

Programmier-Handbuch
Datenverarbeitungssystem
820/15, 820/25, 820/35

Gesamtliste der Befehle, G
Liste der Befehle 820/15, L
Beschreibung der Befehle, B
Systembeschreibung S
Hochleistungsdrucker HD

NIXDORF
COMPUTER

Bedienungsanleitung für NIXDORF-Assembler Serie 820

NIXDORF
COMPUTER

					2.9	2.14	1.1	1.6	I F	grün	
C	7	8	9	2.2	2.4	2.8	2.13	1.2	1.7	II	rot
	4	5	6	←	↶	2.7	2.12	1.3	1.8	III	oran- ge
	1	2	3	2.1	2.3	2.6	2.11	1.4	1.9	IV	gelb
	0	00	,	→	↷	2.5	2.10			V	weiß

Tastenfunktionen

- 1.1 gesetzt Stop-Taste bewirkt „STOP“ der LK-Eingabe, Druck der Protokolle „Symbol Table“ und „Undefined Symbol“. Zusammen mit Taste 2.2 Einzelschrittverarbeitung. In dieser STOP-Stellung neue Programmwahl möglich.
- 1.2 „ Keine Ausgabe der Objektkarten bzw. Objektstreifen.
- 1.3 „ Kein Ausdruck des Maschinen-Codes im Protokoll.
- 1.4 „ Keine Prüfung der Folge-Nummer.
- 1.6 „ Keine Auflistung des symbolischen Programms.
- 1.9 „ Stanzen des Objekt-LS an Stelle von Objekt-LK
- 2.2 „ Start der Programme.
- 2.9 „ Auslösung der Programmwahl.
- F „ Löschen der orangefarbenen Lampe.
- C „ Löschen der roten Lampe.

ENTWICKLUNGSBEREICHE ■ BASIS-SOFTWARE/FIRMWARE

Bedienungsanleitung

Taste Bemerkung

V Taste V setzen und Quelldeck einlegen

C,WZ, Einschalten

Zeilenschaltung
1.1 1zeilig
1.2 4zeilig

1 ASM-PASS I

2 ASM-PASS II

3 ASM-PASS I
ADDITIONS

4 SYMBOL
TABLE

2.9 Programmwahl

2.2 Programm-Start

Fehlerschlüssel

- E 1 Spalte 1 ungleich Blank, Stern oder Gleichheitszeichen
- E 2 Spalte 2 ungleich Buchstabe
- E 3 Name größer als 6 Zeichen
- E 4 Unzulässiges Zeichen (ALC-Code größer als 3.15)
- E 5 Unzulässiger Name (Reservierte Symbole)
- E 6 Doppeldefinition (Name mehrfach definiert)
- E 7 Fehlerhafte Zahl oder Wert größer als 262 143
- E 8 1. Durchlauf: im Befehlsbereich einer Wertzuweisung auftretender Name vorher nicht definiert
2. Durchlauf: Im Befehlsbereich eines Befehls auftretender Name nicht definiert
1. + 2. Durchlauf: Im Befehlsbereich „Setzen Befehlszähler“ auftretender Name nicht definiert
- E 9 Formaler Fehler
- E10 Name in Alphatext-Folgeaussage definiert
- E11 Erstes Zeichen nach „_“ oder „.“ ungleich BU, ZI oder „/“

- E12 Erstes Zeichen nach „/“ ungleich „_“ oder „.“ bzw. ungleich „+“ oder „-“
- E13 In Alphatext-Tabellen folgt nach „/“ Zeichen ungleich BU
- E14 Lochkartenreihenfolge falsch
- E15 Fehler bei „Setzen Befehlszähler“. Spezifikation durch zusätzliche Fehlerschlüssel

Es können Kombinationen von Fehlerschlüsseln auftreten.

Chassisbelegung (Chassis 540)

MSKZ 5/6	MSKZ 3/4	MSKZ 1/2	Umschaltplatte 402	Rechner 154				E/A 184					
12	11	10	9	8	7	6	5	4	3	2	1		
Arbeitsspeicher		Adreßbuch 3		Adreßbuch 1		Adreßbuch 2							
24	23	22	21	20	19	18	17	16	15	14	13		

Bedienungsanleitung zum NIXDORF-Re-Assembler

NIXDORF
COMPUTER

V	Muß zum re-assemblieren gesetzt sein.		
C, WZ, 'r	Einschalten (Monitorebene)		
1	Programmwahl RE-ASM I		Wiederanlauf oder erster Start des Programms?
2	RE-ASM II		START (Karten lesen).
3	RE-ASM III	2.2 (MLAR)	RESTART RE-ASM I / RE-ASM II
2.9 (P)	Ausdruck des Programmnamens	2.1 (MRAR)	WIEDERANLAUF
2.3 (MRSK)	Papierpositionierung		Parameterkarten werden nicht verarbeitet.
1.3, 1.4, 1.9	Den Objektkarten entsprechende Funktionstaste drücken (Im RE-ASM II unbedingt die gleichen Tasten wie im RE-ASM I).		RESTART nur sinnvoll, wenn der vorherige Programmteil ordnungsgemäß mit Blank-Karte beendet wurde.
1.2, 1.6, 1.7, 1.8	Tastenauswahl treffen.		Sonderfall: Bei Netzausfall, Pausen usw. Anlage mit Taste C einschalten.
	Parameter- und Objektkarten im RE-ASM I, nur Objektkarten im RE-ASM II einlegen, letzte Karte muß Blank-Karte sein.	C	

Taste	wirksam im RE-ASM	Bedeutung
1.1	I, II, III	STOP-Taste unterbricht die Karteneingabe (I, II) und die Druckausgabe (III). Ermöglicht Einzelschrittverarbeitung und Programmwechsel.
1.2	II	Keine Ausgabe des Quellprogramms auf Lochkarten.
1.3	I, II	Bei Verarbeitung von Mikro-Monitor-Objektkarten.
1.4	I, II	Keine Prüfziffernkontrolle (vgl. Taste 1.9).
1.6	I, II	Das Protokoll des Quellprogramms wird unterdrückt.
1.7	II	Ausgabe von Folgenummern auf den Quellkarten.
1.8	I, II, III	Protokoll wird zweizeilig gedruckt.
1.9	I, II	Zur Prüfziffernerrechnung werden die ersten 6 Spalten nicht berücksichtigt.
2.1	I, II	Erneuter Start des Programms (siehe Bedienungsanleitung).
2.2	I, II, III	Start mit Einlesen des Objektprogramms (siehe Taste 1.1).
2.3	I, II, III	Papierpositionierung.
2.9	I, II, III	Programmwahl. In der Stop-Phase neue Programmwahl möglich.

ENTWICKLUNGSBEREICHE ■ BASIS-SOFTWARE/FIRMWARE

Fehlermeldung

1. PRÜFZIFFER FALSCH
2. PARAMETER LEB FEHLT
3. PARAMETER SYMBOL FEHLT
4. TYPE FALSCH
5. ENDE PARAMETER FEHLT
6. LABELTABELLE ÜBERLAUF
7. WIEDERANLAUF NICHT MÖGLICH
8. ARBEITSSPEICHER ZERSTÖRT
9. LAUF 2 OHNE LAUF 1

Fehlerbeseitigung

Prüfen, ob Taste 1.9 richtig gesetzt ist. Karte prüfen, ggf. korrigieren. Fortsetzungsmöglichkeiten: Taste 2.2 oder 2.9. Reihenfolge der Parameterkarten prüfen. Fortsetzungsmöglichkeiten siehe 1.

