomer Transfers in Eingabe-Richtung (BATE)
- 4 Sequentielle Operationen im Q-Mode (SQ)
- 5 Befehls-Operation (Command Operation) (CO)
- 6 Alarm-Operation (Demand Operation) (DO)
- 7 Modul-zu-Modul-Transfer (MMT)

Der Hauptzustand kann nach jedem Zyklus wechseln. Eine Ausnahme bilden die schnellen autonomen Transfers bei der Eingabe. Wenn gleichzeitig mehrere Anforderungen auf solche Transfers vorliegen, so müssen diese erst sämtlich nach einem Prioritätsschema abgearbeitet worden sein, ehe sich der System Controller in einen neuen Hauptzustand bringen läßt. Der Grund dafür liegt in folgender Überlegung:

Moderne schnelle ADC's haben Konversionszeiten zwischen 10 und 100  $\mu$ s. Der Abtransport der Daten muß also vorrangig und sehr schnell geschehen, damit keine Meßwerte verloren gehen. Da mehrere ADC's in einem System vorhanden sein können, muß ihnen die höchste Dringlichkeit eingeräumt werden. Daher darf der Abtransport der Daten aus bis zu sechzehn schnellen Quellen nicht durch andere zwischengeschobene Operationen verzögert werden. Die eingehende Beschreibung des Ablaufes aller Funktionen erfolgt im Kapitel 4.

Wenn der sehr unwahrscheinliche, aber immerhin denkbare Fall eintreten sollte, daß konkurrierende Ansprüche gleichzeitig zwei oder mehr von den sieben Hauptzuständen erzeugen, so wird daraus ein Signal "Unerlaubte Betriebsart" (UB) abgeleitet. Erzeugt wird dieses Signal im Modul "Control- und Statusregister". Dieses Signal löscht

die Hauptzustände und gibt den System Controller wieder frei. Jetzt kann sich der Hauptzustand etablieren, der die höchste Priorität besitzt. Um zu verhindern, daß der System Controller von einem Hauptzustand blockiert ist, der nicht abgearbeitet werden kann, (Beispiel: BTA wird generiert, aber BTB vom Crate Controller bleibt aus oder wird nicht zurückgenommen), wird ein "Time Out" (TO) erzeugt und in das Statusregister eingeschrieben. Dieses Signal TO ersetzt das BTB-Signal, so daß der fehlerhafte Zyklus ablaufen kann. Gleichzeitig ist dafür gesorgt, daß ein Programm-Unterbrechungssignal IT8 erzeugt wird. Es wird Programm-Assistenz angefordert und der Rechner kann das Statuswort lesen. Damit ist eine einfache Fehlerdiagnose möglich. Der Time-Out-Generator wird zu Beginn eines jeden Zyklus gestartet und am Ende wieder gelöscht, es sei denn, es vergeht eine längere als die für jeden Zyklus festgelegte Zeit.

Gelöscht wird das Time-Out-Signal mit "Lösche Interrupt 8" (LIT8) oder "Clear System Controller" (CLR).

#### 3.3.3.2. Enable System Controller

Ein Enable/Disable Flipflop (ESC) gestattet es, die Kommunikation zwischen System Controller und angeschlossenem Crate Controller vom Rechner aus zu erlauben oder zu verbieten.

#### 3.3.4. Modul "Control- und Statusregister"

Das Statusregister hat eine Kapazität von 22 Bit und ist vom Rechner lesbar. Es soll dem Rechner die wichtigsten Statusmeldungen zur Erleichterung einer Fehlerdiagnose geben. Die Belegung der Bits 1 ... 13 des Registers und die Zuordnung zu den Informations-Leitungen ist aus Fig. 11 ersichtlich. Bit Nr. 14 und 15 bleiben frei, die

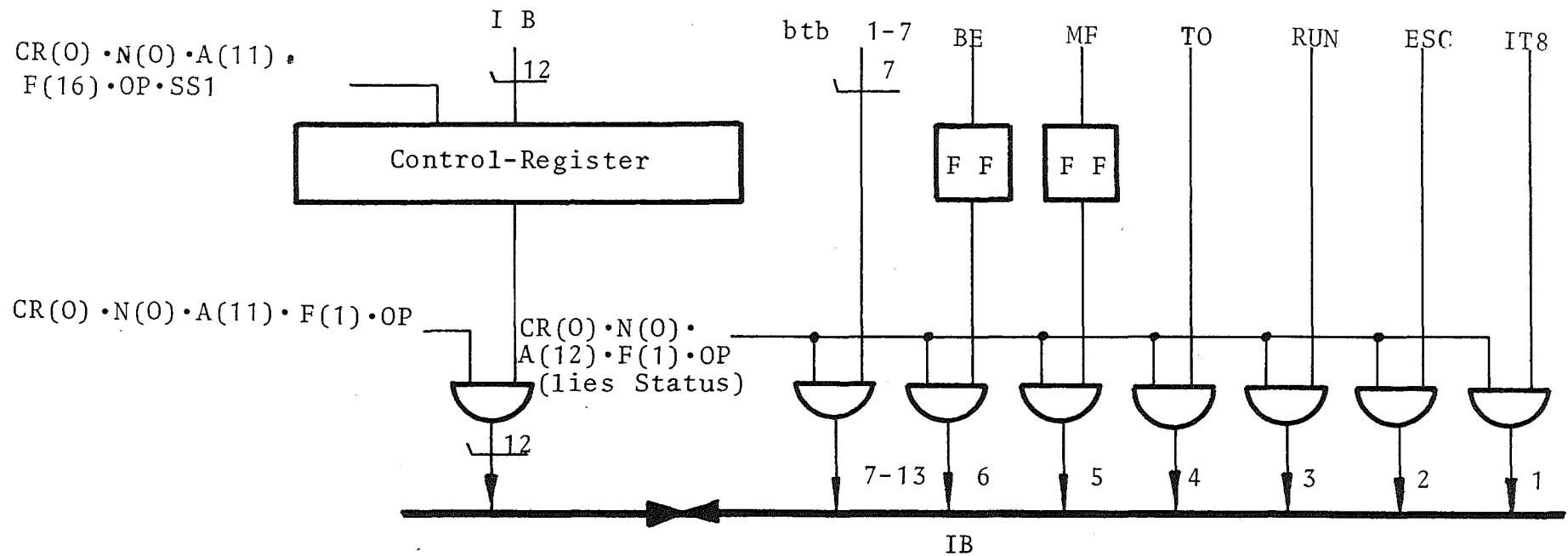


Fig. 11: Control- und Statusregister



Bits 16 ... 24 sind aus Fig. 12 (unten) zu ersehen. Das Statusregister hat die Subadresse A(12).

Das Control-Register hat eine Kapazität von 12 Bit und ist vom Rechner lesbar mit A(11). Die Zuordnung zu den Informations-Busleitungen ist:

Interner Bus	Bit 1	ENBATE	( <u>E</u> n <u>B</u> l <u>a</u> ck <u>a</u> u <u>t</u> o <u>n</u> o <u>m</u> e <u>r</u> <u>T</u> r <u>a</u> n <u>s</u> f <u>e</u> r <u>s</u> <u>E</u> i <u>n</u> g <u>a</u> b <u>e</u> )
"	"	Bit 2	ENITS (Enable <u>I</u> nt <u>e</u> r <u>r</u> u <u>p</u> t <u>s</u> 1 bis 7)
"	"	Bit 3	ENIT8 (Enable <u>I</u> nt <u>e</u> r <u>r</u> u <u>p</u> t 8)
"	"	Bit 4	ADS (Address <u>S</u> can Mode)
"	"	Bit 5	RPM ( <u>R</u> e <u>p</u> e <u>a</u> t <u>M</u> o <u>d</u> e)
"	"	Bit 6	STM ( <u>S</u> t <u>o</u> p <u>M</u> o <u>d</u> e)
"	"	Bit 7	M (Anforderung <u>M</u> o <u>d</u> u <u>l</u> -z <u>u</u> - <u>M</u> o <u>d</u> u <u>l</u> - <u>T</u> r <u>a</u> n <u>s</u> f <u>e</u> r)
"	"	Bit 8	IN ( <u>I</u> nk <u>r</u> e <u>m</u> e <u>n</u> t <u>i</u> e <u>r</u> e Stationsnummer <u>N</u> )
"	"	Bit 9	IA ( <u>I</u> nk <u>r</u> e <u>m</u> e <u>n</u> t <u>i</u> e <u>r</u> e Subadresse <u>A</u> )
"	"	Bit 10	SWD ( <u>S</u> w <u>i</u> t <u>c</u> h <u>D</u> a <u>t</u> a, umschalten auf RDK)
"	"	Bit 11	ENATA (Enable <u>a</u> u <u>t</u> o <u>n</u> o <u>m</u> e <u>r</u> <u>T</u> r <u>a</u> n <u>s</u> f <u>e</u> r <u>A</u> u <u>s</u> g <u>a</u> b <u>e</u> )
"	"	Bit 12	ENATE (Enable <u>a</u> u <u>t</u> o <u>n</u> o <u>m</u> e <u>r</u> <u>T</u> r <u>a</u> n <u>s</u> f <u>e</u> r <u>E</u> i <u>n</u> g <u>a</u> b <u>e</u> )

### 3.3.5. Modul "GL-Register und Interrupt-Behandlung"

Das GL- oder Alarmregister hat eine Kapazität von 24 Bit und wird bei Alarm-Operationen, also bei Anwesenheit des Signalpaares BD/BG (Demand Operation), über die BRW-Leitungen gesetzt (siehe Fig. 13). Die ersten 8 Bit steuern 8 Interrupt-Leitungen des Rechners an, die restlichen 16 Bit werden als Anforderungen für autonome Eingaben in den Kernspeicher des Rechners interpretiert. Diese Bits (9 ... 24) werden nach Abwicklung der nötigen Operationen automatisch zurückgesetzt, die Interrupt-Speicher müssen vom Rechner gezielt über die Schreibleitungen 1 bis 8 gelöscht werden. Der Ablauf der einzelnen Operationen wird in Kapitel 4 genauer besprochen.

Fig. 12: Auto Increment N.A.

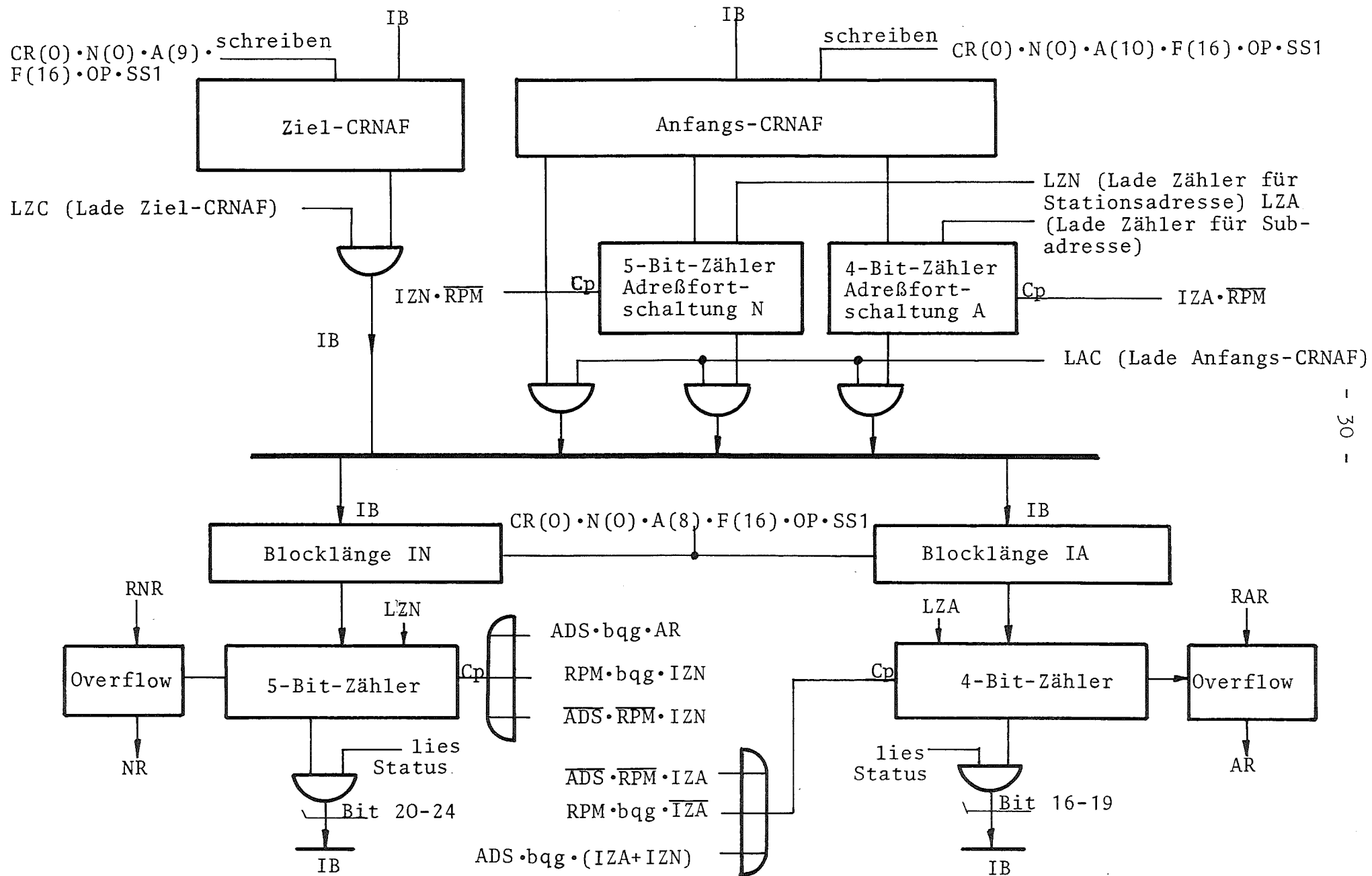
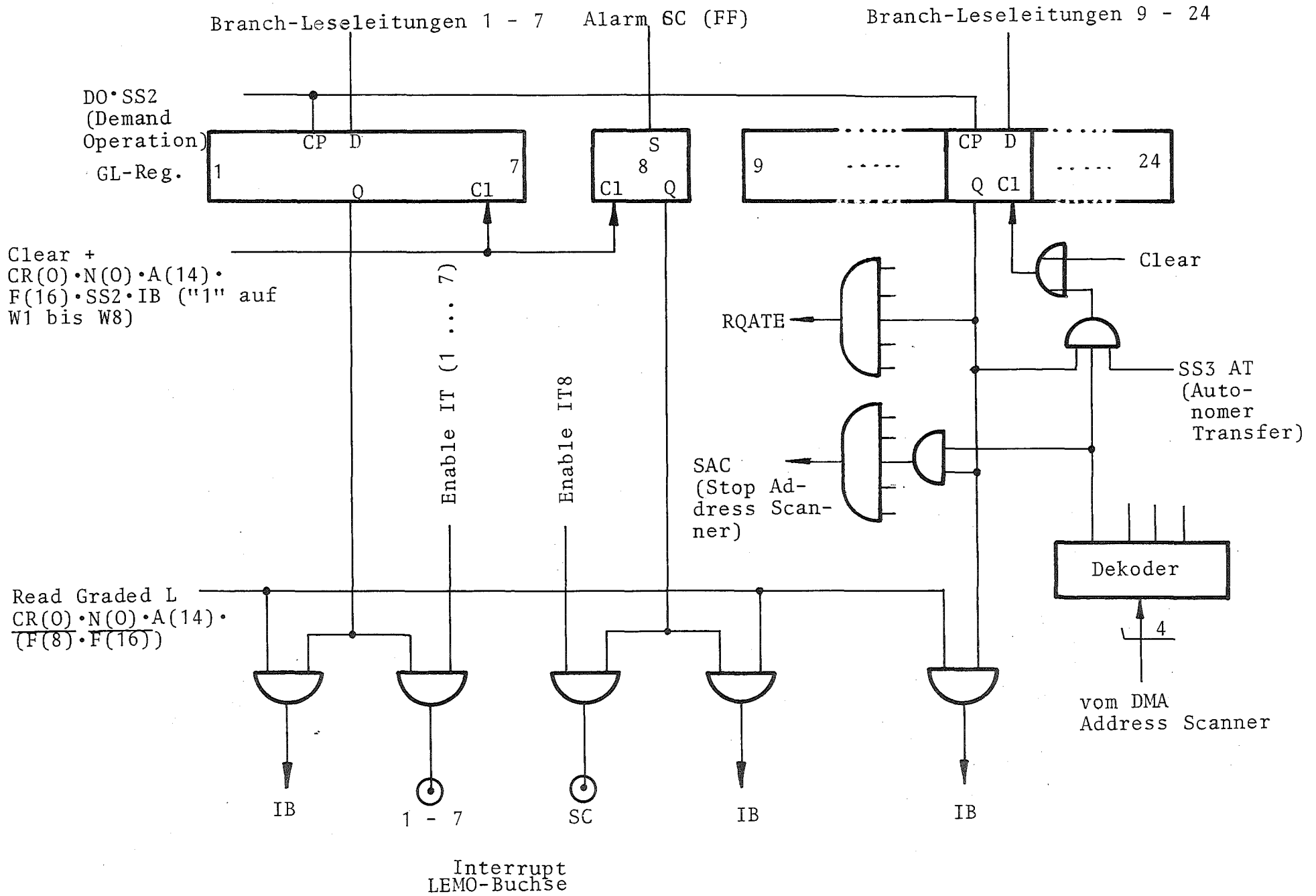


Fig. 13: GL-Register und Interrupt-Behandlung



### 3.3.6. Modul "Auto Inkrement N.A."

Zur Blocktransfer-Einrichtung gehören nach Fig. 12 folgende Bestandteile, um die Stationsnummer und/oder Subadresse zu inkrementieren:

Ein Speicher für den ersten Befehl des Blockes "Anfangs-CRNAF", eine Adreßfortschaltung für N und A, je ein Pufferspeicher für die Anzahl der Transfers in N und A, je ein Zähler für die Anzahl der Transfers in N und A, ein Blockstartspeicher.

