

HANDBUCH  
ZUR  
PROGRAMMIERUNG  
DER DIGITALRECHNER  
MINCAL 4E/4N

Teil 1: Aufbau und FunktionenBlatt

Wortaufbau	1-1
Speicheradressen	1-2
Register	1-3
Indexregister	1-4
W-Register	1-5
X-Register	1-6
Codezahlen	1-7
Vorrangsystem	1-8...1-9
Sensoren	1-10
E-Puffer	1-11
Schreiumaschine	1-12
Streifenleser/-locher	1-13...1-14

Teil 2: Symbolische Programmsprache

2-1...2-5

Teil 3: Programminstruktionen

Instruktionen (Zusammenstellung)	3-1...3-3
Instruktionen (Einzelbefehle)	3-4...3-38

Teil 4: Symbolische Definitionen

Definitionen (Zusammenstellung)	4-1
Definitionen (Einzelklärung)	4-2...4-6

Anhang: Tabelle BCD-Code  
 Potenzen von 2  
 16-bit-Hilfsblätter  
 Instruktionslisten

		MINCAL 4E/4N	HEINRICH DIETZ
		PROGRAMMIERUNG	INDUSTRIE-ELEKTRONIK

Die Wortlänge beim MINCAL 4E/4N beträgt

18 Datenbit (können zur bequemeren Formulierung in 6 Oktalstellen zusammengefaßt werden)

1 Vorzeichenbit (negativ, wenn 1)

1 Prüfbit (nur für Kernspeicher; wird automatisch erzeugt und geprüft)

Je nach Art der Information können die 19 Binärstellen (Vorzeichen + Datenbits) eines Wortes folgende Bedeutung haben:

1. Instruktion: -BBB BBB EEE AAA AAA AAA

<sup>\*)</sup>  
B = Befehl; E = Ergänzung; A = Adresse

2. Binärzahl: -H11 111 111 111 111 111 11N

H = höchste Stelle ( $2^{-1}$ ); N = niedrigste Stelle ( $2^{-18}$ )

3. Linkscomma-Dezimalzahl: -842 184 218 421 842 184

(5 Stellen)

4. Rechtscomma-Dezimalzahl: -218 421 842 184 218 421

(5 Stellen)

5. Oktalzahl: -421 421 421 421 421 421

(6 Stellen)

6. Alpha-Wert: -BA8 421 BA8 421 BA8 421

(s. hierzu Tabelle BCD-Code)

<sup>\*)</sup>Die erste Oktalstelle gibt die Befehlsgruppe an, die zweite definiert den einzelnen Befehl innerhalb einer Gruppe.

Die Kernspeicherkapazität beträgt bei MINCAL 4E/4N je nach Ausbauzustand minimal 4096 (4K) Worte, maximal 32768 (32K) Worte.

Die Adressen der einzelnen Speicherplätze werden oktal wie folgt nummeriert:

00000	Platz	0	1. Speichereinheit (Grundausrüstung)
00001	"	1	
....	....		
07776	"	4094	
07777	"	4095	
10000	Platz	4096	2. Speichereinheit
....	....		usw.
....	....		
77777	"	32767	8. Speichereinheit

Der Speicher ist in "Seiten" von jeweils 512 (oktal: 777) Wörtern aufgeteilt. Bei speicherbezogenen Instruktionen können nur die Plätze direkt adressiert werden, die zur gleichen Seite gehören; Zugang zu anderen Seiten wird durch indirekte Adressierung (oder auch durch Adressmodifikation mit Indexregister) geschaffen. Plätze in der Seite 0 sind jedoch von allen Seiten aus zugänglich; dies wird durch negatives Vorzeichen der Instruktion gekennzeichnet. Die bezogene Adresse wird in den 3 Adressstellen der Instruktion oktal angegeben. Die Seitenaufteilung ist wie folgt:

00000	Seite 0
....	
00777	
01000	Seite 1
....	
01777	
02000	Seite 2
....	usw.

Bei der Formulierung von Programmen in symbolischer Schreibweise braucht auf die Seitengrenzen keine Rücksicht genommen zu werden, da das Umwandlungsprogramm automatisch indirekte Adressierung einführt, wenn dies erforderlich ist.

Der Speicher kann in beliebiger Weise mit Instruktionen, Festwerten und Variablenplätzen belegt werden, ausgenommen sind die Plätze für die Register und die Codezahlen.

Die für die Programmierung wichtigen Register befinden sich beim MINCAL 4E/4N im Kernspeicher; hierfür sind die ersten Plätze reserviert. Die Registerplätze sind über ihre Adressen erreichbar, werden im Übrigen aber von der Operationslogik automatisch angewählt.

Jedes Niveau, d.h. jede Programmebene des Prioritätssystems, hat einen eigenen Satz von 8 Registern, die je nach dem gerade laufenden Niveau automatisch selektiert werden; die Sätze unterscheiden sich durch die oktale 1er-Adresse:

Adressen	00000 ... 00007	- Register zu Niveau 0
"	00010 ... 00017	- " " " 1
"	00020 ... 00027	- " " " 2 (nur MINCAL 4N)
"	00030 ... 00037	- " " " 3
"	00040 ... 00047	- " " " 4
"	00050 ... 00057	- " " " 5
"	00060 ... 00067	- " " " 6
"	00070 ... 00077	- " " " 7

Die 8 Register eines Niveaus werden durch die oktale 1er-Adresse gekennzeichnet:

1er-Adresse #	Symbol #	-	Startadresse
" 1 "	1	-	Indexregister 1
" 2 "	2	-	Indexregister 2
" 3 "	3	-	Indexregister 3
" 4 "	4	-	Indexregister 4
" 5 "	N	-	Programmstand (Instruktionszähler)
" 6 "	K	-	K-Register (Arbeitsregister, Akkumulator)
" 7 "	X	-	X-Register (Hilfsregister, Akku-Verlängerung)

Die Funktion der Register 0 und N ist unter "Vorrangsystem", die der übrigen auf den folgenden Blättern beschrieben.

Viele Operationen beziehen sich ausdrücklich auf einzelne Register, z.B. durch Abfragen oder Verändern ihres Inhalts, in Form einer Adreßmodifikation durch Indexregister oder durch Beeinflussen einer Verzweigung durch den Inhalt eines Registers. In diesen Fällen wird das gewünschte Register mit seiner 1er-Adresse in der Ergänzungsstelle der Instruktion angegeben. Zu beachten ist auch hier, daß stets die Register des jeweils laufenden Programm niveaus benutzt werden.

1-3		MINCAL 4E/4I PROGRAMMIERUNG	HEINRICH DIETZ INDUSTRIE-ELEKTRONIK
-----	--	--------------------------------	--

Für jedes Programmenniveau sind 4 Indexregister (IXR1...4) vorhanden.

Die Aufgabe der Indexregister besteht in folgendem:

- Bei speicherbezogenen Operationen (mit Ausnahme von Verzweigung und Unterprogramm-Sprung) sowie bei Ein/Ausgabe-Operationen modifizieren sie die Adresse, indem der Inhalt des betreffenden Indexregisters zur Adresse addiert bzw. bei negativem IXR-Inhalt von der Adresse subtrahiert und so eine neue Adresse gebildet wird, die dann im Verlauf der Operation benutzt wird.

Bei direkter Adressierung und bei Ein/Ausgabeoperationen wird die in der Instruktion programmierte Adresse modifiziert. Bei indirekter Adressierung wird die indirekte (effektive) Adresse modifiziert, bei mehrfach indirekter Adressierung die letzte indirekte (effektive) Adresse.

- Bei den mit Nullinhalt, positivem oder negativem Inhalt eines Registers bedingten Verzweigungsbefehlen kann es sich um Indexregister handeln.
- Beim Unterprogramm-Sprung wird die Rückkehradresse vorzugsweise in einem der Indexregister gespeichert.

In allen Fällen wird das Indexregister in der Ergänzungsstelle der Instruktion durch Einsetzen der Ziffern 1...4 angegeben. Im Falle (1) findet Adressmodifikation nur dann statt, wenn eine der Ziffern 1...4 angegeben ist.

Der Indexregister-Inhalt kann durch die Befehle LDC, ADC und LDR verändert werden; ferner durch den Register-Transfer TRR.

Beispiele:

LDC1 0.	101000	Setze IXR1 auf Null
LDC1 511	101777	" " " 511
LDC1-511	-101777	" " " -511
ADC1 1.	141001	Erhöhe IXR1 um 1
ADC1-1.	-141001	Erniedrige IXR1 um 1 (bei Subtraktion durch Null Vorzeichenwechsel und richtige Darstellung des Betrages)
LDR1	111000	Lade IXR1 mit Inhalt des W-Registers
TRRW 1.	156001	Transfer W-Inhalt nach IXR1

Jedes Programmneveau besitzt ein W-Register (Arbeitsspeicher, Akkumulator). Es speichert den jeweiligen Datenstand.

Der Inhalt des W-Registers wird verändert:

1. durch die Vorzeichen- und Schiftbefehle,
2. durch die Wertwertoperationen LDC und MLC, wenn das W-Register in der Ergänzung angegeben ist,
3. durch die Ladebefehle LD und LDY,
4. durch den Register-Transfer TRR, wenn in der 1er-Adresse eine "6" angegeben ist,
5. durch die arithmetischen, logischen und Umwandlungsoperationen (Befehlsgruppen 2 und 3),
6. durch die Übernahme- bzw. Eingabebefehle GB, IBG, GX und IXG.

Der Inhalt des W-Registers wird angefragt, bleibt jedoch unverändert:

7. beim Register-Ladebefehl LDR; beim Register-Transfer TRR, wenn das W-Register in der Ergänzung angegeben ist,
8. bei den Transferbefehlen TR und TRY,
9. bei den inhaltsbedingten Verzweigungen (wenn Null, Plus oder Minus) der Befehlsgruppe 4, wenn das W-Register in der Ergänzung angegeben ist,
10. bei den Ausgabebefehlen OB... und OX...

Bei allen übrigen Befehlen wird das W-Register weder verändert noch berücksichtigt.

Jedes Programm niveau besitzt ein X-Register (Hilfsregister). Seine wesentliche Aufgabe besteht darin, Nebenergebnisse aufzunehmen, wenn der Dateninhalt des W-Registers verändert wird und die Nebenergebnisse in diesem nicht Platz finden.

Der Inhalt des X-Registers wird verändert:

1. durch die Schiftbefehle SR, SL und SRR durch Aufnahme der aus dem X-Register herauslaufenden Binärstellen, sowie bei SLN, wo es die Anzahl der Schiebstellen enthält,
2. durch die Festwertoperationen LDC und AUC und den Register-Ladebefehl LDR, wenn das X-Register in der Ergänzung angegeben ist; durch den Register-Transfer TRR, wenn in der 1er-Adresse eine "7" enthalten ist,
3. durch einen Überlauf im W-Register bei Addition und Subtraktion (Befehle AD, ADY, SB, SBY) und bei Überlauf beim Befehl AUC durch Addieren einer rechtsbündigen 1 zum Betrag,
4. bei Multiplikation durch Aufnahme der unbedeutenden Produktstellen,
5. bei Division durch Aufnahme des Rests (bzw. eines negativen Vorzeichens, wenn die Division nicht durchführbar ist),
6. beim Unterprogramm-Sprung, wenn das X-Register zwecks Aufnahme der Rückkehradresse in der Ergänzung angegeben sein sollte,
7. bei Code-Umwandlung durch Verwendung als Zwischenspeicher.

Der Inhalt des X-Registers wird abgefragt, bleibt jedoch unverändert:

8. beim Register-Transfer TRR, wenn das X-Register in der Ergänzung angegeben ist,
9. bei den inhaltsbedingten Verzweigungen der Befehlsgruppe 4, wenn das X-Register in der Ergänzung angegeben ist.

Bei allen übrigen Befehlen wird das X-Register weder verändert noch berücksichtigt.

Für die Umwandlungsbefehle VBL, VBR (Dezimal-binär) und VDL, VDR (Binär-Dezimal) werden positive Festwerte (Codezahlen) benutzt, die von der Operationslogik automatisch abgefragt werden.

Sie befinden sich in folgenden Plätzen:

Adresse: 10102	Inhalt (oktal): 631464
103	050754
104	004062
105	000323
106	000025
00112	047040
113	017500
114	001440
115	000120
116	000010

Diese Plätze dürfen für andere Zwecke nicht verwendet werden und sind bei Programmierung im Maschinencode richtig zu belegen, wenn man die Umwandlungsbefehle benutzen will. Bei symbolischen Programmen ist letzteres nicht notwendig, da das Umwandlungsprogramm die Codezahlen automatisch dort einsetzt.

Das Vorrangsystem des Rechners MINCAL 4E umfaßt 2 Programmebenen (Niveau 0 und 1); der MINCAL 4N besitzt deren 8 (Niveaus 0 bis 7).

Das Prioritatsystem unterscheidet folgende Zustände:

- a) im "Ablauf-Zustand" sind alle die Niveaus, die einmal gestartet, aber noch nicht durch einen End-Befehl (oder manuell) ausgeschaltet worden sind,
- b) im "Start-Zustand" sind alle die Niveaus, die sich im "Ablauf" befinden und nicht durch einen Halt-Befehl (oder manuell) unterbrochen worden sind,
- c) wirksam ist nur das Niveau mit der höchsten Vorrangstufe (Niveau-Ziffer), das sich gerade im Start-Zustand befindet.

Nur wenn mindestens ein Niveau im Start-Zustand ist, läuft im Rechner ein Programm.

Der Ablaufzustand wird durch einen erstmaligen Niveau-Start hergestellt; der Inhalt des (nivea zugehörigen) Registers 0 wird als Startadresse in das Register N (Programmstand-, Instruktionszähler) übertragen, und das Programm wird an dieser Stelle begonnen. Ein End-Befehl (END, ECP) beendet den Ablaufzustand. Der nächste Niveau-Start läßt das Programm von vorne beginnen.

Mit dem Ablaufzustand wird zugleich der Startzustand hergestellt; er kann durch einen Halt-Befehl (HLT, HCP) im Programm ausgeschaltet werden, ohne daß der Ablaufzustand beendet wird. Ein neuer Niveau-Start stellt den Startzustand wieder her und setzt das Programm an der unterbrochenen Stelle, die durch den Inhalt des Registers N bestimmt wird, fort.

Niveau-Starts können erfolgen:

1. von außen über eine Startleitung je Niveau,
2. manuell über eine Start-Taste je Niveau,
3. durch Prioritätswechsel-Befehle (NCP, HCP, ECP) im Programm vom anderen Niveau aus,
4. durch einen Puffer, wenn dessen Arbeitszyklus beendet ist und der auslösende Befehl (IBP, OBP) den Niveau-Start vorsah.

Das Programm im jeweiligen Niveau wird am Ende der laufenden Operation unterbrochen, wenn ein Niveau mit höherem Rang gestartet wird. Wird dieses angehalten (Halt-Befehl) oder beendet (End-Befehl), so wird das Programm im ursprünglichen Niveau an der Stelle fortgesetzt, wo es unterbrochen wurde.

Ein Niveau-Start während des Laufs eines höheren Niveaus wird gespeichert und führt zum Ziel, sobald alle höheren Niveaus angehalten oder beendet sind.

Der Wechsel des wirksamen Niveaus erfolgt jeweils am Ende einer Operation; die nächste Operation des höheren Niveaus wird sofort begonnen. Die Programmierung braucht, abgesehen von ihrer Vergabe nach Prioritäts-Gesichtspunkten, auf die Programmebenen keine Rücksicht zu nehmen, da sie unabhängig voneinander sind. Alle Niveaus können jedoch gemeinsame Teile des gespeicherten Programms benutzen.

1-8	/	—	MINCAL 4 / 4N	HEINRICH DIETZ
			PROGRAMM	

Das jeweils wirksame Niveau hat folgenden Einfluß:

- I) es wählt den ihm eigenen Satz von Registern (Startadresse, 4 Indexregister, Programmstand N, Arbeitsregister W und Hilfsregister X) automatisch an,
- II) es schaltet die niveaugebundenen Sensoreingänge durch,
- III) beim MINCAL 4N wählt es bei Ein/Ausgabebefehlen der Gruppe 6 automatisch den niveauzugehörigen Puffer an; lediglich der Puffer Ø ist von allen Niveaus aus zu bedienen. (Der Niveau-Start aus dem Puffer geht stets auf das auslosende Niveau).

