

**E**inheitlicher  
**Z**usammenstand  
**E**lektronischen  
**R**echengeräte

Urbeschreibung

---

Daniel CAMPOS DO NASCIMENTO © 2020

Licensed under Creative Commons CC-BY-SA 4.0.

Reichstandortsgemeinschaft – Rsg

Reichsforschungsgemeinschaft für Rechenwissenschaft und -lehre – Rfg-r

British Forsee Fellowship for Recon- and Telllore

Norræn Rannsóknar Samfélag

© 1963-1970, 1972-1992

Unter der Reichsgemeinnutzerlaubnis erlaubt.

© 1993-

Unter der Reichsgemeinnutz- und -verbreitungserlaubnis erlaubt.

# Inhaltsverzeichnis

<b>I</b>	<b>Zugriff</b>	<b>3</b>
<b>1</b>	<b>Hauptspeicher</b>	<b>5</b>
	Stellenketten . . . . .	5
	Achtheit . . . . .	5
	Sechzehnheit . . . . .	5
	Zweiunddreißigkeit . . . . .	5
	Vierundsechzigheit . . . . .	5
	Hundertachtundzwanzigkeit . . . . .	5
	Zweihundertsechsfünzigheit . . . . .	6
	Anschriften . . . . .	6
	Wahranschriften . . . . .	6
	Kunstanschriften . . . . .	6
	1-Gestalt . . . . .	6
	2-Gestalt . . . . .	6
<b>2</b>	<b>Einheiten</b>	<b>7</b>
	Wahranschriften . . . . .	7
	Übertragung . . . . .	7
	Lesen . . . . .	7
	Schreiben . . . . .	7
	Zugriffolge . . . . .	7
	Kunstanschriften . . . . .	7
	Seitenfeld . . . . .	7
	Grundfeld . . . . .	7
	2-Seitetor . . . . .	8
	1-Seitefeld . . . . .	8
	1-Seitetor . . . . .	9
	Ablauf . . . . .	9
	Seitenfeldlauf . . . . .	9
	n-Seitezugriff . . . . .	9
<b>II</b>	<b>Unterbrechung</b>	<b>11</b>
<b>3</b>	<b>Anbauen</b>	<b>13</b>
	Ziel . . . . .	13
	Tor . . . . .	13
	Eintrag . . . . .	13
	Quelle . . . . .	13
<b>4</b>	<b>Ablauf</b>	<b>15</b>
<b>III</b>	<b>Ausführung</b>	<b>17</b>
<b>5</b>	<b>Betrieb</b>	<b>19</b>
	Anweisungskreislauf . . . . .	19
	Sonderfälle . . . . .	19

<b>6 Recheneinheiten</b>	<b>21</b>
Nahspeicher	21
0-15 – Angaben	21
AZ – Anweisungszeiger	21
ZE – Zustandseintrag	22
Angaben	22
Standbeistrichzahlen	22
U/Z1	22
U/Z2	22
U/Z4	22
Gleitbeistrichzahlen	22
X4	22
X8	23
X16	23
X32	23
Anweisungen	23
Regelungsanweisungen	23
WGL – Weglass	24
SPR – Sprung	24
Bewegungsanweisungen	24
BWG – Bewegung	24
UWG – Ungeschnittende Bewegung	25
VWE – Verfahrenwechsel	25
UTB – Unterbrechung	26
Stellenweiseanweisungen	26
LVS – Linksverschiebung	27
RNS – Rechtsnullverschiebung	27
RZS – Rechtsvorzeichenverschiebung	27
DRH – Umdreh	27
UND – Durchschnitt	27
ODR – Vereinigung	27
NIC – Nicht	28
VZW – Vorzeichenwechsel	28
Rechenanweisungen	28
ZGB – Zugabe	29
UNT – Unterschied	29
VFL – Verfaltung	29
VTL – Verteilung	29
<b>7 Durchgangeinheiten</b>	<b>31</b>
Nahspeicher	31
AZ – Anweisungszeiger	31
UTZ – Unterbrechungstorzeiger	31
UEZ – Unterbrechungseintragzeiger	31
ZE – Zustandseintrag	31
Anweisungen	32
Regelungsanweisung	32
Übertragungsanweisung	32
Lesung	33
Schreibung	33
Betrieb	33
Verfahren	33
Fallen	33

# Einführung

Der EZER ist ein Entwurf Zusammenstände, die als Ziel die Rechnung und Austausch der Angaben vereinfachen hat. Ein EZER Zusammenstand besteht aus einem **Hauptspeicher**, einigen **Recheneinheiten** und einigen **Durchgangeinheiten**.