Folgende Rechner-Operationen sind nötig, um den System Controller für die Betriebsart des Blocktransfers vorzubereiten:

- 1 Der Rechner schickt einen Befehl mit Operand, wobei der Operand den Anfangs-CRNAF darstellt, der in das dafür vorgesehene Register mit der Subadresse A(10) eingeschrieben wird. Dies ist ein Beispiel für einen Dialog Rechner - System Controller, der mit der möglichen Maximalgeschwindigkeit abläuft.
  
- 2 Durch einen zweiten Befehl mit Operand lädt der Rechner das 2-Komplement der Anzahl der in N bzw. A gewünschten Transfers in die Pufferspeicher mit der Subadresse A(8).

Damit ist der System Controller für einen Blocktransfer vorbereitet. Erscheint der Spezialbefehl "Blockstart" (BS) CR(0) · N(0) · A(6) · F(26), so wird der Blocktransfer gestartet. Der Rechner braucht immer nur Daten abzuholen oder zu schicken und keine Befehle mehr zu senden, da diese sich bei dieser Betriebsart selbsttätig im System Controller durch Inkrementieren von Stationsnummer N oder Subadresse A bilden (Bits IN und IA im Control-Register). Beim letzten Transfer wird das Signal "Blockende" (BE) gebildet und an den Rechner gegeben. Es wird aus den Overflows der N- bzw. A-Zähler nach der in Tabelle 4 gegebenen Übersicht erzeugt.

Tabelle 4: Bildung des Signals "Blockende" (BE)

IN	IA	BE wird gebildet aus	Signal-Name
0	0	—	—
0	1	A-Zähler	AR
1	0	N-Zähler	NR
1	1	A- und N-Zähler	NR u. AR

Soll in den Betriebsarten ADS oder RPM die Blocklänge vorherbestimmt werden, so wird der Zähler für die Anzahl der Transfers für N mit dem für A zu einem 9-Bit-Zähler gekettet. Dieser Zähler kann mit dem Befehl "lies Status" ausgelesen werden.

Da die Information (Blocklänge) in A(10) und A(8) beim Abarbeiten des Blockes nicht zerstört wird, kann der Block immer wieder neu gestartet werden.

### 3.3.6.1. Modul-zu-Modul-Transfer

Der Modul-zu-Modul-Transfer benötigt folgende Hardware:

- 1 Die gesamte in Abschnitt 3.3.6 genannte,
- 2 das Lesebuffer-Register (vgl. Abschnitt 3.3.2),
- 3 einen Speicher für den Befehl, der die Daten zu ihrem Ziel befördert ("Ziel-CRNAF"),
- 4 eine automatische Umschaltung der Transfer-Richtungen Eingabe und Ausgabe, beginnend mit Eingabe,
- 5 einen Spezialbefehl zur Einstellung der gewünschten Betriebsart (Control-Register Bit 7).

Die Figuren 7 (S. 20), 12 (S. 30) und 15,16 (S. 38, 39) zeigen ein Schema der kompletten Hardware. Zu den in Abschnitt 3.3.6. genannten Einstell-Operationen gehört nun ergänzend noch das Laden des "Ziel-CRNAF" in das Register mit der Subadresse A(9).

Mit dem Befehl "Blockstart" an den System Controller wird der Block aktiviert und zuerst der in Subadresse A(10) befindliche "Anfangs-CRNAF" ins Befehlsregister geholt bei gleichzeitiger zwangsweiser Umschaltung der Transfer-Richtung auf Eingabe. Die gelesenen Daten werden im Lesebuffer abgelegt. Diese Daten werden beim zweiten Zyklus des System Controller unter der Kontrolle des in A(9) befindlichen und ins Befehlsregister transportierten Befehls "Ziel-CRNAF" zum Ziel transportiert. Zu Beginn des zweiten Zyklus war auf Ausgabe umgeschaltet worden. Gelesen werden kann wahlweise entweder immer derselbe Modul oder mit Hilfe der Inkrementierung ein Block von Modulen. Für jede Ausgabe ist das Ziel das gleiche.

Das Signal "Blockende" wird bei der letzten Ausgabe gebildet. Wird eine Wiederholung gewünscht, dann muß der Befehl "Blockstart" neu gegeben werden. In der Betriebsart Modul-zu-Modul-Transfer bleibt der Rechner völlig unbelastet, er erhält lediglich zu gegebener Zeit das Signal "Blockende".

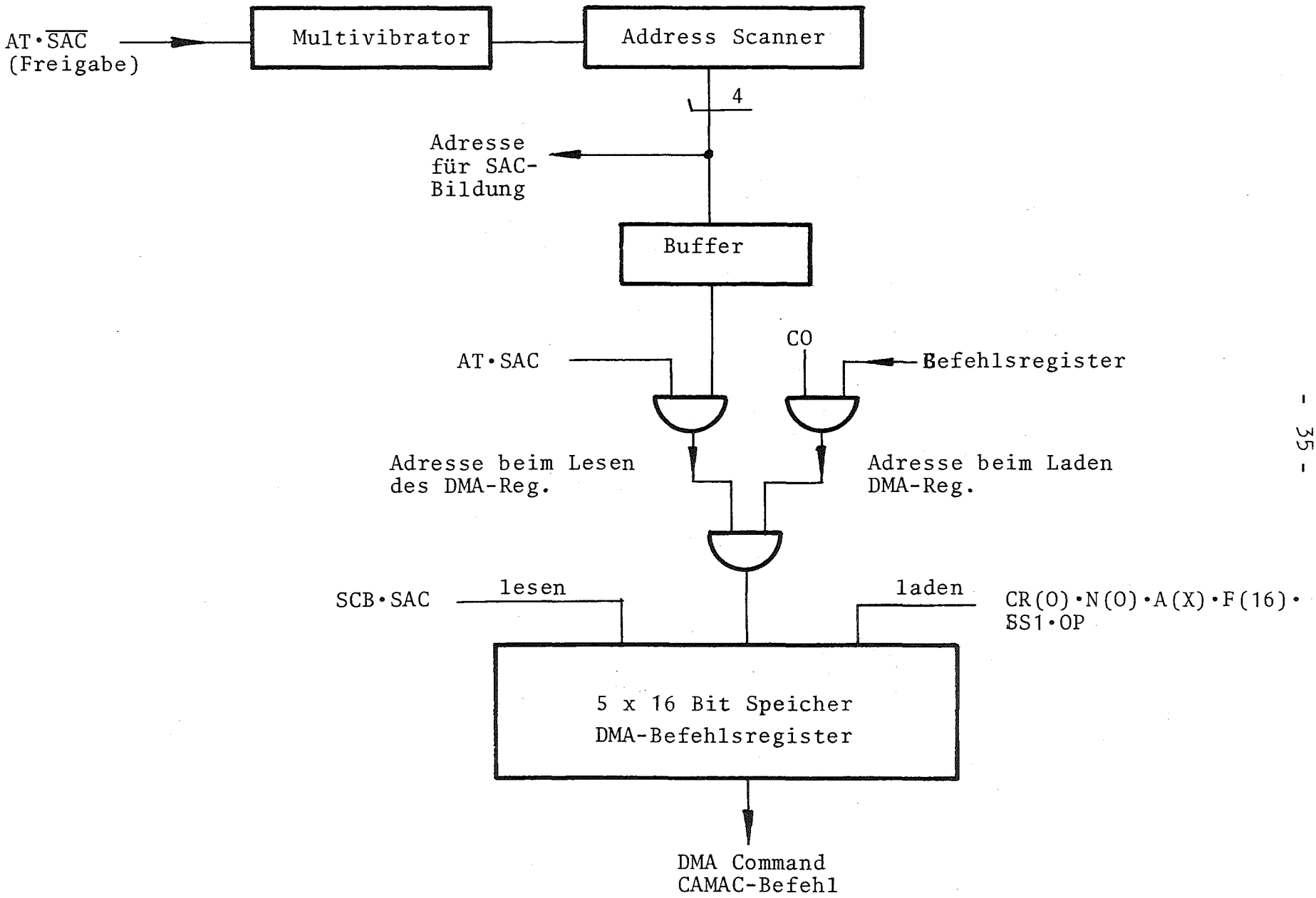
Es ist möglich, daß Hauptzustände höherer Priorität den Blocktransfer sowie den Modul-zu-Modul-Transfer unterbrechen. Sie lassen sich beide nach dem Verschwinden der Hauptzustände höherer Dringlichkeit an der Stelle fortsetzen, an der sie suspendiert wurden, und zwar durch den Befehl "Blockstart". Es geht dabei keine Information verloren, wenn der Lesebuffer vor der Unterbrechung gelesen wurde.

### 3.3.7. Modul "DMA-Befehlsregister"

#### 3.3.7.1. Autonomer Transfer in Eingabe-Richtung

Autonome Transfers in Eingabe-Richtung stehen gewöhnlich unter Zeitdruck, wie schon erläutert. Daher wurde die Hardware auf größtmögliche Geschwindigkeit ausgelegt. In Fig. 13 (S. 31) und 14 ist die beteiligte Hardware zu sehen. Bis zu 16 im ganzen System verstreute Einheiten können an diese Einrichtung angeschlossen werden.

Fig. 14: DMA-Befehlsregister



Sie geben ihre Anforderungen auf autonome Übertragungen (ak) in einer normalen Alarm-Operation (also unter der Kontrolle des Signalpaares BD/BG) in das Alarmregister des System Controller, und zwar in die oberen 16 Bits. Die Wertigkeit zählt vom Bit 24. Ein Scanner prüft, welche der Anforderungen gekommen ist, und stoppt, wenn er eine gefunden hat. Dann wird aus einem von 16 Vorratsspeichern, ein der Quelle der Anforderung zugeordneter Befehl aufgerufen und ins Befehlsregister geladen. Es etabliert sich der Hauptzustand ATE (Autonomer Transfer Eingabe). Die folgende normale Lese-Operation liefert ein 24-Bit-Wort über die BRW-Leitungen, das Lesebuffer-Register und die IB-Leitungen auf die 24 FKAS-Leitungen. Die Anwesenheit dieser Daten wird dem Rechner durch das Signal RMTE angezeigt. Der Rechner übernimmt die Daten. Der Rechner muß seine Bereitschaft zum Empfang solcher Daten durch das Signal FAZ ausdrücken. Nach der Abgabe der Information löscht der System Controller die zugehörige Anforderung in seinem Alarmregister. Liegen immer noch unerledigte Anforderungen vor, so wird der Hauptzustand ATE wieder hergestellt, und der Scanner startet erneut seine Suche. Das Löschen der letzten Anforderung dieser "Charge" gibt den System Controller wieder frei für andere Hauptzustände. Das Programm des Rechners ist bei diesen Operationen nicht beteiligt. Die hereinkommenden Daten gehen direkt in den Kernspeicher in Adressen, die erst in der RDK-Elektronik festgelegt werden.

### 3.3.7.2. Autonomer Transfer in Ausgabe-Richtung

Der System Controller ist bis auf die Schnittstelle des RDK-Kanals für diese Betriebsart voll ausgebaut. Da der Radar-Daten-Kanal nur in Eingabe-Richtung betrieben werden kann, entfällt diese Betriebsart.



### 3.3.8. Modul "Control Unit"

Der Modul besteht im wesentlichen aus Verknüpfungslogik, die verschiedene Signale des System Controller miteinander verknüpft. Erzeugt werden folgende Kontroll-Signale:

siehe Fig. 15, 16 und Fig.

ENRDK/DIRDK	Enable/Disable Radar-Daten-Kanal	3
ta	Timing-Signal für Branch Driver	5
bg	Graded-L Request Signal (Demand Operation)	5/6
STBR	Strobe für Befehlsregister	6
E <sup>+</sup>	Signal zur Umschaltung zwischen Eingabe und Ausgabe bei Modul-zu-Modul-Transfer	7
RNR/RAR	Reset "N Transfers ready" oder "A Transfers ready"	12
IZN/IZA	Strobe zum Inkrementieren von Zähler und Adreßfortschaltung N oder A	12
LAC	Lade Befehlsregister aus Speicher für Anfangs-CRNAF	12
LZC	Lade Befehlsregister aus Speicher für Ziel-CRNAF	12
LZN/LZA	Lade Zähler N oder A	12
FF	Fehlfunktion (Interrupt IT8)	13
MF	Malfunktion	16
BS	Blockstart	
BE	Blockende	

Damit sind alle wesentlichen Grund- und Zusatzfunktionen des System Controller beschrieben. Im nächsten Abschnitt werden anhand von Zeitdiagrammen und Signalflußbildern einige wichtige Abläufe im Detail geschildert.