Der Puffer des MINCAL 4E wird von beiden Niveaus aus bedient; der Niveau-Start aus dem Puffer geht stets auf Niveau 1.

Sensoren dienen zur Beeinflussung des Programmverlaufs von außen mit Hilfe von Verzweigungen. Es stehen 7 gemeinsame Sensoreingänge und je Niveau 7 niveaugebundene zur Verfügung. Sensor 1 und 2 (bei MINCAL 4N) haben eine besondere, vorbestimmte Funktion (s. Hinweis).

Die Verzweigungsbefehle SR und BRY sind sensorabhängig programmierbar. Ist in der Ergänzung eine der Ziffern 1 bis 7 angegeben, so wird die Verzweigung nur dann ausgeführt, wenn der betreffende Sensor erregt ist; andernfalls geht das Programm mit der nächsten Instruktion weiter.

Ein Sensor ist erregt, wenn

1. der betreffende der 7 Sensor-Schalter betätigt ist, oder
2. der betreffende der 7 gemeinsamen Sensor-Eingänge positives Potential hat, oder
3. der betreffende der 7 niveaugebundenen Sensor-Eingänge, der zum gerade laufenden Niveau gehört, positives Potential hat.

Hinweis: Um den aus dem niveaugebundenen Puffer kommenden Niveau-Start von anderen im Programm unterscheiden zu können, wird zugleich mit dem Startsignal der niveaugebundene Sensor-Eingang 1 erregt. Dieser Zustand bleibt, wie das Startsignal, bis zum nächsten Übernahme-, Eingabe- oder Ausgabebefehl für diesen Puffer erhalten. Der Start aus Puffer 0 ist durch den gleichzeitig erregten Sensor 2 zu identifizieren (bei MINCAL 4N).

Die gemeinsamen Sensor-Eingänge heißen SR1...SR7.

Die niveau-gebundenen Sensor-Eingänge heißen

SR01...07	für Niveau 0	
11...17	"	1
21...27	"	2
31...37	"	3
41...47	"	4
51...57	"	5
61...67	"	6
71...77	"	7

Die Ein- und Ausgabe von Daten geschieht bei den Rechnern MINCAL 4E/4N vor allem über Puffer von Wortlänge; der Puffer-Typ E ist bei MINCAL 4E vorgesehen, sowie stets für den Puffer Ø des MINCAL 4N. Auch für die übrigen Puffer des MINCAL 4N wird vorzugsweise dieser Typ verwendet.

Puffer werden durch Instruktionen der Befehlsgruppe 6 gesteuert.

Der E-Puffer erlaubt folgende Ein/Ausgabe-Funktionen:

1. Stellenserielle, 4-bit-parallele Ein- und Ausgabe von Daten (z.B. BCD-Ziffern) einschließlich Vorzeichen aus externen Informationsträgern bzw. in externe Register. Durch die programmierte Adresse sind 512 (oktal 177) verschiedene externe Datengeber und -empfänger anwählbar; ihre Stellen (z.B. BCD-Stellen) werden nacheinander von einem Stellenkanal selektiert. Die Ein/Ausgebegeschwindigkeit ist entsprechend den äußeren Verhältnissen im Puffer fest einstellbar; die Dauer des Ein/Ausgabezyklus' liegt damit fest. Rechts- oder Linkskomma-Ein/Ausgabe wird im Puffer vorgegeben.

Diese Betriebsart ist dann wirksam, wenn in der Ergänzungsstelle der Instruktion eine der Ziffern 1...4 programmiert ist, wobei die programmierte Adresse durch das betreffende Indexregister 1...4 modifiziert wird, wenn 1...4 in der Ergänzung angegeben ist.

2. Stellenserielle Ein- und Ausgabe in Verbindung mit seriellen Eingabe- und Registriergeräten. Es sind jeweils 3 Geräte anwählbar, gekennzeichnet durch die Ziffern 5, 6 und 7 in der Ergänzungsstelle der Instruktion. Die Adressstellen enthalten in diesem Falle Angaben über das Ein/Ausgabeformat (näheres siehe Formatliste, Schreibmaschine, Streifenleser, -locher). Die Ein/Ausgebegeschwindigkeit hängt vom Eingabe- bzw. Registriergerät ab.

Es sind jeweils 3 verschiedene Eingabe- und Ausgabebefehle sowie ein Übernahmefehl in Verbindung mit den Puffern vorgesehen (s. Instruktionsliste). Die Befehle IB... bzw. OB... lösen einen Eingabe- bzw. Ausgabevorgang aus, indem sie einen Arbeitszyklus im Puffer starten. Dieser Zyklus läuft unabhängig vom zentralen Teil des Rechners, der in dieser Zeit entweder wartet (IBG, OBW) oder das Programm fortsetzt.

Das Ende des Pufferzyklus' äußert sich je nach dem Befehl, der den Pufferzyklus auslöste: Bei IBG wird die eingegebene Information ins W-Register des Rechners übernommen, und das Programm läuft, ebenso wie bei OBW, weiter. Bei IBR und OBR bewirkt das Ende des Pufferzyklus' nichts. Dagegen wird bei IBP und OBP am Ende des Zyklus' ein Niveau-Start ausgelöst, gleichzeitig mit der Erregung des niveaugebundenen Sensor-eingangs 1 (bzw. 2). Start und Sensor bleiben so lange erregt, bis ein erneuter Befehl der Gruppe 6 auf den Puffer angewendet wird.

Außerdem gibt es einen Befehl GB, mit dem der Inhalt des Puffers in das W-Register übernommen wird; er löst keinen Pufferzyklus aus.

Bei Ein/Ausgabebefehlen nach Punkt 2 (mit Formatangaben) werden statt IB... bzw. OB... die symbolischen Befehle K..., R..., Z... bzw. W..., P..., X... verwendet.

Zu beachten ist, daß beim MINCAL 4N stets der zum jeweiligen Programm-Niveau gehörige Puffer angesteuert wird (die Puffer Ø...7 sind den Niveaus Ø...7 zugeordnet; siehe Vorrangsystem). Nur der Puffer Ø ist von jedem Niveau aus anwählbar, wenn die betreffende Instruktion mit negativem Vorzeichen versehen wird. Der aus dem Puffer kommende Niveau-Start wirkt stets auf das Niveau, das den Puffer ausgelöst hatte. Gleichzeitig wird zur Identifizierung der niveaugebundene Sensor 2 erregt.

Der Puffer des MINCAL 4E kann von beiden Niveaus aus gestartet werden; der Start geht jedoch stets auf Niveau 1.

1-11		MINCAL 4E/4N PROGRAMMIERUNG	HEINRICH DIETZ INDUSTRIE-ELEKTRONIK
------	--	--------------------------------	--

Beim MINCAL 4E/4N wird eine elektrische Eingabe/Ausgabe-Schreibmaschine für den Programmierbetrieb sowie für manuelle Dateneingabe und Klartext-Registrierung verwendet.

Diese Schreibmaschine hat folgende Eigenschaften:

Typ: IBM 735 BCD mit Sonderverdrähtung (Kugelkopf)  
 Schreibgeschwindigkeit: 15 Zeichen pro Sekunde  
 Schreibbreite: 130 Zeichen pro Zeile; 10 Zeichen pro Zoll  
 Zeilenabstand: 1/6 Zoll  
 Stachelpulze: auf Wunsch; dann ca. 125 Zeichen/Zeile  
 Zeichenvorrat: 10 Ziffern, 26 Buchstaben, 27 Symbole  
 - 63 Charaktere (auf beiden Kugelkopf-Hälften)  
 (siehe Tabelle BCD-Code)  
 Funktionen: Leerschritt, Tabulator, Zeilenwechsel, Schreibkopfrücklauf, Umschaltung auf Sonderfarbe (Rot) und Normalfarbe (Schwarz), Rückschritt

Bei der Programmierung ist folgendes zu beachten:

1. Dezimale Datenstellen beinhalten die Ziffern 0...9; der Ziffer 0 entspricht kein bit in einer 4 bit-Gruppe (BCD-Stelle) des W-Registers.
2. Oktale Datenstellen beinhalten die Ziffern 0...7; der Ziffer 0 entspricht kein bit in einer 3 bit-Gruppe (Oktalstelle) des W-Registers.
3. Alphanumerische Datenstellen umfassen alle 63 möglichen Charaktere; der Ziffer 0 entsprechen, von rechts, das 2. und 4. bit (Bedeutung  $2 + 8 = 10$ ) einer 6 bit-Gruppe (Alpha-Stelle) des W-Registers. Kein bit in dieser Gruppe entspricht einem Leerschritt. Die Umschaltung der beiden Kugelkopfhälften erfolgt automatisch.
4. Bei Eingabe ist nur "Leertaste" als Endzeichen programmierbar und wirksam.
5. Bei Eingabe des Vorzeichens ist für Minus das Minuszeichen (-), für Plus Leertaste zu benutzen; entsprechend erfolgt die Ausgabe.
6. Bei dezimaler und oktaler Eingabe wird Leertaste als 0 interpretiert.
7. Alle Funktionen können bei der Ausgabe nur als "Endzeichen" programmiert werden. Ausnahme: Leerschritt (siehe 3 und 5).
8. Schreibkopfrücklauf ist mit Zeilenwechsel verbunden; Dauer von Rücklauf und Tabulation ca. 1 sec/10". Tabulatoren sind beliebig setzbar (Mindestabstand: 3 Zeichen).

Beim MINCAL 4E/4N werden 5-Kanal-Streifenleser und 8-Kanal-Streifenlocher für Programmierung und Daten-Ein/Ausgabe benutzt. Sie haben folgende Eigenschaften:

	MINCAL 4E	MINCAL 4N (auf Wunsch bei 4E)
Leser	30 Z/s (mechanisch)	300 Z/s (optisch)
Locher	20 Z/s	75 Z/s

Im allgemeinen, bei Programm-Ein/Ausgabe im Standardformat jedoch stets, wird der BCD-Code benutzt (siehe Tabelle BCD-Code). Es ist zu beachten:

1. Dezimale Datenstellen beinhalten die Ziffern 0...9; der Ziffer 0 entspricht kein bit in einer 4 bit-Gruppe (BCD-Stelle) des W-Registers, im Lochstreifen jedoch die Kanäle 2 und 4. Beim dezimalen Lesen werden nur die Kanäle 1 bis 4 berücksichtigt.
2. Oktale Datenstellen beinhalten die Ziffern 0...7; der Ziffer 0 entspricht kein bit in einer 3 bit-Gruppe (Oktalstelle) des W-Registers, im Lochstreifen jedoch die Kanäle 2 und 4. Beim oktalen Lesen werden nur die Kanäle 1 bis 3 berücksichtigt.
3. Alphanumerische Datenstellen entsprechen 64 Kombinationen der Kanäle 1, 2, 3, 4, 6 und 7. Der Ziffer 0 entsprechen, von rechts, das 2. und 4. bit ( $8 + 2 = 10$ ) einer 6 bit-Gruppe (Alpha-Stelle) des W-Registers und im Lochstreifen die Kanäle 2 und 4. Kein bit in einer Alpha-Stelle entspricht dem ZWR-Zeichen.
4. Es ist nur "ZWR" und "WR" als Endzeichen programmierbar.
5. Bei Eingabe des Vorzeichens ist für Minus das Minuszeichen (Kanal 7), für Plus das ZWR-Zeichen vorzusehen; entsprechend erfolgt die Ausgabe. Ist beim Lesen "Vorzeichen" programmiert, so wird auf Minus- oder ZWR-Zeichen geprüft; falls nicht vorhanden, erfolgt Prüfanzeige und Stop.
6. Bei der Ausgabe wird Kanal 5 (Prüfbit) automatisch so erregt, daß die Anzahl der Locher je Zeichen ungerade ist; beim Lesen wird darauf geprüft (bei Parity-Fehler: Prüfanzeige und Rechner-Stop).
7. Das ZWR-Zeichen besteht aus einem Loch in Kanal 5 (Prüfbit). Beim dezimalen oder oktalen Lesen wird ein ZWR-Zeichen als 0 interpretiert.
8. Das WR-Zeichen besteht aus einem Loch in Kanal 8. Ist dies beim Lesen als Endzeichen programmiert und wird kein WR-Zeichen gelesen, erfolgt Prüfanzeige und Rechner-Stop (Stellenkontrolle).
9. Die Zuführlochung (handbetätigt) besteht aus Lüchern in allen 8 Kanälen. Diese Kombination wird beim Eingeben überlesen.

Der Streifenlocher ist auf "8 KANAL-CODE" umschaltbar (handbetätigter Vorwahlschalter). Gegenüber dem BCD-Code ergeben sich folgende Unterschiede:

1. Die Ziffer 0 wird als Kanal 6 gelucht (identisch mit Zeichen 5).
2. Die Zuführlochung (handbetätigt) besteht aus Lüchern in den Kanälen 1 bis 7. (Beim Einlesen wird diese Kombination als Zeichen aufgefaßt).

1-13	77 76 75 74	MINCAL 4E/BN	HEINRICH DIETZ INDUSTRIE-ELEKTRONIK
		PROGRAMMIERUNG	

Streifenleser und -locher sind außerdem auf "5-KANAL-CODE" umschaltbar, um auch 5-Kanal-Streifen zu verarbeiten.

Dabei ist zu beachten:

1. Die Code-Umwandlung ist durch Programm vorzunehmen.
2. Die 5-Kanal-Zeichen sind beim Lesen und Lösen als alphanumerische Datenstellen aufzufassen. Vor- und Endzeichen sind im Format nicht anzugeben.
3. Die Zuordnung der 5 Kanäle zu den 6 bit einer Alpha-Stelle ist:

<u>5-Kanal-Streifen</u>	<u>Alpha-Stelle</u>
Kanal 5	bit 1 (rechts)
4	2
3	4
2	8
1	A
(stets leer)	B (links)

4. Prüfbit-Erzeugung und -Prüfung unterbleibt.
5. Der 5-Kanal-Streifen wird so eingelegt, daß Kanal 5 dort ist, wo beim 8-Kanal-Streifen Kanal 1 liegt.
6. Die Zuführlochung (handbetätigt) besteht aus Löchern in allen 5 Kanälen. (Beim Einlesen wird diese Kombination als Zeichen aufgefaßt).

Die symbolische Programmsprache (SPS) der Rechner MINCAL 4E/4N erlaubt die Formulierung beliebiger Programme mit leicht merkbaren symbolischen Bezeichnungen für Befehle, Adressen, Formate und Definitionen. Ein symbolisch geschriebenes Programm wird mit Hilfe eines Umwandlungsprogramms (Assembler) in eine Form gebracht, die der Maschinensprache des MINCAL 4E/4N entspricht.

Ein SPS-Programm ist eine Folge von Anweisungen; jede Anweisung entspricht im Klartext einer Zeile bzw. im Lochstreifen einem Satz von Zeichen. Die Anweisungen enthalten verschiedene Arten von Angaben, die in bestimmten Spalten angeordnet sind.

Bei der Aufstellung eines symbolischen Programms ist folgendes zu beachten:

**1. Spalte "Befehl"**

Jede Anweisung enthält entweder eine Instruktion oder eine Definition. Instruktionen sind durch 2 oder 3 Buchstaben gekennzeichnet, die linksbündig in den 3 Befehlsspalten stehen. Definitionen sind durch einen Buchstaben oder einen Bruchstaben und eine Ziffer gekennzeichnet, die ebenfalls linksbündig in die Befehlsspalten eingetragen werden. Es sind nur die in der SPS vorgesehenen Symbole zulässig.

