



Interncode

ZUSE KG · BAD HERSFELD
Elektronische Rechenanlagen



Der Interncode der ZUSE Z 31

Einleitung

Diese Schrift befaßt sich nur mit der Wirkungsweise der Internbefehle der ZUSE Z 31. Aufbau und Arbeitsweise dieser Rechenanlage werden hier nicht erläutert. Dazu wird auf die Beschreibung

"Einführung in die Arbeitsweise der programmgesteuerten elektronischen Rechenanlage ZUSE Z 31"

verwiesen.

Darin werden u.a. behandelt:

Informationsdarstellung (Zahl, Befehl, Text),
Speicherwerk, Rechenwerk, Leitwerk,
Ein- und Ausgabe der Informationen
Bedienungspult usw.

Dort werden auch alle Begriffe erklärt, die zum Verständnis des Interncodes notwendig sind.

Wir können uns daher hier darauf beschränken, die Wirkung der Befehle in der symbolischen Schreibweise in Form von Kennzeichen darzustellen und nur wenige Erläuterungen zu geben.

Die Programmierung im Interncode der ZUSE Z 31 gestattet es, den sehr flexiblen Befehlscode der Anlage voll auszunutzen, d.h. befehlssparende und doch schnelle Programme aufzustellen.

Alle Grundprogramme der ZUSE Z 31 wurden deshalb im Interncode programmiert. (s.a. "Beschreibung der Grundprogramme der ZUSE Z 31"). Diese Programme sind in einem Programmspeicher fest verdrahtet und können als Unterprogramme aufgerufen werden.

Für bestimmte Anwendungen kann es vorteilhaft sein, auch Hauptprogramme im Interncode zu programmieren. Im allgemeinen ist jedoch die einfachere Programmierung im Externcode der ZUSE Z 31 vorzuziehen. Der Externcode der ZUSE Z 31 wird in der Druckschrift "ZUSE Z 31-Programmierungsanleitung" beschrieben;

	<u>Inhaltsübersicht</u>	Seite
	Einleitung	1
1.	Darstellung der Internbefehle	4
2.	Schreibweise der Internbefehle	7
3.	Erklärung der Internbefehle	7
3.1.	Sprungbefehle	11
3.2.	Transportbefehle	11
3.2.1.	Transport von <sn> bzw. <pn>	11
3.2.2.	Transport von <Y> mit dem Vorzeichen des X-Registers	12
3.2.3.	Vertauschen der Rechenregisterinhalte	12
3.2.4.	Transport des Bedingungspeicherblocks	12
3.2.5.	Gemeinsamer Transport des Bedingungspeicher- blocks, der Zählregister und des Rückkehr- adressenspeichers	13
3.2.6.	Transport der Folgeadresse nach RAS	13
3.2.7.	Blocktransfer	14
3.2.7.1.	Transport aller Wörter eines Blocks..	14
3.2.7.2.	Transport einer bestimmten Anzahl von Wörtern	15
3.2.7.3.	Magnetband-Vorlauf	15
3.3.	Befehle zur Ausführung arithmetischer Operationen	16
3.3.1.	Addition und Subtraktion	16
3.3.2.	Übertrag	16
3.3.3.	Zählbefehle	17
3.3.3.1.	Zählen in den Zählregistern	17
3.3.3.2.	Zählen in den Rechenregistern	17
3.3.3.3.	Zählen in jeder beliebigen Schnell- speicherzelle	17
3.3.4.	Befehle zum Adressenrechnen	17
3.3.4.1.	Adresse „n“ des Befehles als Operand	17
3.3.4.2.	Adressensubstitution	18
3.3.4.3.	Adressenmodifikation	20
3.4.	Verschieben der Rechenregisterinhalte	20
3.5.	Befehle zur Ausführung logischer Operationen	22
3.5.1.	Intersektion ("Und-Verknüpfung")	22
3.5.2.	Kongruenzvergleich	22
3.5.3.	Disjunktion ("Oder-Verknüpfung")	23

	Seite
3.6. Markieren der Zählregister und Bedingungsspeicher	24
3.7. Bedingte Befehle	24
3.7.1. Bedingungen bezüglich $\langle ZR1 \rangle$ und $\langle ZR2 \rangle$	24
3.7.2. Bedingungen bezüglich der Stellung der Bedingungsspeicher B1...B6	25
3.7.3. Bedingungen bezüglich $\langle X \rangle$ und $\langle Y \rangle$	25
3.7.4. Bedingungen aufgrund der letzten Vergleichs- operationen	25
3.7.5. Wiederholungsbefehle	25
3.8. Programmstop	26
4. Befehlskombinationen	26
4.1. Allgemeines	26
4.2. Hinweise zur Verwendung der Kombinationstabellen	26
4.3. Kombinationsübersicht I	29
4.4. Kombinationsübersicht II (Befehle mit Zusatz C)	30
4.5. Kombinationsübersicht III (Sonderbefehle)	31

1. Darstellung der Internbefehle

Ein Internbefehl, im folgenden nur "Befehl" genannt, kann in der Rechenanlage erst dann die gewünschte Operation auslösen, wenn er sich im Befehlsregister der Anlage befindet. Jeder Befehl besteht aus 11 Dezimalziffern, die im folgenden mit D1-D11 bezeichnet werden. Jede Dezimalziffer im Befehl hat eine bestimmte Bedeutung. Aus mnemotechnischen Gründen sind den verschiedenen Dezimalziffern im Befehl mit Ausnahme der Adressenstellen (D4-D1) Symbole zugeordnet.

Nachstehende Tabelle zeigt die Zuordnung dieser Symbole zu den Ziffern des Befehles.

Dez. Wert	Kennzeichen- Dezimalstelle D11	Bedingungs- teil		Operationsteil				Adressenteil			
		D10	D9	D8	D7	D6	D5	D4	D3	D2	D1
0	+										
1	Normalbefehl	(J1)	(NE)	A	R	V1	E				
2	G	(J2)	(PO)	S	L	Z1	P				
3	(W)	(J3)	(NU)	AT	RR	Ji	V2				
4	TE	(N4)	(SU)	ST	LL	Ni	Z2				
5	TV	(N5)	(NG)	U	ED	VX	I1				
6	VB	(N6)	(GR)	K	Xi+	ZX	I2				
7	(Y0)	(YZ)	(KL)	O	Xi-	C	VY				
8	(10)	(1Z)	(Y5)	B	Z0	Q	ZY				
9	-	(20)	(2Z)	T	M	F	Y				

Die Kennzeichendezimale D11 entscheidet, ob das Wort (s. ZUSE Z 31 - Beschreibung) überhaupt ein Befehl sein soll. Die Ziffern 1, 2, 3, 6, 7, 8 in D11 kennzeichnen verschiedene Befehlstypen.