Fig. 15: Control Unit

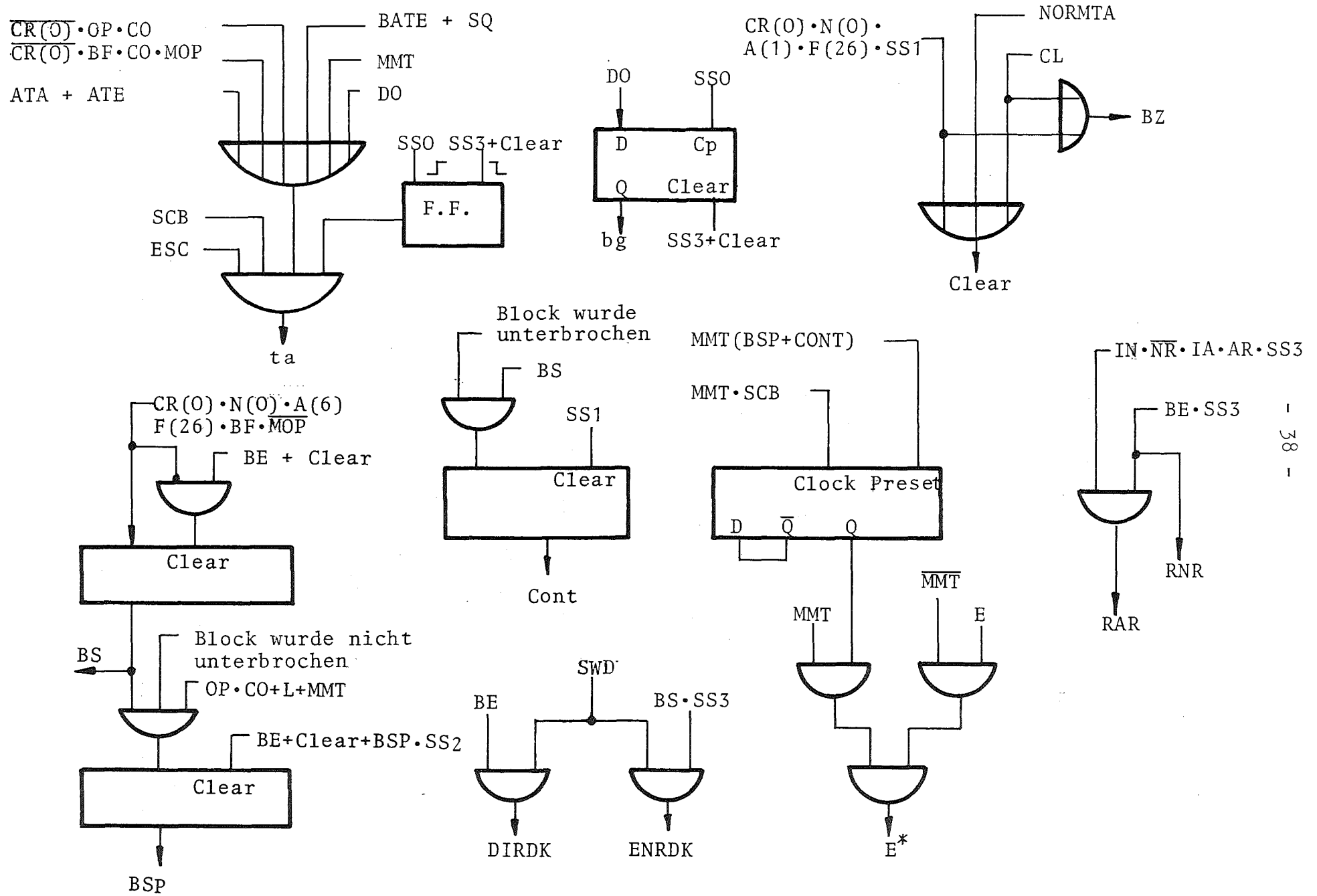
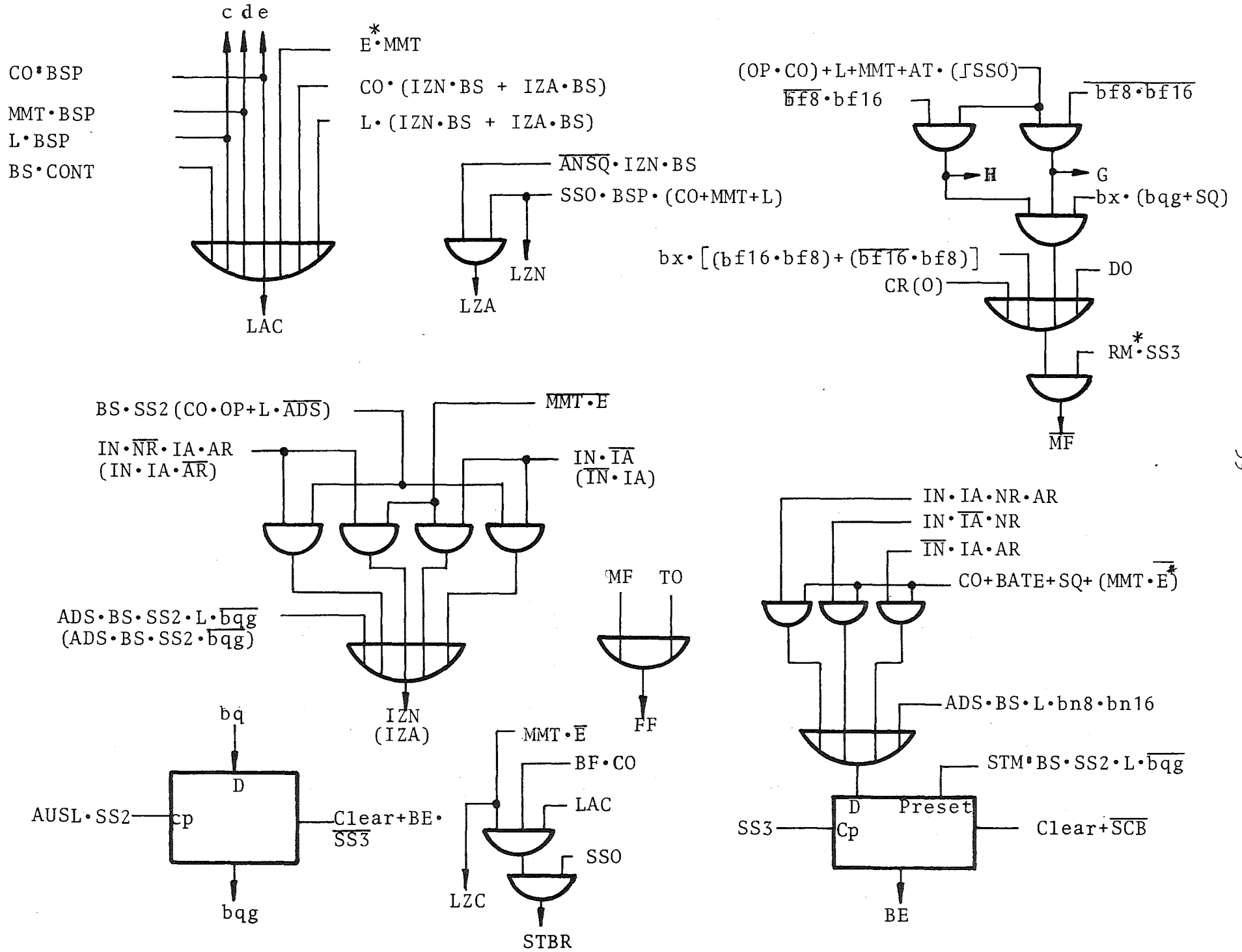


Fig. 16: Control Unit



#### 4. Funktionsabläufe

Fig. 17 gibt einen Überblick über die unterschiedlichen Arbeitsweisen des System Controller. Zunächst sollte zwischen Befehls- und Alarm-Operationen unterschieden werden, wie im CAMAC-System üblich. Nach bewährtem Muster erfolgt dann die weitere Differenzierung nach Einadreß- und Multi-Adreß-Operationen, wobei bei ersteren der Dialogverkehr System Controller - Rechner eine eigene Klasse bildet. Nach vorheriger Einstellung des System Controller durch Dialog-Operationen werden die Betriebsweisen "Sequentielle Operationen im Q Mode" (SQ) sowie "Blocktransfer" wie Einadreß-Operationen abgewickelt. Der Modul-zu-Modul-Transfer nimmt eine Sonderstellung ein, da er nicht vom Rechner überwacht wird.

Nach Karlsruher Vorstellungen wird die durch den BD/BG-Automatismus ins Alarmregister gelieferte Information in zwei Gruppen geteilt und mit ganz verschiedenen Methoden weiterbehandelt. Es sind die aus der Rechnertechnik bekannten Methoden der Interrupt-Verarbeitung einerseits und das Cycle-Stealing-Verfahren andererseits. Diese Methoden hängen nur insofern vom Rechner ab, als ein Direktzugriffs-Kanal und wenigstens eine Interrupt-Ebene gebraucht werden. Natürlich sind 8 Interrupt-Ebenen wie im vorliegenden Falle dem Problem angemessener. In den folgenden Abschnitten werden einige Befehls-Operationen und die Alarm-Verarbeitung besprochen.

##### 4.1. Befehls-Operationen

Von den Befehls-Operationen seien zuerst die Einadreß-Operationen behandelt, und zwar der Dialogverkehr Rechner - System Controller. Die zum System Controller fließende Information ist markiert. Der leichteren Übersicht halber sind die Markierungsbits in Tabelle 5 zusammengestellt und die Bedeutung der möglichen Kombinationen erklärt.

Fig. 17: Funktionen des System Controller

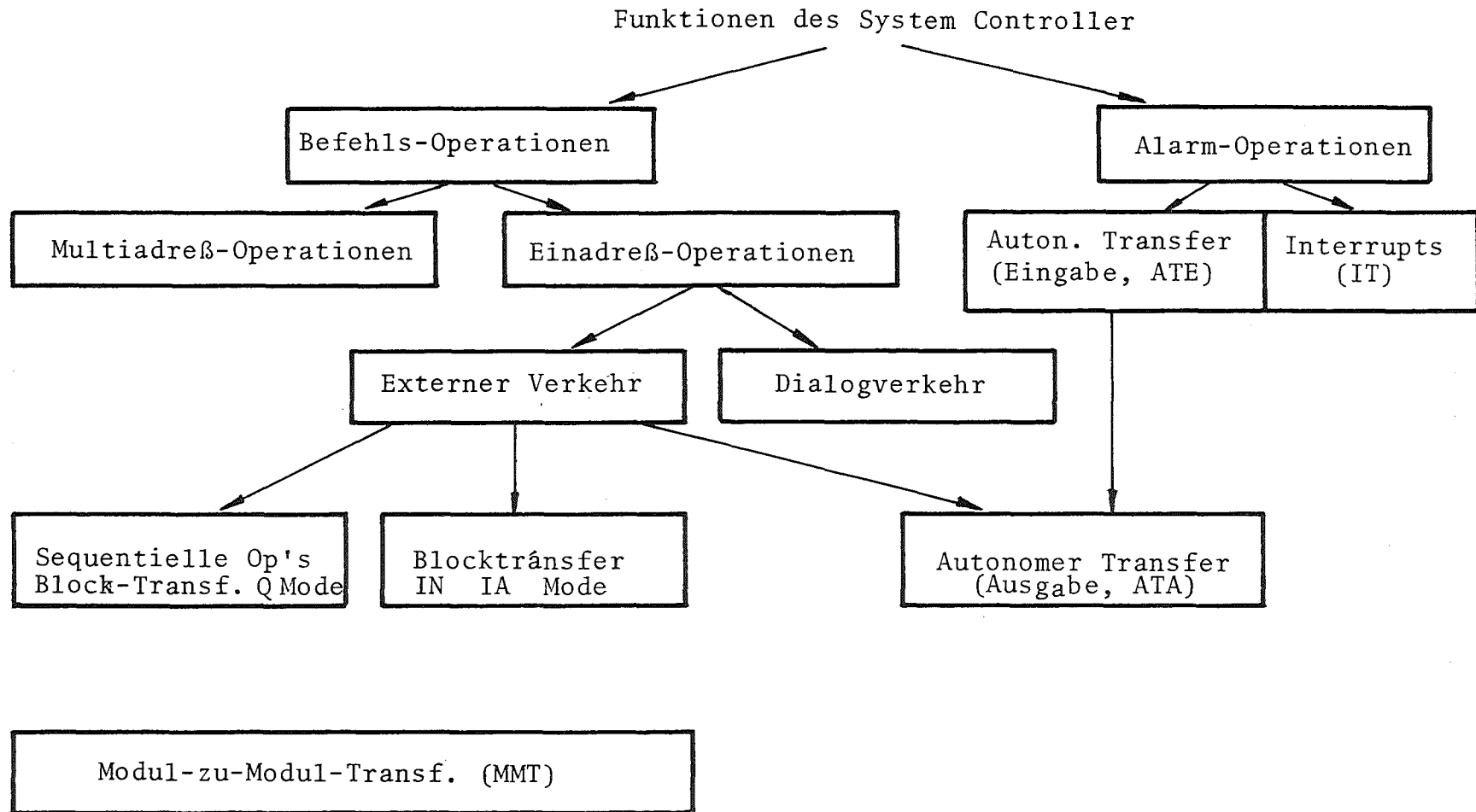


Tabelle 5: Kennung der vom Rechner gelieferten Information

Trans- fertyp	Signal					Bedeutung
	BF	OP	MOP (Bit 19)	E (Bit 20)	cr0	
1	1	0	0	0	1	Befehl ohne Operand an System Controller
2	1	0	1	0	1	Befehlsteil eines Befeh- les mit Ope- rand für Sy- stem Controller
2a	0	1	-	-	-	Daten an System Controller
3	1	0	1	1	0	Befehlsteil eines Lese- befehles für System
3a	0	1	-	-	-	Abholung der Lesedaten
4	1	0	1	0	0	Befehlsteil eines Schreib- befehles für System
4a	0	1	-	-	-	Anlieferung der Schreib- daten
5	1	0	0	0	0	Befehl ohne Operand an System

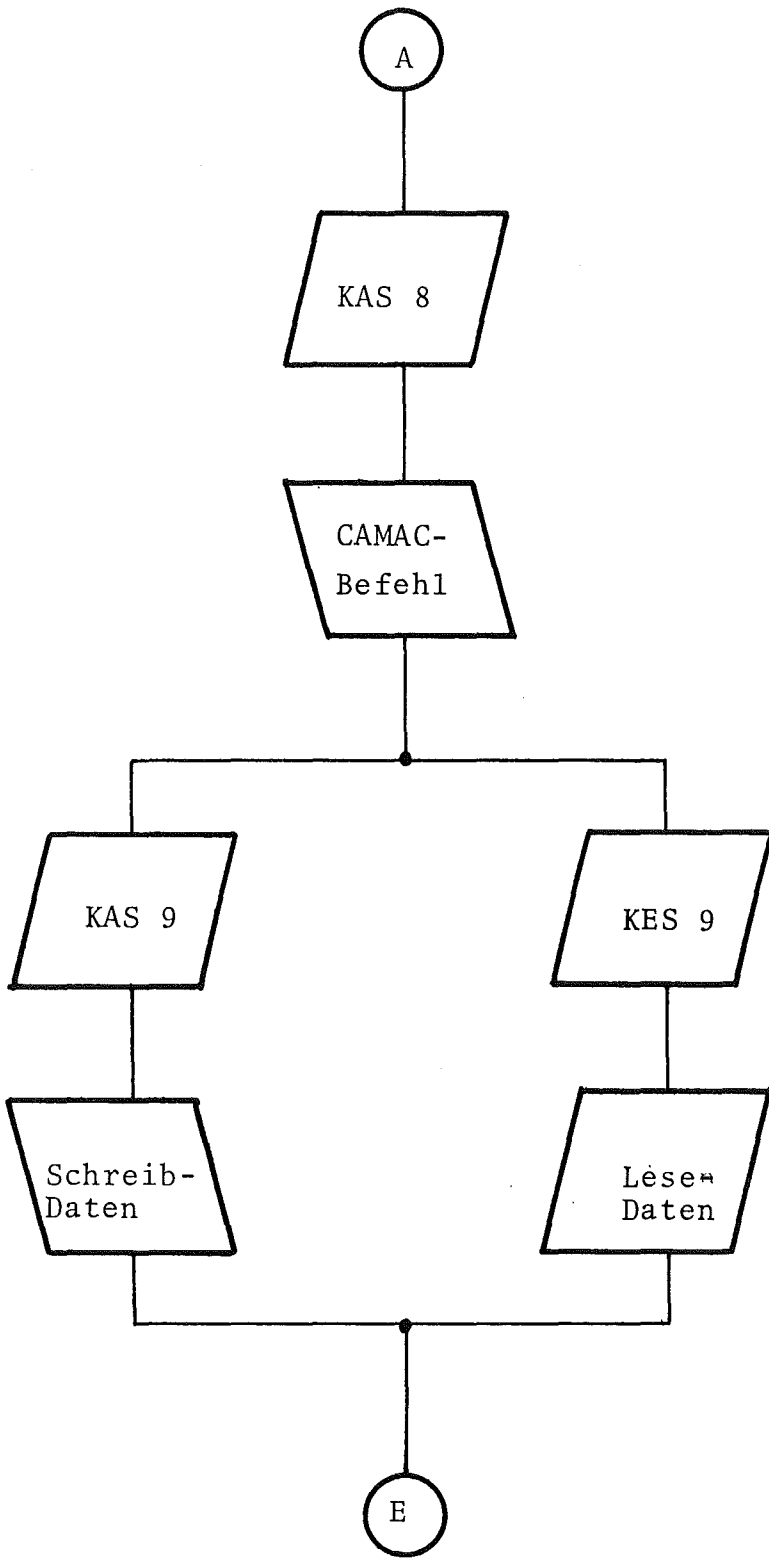
#### 4.1.1. Dialog Rechner - System Controller ohne Operand

In dem Timing-Diagramm von Fig. 10 (S. 25) ist als Beispiel für diese Operationen die Übernahme des Befehles "Blockstart" eingetragen. Die Pfeile in dem Diagramm stellen Wirkungslinien dar.