Im EZER werden die Rechnungen und Austausche durch **Verfahren** ausgedrückt. Wenn ein Verfahren ausgeführt wird, finden drei Verläufe statt:

- der Zugriff;
- die Unterbrechung; und
- die Ausführung.

## Zugriff

Beim Zugriff wird eine **Angabe** zwischen einer Einheit und dem Hauptspeicher übertragen.

## Unterbrechung

Bei der Unterbrechung macht eine Einheit eine Recheneinheit ein neues Verfahren ausführen.

## Ausführung

Bei der Ausführung führt eine Einheit ein Verfahren.



**Teil I**

**Zugriff**





# Abschnitt 1

## Hauptspeicher

Der Hauptspeicher besteht aus **Orte**.

### Stellenketten

Die Orte speichern **Stellenketten**. In einer Stellenkette werden die Stellen ab 1 gezählt und gleichen sie entweder 0 oder 1. Eine Kette, die aus  $n$  Stellen besteht, wird eine  **$n$ -heit** genannt.

Jeder Ort im Hauptspeicher speichert eine einzige Achtheit. Um eine Stellenkette zu speichern, die größere als eine Achtheit ist, wird sie in Achtheiten geteilt und die Achtheiten in folgenden Orten gespeichert.

#### Achtheit

1	8
1	

#### Sechzehnheit

1	8 9	16
1	2	

#### Zweiunddreißigkeit

1	8 9	16 17	24 25	32
1	2	3	4	

#### Vierundsechzigheit

1	8 9	16 17	24 25	32
1	2	3	4	
33	40 41	48 49	56 57	64
5	6	7	8	

#### Hundertachtundzwanzigkeit

1	8 9		16 17		24 25		32
1	2		3		4		
33	40 41		48 49		56 57		64
5	6		7		8		
65	72 73		80 81		88 89		96
9	10		11		12		
97	104 105		112 113		120 121		128
13	14		15		16		

## Zweihundertsechsfünzigheit

1	8 9	16 17	24 25	32
1	2	3	4	
33	40 41	48 49	56 57	64
5	6	7	8	
65	72 73	80 81	88 89	96
9	10	11	12	
97	104 105	112 113	120 121	128
13	14	15	16	
129	136 137	144 145	152 153	160
17	18	19	20	
161	168 169	176 177	184 185	192
21	22	23	24	
193	200 201	208 209	216 217	224
25	26	27	28	
225	232 233	240 241	248 249	256
29	30	31	32	

## Anschriften

Eine **Anschrift** ist eine ganze Zahl, die einen Ort bestimmt. Eine Stellenkette, die größere als eine Achtheit ist, wird durch die Anschrift ihrer ersten Achtheit bestimmt.

Jede Anschrift ist eine **Wahranschrift** oder eine **Kunstanschrift**.

## Wahranschriften

Eine Wahranschrift bestimmt einen einzigen Ort im Hauptspeicher. Die Wahranschrift 0 bestimmt den ersten Ort, und folgende Wahranschriften bestimmen folgenden Orte.

## Kunstanschriften

Eine Kunstanschrift hat zwei Gestalten.

### 1-Gestalt

1	10 11	20 21	32
G	1-S	A	

G Grundbeschreibungszahl

1-S 1-Seiteschlüsselszahl

A Abstand

### 2-Gestalt

1	10 11	32
G	A	

G Grundbeschreibungszahl

A Abstand

## Abschnitt 2

# Einheiten

Eine Einheit greift den Hauptspeicher zu, wenn eine Stellenkette zwischen dieser Einheit und bestimmten Orte darin übertragen wird.

1. Die Einheit ergibt die Wahrschrift, die den ersten zugehörigen Ort im Hauptspeicher bestimmt.
2. Die Stellenkette werden zwischen der Einheit und dem Hauptspeicher übertragen.