Der Bedingungsteil (D10, D9) gestattet es, jeden Befehl bedingt zu geben. Wird in D10 und D9 je eine Bedingung gegeben, so wird der Befehl nur ausgeführt, wenn beide Bedingungen erfüllt sind. Die Ziffer Null bedeutet "keine Bedingung" in der entsprechenden Dezimalstelle.

In jeder der Dezimalstellen D8-D5 des Operationsteiles kann eine Ziffer gegeben sein. Es können somit bis zu vier verschiedene Operationen durch einen Befehl gesteuert werden, sofern sich die einzelnen Operationen nicht gegenseitig ausschließen. Eine Übersicht am Ende dieser Abhandlung gibt Auskunft über die Kombinationsmöglichkeiten der einzelnen Befehlssymbole. Die Ziffer Null in einer Dezimalstelle bedeutet "keine Operation" für diese Dezimalstelle.

Normalerweise bewirkt jedes Befehlssymbol die ihm zugewiesene Operation. Bestimmten Befehlskombinationen kommt jedoch eine Sonderbedeutung zu. Sie werden deshalb bei der Erklärung ausdrücklich als Sonderbefehle gekennzeichnet.

In den Dezimalstellen D4-D1 kann eine Adresse angegeben werden. Steht in D5 das Befehls-Symbol P, so bezieht sich die Adresse auf eine bestimmte Programmspeicherzelle, sofern nicht vor der Befehlsausführung eine Modifikation oder Substitution der Adresse durchgeführt wird (Xi+,Xi- oder G-Befehl). Fehlt das Symbol P, so haben die Zahlen in den Stellen D4-D1 folgende Bedeutung:

0000 - 0010 interne Sonderadressen,
0011 - 0999 externe Sonderadressen,
1000 - 9999 Schnellspeicheradressen.

Durch interne Sonderadressen werden Register, durch externe Sonderadressen Anschlußgeräte angewählt.

Nachstehende Tabelle zeigt die Zuordnung der Adressen.

Sonderadressen

D4	D3	D2	D1	
0	0	0	0	Nullspeicher (liefert beim Aufruf dezimale Nullen)
0	0	0	1	frei
0	0	0	2	Konstantenregister
0	0	0	3	Befehlszählregister
0	0	0	4	X-Register
0	0	0	5	Rückkehradressenspeicher
0	0	0	6	Y-Register
0	0	0	7	Zählregister 1
0	0	0	8	Zählregister 2
0	0	0	9	Schnelles Multiplizierwerk
0	0	1	0	Bedingungsspeicherblock
0	0	1	1	1. Lochstreifenleser (ZUSE-Code)
0	0	1	2	1. Lochstreifenleser (beliebiger Lochstreifencode)
0	0	1	3	2. Lochstreifenleser (ZUSE-Code)
0	0	1	4	2. Lochstreifenleser (beliebiger Lochstreifencode)
0	0	1	5	frei
0	0	1	6	1. Lochstreifenstanzer (Zahlen)
0	0	1	7	1. Lochstreifenstanzer (ZUSE-Code)
0	0	1	8	1. Lochstreifenstanzer (beliebiger Code)
0	0	1	9	2. Lochstreifenstanzer (s. 0016)
0	0	2	0	2. Lochstreifenstanzer (s. 0017)
0	0	2	1	2. Lochstreifenstanzer (s. 0018)
0	0	2	2	frei für Sonderzwecke
0	0	2	3	frei für Sonderzwecke
0	0	2	4	frei für Sonderzwecke
0	0	2	5	Schreibmaschine (Zahlen)
0	0	2	6	Schreibmaschine (ZUSE-Code)
0	0	2	7	Weitere externe (z.B. für Lochkartenlese- und Sonderadressen -stanzgeräte, Zeilendrucker usw.) Sonderadressen
0	9	9	9	Sonderadressen

Schnellspeicheradressen

1000-1999	Grundeinheit (1000 Worte); Indexregister 1000-1009
2000-5999	Zusatzschrank Nr. 1 (4000 Worte)
6000-9999	Zusatzschrank Nr. 2 (4000 Worte)

Programmspeicheradressen

0-1299	Haupt- bzw. Spezialprogramme
1300-2599	Grundprogramme

2. Schreibweise der Internbefehle

Die vom Programmierer aufgestellten Befehle werden im allgemeinen über Lochstreifen (oder Lochkarten) in die Anlage eingegeben. Die Symbole werden auf dem Lochstreifen oder der Lochkarte in der Reihenfolge abgeloht, in der sie geschrieben wurden. Es sind deshalb bestimmte Regeln beim Schreiben der Befehle einzuhalten, damit das Befehlsprogramm die Symbole der Befehle in die zugehörigen Ziffern umwandelt und diese in die richtige Dezimalstelle des betreffenden Befehles setzt.

- 1.) Soll in einer der Dezimalstellen D10-D5 keine Bedingung oder keine Operation gegeben sein, so darf keine Null geschrieben werden. (Das bedeutet für die Praxis eine Vereinfachung).
- 2.) Handelt es sich um einen Normalbefehl, so wird D11 nicht besonders gekennzeichnet (es ist kein Symbol dafür vorhanden).
- 3.) Bedingungen müssen am Anfang des Befehles gegeben werden. Die Symbole müssen in Klammern gesetzt werden; bei verschiedenen Bedingungen können die jeweiligen Symbole in einer Klammer stehen und durch Punkte getrennt werden.
- 4.) Die Operationssymbole in D8-D5 können in beliebiger Reihenfolge geschrieben werden.
- 5.) Zwischen die Operationssymbole können aus Gründen der besseren Übersicht Punkte gesetzt werden?
- 6.) Innerhalb eines Befehles dürfen keine Zwischenräume geschrieben (geloht) werden. Ein Zwischenraum kennzeichnet das Ende des Befehles ebenso wie Wagenrücklauf - Zeilenvorschub, Tabulatorsprung, Komma oder Semikolon.

3. Erklärung der Internbefehle

Bei der Erklärung werden die Befehle in bestimmte Gruppen zusammengefaßt (Sprungbefehle, Transportbefehle, arithmetische Befehle usw.). In den meisten Fällen stellt ein Symbol in einer der Dezimalstellen D8-D5 schon einen Befehl dar. Durch Hinzusetzen weiterer Symbole kann die Bedeutung des Befehles abgewandelt werden.

Im Interncode der ZUSE Z 31 gibt es drei Symbole, die für sich allein noch keinen Befehl darstellen, sondern nur zur Abwandlung eines anderen Befehles dienen.