- 1.1 Die erste Anweisung eines SPS-Programms ist eine Definition C mit der Angabe der verfügbaren Kernspeicherkapazität.
- 1.2 Die folgenden, maximal 8 Anweisungen sind Definitionen S... zur Angabe der Startadressen der einzelnen Programm niveaus. Nicht benutzte bzw. nicht vorhandene Niveaus bedürfen keiner Anweisung.
- 1.3 Die erste darauf folgende, andersartige Anweisung ist die erste echte Programm-Instruktion oder -Definition; sie belegt den Speicherplatz 120 der Seite 0 (Adresse 00120) im umgewandelten Programm.
- 1.4 Jeder weiteren Anweisung entspricht im umgewandelten Programm ein weiterer Speicherplatz in aufsteigender Reihenfolge der Adressen.

Von dieser Regel gibt es jedoch folgende Ausnahmen:

- 1.4.1 Die Definitionen M und X dienen der Zuordnung von symbolischen und echten Adressen; ihnen entspricht kein Speicherplatz im umgewandelten Programm. Sie können im SPS-Programm an beliebiger Stelle stehen.
- 1.4.2 Der Definition F (Variablen-Feld) entspricht im umgewandelten Programm ein zusammenhängender Bereich von Speicherplätzen.
- 1.4.3 Bei der Umwandlung werden am Ende der einzelnen Speicher-Seiten automatisch indirekte Adressen in der notwendigen Zahl bereitgestellt, die durch einen Verzweigungsbefehl Übersprungen werden. Im SPS-Programm wird dies nicht berücksichtigt.
- 1.5 Nach der letzten echten Programmanweisung ist die Anweisung Z (Programmende) zu geben, d.h. Ende des symbolischen Programms.

## 2. Spalte "Ergänzung"

Die Ergänzung wird nur bei folgenden Instruktionen angegeben (wie bei Definitionen):

- 2.1 Bei registerbezogenen Instruktionen muß in der Ergänzung das Register angegeben werden; zu verwenden sind die Symbole 1, 2, 3, 4, H, K und X.
- 2.2 Das gleiche gilt für die mit Nullinhalt, positivem oder negativem Inhalt eines Registers bedingten Verzweigungs-Instruktionen.
- 2.3 Bei den speicherbezogenen Lade-, Transfer-, arithmetischen und logischen Instruktionen ist in der Ergänzung dann eine Ziffer einzutragen, wenn Adressmodifikation über ein bestimmtes Indexregister erwünscht ist; zu verwenden sind die Ziffern 1, 2, 3 oder 4.
- 2.4 Das gleiche gilt für die Eingabe/Ausgabe-Instruktionen über Puffer und Parallelkanal, wenn Adressmodifikation erwünscht ist.
- 2.5 Bei den Verzweigungs-Instruktionen BR und BRY wird dann eine Ziffer eingetragen, wenn die Verzweigung mit einem der Sensor-Eingänge bedingt ist; zu verwenden sind die Ziffern 1, 2, 3, 4, 5, 6 oder 7.
- 2.6 Beim Unterprogramm-Sprung CS und CSY ist stets das Register anzugeben, welches die Rückkehradresse speichert, zu verwenden sind die Ziffern 1, 2, 3 oder 4 (eventuell auch der Buchstabe X).

## 3. Spalte "Vorzeichen"

Diese Spalte wird bei Bedarf mit einem Minuszeichen (-) belegt; es ist in folgenden Fällen erforderlich:

- 3.1 Bei den Festwert-Instruktionen LDC und ADC kennzeichnet es einen negativen Festwert.
- 3.2 Bei allen speicherbezogenen Instruktionen bedeutet ein Minuszeichen, daß das folgende (3-stellig oktale) Merkmal die echte oktale Adresse in Seite 0 des Speichers bezeichnet.
- 3.3 Bei den Ein/Ausgabe-Instruktionen der Befehlsgruppe 6 bedeutet negatives Vorzeichen, daß unabhängig vom jeweiligen Niveau stets der Puffer 0 angewählt wird (gilt für MINCAL 4N).
- 3.4 Bei den Definitionen B, C, D und A wird damit das negative Vorzeichen des Festwerts festgelegt.
- 3.5 Bei der Definition Y bei einer doppelt indirekten Adresse.
- 3.6 Bei der Definition M: wie in 3.2 beschrieben.
- 3.7 Bei der Definition X: wie in 3.3 beschrieben.

#### 4. Spalte "Adressmerkmal"

Diese Gruppe umfaßt 6 Spalten, von denen meist weniger belegt sind. Eintragungen sind hier stets linksbündig vorzunehmen; sie haben folgende Bedeutung:

- 4.1 Bei den Schritt-Instruktionen SR, SL und SRR ist hier die Anzahl der Schiebstellen dezimal anzugeben. Die Zahl ist 1- oder 2-stellig, mit einem Dezimalpunkt bei einer Stelle (1 bis 17).
- 4.2 Bei den Festwert-Instruktionen LJC und ADC wird der Betrag des Festwerts (0 bis 511) dezimal eingetragen; als 1-, 2- oder 3-stellige Ganzzahl; nach einer 1- oder 2-stelligen Zahl ist ein Dezimalpunkt vorgeschrieben.
- 4.3 Bei allen speicherbezogenen Instruktionen wird hier die Adresse angegeben, und zwar entweder:
  - 4.3.1 als symbolisches Merkmal; zulässig sind 1-, 2- oder 3-stellige, linksbündig zusammenhängende alphanumerische Merkmale aus den 26 Buchstaben A...Z, den Ziffern 0...9 und den Zeichen - (Minus), & (Und) sowie / (Schrägstrich); es dürfen jedoch nicht nur Ziffern darin vorkommen;
  - 4.3.2 als 3-stellige oktale, echte Adressen in Seite # des Speichers (mit einem Minuszeichen in der Vorzeichenpalte; siehe 3.2).

Im allgemeinen wird der erste Fall angewendet; das Umwandlungsprogramm ermittelt aus den symbolischen Adressen aufgrund der Instruktionsmerkmale (s. später) die echte Adresse.

Die zweite Methode wird dann benutzt, wenn Plätze mit Adressen unter 120, insbesondere Registerplätze, angesprochen werden sollen.

Symbolische Adressmerkmale müssen eindeutig sein, d.h. sie dürfen im Programm nur an einer einzigen Stelle als Instruktionsmerkmal verwendet werden. Eine Übereinstimmung zwischen speicherbezogenen und externen Merkmalen (s. 4.4) ist jedoch zulässig.

- 4.4 Bei allen Ein/Ausgabe-Instruktionen (außer solchen in Formatschreibweise) enthalten diese Spalten die externe Adresse, entweder

4.4.1 als symbolisches Merkmal (Vorschriften siehe 4.3.1),  
oder

4.4.2 als 3-stellige Watzzahl, die der externen Adresse entspricht.

Für den Fall 4.4.1 ist stets eine X-Definition erforderlich, die dem Umwandlungsprogramm die symbolische der echten Adresse zuordnen hilft.

- 4.5 Bei Ein/Ausgabe-Instruktionen in Formatschreibweise ist hier das Format einzutragen, und zwar

in die erste Spalte die Darstellung (L, R, 0, A),  
in die zweite Spalte, falls Ein/Ausgabe mit Vorzeichen vorgesehen, ein Minuszeichen (sonst leer),  
in die dritte Spalte die Stellenzahl (Ziffern 0 bis 6),  
in die vierte Spalte, falls vorgesehen, das Endzeichen (X, T, Z, E, B, N, F).

Ist z.B. nur Endzeichen erwünscht, kann dessen Symbol in die 1. Spalte mit Dezimalpunkt danach eingetragen werden.

2-3	✓ T ✓ ✓	MINCAL 4E/4N PROGRAMMIERUNG	HEINRICH DIETZ INDUSTRIE-ELEKTRONIK
-----	---------------	--------------------------------	--

- 4.6 Bei den Instruktionen **MCP**, **HCP** und **ECR** ist in der ersten Spalte des Adreßmerkmals mit einer Ziffer 0...7 das Niveau anzugeben, das gestartet werden soll.
- 4.7 Beim Register-Transfer **TRR** ist in der ersten Spalte des Adreßmerkmals das Register anzugeben, in das transferiert wird (1 Stelle mit 1, 2, 3, 4, N, W oder X).
- 4.8 Zu den Definitionen **D** und **B** ist hier entweder eine dezimale Ganzzahl oder ein dezimaler Bruch anzugeben. Die Ganzzahl kann 1 bis 5 Stellen haben hinter der letzten Stelle ( $10^0$ ) ist ein Dezimalpunkt zu setzen. Ein Bruch ist dadurch gekennzeichnet, daß er mit einem Dezimalpunkt beginnt, darauf folgen 1 bis 5 Stellen; die erste hat den Wert  $10^{-1}$ , Leerstellen am Schluß werden als 0 gelesen.
- 4.9 Zur Definition **0** ist stets eine 6-stellige Oktalzahl einzutragen.
- 4.10 Die Definition **A** ist mit einem 3-stelligen alphanumerischen Wort in den ersten 3 Adreßmerkmal-Spalten zu versehen; dieses Wort kann sich in beliebiger Weise aus den 63 Charakteren des BCD-Codes sowie aus Leerstellen zusammensetzen.
- 4.11 Für die Definition **F** ist die Feldgröße als 1-, 2- oder 3-stellige Dezimalzahl einzutragen. Bei weniger als 3 Stellen ist hinter der letzten Stelle ein Dezimalpunkt vorzusehen.
- 4.12 Für die Definition **E** ist die Lage des Elements im Feld in gleicher Weise anzugeben.
- 4.13 Bei der Definition **Y** (indirekte Adresse) ist ein Adreßmerkmal nach den in 4.3 aufgestellten Regeln anzugeben.
- 4.14 Die Zuordnungs-Definitionen **M** und **X** erfordern eine 3-stellige oktale Adresse.

### 5. Spalten "Instruktionsmerkmal"

Das Instruktionsmerkmal dient zur Identifizierung bestimmter Anweisungen im SPS-Programm hinsichtlich ihrer Lage; sie werden vom Umwandlungsprogramm mit den Adreßmerkmalen verglichen, um so die echten Adressen herauszufinden.

**Wichtig:** Die Instruktions-Merkale müssen eindeutig sein; jedes darf im Programm nur einmal vorkommen. (Ausnahme: Eine symbolische externe Adresse vor einer X-Definition darf mit einem Instruktions-Merkmal übereinstimmen).

Jedes symbolische Adreßmerkmal muß seine Entsprechung in einem und nur einem exakt gleichen Instruktionsmerkmal finden.

Das Instruktionsmerkmal ist stets symbolisch (Regeln siehe 4.3.1) und steht in 3 Spalten vor den Befehlsspalten.

Es muß bei folgenden Anweisungen eingeführt werden:

- 5.1 Bei allen Instruktionen, auf die von irgend einer Stelle des Programms her verzweigt wird.
- 5.2 Am Beginn von Unterprogrammen.
- 5.3 Bei allen Speicherplätzen, die von speicherbezogenen Instruktionen angesprochen werden, insbesondere also bei den Definitionen V, F, E, D, G und A; aber auch vor Instruktionen, die vom Programm abgefragt, verändert oder ausgetauscht werden.
- 5.4 Vor indirekten Adressen (Definition Y).
- 5.5 Als symbolische Merkmale vor den Zuordnungs-Definitionen N und X.

## 6. Bemerkungs-Spalten

Es ist möglich, im SPS-Programm zu jeder Anweisung einen Kommentar zu schreiben, der mit einem Schrägstrich hinter der letzten Adreß-Merkmal-Spalte beginnt und maximal 51 Zeichen umfaßt. Ebenso kann durch Eingabe eines Schrägstrichs in der 1. Spalte des Satzes überall eine Kommentar-Zeile eingefügt werden. Es sind alle 63 alphanumerischen Charaktere erlaubt, ebenso natürlich Leerstellen.

Die Informationen in den Bemerkungs-Spalten haben keinen Einfluß auf die Programm-Umwandlung.

Ebenso kann durch Eingabe eines Schrägstrichs in der 1. Spalte des Satzes überall eine Kommentar-Zeile eingefügt werden.

<u>Symbol</u>	<u>Oktalcode</u>	<u>Bem.</u>	<u>Dauer (μs)</u>	<u>Bedeutung</u>
NOP	000000		40	Keine Operation
MSP	010000		60	Vorzeichen Plus
MSM	020000		60	Vorzeichen Minus
MCS	030000		60	Vorzeichen Wechselt
SR	0400BB		100	Schiften rechts
SL	0500BB		100	Schiften links
SRR	0600BB		100+r	Schiften rechts mit Runden
SLN	070000		100	Normalisieren
LDC	*10RCCC	c	60	Laden Festwert
LDR	11R000	s	80	Laden Register
LD	*121AAA	a	80+i	Laden
LDY	*131AAA	a	100+i+y	Laden indirekt
ADC	*14RCCC	c	80+n,u	Addieren Festwert
TRR	15R00A		80	Transfer Register in Register
TR	*161AAA	a	80+i	Transfer
TRY	*171AAA	a	100+i+y	Transfer indirekt
AD	*201AAA	a	80+i+n,u	Addieren
ADY	*211AAA	a	100+i+n,u+y	Addieren indirekt
SB	*221AAA	a	80+i+n,u	Subtrahieren
SBY	*231AAA	a	100+i+n,u+y	Subtrahieren indirekt
FO	*241AAA	a	80+i	ODER
FOY	*251AAA	a	100+i+y	ODER indirekt
FA	*261AAA	a	80+i	UND
FAY	*271AAA	a	100+i+y	UND indirekt
MP	*301AAA	a	840+i	Multiplizieren
MPY	*311AAA	a	860+i+y	Multiplizieren indirekt
DV	*321AAA	a	1240+i	Dividieren
DVY	*331AAA	a	1260+i+y	Dividieren indirekt
VBL	340000		900	Dezimal-Binär Linkskomma
VBR	350000		900	Dezimal-Binär Rechtskomma
VDL	360000		900...1260	Binär-Dezimal Linkskomma
VDR	370000		900...1260	Binär-Dezimal Rechtskomma
BR	*40SAAA	a	40+y	Verzweigen
BRY	*41SAAA	a	60+v+y	Verzweigen indirekt
BZ	*42RAAA	a	60+v	Verzweigen wenn Null
BZY	*43RAAA	a	60+v+y	Verzweigen wenn Null indirekt
BP	*44RAAA	a	60+v	Verzweigen wenn Plus
BPY	*45RAAA	a	60+v+y	Verzweigen wenn Plus indirekt
BM	*46RAAA	a	60+v	Verzweigen wenn Minus
BMY	*47RAAA	a	60+v+y	Verzweigen wenn Minus indirekt
CS	*50RAAA	a	100	Unterprogramm
CSY	*51RAAA	a	120+y	Unterprogramm indirekt

<u>Symbol</u>	<u>Oktalcode</u>	<u>Bem.</u>	<u>Dauer (us)</u>	<u>Bedeutung</u>
NCP	5300L0		60	Niveauwechsel
HLT	540000		60	Halt
HCP	5500L0		60	Halt mit Niveauwechsel
END	560000		60	Ende
ECP	5700L0		60	Ende mit Niveauwechsel
GB	*600000	b	80	Übernehmen Puffer
IBG	*611AAA	b	100+i+w	Eingabe in Puffer, Übernehmen
IBR	*621AAA	b	60+i	Eingabe in Puffer
IBP	*631AAA	b	60+i	Eingabe in Puffer mit Niveau-Start
OBW	*651AAA	b	80+i+w	Ausgabe über Puffer, Warten
OBR	*661AAA	b	80+i	Ausgabe über Puffer
OBP	*671AAA	b	80+i	Ausgabe über Puffer mit Niveau-Start
KG	*61DFFF	b	100+w	auch: RG , XC (sonst wie oben)
KR	*62DFFF	b	60	" RR , XR "
KP	*63DFFF	b	60	" RP , XP "
WW	*65DFFF	b	80+w	" PW , ZW "
WR	*66DFFF	b	60	" PR , ZR "
WP	*67DFFF	b	80	" PP , ZP "
GX	700000		60	Übernehmen parallel
IXG	711AAA		100+i+w	Eingabe parallel, Übernehmen
IXR	721AAA		60+i	Eingabe parallel
IXP	731AAA		60+i	Eingabe parallel mit Niveau-Start
OXW	751AAA		80+i+w	Ausgabe parallel, Warten
OXR	761AAA		80+i	Ausgabe parallel
OXP	771AAA		80+i	Ausgabe parallel mit Niveau-Start

Erklärungen:

- \* Negatives Vorzeichen bedeutet:
  - a Adresse liegt in Seite 0 (sonst jeweilige Seite)
  - b Adresse des Puffers 0 (sonst niveaugehöriger Puffer)  
(bei MINCAL 4N)
  - c negativer Festwert (sonst positiver Festwert)

A	Adresse	000...777	oktale Adresse
B	Größe der Verschiebung	01...21	Binärstellen (oktal)
C	Festwert	000 77	oktaier Festwert
D	Gerät	5	1.Gerät
		6	2.Gerät
		7	3.Gerät
F	Format		s. Formatliste
I	Indexregister, niveaugebunden	0	keine Adreßmodifikation
L	Niveau	1... 4	Adreßmodifikation, durch IXR1...4
R	Allgemeines Register, niveaugebunden	0... 7	Niveau 0...7
S	Sensor (Bedingungseingang)	0... 7	Startadresse, IXR1...4, N-, W-, X-Register
		0	unbedingt
		1... 7	bedingt mit Sensor 1...7

i	bei Adreßmodifikation	20 $\mu$ s
n	bei Addition/Subtraktion durch Null	20 $\mu$ s
r	bei Aufrundung	20 $\mu$ s
u	bei Überlauf	20 $\mu$ s
v	wenn Verzweigung stattfindet	20 $\mu$ s
w	Wartezeit	(unbestimmt)
y	bei mehrfach indirekter Auressierung	je 20 $\mu$ s

Die Befehle der Gruppe 3 sind bei MINCAL 4E nur bei entsprechender Ausrüstung zulässig.