## Wahrschriften

Wenn eine Stellenkette bestimmt wird, stillschweigend werden allen ihre Achtheite bestimmt.

## Übertragung

Je nach der Richtung der Übertragung entweder **liest** oder **schreibt** die Einheit beim Zugriff.

### Lesen

Wenn eine Einheit aus dem Hauptspeicher liest, wird die Stellenkette aus dem bestimmten Ort im Hauptspeicher zur Einheit geschickt.

### Schreiben

Wenn eine Einheit in den Hauptspeicher schreibt, wird die Stellenkette von der Einheit zum bestimmten Ort im Hauptspeicher geschickt.

## Zugriffsfolge

Für jede Einheit und für jeden Hauptspeicherort, eine Lesung von der Einheit aus dem Hauptspeicherort überträgt die Wert, die die letzte Schreibung von derselben Einheit zu demselben Hauptspeicherort überträgt hat.

## Kunstanschriften

Eine Kunstanschrift wird zu einer Wahrschrift verwandelt.

## Seitenfeld

Das **Seitenfeld** speichert die Angaben, die für die Verwandlung nützlich sind.

## Grundfeld

Ein Grundfeld enthält entweder 2-Seitebeschreibungen, 2-Seiteschlüssel oder 1-Seitenfeldbeschreibungen.







## **Teil II**

# **Unterbrechung**





## Abschnitt 3

# Anbauen

In einer Unterbrechung wird ein Eintrag von einer Quelle zu einem Ziel übertragen.

### Ziel

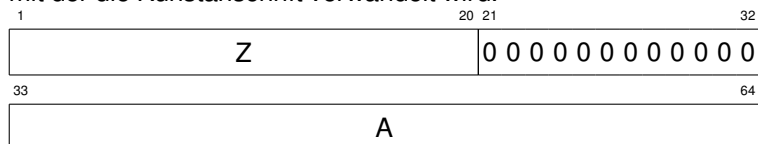
Ein Ziel ist eine Recheneinheit.

### Tor

Ein Unterbrechungstor ist ein Hauptspeicherort, das mit einem Ziel verbunden wird.

### Eintrag

Ein Unterbrechungseintrag enthält eine Kunstanschrift nach einer Angabe und einen Zeiger nach der Grundfeld, mit der die Kunstanschrift verwandelt wird.



Z Grundfeldzeiger

A Angabekunstanschrift

### Quelle

Die Quelle ist die Einheit, die die Unterbrechung an fängt.



## Abschnitt 4

### Ablauf

Eine Unterbrechung statt findet, wenn die Quelle einen Unterbrechungseintrag in das Unterbrechungstor des Ziels schreibt.

1. Die Quelle schreibt den Eintrag in das Tor des Ziels.
2. Das Ziel übersetzt die Eintragkunstschrift mit dem Grundfeld, den der Eintraggrundfeldzeiger zeigt.
3. Das Ziel verarbeitet die Angabe, die die gerade ausgegebene Wahrschrift zeigt.

Das Ziel muss die Anweisungskreislauf anfangen, nachdem es die Angabe verarbeitet hat.



## **Teil III**

# **Ausführung**



## Abschnitt 5

# Betrieb

Im EZER folgen die Recheneinheiten einen genauen Betrieb.

### Anweisungskreislauf

Eine Recheneinheit folgt der Anweisungskreislauf.

1. Die Einheit liest eine Anweisung.
2. Wenn die Anweisung nicht erkannt ist, findet der Sonderfall **AEA** statt.
3. Die Einheit macht alle Zugriffe, die vor der Anweisung gemacht werden sollen.
4. Die Einheit führt die Anweisung aus.
5. Die Einheit macht alle Zugriffe, die nach der Anweisung gemacht werden sollen.

### Sonderfälle

Wenn ein Sonderfall statt findet, führt die Einheit eine **Falle** aus.

Eine Falle ist ein sonderes Verfahren, das den Zustand der Einheit beim Sonderfall als Eingabe bekommt.





## Abschnitt 6

# Recheneinheiten

Im EZER werden die Recheneinheiten nach einen bestimmten Entwurf gebaut.