Reihenfolge der Parameter prüfen. Mit Ausnahme des LEB-Parameters neu einlesen. Fortsetzungsmöglichkeiten siehe 1. Gelesene KORR-Karten prüfen.

Karte erstellen und vor die erste Objektkarte legen. Fortsetzungsmöglichkeiten siehe 1.

Angaben des LEB-Parameters prüfen. Evtl. einen weiteren Speicher hinzufügen. Max. 3 Speicher zulässig, sonst Programmabbruch.

Wiederanlauf-taste irrtümlich gesetzt? Mit Taste 2.2 erneut starten. Andernfalls wurde der Speicherinhalt verändert. Mit RE-ASM I neu beginnen. Neuer Start RE-ASM I ist unbedingt erforderlich (RESTART-FEHLER).

Vorangegangener RE-ASM I nicht ordnungsgemäß beendet bzw. der Arbeitsspeicher wurde anderweitig verwendet. RE-ASM I neu starten.

Kommentare zum RE-ASM II

KORREKTUR	KORR-Parameter notwendig.
INDIZIERUNG	Adreßteil ungleich Null. Prüfen, woher der Wert des Indexregisters kommt.
ABSOLUTE INDIZ.	Adreßteil gleich Null. Prüfung analog INDIZIERUNG.
BVB	Blockvorbefehl für Festspeicher.
BVB WIRKSAM	Verzweigungs-bef. nach einem BL. Programmverschiebung? Adressen prüfen!
XXXXX	Anzahl und Reihenfolge der Objektkarten in I und II nicht identisch.

Parameterkarten 1. Spalte immer Blank (□)

KSP - Parameter □ LEB □ AUF □ XXYYZZ
X, Y u. Z. geben Zuordnungen an, z. B. 070503

Symbol-Parameter □ SYMBOLE □ XX □ YY
XX = Symbol für TT-Befehlsadressen
YY = Symbol für restliche Adressen

Korr-Param. □ KORR □ S □ B □ AAAAA □ EEEEE
S = Korrektur-Schlüssel / B = Blocknummer

A = Anfangsadresse / E = Endadresse
Ende-Parameter □ ENDE □ PARAMETER
KORR-Schlüssel definiert den KORR-Bereich

1 = Befehlsbereich (A = E)
2 = Konstantentabelle (OPi- u. AD-Teil)
3 = ALC-Bereich
4-9 = Konstante (OPi-) und Label (AD-Teil)
Schlüssel minus 4 = BL-Nr. des generierten Labels

Hinweise zur Benutzung

Die nachfolgende Liste enthält alle Befehle, die bei den Modellen 820/15, 820/25 und 820/35 anwendbar sind.

In der Spalte "Betriebsprogramm" ist eine Kennzahl angegeben, die besagt, von welchem Betriebsprogramm der entsprechende Befehl interpretiert wird.

Kennzahl	Bedeutung
keine	Der Befehl ist bei allen Betriebsprogrammen zugelassen
1	Der Befehl ist mit Betriebsprogramm MSKZ1 und SKZA2 anwendbar
2	MSKZ1 + MSKZ2/LU
3	MSKZ1 + SKZA2 + MSKZ3
4	MSKZ1 + MSKZ2/LU + MSKZ3
5	Der Befehl ist mit Betriebsprogramm MFA1 oder MFAGS1 anwendbar
5.1	Zusätzlich ist der GP Modul notwendig
5.2	GP + MC
5.3	GP + PC
5.4	GP + PT
5.5	GP + TC
5.6	ST (ohne GP)
5.7	IN (ohne GP)

OP	Symbol	Funktion	Merker			Betriebs- programm
0.0	NOP	Leerbefehl, AD_{1m} : beliebig				
0.1	ACC	(E) \rightarrow SW AD_{1m} AD_r : Anzahl der Nachkommastellen falls Eingabe ohne Komma SW C, Stelle 13: Anzahl der Vorkommastellen Stelle 15: Anzahl der Nachkommastellen Stelle 12, 14: gelöscht				
0.2	MVH	$(SW AD_r) \rightarrow SW AD_{1m}$				
0.3	MV	$(SW AD_{1m}) \rightarrow SW AD_r$				
0.4	ADH	$(SW AD_{1m}) + (SW AD_r) \rightarrow SW AD_{1m}$			MC	
0.5	AD	$(SW AD_r) + (SW AD_{1m}) \rightarrow SW AD_r$			MC	
0.6	SBH	$(SW AD_{1m}) - (SW AD_r) \rightarrow SW AD_{1m}$			MC	
0.7	SB	$(SW AD_r) - (SW AD_{1m}) \rightarrow SW AD_r$			MC	
0.8	MLH	$(SW AD_{1m}) \times (SW AD_r) \rightarrow SW AD_{1m,C}$			MC	
0.9	ML	$(SW AD_r) \times (SW AD_{1m}) \rightarrow SW AD_{r,C}$ Bei (C) $\neq 0$ ist MC = 1			MC	
0.10	DVH	$(SW AD_{1m}) : (SW AD_r) \rightarrow SW AD_{1m,C}$			MC	
0.11	DV	$(SW AD_r) : (SW AD_{1m}) \rightarrow SW AD_{r,C}$ Bei Überlauf im Zielspeicherwort erfolgt Abbruch der Operation und Merker MC = 1			MC	
0.12	CPH	Vergleich (SW AD_{1m}) mit (SW AD_r)	ML	MU		
0.13	CP	Vergleich (SW AD_r) mit (SW AD_{1m})	ML	MU		
0.14	CPZ	Vergleich (SW AD_{1m}) mit Null Merker entsprechend Vergleichsergebnis	ML	MU		
0.15	CLR SGNIN CLRAL	$AD_r = 0$: Löschen (SW AD_{1m}) $AD_r = 1$: Vorzeichenwechsel in SW AD_{1m} $AD_r = 15$: Löschen AD_{1m} Speicherworte des ALC-Bereiches ab Bereichsanfang				1, 2, 3, 4

OP	Symbol	Funktion	Merker			Betriebs- programm
1.0	BR	Sprung nach AD_{lmr}				
1.1	BR1	Sprung wenn $M1 = 1$ nach AD_{lmr}				
1.2	BR2	Sprung wenn $M2 = 1$ nach AD_{lmr}				
1.3	BR3	Sprung wenn $M3 = 1$ nach AD_{lmr}				
1.4	BR4	Sprung wenn $M4 = 1$ nach AD_{lmr}				
1.5	BR5	Sprung wenn $M5 = 1$ nach AD_{lmr}				
1.6	BRL	Sprung wenn $ML = 1$ nach AD_{lmr}				
1.7	BRU	Sprung wenn $MU = 1$ nach AD_{lmr}				
1.8	BRC	Sprung wenn $MC = 1$ nach AD_{lmr}				
1.9	BXG	Sprung wenn $(I) > 1023$ nach AD_{lmr}				
1.10	BXU	Sprung wenn $(I) \neq 0$ nach AD_{lmr}				
1.11	BRS	Unterprogrammprung nach AD_{lmr} (max. 5 Unterprogrammstufen)				
1.12	BRR	Rücksprung aus Unterprogramm nach Rückkehradresse + AD_{lmr} modulo 2^{11}				
1.13	SST	Substitution des Befehls unter der Adresse AD_{lmr}				
1.14	ICA	AD_{lmr} von (AD_{lmr}) nach SW A vor das Komma			MC	
1.15	CA	AD_{lmr} nach SWA vor das Komma			MC	
2.0	SR	Rechtshift (SW AD_{lm}) um AD_r Stellen			MC	
2.1	SL	Linksshift (SW AD_{lm}) um AD_r Stellen			MC	
2.2	XF XFR	$AD_1 = 0, AD_m + 0$: Indexregistervorbefehl für einen Befehl je Anweisung $AD_m + 12$: Indexregistervorbefehl bis zum nächsten Vorbefehl gültig AD_m : Indexregister für Indizierung (0 - 3) AD_r : Indexregister für Operation (0 - 3)				