Es sind dies die Symbole M, C und Y.

Sie bedeuten im einzelnen:

M Wird dieses Symbol in Verbindung mit einem Transportbefehl (B,T) gegeben, so wird der negierte Wert des Wortes (10er-Komplement) transportiert. In Verbindung mit dem Befehl Un wird das 9er-Komplement zur Intersektion verwendet. Das Symbol M dient außerdem in einigen Fällen zur Bildung der schon erwähnten Sonderbefehle.

C Dieses Symbol bewirkt, daß anstelle des Operanden, der bei der Operation nicht verändert wird, die Adressenstellen des Befehles eingesetzt werden. Da die Adresse nur vierstellig ist, werden dem Operationswerk als restliche Dezimalziffern (D5-D11) dezimale Nullen (OOLL) zugeführt.

An dieser Stelle sei erwähnt, daß in allen Fällen, in denen Register, die weniger als 11 Stellen enthalten, einen Operanden liefern, die fehlenden Stellen mit dezimalen Nullen aufgefüllt werden. Ist ein solches "nicht voll ausgebautes" Register Ziel einer Operation, so werden nur soviele Dezimalstellen des Ergebnisses übernommen, wie das Register aufnehmen kann, beginnend mit der niedrigsten Stelle des Ergebnisses. Die restlichen Dezimalstellen gehen verloren.

Y Dieses Symbol bewirkt, daß bei einem Befehl, bei dem sonst das X-Register definitionsgemäß an der Operation beteiligt war, das Y-Register alle bisherigen Funktionen des X-Registers übernimmt. (Lieferung des Operanden, Aufnahme des Ergebnisses).

Vor der Erklärung der einzelnen Befehle sei noch etwas über den Zeitbedarf gesagt.

Wenn es nicht besonders vermerkt ist, benötigt jeder Befehl zu seiner Ausführung eine Wortzeit.

Erfolgt der (lineare) Programmablauf aus dem Programmspeicher, so wird der Befehl jeweils in der Schaltzeit ins Befehlsregister gebracht. Für das "Holen des Befehles" wird also keine zusätzliche Zeit gebraucht.

Der (lineare) Ablauf des Programmes aus dem Schnellspeicher erfolgt im sogenannten Zweitaktbetrieb; in einer Wortzeit wird der Befehl "geholt", in einer weiteren ausgeführt usw.

Einige Befehle benötigen jedoch zu ihrer Ausführung mehrere Wortzeiten. Es wird je eine zusätzliche Wortzeit gebraucht, wenn einer der Operanden im Programmspeicher steht oder wenn eine Adressenmodifikation oder -substitution vor der Ausführung des Befehles durchgeführt wird.

Jeder Befehl, der wegen nicht erfüllter Bedingung nicht ausgeführt wird, benötigt ebenfalls nur eine Wortzeit.

Ein Wiederholungsbefehl erfordert $w \cdot k + 1$ Wortzeiten.

k = Anzahl der wiederholten Ausführungen des Befehles,

w = Anzahl der Wortzeiten für eine einmalige Ausführung des Befehles.

Symbole zur Erklärung der Befehle

Die folgenden Symbole werden bei der Erläuterung der Wirkungsweise der Befehle verwendet:

- X 1. Rechenregister, X-Register
 - Y 2. Rechenregister, Y-Register
 - b Befehlsregister
 - c Befehlszählregister
 - n eine max. 4-stellige Zahl im Adressenteil des Befehles, der im Befehlsregister steht. (Bereich von 0 bis 9999)
 - \tilde{n} modifizierter Adressenteil eines Befehles im Befehlsreg.
 - sn Schnellspeicherzelle, welche mit der Adresse n aufgerufen wird. Der Schnellspeicheradressenbereich erstreckt sich von 1000 bis 9999, wobei die Schnellspeicherzellen von 1000 bis 1009 als Indexzellen verwendbar sind.
 - pn Programmspeicherzelle, welche mit der Adresse n aufgerufen wird. Der Programmspeicheradressenbereich erstreckt sich von 0 bis 2599
 - RAS Rückkehradressenspeicher
 - ZR1 1. Zählregister
 - ZR2 2. Zählregister
 - Di Dezimale i eines Wortes; z.B. D10
 - ...i Dezimalstelle i von ...; z.B. X₁
 - ...i-j Dezimalstelle i bis j von ...; z.B. X₄₋₁
 - ...ia Bezeichnung der 1. bit-Stelle von ...i, z.B. X_{1a}
(entspr. ...ib, ...ic, ...id) z.B. für X1:
- | | | | |
|-----------------|-----------------|-----------------|-----------------|
| X _{1d} | X _{1c} | X _{1b} | X _{1a} |
|-----------------|-----------------|-----------------|-----------------|
- <...> Inhalt von ... z.B. <X>
 - ... => ... "ergibt": Der links vom Doppelpfeil angegebene Ausdruck ergibt den rechtsstehenden
 - ... -> ... "nach": Die links vom Pfeil angegebene Information wird nach der rechts angegebenen Stelle transportiert.
 - z' Eine ganze Zahl mit der Einerstelle auf D1, z.B. 1'
 - bin. 0 "binäre Null", d.h. alle bits der Tetrade sind Null
(0000)
 - dez. 0 "dezimale Null", d.h. eine Null im 3-Exzesscode (00LL)

In der Beschreibung der Wirkungsweise der Befehle werden im folgenden nur die Hauptoperationen **behandelt**, nicht aber Nebenoperationen, wie Regenerationen von Registern, die als Operand benutzt, dabei aber nicht verändert werden, sowie selbständige Kontrollabläufe ("Kreisen" von Registern), die in keiner Beziehung zur Operation selbst stehen.

3.1. Sprungbefehle

En	<sn>	-->	b
	Er	-->	c
Pn	<pn>	-->	b
	Pn	-->	c

Zusatz F bewirkt Notierung der Rückkehradresse in RAS:

EFn	<c> +1	-->	RAS
	<sn>	-->	b
	En	-->	c
PFn	<c> +1	-->	RAS
	<pn>	-->	c
	Pn	-->	c

3.2. Transportbefehle

3.2.1. Transport von <sn> bzw. <pn>

Bn	<sn>	-->	X
BYn	<sn>	-->	Y
Tn	<X>	-->	sn
TYn	<Y>	-->	sn
BCn	n	-->	X
BCYn	n	-->	Y
TCn	n	-->	sn

Transport einer Konstanten aus dem Programmspeicher:

BPN 1. Wortzeit: Pn --> c
 2. Wortzeit: <pn> --> X ; <c> +1 --> c

Das Programm wird an der Stelle pn+1
des Programmspeichers fortgesetzt.

