

**KERNFORSCHUNGSZENTRUM
KARLSRUHE**

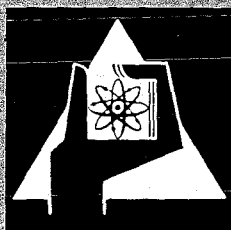
Juni 1971

KFK 1412

Labor für Elektronik und Meßtechnik

CAMAC System Controller für CALAS-Endstelle

J. G. Ottes



GESELLSCHAFT FÜR KERNEFORSCHUNG 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

Juni 1971

KFK 1412

Labor für Elektronik und Meßtechnik

CAMAC System Controller für CALAS-Endstelle

von

J. G. Ottens

Zusammenfassung

Es wird ein CAMAC System Controller beschrieben, der einen voll ausgebauten CAMAC Branch handhaben kann und für den Anschluß an eine CALAS-Datenendstelle ausgelegt ist. Mit dem System Controller lassen sich Einzelbefehle, Blocktransfers, Modul-zu-Modul-Transfers sowie autonome Datentransfers in beiden Richtungen im Cycle-Stealing-Verfahren abwickeln. Durch relativ geringfügige Änderungen innerhalb einer Baugruppe läßt sich der Controller umrüsten zum Anschluß anderer Rechner.

Abstract

A CAMAC System Controller capable of driving a full CAMAC Branch in conjunction with a CALAS Terminal is presented. Single operations, block transfers, modul-to-modul-transfers and autonomous transfers in both directions on a cycle-stealing basis can be handled by the Controller. The device can be adapted to other computers by changing some parts of its computer peripheral electronics.

Inhaltsverzeichnis

	Seite
1. Einleitung	4
2. Nahtstelle zum CALAS-System	5
3. Hauptfunktionsgruppen des System Controller	9
3.1. Rechnerperipherie	9
3.2. Branch Driver	10
3.3. Interne Logik	10
3.3.1. Befehlsregister	10
3.3.2. Bildung der Crate-Adresse	12
3.3.3. Daten-Leitungen	14
3.3.4. Timing-Generator	14
3.3.5. Hauptzustands-Generator	16
3.3.6. Statusregister	18
3.3.7. Alarmregister	18
3.3.8. Anzeige- und Bedienungs-feld	19
3.3.9. Blocktransfer-Einrichtung	19
3.3.10. Modul-zu-Modul-Transfer	21
3.3.11. Autonomer Transfer in Eingabe-Richtung	22
3.3.12. Autonomer Transfer in Ausgabe-Richtung	23
4. Funktionsabläufe	24
4.1. Befehls-Operationen	24
4.1.1. Dialog Rechner - System Controller ohne Operand	25
4.1.2. Dialog Rechner - System Controller mit Operand	25
4.1.3. Einadreß-Lesebefehle	27
4.1.4. Einadreß-Schreibbefehle	28
4.1.5. Befehle ohne Operand	28
4.1.6. Blocktransfer	29
4.2. Modul-zu-Modul-Transfer	30
4.3. Alarm-Verarbeitung	31
4.3.1. Interrupt-Behandlung	32
4.3.2. Autonome Transfers (Eingabe)	32
4.4. Autonome Transfers (Ausgabe)	34

	Seite
5. Testmöglichkeiten mit dem System Controller	34
5.1. Benutzung des BQ-Signals	34
5.2. Prüfen des Befehlsregisters	35
5.3. Lesen des Statusregisters	35
5.4. Tests über die Frontplatte	36
Literatur	37
Anhang: Zusammenstellung benutzter Signalnamen, Subadressen etc.	64
1. Subadressen und Funktionen	64
2. Befehlsliste	65
3. Liste der Signalnamen	67

1. Einleitung

System Controllers für das CAMAC-System [1] haben ganz allgemein die Aufgabe, einem Rechner oder einer anderen digitalen Steuereinheit den Zugriff zum Branch Highway zu ermöglichen. Der planende Ingenieur hat dabei zunächst grundsätzlich zu entscheiden, wieviel Fähigkeiten er in das Gerät hineinbauen möchte. Als allgemeine Richtlinie sei gesagt, daß man dem Rechner soviel wie möglich überlassen und gleichzeitig den System Controller entsprechend knapp halten sollte. Falls also zyklische Bedienung durch das Programm ausreicht, dann sollte kein Zugriff zur Interrupt-Einrichtung eingebaut werden, und falls der Datentransfer per Programm abgewickelt werden kann, ist kein Anschluß an die Direktzugriffs-Kanäle erforderlich. Natürlich wird diese Vorentscheidung auch von dem in Frage stehenden Rechner und der Konfiguration des Gesamt-Systems beeinflußt. Im vorliegenden Falle handelt es sich um einen CAMAC System Controller, der an einer Datenendstelle des CALAS-Systems¹⁾ betrieben werden soll. Dabei kann ein zentraler Prozeßrechner (Telefunken, TR 86) über das CALAS-System mehrere Experimente bzw. Prozesse gleichzeitig fahren. Hier ist es natürlich angezeigt, mit einem Minimum von Ein- und Ausgabe-Operationen auszukommen. Aus diesem Grunde wurde z. B. eine Blocktransfer-Einrichtung in den Controller eingebaut; außerdem können schnelle Datentransfers als sogenannte autonome Transfers über einen gesonderten Kanal des CALAS-Systems stattfinden. Einen wesentlichen Beitrag zu dem angestrebten Ziel, den Rechner zu entlasten, liefert auch die Möglichkeit des Modul-zu-Modul-Transfers über den System Controller.

Die Durchführung eines Blocktransfers unter Programm-Kontrolle erspart durch eine automatische Befehls-Erzeugung im System Controller etwa die Hälfte der bei Einzel-Abwicklung nötigen

1)
CALAS - Computer Aided Labo-
ratory Automation System [2]

Manuskript eingegangen
am 23. 6. 1971

Transfers. Beim Modul-zu-Modul-Transfer ist der Rechner, abgesehen von der ursprünglichen Einstellung dieser Betriebsart, überhaupt nicht mehr in Aktion. Autonome Transfers finden auf der Basis des sogenannten Cycle Stealing statt und benötigen keine Programm-Assistenz. Der zu beschreibende System Controller kann bis zu 16 verschiedene Quellen autonomer Daten in einem CAMAC-System handhaben.

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", 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.

Durch die beiden Nahtstellen zum Branch Highway einerseits und zum Rechner andererseits und die gewünschten Fähigkeiten ist der System Controller weitgehend bestimmt. In Kapitel 2 wird die Nahtstelle zum CALAS-System beschrieben, in den folgenden die interne Logik des System Controller.

2. Nahtstelle zum CALAS-System

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

IO.i ($1 \leq i \leq 24$) (Information)

Tabelle 1: Leitungen der Nahtstelle zum CALAS-System

Benennung	Bedeutung	Anzahl der Leitungen
IO.i	Informations-Leitungen (bidirektional, für CAMAC-Befehle und Daten)	24
BF	CAMAC-Befehl	1
OP	Operand, CAMAC-Daten	1
BE	Blockende	1
ANF	Anforderung (Timing-Signal 1)	1
RM	Rückmeldung (Timing-Signal 2)	1
ITm	Interrupt-Anforderung	8
$\overline{\text{LIT}}$	Lösche nicht Interrupt	1
$\overline{\text{LITm}}$	Lösche nicht Interrupt-Anforderung ITm	8
$\overline{\text{CL}}$	Clear not	1
FF	Fehlfunktion	1
ATi	Informations-Leitungen autonomen Transfer (bidirektional)	24
ANFATE	Anforderung für autonomen Transfer in Eingabe-Richtung (Timing-Signal 3)	1
RMATE	Rückmeldung für autonomen Transfer in Eingabe-Richtung (Timing-Signal 4)	1
ANFATA	Anforderung für autonomen Transfer in Ausgabe-Richtung (Timing-Signal 5)	1
RMATA	Rückmeldung für autonomen Transfer in Ausgabe-Richtung (Timing-Signal 6)	1

Über die insgesamt 24 bidirektionalen Informations-Leitungen IO.i 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.

BF (Befehl)

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

OP (Operand)

Die Leitung führt "L", wenn es sich bei der Information auf den IO.i-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)

Der 0 → L-Übergang dieses vom Rechner kommenden Signals bedeutet den Beginn einer programmgesteuerten Operation.

RM (Rückmeldung)

Das Signal meldet dem Rechner den Vollzug des Befehles. Das Signalpaar ANF und RM verwirklicht ein Ping-Pong- bzw. Hand-Shake-Timing-System.

IT_m ($1 \leq m \leq 8$) (Interrupt-Anforderung)

Die Signale auf den Leitungen $IT_1 \dots 8$ zwingen den Rechner in Interrupt-Routinen. IT_1 bis IT_7 stehen den zugeordneten Rahmen des CAMAC-Systems zur Verfügung, über IT_8 kann der System Controller selbst das Rechnerprogramm unterbrechen.

\overline{LIT} (Lösche nicht Interrupt)

Nach der Erledigung der von einem Interrupt-Signal geforderten Operationen bereitet der Rechner mit diesem Signal die Rücksetzung des oder der IT 's vor. Die wirkliche Ausführung der Löschung erfolgt gezielt durch das Signal \overline{LIT}_m .

\overline{LIT}_m ($1 \leq m \leq 8$) (Lösche nicht Interrupt-Anforderung IT_m)

Der Interrupt-Request IT_m wird zu gegebener Zeit durch die gleichzeitige Anwesenheit von $\overline{LIT} = 0$ und $\overline{LIT}_m = 0$ gelöscht.

\overline{CI} (Clear not)

Das Clear-Signal wird im System Controller an einigen Stellen zum Löschen benutzt.

FF (Fehlfunktion)

Dieses Signal meldet dem Rechner, daß eine Störung vorliegt und deshalb die verlangte Funktion nicht ausgeführt werden konnte.

AT_i ($1 \leq i \leq 24$) (Autonome Transfer-Daten)

Auf den 24 bidirektionalen AT_i -Leitungen findet der Datentransfer im Cycle-Stealing-Verfahren direkt vom und zum Kernspeicher statt.

$ANFATE$ (Anforderung autonomer Transfer Eingabe)

Wenn der Rechner autonome Transfers zulassen will, so setzt er diese Leitung auf L. Den wirklichen Zeitpunkt des Transfers kann er nicht bestimmen, sondern er wird ihm durch $RMATE$ mitgeteilt.

RMATE (Rückmeldung autonomer Transfer Eingabe)

Dieses Signal bedeutet, daß autonome Daten, die in den Kernspeicher abgelegt werden sollen (Eingabe), zur Verfügung stehen.

ANFATA (Anforderung autonomer Transfer Ausgabe)

Das Signal entspricht in seiner Bedeutung dem ANFATE und gilt für die Ausgabe-Richtung.

RMATA (Rückmeldung autonomer Transfer Ausgabe)

Dieses Signal entspricht RMATE und steht für die erfolgte Übernahme der ausgegebenen autonomen Daten.

3. Haupt-Funktionsgruppen 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 zur CALAS-Datenendstelle. Hier werden nur die Signale auf den in Kapitel 2 aufgezählten Leitungen gepuffert und verstärkt, sonst enthält diese Funktionsgruppe keine weitere Elektronik.

3.2. Branch Driver

Die Funktionsgruppe Branch Driver enthält ausgangsseitig die Terminationen und die bereits durch die Beschreibung des Branch Highway [4] bekannten und in Tabelle 2 noch einmal zusammengestellten Leitungen. Die Leitungen des Branch Highway werden aus Gründen der Störsicherheit auf Kosten eines in Kauf genommen höheren Energieverbrauches an beiden Enden nahezu angepaßt abgeschlossen. Die Signale auf den Leitungen des Branch Highway unterscheiden sich von denen auf den in Fig. 2 (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.

3.3. Interne Logik

Die in diesem Abschnitt beschriebenen Baugruppen des System Controller haben unterschiedlich große Bedeutung. Die unter 3.3.1. bis 3.3.8. aufgeführten sind unbedingt und in jedem Falle erforderlich, die übrigen erhöhen nur den Komfort, die Komplexität und - leider - auch die Kosten.

3.3.1. Befehlsregister

Ein Pufferspeicher für Befehle ist primär dadurch nötig, daß es nur einen Satz von 24 bidirektionalen Leitungen gibt, über die

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	<u>1</u>
BX	Reserve	9 =====
		66

Befehle und Daten geschickt werden. Befehle werden zuerst geschickt und zwischengespeichert. In Fig. 3 ist die Zuordnung der einzelnen Bitgruppen des CAMAC-Befehls CRNAF zu den Speicherstufen des Registers zu erkennen. Das Befehlsregister (BR) hat eine Kapazität von 22 Bit. Der eigentliche Befehl besteht nur aus 17 Bit, nämlich

CR	3
N	5
A	4
F	<u>5</u>
	17

Die übrigen 5 Bit werden für Steuerzwecke gebraucht. So gehören z. B. die Bits 19 und 20 zur Blocktransfer-Einrichtung (siehe Abschnitt 3.3.9.). Ist "IN" gesetzt, so wird automatisch während jedes Transfers der N-Teil des gültigen Befehles um 1 erhöht. Durch diese N-Fortschaltung wird der Befehl für den nächsten Transfer im System Controller automatisch gebildet. Analoges gilt für die Subadresse A. Sind beide Bits gesetzt, also IN = L und IA = L, so ist vereinbart, daß zunächst in einer Station N alle Subadressen vom festgelegten Anfangswert bis zum Endwert bedient werden. Dann wird die Stationsnummer N um 1 erhöht und gleichzeitig A wieder auf den Anfangswert zurückgesetzt.

Jede Stufe des Befehlsregisters kann aus verschiedenen Quellen gesetzt werden, eine davon sind die Rechner-Leitungen. Die anderen Zugänge stammen aus internen Registern, über die noch gesprochen wird.

3.3.2. 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 Bit bcr1, bcr2 und bcr4 des Befehlsregisters lassen sich 8 Adressen unterbringen, 7 für die CAMAC-Crates und cr0 für den System Controller (vgl. Fig. 3). 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. 3 ist zu sehen, daß sowohl bei rechnerkontrollierten Operationen als auch bei Alarm-Verarbeitung (DO-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 "Crate not on-line" (CRnOL) erzeugt und ins Statusregister des System Controller eingeschrieben. Dies erleichtert dem Rechner die Fehlerdiagnose.

3.3.3. Daten-Leitungen

Im einfachsten Falle können die Informations-Leitungen vom Rechner zum Transport von Daten auf die Treiber der BRW-Leitungen gegeben werden. Letztere können nach Fig. 4a auch auf die ATi-Leitungen geschaltet und von ihnen angesteuert werden. Wird der System Controller so weit ausgebaut, daß er Modul-zu-Modul-Transfer bewältigen kann, dann wird nach Fig. 4b 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.

3.3.4. Timing-Generator

In dieser Baugruppe werden die für den ordnungsgemäßen Betrieb des System Controller benötigten Zeitsignale erzeugt (s. Fig. 5). 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", "Einzeloperation" 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 "Einzeloperation" und "Step" möglich. Bei "Einzeloperation" 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 eine zweite Taste 5mal gedrückt werden. Bei beiden Betriebsarten bleibt der Rechner mit dem System verbunden, der zeitliche Ablauf jedoch dem Operator überlassen, der dadurch im 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. 6 dargestellt.

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	Rückflanke	Erzeugung des Timing-Signals BTA
SS2		Adreßfortschaltung A, N;
	Rückflanke	Rücknahme des Timing-Signals BTA;
	Rückflanke	Erzeugung der Rückmeldungen
SS3		Rücknahme der Befehle (nach Rücknahme von TB 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. 15).

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)

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

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.

3.3.5. Hauptzustands-Generator

Der wichtigste Teil der Steuerlogik des System Controller ist der sogenannte Hauptzustands-Generator (siehe Fig. 7), der die Zyklen des System Controller nach einem fest eingebauten Priori-

tätssystem verteilt. Der Generator kann sich in Ruhe befinden oder in einem (und nur in einem) der fünf 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 Befehls-Operation (Command Operation) (CO)
- 4 Alarm-Operation (Demand Operation) (DO)
- 5 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 µs. 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 fünf Hauptzuständen erzeugen, so wird daraus ein Signal "Unerlaubte Betriebsart" (UB) abgeleitet und in das Statusregister eingeschrieben. Gleichzeitig ist

dafür gesorgt, daß über einen Time-out-Generator ein Programm-Unterbrechungs-Signal IT8 gesetzt 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 beim Ende wieder gelöscht, es sei denn, es vergeht eine längere als die für jeden Zyklus festgelegte Zeit. (Dann erscheint ein Signal TOn - Time-out-Signal, Stufe n -, das die Interrupt-Anforderung IT8 setzt.)

3.3.6. Statusregister

Das Statusregister hat eine Kapazität von 14 Bit und ist vom Rechner lesbar. Es soll dem Rechner die wichtigsten Statusmeldungen zur Erleichterung einer Fehlerdiagnose geben. Die Belegung der einzelnen Bits des Registers und die Zuordnung zu den Informations-Leitungen ist aus Fig. 8 ersichtlich. Das Statusregister hat wie üblich die Subadresse A(15).