OP	Symbol	Funktion	Merker		Betriebsprogramm
2.2	MVX MVXH	$AD_1 = 0, AD_m: 0 - 3, AD_r: 0 - 3$ und 5 $AD_1 = 0, AD_m: 0 - 3, AD_r: 0 - 7$ $AD_m + 4: (\text{Ireg. } AD_r) \rightarrow \text{Ireg. } AD_m$ $AD_m + 8: (\text{Ireg. } AD_m) \rightarrow \text{Ireg. } AD_r$			5 1, 2, 3, 4
	INCX XINC	$AD_1 = 0, AD_r + 8: (\text{Befehlszähler } AD_r: 0 - 5)$ $AD_m = 4: (\text{Befehlszähler } AD_r) \rightarrow I_0$ (Blockzähler AD_r) $\rightarrow I_1$ $AD_m = 8: (I_0) \rightarrow \text{Befehlszähler } AD_r$ $(I_1) \rightarrow \text{Blockzähler } AD_r$			1, 2, 3, 4 1, 2, 3, 4
	CPX	$AD_1 = 1, AD_m: 0 - 3, AD_r: 0 - 3$ Vergleich (Ireg. AD_m) mit (Ireg. AD_r)	ML	MU	
	SSTX MVOX	$AD_1 = 2, AD_m: 0 - 3, AD_r: 0 - 3$ $(\text{Ireg. } AD_m) = OP$ $MC = AD_1$ $(\text{Ireg. } AD_r) = AD_{1mr}$	} Ergibt einen neuen Befehl		2, 4
		$AD_m + 0: \text{Substitution: Der neue Befehl wird ausgeführt}$			
		$AD_m + 8: \text{Befehlsspeicherung: Der neue Befehl wird unter der Adresse } (I_3) \text{ in dem durch Vorbefehl Blockumschaltung bestimmten Kernspeicherblock gespeichert. Danach ist } I_3 \text{ um 1 erhöht.}$			1, 2, 3, 4, 5.1, 5.2, 5.3, 5.4, 5.5
	CLX	$AD_1 = 3, AD_m: 0 - 3, AD_r: 0 - 3$ $2048 - (\text{Ireg. } AD_r) \rightarrow \text{Ireg. } AD_m$			
	ADX	$AD_1 = 4, AD_m: 0 - 3, AD_r: 0 - 3$ $(\text{Ireg. } AD_m) + (\text{Ireg. } AD_r) \rightarrow \text{Ireg. } AD_m$			MC
	SBX	$AD_1 = 5, AD_m: 0 - 3, AD_r: 0 - 3$ $(\text{Ireg. } AD_m) - (\text{Ireg. } AD_r) \rightarrow \text{Ireg. } AD_m$			MC
	SRX	$AD_1 = 6, AD_m: 0 - 3, AD_r: 0 - 11$ Rechtsshift (Ireg. AD_m) um AD_r Stellen			MC
SLX	$AD_1 = 7, AD_m: 0 - 3, AD_r: 0 - 11$ Linksshift (Ireg. AD_m) um AD_r Stellen			MC	

OP	Symbol	Funktion	Merker			Betriebs- programm
2.3	RND	Runden (SW AD_{1m}) mit Stelle AD_r (Absolute Addition von 5 in Stelle AD_r , da- nach werden diese und die rechts anschließen- den Stellen auf 0 gesetzt).			MC	
2.4	CNT	Zählen (SW AD_{1m}) in Stelle AD_r (Absolute Addition von 1 in Stelle AD_r)			MC	
2.5	DC1 bis DC4	$AD_1 + 0$: Transport (I) nach SW AD_m ab Stelle AD_r zuzüglich AD_1 Stellen links anschließend			MC	
	,PL	$AD_1 + 4$: Zusätzlich Addition vom angesproch- enen Teil des Speicherwortinhalts			MC	
2.6	XR1 bis XR4	$AD_1 + 0$: Transport (SW AD_m) ab Stelle AD_r zuzüglich AD_1 Stellen links nach Indexregister I_{1mr}			MC	
	,PL	$AD_1 + 4$: Zusätzlich Addition des ursprüng- lichen Indexregisterinhalts			MC	
2.7	XM1 bis XM4	$AD_1 + 0$: Transport (SW AD_m) ab Stelle AD_r zuzüglich AD_1 Stellen links nach Indexregister I_{1m} und $I_r = 0$			MC	
	,PL	$AD_1 + 4$: Zusätzlich Addition des ursprüng- lichen Indexregisterinhalts			MC	
2.8	XL1 bis XL4	$AD_1 + 0$: Transport (SW AD_m) ab Stelle AD_r zuzüglich AD_1 Stellen links nach Indexregister I_1 und $I_{1mr} = 0$			MC	
	,PL	$AD_1 + 4$: Zusätzlich Addition des ursprüng- lichen Indexregisterinhalts			MC	
2.9	ICX	AD_{1mr} von (AD_{1mr}) nach Indexregister				
2.10	CX	AD_{1mr} nach Indexregister				
2.11		$AD_1 = 0, AD_m = 0$: Kugelkopposition nach Indexregister				
	POSX	$AD_r = 2$: Serialdrucker 1				1, 2, 3, 4, 5
	POSXS	$AD_r = 3$: Serialdrucker 2				1, 2, 3, 4

OP	Symbol	Funktion	Merker			Betriebsprogramm
2.11	PBG	AD ₁ = 0, AD _m + 8: Anfang des PCT-Bereichs vorgeben Bit 1 bis 7: Anfangsspeicherwort				
	,RD	AD ₁ + 0: Eingabebereich, RD-Bereich (zum Einlesen von Lochkarten)				1, 3, 4, 5.3
	,RD	AD ₁ + 0: Eingabebereich, RD-Bereich (zum Einlesen von Lochstreifen)				1, 3, 4, 5.4
	,RD	AD ₁ + 0: Eingabebereich, RD-Bereich (zum Einlesen von Cassette)				3, 4, 5.5
	,PN	AD ₁ + 2: Ausgabebereich, PN-Bereich (zum Stanzen von Lochkarten)				1, 3, 4, 5.3
	,PN	AD ₁ + 2: Ausgabebereich, PN-Bereich (zur Ausgabe auf Cassette) (Der PN-Bereichsanfang fällt intern mit dem ALC-Bereichsanfang zusammen)				3, 4, 5.5
	PBP	AD ₁ = 1: PCT-Bereichszeiger-Zugriff Bit 1 bis 7: Nummer AD _m + 0: Laden des PCT-Bereichszeigers mit Nummer				
	PBPX	AD _m + 8: Inhalt des PCT-Bereichszeigers nach Indexregister Vergleich mit Nummer	ML	MU		1, 3, 4, 5.3-5.5
	,RD	AD ₁ + 0: Eingabebereichszeiger, RDP				1, 3, 4, 5.3-5.5
	,PN	AD ₁ + 2: Ausgabebereichszeiger, PNP				1, 3, 4, 5.3, 5.5
	MVBLK	AD ₁ = 6: Löschen des PCT-Ausgabebereichs ab Inhalt Ausgabebereichszeiger PNP Bit 1 bis 7: Anzahl der Spalten (entspricht der Ausgabe von Leerspalten auf der Lochkarte. Für eine Spalte werden zwei Kernspeicherstellen gelöscht)				1, 3, 4, 5.3, 5.5
	BL	AD ₁ = 4: Vorbefehl Blockumschaltung AD _r : 0-7 AD _m = 0: Programmblock im Festspeicher				1 - 4
	,LIV	AD _r : 0-2 AD _m = 1: Programmblock im Kernspeicher AD _r				5 1 - 4, 5.1 - 5.5
2.12	SMWT	Setze Merker ML, MU, MC nach Beendigung				
	,MTW	AD _{1mr} = 0.10.0: Ausgabe Simultangerät 1	ML	MU	MC	
	,MTWS	AD _{1mr} = 0.10.1: Ausgabe Simultangerät 2	ML	MU	MC	1, 2, 3, 4, 5.1-5.5