BFPn 1. Wortzeit: <c>+1 --> RAS, Pn --> c
 2. Wortzeit: <pn> --> X, <RAS> --> c

Keine Unterbrechung des linearen Programm-
ablaufes aus dem Programm- oder Schnell-
speicher.

Der Zusatz M bewirkt bei diesen vorangegangenen Transport-
befehlen, daß der negierte Wert transportiert wird, z.B.

BMPn - <pn> --> X

TMyn - <Y> --> sn

3.2.2. Transport von <Y> mit dem Vorzeichen des X-Registers

TQYn <X₁₂; Y₁₀₋₁> --> sn

TMQYn -<X₁₂; Y₁₀₋₁> --> sn

3.2.3. Vertauschen der Rechenregisterinhalte (Sonderbefehle)

B6	<Y> --> X, <X> --> Y	6 und 4 sind die
BY4	<X> --> Y, <Y> --> X	den Y- und X-Registern
BM6	-<Y> --> X, <X> --> Y	zugeordneten internen
BYM4	-<X> --> Y, <Y> --> X	Sonderadressen

3.2.4. Transport des Bedingungspeicherblocks

Der unter der Sonderadresse 10 aufrufbare Bedingungspeicher-
block verfügt über 6 bit-Stellen (sogenannte Bedingungs-
speicher B1 bis B6), die einzeln über das Bedienungspult
oder per Programm in 0 bzw. 1 gesetzt und dadurch in Nein-
bzw. Ja-Stellung gebracht werden können.

B10 <B1> ... <B6> --> X_{1a-2b}
 bin.0 --> X_{2c-10d}
 Textkennzeichen 4 --> X_{11,12}

Der Inhalt der Bedingungspeicher bleibt, da
der Bedingungspeicherblock wie eine normale
Schnellspeicherzelle behandelt wird, erhalten.
Eine Kombination mit M ist verboten!

T10 $\langle X_{1a-2b} \rangle \rightarrow B1, \dots, B6$

TM10 $-\langle X_{1a-2b} \rangle \rightarrow B1, \dots, B6$

Durch den Zusatz Y werden die Befehle statt mit dem X-Register mit dem Y-Register ausgeführt.

3.2.5. Gemeinsamer Transport des Bedingungsspeicherblocks, der Zählregister und des Rückkehradressenspeichers

Diese 3 Register können durch einen Sonderbefehl in eine beliebige Schnellspeicherzelle transportiert werden, durch einen zweiten Sonderbefehl kann $\langle sn \rangle$ in entsprechender Weise auf die 3 Register aufgeteilt werden.

BFn	$\langle sn_{1a\dots 2b} \rangle \rightarrow B1, \dots, B6$	
	$\langle sn_3 \rangle \rightarrow ZR1$	Der Inhalt der Schnellspeicherzelle bleibt dabei unverändert.
	$\langle sn_4 \rangle \rightarrow ZR2$	
	$\langle sn_{6-10} \rangle \rightarrow RAS$	

TFn	$\langle B1 \rangle \dots \langle B6 \rangle \rightarrow sn_{1a\dots 2b}$	
	0 $\rightarrow sn_{2c, 2d}$	Der Inhalt von Bedingungsspeicher, Zählregister und RAS bleibt unverändert. Kombination mit M ist nicht erlaubt.
	$\langle ZR1 \rangle \rightarrow sn_3$	
	$\langle ZR2 \rangle \rightarrow sn_4$	
	bin. 0 $\rightarrow sn_5$	
	$\langle RAS \rangle \rightarrow sn_{6-10}$	
	4 $\rightarrow sn_{11}$	

3.2.6. Transport der Folgeadresse nach RAS

F $\langle c \rangle + 1' \rightarrow RAS$
 Der im Befehlszählregister stehende Sprungbefehl wird mit um 1 erhöhter Adresse im Rückkehradressenspeicher gespeichert.
 (Vgl. 3.1. Sprungbefehle).

3.2.7. Blocktransfer

Der Blocktransfer gestattet die Übertragung von Wortgruppen zwischen dem Nachschubspeicher und dem Schnellspeicher (in beiden Richtungen). Die Wörter sind im Nachschubspeicher in sogenannte Wortblöcke geteilt. Jeder Wortblock ist adressierbar. Die einzelnen Blöcke können verschiedene Länge haben und sind durch sogenannte Blocktrennungszeichen getrennt. Durch 2 Befehle können alle Wörter eines Blockes übertragen werden, durch einen zusätzlichen Befehl kann die Anzahl der zu übertragenden Wörter vorher bestimmt werden.

Die zur Erklärung des Blocktransfers verwendeten Adressensymbole haben folgende Bedeutung:

- n die um 1 verminderte Anzahl der zu übertragenden Wörter.
- m Blocknummer des Wortblockes im Nachschubspeicher.
- p vierstellige Adresse, wobei p_{4-3} die Anfangsadresse eines Hunderterblockes im Schnellspeicher (Blocknummer des Schnellspeichers) und p_{2-1} die Adresse des Nachschubspeichers angeben. Durch p_{2-1} wird also einer von mehreren möglichen Nachschubspeichern ausgewählt.

3.2.7.1. Transport aller Wörter eines Blockes

Es sind folgende Befehle nötig:

- VBm "Vorbefehl", hierdurch Angabe der Blockadresse
- WTP wiederholte Übertragung eines Wortes vom Nachschubspeicher in den Schnellspeicher. Die Übertragung wird durch das Blocktrennungszeichen beendet.

Es sei erwähnt, daß hier ausnahmsweise der Wiederholungsbefehl ohne eine Bedingung gegeben werden kann.

3.2.7.2. Transport einer bestimmten Anzahl von Wörtern

BCn-1 Wahl der Anzahl der zu übertragenden Wörter;
VBm Vorbefehl, s.o.
(WFO)TZXP

Der letzte Befehl bewirkt wieder den Transport vom Nachschubspeicher in den Schnellspeicher; durch die Bedingung (FO) in Verbindung mit dem Zählbefehl ZY wird die Übertragung beendet, wenn die gewünschte Anzahl der Wörter transportiert worden ist.

(WFO)BZXP bewirkt den Transport vom Schnellspeicher in den Nachschubspeicher.

3.2.7.3. Magnetband-Vorlauf

Um Wartezeiten der Rechenanlage beim Suchen des Wortblockes auf dem Magnetband zu vermeiden, kann man die Befehlsfolge

VBm
Fp₂₋₁

geben.