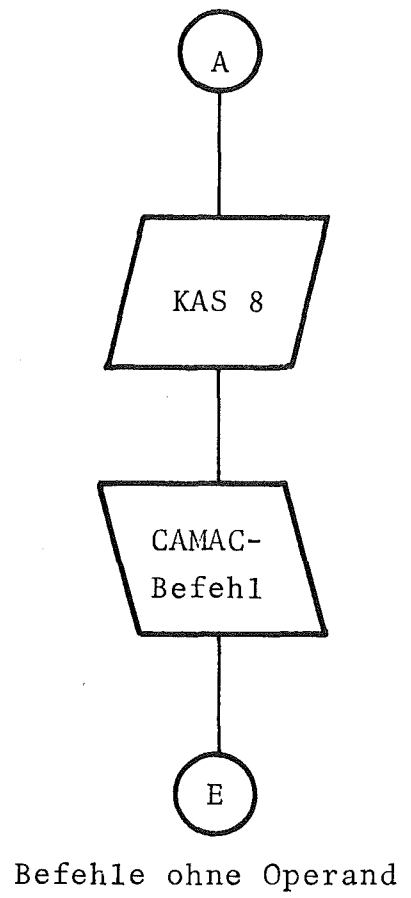
Es gilt Zeile 1 in Tabelle 5, denn es ist ein Befehl ohne begleitende Daten (Operand) an den System Controller. Die Information auf den IB-Leitungen sowie auf den Leitungen BF und OP erscheint zuerst und steht am System Controller an, der sich in Ruhe befindet. Er wird erst durch RAR\* aktiviert. Dabei wird vorausgesetzt, daß kein Hauptzustand höherer Ordnung herrscht. Das Signal RAR\* erzeugt im Hauptzustands-Generator den Zustand "Command Operation" (CO) und schützt den System Controller gegen jeden weiteren Anspruch durch das Signal SCB (System Controller Busy). CO triggert den Timing-Generator über den Eingang für den Fremdstart, wobei auf hohe Verarbeitungsgeschwindigkeit Wert gelegt wird. Ein Uhrimpuls erzeugt nach einer Pause den nächsten. Die Rückflanke von SS2 gibt dem Rechner die erwartete Rückmeldung ZM über den Vollzug des Befehles. Rund 250 ns nach SS3 ist der System Controller wieder frei. Neben dem Taktdiagramm kann der Programmablauf nach Fig. 18 nützlich sein, den Ablauf der Operation zu verdeutlichen.

#### 4.1.2. Dialog Rechner - System Controller mit Operand

Befehle für den System Controller mit Operand sind in Tabelle 5 als Typ 2 bezeichnet. Es sind zweizyklische Operationen wie die entsprechenden Befehle an das CAMAC-System. Der Unterschied liegt darin, daß beim zweiten Zyklus derselbe rasche Ablauf stattfindet wie bei der Absetzung des Befehls selbst (siehe Fig. 10, S. 25).



Befehle mit Operand



Befehle ohne Operand

Fig. 18: Programmablauf für Befehle mit und ohne Operand



#### 4.1.3. Einadreß-Lesebefehle

Zur Beschreibung des Ablaufes gehören die Figuren 19 und 20 sowie die Zeilen 3 und 3a der Tabelle 5. Die Operation erfordert 2 Zyklen des System Controller, im ersten wird der Lesebefehl empfangen und im Befehlsregister abgelegt, im zweiten werden die Lesedaten an der Quelle abgeholt und dem Rechner zusammen mit der Rückmeldung ZM übergeben. Beim zweiten Zyklus ist eine Laufzeit der Signale im CCA und auf dem Branch Highway angenommen. Unter diesen Umständen ergibt sich eine Gesamtdauer für den zweiten Zyklus von 2,6  $\mu$ s, wogegen die Absetzung des Befehles in 1,8  $\mu$ s geschehen kann. In Fig. 19 ist noch die Verwirklichung einiger stets befolgter Regeln zu sehen:

- 1 Die Rückflanke von SSO erzeugt ein Signal TA, womit der Befehl (bei Schreibbefehlen auch die Schreibdaten) auf den Branch Highway gegeben werden, damit sie einen Vorsprung vor BTA erhalten;
- 2 der Uhrimpuls SS1 folgt nach einer Pause von ca. 100 ns auf SSO und mit seiner Rückflanke wird BTA erzeugt. Die nun folgende Pausenzeit ist variabel und hängt u. a. von der Länge des Branch Highway ab.
- 3 der Uhrimpuls SS2 wird erzeugt, wenn die logische Gleichung (1) erfüllt ist. Von seiner Rückflanke wird die Rückmeldung (ZM) an den Rechner und die Rücknahme von BTA abgeleitet;
- 4 die Erfüllung der Gleichung (2) erzeugt den Impuls SS3. Etwa 250 ns nach seiner Rückflanke ist der System Controller wieder frei.

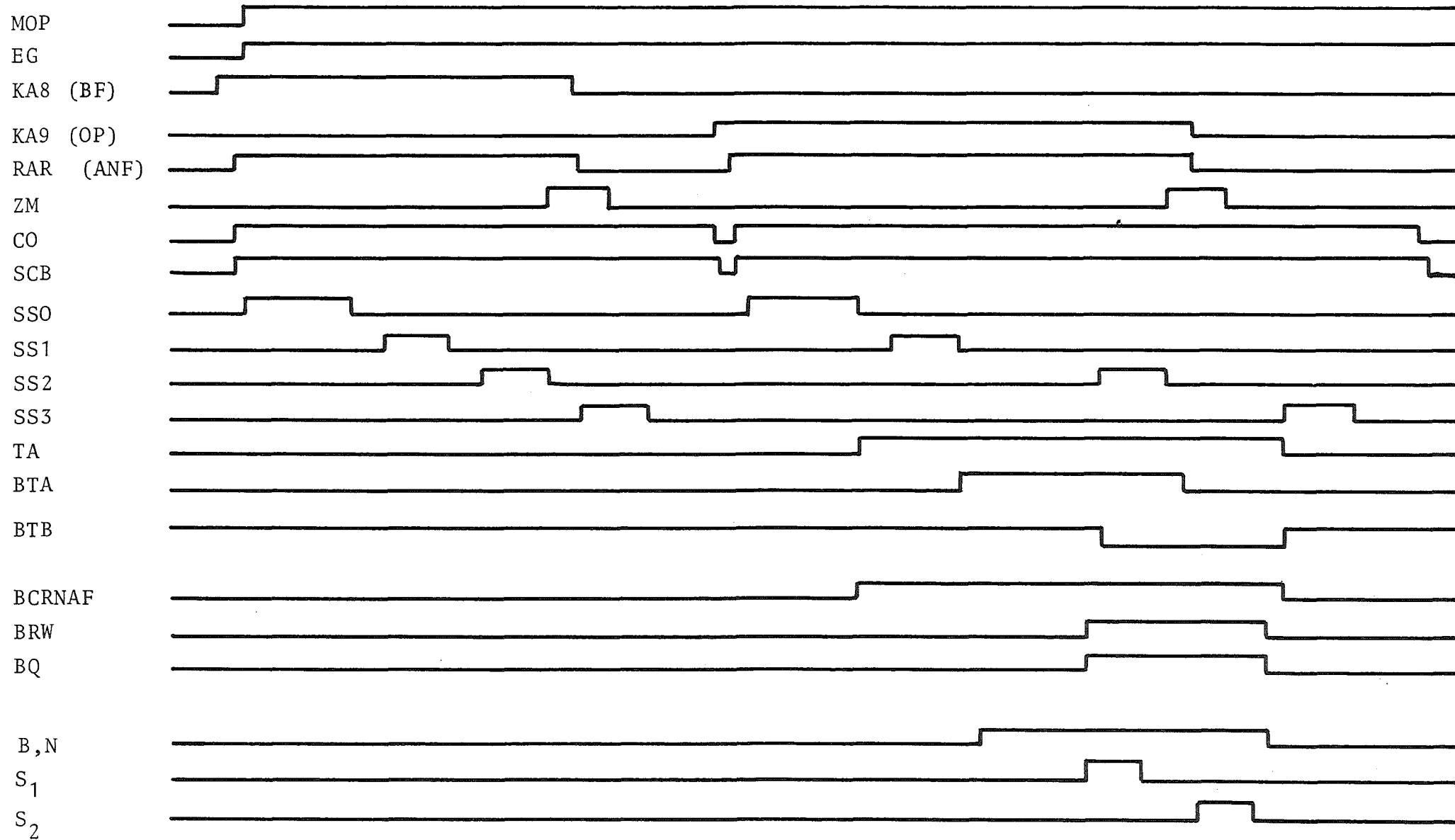


Fig. 19: Timing-Diagramm für Lesebefehle

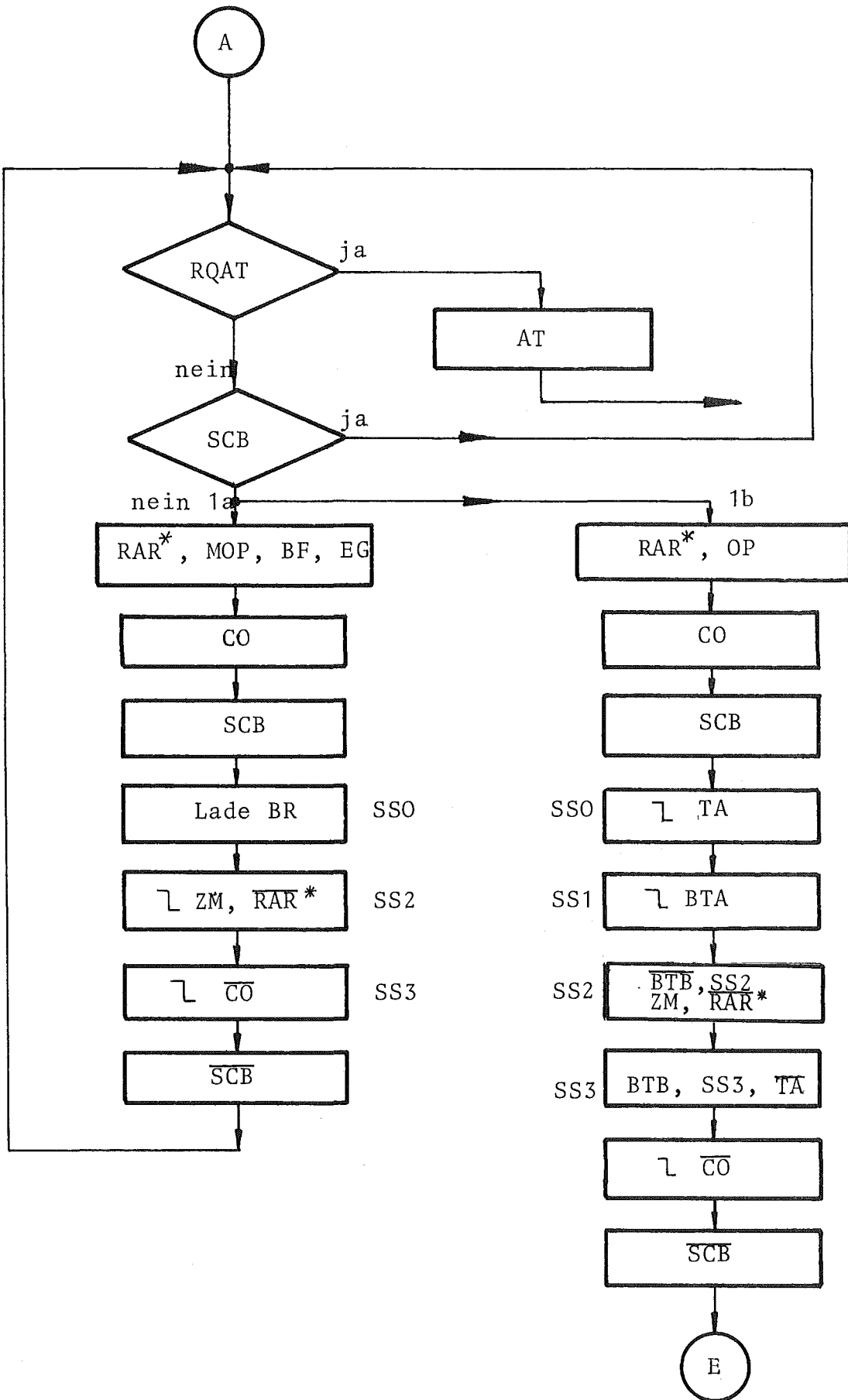


Fig. 20: Signalfluß-Schema für Lesebefehle

#### 4.1.4. Einadreß-Schreibbefehle

Diese Befehle unterscheiden sich von den Lesebefehlen praktisch nur darin, daß die Transfer-Richtung umgekehrt wird (siehe Tabelle 5, Zeile 4 und 4a). Deshalb wird hier auf eine eingehendere Besprechung verzichtet.

#### 4.1.5. Befehle ohne Operand

Bei Befehlen ohne Operand werden die BRW-Leitungen nicht benutzt, d. h. der System Controller braucht nicht mit der Ausführung zu zögern. Deshalb sind es einzyklische Operationen. Es galten Zeile 5 in Tabelle 5 und die Figuren 18 (S. 44) und 21. Der Befehl wird unmittelbar nach dem Empfang auf den Branch Highway gegatet zur direkten Ausführung im System.

Damit sind alle Grundtypen der Befehls-Operationen besprochen. Auch Multi-Adreß-Operationen bedürfen keiner weiteren Erläuterung. Multi-Station-Operationen sind für den System Controller ohnehin ganz normale Einadreß-Operationen, denn die Vielfach-Adressierung besorgt ja die Rahmensteuerung nach entsprechender Voreinstellung. Was Multi-Crate-Operationen anbelangt, so sind auch für diese die Gleichungen (1) und (2) gültig.

Komplexere Befehls-Operationen setzen sich aus mehreren Operationen der eben beschriebenen Grundtypen zusammen. Als Beispiel sei der rechnergesteuerte Blocktransfer behandelt.