### Nahspeicher

Der Nahspeicher hängt nicht vom Hauptspeicher ab.

### 0-15 – Angaben

1		32
	0	
	1	
	2	
	3	
	4	
	5	
	6	
	7	
	8	
	9	
	10	
	11	
	12	
	13	
	14	
	15	

### AZ – Anweisungszeiger

1		31	32
	A		Z

A Zeiger

Z Zustand

0 Laufend

1 Stehend

## ZE – Zustandseintrag

1		20	21	29	30	31	32
	G	0	0	0	0	0	0
33				B	0		
	AZ						

G Grundfeldzeiger

B Bedingung

AZ Anweisungszeiger

## Angaben

Eine Angabe kann entweder eine Standbeistrichzahl oder eine Gleitbeistrichzahl sein.

### Standbeistrichzahlen

Sei  $Z_U(S, M)$  die Zahl, die eine Stellenkette  $S$  mit wenigsten Macht  $M$  in einer unvorzeichnenden Deutung aus drückt.

$$Z_U(S, M) = 2^M \sum_{i=1}^{|S|} 2^{|S|-i} S_i$$

Sei  $Z_Z(S, M)$  die Zahl, die eine Stellenkette  $S$  mit wenigsten Macht  $M$  in einer vorzeichnenden Deutung aus drückt.

$$Z_Z(S, M) = 2^M \left( -2^{|S|-1} S_1 + \sum_{i=2}^{|S|} 2^{|S|-i} S_i \right)$$

### U/Z1

1	8
S	

### U/Z2

1	16
S	

### U/Z4

1	32
S	

### Gleitbeistrichzahlen

Sei  $X(V, M, B)$  die Zahl, die eine Vorzeichenstelle  $V$ , eine Machtstellenkette  $M$  und eine Bruchstellenkette  $B$  aus drücken.

$$X(V, M, B) = (-1)^V 2^{Z_U(M, 0) - 2^{|M|-1}} Z_U(B, -1 - |B|)$$

### X4

1	2	9	10	32
V	M	B		

### X8

1	2			12	13					32
V			M					B		
33										64
								B		

### X16

1	2					16	17			32
V				M				B		
33										64
								B		
65										96
								B		
97										128
								B		

### X32

1	2							20	21			32
V					M					B		
33												64
										B		
65												96
										B		
97												128
										B		
129												160
										B		
161												192
										B		
193												224
										B		
225												256
										B		

## Anweisungen

Eine Anweisung beschreibt eine Änderung des Zustands der Einheit. Jede Anweisung hat einige **Gestalten**.

### Regelungsanweisungen

Eine Regelungsanweisung ändert die Ausführung des Verfahrens, wenn ihr Bedingung befriedigt wird. Jede Stelle der Anweisungsbedingung bestimmt einen möglichen Wert der ZE-Bedingung, und die Anweisungsbedingung wird befriedigt, wenn die ZE-Bedingung einen Wert hat, dessen Anweisungsbedingungsstelle 1 gleicht.

Eine Regelungsanweisung hat drei Gestalten.

### Beständergestalt

1	3	4	5		8	9		12	13			32
0	0	0	*		B		R					W

In dieser Gestalt:

- ist das Ziel das Nahspeicherort  $R$ ;
- ist die Quelle  $W$ .

### Hauptspeichergestalt

1	3	4	5	8	9	12	13	16	17	32
0	1	0	*	B	R	Q	A			

In dieser Gestalt:

- ist das Ziel das Nahspeicherort  $R$ ;
- ist die Quelle das Hauptspeicherort, das die Zugabe  $A$  und des Nahspeicherortes  $Q$  bestimmt.

### Nahspeichergestalt

1	3	4	5	8	9	12	13	16
1	1	0	*	B	R	Q		

In dieser Gestalt sind das Ziel und die Quelle die Nahspeicherorte  $R$  und  $Q$ .

### WGL – Weglass

Nach dieser Anweisung:

- gleicht  $AZ$  der Zugabe  $AZ$  und der Quelle;
- gleicht das Ziel  $AZ$  vor der Ausführung.