Die Befehle der Gruppe 6 gelten für Puffer vom E-Typ.

Die Befehle der Gruppe 7 sind nur bei Ausrüstung mit Parallel-Ein/Ausgang sinnvoll.

Der Neustart eines Niveaus (nach Ende) benötigt 20  $\mu$ s zusätzlich.

NOP

\*00XXXX

KEINE OPERATION

Bedeutung: Diese Operation wird übersprungen.

Bemerkung: Das Vorzeichen \* und die 4 Oktalstellen XXXX der Instruktion sind ohne Bedeutung. Positives Vorzeichen und 0000 für XXXX werden empfohlen.

Beispiel:

NOP

000000

Keine Operation

MSP	*01XXXX	VORZEICHEN PLUS
MSM	*02XXXX	VORZEICHEN MINUS
MCS	*03XXXX	VORZEICHEN WECHSELN

Bedeutung: Das Vorzeichen des W-Registers wird auf Plus gesetzt (MSP), auf Minus gesetzt (MSM) oder in sein Gegenteil verkehrt (MCS). Der Betrag des W-Registers bleibt erhalten.

Bemerkung: Das Vorzeichen \* und die 4 Oktalstellen XXXX der Instruktion sind ohne Bedeutung. Positives Vorzeichen und 0000 für XXXX werden empfohlen

Beispiele:

MSP	010000	Setze Vorzeichen Plus
MSM	020000	Setze Vorzeichen Minus
MCS	030000	wechsle Vorzeichen

SR	*04XXBB	SCHIFTFEN RECHTS
SL	*05XXBB	SCHIFTFEN LINKS
SRR	*06XXBB	SCHIFTFEN RECHTS MIT RUNDEN

**Bedeutung:** Der Inhalt des W-Registers wird um die in den Oktalstellen BB angegebene Zahl von Binärstellen nach rechts (SR, SRR) oder nach links (SL) verschoben. Die freierdenden Stellen des W-Registers bekommen Nullinhalt; das Vorzeichen des W-Registers bleibt erhalten.

Die Größe der Verschiebung kann 1 bis 17 Binärstellen betragen.

Die beim Rechtsschiften rechts überlaufenden Stellen werden linksbündig in das X-Register gebracht und dessen übrige Stellen Null gesetzt; das Vorzeichen des X-Registers wird Plus.

Die beim Linksschiften links überlaufenden Stellen werden rechtsbündig in das X-Register gebracht und dessen übrige Stellen Null gesetzt; das Vorzeichen des X-Registers wird Plus.

Beim Rechtsschiften mit Runden (SRR) wird der Absolutbetrag der verschobenen Zahl im W-Register um 1 erhöht, wenn die höchste rechts überlaufende Stelle den Inhalt 1 hat.

**Bemerkung:** Das Vorzeichen \* und die 2 Oktalstellen XX der Instruktion sind ohne Bedeutung. Positives Vorzeichen und ØØ für XX werden empfohlen.

**Beispiele:**

SR	1.	040001	Schifte 1 Stelle rechts
SR	17	040021	" 17 Stellen "
SL	1.	050001	Schifte 1 Stelle links
SL	17	050021	" 17 Stellen links
SRR	12	060014	Schifte 12 Stellen rechts und runde

SLN

\* 07XXXX

NORMALISIEREN

**Bedeutung:** Der Inhalt des W-Registers wird so weit nach links geschoben, daß die höchste Binärstelle eine 1 enthält. Die freiwerdenden Stellen des W-Registers bekommen Nullinhalt; das Vorzeichen des W-Registers bleibt erhalten.

Die Anzahl der Binärstellen, um die der Inhalt des W-Registers nach links verschoben wurde, steht rechtsbündig (binär) im X-Register; sein Vorzeichen wird Plus gesetzt. Hatte das W-Register Nullinhalt, steht im X-Register rechtsbündig die Oktalzahl 22.

**Bemerkung:** Das Vorzeichen \* und die 4 Oktalstellen XXXX der Instruktion sind ohne Bedeutung. Positives Vorzeichen und 0000 für XXXX werden empfohlen.

**Beispiel:**

SLN

070000

Normalisiere

LDC	*10RCCC	LADEN FESTWERT
ADC	*14RCCC	ADDIEREN FESTWERT

Bedeutung: Das in der Ergänzung angegebene Register R wird gelöscht und mit dem programmierten Festwert geladen (LDC).

Zum Inhalt des in der Ergänzung angegebenen Registers R wird der programmierte Festwert vorzeichenrichtig addiert (ADC). Die Summe steht im Register R. Ein eventueller Überlauf bewirkt, daß zum Absolutbetrag des X-Registers eine rechtsbündige 1 addiert wird. Addition durch Null wird hinsichtlich Vorzeichen und Betrag richtig durchgeführt.

Festwert: Das Vorzeichen \* der Instruktion gibt das Vorzeichen des Festwerts an, die 3 Oktalstellen CCC den Betrag. Es sind Festwerte von -00777 bis +00777 programmierbar, entsprechend natürlichen Zahlen von -511 bis +511. Der Festwert Null ist zulässig.

Hinweis: Wenn der Befehl ADC auf das X-Register angewendet wird, ist ein Überlauf zu vermeiden.

Beispiele:

LDC1 8.	101010	Lade in IXR1 Festwert 8
LDC2 8.	102010	" " IXR2 "
LDC3 8.	103010	" " IXR3 "
LDC4 8.	104010	" " IXR4 "
LDCW 8.	106010	Lade in W-Register Festwert 8
LDCX 8.	107010	" " X-Register "
LDCW 0.	106000	Lade in W-Register Null
LDCW 511	106777	" " " Festwert 511
LDCW-511	-106777	" " " " -511
ADCW 1.	146001	Addiere zu W-Register Festwert 1
ADCW 511	146777	" " " " 511
ADCW-1.	-146001	Subtrahiere von W-Register Festwert 1
ADCW-511	-146777	" " " " -511

LDR	*11RXXX	LADE REGISTER
TRR	*15R00A	TRANSFER REGISTER IN REGISTER

Bedeutung: Das in der Ergänzung angegebene Register R wird gelöscht und mit dem Inhalt des X-Registers, einschließlich Vorzeichen, geladen (LDR).

Der Inhalt des in der Ergänzung angegebenen Registers wird, einschließlich Vorzeichen, in ein anderes Register transferiert, das vorher gelöscht wurde (TRR).

Bemerkung: Bei LDR ist das Vorzeichen \* und die 3 Oktalstellen XXX der Instruktion ohne Bedeutung. Positives Vorzeichen und 000 für XXX werden empfohlen.

Bei TRR ist die Einer-Adresse des Registers, in das transferiert wird, in der 1er-Adresse anzugeben; die 10er- und 10er-Adresse müssen stets 0 sein. Das Vorzeichen \* ist ohne Bedeutung; positives Vorzeichen wird empfohlen.

Hinweis: Die Instruktion TRR dient auch zur Rückverzweigung am Ende eines Unterprogramms.

Beispiele:

LDR1	111000	Lade IXR1 mit Inhalt W-Register
LDR2	112000	" IXR2 " " "
LDR3	113000	" IXR3 " " "
LDR4	114000	" IXR4 " " "
LDRX	117000	Lade X-Register mit Inhalt W-Register
TRR1 W	151006	Transferiere Inhalt IXR1 in W-Register
TRR2 W	152006	" " IXR2 " "
TRR3 W	153006	" " IXR3 " "
TRR4 W	154006	" " IXR4 " "
TRRX W	157006	Transferiere Inhalt X-Register in W-Register
TRRW X	156007	" " W-Register in X-Register
TRR3 N	153005	" " IXR3 in N-Register (= Ende Unterprogramm; Rücksprung zu der in IXR3 gespeicherten Rückkehradresse)

LD	*121AAA	LADEN
LDY	*131AAA	LADEN INDIREKT

**Bedeutung:** Das W-Register wird gelöscht und mit dem Inhalt des adressierten Speicherplatzes, einschließlich dessen Vorzeichen, geladen.

**Adresse:** bei direkter Adressierung wird die programmierte Adresse AAA benutzt, die zur jeweiligen Seite gehört; ist das Vorzeichen \* der Instruktion negativ, die Adresse liegt der Seite 0.

Bei indirekter Adressierung wird ein Speicherplatz benutzt, dessen Adresse sich in AAA befindet, wobei AAA je nach positivem oder negativem Vorzeichen \* der Instruktion in der jeweiligen Seite oder in Seite 0 liegt.

**Indexregister:** Ist in der Ergänzungsstelle I anstelle von 0 eine der Ziffern 1, 2, 3 oder 4 angegeben, so wird Speicheradresse durch das Indexregister 1, 2, 3 oder 4 additiv verändert.

**Beispiele:**

LD	ABC	120375	Lade Inhalt der Adresse NN375
LD	DEF	-120624	" " " " 00624
LDY	ABC	130375	Lade Inhalt der in NN375 stehenden Adresse
LDY	DEF	-130624	" " " " 00624 " "
LD 4	ABC	124375	Lade Inhalt der Adresse (NN375 + IXR4)
LDY4	ABC	134375	" " " in NN375 stehenden, um (IXR4) veränderten Adresse

TR \*161AAA TRANSFER

TRY \*171AAA TRANSFER INDIREKT

**Bedeutung:** Der Inhalt des  $\omega$ -Registers wird, einschließlich Vorzeichen, in den adressierten Speicherplatz übertragen, der vorher gelescht wurde.

Das  $\omega$ -Register behält seinen Inhalt.

**Adresse:** Bei direkter Adressierung wird die programmierte Adresse AAA benutzt, die zur jeweiligen Seite gehört; ist das Vorzeichen \* der Instruktion negativ, die Adresse AAA der Seite Ø.

Bei indirekter Adressierung wird ein Speicherplatz benutzt, dessen Adresse sich in AAA befindet, wobei AAA je nach positivem oder negativem Vorzeichen \* der Instruktion in der jeweiligen Seite oder in Seite Ø liegt.

**Indexregister:** Ist in der Ergänzungsstelle I anstelle von Ø eine der Ziffern 1, 2, 3 oder 4 angegeben, so wird die Speicheradresse durch das Indexregister 1, 2, 3 oder 4 additiv verändert.

**Beispiele:**

TR ABC 160375 Transferiere nach Adresse NN375  
 TR DEF -160624 " " " 00624

TRY ABC 170375 Transferiere nach in NN375 stehender Adresse  
 TRY DEF -170624 " " " 00624 stehender Adresse

TR 4 ABC 164375 Transferiere nach Adresse (NN375 + IXR4)  
 TRY4 ABC 174375 Transferiere nach in NN375 stehender, um (IXR4) veränderter Adresse

AD \*201AAA ADDIEREN

ADY \*211AAA ADDIEREN INDIREKT

**Bedeutung:** Der Inhalt des adressierten Speicherplatzes wird vorzeichengerecht zum Inhalt des W-Registers addiert; die Summe steht im W-Register. Ein eventueller Überlauf bewirkt, daß zum Absolutbetrag des X-Register-Inhalts eine rechtsbündige 1 addiert wird. Addition durch Null wird hinsichtlich Vorzeichen und Betrag richtig ausgeführt.

**Adresse:** Bei direkter Adressierung wird die programmierte Adresse AAA benutzt, die zur jeweiligen Seite gehört; ist das Vorzeichen \* der Instruktion negativ, die Adresse AAA der Seite 0.

Bei indirekter Adressierung wird ein Speicherplatz benutzt, dessen Adresse sich in AAA befindet, wobei AAA je nach positivem oder negativem Vorzeichen \* der Instruktion in der jeweiligen Seite oder in Seite 0 liegt.

**Indexregister:** Ist in der Ergänzungsstelle I anstelle von 0 eine der Ziffern 1, 2, 3 oder 4 angegeben, so wird die Speicheradresse durch das Indexregister 1, 2, 3 oder 4 additiv verändert.

**Beispiele:**

AD ABC 200375 Addiere Inhalt der Adresse NN375  
AD DEF -200624 " " " " 00624

ADY ABC 210375 Addiere Inhalt der in NN375 stehenden Adresse  
ADY DEF -210624 " " " " 00624 " "

AD 4 ABC 204375 Addiere Inhalt der Adresse (NN375 + IXR4)  
ADY4 ABC 214375 Addiere Inhalt der in NN375 stehenden, um (IXR4) veränderten Adresse

SB \*21AAA SUBTRAHIEREN

SBY \*231AAA SUBTRAHIEREN INDIREKT

**Bedeutung:** Der Inhalt des adressierten Speicherplatzes wird vorzeichenrichtig vom Inhalt des W-Registers subtrahiert; die Differenz steht im W-Register. Ein eventueller Überlauf bewirkt, daß zum Absolutbetrag des X-Register-Inhalts eine rechtsbündige 1 addiert wird. Subtraktion durch Null wird hinsichtlich Vorzeichen und Betrag richtig durchgeführt.

**Adresse:** Bei direkter Addressierung wird die programmierte Adresse AAA benutzt, die zur jeweiligen Seite gehört; ist das Vorzeichen \* der Instruktion negativ, die Adresse AAA der Seite  $\emptyset$ .

Bei indirekter Addressierung wird ein Speicherplatz benutzt, dessen Adresse sich in AAA befindet, wobei AAA je nach positivem oder negativem Vorzeichen \* der Instruktion in der jeweiligen Seite oder in Seite  $\emptyset$  liegt.

**Indexregister:** Ist in der Ergänzungsstelle I anstelle von  $\emptyset$  eine der Ziffern 1, 2, 3 oder 4 angegeben, so wird die Speicheradresse durch das Indexregister 1, 2, 3 oder 4 additiv verändert.

**Beispiele:**

SB ABC 220375 Subtrahiere Inhalt der Adresse NN375  
 SB DEF -220624 " " " " 00624

SBY ABC 230375 Subtrahiere Inhalt der in NN375 stehenden Adresse  
 SBY DEF -230624 " " " " 00624 " "

SB 4 ABC 224375 Subtrahiere Inhalt der Adresse (NN375 + IXR4)  
 SBY4 ABC 234375 Subtrahiere Inhalt der in NN375 stehenden, um (IXR4) veränderten Adresse

FO	*241AAA	ODER
FOY	*251AAA	ODER INDIREKT

**Bedeutung:** Alle Binärstellen des adressierten Speicherplatzes werden mit den entsprechenden Binärstellen des W-Registers in ODER-Verknüpfung gebracht; das Ergebnis steht im W-Register.