#### 4.1.6. Blocktransfer IN, IA Mode

Um einen Blocktransfer mit Inkrementierung von N und A in Gang zu bringen, müssen verschiedene, aus Fig. 22 erkennbare Einstellungen des System Controller vorgenommen werden. Der letzte Befehl

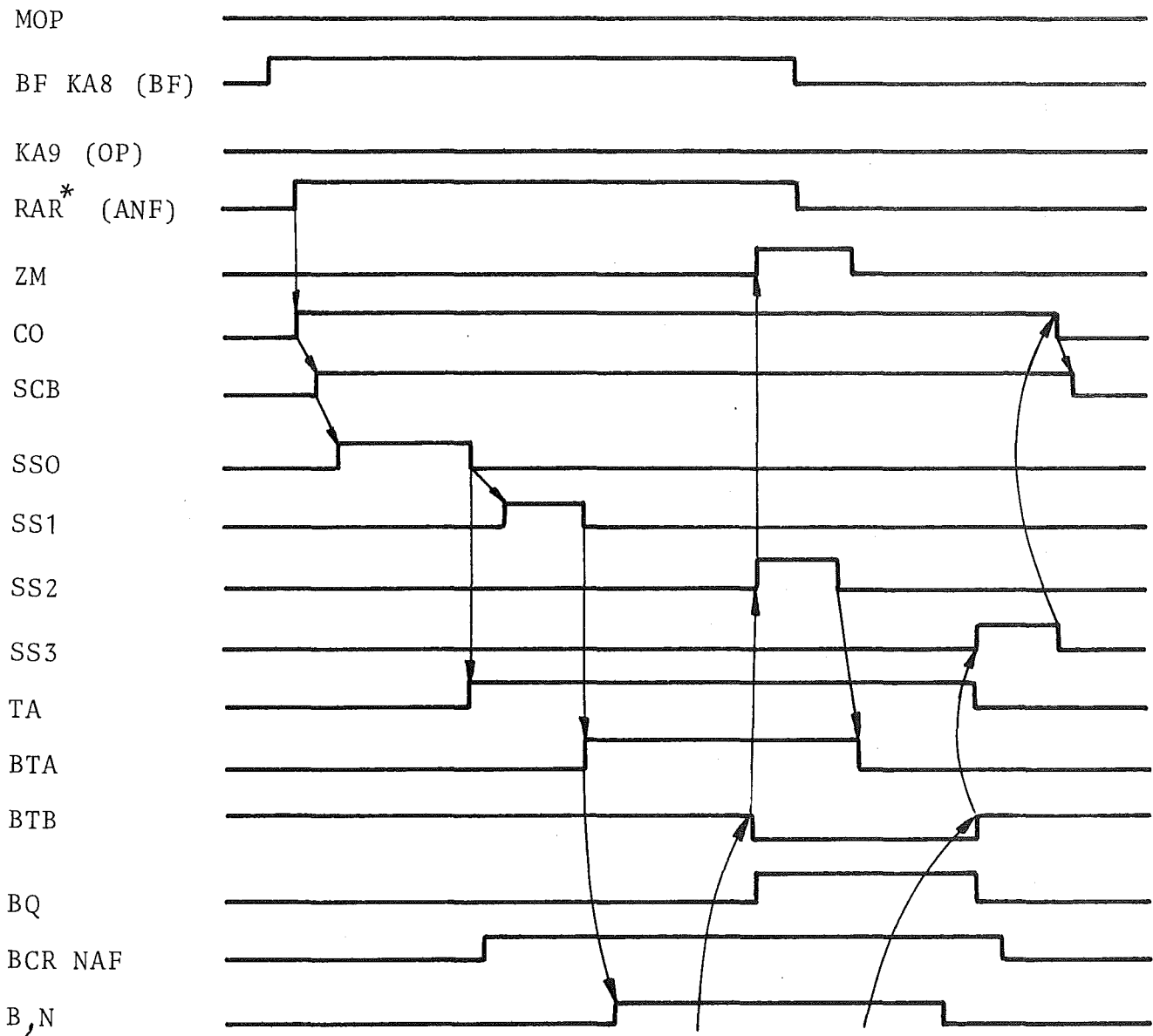


Fig. 21: Timing-Diagramm für Befehle ohne Operand

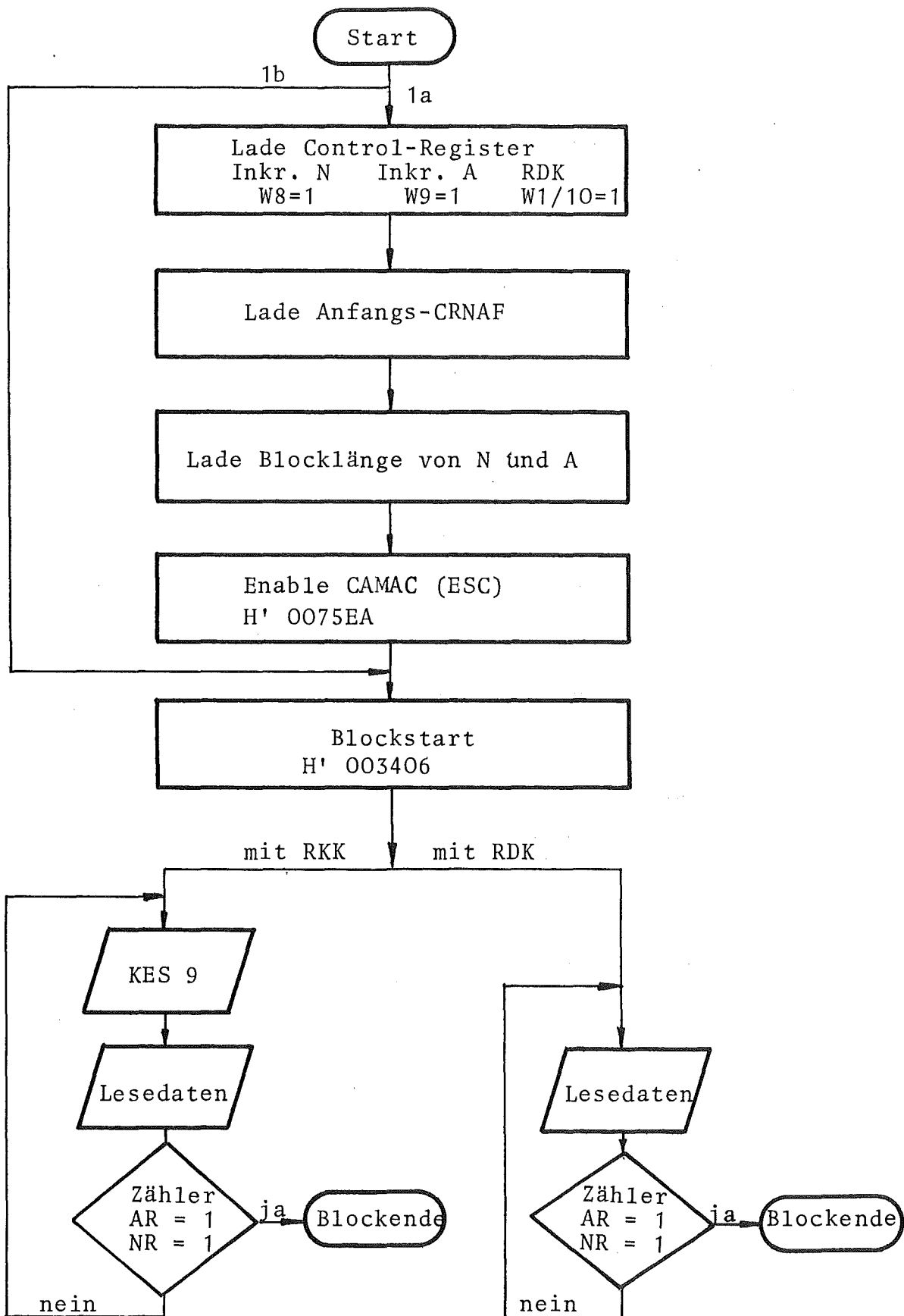


Fig. 22: Flußdiagramm Blocktransfer mit Inkrementieren von N und A

der Einstellphase heißt "Blockstart". Bei diesem Blocktransfer gibt es drei Möglichkeiten, die sich durch die Kombinationen der Bits IN und IA im Control-Register ausdrücken lassen, wie in der folgenden Tabelle 6 gezeigt.

Tabelle 6: Mögliche Typen des Blocktransfers IN, IA Mode

	Bit 8 (IN)	Bit 9 (IA)
1	0	1
2	1	0
3	1	1

Betrachtet wird Fall 3, in dem die beiden anderen enthalten sind. Der Block startet beim ersten Datentransfer (BF = 0, OP = 1) an der Stelle, die von dem Anfangs-CRNAF angegeben wird. Der Transfer selbst ist vom Typ 3 oder 4 der Tabelle 5 je nach Richtung. Zur Taktzeit SS2 wird die Subadresse A jeweils um 1 erhöht und damit ein modifizierter Befehl für den nächsten Transfer vorbereitet. Gleichzeitig mit der Erhöhung von A wird auch der Zähler für die Anzahl der Transfers in A um 1 erhöht. Erscheint nach dieser Erhöhung der Overflow AR (A Transfers Ready) dieses Zählers, dann wird während des nächsten Transfers folgendes erledigt:

- 1 Die Subadresse A wird mit SSO durch erneuten Zugriff zum Vorratsspeicher für den Anfangs-CRNAF auf den Anfangswert zurückgesetzt. Falls auch in N eine Erhöhung nötig ist, geschieht das mit SS2, aber nur wenn AR erschienen ist.
- 2 Der Zähler für die Anzahl der Transfers in A wird erneut aus seinem Vorratsspeicher geladen (SSO).
- 3 Die Fertigmeldung AR wird gelöscht (SS1). In dieser Weise geht es fort, bis beide Zähler für die Anzahl der Transfers ihre Overflow-Signale produzieren.

Aus diesen Meldungen wird das Signal "Blockende" (BE) hergeleitet und dem Rechner mit der Rückmeldung ZM des letzten Transfers gegeben. Der Rechner braucht bei dieser Methode also nicht selbst buchzuführen, sondern kann auf das BE-Signal warten. Durch die automatische Bildung der Adresse ergibt sich eine Ersparnis in Input/Output-Operationen von rund 50 %.

Da diese Betriebsart nicht die höchste Dringlichkeitsebene belegt, kann sie durch höherwertige Anforderungen unterbrochen werden. Wurde im Zuge einer zwischenzeitlichen Operation das Befehlsregister des System Controller benutzt, so ist der Hauptzustand Blocktransfer zerstört. Der Rechner muß ihn wieder herstellen durch erneuten Blockstart (BS). Für den Fall, daß ein Block nicht regulär beendet, sondern unterbrochen wurde, bewirkt das erneute Signal BS eine Weiterführung an der Stelle, wo die Unterbrechung stattfand. Es geht keine Information verloren.

#### 4.1.7. Blocktransfer Q Mode

Unter Q Mode verstehen wir verschiedene Methoden des Blocktransfers mit Hilfe des Response-Signals Q. Wie der Status von Q in den verschiedenen Modes ausgewertet wird, zeigt Tabelle 7.

Tabelle 7: Blocktransfers Q Mode

Response	Address Scan	Repeat	Stop
Q = 1	Register vorhanden	Register bereit	im Block
Q = 0	" fehlt	" nicht bereit	Blockende



#### 4.1.7.1. Blocktransfer im Address Scan Mode

Befinden sich in einem CAMAC Crate Module mit unterschiedlicher Anzahl von belegten Subadressen, ist die Abarbeitung als Block nur im Address Scan Mode (ADS) möglich. Dieser Mode paßt sich der jeweiligen Subadressenlänge automatisch an. Die Abarbeitung des Blocks beginnt immer mit der Adresse "Anfangs-CRNAF". Der System Controller inkrementiert automatisch nach jedem Transfer die Subadresse. Wird eine Subadresse angesprochen, die nicht belegt ist ( $Q = 0$ ), wird die Stationsnummer inkrementiert und die Subadresse auf  $A(0)$  zurückgesetzt. Dieser Leer-Transfer läuft im System Controller ab, wird aber nicht im Radar-Daten-Kanal übertragen, er bringt nämlich keine gültigen Daten.

Die Vorbereitung zum Address Scan Mode ist ersichtlich aus dem Flußdiagramm (Fig. 23 und Fig. 24). Die Erzeugung des Blockende-Signales ( $BE = 1$ ) ist auf zwei Arten möglich:

- a) nach Erreichen der Adresse  $CR(x) \cdot N(23) \cdot A(16)$
- b) nach Abarbeitung der vorgewählten Transfer-Anzahl

#### 4.1.7.2. Blocktransfer im Repeat Mode

Sollen größere Mengen von Daten über ein CAMAC-Modul mit Geräten ausgetauscht werden, die nicht CAMAC-synchron arbeiten, z. B. bei Rechner-Rechner-Kopplung, kann der Repeat Mode (RPM) benutzt werden. Der System Controller benutzt dabei den bereits besprochenen Anfangs-CRNAF als Adresse. Er regeneriert nun laufend diese Adresse; wird sie mit  $Q = 1$  beantwortet, läuft ein gültiger Transfer ab, bei  $Q = 0$  ist der Transfer ungültig und wird nicht übertragen.

Das Blockende-Signal wird erzeugt bei Erreichen der vorgewählten Anzahl gültiger Transfers (s. Flußdiagramm Fig. 25).