3.3.7. Alarmregister

Das Alarmregister hat eine Kapazität von 24 Bit und wird bei Alarm-Operationen, also bei Anwesenheit des Signalpaares BD/BG, über die BRW-Leitungen gesetzt (siehe Fig. 9). Die ersten 8 Bit steuern 8 Interrupt-Leitungen des CALAS-Systems, also letzten Endes des Rechners an, die restlichen 16 Bit werden als Anforderungen für autonome Eingaben in den Kernspeicher des Rechners interpretiert. Diese Bits (a9 ... a24) werden nach Abwicklung der nötigen Operationen automatisch zurückgesetzt, die Interrupt-Speicher müssen vom Rechner gezielt gelöscht werden (durch $\overline{LIT} = 0$ und $\overline{LITm} = 0$). Der Ablauf der einzelnen Operationen wird in Kapitel 4 genauer besprochen.

3.3.8. Anzeige- und Bedienungsfeld

Die Anzeige- und Bedienungselemente an der Frontplatte des System Controller zeigt Fig. 10.

Der Betriebsarten-Wahlschalter hat die Stellungen "Löschen", "Step", "Einzeloperation" und "Run". Nur in der Stellung "Löschen" läßt sich das Netz einschalten. Das hat den Vorteil, daß beim Einschalten des Gerätes zwangsweise ein definierter Anfangszustand hergestellt wird. In der Betriebsart "Step" kann ein Zyklus des System Controller für Testzwecke mit Hilfe der Taste 1 in seine fünf einzelnen Phasen zerlegt werden. In der Betriebsart "Einzeloperation" läuft beim Drücken der Taste 2 ein Zyklus ab. In der Betriebsart "Run" übernimmt der Rechner die volle Kontrolle.

Das Signal BZ läßt sich außer durch einen Spezialbefehl und das Clear-Signal vom Rechner noch durch eine Taste Z am System Controller von Hand erzeugen.

Mit Hilfe eines Wahlschalters mit den Stellungen DRB (Display Read Buffer), DCR (Display Command Register) und DDR (Display Demand Register) kann der Inhalt des Lesebuffer-, Befehls- oder Alarmregisters an einem 24-Bit-Display dargestellt werden.

3.3.9. Blocktransfer-Einrichtung

Zur Blocktransfer-Einrichtung gehören nach Fig. 11 folgende Bestandteile:

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 Startspeicher.

In Fig. 11 sind auch die zu den einzelnen Hardware-Komponenten gehörenden Subadressen zu erkennen.

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(11) 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(9).

Damit ist der System Controller für einen Blocktransfer vorbereitet. Erscheint der Spezialbefehl "Blockstart" (BS) $CR(0) \cdot N(0) \cdot A(11) \cdot F(26)$, 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 Stationsadresse N oder Subadresse A bilden (Bits IN und IA im Befehlsregister, vgl. Abschnitt 3.3.1.). Beim letzten Transfer wird das Signal "Blockende" (BE) gebildet und an den Rechner gegeben. Es wird aus den Overflows (OF) der N- bzw. A-Zähler gebildet nach der in Tabelle 4 gegebenen Übersicht.

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

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

Da die Werte in A(11) und A(9) zerstörungsfrei gelesen werden, kann der Block immer wieder neu gestartet werden.

3.3.10. Modul-zu-Modul-Transfer

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

- 1 Die gesamte in Abschnitt 3.3.9. genannte,
- 2 das Lesebuffer-Register (vgl. Abschnitt 3.3.3.),
- 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 $CR(O) \cdot N(O) \cdot A(8) \cdot F(26) = M$

Fig. 12 zeigt ein Schema der kompletten Hardware. Zu den in Abschnitt 3.3.9. genannten Einstell-Operationen gehört nun ergänzend noch das Laden des "Ziel-CRNAF" in das Register mit der Subadresse A(10) sowie die Übersendung von "M".

Mit dem Befehl "Blockstart" an den System Controller wird der Block aktiviert und zuerst der in Subadresse A(11) 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(10) befindlichen und ins Befehlsregister transportierten Befehls "Ziel-CRNAF" zum Ziel transportiert. Zu Beginn des zweiten Zyklus war auf Ausgabe umgeschaltet worden. Für jede Ausgabe ist das Ziel das gleiche.

Das Signal "Blockende" wird bei der letzten Ausgabe gebildet. Wird eine Wiederholung gewünscht, dann müssen die Befehle "M"

und "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 der Blocktransfer durch den Befehl "Blockstart", der Modul-zu-Modul-Transfer durch die Befehle M und Blockstart. Es geht dabei keine Information verloren.

3.3.11. 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 ist die beteiligte Hardware zu sehen. Bis zu 16 im ganzen System verstreute Einheiten können an diese Einrichtung angeschlossen werden. 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. Nun etabliert sich der Hauptzustand ATE (Autonomer Transfer Eingabe) und bleibt so lange erhalten, bis auch die letzte Eingabe erfolgt ist. 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 Lesebefehl aufgerufen und ins Befehlsregister geladen. Die folgende normale Lese-Operation liefert ein 24-Bit-Wort über die BRW-Leitungen und das Lesebuffer-Register auf die 24 AT-Leitungen. Die Anwesenheit dieser Daten wird dem Rechner durch das Signal RMATE angezeigt. Der Rechner übernimmt die Daten.

Diese Lesedaten enthalten 8 Bit Quelladresse (Bits 24 bis 17) und 16 Bit eigentliche Lese-Information. Der Rechner muß seine Bereitschaft zum Empfang solcher Daten durch das Signal ANFATE 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 CALAS-Elektronik festgelegt werden.

3.3.12. Autonomer Transfer in Ausgabe-Richtung

Beim autonomen Transfer in Ausgabe-Richtung ist der elektronische Aufwand viel kleiner. Benutzt werden wieder die bidirektionalen ATi-Leitungen, diesmal aber mit einem anderen Paar von Timing-Signalen (ANFATA/RMATA), die BRW-Leitungen und das Befehlsregister. Der im Befehlsregister stehende Befehl nimmt die auf den ATi-Leitungen anstehenden Signale in Empfang und transportiert sie an die richtige Adresse.

Damit sind alle wesentlichen Grund- und Zusatzfunktionen des System Controller beschrieben. Im nächsten Abschnitt werden an Hand von Zeitdiagrammen und Signalflußbildern einige wichtige Abläufe im Detail geschildert, wobei wieder die Einteilung nach lebensnotwendigen Grundfunktionen und zeitsparenden Zusätzen beibehalten wird.

4. Funktionsabläufe

Fig. 14 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 "Sequentielles Lesen/Schreiben" (SQ) sowie "Blocktransfer" wie Einadreß-Operationen abgewickelt. Der Blocktransfer kann auch zyklisch stattfinden (ZYB). Der Modul-zu-Modul-Transfer sowie seine zyklische Wiederholung (ZYM) 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 nicht von der zufälligen Anwesenheit des CALAS-Systems ab und auch nur insofern vom Rechner, als ein Direktzugriffs-Kanal und wenigstens eine Interrupt-Ebene gebraucht wird. 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.

4.1.1. Dialog Rechner - System Controller ohne Operand

In dem Timing-Diagramm von Fig. 15 ist als Beispiel für diese Operationen die Übernahme des Befehls "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 IO.i-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 ANF aktiviert. Dabei wird vorausgesetzt, daß kein Hauptzustand höherer Ordnung herrscht. Das Signal ANF 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 RM über den Vollzug des Befehls. Rund 100 ns nach SS3 ist der System Controller wieder frei. Neben dem Takt-diagramm kann das Signalflußschema Fig. 16 nützlich sein, den Ablauf der Operation bei Beachtung aller Nebenbedingungen 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

Tabelle 5: Kennung der vom Rechner gelieferten Information

Trans- fertyp	Si- gnal	BF	OP	mOP (Bit 21)	E (Bit 22)	crO	Bedeutung
1		L	0	0	L	L	Befehl ohne Operand an System Controller
2		L	0	L	0	L	Befehlsteil eines Befeh- les mit Ope- rand für System Controller
2a		0	L	-	-	-	Daten an System Controller
3		L	0	L	L	0	Befehlsteil eines Lesebe- fehles für System
3a		0	L	-	-	-	Abholung der Lesedaten
4		L	0	L	0	0	Befehlsteil eines Schreib- befehles für System
4a		0	L	-	-	-	Anlieferung der Schreib- daten
5		L	0	0	-	0	Befehl ohne Operand an System

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.

4.1.3. Einadreß-Lesebefehle

Zur Beschreibung des Ablaufes gehören die Figuren 17 und 18 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 die Lesedaten an der Quelle abgeholt und dem Rechner zusammen mit der Rückmeldung RM übergeben. Beim zweiten Zyklus ist eine Laufzeit der Signale auf dem Branch Highway von 100 ns angenommen. Unter diesen Umständen ergibt sich eine Gesamtdauer für den zweiten Zyklus von 2,4 μ s, wogegen die Absetzung des Befehles in 1,55 μ s geschehen kann. In Fig. 17 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. 200 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 (RM) an den Rechner und die Rücknahme von BTA abgeleitet;

- 4 die Erfüllung der Gleichung (2) erzeugt den Impuls SS3, Etwa 100 ns nach seiner Rückflanke ist der System Controller wieder frei.

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 gilt Zeile 5 in Tabelle 5 und das Diagramm von Fig. 19. Der Befehl wird unmittelbar nach dem Empfang auf den Branch Highway getatet 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 rechnergeführte Blocktransfer behandelt.

4.1.6. Blocktransfer

Um einen Blocktransfer in Gang zu bringen, müssen verschiedene, aus Fig. 20 erkennbare Einstellungen des System Controller vorgenommen werden. Der letzte Befehl der Einstellphase heißt "Blockstart". Beim Blocktransfer gibt es drei Möglichkeiten, die sich durch die Kombinationen der Bits IN und IA ausdrücken lassen, wie in der folgenden Tabelle 6 gezeigt.

Tabelle 6: Mögliche Typen des Blocktransfers

	Bit 19 (IN)	Bit 20 (IA)
1	O	L
2	L	O
3	L	L

Betrachtet wird Fall 3, in dem die beiden anderen enthalten sind. Der Block startet beim ersten Datentransfer (BF = 0, OP = L) 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 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 RM 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.

4.2. 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 Befehlsoperationen. In Fig. 21 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" und "M" wiederhergestellt werden. Dann geht der Modul-zu-Modul-Transfer an der Stelle weiter, wo er unterbrochen wurde.

4.3. Alarm-Verarbeitung

Über das Karlsruher Gesamtkonzept der Alarm-Verarbeitung wird in einem eigenen Beitrag berichtet [5]. 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. 22 ist der zeitliche Ablauf einer solchen Operation aufgezeichnet, in Fig. 23 das zugehörige Signalfluß-Schema. 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.

4.3.1. Interrupt-Behandlung

Die Abarbeitung von Interrupt-Requests geht in der von der Rechnertechnik 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 durch die gleichzeitige Anwesenheit der Signale $\overline{LIT} = 0$ und $\overline{LITm} = 0$ wieder 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 24 und 25. Der autonome Transfer in Eingabe-Richtung ist bei diesem System Controller der Hauptzustand höchster Priorität und damit nicht unterbrechbar. In Fig. 24 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 \overline{ATR} (Autonomer Transfer not Ready). Die Anwesenheit dieses Signals verhindert die anderweitige Benutzung des System Controller. Es etabliert sich der Hauptzustand ATE. Unter seiner Kontrolle startet ein Suchgenerator (Priority Sorter), der die einzelnen Bits des Alarmregisters abfragt, beginnend mit Nr. 24. 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, d. h. für das Bit Nr. 24). Von jedem ACK-Signal wird dasselbe Start-Signal SLC (Start Lese-CRNAF) für den Timing-Generator (Eingang Selbststart) abgeleitet. Zur Taktzeit SSO wird das von AC24 aufgerufene Befehlsword aus dem Vorratsspeicher in das Befehlsregister umgeladen. Dabei werden am Befehlsregister die Eingänge LLB (Lade Lesebefehl) benutzt. Es läuft also 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 ATi-Leitungen geschaltet. Dem Rechner wird durch das Signal RMATE bekanntgegeben, daß Information bereitgestellt ist. Die Signale AC24 und SLC verschwinden mit der Vorderflanke von SS3, nicht aber \overline{ATR} , denn vereinbarungsgemäß sollte ja ein zweiter Request (a22) gleichzeitig erschienen sein. Durch $\overline{ATR} = L$ wird unmittelbar nach dem Verschwinden von SCB, also sofort wenn der System Controller wieder frei ist, erneut ATE gebracht und der Priority Sorter wieder gestartet. Er sucht so lange, bis er wieder einen Request findet. Es folgen so lange autonome Lese-Operationen, bis $\overline{ATR} = 0$ geworden ist.

Es soll noch eine Abschätzung des maximalen Zeitbedarfes 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	2,2 μ s
16 x 100 ns für Scanner	1,6 μ s
16 x 2,4 μ s für Lese-CRNAF's	38,4 μ s
		<hr/>
	Summe	42,2 μ s

4.4. Autonome Transfers (Ausgabe)

Autonome Datenausgaben werden vom Rechner unter Programm-Kontrolle vorbereitet. Dazu wird das Timing-Signal ANFATA benutzt, welches im Hauptzustands-Generator das Signal ATA erzeugt. Die genaue Beschreibung des Ablaufes kann dem Schema von Fig. 26 entnommen werden. Bei dieser Art der Datenausgabe bedeutet das Anforderungssignal, daß Daten vorhanden sind. Der Weitertransport der Daten erfolgt unter der Leitung des im Befehlsregister stehenden Befehlswordes, welches in der Vorbereitungsphase vom Rechner dort hineingeschrieben wurde. Es ist ein Schreib-CRNAF. Alle Transfers dieser Art gehen an die durch CR, N und A bestimmte Stelle. Für den Branch Highway ist ein solcher Transfer eine Einadreß-Schreib-Operation.

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 [1] 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 (als FF-Signal).

5.2. Prüfen des Befehlsregisters

Das Befehlsregister BR hat ein auslesbares Pufferregister BR2 nachgeschaltet, in welches normalerweise alle Befehle eingegeben werden. BR übernimmt Befehle mit dem Strobe SS0, BR2 übernimmt sie mit SS1, wenn es sich nicht um den Lesebefehl $CR(0) \cdot N(0) \cdot A(2) \cdot F(0)$ handelt. Auf diese Weise kann posthum geprüft werden, ob ein Befehl richtig abgesetzt wurde.

5.3. Lesen des Statusregisters

Die Analyse der Information 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.
- 3 Der System Controller befindet sich im Zustand der unerlaubten Betriebsart (UB), weil zwei oder mehr Hauptzustände gleichzeitig herrschen.
- 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".