Das Vorzeichen des W-Registers bleibt erhalten.

**Adresse:** Bei direkter Adressierung wird die programmierte Adresse AAA benutzt, die zur jeweiligen Seite gehört, ist das Vorzeichen \* der Instruktion negativ, die Adresse AAA der Seite Ø.

Bei indirekter Adressierung wird ein Speicherplatz benutzt, dessen Adresse sich in AAA befindet, wobei AAA je nach positivem oder negativem Vorzeichen \* der Instruktion in der jeweiligen Seite oder in Seite Ø liegt.

**Indexregister:** Ist in der Ergänzungsstelle I anstelle von Ø eine der Ziffern 1, 2, 3 oder 4 angegeben, so wird die Speicheradresse durch das Indexregister 1, 2, 3 oder 4 additiv verändert.

**Beispiele:**

FO	ABC	240375	ODER mit Inhalt der Adresse NN375
FO	DEF	-240624	" " " " 00624
FOY	ABC	250375	ODER mit Inhalt der in NN375 stehenden Adresse
FOY	DEF	-250624	" " " " 00624" "
FO 4	ABC	244375	ODER mit Inhalt der Adresse (NN375 + IXR4)
FOY4	ABC	254375	ODER mit Inhalt der in NN375 stehenden, um (IXR4) veränderten Adresse

FA \*261AAA UND

FAY \*271AAA UND INDIREKT

**Bedeutung:** Alle Binärstellen des adressierten Speicherplatzes werden mit den entsprechenden Binärstellen des W-Registers in UND-Verknüpfung gebracht; das Ergebnis steht im W-Register.  
Das Vorzeichen des W-Registers bleibt erhalten.

**Adresse:** Bei direkter Adressierung wird die programmierte Adresse AAA benutzt, die zur jeweiligen Seite gehört; ist das Vorzeichen \* der Instruktion negativ, die Adresse AAA der Seite Ø.

Bei indirekter Adressierung wird ein Speicherplatz benutzt, dessen Adresse sich in AAA befindet, wobei AAA je nach positivem oder negativem Vorzeichen \* der Instruktion in der jeweiligen Seite oder in Seite Ø liegt.

**Indexregister:** Ist in der Ergänzungsstelle I anstelle von Ø eine der Ziffern 1, 2, 3 oder 4 angegeben, so wird die Speicheradresse durch das Indexregister 1, 2, 3 oder 4 additiv verändert.

**Beispiele:**

FA ABC 260375 UND mit Inhalt der Adresse NN375  
FA DEF -260624 " " " " " 00624

FAY ABC 270375 UND mit Inhalt der in NN375 stehenden Adresse  
FAY DEF -270624 " " " " " 00624 " "

FA 4 ABC 264375 UND mit Inhalt der Adresse (NN375 + IXR4)  
FAY4 ABC 274375 UND mit Inhalt der in NN375 stehenden, um (IXR4) veränderten Adresse

MP \*301AAA MULTIPLIZIEREN

MPY \*311AAA MULTIPLIZIEREN INDIREKT

Bedeutung: Der Inhalt des adressierten Speicherplatzes wird vorzeichengerecht mit dem Inhalt des W-Registers multipliziert; die bedeutenden Stellen des Produkts und sein Vorzeichen stehen im W-Register, die unbedeutenden Stellen (mit stets positivem Vorzeichen) im X-Register.

Rechengang: W vorher Operand W nachher X nachher

-400000	· 400000	= -200000	000000
000001	· 000001	= 000000	000001

Adresse: Bei direkter Adressierung wird die programmierte Adresse AAA benutzt, die zur jeweiligen Seite gehört; ist das Vorzeichen \* der Instruktion negativ, die Adresse AAA der Seite 0.

Bei indirekter Adressierung wird ein Speicherplatz benutzt, dessen Adresse sich in AAA befindet, wobei AAA je nach positivem oder negativem Vorzeichen \* der Instruktion in der jeweiligen Seite oder in Seite 0 liegt.

Indexregister: Ist in der Ergänzungsstelle I anstelle von 0 eine der Ziffern 1, 2, 3 oder 4 angegeben, so wird die Speicheradresse durch das Indexregister 1, 2, 3 oder 4 additiv verändert.

Beispiele:

MP ABC 300375 Multiplizierte mit Inhalt der Adresse NN375  
 MP DEF -300624 " " " " " 00624

MPY ABC 310375 Multiplizierte mit Inhalt der in NN375 stehenden Adresse  
 MPY DEF -310624 " " " " " 00624 " "

MP 4 ABC 304375 Multiplizierte mit Inhalt der Adresse (NN375 + IXR4)  
 MPY4 ABC 314375 Multiplizierte mit Inhalt der in NN375 stehenden, um (IXR4) veränderten Adresse

Bei MINCAL 4E nur, wenn für Befehlsgruppe 3 ausgerüstet

DV \*321AAA DIVIDIEREN

DVY \*331AAA DIVIDIEREN INDIREKT

Bedeutung: Der Inhalt des W-Registers wird vorzeichengerecht durch den Inhalt des adressierten Speicherplatzes dividiert. Der Quotient steht einschließlich Vorzeichen im W-Register, der positive Rest im X-Register.

Ist der Quotient gleich oder größer als 1, jedoch kleiner als 2, so stehen seine binären Stellen ( $2^{-1} \dots 2^{-10}$ ) richtig im W-Register und der Rest im X-Register, das in diesem Falle Minus gesetzt wird. Bei Quotienten gleich oder größer als 2 enthalten alle Binärstellen des W-Registers eine 1, und das Vorzeichen des X-Registers wird Minus gesetzt.

Rechengang:	W vorher	Operand	W nachher	X nachher
	-200000	: 400000	- -400000	Rest 000000
	000001	: 400000	- 000002	Rest 000000
	200000	: 300000	- 525252	200000
	200000	: 200000	- 000000	-000000 Quotient = 1
	400000	: 200000	- 777777	-400000 Quotient = 2

Adresse: Bei direkter Adressierung wird die programmierte Adresse AAA benutzt, die zur jeweiligen Seite gehört; ist das Vorzeichen \* der Instruktion negativ, die Adresse AAA der Seite 0.

Bei indirekter Adressierung wird ein Speicherplatz benutzt, dessen Adresse sich im AAA befindet, wobei AAA je nach positivem oder negativem Vorzeichen \* der Instruktion in der jeweiligen Seite oder in Seite 0 liegt.

Indexregister: Ist in der Ergänzungsstelle I anstelle von 0 eine der Ziffern 1, 2, 3 oder 4 angegeben, so wird die Speicheradresse durch das Indexregister 1, 2, 3 oder 4 additiv verändert.

Beispiele:

DV ABC 320375 Dividiere durch Inhalt der Adresse NN375  
 DV DEF -320624 " " " " " 0624

DVY ABC 330375 Dividiere durch Inhalt der in NN375 stehenden Adresse  
 DVY DEF -330624 " " " " " 0624 " "

DV 4 ABC 324375 Dividiere durch Inhalt der Adresse (NN375 + IXR4)  
 DVY4 ABC 334375 Dividiere durch Inhalt der in NN375 stehenden, um (IXR4) veränderten Adresse

Bei MINCAL 4E nur, wenn für Befehlsgruppe 3 ausgerüstet

3-17	/ / /	MINCAL 4E/4N PROGRAMMIERUNG	HEINRICH DIETZ INDUSTRIE ELEKTRONIK
------	-------	--------------------------------	--

VBL	*34XXXX	DEZIMAL-BINAER LINKSKOMMA
VBR	*35XXXX	DEZIMAL-BINAER RECHTSKOMMA
VDL	*36XXXX	BINAER-DEZIMAL LINKSKOMMA
VDR	*37XXXX	BINAER-DEZIMAL RECHTSKOMMA

Bedeutung: Der BCD-codierte, dezimale Inhalt des W-Registers wird in eine Binärzahl umgeformt (VBL, VBR).

Der binäre Inhalt des W-Registers wird in eine BCD-codierte Dezimalzahl umgeformt (VDL, VDR).

Die Zahlen werden als echte Brüche (VBL, VDL) oder als ganze Zahlen (VBR, VDR) aufgefaßt.

Das Vorzeichen des Rechenwerks bleibt erhalten.

Die größtmögliche natürliche Ganzzahl ist 39999; der größtmögliche natürliche Bruch ist 0.99998.

Rechengang:	X vorher	Operation	W nachher
	240000	VBL	400000 (0.5)
	000020	VBR	000012 (10)
	400000	VDL	240000 (0.5)
	000012	VDR	000007 (10)

Bemerkung: Das Vorzeichen \* und die 4 Oktalstellen XXXX sind ohne Bedeutung. Positive Vorzeichen und 0000 für XXXX werden empfohlen.

Der Inhalt des X-Registers wird verändert, da das X-Register als Zwischenspeicher benutzt wird.

Beispiele:

VBL	340000	Binärumformung Bruch
VBR	350000	" Ganzzahl
VDL	360000	Dezimalumformung Bruch
VDR	370000	" Ganzzahl

Bei MINCAL 4E nur, wenn für Befehlsguppe 3 ausgerüstet

3-18	/	/	MINCAL 4E/4N	HEINRICH DIETZ
			PROGRAMMIERUNG	

BR \*40SAAA

VERZWEIGT

BRY \*41SAAA

VERZWEIGT INDIREKT

**Bedeutung:** Das Programm verzweigt zum adressierten Platz. Ist in der Ergänzung S der Instruktion anstelle von 0 eine der Ziffern 1...7 angegeben, so findet die Verzweigung nur dann statt, wenn der betreffende Sensor-Eingang erregt bzw. der betreffende Sensor-Schalter betätigt ist, andernfalls geht das Programm mit der folgenden Instruktion weiter.

**Adresse:** Bei direkter Adressierung wird zur programmierten Adresse AAA verzweigt, die zur jeweiligen Seite gehört; ist das Vorzeichen \* der Instruktion negativ, zur Adresse AAA der Seite 0.

Bei indirekter Adressierung wird zu einem Platz verzweigt, dessen Adresse sich in AAA befindet, wobei AAA je nach positivem oder negativem Vorzeichen \* der Instruktion in der jeweiligen Seite oder in Seite 0 liegt.

**Beispiele:**

BR ABC 400375 Verzweige nach NN375, unbedingt  
 BR DEF -400624 " " 00624, "

BRY ABC 410375 Verzweige nach in NN375 stehender Adresse, unbedingt  
 BRY DEF -410624 " " 00624 " " , "

BR 4 ABC 404375 Verzweige nach NN375, wenn Sensor 4 erregt  
 BRY4 ABC 414375 Verzweige nach in NN375 stehender Adresse, wenn Sensor 4 erregt

BZ	*42RAAA	VERZWEIGE WENN NULL
BZY	*43RAAA	VERZWEIGE WENN NULL INDIREKT
BP	*44RAAA	VERZWEIGE WENN PLUS
BPY	*45RAAA	VERZWEIGE WENN PLUS INDIREKT
BM	*46RAAA	VERZWEIGE WENN MINUS
BMY	*47RAAA	VERZWEIGE WENN MINUS INDIREKT

**Bedeutung:** Das Programm verzweigt zum adressierten Platz, wenn der Inhalt des in der Ergänzung angegebenen Registers R dem Betrage nach Null ist (BZ, BZY), positives Vorzeichen (BP, BPY) oder negatives Vorzeichen (BM, BMY) hat; andernfalls geht das Programm mit der folgenden Instruktion weiter.

**Adresse:** Bei direkter Adressierung wird zur programmierten Adresse AAA verzweigt, die zur jeweiligen Seite gehört; ist das Vorzeichen \* der Instruktion negativ, zur Adresse AAA der Seite Ø.

Bei indirekter Adressierung wird zu einem Platz verzweigt, dessen Adresse sich in AAA befindet, wobei AAA je nach positivem oder negativem Vorzeichen \* der Instruktion in der jeweiligen Seite oder in Seite Ø liegt.

**Beispiele:**

BZ 1 ABC	401375	Verzweige nach NN375, wenn IXR1 Null
BZ 2 ABC	402375	" " " " IXR2 "
BZ 3 ABC	403375	" " " " IXR3 "
BZ 4 ABC	404375	" " " " IXR4 "
BZ W ABC	406375	Verzweige nach NN375, wenn W-Register Null
BZ X ABC	407375	" " " " X-Register "
BZ W DEF	-406624	Verzweige nach ØØ624, wenn W-Register Null
BZYW ABC	416375	Verzweige nach in NN375 stehender Adresse, wenn W = Null
BZYW DEF	-416624	" " " ØØ624 " " "
BP W ABC	426375	Verzweige nach NN375, wenn W = Plus
BPYW ABC	436375	" " in NN375 stehender Adresse, wenn W = Plus
BM W ABC	446375	" " NN375, wenn W = Minus
BMYW ABC	456375	" " in NN375 stehender Adresse, wenn W = Minus

CS \*50RAAA

UNTERPROGRAMM

CSY \*51RAAA

UNTERPROGRAMM INDIREKT

**Bedeutung:** Das Programm verzweigt zum adressierten Platz, an dem ein Unterprogramm beginnt. Die um 1 erhöhte Adresse der Instruktion (Rückkehradresse) wird in dem in der Ergänzung angegebenen Register R gespeichert.

**Adresse:** Bei direkter Adressierung wird zur programmierten Adresse AAA verzweigt, die zur jeweiligen Seite gehört; ist das Vorzeichen \* der Instruktion negativ, zur Adresse AAA der Seite Ø.

Bei indirekter Adressierung wird zu einem Platz verzweigt, dessen Adresse sich in AAA befindet, wobei AAA je nach positivem oder negativem Vorzeichen \* der Instruktion in der jeweiligen Seite oder in Seite Ø liegt.

**Bemerkung:** Als Register R ist vorzugsweise eines der 4 Indexregister zu benutzen; hierzu wird eine der Ziffern 1...4 in der Ergänzung angegeben; eventuell kommt auch das X- oder das N-Register infrage. Angabe von Ø (Startadresse) oder N (Programmstand) ist verboten!

Am Ende des Unterprogramms wird zum Hauptprogramm verzweigt, indem das N-Register (Programmstand) mit dem Inhalt des Indexregisters (Rückkehradresse) geladen wird; dies geschieht mit 1 Abschluß-Instruktion, z.B.:

TRR3 N. 153005 Rückkehradresse aus IXR3 nach N-Register

**Beispiele:**

CS 1 ABC 501375	Unterprogramm-Sprung nach NN375, Rückkehradresse in IXR1
CS 2 ABC 502375	" " " " " IXR2
CS 3 ABC 503375	" " " " " IXR3
CS 4 ABC 504375	" " " " " IXR4

CS 1 DEF -501624 Unterprogramm-Sprung nach Ø0624, Rückkehradresse in IXR1

CSY1 ABC 511375	Unterprogramm-Sprung nach in NN375 stehender Adresse
CSY1 DEF -511624	" " " " Ø0624 " "
(beides mit Rückkehradresse in IXR1)	

NCP	*53XXLX	NIVEAUWECHSEL
HLT	*54XXXX	HALT
HCP	*55XXLX	HALT MIT NIVEAUWECHSEL
END	*56XXXX	ENDE
ECP	*57XXLX	ENDE MIT NIVEAUWECHSEL

Bedeutung: Das in der Oktalstelle L der Instruktion angegebene Niveau wird gestartet (NCP).

Das Programm im laufenden Niveau wird angehalten; bei einem neuen Start wird es mit der nächsten Instruktion weitergehen (HLT). Mit dem Anhalten kann der Start eines anderen, in L angegebenen Niveaus verbunden werden (HCP).