Das Magnetbandgerät (durch p₂₋₁ angewählt) sucht dann automatisch die Blockadresse m und bleibt stehen, nachdem der Block gefunden worden ist. Durch die Befehlsfolgen unter 3.2.7.1. oder 3.2.7.2. kann dann sofort der Blocktransfer eingeleitet werden. Während des Suchens kann die Rechenanlage beliebig operieren, da sie nach dem Befehl Fp₂₋₁ sofort zum nächsten Befehl des Programmes übergeht.

3.3. Befehle zur Ausführung arithmetischer Operationen

3.3.1. Addition und Subtraktion

Addition:

An $\langle X \rangle + \langle sn \rangle \rightarrow X$

AYn $\langle Y \rangle + \langle sn \rangle \rightarrow Y$

ATn $\langle sn \rangle + \langle X \rangle \rightarrow sn$

ATYn $\langle sn \rangle + \langle Y \rangle \rightarrow sn$

APn 1. Wortzeit: Pn $\rightarrow c$

2. Wortzeit: $\langle X \rangle + \langle pn \rangle \rightarrow X; \langle c \rangle + 1 \rightarrow c$

AFPn 1. Wortzeit: $\langle c \rangle + 1 \rightarrow RAS; Pn \rightarrow c$

2. Wortzeit: $\langle X \rangle + \langle pn \rangle \rightarrow X; \langle RAS \rangle \rightarrow c$

Subtraktion:

Sn $\langle X \rangle - \langle sn \rangle \rightarrow X$

SYn $\langle Y \rangle - \langle sn \rangle \rightarrow Y$

STn $\langle sn \rangle - \langle X \rangle \rightarrow sn$

STYn $\langle sn \rangle - \langle Y \rangle \rightarrow sn$

SPn 1. Wortzeit: Pn $\rightarrow c$

2. Wortzeit: $\langle X \rangle - \langle pn \rangle \rightarrow X; \langle c \rangle + 1 \rightarrow c$

SFPn 1. Wortzeit: $\langle c \rangle + 1 \rightarrow RAS; Pn \rightarrow c$

2. Wortzeit: $\langle X \rangle - \langle pn \rangle \rightarrow X; \langle RAS \rangle \rightarrow c$

3.3.2. Übertrag

Q Addition eines Übertrages, der in Y_{11} entstanden sein kann, auf die Stelle X_1 . Dieser Befehl dient vor allem zum Rechnen mit mehrfacher Zahlenlänge.

Operation:

$\langle Y_{12} \rangle + \langle X_{12} \rangle \rightarrow X_{12}$

$\langle Y_{12} \rangle + \langle X_{11} \rangle \rightarrow X_{11}$

$\langle Y_{12} \rangle + \langle X_2 \rangle \rightarrow X_2$

$\langle Y_{11} \rangle + \langle X_1 \rangle \rightarrow X_1$

Hierbei werden die Dezimalüberträge wie bei jeder Addition behandelt.

3.3.3. Zählbefehle

3.3.3.1. Zählen in den Zählregistern

V1	$\langle ZR1 \rangle + 1$	\rightarrow	ZR1
Z1	$\langle ZR1 \rangle - 1$	\rightarrow	ZR1
V2	$\langle ZR2 \rangle + 1$	\rightarrow	ZR2
Z2	$\langle ZR2 \rangle - 1$	\rightarrow	ZR2

3.3.3.2. Zählen in den Rechenregistern

VX	$\langle X \rangle + 1$	\rightarrow	X
ZX	$\langle X \rangle - 1$	\rightarrow	X
VY	$\langle Y \rangle + 1$	\rightarrow	Y
ZY	$\langle Y \rangle - 1$	\rightarrow	Y

3.3.3.3. Zählen in jeder beliebigen Schnellspeicherzelle

MVXn	$\langle sn \rangle + 1$	\rightarrow	sn	"Sonderbefehl"; M verliert
MZXn	$\langle sn \rangle - 1$	\rightarrow	sn	seine ursprüngliche Bedeutung

3.3.4. Befehle zum Adressenrechnen

3.3.4.1. Adresse „n“ des Befehles als Operand

C Die Adresse n vertritt denjenigen Operanden, der bei der normalen Operation, wenn diese ohne den Zusatz C ausgeführt wird, nicht verändert wird.

Folgende Kombinationen sind möglich:

ACn	$\langle X \rangle + n$	\rightarrow	X	
SCn	$\langle X \rangle - n$	\rightarrow	X	Vgl. Addition und Sub-
ATCn	$\langle sn \rangle + n$	\rightarrow	sn	traktion 3.3.1.
STCn	$\langle sn \rangle - n$	\rightarrow	sn	
UCn	$\langle X \rangle \wedge n$	\rightarrow	X	Vgl. Befehle zur Ausführung
KCn	$n \cong \langle X \rangle$	\rightarrow	X	logischer Operationen
OCn	$\langle X \rangle \vee n$	\rightarrow	X	3.5.

BCn	$n \rightarrow X$	Vgl. Transportbefehle
TCn	$n \rightarrow sn$	3.2.

Der Zusatz von M ist bei den drei Befehlen BCn, TCn und UCn möglich und verändert ihre Wirkung wie folgt:

BMCn	$- n \rightarrow X$	
TMCn	$- n \rightarrow sn$	
UMCn	$\langle X \rangle \wedge - n \rightarrow X$	(-n hier 9er-Komplement)

Durch Hinzusetzen von Y wird anstelle des X-Registers das Y-Register zur Operation verwendet.

3.3.4.2. Adressensubstitution

Gn In der ersten Operationszeit wird eine neue Adresse in den Adressenteil des im Befehlsregister stehenden Befehles eingesetzt; in der zweiten Operationszeit wird der Befehl dann mit der neuen Adresse ausgeführt.

1. Wortzeit: $\langle sn_{4-1} \rangle \Rightarrow \tilde{n}$

2. Wortzeit: Ausführung des Befehles mit \tilde{n}

z.B. Befehl GB1279 mit $\langle s1279 \rangle = 12345671654$

1. Wortzeit: $\langle s1279_{4-1} \rangle \rightarrow b_{4-1}$, d.h. 1654 $\Rightarrow \tilde{n}$

2. Wortzeit: $\langle s1654 \rangle \rightarrow X$

3.3.4.3. Adressenmodifikation

In der ersten Operationszeit wird der Adressenteil der ausgewählten Indexzelle zu dem im Befehlsregister stehenden Adressenteil des Befehles addiert.

Mit der Stelle D10 des Befehlswortes, die bei diesen Befehlen ihre Bedeutung als Bedingungsdezimale verliert, können für $i = 0, 1, 2, \dots, 9$ die 10 Indexzellen 1000...1009 angewählt werden.