5.4. Tests über die Frontplatte

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 Frontplatte 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

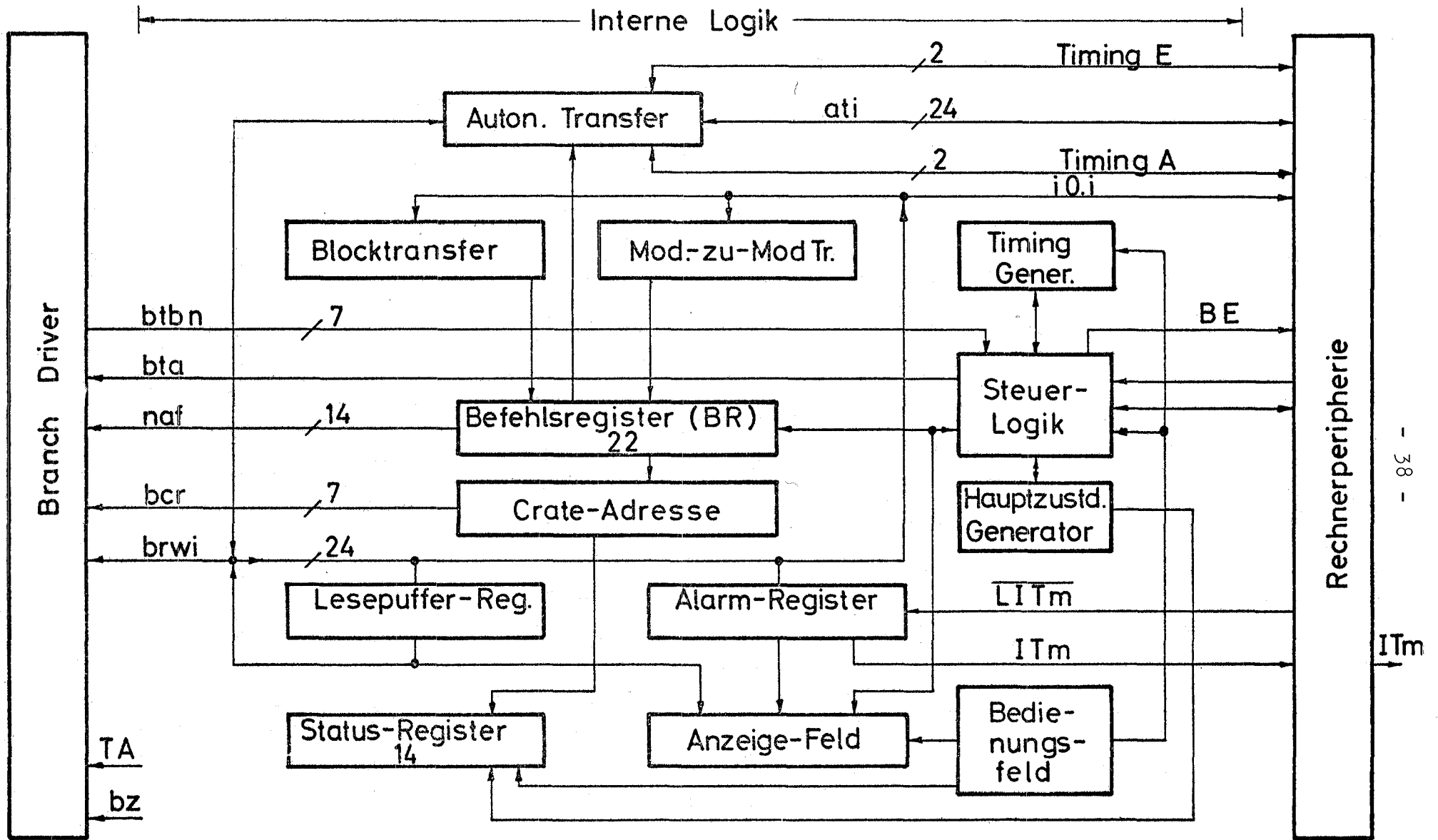
- [1] CAMAC - A Modular Instrumentation System for Data Handling -
Description and Specification
Euratombericht EUR 4100 e, Luxembourg 1969

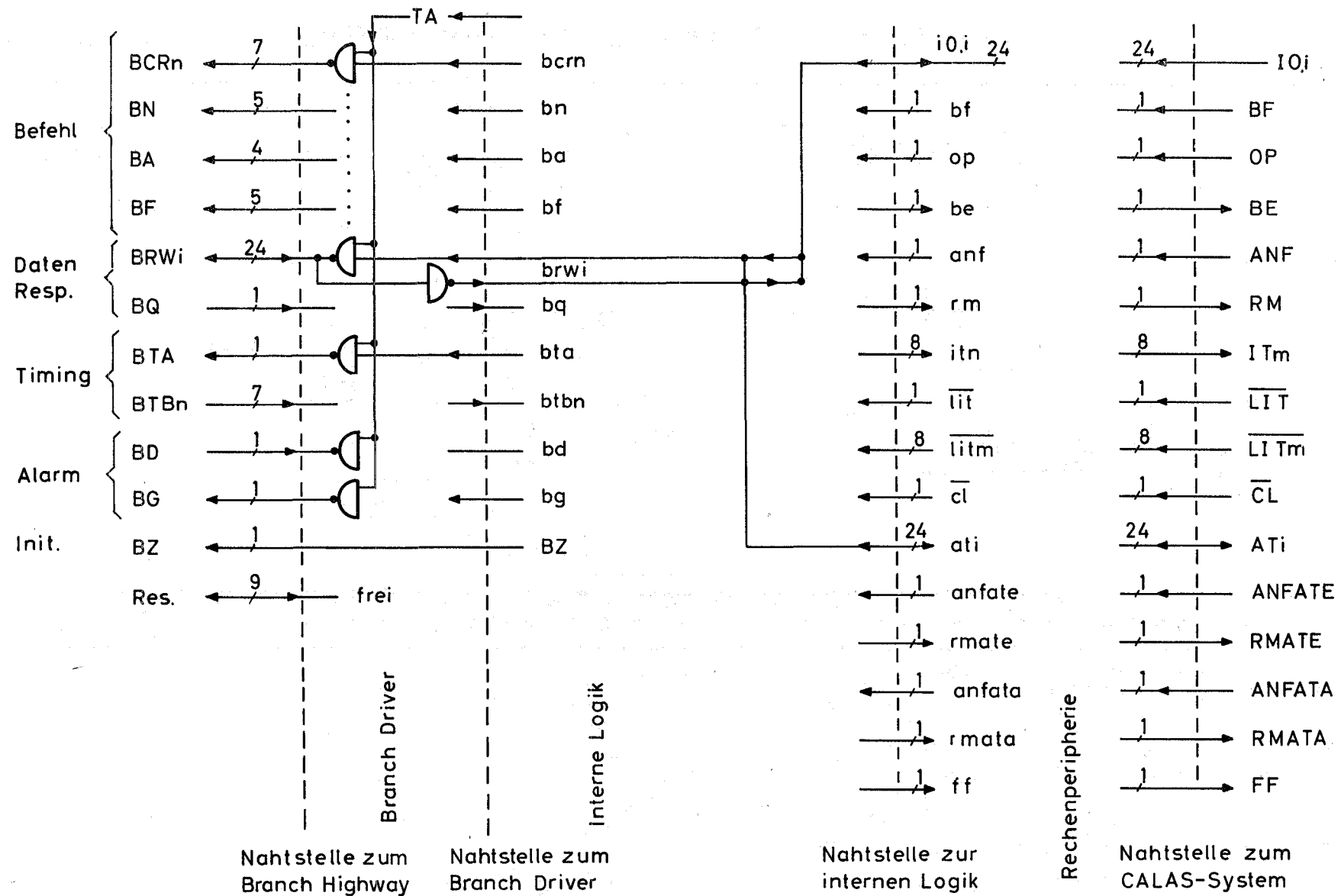
- [2] Gagel, G.; Hepke, G.; Herbstreith, H.; Nehmer, J.
CALAS 68 - ein computergestütztes Vielfachzugriffssystem
zur Laborautomatisierung
Kernforschungszentrum Karlsruhe, Externer Bericht 19/69-1,
1970

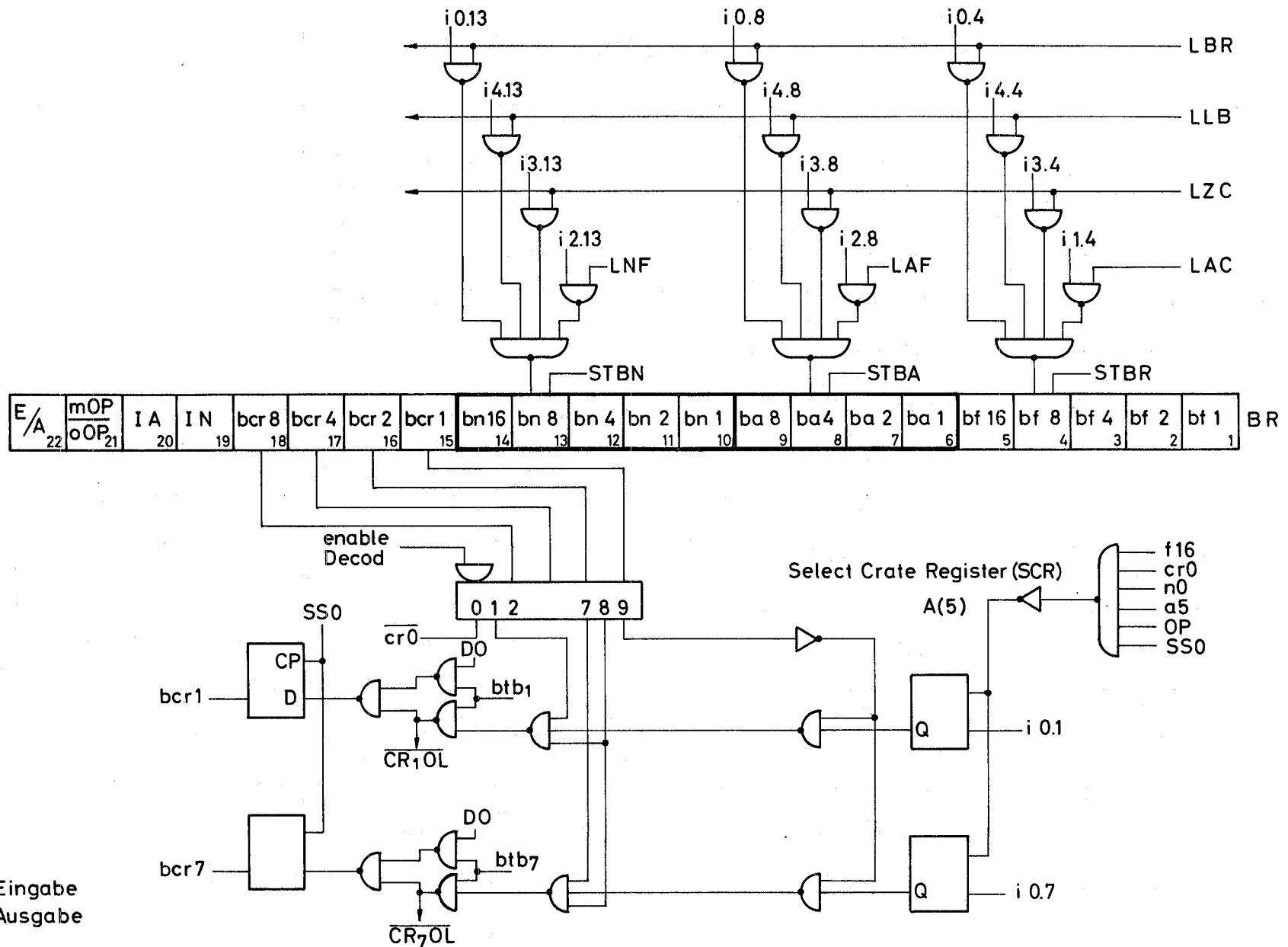
- [3] CAMAC - Organisation of Multi-Crate Systems - Specification
of the Branch Highway and CAMAC Crate Controller Type A
Euratombericht EUR 4600 e (im Druck)

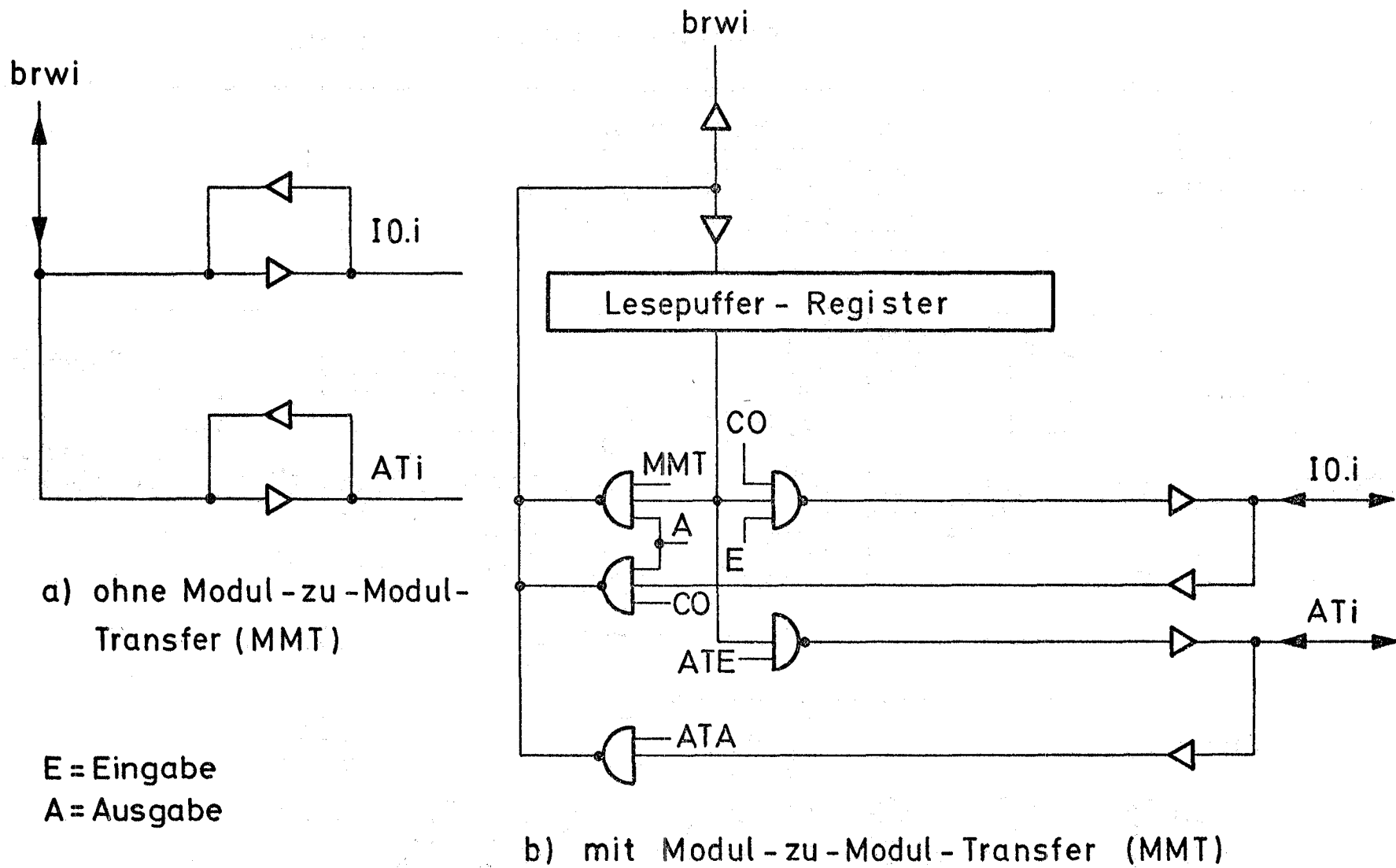
- [4] Ottes, J.
CAMAC - Ein System rechnergeführter Elektronik. Beschreibung
der gleichbleibenden Systemteile.
Kernforschungszentrum Karlsruhe, Bericht KFK 1402, 1971

- [5] Heep, W.; Ottes, J.; Tradowsky, K.
Alarm-Verarbeitung und autonomer Datentransfer im CAMAC-
System
Kernforschungszentrum Karlsruhe, Externer Bericht 22/71-7
(im Druck)









- 41 -

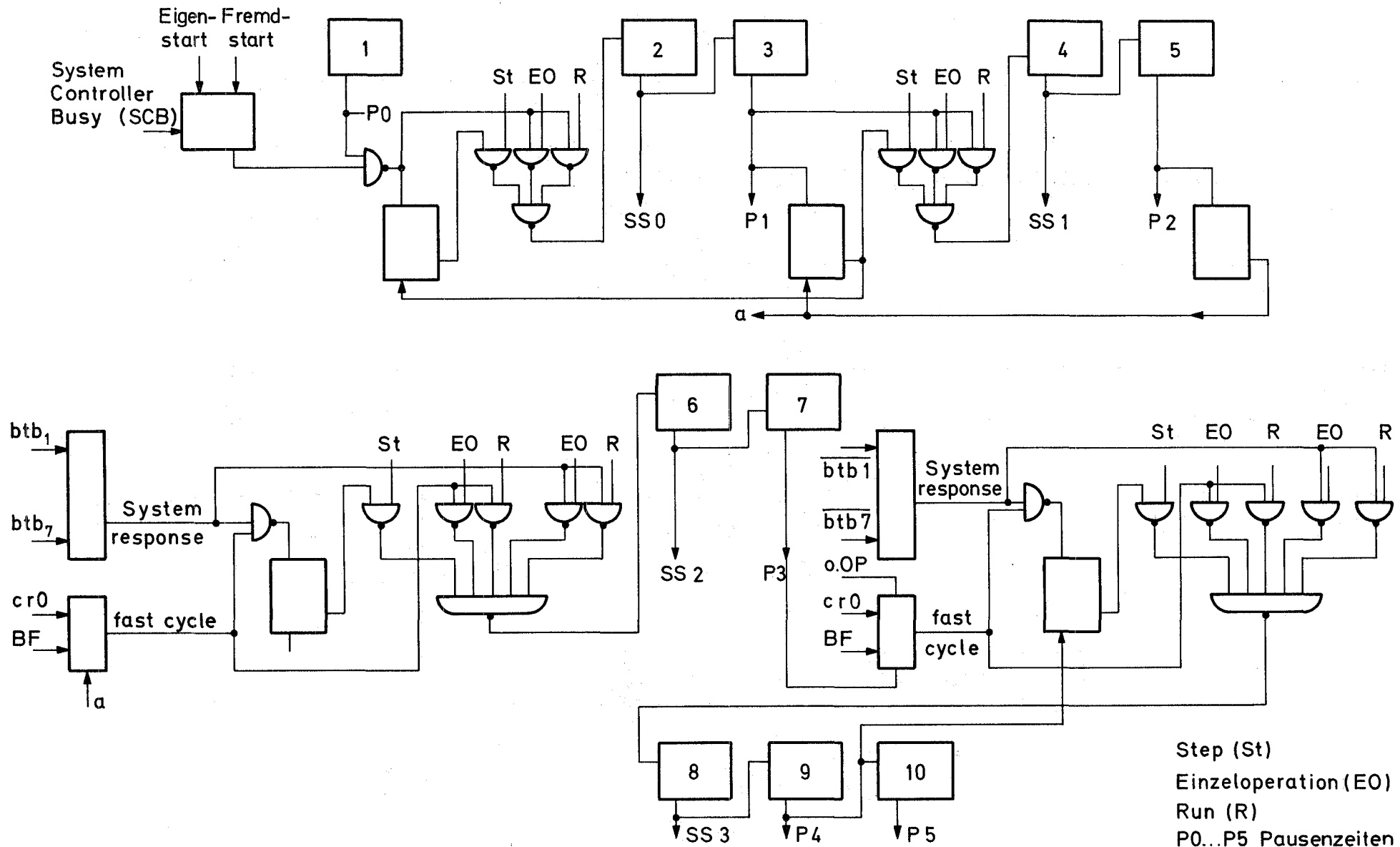
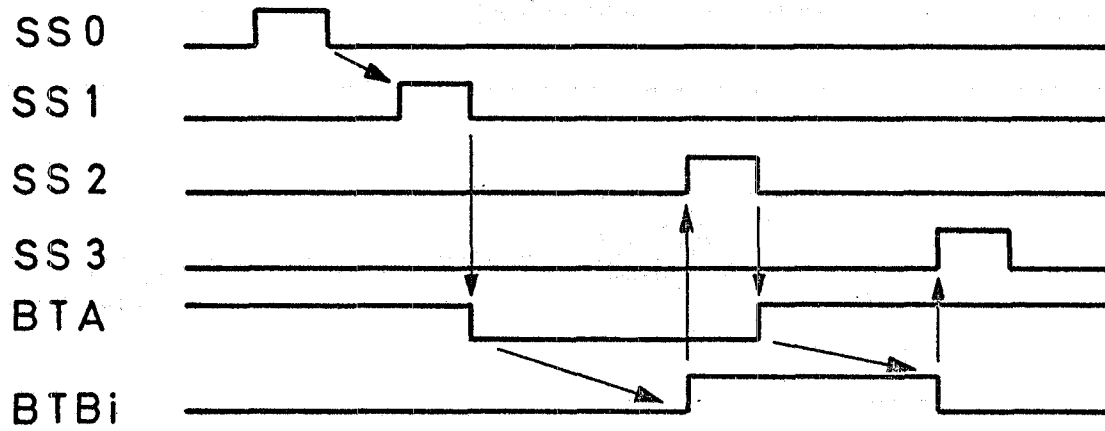
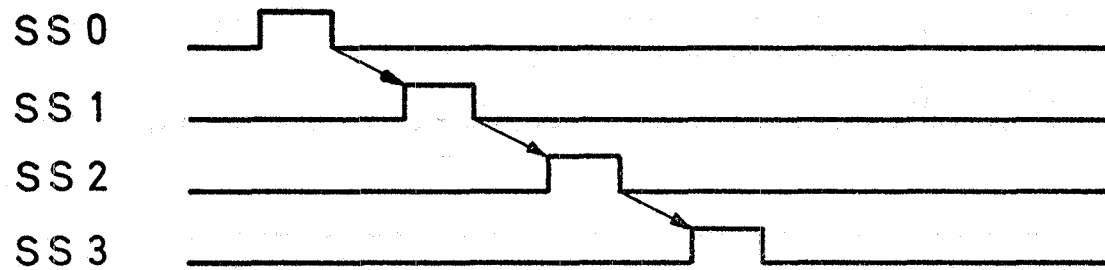