OP	Symbol	Funktion	Merker		Betriebs- programm																																																																																									
2.12	WTC	AD _{1mr} = 0.0.0: Warte auf Taste C (Lampe rot)																																																																																												
	WTF	AD _{1mr} = 0.0.1: Warte auf Taste F (Lampe orange)																																																																																												
	EFK	AD _{1mr} = 0.7.15: Funktionstastenauswurf																																																																																												
	WT SM1 bis SM5 SML und SMU .ZERO .ONE	AD ₁ = 0: Warte auf Bedingung AD _{mr} AD ₁ ≠ 0: Setze Merker AD ₁ auf 1, wenn Bedingung AD _{mr} erfüllt ist. AD _{mr} = 0.0: Merker AD ₁ auf 0 setzen AD _{mr} = 3.12: Merker AD ₁ auf 1 setzen	ML	MU																																																																																										
		<table border="1"> <thead> <tr> <th>AD_{mr}</th> <th>Symbol</th> <th>Taste</th> <th>AD_{mr}</th> <th>Symbol</th> <th>Taste</th> </tr> </thead> <tbody> <tr> <td>1.1</td> <td>MLFM</td> <td></td> <td>2.1</td> <td>MRAR</td> <td></td> </tr> <tr> <td>1.2</td> <td>MLFMB</td> <td></td> <td>2.2</td> <td>MLAR</td> <td></td> </tr> <tr> <td>1.3</td> <td>MCM</td> <td></td> <td>2.3</td> <td>MRSK</td> <td></td> </tr> <tr> <td>1.4</td> <td>MCMB</td> <td></td> <td>2.4</td> <td>MLSK</td> <td></td> </tr> <tr> <td>1.6</td> <td>MLFLU</td> <td></td> <td>2.5</td> <td>MFTS</td> <td></td> </tr> <tr> <td>1.7</td> <td>MLFL</td> <td></td> <td>2.6</td> <td>MIT</td> <td></td> </tr> <tr> <td>1.8</td> <td>MNA</td> <td></td> <td>2.7</td> <td>MMIN</td> <td></td> </tr> <tr> <td>1.9</td> <td>MSTO</td> <td></td> <td>2.8</td> <td>MD</td> <td>D</td> </tr> <tr> <td></td> <td></td> <td></td> <td>2.9</td> <td>MP</td> <td>P</td> </tr> <tr> <td></td> <td></td> <td></td> <td>2.10</td> <td>MPML</td> <td>%</td> </tr> <tr> <td></td> <td></td> <td></td> <td>2.11</td> <td>MPCT</td> <td>%</td> </tr> <tr> <td></td> <td></td> <td></td> <td>2.12</td> <td>MPD</td> <td>q</td> </tr> <tr> <td></td> <td></td> <td></td> <td>2.13</td> <td>MS</td> <td>S</td> </tr> <tr> <td></td> <td></td> <td></td> <td>2.14</td> <td>MW</td> <td>W</td> </tr> </tbody> </table>	AD _{mr}	Symbol	Taste	AD _{mr}	Symbol	Taste	1.1	MLFM		2.1	MRAR		1.2	MLFMB		2.2	MLAR		1.3	MCM		2.3	MRSK		1.4	MCMB		2.4	MLSK		1.6	MLFLU		2.5	MFTS		1.7	MLFL		2.6	MIT		1.8	MNA		2.7	MMIN		1.9	MSTO		2.8	MD	D				2.9	MP	P				2.10	MPML	%				2.11	MPCT	%				2.12	MPD	q				2.13	MS	S				2.14	MW	W		
AD _{mr}	Symbol	Taste	AD _{mr}	Symbol	Taste																																																																																									
1.1	MLFM		2.1	MRAR																																																																																										
1.2	MLFMB		2.2	MLAR																																																																																										
1.3	MCM		2.3	MRSK																																																																																										
1.4	MCMB		2.4	MLSK																																																																																										
1.6	MLFLU		2.5	MFTS																																																																																										
1.7	MLFL		2.6	MIT																																																																																										
1.8	MNA		2.7	MMIN																																																																																										
1.9	MSTO		2.8	MD	D																																																																																									
			2.9	MP	P																																																																																									
			2.10	MPML	%																																																																																									
			2.11	MPCT	%																																																																																									
			2.12	MPD	q																																																																																									
			2.13	MS	S																																																																																									
			2.14	MW	W																																																																																									
	AD _{mr}	Symbol	Bedingung																																																																																											
			<u>Operation beendet :</u> <u>Simultangerät 1 :</u> Serialdrucker 1 <u>Simultangerät 2 :</u> Serialdrucker 2 Kartenlocher 1 Kartenlocher 2 Streifenlocher 1 Streifenlocher 2 Cassette		ML MU																																																																																									
	8.0	MTW																																																																																												
	8.1	MTWS			1, 2, 3, 4																																																																																									
	8.1	MPC			1, 3, 4, 5.3																																																																																									
	8.1	MPCS			1, 3, 4																																																																																									
	8.1	MPT			1, 2, 3, 4, 5.4																																																																																									
	8.1	MPTS			1, 2, 3, 4 3, 4, 5.5																																																																																									

OP	Symbol	Funktion			Merker			Betriebsprogramm
2.12		AD _{mr}	Symbol	Bedingung				
		8.4	MMAG	<u>Karte da:</u> Magnetkontenschacht 1				2, 4
		8.8	MMAGB	Magnetkontenschacht 2				2, 4, 5.2
		8.4		Intromat 1				1, 3, 5.7
		8.8		Intromat 2				1, 3, 5.7
		8.4		Steinhilber				1, 3, 5.6
		8.12	MPLU	<u>Kein Papier:</u> Lep. 1, Drucker 1				
		8.13	MPLL	Lep. 2, Drucker 1				
		8.14	MPLSU	Lep. 1, Drucker 2				
		8.15	MPLSL	Lep. 2, Drucker 2				
		4.0	MFPT	<u>Meldungen:</u> Lochstreifenende 1				1, 3, 4, 5.4
		4.1	MFPTS	Lochstreifenende 2				1, 3, 4,
		4.2	MWPT	<u>Stanzfehler:</u> Streifenlocher 1				1, 2, 3, 4, 5.4
		4.3	MWPTS	Streifenlocher 2				1, 3, 4
	4.4	MHPT	<u>Kein Karten-</u> <u>führungsloch:</u> Streifenlocher 1				1, 3, 4, 5.4	
	4.5	MHPTS	Streifenlocher 2				1, 3, 4	
	WT ,SEP	AD _{1mr} = 0.4.12 :Setzen Anfang des Befehlsbereiches. Befehl muß unmittelbar auf Vorbefehl EP folgen.						5.1, 5.2, 5.3, 5.4 5.5
	,CEP	AD _{1mr} = 0.4.11 :Freigabe des Befehlsbereiches als Datenbereich						5.1, 5.2, 5.3, 5.4, 5.5
2.13	TAB	Tabulation nach Position AD _{mr} AD ₁ = 0 : Serialdrucker 1						
	TABS	AD ₁ = 1 : Serialdrucker 2						1, 2, 3, 4
	TW	AD ₁ = 2 : Freigabe Serialdrucker 1 bis Position AD _{mr} - 1			ML	MU	MC	