Das Programm im laufenden Niveau wird angehalten; bei einem neuen Start wird es mit der in Register Ø enthaltenen Adresse (Startadresse) weitergehen (END). Mit dem Endbefehl kann der Start eines anderen, in L angegebenen Niveaus verbunden werden (ECP).

Bemerkung: Das Vorzeichen \* und die 2 + 1 Oktalstellen XX,X sind ohne Bedeutung. Positives Vorzeichen und 00,Ø für XX,X werden empfohlen.

Beispiele:

NCP 1	530010	Starte Niveau 1
HLT	540000	Halt
HCP 7	550070	Halt, starte Niveau 7
END	560000	Ende
ECP Ø	570000	Ende, starte Niveau Ø

GB

\*60XXXX

UEBERNEHMEN PUFFER

**Bedeutung:** Der aus einem Eingabevorgang stammende Inhalt des Datenregisters im Puffer wird, einschließlich Vorzeichen, in das W-Register übernommen, das vorher gelöscht wurde.

**Puffer:** Beim MINCAL 4N wird der niveau-zugehörige Puffer angesprochen; ist das Vorzeichen \* der Instruktion negativ, so wird jedoch der Puffer Ø angesprochen, unabhängig vom Niveau, in dem die Operation abläuft.

**Bemerkung:** Die 4 Oktalstellen XXXX sind ohne Bedeutung; die Angabe von 0000 wird hierfür empfohlen.

Ein eventuell anstehender Niveau-Start aus dem Puffer wird mit dieser Operation beendet. Wird dieser Befehl auf einen Puffer angewendet, dessen Arbeitszyklus noch läuft, so hält der Rechner vor Ausführen der Operation an und bleibt bis zum Ende des Pufferzyklus blockiert.

**Beispiele:**

GB 600000 Übernimm aus Puffer

GB - -600000 Übernimm aus Puffer Ø MINCAL 4N

IBG

\*611AAA

EINGABE IN PUFFER, UEBERNEHMEN

**Bedeutung:** Die Adresse wird in das Adressregister des Puffers übertragen, und ein Eingabezyklus wird gestartet.

Der Rechner hält innerhalb der Operation bis zum Ende des Pufferzyklus' an; dann wird der neue Inhalt des Puffer-Datenregisters ins W-Register übernommen und das Programm fortgesetzt. Für die Dauer des Pufferzyklus' ist der Rechner blockiert.

**Eingabemodus:** Die Eingabe der externen Information in den Puffer erfolgt 4-bit-weise parallel/stellen-  
seriell mit der im Puffer fest eingestellten Geschwindigkeit.

**Puffer:** Beim MINCAL 4N wird der niveau-zugehörige Puffer angesprochen; ist das Vorzeichen \* der Instruktion negativ, so wird jedoch der Puffer Ø angesprochen, unabhängig vom Niveau, in dem die Operation abläuft.

**Adresse:** Der externe Informationsgeber wird durch die oktale Adresse AAA identifiziert und durch den Adresskanal des angesprochenen Puffers ausgewählt. Ist in der Ergänzungsstelle I statt Ø eine der Ziffern 1 bis 4 angegeben, so wird die Adresse AAA durch eines der Indexregister 1...4 additiv verändert.

**Bemerkung:** Ein eventuell anstehender Niveau-Start aus dem Puffer wird mit dieser Operation beendet. Wird dieser Befehl auf einen Puffer angewendet, dessen Arbeitszyklus noch läuft, so hält der Rechner vor Ausführen der Operation an und bleibt bis zum Ende des Pufferzyklus' blockiert.

**Beispiele:**

IBG 256 61Ø256 Eingabe aus Adresse 256  
IBG3 256 613256 " " " (256 + IXR3)

IBG -256 -61Ø256 Eingabe aus Adresse 256 /Puffer Ø MINCAL 4N  
IBG3-256 -613256 " " " (256 + IXR3)/Puffer Ø MINCAL 4N

Nur gültig in Verbindung mit Puffern vom E-Typ

IBR

\*621AAA

## EINGABE IN PUFFER

**Bedeutung:** Die Adresse wird in das Adreßregister des Puffers übertragen, und ein Eingabezyklus wird gestartet.

Das Programm wird sofort mit der nächsten Instruktion fortgesetzt, d.h. Rechner und Puffer arbeiten gleichzeitig und unabhängig voneinander.

Die vom Puffer abgefragte externe Information muß später mit einem Befehl GB in das X-Register übernommen werden; wird GE vom Programm noch im Arbeitszyklus des Puffers erreicht, hält das Programm an, und der Rechner bleibt bis zum Ende des Zyklus' blockiert.

**Eingabemodus:** Die Eingabe der externen Information in den Puffer erfolgt 4-bit-weise parallele/stellen- seriell mit der im Puffer fest eingestellten Geschwindigkeit.

**Puffer:** Beim MINCAL 4N wird der niveau-zugehörige Puffer angesprochen; ist das Vorzeichen \* der Instruktion negativ, so wird jedoch der Puffer # angesprochen, unabhängig vom Niveau, in dem die Operation abläuft.

**Adresse:** Der externe Informationsgeber wird durch die oktale Adresse AAA identifiziert und durch den Adreßkanal des angesprochenen Puffers ausgewählt. Ist in der Ergänzungsstelle I statt # eine der Ziffern 1 bis 4 angegeben, so wird die Adresse AAA durch eines der Indexregister 1...4 additiv verändert.

**Bemerkung:** Ein eventuell anstehender Niveau-Start aus dem Puffer wird mit dieser Operation beendet. Wird dieser Befehl auf einen Puffer angewendet, dessen Arbeitszyklus noch läuft, so hält der Rechner vor Ausführung der Operation an und bleibt bis zum Ende des Pufferzyklus' blockiert.

**Beispiele:**

IBR 256 620256 Eingabe aus Adresse 256  
IBR3 256 623256 " " " (256 + IXR3)

IBR -256 -620256 Eingabe aus Adresse 256 /Puffer # MINCAL 4N  
IBR3-256 -623256 " " " (256 + IXR3) / " " " "

Nur gültig in Verbindung mit Puffern vom E-Typ

IBP

\*631AAA

EINGABE IN PUFFER MIT NIVEAU-START

**Bedeutung:** Die Adresse wird in das Adressregister des Puffers übertragen, und ein Eingabezyklus wird gestartet.

Das Programm wird sofort mit der nächsten Instruktion fortgesetzt, d.h. Rechner und Puffer arbeiten gleichzeitig.

Im Programm ist etwas später, jedenfalls vor dem voraussichtlichen Ende des Pufferzyklus', ein Befehl HAL vorzusehen, mit dem das Niveau ausgeschaltet und das Programm angehalten wird. Am Ende des Pufferzyklus' erfolgt automatisch ein Start desselben Niveaus, mit dem das Programm fortgesetzt wird. Der Niveau-Start dauert an bis zum nächsten Befehl, der sich auf diesen Puffer bezieht, und ist mit dem niveau-gebundenen Erregung des Sensors 1 verbunden.

Die vom Puffer abgefragte externe Information ist mit einem Befehl GB zu übernehmen.

**Eingabemodus:** Die Eingabe der externen Information in den Puffer erfolgt 4-bit-weise parallel/stellen seriell mit der im Puffer fest eingestellten Geschwindigkeit.

**Puffer:** Beim MINCAL 4N wird der niveau-zugehörige Puffer angesprochen. Ist das Vorzeichen \* der Instruktion negativ, so wird jedoch der Puffer Ø angesprochen, unabhängig vom Niveau, in dem die Operation abläuft; zur Identifizierung wird dann mit dem Niveau-Start der niveau-gebundene Sensor Z erregt.

**Adresse:** Der externe Informationsgeber wird durch die oktale Adresse AAA identifiziert und durch den Adresskanal des angesprochenen Puffers ausgewählt. Ist in der Ergänzungsstelle I statt Ø eine der Ziffern 1...4 angegeben, so wird die Adresse AAA durch eines der Indexregister 1...4 additiv verändert.

**Bemerkung:** Ein eventuell anstehender Niveau-Start aus dem Puffer wird mit dieser Operation beendet. Wird dieser Befehl auf einen Puffer angewendet, dessen Arbeitszyklus noch läuft, so hält der Rechner vor Ausführung der Operation an und bleibt bis zum Ende des Pufferzyklus' blockiert.

**Beispiele:**

IBP 256 630256 Eingabe aus Adresse 256  
IBP3 256 633256 " " " (256 + IXR3)

IBP -256 -630256 Eingabe aus Adresse 256 /Puffer Ø MINCAL 4N  
IBP3-256 -633256 " " " (256 + IXR3) / " " " "

Nur gültig in Verbindung mit Puffern vom E-Typ

OBW

\*651AAA

## AUSGABE UEBER PUFFER, WARTEN

Bedeutung: Der Inhalt des W-Registers wird in das Datenregister und die Adresse in das Adreßregister des Puffers übertragen, und ein Ausgabezyklus wird gestartet.  
Der Rechner hält innerhalb der Operation bis zum Ende des Pufferzyklus' an; dann wird das Programm fortgesetzt. Für die Dauer des Pufferzyklus' ist der Rechner blockiert.

**Ausgabemodus:** Die Ausgabe der Information aus dem Puffer erfolgt 4-bitweise parallel/stellenseriel mit der im Puffer fest eingestellten Geschwindigkeit.

**Puffer:** Beim **MINCL 4N** wird der niveau-zugehörige Puffer angesprochen; ist das Vorzeichen \* der Instruktion negativ, so wird jedoch der Puffer  $\beta$  angesprochen, unabhängig vom Niveau, in dem die Operation abläuft.

**Adresse:** Der externe Informationsgeber wird durch die oktale Adresse AAA identifiziert und durch den Adresskanal des angesprochenen Puffers ausgewählt. Ist in der Ergänzungsstelle I statt Ø eine der Ziffern 1 bis 4 angegeben, so wird die Adresse AAA durch eines der Indexregister 1...4 additiv verändert.

**Bemerkung:** Ein eventuell anstehender Niveau-Start aus dem Puffer wird mit dieser Operation beendet. Wird dieser Befehl auf einen Puffer angewendet, dessen Arbeitszyklus noch läuft, so hält der Rechner vor Ausführen der Operation an und bleibt bis zum Ende des Pufferzyklus blockiert.

### Beispiele:

OBW 256 650256 Ausgabe in Adresse 256  
OBW3 256 653256 " " " (256 + IX83)

OBW -256 -650256 Ausgabe in Adresse 256 /Puffer # MINCAL 4N  
 OBW3-256 -653256 " " " (256 + IXR3)/ " " " "

Nur gültig in Verbindung mit Puffern vom E-Typ

1

/	
/	
/	

MINCIAI BEGAN

## PROGRAMMIERUNG

# HEINRICH DIETZ

## INDUSTRIE-ELEKTRONIK

OBR

\*661AAA

AUSGABE UEBER PUFFER

**Bedeutung:** Der Inhalt des W-Registers wird in das Datenregister und die Adresse in das Adreßregister des Puffers übertragen, und ein Ausgabezyklus wird gestartet.

Das Programm wird sofort mit der nächsten Instruktion fortgesetzt, d.h. Rechner und Puffer arbeiten gleichzeitig und unabhängig voneinander.

**Ausgabemodus:** Die Ausgabe der Information aus dem Puffer erfolgt 4-bit-weise parallel/stellenseriel mit der im Puffer fest eingestellten Geschwindigkeit.

**Puffer:** Beim MINCAL 4N wird der niveau-zugehörige Puffer angesprochen; ist das Vorzeichen \* der Instruktion negativ, so wird jedoch der Puffer # angesprochen, unabhängig vom Niveau, in dem die Operation abläuft.

**Adresse:** Der externe Informationsgeber wird durch die oktale Adresse AAA identifiziert und durch den Adreßkanal des angesprochenen Puffers angewählt. Ist in der Ergänzungsstelle I statt # eine der Ziffern 1 bis 4 angegeben, so wird die Adresse AAA durch eines der Indexregister 1...4 additiv verändert.

**Bemerkung:** Ein eventuell anstehender Niveau-Start aus dem Puffer wird mit dieser Operation beendet. Wird dieser Befehl auf einen Puffer angewendet, dessen Arbeitszyklus noch läuft, so hält der Rechner vor Ausführen der Operation an und bleibt bis zum Ende des Pufferzyklus blockiert.

**Beispiele:**

OBR 256 66#256 Ausgabe im Adresse 256  
CBR3 256 663256 " " " (256 + IXR3)

OBR -256 -66#256 Ausgabe im Adresse 256 /Puffer # MINCAL 4N  
OBR3-256 -663256 " " " (256 + IXR3) / " " " "

Nur gültig in Verbindung mit Puffern vom E-Typ

OBP

\*671AAA

AUSGABE UEBER PUFFER MIT NIVEAU-START

**Bedeutung:** Der Inhalt des W-Registers wird in das Datenregister und die Adresse in das Adressregister des Puffers übertragen, und ein Ausgabezyklus wird gestartet.

Das Programm wird sofort mit der nächsten Instruktion fortgesetzt, d.h. Rechner und Puffer arbeiten gleichzeitig und unabhängig voneinander.

Im Programm ist etwas später, jedenfalls vor dem voraussichtlichen Ende des Pufferzyklus', ein Befehl HLT vorzusehen, mit dem das Niveau ausgeschaltet und das Programm anhalten wird. Am Ende des Pufferzyklus' erfolgt automatisch ein Start desselben Niveaus, mit dem das Programm fortgesetzt wird. Der Niveau-Start dauert an bis zum nächsten Befehl, der sich auf diesen Puffer bezieht, und ist mit der niveau-gebundenen Erregung des Sensors 1 verbunden.

**Ausgabemodus:** Die Ausgabe der Information aus dem Puffer erfolgt 4-bit-weise parallel/stellenseiell mit der im Puffer fest eingestellten Geschwindigkeit.

**Puffer:** Beim MINCAL 4M wird der niveau-zugehörige Puffer angesprochen. Ist das Vorzeichen \* der Instruktion negativ, so wird jedoch der Puffer Ø angesprochen, unabhängig vom Niveau, in dem die Operation abläuft. Zur Identifizierung wird dann mit dem Niveau-Start der niveau-gebundene Sensor 2 erregt.

**Adresse:** Der externe Informationsgeber wird durch die oktale Adresse AAA identifiziert und durch den Adresskanal des angesprochenen Puffers ausgewählt. Ist in der Ergänzungsstelle I statt Ø eine der Ziffern 1 bis 4 angegeben, so wird die Adresse AAA durch eines der Indexregister 1...4 additiv verändert.

**Bemerkung:** Ein eventuell anstehender Niveau-Start aus dem Puffer wird mit dieser Operation beendet. Wird dieser Befehl auf einen Puffer angewendet, dessen Arbeitszyklus noch läuft, so hält der Rechner vor Ausführen der Operation an und bleibt bis zum Ende des Pufferzyklus' blockiert.

**Beispiele:**

OBP 256 670256 Ausgabe in Adresse 256  
OBP3 256 673256 " " " (256 + IXR3)

OBP -256 -670256 Ausgabe in Adresse 256 /Puffer Ø MINCAL 4N  
OBP3-256 -673256 " " " (256 + IXR3) / " " " "

Nur gültig in Verbindung mit Puffern vom E-Typ

KG	*615FFF	MANUELLE EINGABE, UEBERNEHMEN
KR	*625FFF	MANUELLE EINGABE
KP	*635FFF	MANUELLE EINGABE MIT NIVEAU-START

Bedeutung: Die Befehle KG, KR und KP entsprechen den Befehlen IBG, IBR und IBP, beziehen sich jedoch auf das 1. Eingabegerät des Puffers (im allgemeinen die Tastatur einer Schreibmaschine).

Eingabemodus: Die Eingabe der Informationen in den Puffer geschieht in der Reihenfolge Vorzeichen - Datenstellen - Endzeichen je nach dem programmierten Format und mit der Geschwindigkeit des Eingabegerates bzw. -vorganges.

Puffer: Beim MINCAL 4N wird der niveau-zugehörige Puffer angesprochen; ist das Vorzeichen \* der Instruktion negativ, so wird jedoch der Puffer Ø angesprochen, unabhängig vom Niveau, in dem die Operation abläuft.