Fig. 5 Timing-Generator

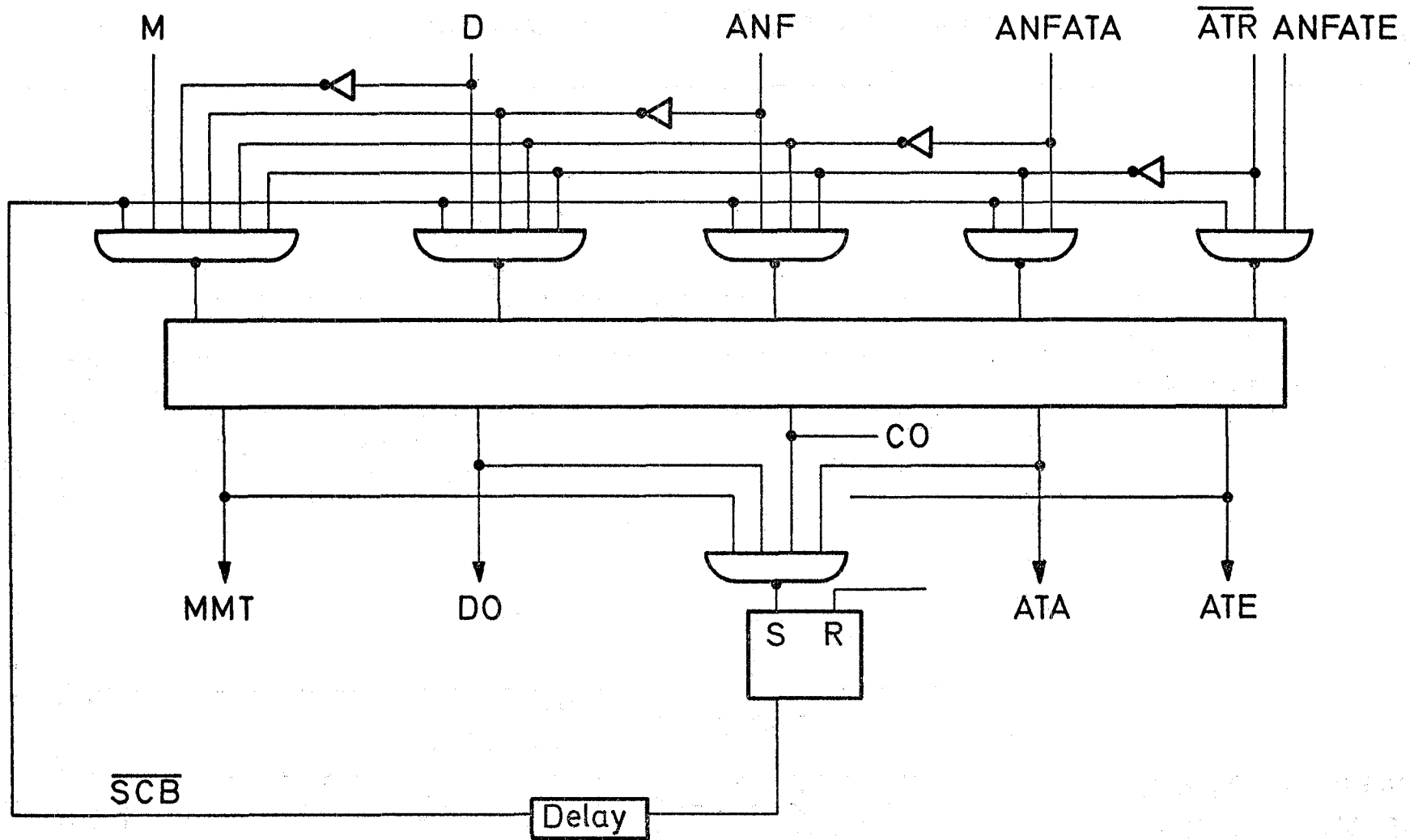


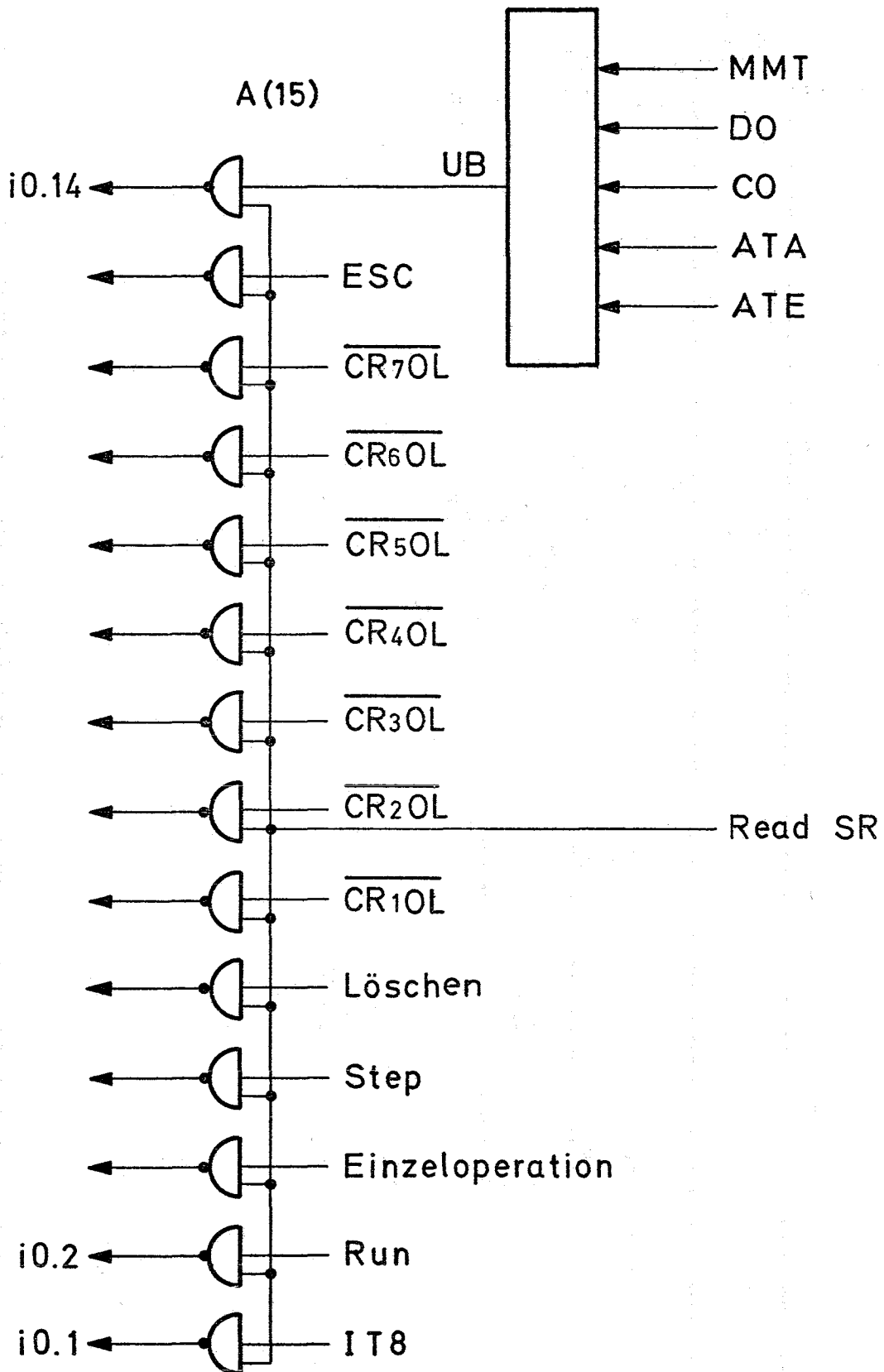
a) Verkehr des System Controller mit dem CAMAC - System

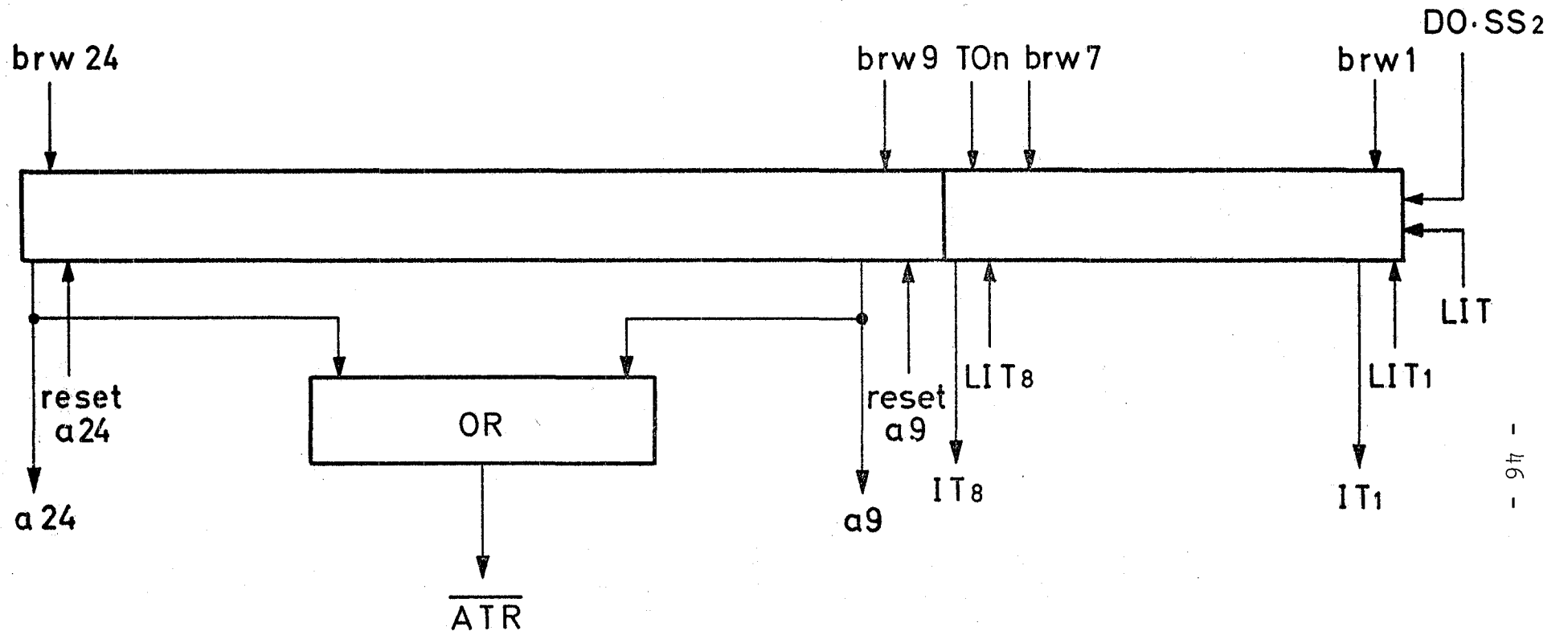


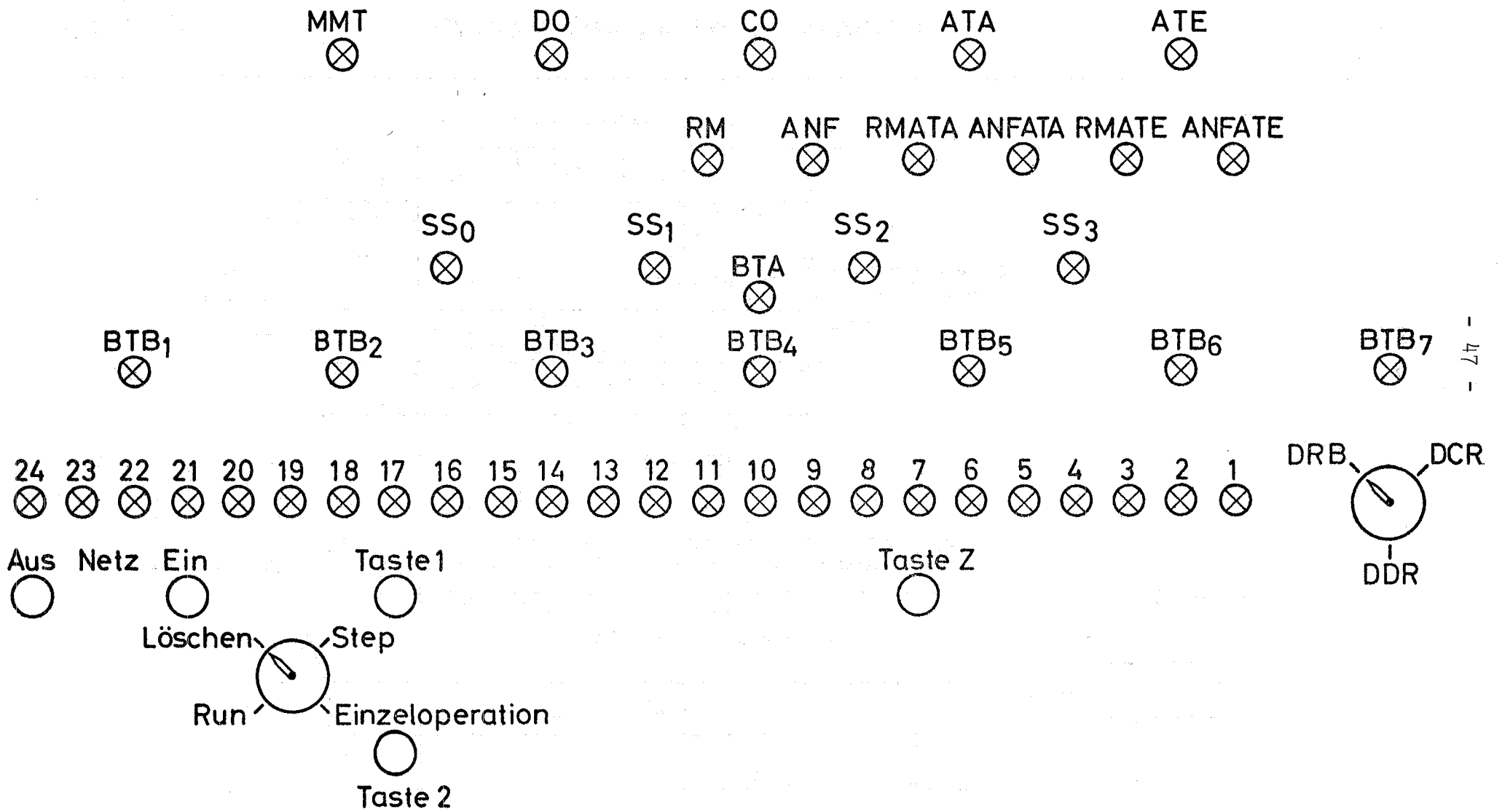
b) Schneller Dialogverkehr des Rechners mit dem System Controller