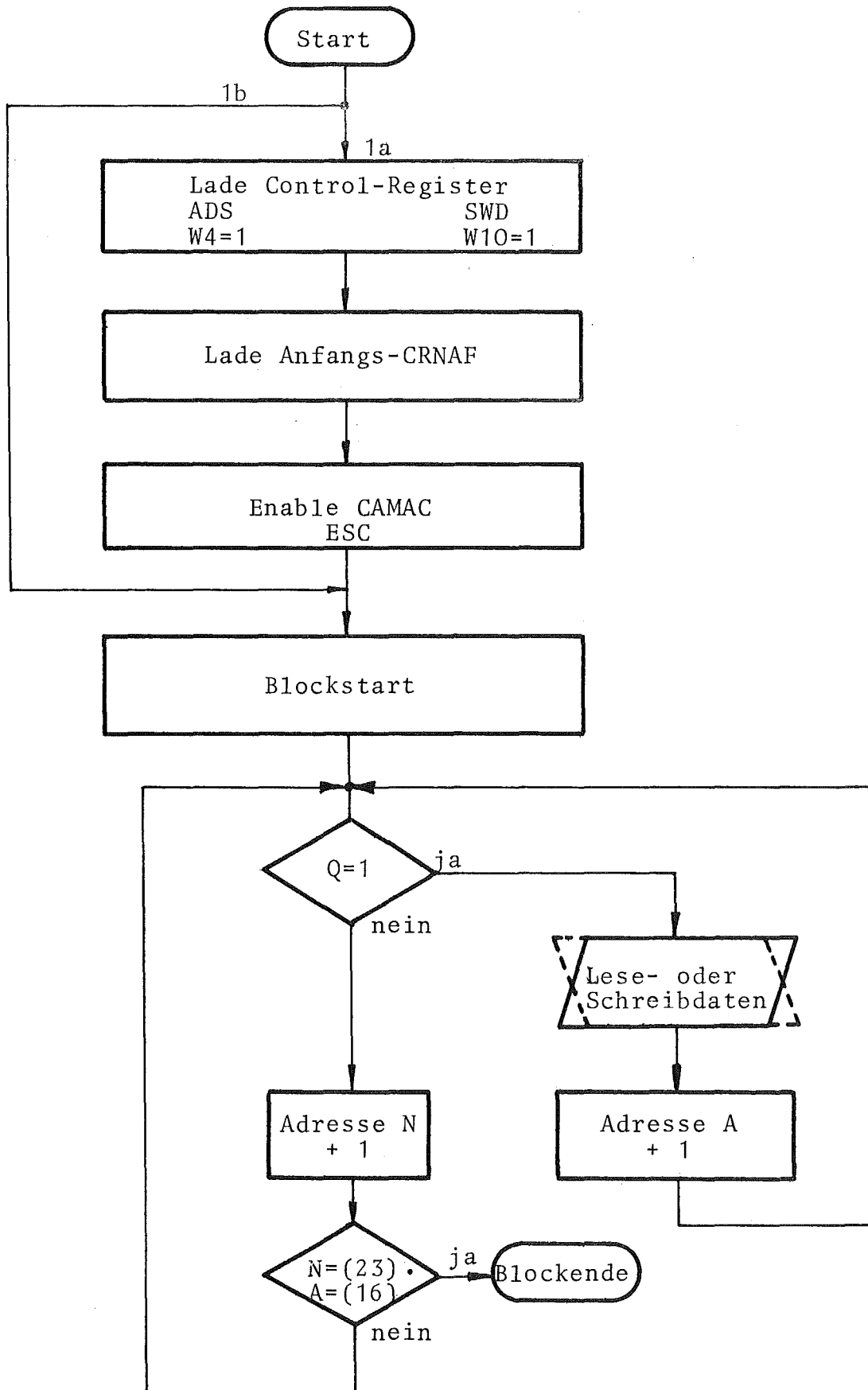


Fig. 23: Flußdiagramm Blocktransfer im Address Scan Mode

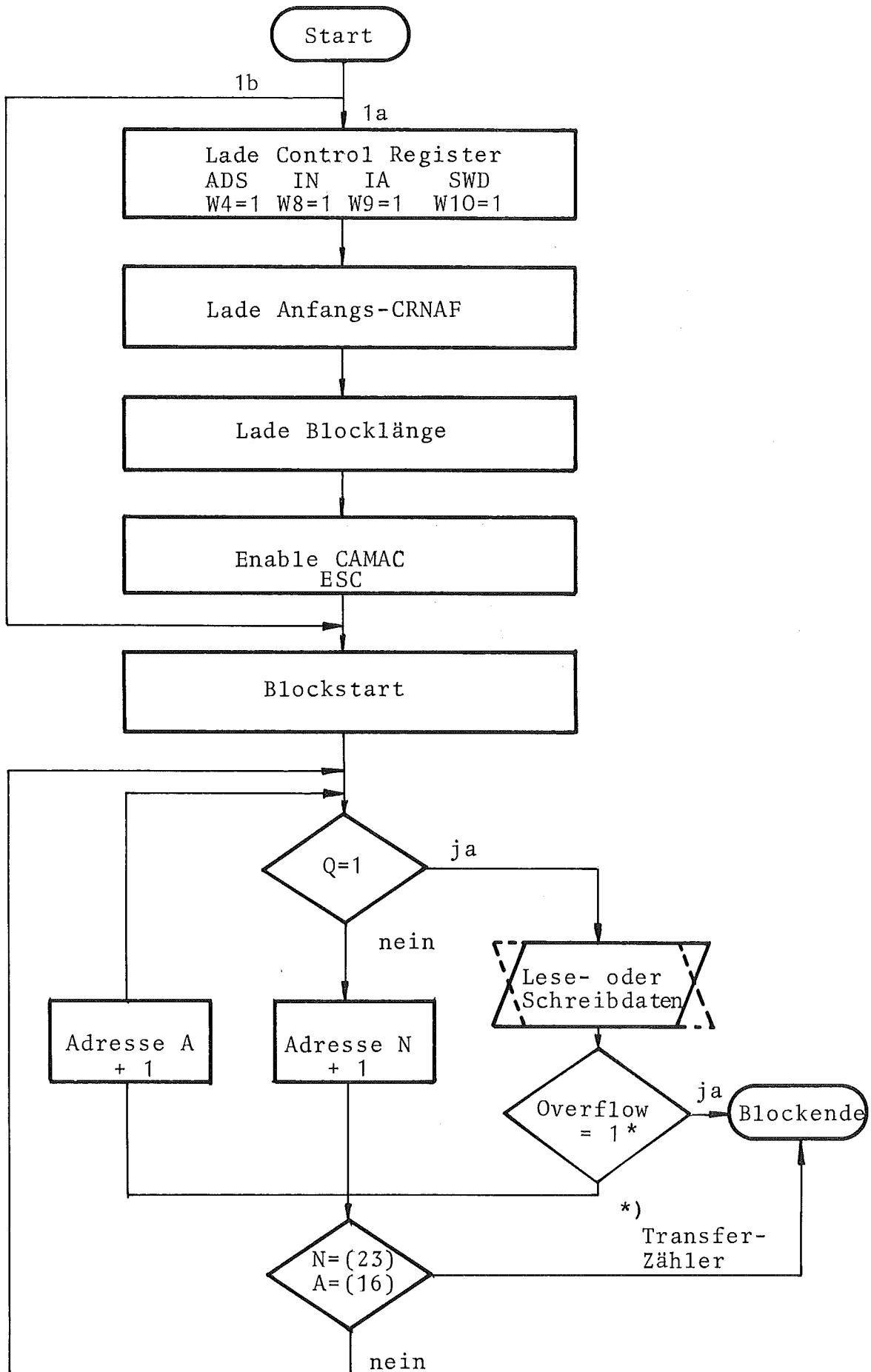


Fig. 24: Flußdiagramm Blocktransfer im ADS Mode mit vorgewählter Blocklänge

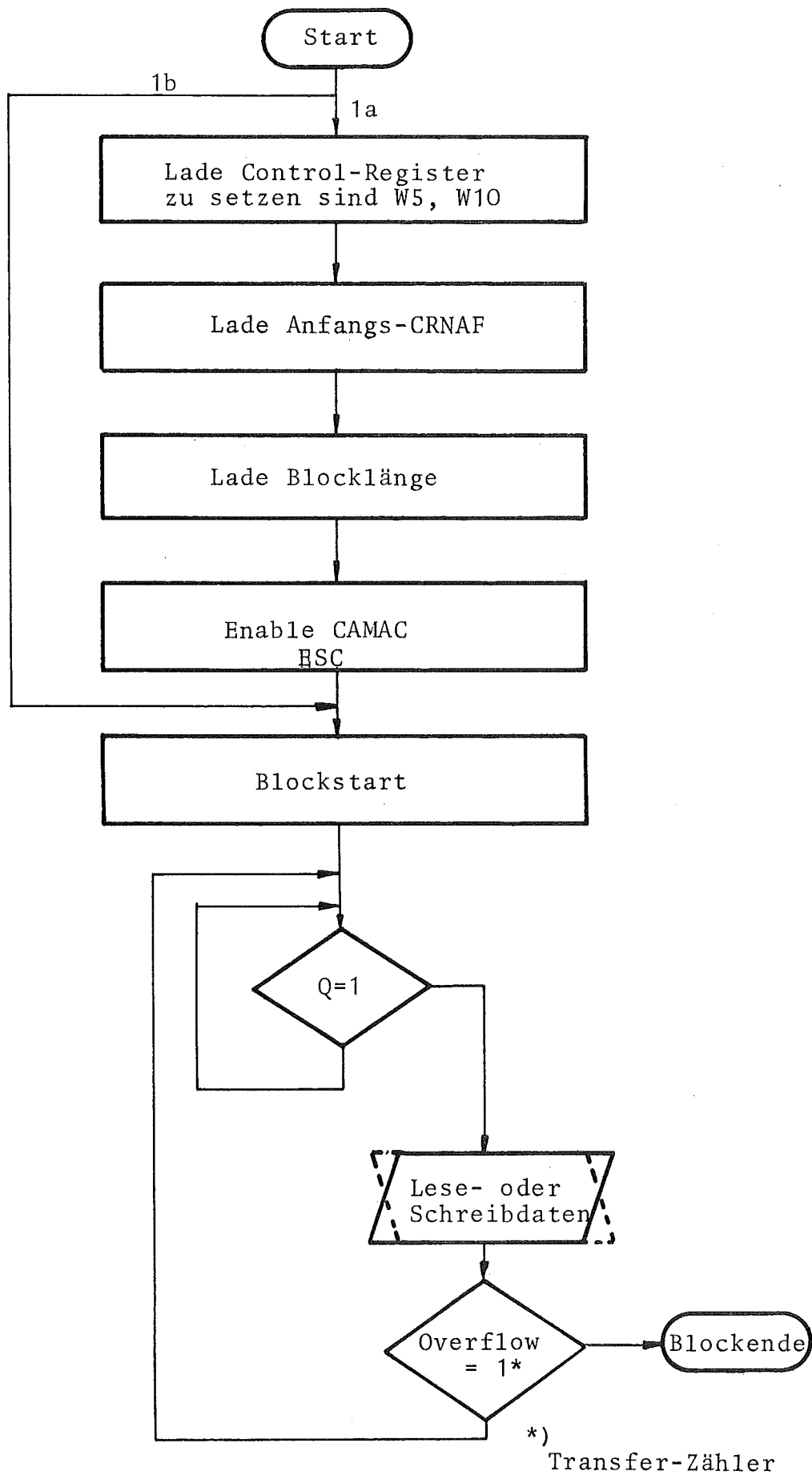


Fig. 25: Flußdiagramm Blocktransfer im Repeat Mode

#### 4.1.7.3. Blocktransfer im Stop Mode

Der Stop Mode (STM) arbeitet wie der Repeat Mode, nur muß hier gewährleistet sein, daß der abzuarbeitende Modul seine Daten schneller zur Verfügung stellt als der Abarbeitungszyklus des System Controller abläuft. Das Blockende-Signal wird durch  $Q = 0$  erzeugt (s. Flußdiagramm Fig. 26).

Eine Mischung von Repeat Mode und Stop Mode ist möglich, wenn nur eine vorbestimmte Anzahl gültiger Transfers übertragen werden soll, aber der Block auch kürzer sein kann. In diesem Falle kann die genaue Blocklänge aus dem Transferzähler mit Hilfe des Befehles "Lies Status" ausgelesen werden (s. Flußdiagramm Fig. 27).

#### 4.2. Modul-zu-Modul-Transfer

Der Modul-zu-Modul-Transfer ist die Betriebsart niedrigster Priorität. Er nimmt insofern eine Sonderstellung ein, als er vom Rechner zwar in Gang gesetzt, dann aber nicht weiter überwacht wird, sondern eigengesetzlich abläuft. Es ist sozusagen ein autonomer Transfer mit den Mitteln der normalen Befehls-Operationen. In Fig. 28 sind alle dabei notwendigen Einzelschritte gezeigt. Dabei wurde unterstellt, daß eine Zählerbatterie ausgelesen wird und die gelesenen Werte alle an dieselbe Stelle transportiert werden, von wo aus sie z. B. auf einem Bildschirm sichtbar gemacht werden sollen. Diese Betriebsart kann nach jedem einzelnen Zyklus des System Controller unterbrochen werden. Bei zwischenzeitlicher Benutzung des Befehlsregisters bricht sie zusammen und muß durch den Befehl "BS" wiederhergestellt werden. Dann geht der Modul-zu-Modul-Transfer an der Stelle weiter, wo er unterbrochen wurde.