Format: Die 3 Oktalstellen FFF geben das Format an; sie enthalten Angaben über das Vorzeichen, die Daten-Darstellung, die Anzahl der Datenstellen und das Endzeichen (siehe Formatliste).

Beispiele:

KG	L 5	615000	Eingabe dezimal (Linkskomma) 5-stellig, übernehmen
KR	L 5	625000	" " " "
KP	L 5	635000	" " " " mit Start
KR	L-5X	625101	Eingabe Vorzeichen, dezimal (Linkskomma) 5-stellig, Leertaste
KR	R 5	625200	" dezimal (Rechtskomma) 5-stellig
KR	O 6	625400	" oktal 6-stellig
KR	A 3	625600	" alphanumerisch 3-stellig
KR	A 1	625620	" " 1-stellig
KR	A 3X	625601	" " " 3-stellig, Leertaste
KR	X.	625631	" Leertaste
KG	-L 5	-615000	wie 1. Beispiel, jedoch Puffer Ø (MINCAL 4E)

Nur gültig in Verbindung mit Puffern vom E-Typ

3-30	/	MINCAL 4E/4N	HEINRICH DIETZ
	/	PROGRAMMIERUNG	INDUSTRIE-ELEKTRONIK

RG	*616FFF	LESEN, UEBERNEHMEN
RR	*626FFF	LESEN
RP	*636FFF	LESEN MIT NIVEAU-START

**Bedeutung:** Die Befehle RG, RR und RP entsprechen den Befehlen IBG, IBR und IBP, beziehen sich jedoch auf das 2. Eingabegerät des Puffers (im allgemeinen ein Streifenleser).

**Eingabemodus:** Die Eingabe der Informationen in den Puffer geschieht in der Reihenfolge Vorzeichen - Datenstellen - Endzeichen je nach dem programmierten Format und mit der Geschwindigkeit des Eingabegerätes bzw. -vorganges.

**Puffer:** Beim MINCAL 4N wird der niveau-zugehörige Puffer angesprochen; ist das Vorzeichen \* der Instruktion negativ, so wird jedoch der Puffer Ø angesprochen, unabhängig vom Niveau, in dem die Operation läuft.

**Format:** Die 3 Oktalstellen FFF geben das Format an; sie enthalten Angaben über das Vorzeichen, die Daten-Darstellung, die Anzahl der Datenstellen und das Endzeichen (siehe Formatliste).

**Beispiele:**

RG	L 5	616000	Lies dezimal (Linkskomma) 5-stellig, übernimmt
RR	L 5	626000	" " " "
RP	L 5	636000	" " " " mit Start
RR	L-5X	626101	Lies Vorzeichen, dezimal (Linkskomma) 5-stellig, ZWR-Zeichen
RR	R 5	626200	" dezimal (Rechtskomma) 5-stellig
RR	O 6	626400	" oktal 6-stellig
RR	A 3	626600	" alphanumerisch 3-stellig
RR	A 1	626620	" " 1-stellig
RR	A 3X	626601	" " 3-stellig, ZWR-Zeichen
RR	A 2E	626614	" " 2-stellig, WR-Zeichen
RR	E.	626634	" WR-Zeichen
RG	-L 5	-616000	wie 1. Beispiel, jedoch Puffer Ø (MINCAL 4N)

Nur gültig in Verbindung mit Puffern vom E-Typ

3-31		MINCAL 4E/4N	HEINRICH DIETZ
		PROGRAMMIERUNG	INDUSTRIE-ELEKTRONIK

ZG	*617FFF	EINGABE 3. GERAET, WARTEN
ZR	*627FFF	EINGABE 3. GERAET
ZP	*637FFF	EINGABE 3. GERAET

**Bedeutung:** Die Befehle ZG, ZR und ZP entsprechen den Befehlen IBG, IBR und IBS, beziehen sich jedoch auf ein näher zu spezifizierendes drittes Eingabegerät, das an den Puffer angeschlossen werden kann.

**Eingabemodus:** Die Eingabe der Informationen in den Puffer geschieht in der Reihenfolge Vorzeichen - Datenstellen - Endzeichen je nach dem programmierten Format und mit der Geschwindigkeit des Eingabegerätes bzw. -vorganges.

**Puffer:** Beim MINCAL 4N wird der niveau-zugehörige Puffer angesprochen; ist das Vorzeichen \* der Instruktion negativ, so wird jedoch der Puffer Ø angesprochen, unabhängig vom Niveau, in dem die Operation abläuft.

**Format:** Die 3 Oktalstellen FFF geben das Format an; sie enthalten Angaben über das Vorzeichen, die Daten-Darstellung, die Anzahl der Datenstellen und das Endzeichen (siehe Formatliste).

**Beispiele:**

ZG	L 5	617000	Eingabe dezimal (Linkskomma) 5-stellig, Übernimmt
ZR	L 5	627000	" " " "
ZP	L 5	637000	" " " " mit Start
ZR	L-5X	627101	Eingabe Vorzeichen, dezimal (Linkskomma) 5-stellig, Leertaste
ZR	R 5	627200	" dezimal (Rechtskomma) 5-stellig
ZR	O 6	627400	" oktal 6-stellig
ZR	A 3	627600	" alphanumerisch 3-stellig
ZR	A 1	627620	" " 1-stellig
ZR	A 3X	627601	" " 3-stellig, Leertaste
ZR	A 2E	627614	" " 2-stellig, WR-Zeichen
ZR	E.	627634	" WR-Zeichen
ZG	-L 5	-617000	wie 1. Beispiel, jedoch Puffer Ø (MINCAL 4N)

Nur gültig im Verbindung mit Puffern vom L-Typ

WW	*655FFF	DRUCKEN, WARTEN
WR	*665FFF	DRUCKEN
WP	*675FFF	DRUCKEN MIT NIVEAU-START

**Bedeutung:** Die Befehle WW, WR und WP entsprechen den Befehlen QW, QSR und QBP, beziehen sich jedoch auf das 1. Registriergerät des Puffers (im allgemeinen das Druckwerk einer Schreibmaschine).

**Ausgabemodus:** Die Ausgabe der Information aus dem Puffer geschieht in der Reihenfolge Vorzeichen - Datenstellen - Endzeichen je nach dem programmierten Format und mit der maximalen Geschwindigkeit des Registriergeräts.

**Puffer:** Beim MINCAL 4N wird der niveau-zugehörige Puffer angesprochen; ist das Vorzeichen \* der Instruktion negativ, so wird jedoch der Puffer Ø angesprochen, unabhängig vom Niveau, in dem die Operation abläuft.

**Format:** Die 3 Oktalstellen FFF geben das Format an; sie enthalten Angaben über das Vorzeichen, die Daten-Darstellung, die Anzahl der Datenstellen und das Endzeichen (siehe Formatliste).

**Beispiele :**

WW	L 5	655000	Drucke dezimal (Linkskomma) 5-stellig, warte
WR	L 5	665000	" " " "
WP	L 5	675000	" " " " mit Start
WR	L-5X	665101	Drucke Vorzeichen, dezimal (Linkskomma) 5-stellig, Zwischenraum
WR	R 5	665200	" dezimal (Rechtskomma) 5-stellig
WR	O 6	665400	" oktal 6-stellig
WR	A 3	665600	" alphanumerisch 3-stellig
WR	A 1	665620	" " 1-stellig
WR	A 3X	665601	" " 3-stellig, Zwischenraum
WR	A 2E	665614	" " 2-stellig, Schreibkopfrücklauf
WR	E.	665634	Schreibkopfrücklauf
WW	-L 5	-655000	wie 1. Beispiel, jedoch Puffer Ø (MINCAL 4N)

Nur gültig in Verbindung mit Puffern vom E-Typ

PW	*656FFF	LOCHEN, WARTEN
PR	*666FFF	LOCHEN
PP	*676FFF	LOCHEN MIT NIVEAU-START

Bedeutung: Die Befehle PW, PR, PP entsprechen den Befehlen DBW, DBR und DBP, beziehen sich jedoch auf das 2. Registriergerät des Puffers (im allgemeinen Streifenlocher)

Ausgabemodus: Die Ausgabe der Information aus dem Puffer geschieht in der Reihenfolge Vorzeichen - Datenstellen - Endzeichen je nach dem programmierten Format und mit der maximalen Geschwindigkeit des Registriergeräts.

Puffer: Beim MINCAL 4N wird der niveau-zugehörige Puffer angesprochen; ist das Vorzeichen \* der Instruktion negativ, so wird jedoch der Puffer Ø angesprochen, unabhängig vom Niveau, in dem die Operation abläuft.

Format: Die 3 Oktalstellen FFF geben das Format an; sie enthalten Angaben über das Vorzeichen, die Daten-Darstellung, die Anzahl der Datenstellen und das Endzeichen (siehe Formatliste).

#### Beispiele:

PW	L 5	656000	Loche dezimal (Linkskomma) 5-stellig, warte
PR	L 5	666000	" " " "
PP	L 5	676000	" " " " mit Start
PR	L-5X	666101	Loche Vorzeichen, dezimal (Linkskomma) 5-stellig, ZWR-Zeichen
PR	R 5	666200	" dezimal (Rechtskomma) 5-stellig
PR	O 6	666400	" oktal 6-stellig
PR	A 3	666600	" alphanumerisch 3-stellig
PR	A 1	666620	" " 1-stellig
PR	A 3X	666601	" " 3-stellig, ZWR-Zeichen
PR	A 2E	666614	" " 2-stellig, WR-Zeichen
PR	E.	666634	Loche WR-Zeichen
PW	-L 5	-656000	wie 1. Beispiel, jedoch Puffer Ø (MINCAL 4N)

Nur gültig in Verbindung mit Puffern vom E-Typ

XW	*657FFF	AUSGABE 3. GERAET, WARTEN
XR	*667FFF	AUSGABE 3. GERAET
XP	*677FFF	AUSGABE 3. GERAET MIT NIVEAU-START

**Bedeutung:** Die Befehle XW, XR und XP entsprechen den Befehlen OBW, OBR und OBP, beziehen sich jedoch auf ein näher zu spezifizierendes drittes Registrergerät, das an den Puffer angeschlossen werden kann. Stattdessen kann es sich auch um eine kombinierte, gleichzeitige Ausgabe an das 1. und 2. Gerät (Schreibmaschine und Locher) handeln.

**Ausgabemodus:** Die Ausgabe der Information aus dem Puffer geschieht in der Reihenfolge Vorzeichen - Datenstellen - Endzeichen je nach dem programmierten Format und mit der maximalen Geschwindigkeit des Registrergeräts.

**Puffer:** Beim MINCAL 4N wird der niveau-zugehörige Puffer angesprochen; ist das Vorzeichen \* der Instruktion negativ, so wird jedoch der Puffer Ø angesprochen, unabhängig vom Niveau, in dem die Operation abläuft.

**Format:** Die 3 Oktalstellen FFF geben das Format an; sie enthalten Angaben über das Vorzeichen, die Daten-Darstellung, die Anzahl der Datenstellen und das Endzeichen (siehe Formatliste).

**Beispiele:**

XW	L 5	657000	Ausgabe dezimal (Linkskomma) 5-stellig, warte
XR	L 5	667000	" " " "
XP	L 5	677000	" " " " mit Start
XR	L-5X	667101	Ausgabe Vorzeichen, dezimal (Linkskomma) 5-stellig, Zwischenraum
XR	R 5	667200	" dezimal (Rechtskomma) 5-stellig
XR	O 6	667400	" oktal 6-stellig
XR	A 3	667600	" alphanumerisch 3-stellig
XR	A 1	667620	" " 1-stellig
XR	A 3X	667601	" " 3-stellig, Zwischenraum
XR	A 2E	667614	" " 2-stellig, WR-Zeichen
XR	E .	667634	Ausgabe WR-Zeichen
XW	-L 5	-657000	wie 1. Beispiel, jedoch Puffer Ø (MINCAL 4N)

Nur gültig in Verbindung mit Puffern vom E-Typ

3-35	/	MINCAL 4E/4N	HEINRICH DIETZ
	/		
	/		
PROGRAMMIERUNG		INDUSTRIE-ELEKTRONIK	

Bei Ein/Ausgabebefehlen über Buffer vom E-Typ enthalten die 3 oktalen Adressstellen FFF eine Formatangabe, wenn in der Ergänzung der Instruktion einer der Ziffern 5, 6 oder 7 (entsprechend dem 1., 2. oder 3. Gerät) angegeben ist. Die Stellen haben folgende Bedeutung:

1. Stelle (100er-Adresse): Gibt die Code-Darstellung an und bestimmt, ob ein Vorzeichen berücksichtigt wird (bei Schreibmaschine und Streifenleser wird bei positivem Vorzeichen Leertaste bzw. ZWR benutzt).

0	symbolisch:	L	Dezimal (Linkskomma)
1		L-	" " mit Vorzeichen
2		R	Dezimal (Rechtskomma)
3		R-	" " mit Vorzeichen
4		O	oktal
5		O-	" mit Vorzeichen
6		A	Alphanumerisch
7		A-	" mit Vorzeichen

2. Stelle (10er-Adresse): Bestimmt, wieviel Datenstellen (von links gerechnet) nicht berücksichtigt werden. In symbolischer Schreibweise wird jedoch die Anzahl der berücksichtigten Stellen angegeben. Bei voller Berücksichtigung sind dezimal 5, oktal 6 und alphanumerisch 3 Stellen vorhanden.

0	alle Stellen berücksichtigt
1	1 Stelle unberücksichtigt
2	2 " "
3	3 " "
4	4 " "
5	5 " "

Wird nur das Vor- oder Endzeichen benutzt, so ist die Angabe "alphanumerisch, 3 Stellen unberücksichtigt", zu programmieren.

3. Stelle (1er-Adresse): Gibt Endzeichen oder auch eine Funktion an

0	symbolisch:		Kein Endzeichen
1		X	Leerschritt /ZWR-Zeichen
2		T	Tabulator
3		Z	Zeilenwechsel
4		E	Schreibkopfrücklauf + Zeilenwechsel / "R-Zeichen
5		B	Rückschrift
6		N	Normalfarbe (Schwarz)
7		F	Sonderfarbe (Rot)

hier gültig in Verbindung mit Befehlen vom E-Typ

GX	*70XXXX	UEBERNEHMEN PARALLEL
IXG	*711AAA	EINGABE PARALLEL, UEBERNEHMEN
IXR	*721AAA	EINGABE PARALLEL
IXP	*731AAA	EINGABE PARALLEL MIT NIVEAU-START

Bedeutung: Die am Paralleleingang anstehende externe Information wird ins W-Register übernommen; der Anwahl- und Übernahmevergäng dauert eine Wortzeit (GX).

Die Adresse wird am Adreßausgang angeboten; der Rechner hält in der Operation an und ist blockiert, bis die Bereitstellung der externen Information am Parallelausgang gemeldet wird; dann wird diese ins W-Register übernommen, und das Programm läuft weiter (IXG).

Die Adresse wird innerhalb einer Wortzeit angeboten und in ein externes Adreßregister, das den externen Informationsgeber anwählt, das Programm läuft weiter (IXR). Nach Bereitstellung der Information in einem externen Datenregister kann ein Niveau-Start ausgelöst werden (IXP). Die Aufschaltung des externen Datenregisters auf den Paralleleingang und die Übernahme ins W-Register hat mit einem Befehl GX zu erfolgen.

Adresse: Die aktive externe Adresse wird in den 3 Oktalstellen AAA der Instruktion programmiert. Ist in der Ergänzung I eine der Ziffern 1, 2, 3 oder 4 statt 0 angegeben, so wird die Adresse AAA über das entsprechende Indexregister additiv verändert; es können dann max. 32768 externe Adressen angesprochen werden.

Die Adresse wird am Parallel-Adreßkanal angeboten.

Bemerkung: Das Vorzeichen \* der Instruktion ist ohne Bedeutung, positives Vorzeichen wird empfohlen.