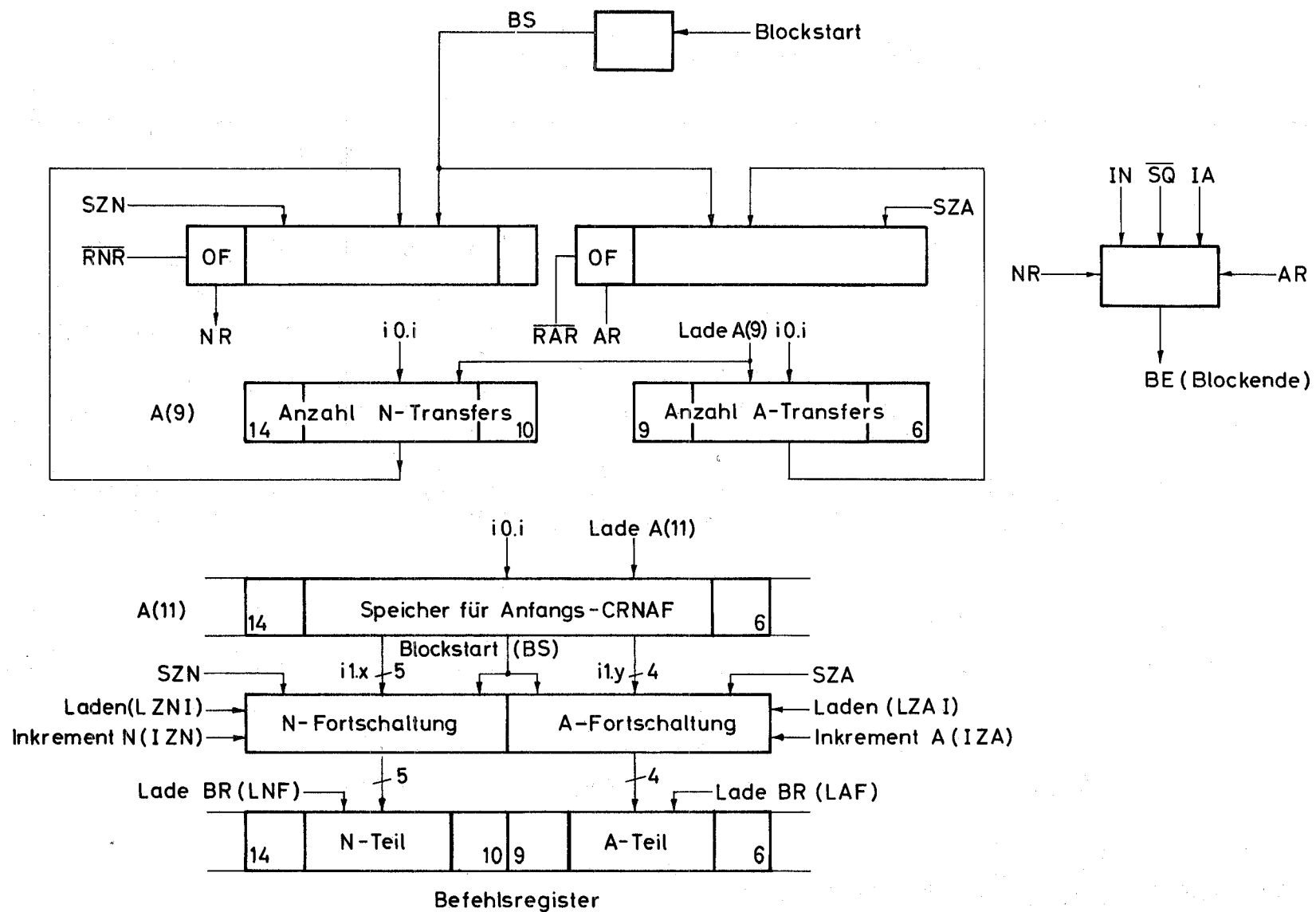
- 43 -

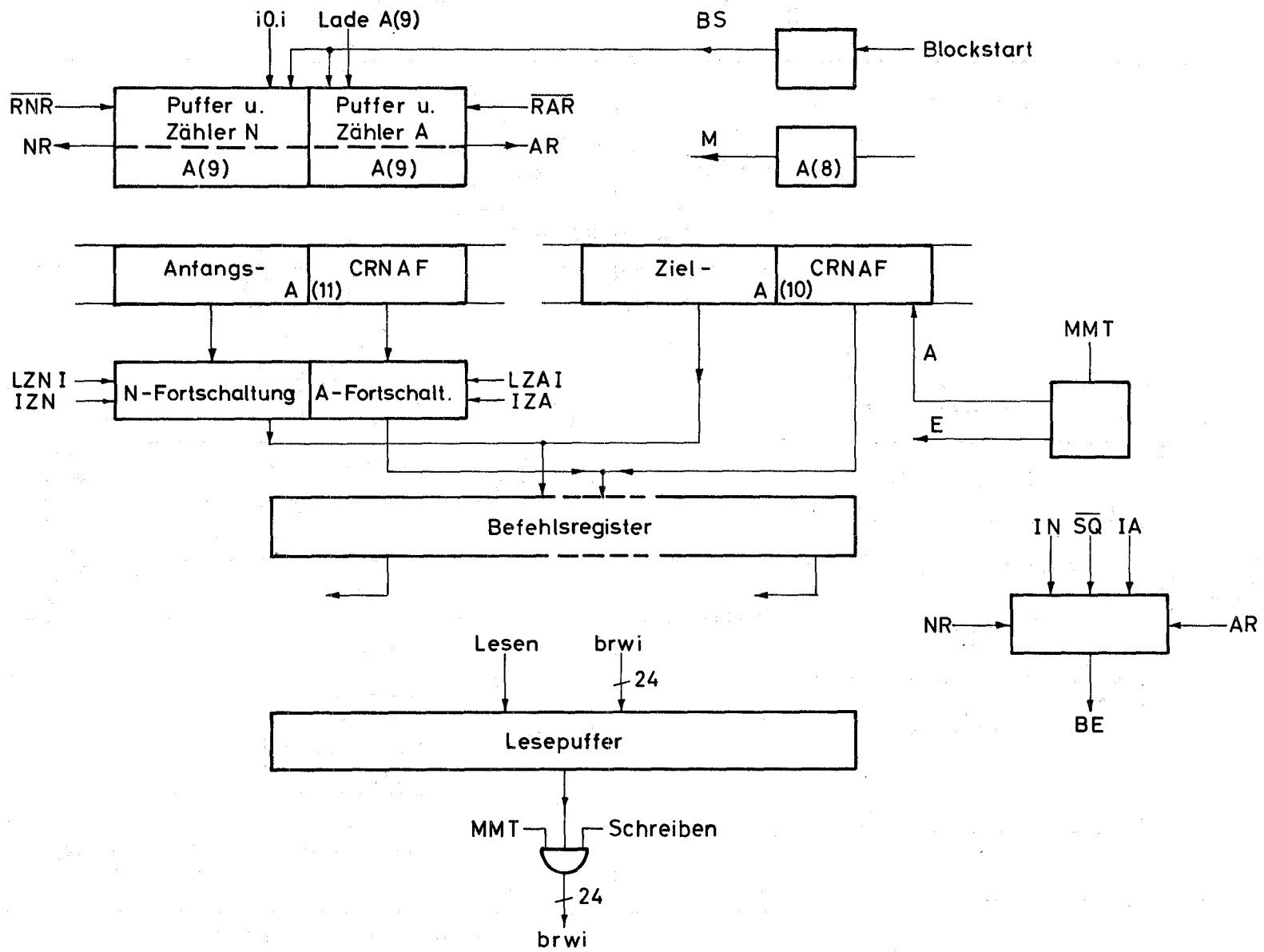




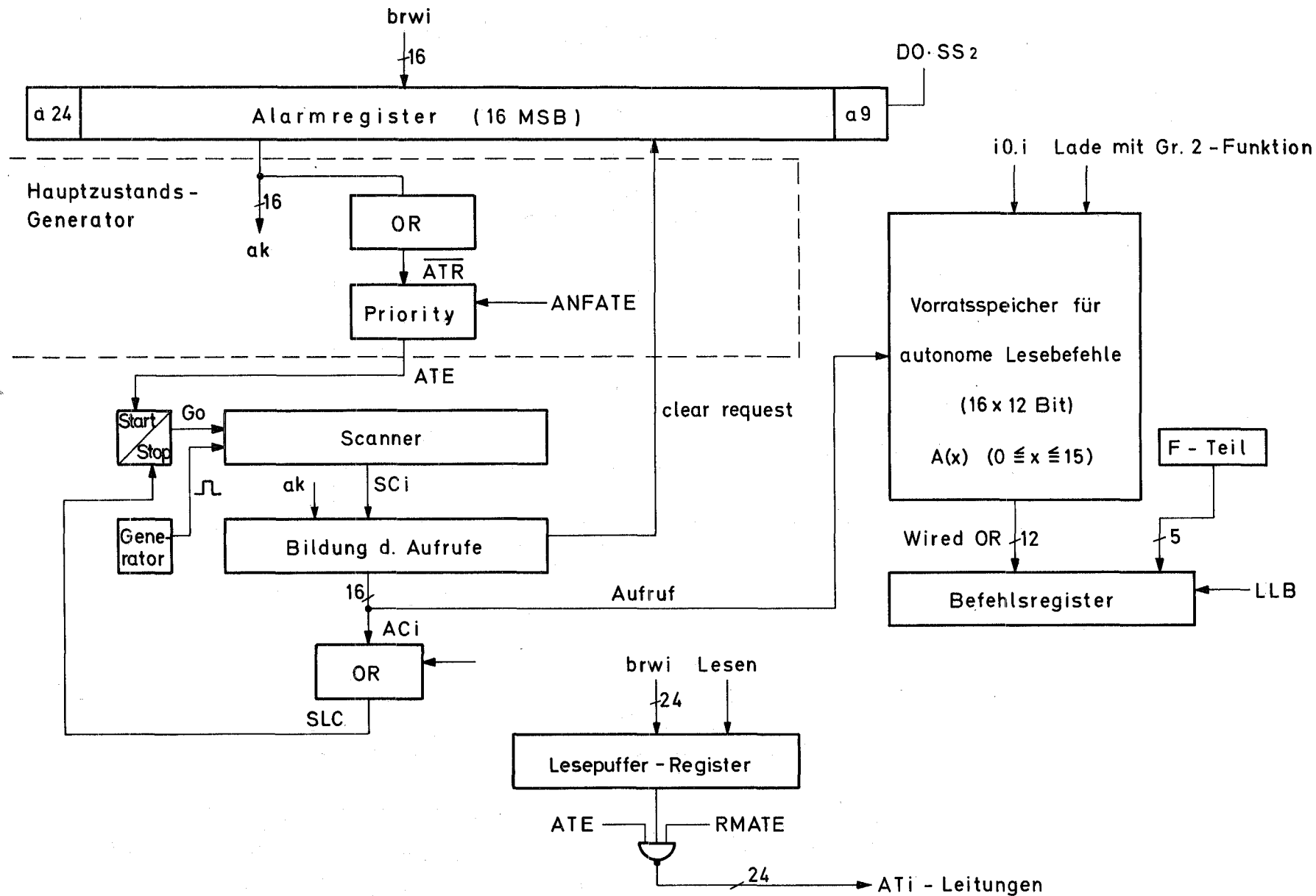




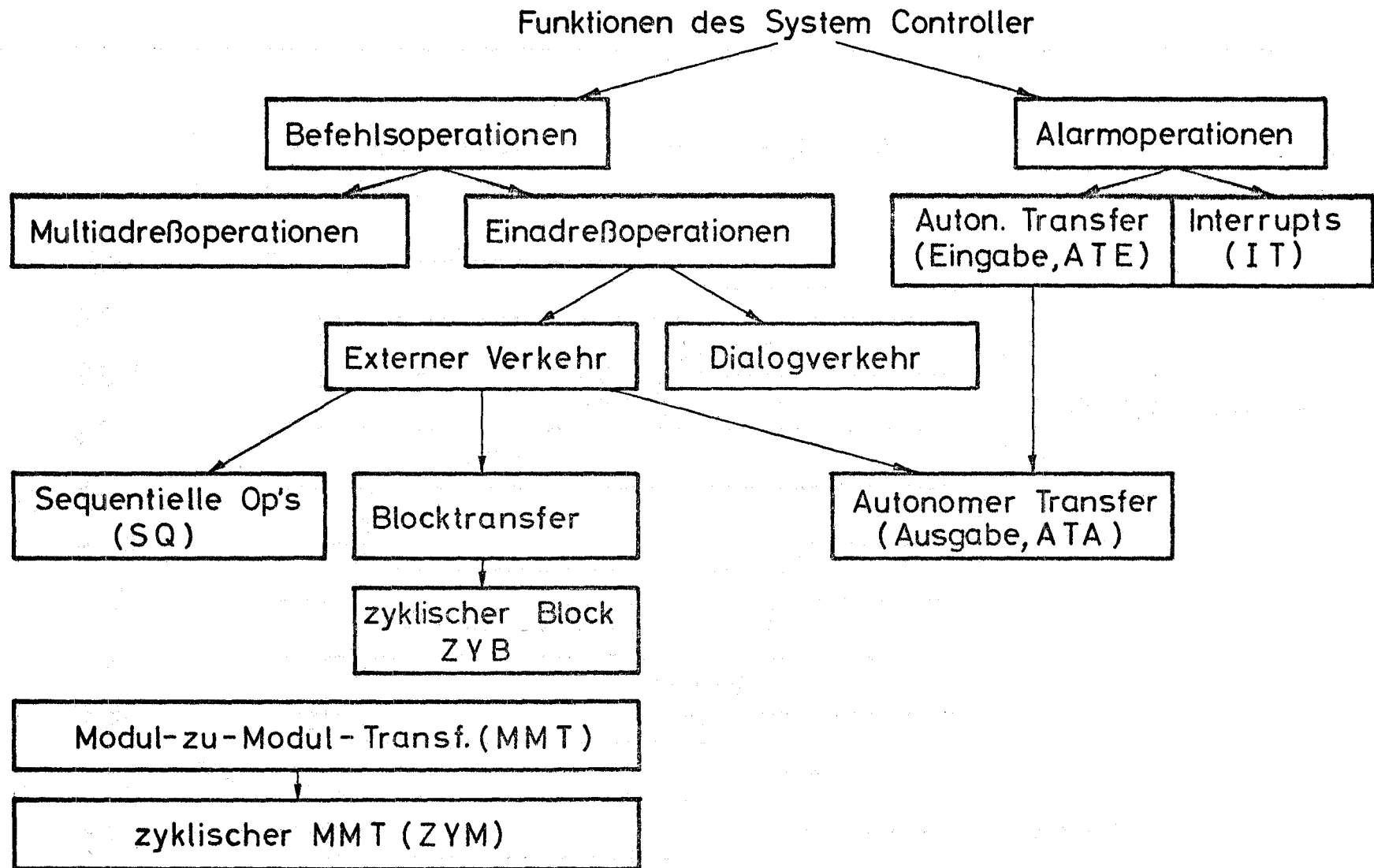




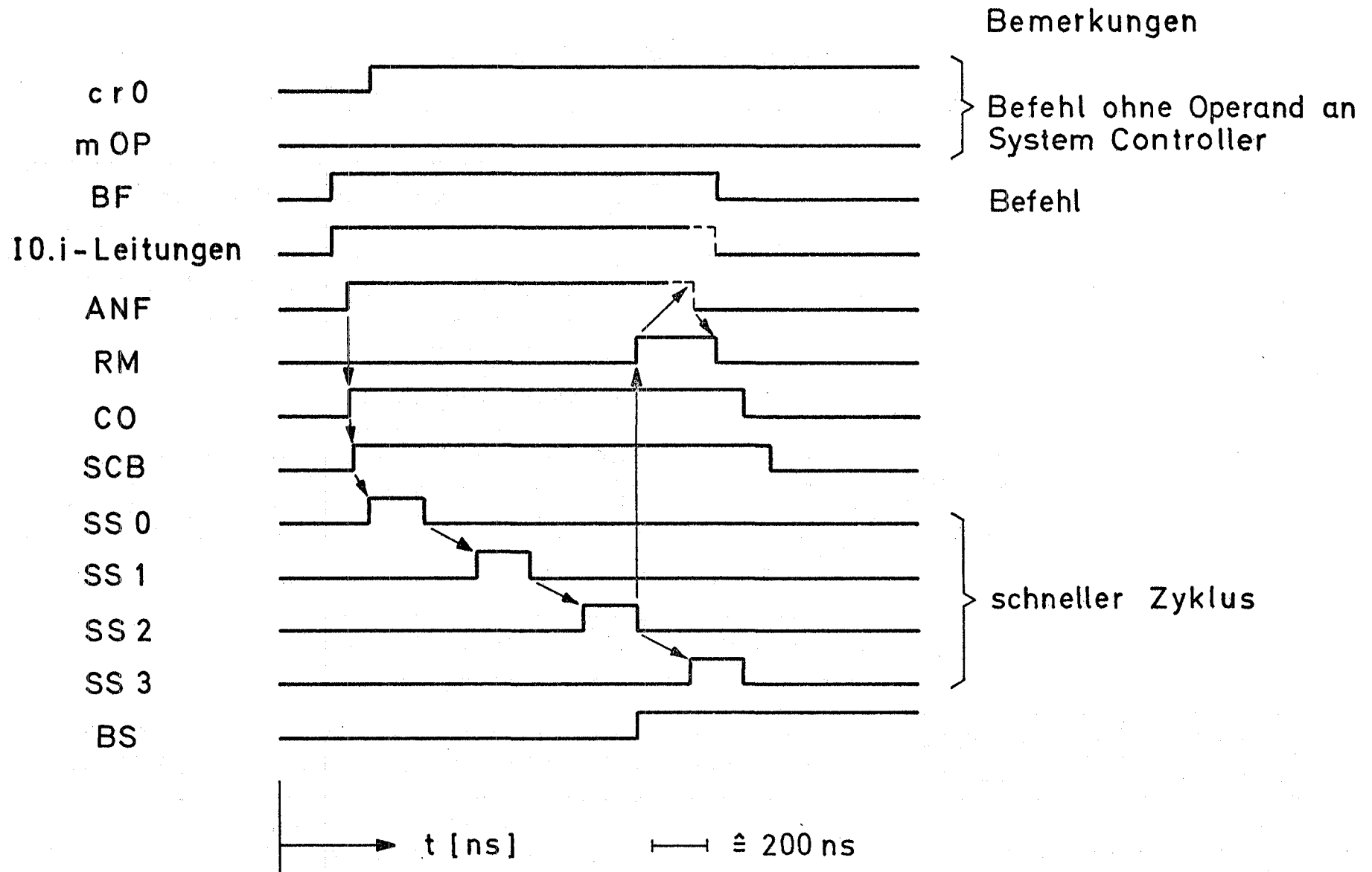
- 69 -

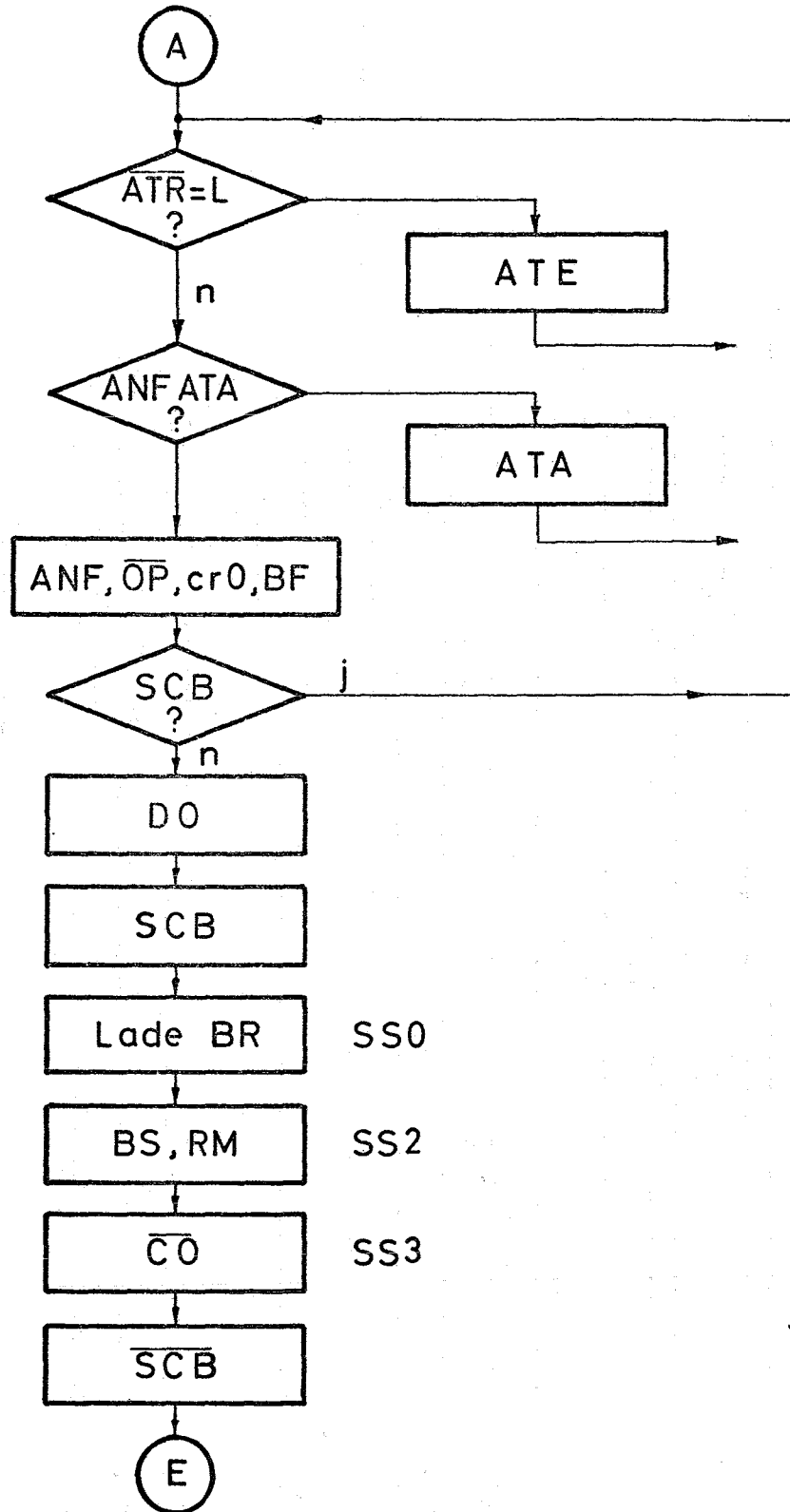


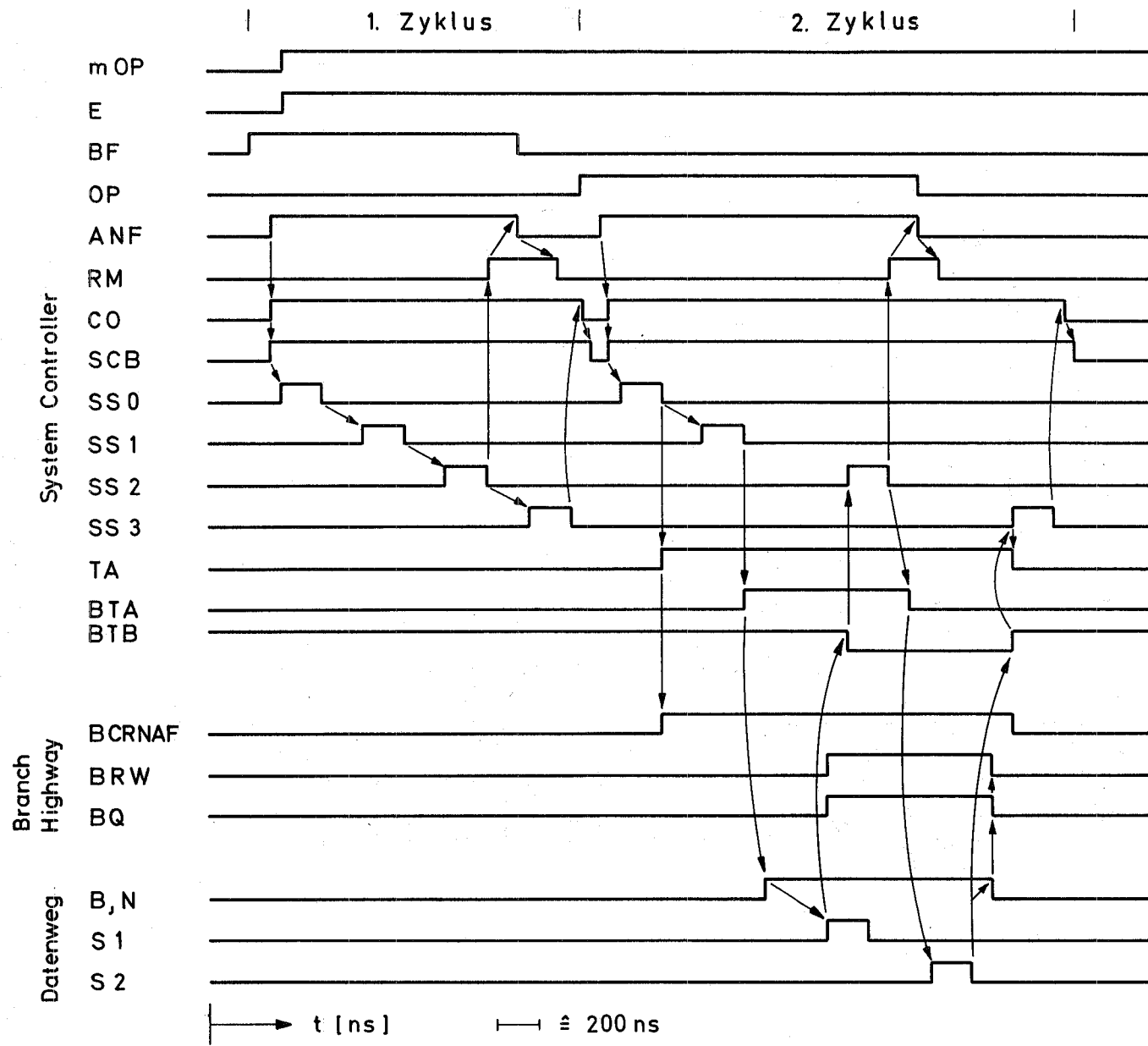
- 50 -



- 51 -



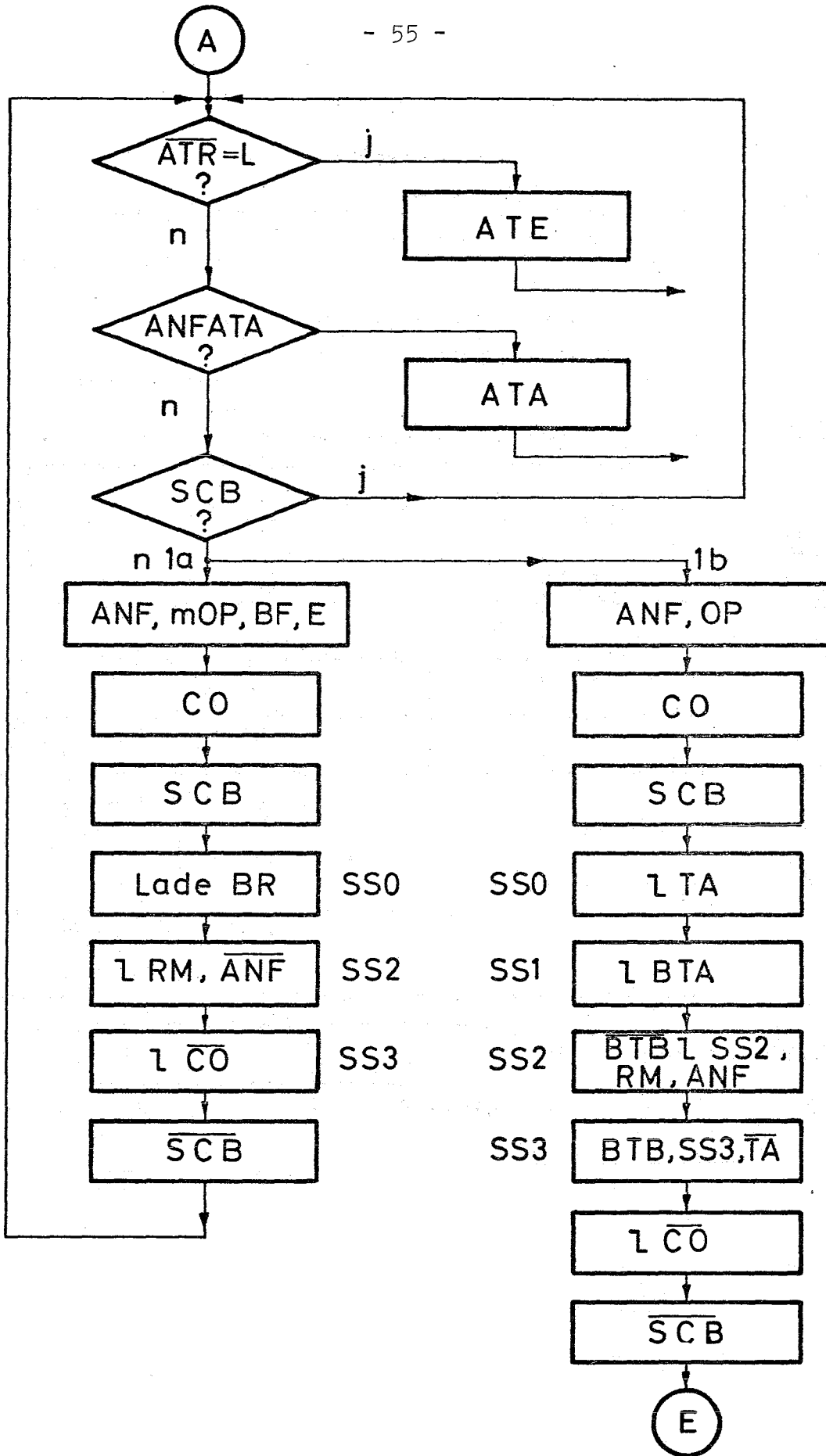


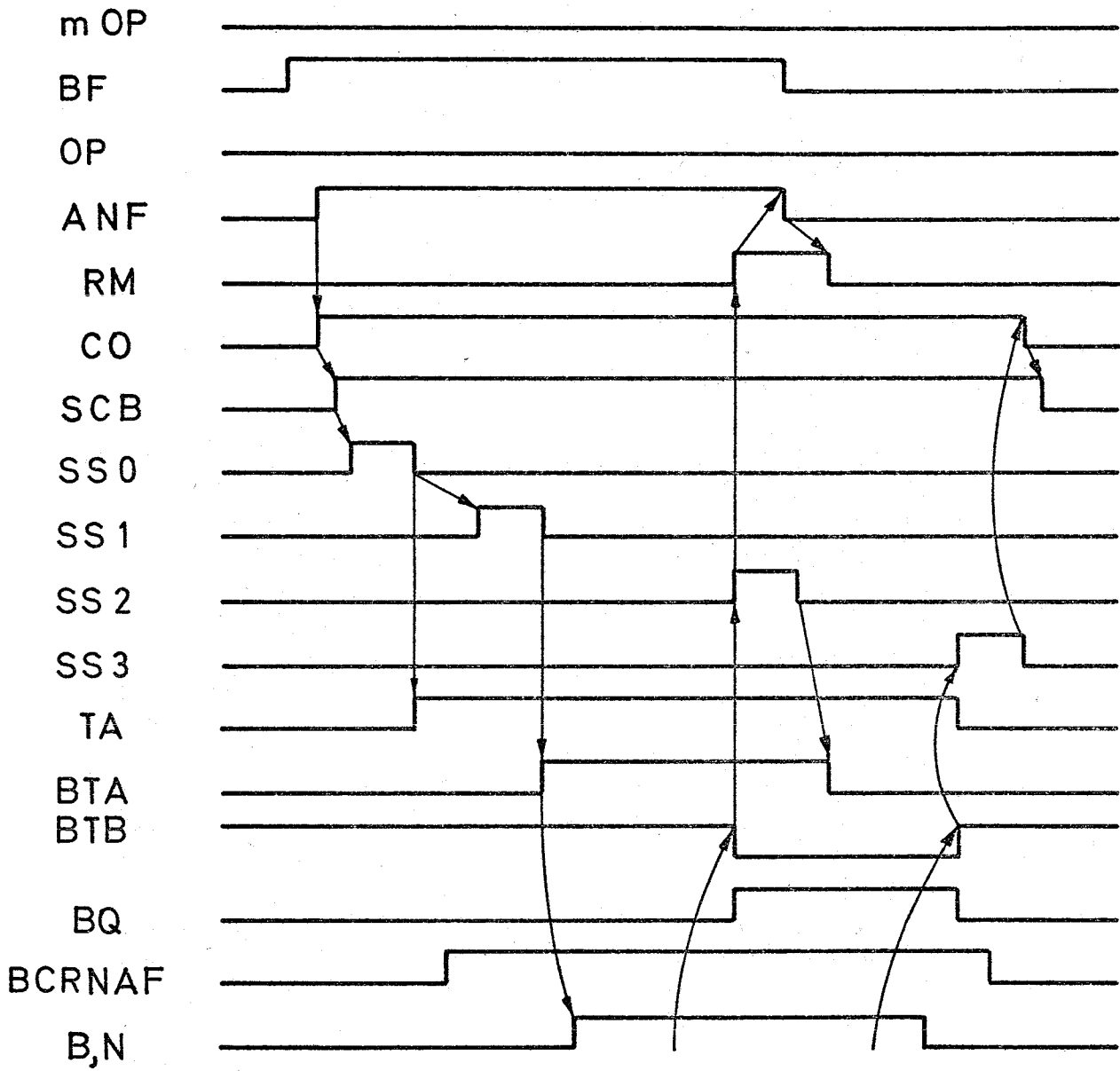


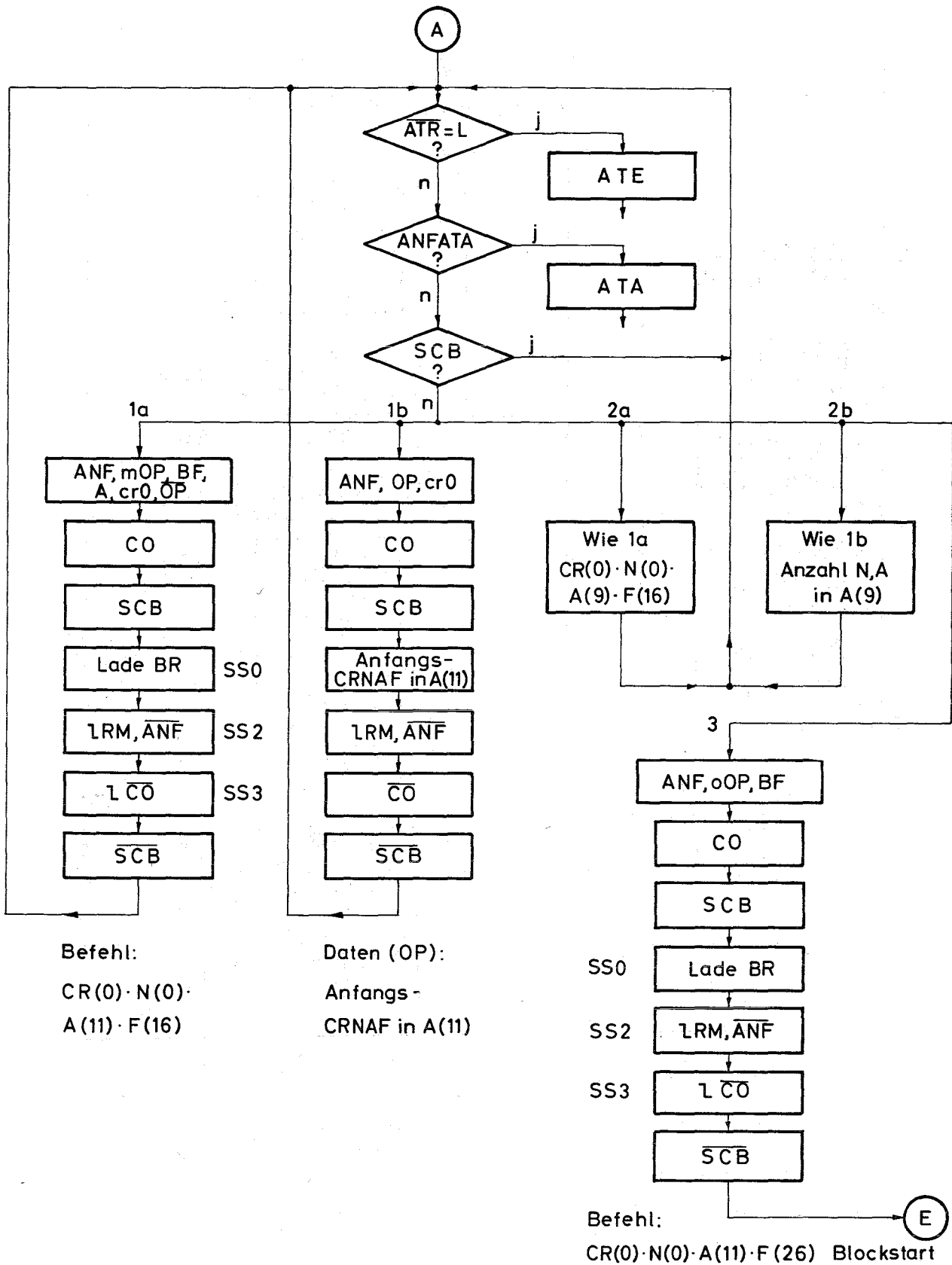
Befehl auf Branch Highway

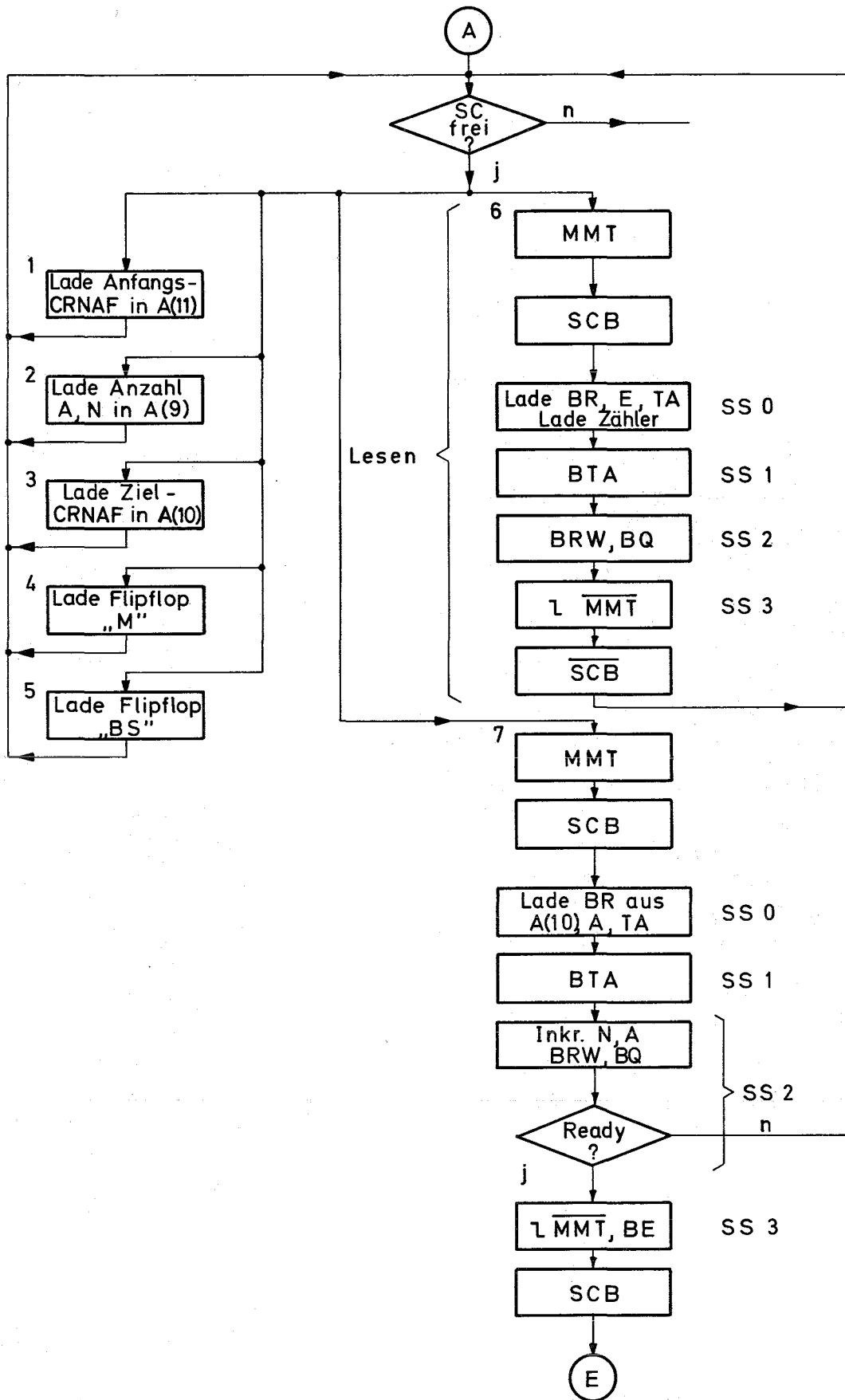
Fig. 17

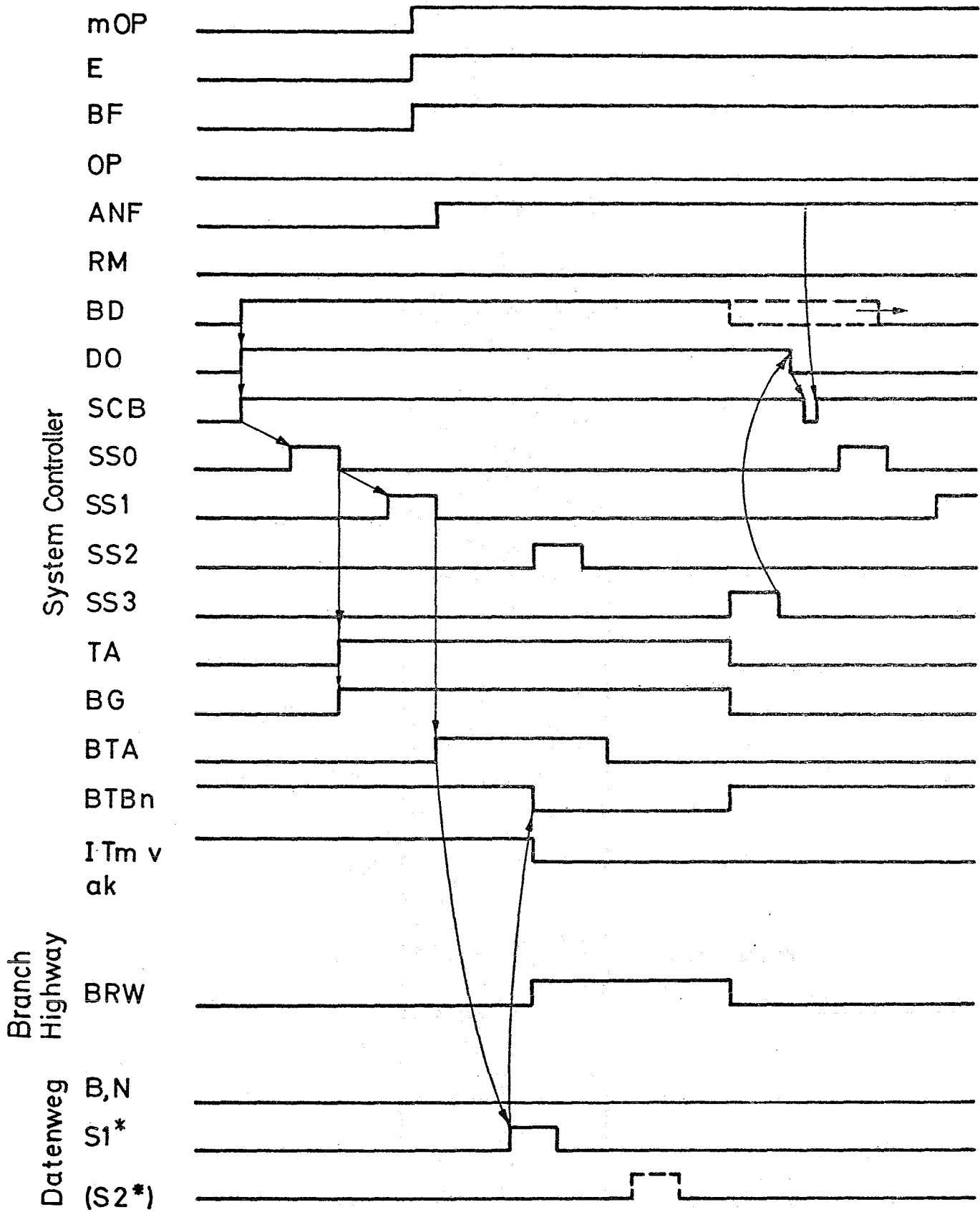
Timing für Lesebefehle





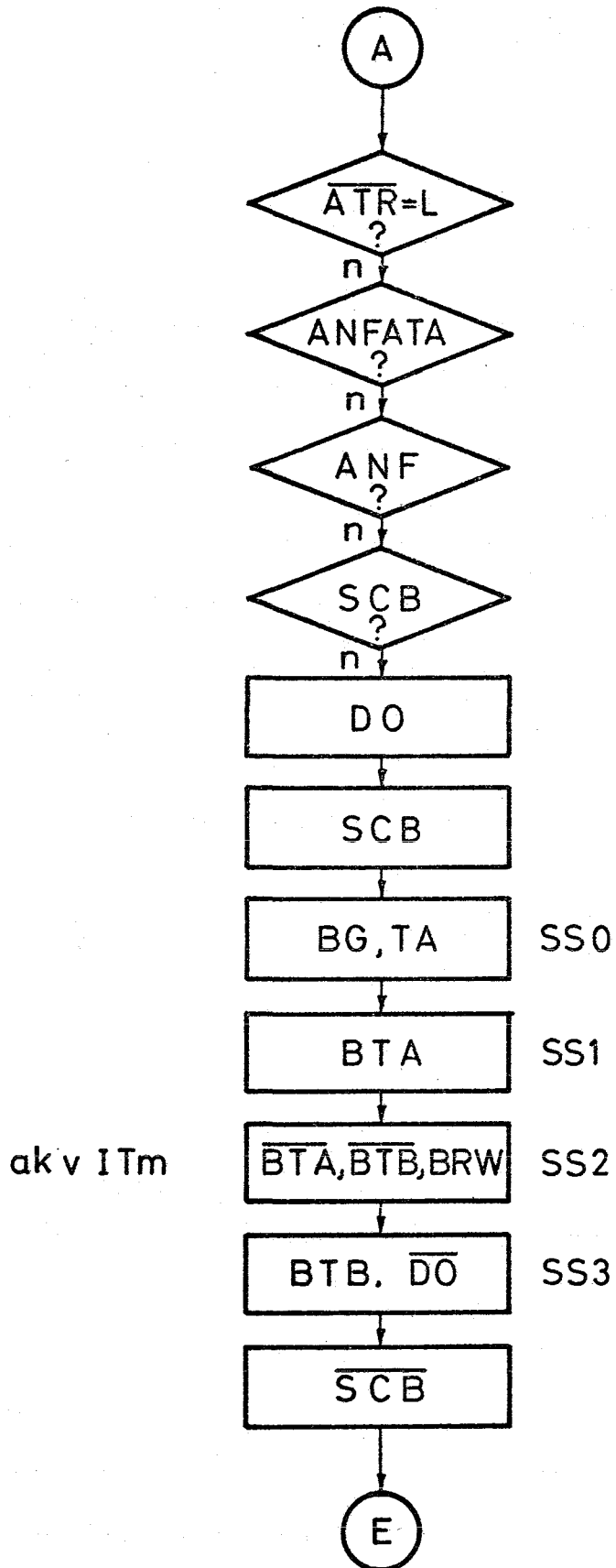


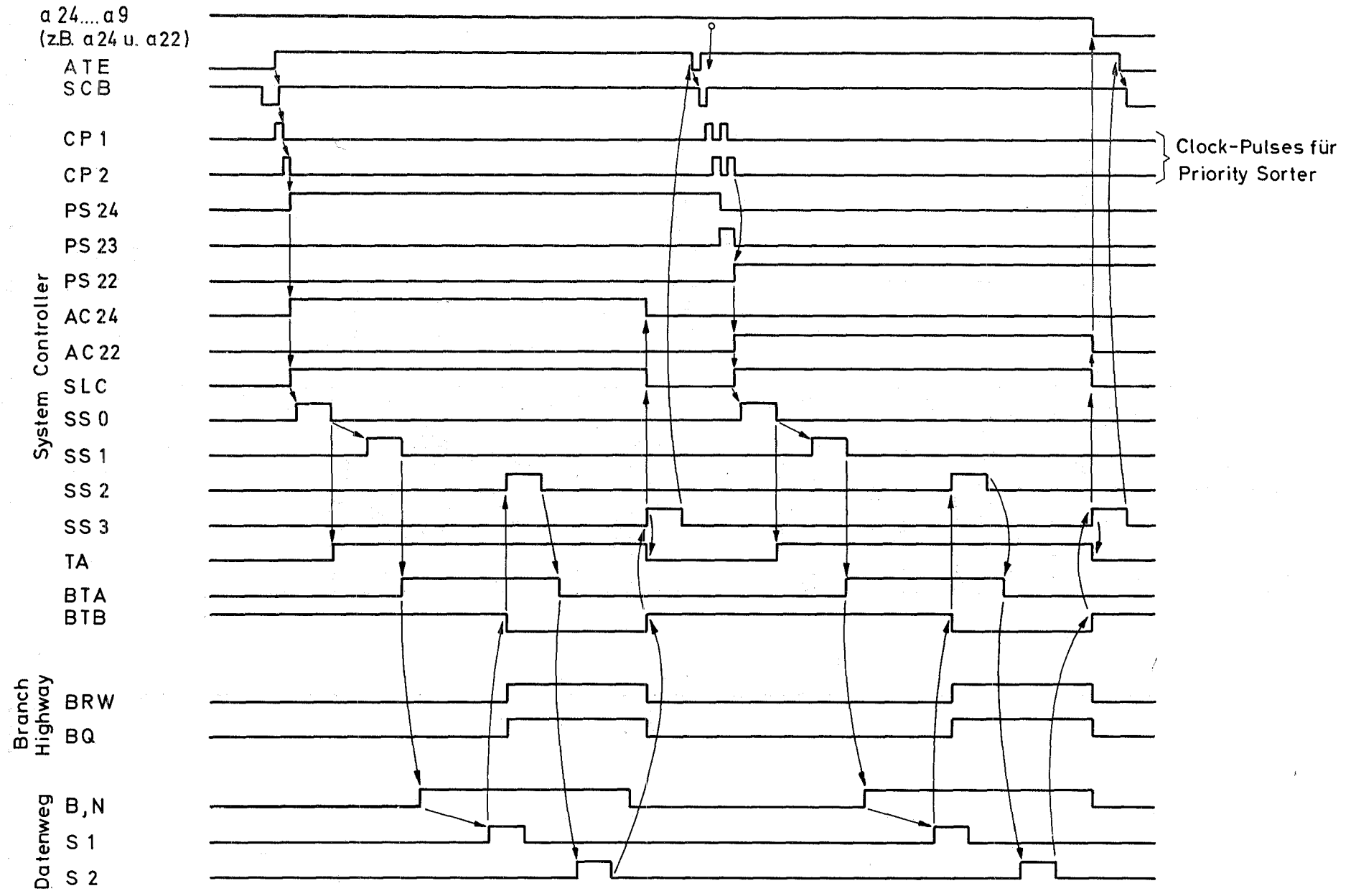


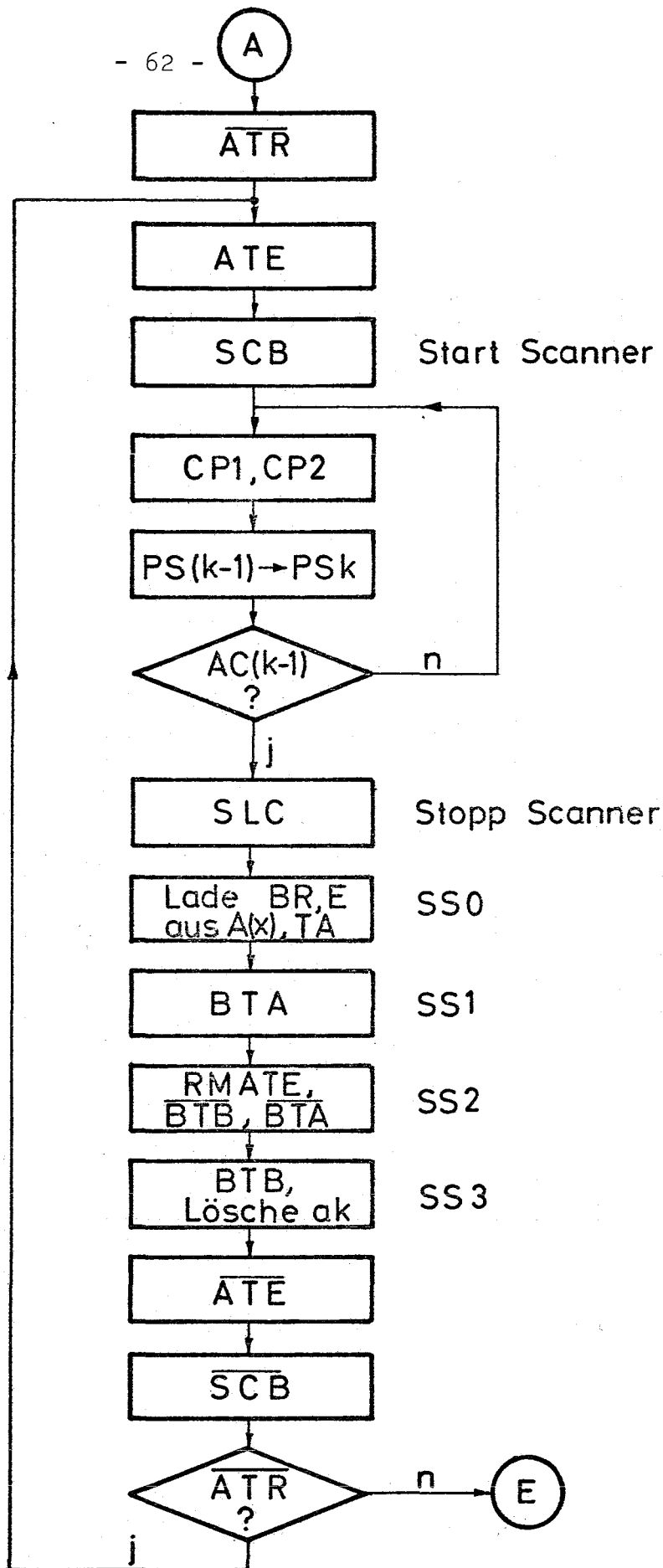


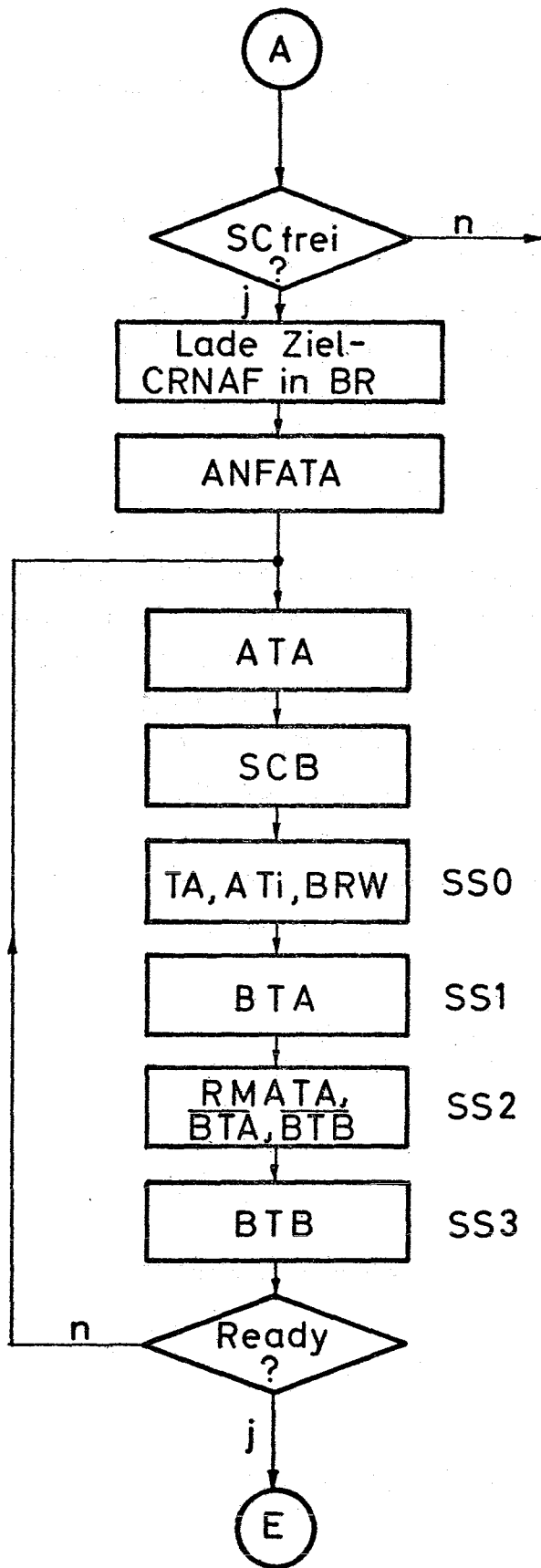
420 | Fig.22 Timing-Diagramm für BD/BG-Operationen

GfK
LEM









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 27 Subadressen gebraucht. Daher muß mit Funktionscodes der Gruppen 1 und 2 (s. Tabelle IV in [1]) gearbeitet werden. So werden 16 Lesebefehle für autonome Transfers in 16 Vorratsspeichern zu je 12 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 Lesebefehle	F(17)
⋮			
⋮			
⋮			
⋮			
⋮			
⋮			
⋮			
⋮			
⋮			
⋮			
⋮			
⋮			
⋮			
⋮			
A(15)			

Subadressen mit Funktionen der Gruppe 1

A(15)	Statusregister	F(0)
A(14)	Control-Register	nicht eingebaut
A(13)	Modul-Erkennungsregister	nicht eingebaut
A(12)	frei	
A(11)	Register für Anfangs-CRNAF	F(16); F(26)

A(10)	Register für Ziel CRNAF	F(16)
A(9)	Pufferspeicher für Anzahl der N- und A-Transfers	F(16)
A(8)	1-Bit-Speicher für M (Einstellung der Betriebsart Modul-zu-Modul-Transfer)	F(26)
A(7)	1-Bit-Speicher für ZYM (zyklischer Modul-zu-Modul-Transfer)	F(24); F(26)
A(6)	1-Bit-Speicher für ZYB (zyklischer Blocktransfer)	F(24); F(26)
A(5)	Select Crate Register (SCR)	F(16)