OP	Symbol	Funktion	Merker			Betriebs- programm	
			ML	MU	MC		
2.13	TWX	AD ₁ = 4: Freigabe 1 Zeichen links von Position AD _m , Zeichen nach 1	ML	MU	MC	1, 2, 3, 4	
	.TWS	AD ₁ + 1: mit Druck auf Serialdrucker 2				1, 2, 3, 4	
		Taste*	(!) bei TWX	ML	MU	MC	
		Zeichentaste (bei TWX)	ALC-Code	0	0	0	1, 2, 3, 4
		Rücktaste (bei TWX)	5.12	0	0	0	1, 2, 3, 4
		Tabulationstaste	6.8	0	1	0	
	Wagenaufzugtaste	6.12	0	0	1		
	Auslösetaste	6.11	1	0	0		
2.14	LF	Zeilenschaltung					
		Bit 1 bis 7: Anzahl der Zeilen					
	.ST	AD ₁ = 0: Steinhilbereinzug				1, 3, 5, 6	
	.INTR	AD ₁ = 0: Intromat Schacht 1				1, 3, 5, 7	
	.INTRB	AD ₁ = 1: Intromat Schacht 2				1, 3, 5, 7	
	.MAG	AD ₁ = 0: Magnetkontenschacht 1				2, 4	
	.MAGB	AD ₁ = 1: Magnetkontenschacht 2				2, 4, 5, 2	
	.BACK	AD _m + 8: in Einzugsrichtung				2, 4, 5, 2	
	.LEP, UP	AD ₁ = 2: Vorschubeinrichtung 1 an Serial- drucker 1					
	.LEP, LOW	AD ₁ = 3: Vorschubeinrichtung 2 an Serial- drucker 1					
	.LEPS, UP	AD ₁ = 4: Vorschubeinrichtung 1 an Serial- drucker 2				1, 2, 3, 4	
	.LEPS, LOW	AD ₁ = 5: Vorschubeinrichtung 2 an Serial- drucker 2				1, 2, 3, 4	
	.SAVPL	AD _m + 8: mit einer Walzenschaltung dto, ohne Zeilenzähler- Zählung				2, 4 1, 3	
	.PLT	AD ₁ = 6: Walze von Serialdrucker 1					
	.PLTS	AD ₁ = 7: Walze von Serialdrucker 2				1, 2, 3, 4	
	Auswurf Intromat:						
.OUT1	AD _{1mr} = 0,3,0: Intromat 1				1, 3, 5, 7		
.OUT2	AD _{1mr} = 1,3,0: Intromat 2				1, 3, 5, 7		

* Die Tabelle zeigt die Stellung der Merker in Abhängigkeit der gedruckten Taste

OP	Symbol	Funktion	Merker			Betriebs- programm
2.15	ALC	Ausgabe eines Zeichens Bit 1 bis 7: ALC-Code des Zeichens				
	,TW1	AD ₁ + 1: Serialdrucker 1				
	,TW2	AD ₁ = 2: Serialdrucker 2				1, 2, 3, 4
	,RED	AD _m + 8: Druck in Rot				
	,PT	AD ₁ = 4: Streifenlocher 1				1, 2, 3, 4, 5, 4
	,PTS	AD ₁ = 6: Streifenlocher 2				1, 2, 3, 4
3.0	TT	Drucke Tabelle (bis Endezeichen 3.15) AD _{1mr} : Anfangsadresse der Tabelle				
		Merker werden durch SMWT (2.12) gesetzt	ML	MU	MC	
		8 YTAB	0	1	0	
		12 (Tab. nach Ausgangspos.) YCAR	0	0	1	
		11, sonst YTRK	1	0	0	
3.1	BUFA ABUF ,MVSN	Transport zwischen ALC-Bereich und SW A AD ₁ = 0: von ALC-Bereich ab (I ₃) nach A AD ₁ = 1: von A nach ALC-Bereich ab (I ₃) AD ₁ + 2: Transport mit Vorzeichen AD _m : Anzahl der Nachkommastellen AD _r : Anzahl der Stellen				
		AD ₁ = 4, AD _r = 0: Transport vom ALC-Bereich ab (I ₃) nach Indexregister				
		AD _r + 0: eine Sedezimalzahl AD _r + 1: zwei Sedezimalzahlen				
	BUFX1 BUFX2	AD ₁ = 4, AD _r = 2: Transport Inhalt Indexre- gister nach ALC-Bereich ab (I ₃)				
		AD _r + 0: eine Sedezimalzahl				
	XBUF1 XBUF2	AD _r + 1: zwei Sedezimalzahlen				
		AD ₁ = 4: Drucken Inhalt ALC-Bereich ab (I ₃) bis Endezeichen Bit 1 bis 7: maximale Anzahl der aus- zudruckenden Zeichen				
	ALOUT6 ALOUT8	AD _m + 0: im 6-Bit-Code gespeichert				1, 2, 3, 4
		AD _m + 8: im 8-Bit-Code gespeichert				

OP	Symbol	Funktion	Merker			Betriebs- programm
			ML	MU	MC	
3.1	.TW1	AD ₁ + 1: Druck auf Serialdrucker 1				1, 2, 3, 4
	.TW2	AD ₁ + 2: Druck auf Serialdrucker 2				
		Merker werden durch SMWT (2.12) gesetzt	ML	MU	MC	
		Ende durch 6.8	0	1	0	
		Ende durch 6.12 (Tab. nach Ausgangspos.)	0	0	1	
		Ende durch 6.11	1	0	0	
		Ende durch Erreichen der Anzahl	0	0	0	
3.2	ED	Drucken Inhalt Druckspeicherwort D1 (ohne Vorbefehl (D1) in Rot falls negativ) Drucken Inhalt Druckspeicherwort D2 AD ₁ : Anzahl der Nachkommastellen AD _{mr} : Druckposition des Kommas				1, 2, 3, 4
3.3		Zeilenzähler-Zugriff Bit 1 bis 7: Zeilennummer				
	LC	AD _m + 0: Zeilenzähler laden				
	LCX	AD _m + 8: Inhalt Zeilenzähler nach I, Vergleich mit Zeilennummer	ML	MU		
	.ST	AD ₁ = 0: Steinhilbereinzug				1, 3, 5, 6
	.INTR	AD ₁ = 0: Intromat Schacht 1				1, 3, 5, 7
	.INTRB	AD ₁ = 1: Intromat Schacht 2				1, 3, 5, 7
	.MAG	AD ₁ = 0: Magnetkontenschacht 1				2, 4
	.MAGB	AD ₁ = 1: Magnetkontenschacht 2				2, 4, 5, 2
	.LEP, UP	AD ₁ = 2 Lep. 1, SM 1				
	.LEP, LOW	3 Lep. 2, SM 1				
	.LEPS, UP	4 Lep. 1, SM 2				1, 2, 3, 4
	.LEPS, LOW	5 Lep. 2, SM 2				1, 2, 3, 4
	.PLT	6 Walze, SM 1				
.PLTS	7 Walze, SM 2				1, 2, 3, 4	
3.4	VP	Vertikalpositionierung auf Zeile Bit 1 bis 7: Zeilennummer				
	.ST	AD ₁ = 0: Steinhilbereinzug				1, 3, 5, 6
	.INTR	AD ₁ = 0: Intromat Schacht 1				1, 3, 5, 7
	.INTRB	AD ₁ = 1: Intromat Schacht 2				1, 3, 5, 7
	.MAG	AD ₁ = 0: Magnetkontenschacht 1				2, 4