1	3	4	5																																	32																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
*	*	*	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*</

### SPR – Sprung

Nach dieser Anweisung:

- gleicht  $AZ$  die Quelle;
- gleicht das Ziel  $AZ$  vor der Ausführung.

1	3	4	5																																	32																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
*	*	*	1	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*</

## Bewegungsanweisungen

Eine Bewegungsanweisung beschreibt eine Bewegung Angabe zwischen dem Hauptspeicher und dem Nahspeicher.

### BWG – Bewegung

Nach dieser Anweisung wird das Ziel durch die Quelle ersetzt.

Diese Anweisung hat vier Gestalten.

### Beständergestalt

1	8	9	12	13	32
00110000			Z	W	

In dieser Gestalt:

- ist das Ziel das Nahspeicherort  $Z$ ;
- ist die Quelle  $W$ .

### Hauptspeichergestalt

1	8	9	12	13	16	17	18	20	21	32		
0	1	1	1	0	0	0	0	Z	Q	Ä	G	A

In dieser Gestalt:

- ist das Ziel das Nahspeicherort Z;
  - wenn die Ä-Stelle 0 gleicht, ist die Quelle das Hauptspeicherort, das die Zugabe A und des Nahspeicherortes Q bestimmt;
- sonst:
- ist die Quelle das Hauptspeicherort, den das Nahspeicherort Q bestimmt;
  - wird A zum Nahspeicherort Q nach der Anweisung zugegeben.

1	8	9	12	13	16	17	18	20	21	32															
1		0		1		1		0		0		0		0		Z		Q		Ä		G		A	

In dieser Gestalt:

- wenn die Ä-Stelle 0 gleicht, ist das Ziel das Hauptspeicherort, das die Zugabe A und des Nahspeicherortes Z bestimmt;
- sonst:
- wird A zum Nahspeicherort Z bevor der Anweisung zugegeben;
  - ist das Ziel das Hauptspeicherort, das der Nahspeicherort Z bestimmt;
- ist die Quelle das Nahspeicherort Q.

### Nahspeichergestalt

1	8	9	12	13	16				
1	1	1	1	0	0	0	0	Z	Q

In dieser Gestalt sind das Ziel und die Quelle die Nahspeicherorte Z und Q.

### UWG – Ungeschnittende Bewegung

Der Ablauf dieser Anweisung hängt von der Gestalt ab.

Diese Anweisung hat zwei Gestalten.

1	8	9	12	13	16	17	18	20	21	32		
0	1	1	1	0	0	0	1	Z	Q	0	G	A

In dieser Gestalt:

- ist das Ziel das Nahspeicherort Z;
- ist die Quelle das Hauptspeicherort, das die Zugabe A und des Nahspeicherortes Q bestimmt.

Außerdem wird der genaue Hauptspeicherort gespeichert.

1	8	9	12	13	16	17	18	20	21	32		
1	0	1	1	0	0	0	1	Z	Q	0	G	A

In dieser Gestalt:

- ist das Ziel das Hauptspeicherort, das die Zugabe A und des Nahspeicherortes Z bestimmt;
- ist die Quelle das Nahspeicherort Q.

Außerdem erschafft diese Anweisung nicht, wenn eine Schreibung zu dem genauen Hauptspeicherort nach der letzten **UWG** danach statt gefunden ist.

### VWE – Verfahrenwechsel

Der Ablauf dieser Anweisung hängt von der Gestalt ab.

Diese Anweisung hat zwei Gestalten.

1	8	9	12	13	16	17	32
0	1	1	1	0	0	1	0
			R		Q		A

In dieser Gestalt:

- wird  $ZE$  in das Hauptspeicherort gespeichert, das das Nahspeicherort  $R$  bestimmt;
- wird einen neuen Zustandseintrag aus dem Hauptspeicherort geladen, das die Zugabe  $A$  und des Nahspeicherortes  $Q$  bestimmt.

1	8	9	12	13	16
1	1	1	1	0	0
			R		Q

In dieser Gestalt:

- wird  $ZE$  in das Hauptspeicherort gespeichert, das das Nahspeicherort  $R$  bestimmt;
- wird einen neuen Zustandseintrag aus dem Hauptspeicherort geladen, das das Nahspeicherort  $Q$  bestimmt.