A(4)	1-Bit-Speicher für SQ (sequentielles Lesen)	F(24); F(26)
A(3)	frei	
A(2)	Befehlsregister, auslesbarer Teil (BR2)	F(0)
A(1)	1-Bit-Speicher für ESC (Enable System Controller)	F(24); F(26)
A(0)	frei	

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

2. Befehlsliste

Befehle mit Funktionen der Gruppe 2

CR(0) · N(0) ·	$\left\{ \begin{array}{l} A(0) \\ A(1) \\ \cdot \\ \cdot \\ \cdot \\ A(15) \end{array} \right\}$	· F(17)	Einspeichern der Lesebefehle für den auto- nomen Transfer
----------------	--	---------	--

Befehle mit Funktionen der Gruppe 1

CR(O)·N(O)·A(15)·F(O)	Lies Statusregister (Read SR)
CR(O)·N(O)·A(11)·F(16)	Lade Register für Anfangs-CRNAF
CR(O)·N(O)·A(11)·F(26)	Blockstart ("BS") der erste Hol-Befehl startet den Blocktransfer wirklich.
CR(O)·N(O)·A(10)·F(16)	Lade Register für Ziel-CRNAF
CR(O)·N(O)·A(9)·F(16)	Lade Register für Anzahl der Transfers in N und A

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

CR(O)·N(O)·A(8)·F(26)	Enable Modul-zu-Modul-Transfer ("M")
CR(O)·N(O)·A(7)·F(26)	Enable zyklischen Modul-zu-Modul- Transfer "ZYM"
CR(O)·N(O)·A(7)·F(24)	Disable zyklischen Modul-zu-Modul- Transfer "ZYM"
CR(O)·N(O)·A(6)·F(26)	Enable zyklischen Blocktransfer "ZYB"
CR(O)·N(O)·A(6)·F(24)	Disable zyklischen Blocktransfer "ZYB"
CR(O)·N(O)·A(5)·F(16)	Lade Select Crate Register SCR mit Crate Pattern
CR(O)·N(O)·A(4)·F(26)	Enable sequentielles Lesen "SQ"
CR(O)·N(O)·A(4)·F(24)	Disable sequentielles Lesen "SQ"
CR(O)·N(O)·A(2)·F(O)	Lies Inhalt des Befehlsregister aus BR2 ("PBR")
CR(O)·N(O)·A(1)·F(26)	Enable System Controller ("ESC")
CR(O)·N(O)·A(1)·F(24)	Disable System Controller (" $\overline{\text{ESC}}$ ")

3. Liste der Signalnamen

BF	<u>B</u> efehl	SQ	Sequentielle Lese/Schreib-Operationen sind erlaubt