OP	Symbol	Funktion	Merker			Betriebsprogramm
3.4	,MAGB	AD ₁ = 1: Magnetkontenschacht 2				2, 4, 5.2
	,LEP, UP	AD ₁ = 2: Vorschubeinrichtung 1 an Serialdrucker 1			MC	
	,LEP, LOW	AD ₁ = 3: Vorschubeinrichtung 2 an Serialdrucker 1			MC	
	,LEPS, UP	AD ₁ = 4: Vorschubeinrichtung 1 an Serialdrucker 2			MC	1, 2, 3, 4
	,LEPS, LOW	AD ₁ = 5: Vorschubeinrichtung 2 an Serialdrucker 2			MC	1, 2, 3, 4
	,PLT	AD ₁ = 6: Walze von Serialdrucker 1			MC	
	,PLTS	AD ₁ = 7: Walze von Serialdrucker 2			MC	1, 2, 3, 4
			(Zellenzähler) > Zeilennummer			MC
	ALBG	AD _{1m} + 0, 8: Bit 1 bis 7: Anfang ALC-Bereich				
		AD ₁ = 0: Magnetkontenvorbefehl: Einzug mit Lesen, Auswurf mit Schreiben				2, 4, 5.2
	,READC	AD ₁ = 4: Magnetkontenvorbefehl: Auswurf mit Schreiben, Kontrolllesen				2, 4
3.5	EDPT	Lochen Inhalt Speicherwort D2 (ohne Vorbefehl nach Tabelle 1)				1, 2, 3, 4, 5.4
	,PT	AD ₁ = 4: Streifenlocher 1				
	,PTS	AD ₁ = 6: Streifenlocher 2 AD _r : Anzahl der Nachkommastellen				1, 2, 3, 4
	PTF,	AD _m = 2: Lochstreifenvorbefehle				1, 2, 3, 4, 5.4
	,TBL1 ,TBL2	AD _m + 0: Tabelle 1 AD _m + 8: Tabelle 2 } als Ergänzung für alle PTF-Vorbefehle				
	PTF	Vorbefehl für ALC (2.15) AD _r : Anzahl der Wiederholungen				1, 2, 3, 4, 5.4
PTF	Vorbefehl für EDPT (3.5) AD _r : Mindestanzahl der auszugebenden Zeichen vor dem Komma				1, 2, 3, 4, 5.4	

OP	Symbol	Funktion	Merker	Betriebs- programm	
3.5	,ZERO	$AD_m + 0$: Vormullen			
	,STAR	$AD_m + 1$: Sicherungssterne			
	,BLNK	$AD_m + 4$: Leertasten			
	,MIN	$AD_1 = 0$: ohne Kennzeichen			
	,SGN	$AD_1 = 1$: Leertaste oder Minuszeichen			
	,ITS	$AD_1 = 3$: Kennzeichen + oder -			
	,FTS	$AD_1 = 5$: Kennzeichen \diamond oder \triangleleft			
		$AD_1 = 7$: Kennzeichen \times oder \times			
	PTF	Vorbefehl für ALPT (3.13)			1, 2, 3, 4, 5, 4
	,PT ,PTS	$AD_1 = 4$: Streifenlocher 1 $AD_1 = 6$: Streifenlocher 2			1, 2, 3, 4
3.6	PIF	Vorbefehl für RDPT (3.13)			1, 3, 4, 5, 4
	,CD5	AD_{m+0} : 6-, 7-, 8-Spur-Lochstreifen			
	,YZI	$AD_m + 1$: 5-Spur-Lochstreifen			
	,YBU	$AD_r = 0$: nach Umschaltzeichen ZI			
		$AD_r = 8$: nach Umschaltzeichen BU			
		Druckvorbefehl für TW und TWX (2.13), ALOUT6 und ALOUT8 (3.1), ALIN6 und ALIN8 (3.13), TT (3.0)			
	EDF	$AD_m = 0$: Vorbefehl für Serialdrucker 1			1, 2, 3, 4
	EDFS	$AD_m = 2$: Vorbefehl für Serialdrucker 2			
	,RED	$AD_m + 8$: Druck in Rot			
		Druckvorbefehl für ALC (2.15)			
	EDF	$AD_m = 0$: Vorbefehl für Serialdrucker 1			1, 2, 3, 4
	EDFS	$AD_m = 2$: Vorbefehl für Serialdrucker 2			
	,RED	$AD_m + 8$: Druck in Rot			
		$AD_r : 0 - 15$: Anzahl der Wiederholungen			

OP	Symbol	Funktion	Merker			Betriebs- programm
3.6		Druckvorbefehl für ED (3.2) AD _r : Mindestanzahl der auszugebenden Zeichen vor dem Komma				1, 2, 3, 4
	,EDF	AD _m = 0: Druck (D1), Serialdrucker 1				
	,EDFS	AD _m = 2: Druck (D2), Serialdrucker 2				
	,ZERO	AD _m + 0: Vormullen				
	,STAR	AD _m + 1: Sicherungssteme				
	,BLACK	AD _m + 0: Druck in Schwarz				
	,REDL	AD _m + 4: Druck in Rot falls (D) negativ				
	,RED	AD _m + 8: Druck in Rot				
	,REDG	AD _m + 12: Druck in Rot falls (D) positiv				
	,MIN	AD ₁ = 0: ohne Kennzeichen AD ₁ = 1: Leertaste oder Minuszeichen				
	,SGN	AD ₁ = 3: Kennzeichen + oder -				
	,ITS	AD ₁ = 5: Kennzeichen ◊ oder ◊				
,FTS	AD ₁ = 7: Kennzeichen × oder ×					
3.7	PBA, PC	AD ₁ = 0: Transport numerischer Zeichen zwischen dem PCT-Bereich und A entsprechend Lochkartenvorbefehl PCF (3.8) AD _m : Anzahl der Nachkommastellen (0-KA) AD _r : Anzahl der Zeichen (1 - 15)	ML	MU	MC	1, 3, 4, 5, 3
		Nicht numerisches Zeichen (Abbruch)			MC	
	DEC, PC	AD ₁ = 1: Decodierung des PCT-Eingabebereichs- inhalts (RD-Bereich) ab (RDP) nach PC-Tabelle Bit 1 bis 7: Anzahl Bei Anzahl = 0: Beendigung nur durch Ende- zeichen (PC-Code für 3.15, 6.8, 6.11 oder 6.12) Bei Anzahl ≠ 0: Beendigung durch Endezeichen oder Erreichen der Anzahl		MU	MC	1, 3, 4, 5, 3
,CD6	AD _m + 0: im 6-Bit-Code				1, 3, 4	
,CD8	AD _m + 8: im 8-Bit-Code nach ALC-Bereich ab (I ₃) bringen Bei Beendigung durch Endezeichen wird dieses decodiert und gespeichert				1, 3, 4, 5, 3	