### UTB – Unterbrechung

1	8	9	12	13	15	16	17	32
1	0	1	1	0	0	1	0	
			Z		Q	0		A

In dieser Anweisung:

- ist die Quelle die Paar Nahspeicherorte  $(Q, Q + 1)$ ;
- ist das Ziel das Hauptspeicherort, das die Zugabe  $A$  und des Nahspeicherortes  $Z$  bestimmt.

### Stellenweiseanweisungen

Eine Stellenweiseanweisung beschreibt eine Rechnung über einzelne Stellen.

Jede Stellenweiseanweisung hat vier Gestalten.

### Beständergestalt

1	4	5	8	9	12	13	32
0	0	1	0	*	*	*	*
			Z				W

In dieser Gestalt:

- ist das Ziel das Nahspeicherort  $Z$ ;
- ist die Quelle  $W$ .

### Hauptspeichergestalt

1	4	5	8	9	12	13	16	17	18	20	21	32
0	1	1	0	*	*	*	*					
			Z		Q		Ä		G			A

In dieser Gestalt:

- ist das Ziel das Nahspeicherort  $Z$ ;
- wenn die Ä-Stelle 0 gleicht, ist die Quelle das Hauptspeicherort, das die Zugabe  $A$  und des Wertes  $Q$  bestimmt;

sonst:

- ist die Quelle das Hauptspeicherort, den das Nahspeicherort  $Q$  bestimmt;
- wird  $A$  zum Nahspeicherort  $Q$  nach der Anweisung zugegeben.

1	4	5	8	9	12	13	16	17	18	20	21	32
1	0	1	0	*	*	*	*					
			Z		Q		Ä		G			A

In dieser Gestalt:

- wenn die Ä-Stelle 0 gleicht, ist das Ziel das Hauptspeicherort, das die Zugabe  $A$  und der Wert  $Z$  bestimmt;
- sonst:
  - wird  $A$  zum Nahspeicherort  $Z$  bevor der Anweisung zugegeben;
  - ist das Ziel das Hauptspeicherort, das das Nahspeicherort  $Z$  bestimmt;



1	4	5	8	9	32
* * *	0	1	0	1	* *
1	4	5	8	9	16
* * *	0	1	0	1	* * * * * * * *

### NIC – Nicht

Nach dieser Anweisung wird jede Stelle des Ziels durch die Wechsel der entsprechenden Stellen der Quelle ersetzt.

1	4	5	8	9	32
* * *	0	1	1	0	* *
1	4	5	8	9	16
* * *	0	1	1	0	* * * * * * * *

### VZW – Vorzeichenwechsel

Nach dieser Anweisung wird das Ziel durch die Wechsel der Quelle ersetzt.

In der Beständergestalt wird die Quelle nullweitert.

1	4	5	8	9	32
* * *	0	1	1	1	* *
1	4	5	8	9	16
* * *	0	1	1	1	* * * * * * * *

## Rechenanweisungen

Eine Rechenanweisung beschreibt eine Rechnung.

Alle Rechenanweisungen rechnen Standbeistrichzahlen.

Jede Rechenanweisung hat vier Gestalten.

### Beständergestalt

1	4	5	8	9	12	13	32
0	0	1	0	* * *	Z		W

In dieser Gestalt:

- ist das Ziel das Nahspeicherort *Z*;
- ist die Quelle *W*.

### Hauptspeichergestalt

1	4	5	8	9	12	13	16	17	18	20	21	32
0	1	1	0	* * *	Z		Q	Ä	G			A

In dieser Gestalt:

- ist das Ziel das Nahspeicherort *Z*;
  - wenn die Ä-Stelle 0 gleicht, ist die Quelle das Hauptspeicherort, das die Zugabe *A* und des Nahspeicherortes *Q* bestimmt;
- sonst:
- ist die Quelle das Hauptspeicherort, das das Nahspeicherort *Q* bestimmt;
  - wird *A* zum Nahspeicherort *Q* nach der Anweisung zugegeben.