In der zweiten Operationszeit wird der Befehl dann mit der umgerechneten Adresse ausgeführt.

Folgende zwei Indexbefehle sind möglich:

X_i+ ohne Veränderung der Indexzelle:

$$1. \text{ Wortzeit: } (n + \langle s1000+i \rangle_{4-1}) \Rightarrow \tilde{n}$$
$$\langle s1000+i \rangle \rightarrow s1000+i$$

2. Wortzeit: Ausführung des Befehles mit \tilde{n}

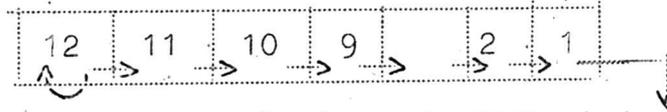
X_i+ mit Veränderung der Indexzelle:

$$1. \text{ Wortzeit: } (n + \langle s1000+i \rangle_{4-1}) \Rightarrow \tilde{n}$$
$$\tilde{n} \rightarrow s1000+i_{4-1}$$

2. Wortzeit: Ausführung des Befehles mit \tilde{n} .
 $\langle s1000+i \rangle_{11-5}$ bleibt erhalten

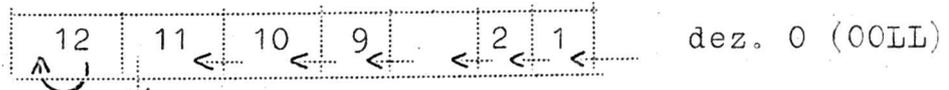
3.4. Verschieben der Rechenregisterinhalte

R Rechtsverschiebung im X-Register
 $\langle X_{12} \rangle$ bleibt erhalten; $\langle X_1 \rangle$ geht verloren:



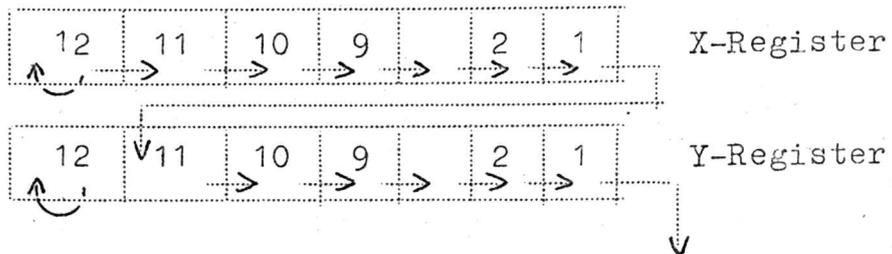
RY Rechtsverschiebung im Y-Register
 Anstelle $\langle X \rangle$ wird $\langle Y \rangle$ verschoben.

L Linksverschiebung im X-Register
 $\langle X_{12} \rangle$ bleibt erhalten; $\langle X_{11} \rangle$ geht verloren;
 dez. Null wird in X_1 geschoben:



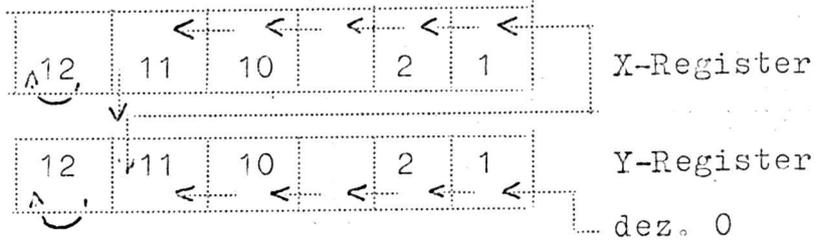
LY Linksverschiebung im Y-Register
 Anstelle $\langle X \rangle$ wird $\langle Y \rangle$ verschoben.

RR Gekoppelte Rechtsverschiebung von $\langle X \rangle$ und $\langle Y \rangle$
 $\langle X_{12} \rangle$ und $\langle Y_{12} \rangle$ bleiben erhalten;
 $\langle X_1 \rangle \rightarrow Y_{11}$; $\langle Y_1 \rangle$ geht verloren:



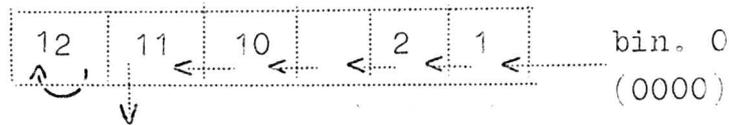
LL Gekoppelte Linksverschiebung von $\langle X \rangle$ und $\langle Y \rangle$

$\langle X_{12} \rangle$ und $\langle Y_{12} \rangle$ bleibt erhalten;
 $\langle Y_{11} \rangle \rightarrow X_1$; $\langle X_{11} \rangle$ geht verloren;
 dez. Null wird in Y_1 geschoben:



LD Linksverschiebung im X-Register mit bin. 0 nach X_1

$\langle X_{12} \rangle$ bleibt erhalten; $\langle X_{11} \rangle$ geht verloren;
 bin. Null wird in X_1 geschoben:



LDY Linksverschiebung im Y-Register mit bin. 0 nach Y_1

Anstelle von $\langle X \rangle$ wird $\langle Y \rangle$ verschoben.

3.5. Befehle zur Ausführung logischer Operationen

3.5.1. Intersektion ("Und-Verknüpfung")

Un $\langle X \rangle \wedge \langle sn \rangle \rightarrow X$ Herausschneiden beliebiger Stellen eines Wortes im X-Register.

Zwei entsprechende Dezimalstellen werden nach folgendem Schema verarbeitet:

Stelle in X (beliebiger Inhalt)	Dezimalziffer in sn	Resultat im X-Register
Z	0	0
Z	$\neq 0$	Z

UIn $\langle X \rangle \wedge - \langle sn \rangle \rightarrow X$; es wird das 9er-Komplement von sn gebildet. Durch Zusatz von Y wird das Y-Register anstelle des X-Registers zur Operation verwendet.

3.5.2. Kongruenzvergleich

Kn $\langle X \rangle \cong \langle sn \rangle$

Nach einem Vergleich von $\langle X \rangle$ und $\langle sn \rangle$ werden, sofern $\langle X \rangle$ ungleich $\langle sn \rangle$ ist, die entsprechenden Indikatoren NG und GR bzw. NG und KL gesetzt.

NG (nicht gleich) bei $\langle sn \rangle \neq \langle X \rangle$

GR (größer) bei $\langle sn \rangle > \langle X \rangle$

KL (kleiner) bei $\langle sn \rangle < \langle X \rangle$

Das Vergleichsergebnis

bleibt solange erhalten, bis ein neuer Kongruenzvergleich erfolgt. Die Indikatoren können durch entsprechende bedingte Befehle beliebig oft abgefragt werden (vgl. 3.7.4.).