OP	Symbol	Funktion	Merker		Betriebsprogramm	
3. 7	, WNI	AD ₁ + 0: bei Ende durch Anzahl kein Endezeichen speichern			1, 3, 4, 5, 3	
		AD ₁ + 2: bei Ende durch Anzahl Endezeichen 6.8 speichern			1, 3, 4, 5, 3	
		Endezeichen nicht aufgetreten Nicht definiertes Zeichen (Abbruch)		MU	MC	
	RDX, PC XPN, PC	AD ₁ = 2: Transport eines alphanumerischen Zeichens zwischen PCT-Bereich und Indexregister. Umcodierung nach PC-Tabelle				
		AD _r = 0: ((RDP)) decodiert nach I			MC	1, 3, 4, 5, 3
		AD _r = 1: (I) codiert nach (PNP)				1, 3, 4, 5, 3
		Nicht definiertes Zeichen			MC	
	PBA, PT	AD ₁ = 4: Transport vom PCT Eingabebereich ab (RDP) nach SW A und I. Umcodierung nach PT-Tabelle AD _m : Anzahl der Nachkommastellen AD _r : max. Anzahl Ziffern, Vornullen, Sicherungsterne, Leertasten Falls Komma, dann kommagerechter Transport ohne Beachtung von AD _m . In SW C: Stelle 13: Anzahl Vorkommastellen Stelle 15: Anzahl Nachkommastellen Stellen 12, 14: gelöscht Beendigung durch Zeichen verschieden von Ziffer, Komma, Stern, Leertaste vor erster Zahl, danach durch Zeichen ungleich Ziffer, Komma, nach Komma durch Zeichen ungleich Ziffer. Dieses Zeichen nach I, Bei -, 0, * kommt - nach SW A, sonst +.		MU	MC	1, 3, 4, 5, 4
		Anzahl der Zeichen von AD _r verschieden Nicht definiertes Zeichen (Abbruch)		MU	MC	
	DEC, PT	AD ₁ = 5: Decodierung PCT-Eingabebereichsinhalt ab (RDP) nach PT-Tabelle Bit 1 bis 7: Anzahl Bei Anzahl = 0: Beendigung nur durch Endezeichen (PT-Code für 3.15) Bei Anzahl ≠ 0: Beendigung durch Endezeichen oder Erreichen der Anzahl		MU	MC	1, 3, 4, 5, 4

OP	Symbol	Funktion	Merker			Betriebsprogramm	
3.7	,CD6	$AD_m + 0$: im 6-Bit-Code				1, 3, 4	
	,CD8	$AD_m + 8$: im 8-Bit-Code nach ALC-Bereich ab (I_3) bringen				1, 3, 4, 5, 4	
	,WN1	$AD_1 + 0$: mit Decodierung des Zeichens nach Endezeichen				1, 3, 4, 5, 4	
		$AD_1 + 2$: ohne Decodierung des Zeichens nach Endezeichen (speichern 6, 8)				1, 3, 4, 5, 4	
		Endezeichen nicht aufgetreten Nicht definiertes Zeichen (Abbruch)		MU		MC	
	RDX	$AD_1 = 6$: Transport eines Zeichens vom PCT-Eingabebereich nach I				MC	1, 3, 4, 5, 4
,PT	$AD_r = 0$: ((RDP)) decodiert nach I						
	Nicht definiertes Zeichen				MC		
3.8	PCH, RD	$AD_1 = 0$: auf Gerät 1 eine Karte lesen				1, 3, 4, 5, 3	
	,PCS	$AD_1 = 1$: auf Gerät 2 eine Karte lesen				1, 3, 4	
	,NCT	AD_{mr} : Anzahl der Spalten der Karte					
		$AD_m + 8$: ohne interne Umwandlung des 12 Bit Lochkartencodes					1, 3, 4
	PCH, PN	$AD_1 = 2$: auf Gerät 1 eine Karte stanzen				1, 3, 4, 5, 3	
	,PCS	$AD_1 = 3$: auf Gerät 2 eine Karte stanzen Beim Nixdorf-Locher ist (I_2) Prüfzeichen				1, 3, 4	
	,PCBI	AD_{mr} : Anzahl der zu stanzenen Spalten					
		$AD_m + 8$: zwei 6-Bit tiefe Kernspeicherstellen in eine Lochkartenspalte					1, 3, 4
		Merker werden durch SMWT (2.12) gesetzt	ML	MU	MC		
		Lochvorgang fehlerhaft durchgeführt Fehler bei Kontrolleung festgestellt Nur Kontrolleung durchgeführt	ML	MU	MC		
PCF	Lochkartenvorbefehl für PBA, PC (3.7)					1, 3, 4, 5, 3	
,RD	$AD_1 = 4$: Transport von RD-Bereich nach A						
,PN	$AD_1 = 6$: Transport von A nach PN-Bereich AD_m : Code für Überlochverarbeitung (vgl. Tabelle Blatt G 21) AD_r : Stelle in A (1 - 15), die dazugehörige Lochkartenspalte trägt die Überlochung (Der Vorbefehl PCF fällt intern mit PTF; PTFS und EDFS zusammen)						

OP	Symbol	Funktion	Merker		Betriebsprogramm
3.8	COD, PC	$AD_1 = 5$: Codierung des ALC-Bereichsinhalts ab (1_3) nach PC-Tabelle Bit 1 bis 7: Anzahl Bei Anzahl = 0: Beendigung nur durch Endezeichen Bei Anzahl \neq 0: Beendigung durch Endezeichen oder Erreichen der Anzahl	MU		1, 3, 4, 5, 3
	, CD6	$AD_m + 0$: vom 6-Bit-Code			1, 3, 4
	, CD8	$AD_m + 8$: vom 8-Bit-Code nach PCT-Ausgabebereich ab (PNP) bringen			1, 3, 4, 5, 3
	, WNI	$AD_1 + 0$: mit Endezeichen (PC-Code für 6.8 6.11, 6.12) $AD_1 + 2$: ohne Endezeichen			1, 3, 4, 5, 3 1, 3, 4, 5, 3
		Endezeichen nicht aufgetreten	MU		
3.9	MVV	Bit 1 bis 10: Adresse des Speicherwortes $AD_1 + 0$: Transport Inhalt von hoher Adresse nach A			
	MVVH	$AD_1 + 4$: Transport Inhalt von A nach hoher Adresse			
	EP	Als Vorbefehl für WT, SEP. 1. Betriebsprogramm MFA 1 AD_{Imr} = Adresse des letzten Speicherwortes im Datenbereich 2. Betriebsprogramm MFA 1 AD_{Imr} = Adresse des letzten Speicherwortes kleiner 123 im Datenbereich Transport des Speicherwortes nach A			5.1, 5.2, 5.3, 5.4, 5.5
3.10	INT	$AD_1 + 0$: Intromat Schacht 1			1, 3, 5, 7
	INTB	$AD_1 + 1$: Intromat Schacht 2			1, 3, 5, 7
	INTO	$AD_{Imr} + 0.0.1$: Intromat Schacht 1 und 2			1, 3, 5, 7
	, IN	$AD_1 = 2$: Einzug			
	, CUT	$AD_1 = 4$: Stanzen Zeilenmarkierung			
	SORTMX	$AD_{Imr} = 6.0.0$: Sortimatwert nach Indexregister			1, 3, 5, 7