1	4	5	8	9	12	13	16	17	18	20	21	32
1	0	1	0	* * *	Z		Q	Ä	G			A

In dieser Gestalt:



1	4	5	8	9	12	13	16
1	1	1	0	*	*	*	*
				Z	Q		

The diagram shows two rows of boxes representing bits. The top row is labeled with indices 1, 4, 5, 8, 9, and 32. It contains a sequence of 32 bits: the first four bits are 1, 0, 0, 0; bits 5 through 8 are 0; and bits 9 through 32 are 0. The bottom row is labeled with indices 1, 4, 5, 8, 9, and 16. It contains a sequence of 16 bits: the first four bits are 1, 0, 0, 0; bits 5 through 8 are 0; and bits 9 through 16 are 0.

The diagram shows an array of 32 elements. The first 16 elements are marked with asterisks (\*), and the last 16 elements are marked with asterisks (\*). The array is divided into two halves of size 16. The left half is sorted, and the two sorted halves are merged back together.

Diagram illustrating the recursive splitting of a 32-bit array into two 16-bit arrays. The top row shows a 32-bit array with indices 1, 4, 5, 8, 9, and 32. The bottom row shows a 16-bit array with indices 1, 4, 5, 8, 9, and 16. The array is split into two 16-bit halves, each containing a 10-bit segment and a 6-bit segment.

Diagram illustrating the iterative step of the merge sort algorithm. It shows two 16-element arrays being merged into a 32-element array. The top array is divided into two 16-element segments, each containing a sequence of asterisks followed by the binary sequence 1011. The bottom array is divided into two 8-element segments, each containing a sequence of asterisks followed by the binary sequence 1011. The resulting 32-element array at the top contains a sequence of asterisks followed by the binary sequence 1011101110111011.



## Abschnitt 7

# Durchgangeinheiten

Die Durchgangeinheiten vereinfachen die Übertragung Angabe zu und aus dem Hauptspeicher. Jede Durchgangeinheit besteht aus einem **Gerät** und einer **Ausführungseinheit**. Hier wird beschrieben den Entwurf der Ausführungseinheit.

## Nahspeicher

### AZ – Anweisungszeiger

1		30	31	32
	A		0	Z

A Zeiger

Z Zustand

0 Laufend

1 Stehend

### UTZ – Unterbrechungstorzeiger

1		32
	Z	

### UEZ – Unterbrechungseintragzeiger

1		32
	Z	

### ZE – Zustandseintrag

1		20	21	24	25	30	31	32
	G		0	0	0		S	B
33								64
	AZ							
65								96
	UTZ							
97								128
	UEZ							

G Grundfeldzeiger

S Sonderfallzeichen

B Bedingung  
AZ Anweisungszeiger  
UTZ Unterbrechungstorzeiger  
UEZ Unterbrechungseintragzeiger

L Lange

### Beständergestalt

1	2	3	4	5	10	11	32
*	*	1	1	0	0	0	0
							L
							32
							64
							A

In dieser Gestalt:

- ist *L* die Zahl Achtheiten zu übertragen;
- ist *A* die Anschrift des Orts, zu dem die Angaben übertragen werden.

### Lesung

Nach dieser Anweisung wird die bestimmte Zahl Achtheiten vom Hauptspeicher in das Gerät übertragen.

1	2	3	4	5	10	11	32
*	*	*	0	0	0	0	0
							A
							32
							64
							L
							32
							64
							A

### Schreibung

Nach dieser Anweisung wird die bestimmte Zahl Achtheiten vom Gerät in den Hauptspeicher übertragen.

1	2	3	4	5	10	11	32
*	*	*	0	0	0	0	0
							A
							32
							64
							L
							32
							64
							A

## Betrieb

### Verfahren

Die Durchgangeinheit fängt ein Verfahren an, wenn sie mit einem Zustandseintrag untergebrochen wird, in dem *Z* von *AZ* **Laufend** gleicht.

### Fallen

Die Durchgangeinheit schreibt *ZE* zum Hauptspeicherort, das *UTZ* + 4 folgt.

Dann führt die Durchgangeinheit ein Verfahren aus, das aus einer einzigen Regelungsanweisung mit der folgenden Felden besteht:

- *Z* gleich 1;
- *U* gleich 1;
- *A* gleich 0.