OP	<u>O</u> perand	ZYM	Zyklischer Modul-zu-Modul-Transfer Der System Controller benutzt seine "Freizeit" für diese Transfers
mOP	mit <u>O</u> perand ($\overline{mOP} \hat{=} oOP$, ohne Operand)	ZYB	Zyklischer Blocktransfer Das Signal \overline{BE} wird dabei unterdrückt.
E	<u>E</u> ingabe ($\overline{E} \hat{=} A$, Ausgabe)	DO	Demand Operation (Alarm-Operation)
\overline{CL}	<u>C</u> lear not	CO	Command Operation (Befehls-Operation)
\overline{CRnOL} ($1 \leq n \leq 7$)	<u>C</u> rate not <u>o</u> n- <u>l</u> ine	MMT	Modul-zu-Modul-Transfer
LBR	Lade <u>B</u> efehlsregister vom Rechner	ATE	Autonomer Transfer in <u>E</u> ingabe-Richtung
LAC	Lade Befehlsregister aus dem Speicher für den <u>A</u> nfangs-CRNAF	ATA	Autonomer Transfer in <u>A</u> usgabe-Richtung
LZC	Lade Befehlsregister aus dem Speicher für den <u>Z</u> iel-CRNAF	ANF	Rechner- <u>A</u> nforderung
LLB	Lade Lesebefehl ins Befehlsregister aus dem Speicher für autonome Lesebefehle	RM	<u>R</u> ückmeldung
STBR	Strobe für das ganze <u>B</u> efehlsregister außer den Teilen A und N	ANFATE	Rechner- <u>A</u> nforderung
LAF	Lade <u>A</u> -Teil des Befehlsregisters aus der Adreßfortschaltung A	RMATE	<u>R</u> ückmeldung
STBA	Strobe für das <u>B</u> efehlsregister, Teil <u>A</u>	ANFATA	Rechner- <u>A</u> nforderung
LNF	Lade <u>N</u> -Teil des Befehlsregisters aus der Adreßfortschaltung N	RMATA	<u>R</u> ückmeldung
STBN	Strobe für das <u>B</u> efehlsregister, Teil <u>N</u>	SCB	System Controller <u>B</u> usy
IN	<u>I</u> nkrementiere Stationsnummer <u>N</u>	SSO ... SS3	Uhrimpulse des Timing-Generators
IA	<u>I</u> nkrementiere Subadresse <u>A</u>	\overline{ATR}	<u>A</u> utonomous Transfer not <u>R</u> eady
PBR	<u>P</u> rüfe Inhalt des <u>B</u> efehlsregisters	UB	<u>U</u> nerlaubte <u>B</u> etriebsart
IZN	Strobe zum <u>I</u> nkrementieren von <u>Z</u> ähler und Adreßfortschaltung <u>N</u>	FF	<u>F</u> ehlfunktion