Der Vergleich ist für alle Wortarten möglich, also auch für Textwörter in beliebigem Code.

Beim Vergleich werden die einzelnen Tetraden der Wörter verglichen; nur wenn alle Tetradenvergleiche Kongruenz ergeben, wird keine der drei angeführten Markierungen gesetzt.

Besteht eine Tetrade aus 4 Nullen, d.h. wird eine binäre Null zum Vergleich herangezogen, so wird der entsprechende Tetradenvergleich als kongruent gewertet. Eine dezimale 4 auf D11 oder D12 gilt ebenfalls als "gleich mit allem".

3.5.3. Disjunktion ("Oder-Verknüpfung")

On $\langle X \rangle \vee \langle sn \rangle \rightarrow X$ Binärstellenweiser Vergleich von $\langle X \rangle$ und $\langle sn \rangle$ nach folgendem Schema:

Binärstelle		
in X	in sn	Ergebnis in X
$\bar{1}$	$\bar{1}$	$\bar{1}$
$\bar{1}$	0	$\bar{1}$
0	$\bar{1}$	$\bar{1}$
0	0	0

Die Zusätze Y, P und PF verändern die Funktion der logischen Operationen in folgender Weise:

- Y Intersektion bzw. Vergleich bzw. Disjunktion erfolgt nicht mit dem X-Register sondern dem Y-Register.
- P Intersektion bzw. Vergleich bzw. Disjunktion erfolgt nicht mit $\langle sn \rangle$ sondern mit $\langle pn \rangle$.
Gleichzeitig Unterbrechung des linearen Programmablaufes, d.h. Sprung auf pn+1.

PF Intersektion bzw. Vergleich bzw. Disjunktion erfolgt nicht mit $\langle sn \rangle$, sondern mit $\langle pn \rangle$. Der lineare Programmablauf wird fortgesetzt.

3.6. Markieren der Zählregister und Bedingungspeicher

I1n $n_1 \rightarrow$ ZR1 Transport der letzten Stelle der
I2n $n_1 \rightarrow$ ZR2 Adresse des im Befehlsregister
stehenden Befehles nach einem der
beiden Zählregister.

Ji Der mit i bezeichnete, in der Dezimalstelle D10
des Befehlswortes gewählte Bedingungspeicher
wird in Ja-Stellung gebracht.
(i = 1,2...6)

Ni Der mit i bezeichnete, in der Dezimalstelle D10
des Befehlswortes gewählte Bedingungspeicher
wird in Nein-Stellung gebracht.
(i = 1,2...6)

Der Bedingungspeicher B6 wird von der Maschine
automatisch in Ja-Stellung gebracht, wenn
 $\langle X_{12} \rangle \neq \langle X_{11} \rangle$ ist, z.B. bei Bereichsüber-
schreitung in X.

3.7. Bedingte Befehle

Bedingte Befehle werden nur ausgeführt, wenn die in
Klammern stehende Bedingung erfüllt ist. Bei nicht er-
füllter Bedingung wird der Befehl überlaufen.

3.7.1. Bedingungen bezüglich $\langle ZR1 \rangle$ und $\langle ZR2 \rangle$

(10)... Befehlsausführung, wenn $\langle ZR1 \rangle = 0$
(1Z)... Befehlsausführung, wenn $\langle ZR1 \rangle \neq 0$
(20)... Befehlsausführung, wenn $\langle ZR2 \rangle = 0$
(2Z)... Befehlsausführung, wenn $\langle ZR2 \rangle \neq 0$

Die Bedingung (10) steht in D11 (Kennzeichenstelle). Ein so gekennzeichneteter Befehl stellt einen von vornherein bedingten Normalbefehl dar. Es können noch zwei weitere Bedingungen in D10 und D9 gegeben werden.

3.7.2. Bedingungen bezüglich der Stellung der

Bedingungsspeicher B1...B6

- (J1)... Befehlsausführung, wenn B1 in Ja-Stellung
- (J2)... Befehlsausführung, wenn B2 in Ja-Stellung
- (J3)... Befehlsausführung, wenn B3 in Ja-Stellung
- (N4)... Befehlsausführung, wenn B4 in Nein-Stellung
- (N5)... Befehlsausführung, wenn B5 in Nein-Stellung
- (N6)... Befehlsausführung, wenn B6 in Nein-Stellung

3.7.3. Bedingungen bezüglich <X> und <Y>

- (NE)... Befehlsausführung, wenn $\langle X \rangle < 0$
 - (FO)... Befehlsausführung, wenn $\langle X \rangle \geq 0$
 - (NU)... Befehlsausführung, wenn $\langle X \rangle = 0$
 - (Y5)... Befehlsausführung, wenn $\langle Y_1 \rangle \geq 5$
 - (Y0)... Befehlsausführung, wenn $\langle Y_1 \rangle = 0$
 - (YZ)... Befehlsausführung, wenn $\langle Y_1 \rangle \neq 0$
 - (SU)... Befehlsausführung, wenn $\langle X_{12} \rangle = 0$ und $\langle Y_{12} \rangle = 9$ oder $\langle X_{12} \rangle = 9$ und $\langle Y_{12} \rangle = 0$
- (Y0) steht in D11, Erklärung siehe Bedingung (10)

3.7.4. Bedingungen aufgrund der letzten Vergleichsoperationen

- (NG)... Befehlsausführung, wenn Indikator NG gesetzt
- (GR)... Befehlsausführung, wenn Indikator GR gesetzt
- (KL)... Befehlsausführung, wenn Indikator KI gesetzt

3.7.5. Wiederholungsbefehl

- W Der Befehl wird solange ausgeführt, bis eine Bedingung, die bei diesem Befehl immer vorhanden sein muß (Ausnahme siehe 3.2.7.2.), nicht mehr erfüllt ist.

Die Operation muß also die Bedingung, welche mit W in der Klammer steht, ändern.

Z.B. (W.N6)E

Es wird solange linksverschoben, bis $\langle X_{12} \rangle$

$\neq \langle X_{11} \rangle$ ist. (Voraussetzung: $\langle X \rangle \neq 0$)

3.8. Programmstop

Z0 bewirkt Stop der Maschine vor Befehlsausführung.

Auf dem Bedienungspult wird durch die Befehlsregisteranzeige der Befehl mit Z0 in Form von Ziffern angezeigt. Zusätzlich leuchtet noch die Anzeige "Stop vor Befehlsausführung" auf dem Bedienungspult auf.