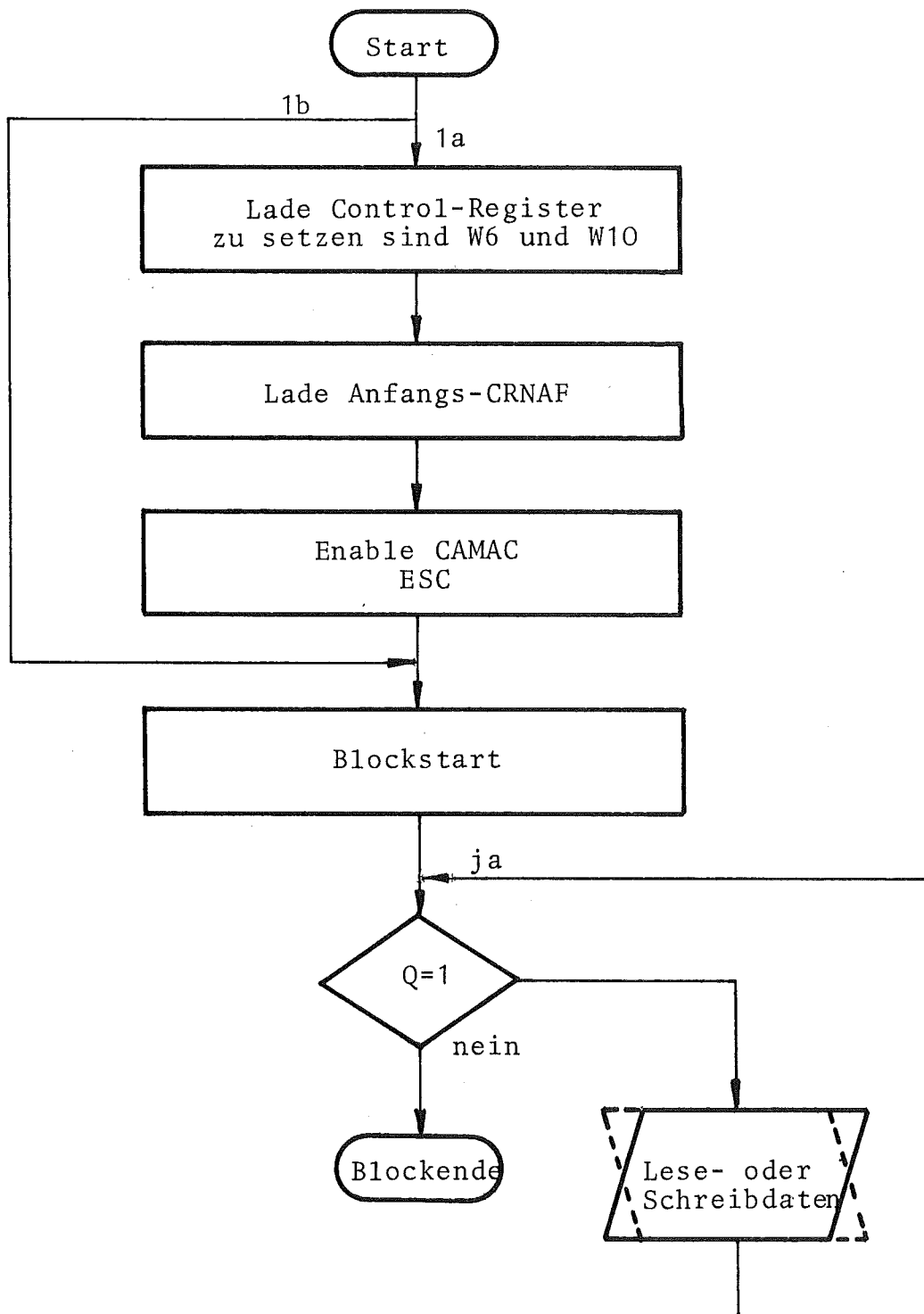


Fig. 26: Flußdiagramm Blocktransfer im Stop Mode

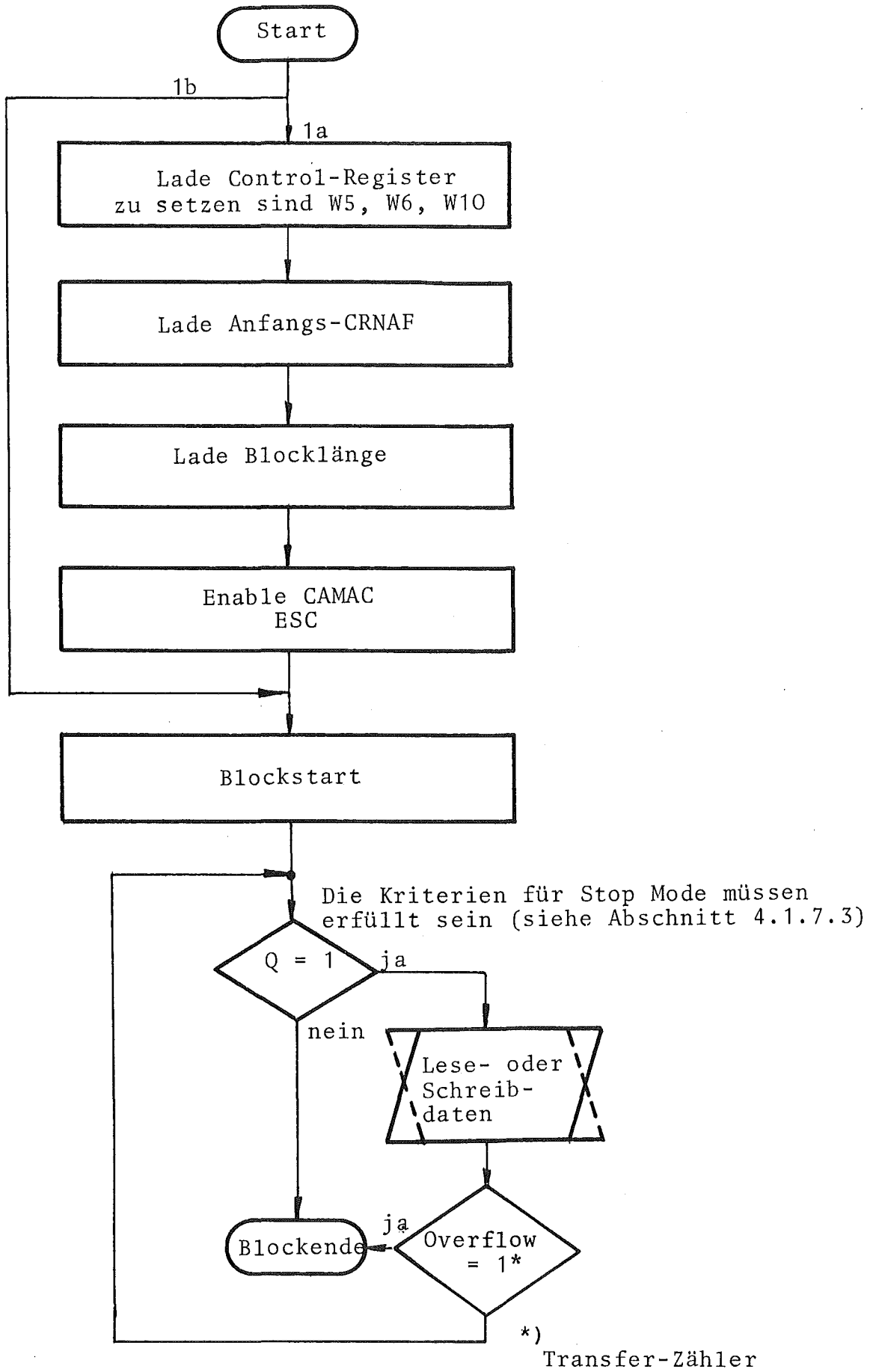
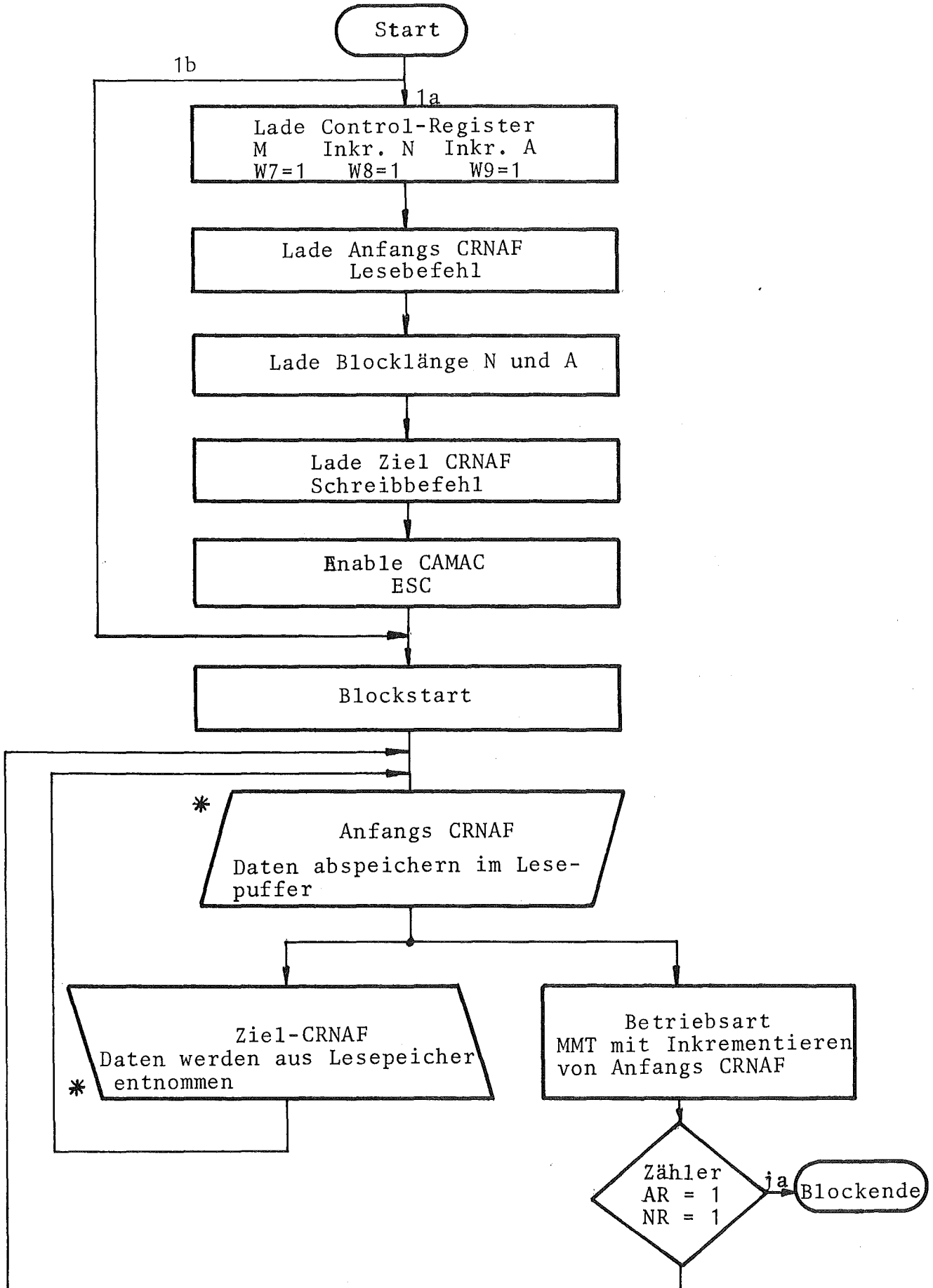


Fig. 27: Flußdiagramm Repeat Mode und Stop Mode



\* Dialog zwischen CAMAC und System Controller ohne Rechner

Fig. 28: Flußdiagramm Modul-zu-Modul-Transfer



### 4.3. Alarm-Verarbeitung

Über das Karlsruher Gesamtkonzept der Alarm-Verarbeitung wurde in einem eigenen Beitrag berichtet [4]. Hier sollen nur die Aktivitäten des System Controller behandelt werden. Die durch eine Alarm-Operation herangeschafften "Demand"-Signale werden verschieden interpretiert. Der System-Ingenieur kann entscheiden, ob ein L-Signal zu einer Programm-Unterbrechung oder zu einem autonomen Transfer führen soll. Die Entscheidung wird im LAM-Grader gefällt. Den nachgeschalteten Verarbeitungen geht aber in jedem Falle eine BD/BG-Operation voraus. In Fig. 29 ist der zeitliche Ablauf einer solchen Operation aufgezeichnet. Diese Operation ist im Effekt eine Multi-Crate-Lese-Operation an allen dialogfähigen Crates. Die Unterschiede zu einer Lese-Operation sollen aber sehr klar herausgestellt werden:

- 1 Der normale 17-Bit-Befehl CRNAF ist ersetzt durch ein einziges Signal, nämlich BG.
- 2 BG bewirkt im Crate keine normale Datenweg-Operation, d. h. es wird weder B noch S1 noch S2 erzeugt. (Nach der Vorschrift müßten ja die L's beim Erscheinen von B vom Datenweg verschwinden.)
- 3 Der System Controller muß neben BG auch die Adreß-Signale BCRn aller dialogfähigen Crates selbst zu Beginn der Operation bilden.

Zum Ende dieses Zyklus sind irgendwelche Bits in dem Alarmregister gesetzt, dabei führen die unteren 8 zu Programm-Unterbrechungen, die oberen 16 zu autonomen Transfers.

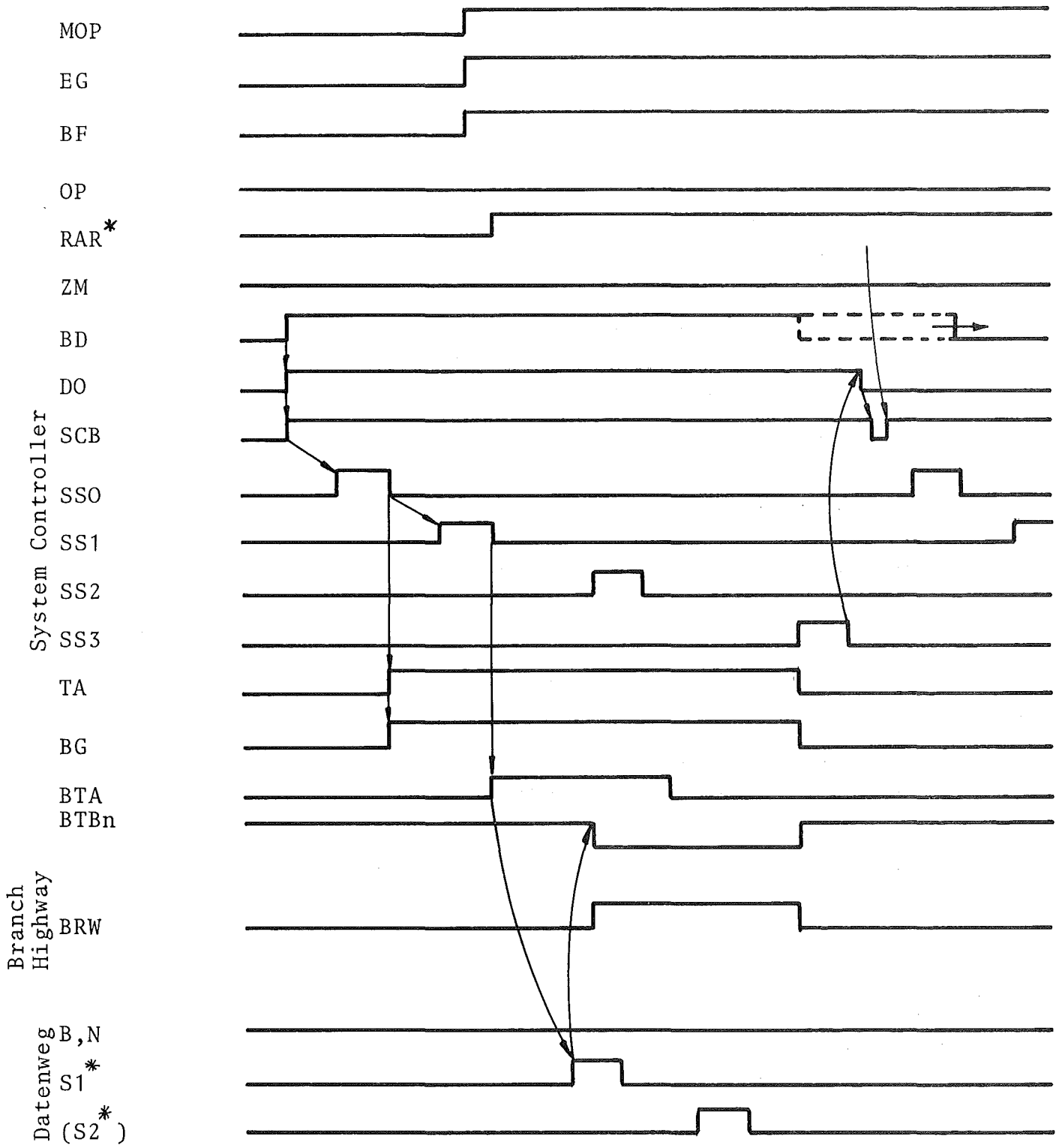


Fig. 29: Timing-Diagramm für BD/BG-Operationen

#### 4.3.1. Interrupt-Behandlung

Die Abarbeitung von Interrupt-Requests geht in der von der Rechner-technik bekannten Art vor sich. Sobald der Rechner frei ist, den Interrupt zur Kenntnis zu nehmen, springt das Programm in die der Interrupt-Ebene zugeordnete Subroutine, in der vielleicht zuerst Schritte zur Identifizierung der eigentlichen Alarmquelle ausgeführt werden müssen, je nachdem ob sich hinter einem IT-Signal viele oder nur eine Quelle verbergen. Die weiteren Schritte der Routine dienen gewöhnlich der Durchführung vorgeplanter Maßnahmen zur Beseitigung der Alarmursache. Dadurch wird die Alarmsituation bereinigt, und als letzter Schritt wird der entsprechende Speicher des Alarmregisters unter der Wirkung des Befehles LIT bei 1-Zustand der gleichnamigen W-Leitung selektiv gelöscht.

#### 4.3.2. Autonome Transfers (Eingabe)

Werden durch eine BD/BG-Operation die Bits 9 bis 24 des Alarmregisters gesetzt, dann werden sie als Anforderungen für autonome Transfers betrachtet. Es gelten für diesen Fall die Figuren 30 und 31. Der autonome Transfer in Eingabe-Richtung ist bei diesem System Controller der Hauptzustand höchster Priorität und damit nicht unterbrechbar. In Fig. 30 ist angenommen, daß zwei Requests gleichzeitig aufgetreten sind, und zwar a24 und a22 (Bits 24 und 22 des Alarmregisters). Durch diese geometrische Ordnung ist auch eine Priorität festgelegt, beginnend mit der höchsten Stufe (Bit 24). Das Oder aller 16 möglichen Requests bildet das Signal RQATE (RequAuTonomer Transfer Eingabe). Die Anwesenheit dieses Signals verhindert die anderweitige Benutzung des System Controller. Es etabliert sich der Hauptzustand ATE. Unter seiner Kontrolle startet ein Suchgenerator, der die einzelnen Bits des Alarmregisters mit Signalen SCK, beginnend mit SC24, abfragt. Im angegebenen Beispiel findet er schon beim ersten Schritt einen gesetzten Speicher. Der Scanner stoppt auf dieser Stellung, und es entsteht das Signal AC24 (Autonomer CRNAF Nr. 24,