Beispiele:

GX	700000	Übernimmt parallel
IXG 514	710514	Eingabebefehl und Übernahme der externen Adresse 514
IXR 514	720514	Eingabebefehl für externe Adresse 514
IXP 514	730514	" " " " 514 mit Start
IXR2 514	722514	Eingabebefehl für externe Adresse (514 + IXR2)

Nur bei Ausrüstung mit Parallel-Ein/Ausgang

OXW	*751AAA	AUSGABE PARALLEL, WARTEN
OXR	*761AAA	AUSGABE PARALLEL
OXP	*771AAA	AUSGABE PARALLEL MIT NIVEAU-START

**Bedeutung:** Die Adresse wird am Adreßausgang und der Inhalt des W-Registers am Parallelausgang angeboten; der Rechner hält in der Operation an und ist blockiert, bis der externe Empfänger der Information die Übernahme gemeldet hat; dann läuft das Programm weiter (OBW).

Adresse und Inhalt des L-Registers werden innerhalb einer Wortzeit am Parallelausgang angeboten zwecks Ausgabe in den externen Informationsempfänger; das Programm läuft weiter (OBR). Wenn in einen externen Zwischenspeicher ausgegeben wird, der die Information dem adressierten Empfänger übermittelt, kann nach Beendigung dieses Vorgangs ein Niveau-Start ausgelöst werden (UBP).

**Adresse:** Die oktale externe Adresse wird in den 3 Oktalstellen AAA der Instruktion programmiert. Ist in der Ergänzung I eine der Ziffern 1, 2, 3 oder 4 statt Ø angegeben, so wird die Adresse AAA über das entsprechende Indexregister additiv verändert; es können dann max. 32768 externe Adressen angesprochen werden.

**Bemerkung:** Das Vorzeichen \* der Instruktion ist ohne Bedeutung, positives Vorzeichen wird empfohlen.

**Beispiele:**

OXW 514 750514	Ausgabe in externe Adresse 514, warten
OXR 514 760514	" " " " 514
OXP 514 770514	" " " " 514 mit Start
OXR2 514 762514	Ausgabe in externe Adresse (514 + IXR2)

Nur bei Ausrüstung mit Parallel-ein/Ausgang

3-38	/ \	MINCAL 4E/4N	HEINRICH DIETZ
	/ \	PROGRAMMIERUNG	INDUSTRIE-ELEKTRONIK

Symbol	Oktalcode	Bem.	Bedeutung	Raum im umgewandelten Programm
V	*XXXXXX	x	Variablen	1 Speicherplatz
F	*XXXXXX	x	Feld	2...999 Speicherplätze
E	*XXXXXX	x	Element	1 Speicherplatz im Feld
D	*CCCCCC	c	Dezimalzahl	1 Speicherplatz
B	*CCCCCC	c	Binärzahl	1 Speicherplatz
O	*CCCCCC	c	Oktalzahl	1 Speicherplatz
A	*CCCCCC	c	Alph.-Wort	1 Speicherplatz
Y	*ØAAAAAA	a	Indirekte Adresse	1 Speicherplatz
M			Speicheradreß-Zuordnung	-
X			Externe Adreß-Zuordnung	-
C			Programmbeginn	-
S...			Startadressen	(Register Ø)
Z			Programmende	-

Erklärungen:

- \* Negatives Vorzeichen bedeutet:
  - a doppelt indirekte Adresse
  - c negativer Festwert
  - x Variable ist negativ

A Adresse

C Festwert

X Variabler Wert

4-1	/	MINCAL 4E/4N		HEINRICH DIETZ INDUSTRIE-ELEKTRONIK	
		PROGRAMMIERUNG			

V	*XXXXXX	VARIABLE
F	*XXXXXX *XXXXXX	FELD
	***	
E	*XXXXXX	ELEMENT

Bedeutung: Das Symbol V kennzeichnet einen einzelnen Speicherplatz, der vom Programm mit Variablen belegt wird.

Das Symbol F bezeichnet einen Bereich (ein Feld) von mindestens 2, maximal 1000 zusammenhängenden Speicherplätzen, die für Variable reserviert sind. Durch das Symbol E wird ein einzelner Speicherplatz dieses Feldes (ein Element) definiert.

Hinweis: Die Symbole V, F und E sind stets mit Instruktionsmerkmalen zu versehen.

Bei Symbol F wird im Adressmerkmal dezimal angegeben, wieviele Plätze das Feld umfaßt. Im Adressmerkmal des Symbols E wird angegeben, wie groß die Adressdifferenz zum Feldbeginn ist. Die bei F angegebene Zahl ist gleich der Anzahl der Speicherplätze minus 1.

Je Feld können beliebig viele Elemente definiert werden; die symbolischen Anweisungen E müssen jedoch unmittelbar auf das Feldsymbol F folgen.

Der Inhalt von Variablenplätzen kann u.U. als Instruktion aufgefaßt werden; es ist daher zu vermeiden, daß das Programm über diese Plätze läuft.

Beispiele:

ABC	V	Variable
ABC	F 999	Feld mit 1000 Plätzen
DEF	F 7.	Feld mit 8 Plätzen
GHI	E 1.	das 2. Element des Feldes DEF
JKL	E 7.	das 8. Element des Feldes DEF

D	*CCCCCC	DEZIMALZAHL
B	*CCCCCC	BINAERZAHL
O	*CCCCCC	OKTALZAHL
A	*CCCCCC	ALPHA-WORT

**Bedeutung:** Das Symbol D kennzeichnet einen Speicherplatz, der eine Dezimalzahl enthält, die dezimal programmiert wird.

Mit Symbol B ist ein Speicherplatz definiert, der eine Binärzahl enthält, deren Größe als Dezimalzahl programmiert wird; die Umformung erfolgt beim Umwandeln.

Das Symbol O bezeichnet einen Speicherplatz, der eine oktal eingegebene binäre Information enthält.

Das Symbol A definiert einen Speicherplatz mit alphanumerischem Inhalt.

**Hinweis:** Der Festwert wird in den Adreßmerkmal-Spalten programmiert. Bei den Symbolen D und B wird nach Brüchen und Ganzzahlen (Links- und Rechtskomma) unterschieden. Der kleinste Bruch ist 0.00004; die größte Ganzzahl 39999.

Negative Festwerte werden durch Minuszeichen in der Vorzeichenpalte bezeichnet.

Festwert-Symbole sind im allgemeinen mit Instruktions-Merkmal zu versehen.

Festwerte können als Instruktionen aufgefaßt werden; daher ist zu vermeiden, daß das Programm über Festwert-Piäte läuft.

**Beispiele:**

D	32767.	623547	Dezimalzahl 32767
D	-10.	-000200	" 10
D	.75	352000	" 0.75
B	32767.	077777	Binärzahl 32767
B	-10.	-000012	" 10
B	.75	600000	" 0.75
O	017740	017740	Oktalzahl
A	MC4	446304	Alpha-wort

Symbol B bei MINCAL 4E nur, wenn mit Befehlsguppe 3 ausgerüstet.

Y

\*0AAAAA

INDIREKTE ADRESSE

**Bedeutung:** Das Symbol Y kennzeichnet einen Speicherplatz, in dem sich eine indirekte Adresse befindet. Nach der Umwandlung steht in den 5 Oktalstellen AAAAA die oktale effektive Adresse. Das Instruktions-Merkmal enthält die Adresse zu einem Befehl mit indirekter Adressierung; im Adressmerkmal steht die effektive Adresse.

**Hinweis:** Das Symbol ` ist stets mit Instruktions- und Adressmerkmal zu versehen.

Indirekte Adressen können u.U. als Instruktion aufgefaßt werden; es ist daher zu vermeiden, daß das Programm über Y-Plätze läuft.

**Bemerkung:** Ist das Vorzeichen \* negativ, so handelt es sich um eine doppelt indirekte Adresse; der durch sie bezeichnete Speicherplatz enthält wiederum eine indirekte Adresse.

**Beispiele:**

ABC Y DEF      0AAAAA      Indirekte Adresse (DEF = effektive Adresse)

ABC Y -DEF      -0AAAAA      Doppelt indirekte Adresse (DEF = 2, indirekte Adresse)

M

SPEICHERADRESS-ZUORDNUNG

X

EXTERNE ADRESS-ZUORDNUNG

**Bedeutung:** Die Definition M ordnet das symbolische Adressmerkmal einer speicherbezogenen Instruktion einem festen Platz in Seite 0 des Kernspeichers zu.

Die Definition X ordnet das symbolische Adressmerkmal einer Ein/Ausgabe-Instruktion einer oktalen, externen Adresse zu.

**Hinweis:** Die Symbole M und X sind stets mit einem Instruktionsmerkmal zu versehen, das die festzulegende symbolische Adresse enthält, sowie einem Adressmerkmal mit der oktalen Adresse.

Die Definition M ist dann erforderlich, wenn ein bestimmter Speicherplatz in Seite 0 (z.B. eines der festen Register) angesprochen werden soll und dessen Adresse zunächst symbolisch angegeben ist.

Die Definition X ist notig, wenn externe Adressen in Ein/Ausgabe-Instruktionen nicht oktal, sondern symbolisch angegeben sind. Die oktale Adresse bezieht sich bei Befehlen der Gruppe 7 auf den Adresskanal des Parallel-Ein/Ausgangs; bei Befehlen der Gruppe 6 ist beim MINCAL 4N der Adresskanal des niveau-zugehörigen Puffers oder des Puffers 0 gemeint (je nach Befehl), beim MINCAL 4E der Adresskanal des einzigen dort eingebauten Puffers.

**Bemerkung:** Den Definitionen M und X entsprechen keine Speicherplätze im umgewandelten Programm. Sie können an beliebigen Stellen des Programms stehen.

**Beispiele:**

ABC M -107

Merkmal ABC hat die Speicheradresse 00107

XYZ X 514

Merkmal XYZ entspricht der externen Adresse 514

C	PROGRAMMBEGINN
S0	STARTADRESSE NIVEAU 0
...	...
S7	STARTADRESSE NIVEAU 7

Bedeutung: Das Symbol C steht als erste Anweisung am Anfang jedes symbolischen Programms; es definiert die Speichergroße.

Darauf folgen die Definitionen S0 und S1 (MINCAL 4E) bzw. S0...S7 (MINCAL 4N) zur Bestimmung der Startadressen.

Das Symbol Z schließt ein symbolisch geschriebenes Programm als letzte Anweisung ab.

Hinweis: In der Adreßmerkmal-Spalte zu C ist die installierte Speicherkapazität anzugeben, z.B.

04	für	4096 Worte (4K)
08	"	8192 " (8K)
...		
32	"	32768 " (32K)

Zu den Symbolen S0...S7 sind als Adreßmerkmal die Instruktions-Merkmale anzugeben, bei denen das Programm in den betreffenden Niveaus beginnt. Bei der Umwandlung werden hierdurch die Register 0 mit Startadressen belegt.

Bemerkung: Den Definitionen C, S... und Z entsprechen keine Speicherplätze im umgewandelten Programm.

Beispiele:

C	12	Speicherkapazität 12K
S0	ABC	Niveau 0 beginnt mit ABC
S1	DEF	" 1 " " DEF
Z		Programmende

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

			18 bit-Hilfsblatt	HEINRICH DIETZ INDUSTRIE-ELEKTRONIK

Komm. Nr.

PROGR. NR.

MASCH.-CODE

Z K Z	SORTIER- BEGRIFF	INSTR- MERK- MAL	BEFEHL BEGRIFF	ADRESS- MERKMA L	Z K Z	BEMERKUNGEN																																											
						0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
2	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	
3	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49		
4	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49			
5	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49				
6	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49					
7	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49						
8	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49							
9	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49								

MC 4 E / N

## INSTRUKTIONSLISTE

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

## BEARBEITER

# HEINRICH DIETZ

## INDUSTRIE-ELEKTRONIK

Komm. Nr.

PROGR. NR.

MC 4 E / N

## INSTRUKTIONSLISTE

**BEARBEITER:**

# HEINRICH DIETZ INDUSTRIE-ELEKTRONIK

Komm. Nr.

PROGR. NR.

MASCH. - CODE									
Z K		Z		Z		Z		Z	
INSTR- MERK- MAL	BEFEHL	ADRES- MERKMA- L	BEFEHL	ADR.	ADR.	BEFEHL	ADR.	BEFEHL	ADR.
1	2	3	4	5	6	7	8	9	10
1	1	0	1	1	1	2	3	4	5
1	2	2	3	4	5	6	7	8	9
1	3	3	4	5	6	7	8	9	10
1	4	4	5	6	7	8	9	10	11
1	5	5	6	7	8	9	10	11	12
1	6	6	7	8	9	10	11	12	13
1	7	7	8	9	10	11	12	13	14
1	8	8	9	10	11	12	13	14	15
1	9	9	10	11	12	13	14	15	16
1	10	10	11	12	13	14	15	16	17
1	11	11	12	13	14	15	16	17	18
1	12	12	13	14	15	16	17	18	19
1	13	13	14	15	16	17	18	19	20
1	14	14	15	16	17	18	19	20	21
1	15	15	16	17	18	19	20	21	22
1	16	16	17	18	19	20	21	22	23
1	17	17	18	19	20	21	22	23	24
1	18	18	19	20	21	22	23	24	25
1	19	19	20	21	22	23	24	25	26
1	20	20	21	22	23	24	25	26	27
1	21	21	22	23	24	25	26	27	28
1	22	22	23	24	25	26	27	28	29
1	23	23	24	25	26	27	28	29	30
1	24	24	25	26	27	28	29	30	31
1	25	25	26	27	28	29	30	31	32
1	26	26	27	28	29	30	31	32	33
1	27	27	28	29	30	31	32	33	34
1	28	28	29	30	31	32	33	34	35
1	29	29	30	31	32	33	34	35	36
1	30	30	31	32	33	34	35	36	37
1	31	31	32	33	34	35	36	37	38
1	32	32	33	34	35	36	37	38	39
1	33	33	34	35	36	37	38	39	40
1	34	34	35	36	37	38	39	40	41
1	35	35	36	37	38	39	40	41	42
1	36	36	37	38	39	40	41	42	43
1	37	37	38	39	40	41	42	43	44
1	38	38	39	40	41	42	43	44	45
1	39	39	40	41	42	43	44	45	46
1	40	40	41	42	43	44	45	46	47
1	41	41	42	43	44	45	46	47	48
1	42	42	43	44	45	46	47	48	49

MC 4 E / N

## INSTRUKTIONSLISTE

110

**BEARBEITER:**

# HEINRICH DIETZ INDUSTRIE-ELEKTRONIK

Komm. Nr.

PROGR. NR.

MC 4 E / N

## INSTRUKTIONSLISTE

**BEARBEITER:**

HEINRICH DIETZ  
INDUSTRIE-ELEKTRONIK

Komm. Nr.

PROGR. NR.

Komm. Nr.

PROGR. NR.

MC 4 E / N

# INSTRUKTIONSLISTE

**BEARBEITER:**

# HEINRICH DIETZ

## INDUSTRIE-ELEKTRONIK

Komm. Nr.

PROGR. NR.

MASCH.-CODE

MC 4 E / N

## INSTRUKTIONSLISTE

### BEARBEITER:

HEINRICH DIETZ  
INDUSTRIE-ELEKTRONIK

Komm. Nr.

PROGR. NR.

MASCH.-CODE

MC 4 E / N

## INSTRUKTIONSLISTE

## BEARBEITER

# HEINRICH DIETZ

## INDUSTRIE-ELEKTRONIK

Komm. Nr.

PROGR. NR.

MASCH. - CODE

BEMERKUNGEN

Z	K	SORTIER - BEGRIFF	INSTR - MERK - MAL	BEFEHL	Z	ADRES - MERKMA L	Z	ADR.
1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9
0								
1								
2								
3								
4								
5								
6								
7								
0								
1								
2								
3								
4								
5								
6								
7								
0								
1								
2								
3								
4								
5								
6								
7								

MC 4 E / N

INSTRUKTIONSLISTE

BEARBEITER:

HEINRICH DIETZ

INDUSTRIE-ELEKTRONIK

Komm. Nr.

PROGR. NR.