4. Befehlskombinationen

4.1. Allgemeines

Anhand des Blockschaltbildes können die Wege der Informationen für einfache Befehlskombinationen unter Zuhilfenahme der bereits erfolgten Beschreibung der einzelnen Befehle prinzipiell verfolgt werden. Eine genaue Auskunft über die einzelnen Wege der Informationen bei der entsprechenden Befehlskombination erhält man nur durch intensives Studium der ausführlichen logischen Pläne der ZUSE Z 31.

Deshalb werden hier die Möglichkeiten von Kombinationen der einzelnen Internbefehle der ZUSE Z 31 anhand von Tabellen erläutert.

4.2. Hinweise zur Verwendung der Kombinationstabellen

Bei der Verwendung der Tabellen sind folgende Punkte zu beachten:

In den Tabellen sind in Anlehnung an die Tabelle der Symbole des Interncodes (vgl. 1.) der ZUSE Z 31 die Befehle spaltenweise angeordnet.

Die möglichen Kombinationen gehen wie folgt daraus hervor:

- a) kombinierbar sind nur Befehle aus verschiedenen Spalten,
- b) die gestrichelten senkrechten Linien lassen Kombinationen zu,
- c) die durchgezogenen waagerechten Linien schließen Kombinationen aus.

Die einzelnen Befehle innerhalb einer Befehlskombination werden im allgemeinen unabhängig voneinander ausgeführt.

Bei einigen nachstehend aufgeführten Befehlen besteht jedoch eine Abhängigkeit der einzelnen Schritte.

1. Das Stopzeichen Z0 hat Vorrang vor allen anderen Befehlen, z.B.:

Z0.AVYGN Z0 stoppt die Maschine; die Befehlskombination AVYGN kommt nicht zur Ausführung, bleibt jedoch in b stehen.

2. Adressensubstitutions- oder -modifikationsbefehle, die in der Kombination vorhanden sind, werden vor allen anderen Befehlen zuerst ausgeführt. Dabei hat der G-Befehl den Vorrang vor den Xi+-bzw. Xi=-Befehl, z.B.:

AVYXi=Gn 1. Wortzeit: Ausführung G-Schritt
 2. Wortzeit: Ausführung Xi=-Schritt
 3. Wortzeit: Unabhängige Ausführung von A und VY.

3. Verschiebungen haben Vorrang vor allen Operationen, die im Operationswerk durchgeführt werden, z. B.:

ARV1.n Der verschobene Inhalt des X-Registers wird zur Addition verwendet; unabhängig davon wird <ZR1> um 1 erhöht.

4. Der Befehl I1 hat Vorrang vor den Befehlen V1 bzw. Z1.

z.B.:

AV1.I1.n

Die letzte Stelle der Adresse des in b stehenden Befehles wird zuerst nach ZR1 gebracht, erst dann wird <ZR1> um 1 erhöht. Völlig unabhängig von diesem Ablauf erfolgt die Addition mit dem X-Register.

Ferner ist zu beachten:

Jeder Befehl kann wiederholt gegeben werden (Zeichen(W) in D11), sofern kein anderes Kennzeichen in der Stelle D11 verwendet wird. Bei einem Wiederholungsbefehl muß aber immer mindestens eine Bedingung aus D10 oder D9 zum Befehl dazugesetzt werden. (Vgl. 3.7.5., Ausnahme: 3.2.7.1.)

Jeder Befehl kann unter je eine Bedingung aus D11, D10 und D9 gestellt werden. Der Befehl wird nur ausgeführt, wenn diese Bedingungen erfüllt sind. (Vgl. 3.7.)

Befehlskombinationen mit Adressenmodifikation dürfen nicht mit den Bedingungen aus D10 kombiniert werden. (Vgl. 3.3.4.3.)

4.3. Kombinationsübersicht I

1	2	3	4
A			
S	R		
AT	L		
ST	RR	V1	V2
U	LL	Z1	Z2
K	ZO	Ji	I1
O		Ni	I2
T			VY
B	ZO	F*	ZY
K			
O	LD		
T (b.Text)			
BM			
UM			
TM			
AT			V2
ST			Z2
T	ZO	VX	I1
K		ZX	I2
AY			
SY			
ATY		V1	
STY	R**	Z1	
UY	L**	Ji	
KY	ZO	Ni	
OY			
TY		F*	
KY			
OY	LD**	VX	
TY (b.Text)		ZX	
BY	ZO		
	R		V2
	L		Z2
	ZO	Q	I1
			I2
	R	V1	
	L	Z1	
	RR	Ji	
	LL	Ni	
	LD		
	ZO	F	
P		VX	
	ZO	ZX	
	R		
	L	Q	
	ZO		
E	ZO	F	

* F gilt jedoch nicht in Verbindung mit B.. oder T., da BFn und TFn Sonderbefehle ergeben.

** Die Verschiebungen beziehen sich auf das Y-Register.

4.4. Kombinationsübersicht II
(Befehle mit Zusatz C)

1	2	3	4
AC	R		
SC	L		
UC	RR		
KC	LL		
OC	ZO		V2
KC			Z2
OC	LD		I1
ATC			I2
STC	ZO		VY
BC			ZY
TC			
BMC			
UMC			
TMC			
ACY			
SCY	R*		
UCY	L*		
KCY	ZO		
OCY			
ATCY			
STCY			
BCY	ZO		
TCY			
BMCY			
UMCY			
TMCY			
KCY	LD*		
OCY			

* Die Verschiebungen beziehen sich auf das Y-Register.

4.5. Kombinationsübersicht III (Sonderbefehle)

1	2	3	4
B6	ZO	V1 Z1	V2 Z2
BM6		Ji Ni	I1 I2
BY4	ZO	V1 Z1	
BMY4		Ji Ni	
AY SY UY KY OY	MVX MZX		
BY *** UY			
A S U K O	MVX MZX		V2 Z2 I1 I2
B *** U			
B10 T10	ZO	V1 Z1	V2 Z2 I1 I2
TM10			VY ZY
BY10 TY10	ZO	V1 Z1 VX	
TMY10		ZX	
BF TF	ZO		VY ZY
TQY	ZO		
AP SP UP KP OP BP	R L RR LL ZO ZO	V1 Z1 Ji Ni	
UMP BMP		F *	
KP OP	LD		
KP	ZO	VX ZX	

* In Verbindung mit dem Befehl BPn wirkt das F allein auf den P-Befehl und nicht auf den B- Befehl.

*** In Verbindung mit MVX bzw. MZX wirkt M auch auf B und U, bzw. BY und UY.