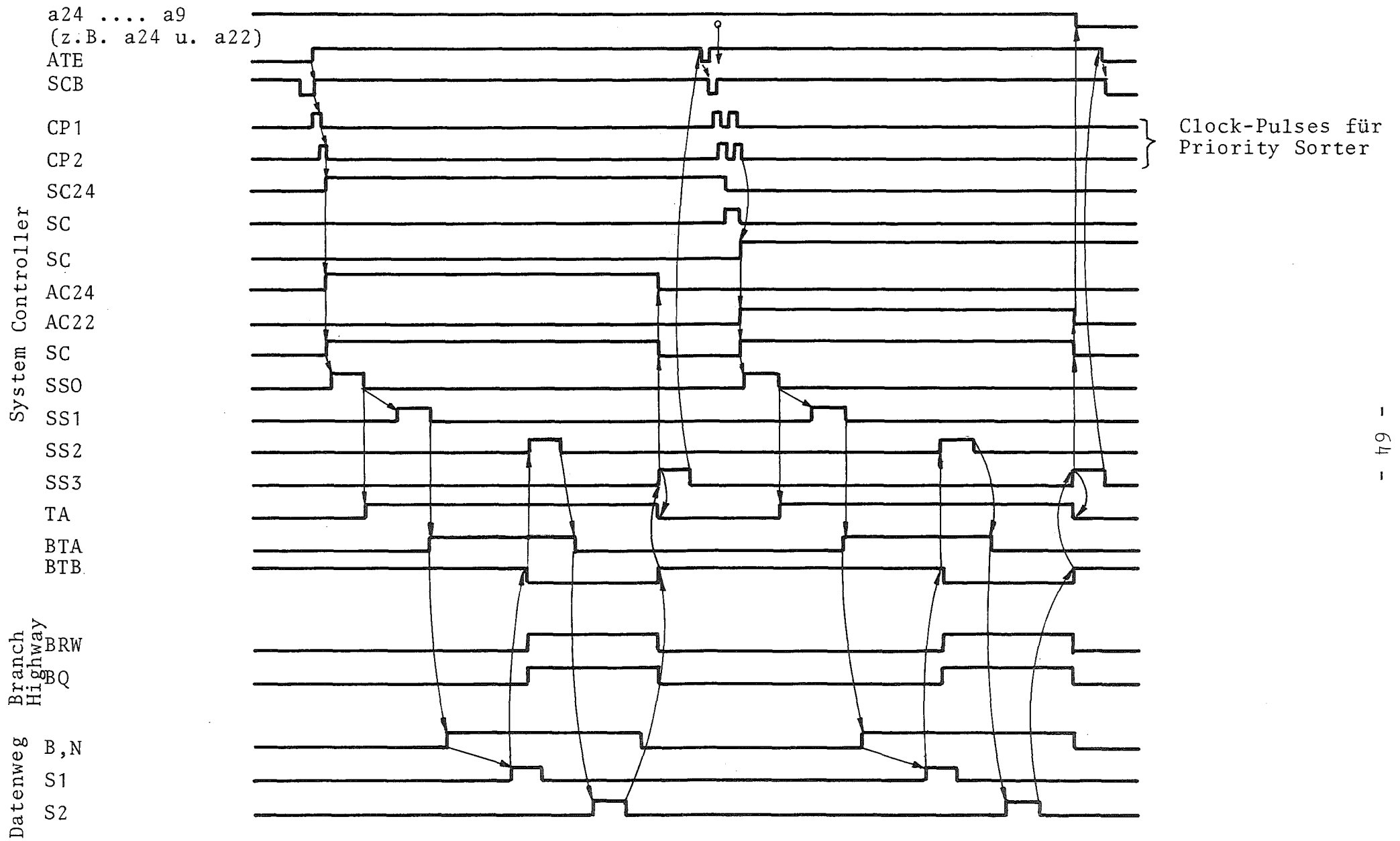


Fig. 30: Timing für autonome Eingabe

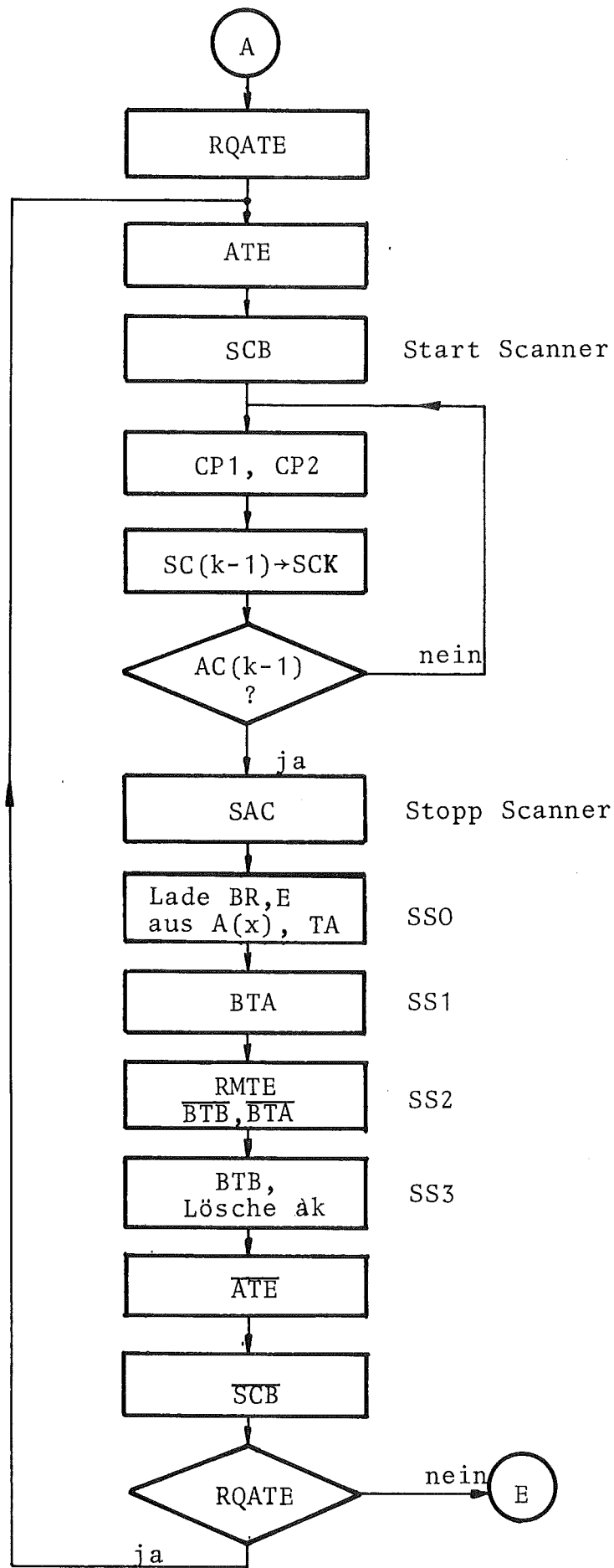


Fig. 31: Signalfluß-Schema für autonome Eingabe

d. h. für das Bit Nr. 24). Von jedem ACK-Signal wird dasselbe Start-Signal SAC für den Timing-Generator (Eingang Selbststart) abgeleitet. Zur Taktzeit SSO wird das von AC24 aufgerufene Befehlswort aus dem Vorratsspeicher auf die Befehlsleitungen gegeben. Es läuft eine normale Lese-Operation ab. Bei dieser Operation wird die Anforderung a24 im Alarmregister mit der Vorderflanke von SS3 zurückgesetzt. Die gelesenen Daten gelangen mit SS2 in den Lesebuffer, und von da werden sie auf die IB-Leitungen geschaltet. Dem Rechner wird durch das Signal CPA1/2 bekanntgegeben, daß Information bereitgestellt ist. Die Signale AC24 und SAC verschwinden mit der Vorderflanke von SS3, nicht aber RQATE, denn vereinbarungsgemäß sollte ja ein zweiter Request (a22) gleichzeitig erschienen sein. Durch RQATE = 1 wird unmittelbar nach dem Verschwinden von SCB, also sofort wenn der System Controller wieder frei ist, erneut ATE gebracht und der Address Scanner wieder gestartet. Er sucht so lange, bis er wieder einen Request findet. Es folgen so lange autonome Lese-Operationen, bis RQATE = 0 geworden ist.

Es soll noch eine Abschätzung des maximalen Zeitbedarfs für autonome Transfers in Eingabe-Richtung gegeben werden. Dazu werden folgende Annahmen zu Grunde gelegt:

- 1 Die Laufzeit auf dem Branch Highway sei 100 ns.
- 2 Die Scanner-Periode sei 100 ns (zwei Impulse zu je 50 ns).
- 3 Es seien alle 16 möglichen Anforderungen für autonome Transfers gekommen.

Dann errechnet sich der Zeitbedarf wie folgt:

1 x BD/BG-Operation	.....	3,5 µs
16 x 100 ns für Scanner	.....	1,6 µs
16 x 3,5 µs für Lese-CRNAF's	.....	56,0 µs
	<hr/>	
	Summe	61,1 µs

## 5. Testmöglichkeiten mit dem System Controller

Der System Controller bietet einige Möglichkeiten der Fehlersuche im on-line-Betrieb.

### 5.1. Benutzung des BQ-Signals

Nach den in [2] gegebenen Definitionen wird das Q-Signal im Datenweg nach folgenden Methoden erzeugt:

- a) Bei Lese- und Schreib-Operationen, wenn ein Modul in der Lage ist, den Befehl auszuführen;
- b) bei den Testfunktionen F(8) und F(27) dann, wenn die Frage bejaht werden soll;
- c) bei allen anderen Funktionen ganz nach dem Belieben des Modul-Entwicklers.

BQ ist also für Testzwecke kein sehr leistungsfähiges Signal. Es bleibt nichts anderes übrig, als das BQ-Signal dem Rechner unverändert zur Verfügung zu stellen (Q=0=Fehlfunktion).

### 5.2. Lesen des Statusregisters

Die Analyse der Informationen aus dem Statusregister kann folgende Gründe für eine Fehlfunktion ergeben:

- 1 Der System Controller war nicht enabled (ESC = 0).
- 2 Das angesprochene Crate war nicht dialogfähig, weil es entweder nicht vorhanden ist, keine Stromversorgung hat oder off-line ist (BTB = 0).
- 3 Der Interrupt IT8 ist gekommen, weil eine Lese- oder Schreib-Operation kein Q = 1 gebracht hat

- 4 Der Interrupt IT8 ist gekommen, weil eine Operation länger als die (über den Time-out-Generator) zugestandene längste Zeit gebraucht hat.
- 5 Der Betriebsarten-Wahlschalter steht nicht auf Stellung "Run".

### 5.3. Tests über die Frontplatten

Die bisher beschriebenen Tests laufen alle in der Betriebsart "Run", werden also vom Rechner durchgeführt. Der Operator kann ebenfalls bis zu einem gewissen Umfang Tests ausführen. Diesem Zweck dienen die Betriebsarten "Einzeloperation" und "Step". Besonders bei "Step" kann der Operator am Display-Teil der Frontplatten den Ablauf der einzelnen Operationen gleichsam mit der Lupe verfolgen und sich neben den Timing-Signalen die Inhalte vom Befehls-, Lesebuffer- und Alarmregister in Ruhe ansehen.

### Literatur

- [1] Ottes, J.G.  
CAMAC System Controller für CALAS-Endstelle  
Kernforschungszentrum Karlsruhe, Bericht KFK 1412, 1971
- [2] CAMAC - A Modular Instrumentation System for Data Handling -  
Revised Description and Specification  
Euratom-Bericht EUR 4100 e (Revised version 1972), Luxembourg 1972
- [3] CAMAC - Organisation of Multi-Crate Systems - Specification  
of the Branch Highway and CAMAC Crate Controller Type A  
Euratom-Bericht EUR 4600 e, Luxembourg 1972
- [4] Heep, W.; Ottes, J.; Tradowsky, K.  
Alarm-Verarbeitung und autonomer Datentransfer im CAMAC-System  
Kernforschungszentrum Karlsruhe, Externer Bericht 22/71-7, 1971



Anhang: Zusammenstellung benutzter Signalnamen, Subadressen etc.

Die nachfolgend gegebene Zusammenstellung der wichtigsten Kenngrößen soll das Arbeiten mit dem System Controller erleichtern, insbesondere dem Programmierer; sie sind aber möglicherweise auch bei der Inbetriebnahme und Fehlersuche von Nutzen.

### 1. Subadressen und Funktionen

Im System Controller werden beim derzeitigen Ausbau insgesamt 26 Subadressen gebraucht. Daher muß mit Funktionscodes der Gruppen 1 und 2 (s. Tabelle IV in [2]) gearbeitet werden. So werden 16 Lesebefehle für autonome Transfers in 16 Vorratsspeichern zu je 20 Bit mit den Subadressen A(0) bis A(15) gehalten. Diese Speicher werden nur überschrieben mit F(17).

#### Subadressen mit Funktionen der Gruppe 2

A(0)	}	Vorratsspeicher für autonome CRNAF's F(17)
.		
.		
.		
.		
.		
.		
.		
.		
.		
.		
.		
.		
.		
.		
A(15)		

#### Subadressen mit Funktionen der Gruppe 1

A(12)	Statusregister	F(1)
A(11)	Control-Register	F(16); F(1)
A(14)	GL-Register (Alarmregister)	F(8) · (16); F(16)
A(10)	Speicher für Anfangs-CRNAF	F(16)

A(9)	Speicher für Ziel-CRNAF	F(16)
A(8)	Pufferspeicher für Anzahl der N- und A-Transfers	F(16)
A(6)	Start Blocktransfer	F(26)
A(7)	Select Crate Register (SCR)	F(16)
A(0)	1-Bit-Speicher für ESC	F(24); F(26)
A(1)	Clear und BZ-Erzeugung	F(26)

Der System Controller hat die Crate-Adresse CR(0).

## 2. Befehlsliste

Befehle mit Funktionen der Gruppe 2

$CR(0) \cdot N(0) \cdot \left\{ \begin{array}{c} A(0) \\ A(1) \\ \cdot \\ \cdot \\ \cdot \\ A(15) \end{array} \right\} \cdot F(17)$       Einspeichern der  
autonomen CRNAF's

Befehle mit Funktionen der Gruppe 1

CR(0) · N(0) · A(12) · F(1)	Lies Statusregister und Transferzähler ("RST")
CR(0) · N(0) · A(10) · F(16)	Lade Speicher für Anfangs-CRNAF
CR(0) · N(0) · A(6) · F(26)	Blockstart ("BS")
CR(0) · N(0) · A(9) · F(16)	Lade Speicher für Ziel-CRNAF
CR(0) · N(0) · A(8) · F(16)	Lade Pufferspeicher für Anzahl der N- und A-Transfers

Anmerkung: N- und A-Teil müssen gleichzeitig mit dem 2-Komplement der Anzahl der gewünschten Transfers geladen werden.

CR(0) · N(0) · A(7) · F(16)	Lade Select Crate Register SCR mit Crate Pattern
CR(0) · N(0) · A(0) · F(26)	Enable System Controller ("ESC")
CR(0) · N(0) · A(0) · F(24)	Disable System Controller (" $\overline{\text{ESC}}$ ")
CR(0) · N(0) · A(1) · F(26)	Clear System Controller (CLR") und erzeuge BZ
CR(0) · N(0) · A(11) · F(1)	Lies Control-Register
CR(0) · N(0) · A(11) · F(16)	Lade Control-Register
CR(0) · N(0) · A(14) · ( $\overline{\text{F8}}$ · $\overline{\text{F16}}$ )	Lies GL-Register
CR(0) · N(0) · A(14) · F(16)	Lösche GL-Register Bits 1 - 8 ("LIT")
CR(0) · N(0) · A(0-15) · F(16)	Lade DMA-Befehlsregister

### 3. Liste der verwendeten Abkürzungen und Signalnamen

ACK ( $9 \leq k \leq 24$ )	Autonomer CRNAF (Erzeugung von SAC)	I2N	Strobe zum Inkrementieren von Zähler und Adreßfortschaltung N
ADS	Address Scan Mode	L	Hauptzustand BATE oder SQ
AN <sup>+</sup>	Sammelsignal aller Anforderungen	LAC	Lade Befehlsregister aus dem Speicher für Anfangs-CRNAF
ANBATE	Anforderung Block autonomer Transfers in Eingabe-Richtung	LITm ( $1 \leq m \leq 8$ )	Lösche Speicher für Interrupt-Anforderung ITm im GL-Register
ANF	Anforderung vom Rechner zur Command Operation (durch RAR*)	LZA	Lade Transferzähler A
ANSQ	Anforderung sequentieller Operationen im Q Mode	LZC	Lade Befehlsregister aus dem Speicher für Ziel-CRNAF
AR	A Transfers Ready (Overflow des A-Zählers)	LZN	Lade Transferzähler N
AT	Autonomer Transfer	M	Anforderung Modul-zu-Modul-Transfer
ATA	Autonomer Transfer in Ausgabe-Richtung	ma	Memory-Adresse von DMA-CRNAF
ATE	Autonomer Transfer in Eingabe-Richtung	MAN	Manueller Einzeltakt
AUSL	Auslösen von SS2 oder SS3	MF	Malfunktion
b	Hilfssignal	MMT	Modul-zu-Modul-Transfer
BATE	Block autonomer Transfers in Eingabe-Richtung	MOP	Mit Operand
BE	Blockende	NORMTA	Clear manuell
BF	Befehl	NR	N Transfers Ready (Overflow des N-Zählers)
BR	Befehlsregister	OP	Operand
BS	Blockstart (statisch im Block)	P4,P5	Hilfstakte im Timing-Generator
BSP	Blockstart (Impuls zu Beginn des Blockes)	POP	Betriebsart Einzeloperation
bqg	Gespeichertes Branch Q	PRUN	Betriebsart Run
c	Hilfssignal	RAR	Reset AR
CL	Clear vom Rechner (falls vorhanden)	RAR*	Recheraufruf (Computer Driver)
CLR	Clear-Signal für System Controller	rm	Rückmeldung
CO	Command Operation	RM <sup>+</sup>	Rückmeldung von allen Betriebsarten
CONT	Continue (wird generiert beim Wiederstarten eines unterbrochenen Blockes)	rmte	Rückmeldung Transfer Eingabe
d	Hilfssignal	RNR	Reset NR
DIRDK	Disable Radar-Daten-Kanal	RPM	Repeat Mode
DO	Demand Operation	RQATA	Request ATA
e	Hilfssignal	RQATE	Request ATE
E	Eingabe (BR Bit 20)	RST	Read Status
E <sup>+</sup>	Eingabe (MMT)	SAC	Start autonomer CRNAF
eg	Eingabe (Rechner)	SCB	System Controller Busy
ENATA	Enable ATA	Sck ( $9 \leq k \leq 24$ )	Address Scanner steht auf Stufe k
ENATE	Enable ATE	SCR	Select Crate Register
ENBATE	Enable Block autonomer Transfers in Eingabe-Richtung	SCEN	Enable Scanner für DMA-CRNAF
ENIT8	Enable Interrupt 8 (System Controller)	SPE	Blockstartpuls "Ende"
ENITS	Enable Interrupts Crate 1 bis 7	SQ	Blocktransfer im Q Mode (Address Scan, Repeat oder Stop Mode)
ENRDK	Enable Radar-Daten-Kanal	SSO ... 3	Uhrimpulse des Timing-Generators
ESc	Enable System Controller (Ohne dieses Signal kann der SC nicht mit dem Branch verkehren.)	STBR	Strobe für Befehlsregister
FF	Fehlfunktion (Auslösung von IT8)	STM	Stop Mode
G	CAMAC-Lesebefehl	SWD	Switch Data, umschalten auf RDK
H	CAMAC-Schreibbefehl	TA	Timing-Signal für Branch-Driver
IA	Inkrementiere Subadresse A	TO	Time Out
IB	Informations-Bus	UB	Unerlaubte Betriebsart
IN	Inkrementiere Stationsnummer N	WSCR	Write in Select Crate Register
ITm ( $1 \leq m \leq 8$ )	Interrupt-Anforderung	ZM	Zeichenmeldung (Timing-Signal zum Rechner)
I2A	Strobe zum Inkrementieren von Zähler und Adreßfortschaltung A		