SZN	Strobe zum parallelen Laden von <u>Z</u> ähler und Adreßfortschaltung <u>N</u>	ak ($9 \leq k \leq 24$)	Anforderung für autonomen Transfer in Eingabe-Richtung
LZNI	Lade <u>Z</u> ähler und Adreßfortschaltung <u>N</u> aus <u>i</u> nterner Quelle (Anfangs-CRNAF)	PSK ($9 \leq k \leq 24$)	Priority <u>S</u> orter steht auf Stufe k
IZA	Strobe zum <u>I</u> nkrementieren von <u>Z</u> ähler und Adreßfortschaltung <u>A</u>	ACK ($9 \leq k \leq 24$)	Autonomer <u>C</u> RNAF
SZA	Strobe zum parallelen Laden von <u>Z</u> ähler und Adreßfortschaltung <u>A</u>	SLC	<u>S</u> tart <u>L</u> ese-CRNAF (bei autonomer Eingabe)
LZAI	Lade <u>Z</u> ähler und Adreßfortschaltung <u>A</u> aus <u>i</u> nterner Quelle (Anfangs-CRNAF)	ITm ($1 \leq m \leq 8$)	<u>I</u> nterrupt-Anforderung
NR	<u>N</u> Transfers <u>R</u> eady (Overflow des N-Zählers)	\overline{LIT}	<u>L</u> ösche nicht <u>I</u> nterrupt
\overline{RNR}	<u>R</u> eset not <u>N</u> R	\overline{LITm} ($1 \leq m \leq 8$)	<u>L</u> ösche nicht <u>I</u> nterrupt-Anforderung ITm
AR	<u>A</u> Transfers <u>R</u> eady (Overflow des A-Zählers)	TOn	<u>T</u> ime-out-Signal, Anzapfung n
\overline{RAR}	<u>R</u> eset not <u>A</u> R	DRB	<u>D</u> isplay <u>R</u> ead <u>B</u> uffer
M	Enable Betriebsart <u>M</u> odul-zu-Modul-Transfer	DCR	<u>D</u> isplay <u>C</u> ommand <u>R</u> egister
BE	<u>B</u> lock <u>e</u> nde	DDR	<u>D</u> isplay <u>D</u> emand <u>R</u> egister
BS	Blockstart Statisches Signal während des gesamten Blockes bis zur Unterbrechung oder Blockende.		
ESC	Enable System Controller Ohne dieses Signal kann der System Controller nicht mit dem CAMAC-Branch verkehren.		

Hauptzustände des System Controller

Timing-Signale für programmierten Transfer

Timing-Signale für autonomen Transfer in Eingabe-Richtung

Timing-Signale für autonomen Transfer in Ausgabe-Richtung