OP	Symbol	Funktion	Merker			Betriebs- programm
3.10	STN	$AD_1 + 0$: Steinhilbereinzug				1, 3, 5, 6
	,OUT	$AD_1 = 0$: Auswurf				
	,IN	$AD_1 = 2$: Einzug				
	,CUT	$AD_1 = 4$: Stanzen Zeilenmarkierung				
	MAGN	$AD_1 + 0$: Magnetkontenschacht 1	ML	MU	MC	2, 4
	MAGNB	$AD_1 + 1$: Magnetkontenschacht 2	ML	MU	MC	
	,OUT	$AD_1 = 0$: Auswurf ohne Schreiben				2, 4, 5, 2
	,IN	$AD_1 = 2$: Einzug ohne Lesen Zeile AD_{mr}				
	,MLOW	$AD_m + 4$ Auswurf nach unten				
	.	Bit 1-6: Anzahl-1 der zu übertragenden SW ab Anfang ALC-Bereich lt. Vorbefehl ALBG (0 - 31) Bit 1 bis 6: Anzahl - 1 der zu übertragenden Worte ab Anfang ALC-Bereich laut Vorbefehl ALBG (0 - 63)				
MAGN	$AD_1 + 0$: Magnetkontenschacht 1	ML	MU	MC	2, 4	
MAGNB	$AD_1 + 1$: Magnetkontenschacht 2	ML	MU	MC		
,WRITE	$AD_1 = 4$: Auswurf mit Schreiben				2, 4, 5, 2	
,READ	$AD_1 = 6$: Einzug mit Lesen					
,MLOW	$AD_m + 4$ Auswurf nach unten					
	Keine Karte da (bei Einzug) Karte richtig ausgeworfen (bei Auswurf) Falsch gelesen (bei Lesen) Leerkarte (bei Lesen)	ML ML	MU MU	MC		
3.10	MCPROC	Lesen $AD_r = \text{Anzahl SW} + 1$				
	,RUN1	$AD_m + 8$ eine Drehung der Trommel (nur mit MSKZ2)				
3.10	MCORG	$AD_{lm} = 0, 8$ Magnetkontenlistbefehl				4
	,SELZ	$AD_r = 0$: Setze ML, wenn Trommel frei				
	,SEL1	$AD_r = 1$: Ablage von Trommel nach Fach 1				
	,SEL2	$AD_r = 2$: Ablage von Trommel nach Fach 2				
	,SEL3	$AD_r = 3$: Ablage von Trommel nach Fach 3				
,RDO	$AD_1 = 1$: Lesen ohne Einzug, ohne Ablage Bit 1-6: Angabe der SW-1					

OP	Symbol	Funktion	Merker			Betriebs- programm
3.10	,RDF	AD ₁ = 2 : Einzug mit Lesen Bit 1-6 : Anzahl der SW-1				
	,SRD	AD _{1m} +0.0 : keine Ablage				
	,SR1E3	AD _{1m} +0.4 : mit Ablage nach Fach 1, bei Lesefehler nach Fach 3				
	,SR1ED	AD _{1m} +1.0 : mit Ablage nach Fach 1, bei Lesefehler Verbleib in der Trommel				
	,SR2E3	AD _{1m} +1.4 : mit Ablage nach Fach 2, bei Lesefehler nach Fach 3				
	,WRO	AD ₁ = 4 : Schreiben ohne Einzug Bit 1-6 : Anzahl der SW-1				
	,SWD	AD _{1m} +0.0 : keine Ablage				
	,SW1ED	AD _{1m} +0.4 : Ablage nach Fach 1, bei Schreibfehler Verbleib in der Trommel				
	,SW2ED	AD _{1m} +1.0 : Ablage nach Fach 2, bei Schreibfehler Verbleib in der Trommel				
			Die Merker werden wie folgt gesetzt	ML	MU	MC
		Leerkarte	0	1	1	
		Falsch gelesen / geschrieben	0	1	0	
		Karte klemmt	0	0	1	
		Keine Karte in der Trommel bei Aufruf zum Schreiben	1	0	0	
3.11	BPX	Laden des ALC-Bereichszeigers AD _{1m} : Adresse des Speicherwortes AD _r : Stellennummer (bewirkt AD _{1mr} + 0.4.0 → 1 ₃) (bewirkt AD _{1mr} + 0.5.0 → 1 ₃)				5, 5.1 - 5.7 1, 2, 3, 4
3.12	MGC	AD _{1mr} = 0.2.0 : Cassettenhauptbefehl (nur mit Folgebefehl anzuwenden) Folgebefehle : AD ₁ + 0 = Cassette 1 AD ₁ + 2 = Cassette 2 AD ₁ + 4 = Cassette 3 AD ₁ + 6 = Cassette 4				3, 4, 5, 5 3, 4 3, 4

OP	Symbol	Funktion	Merker			Betriebsprogramm
0.0	MGR	Folgebefehle für MGC (3.12) AD _{1mr} = 0.0.0: Rückspulen				3, 4, 5, 5
0.0	MGR.1	AD _{1mr} = 0.0.1: Reinigen des Kopfes				
0.1	MGO	AD _{1mr} = 0.0.0: Deckel öffnen				
0.2	MGRC	Bit 1 bis 8: Anzahl - 1 der zu übertragenden 8-Bit-Zeichen Einen Satz lesen				
0.3	MGWC	Einen Satz schreiben				
0.4	MGRFR	Bit 1 bis 8: Anzahl - 1 der zu berücksichtigenden Sätze Vorlauf um vorgegebene Anzahl von Sätzen				
0.5	MGRER	Rücklauf um vorgegebene Anzahl von Sätzen				
		Merker werden durch SMWT (2.12) gesetzt		MU	MC	
		Bandanfangs-, Bandendemarke Lese- oder Schreibvorgang fehlerhaft durchgeführt		MU	MC	
		Bandanfangs-, Bandendemarke und Lese-, oder Schreibvorgang fehlerhaft durchgeführt		MU	MC	
3.13	ALIN6	Eingabe über Serialdrucker 1 in ALC-Bereich ab (I ₃). Freigabe Position AD _{mr} - 1. AD ₁ = 0: im 6-Bit-Code speichern	ML	MU	MC	1, 2, 3, 4
	ALIN8	AD ₁ = 2: im 8-Bit-Code speichern	ML	MU	MC	1, 2, 3, 4
	, TWS	AD ₁ + 1: mit Druck auf Serialdrucker 2				
		Tabulationstaste Wagenaufzugtaste Auslösetaste		MU	MC	
		Endenzeichen 6.8 Endenzeichen 6.12 Endenzeichen 6.11	ML			
	ALPT	AD ₁ = 4: Lochen auf Streifenlocher Inhalt ALC-Bereich ab (I ₃) Bit 1 bis 7: Anzahl Bei Anzahl = 0: Beendigung nur durch Endenzeichen Bei Anzahl ≠ 0: Beendigung durch Endenzeichen oder Erreichen der Anzahl				1, 2, 3, 4, 5, 4
	, CD6	AD _m ≠ 0: im 6-Bit-Code gespeichert				1, 2, 3, 4

OP	Symbol	Funktion	Merker			Betriebs- programm	
3.13	,CD8	$AD_m + 8$: im 8-Bit-Code gespeichert				1, 2, 3, 4, 5, 4	
		$AD_1 + 0$: mit Lochen des Zeichens nach Endezeichen				1, 2, 3, 4, 5, 4	
	,WNI	$AD_1 + 2$: ohne Lochen des Zeichens nach Endezeichen				1, 2, 3, 4, 5, 4	
		Merker werden durch SMWT (2.12) gesetzt	ML	MU	MC		
		Endezeichen nicht aufgetreten		MU			
	RDPT RDPTS	,WNI	$AD_1 = 5$: Lesen mit Streifenleser in PCT-Eingabebereich ab Bereichsanfang				1, 3, 4, 5, 4
			Bit 1 bis 7: Anzahl Bei Anzahl = 0: Beendigung nur durch Endezeichen (PT-Code für 3.15) Bei Anzahl $\neq 0$: Beendigung durch Endezeichen oder Erreichen der Anzahl				
			$AD_m + 0$: Streifenleser 1		MU		1, 3, 4, 5, 4
			$AD_m + 8$: Streifenleser 2		MU		1, 3, 4
			$AD_1 + 0$: mit Lesen des Zeichens nach Endezeichen				1, 3, 4, 5, 4
$AD_1 + 2$: ohne Lesen des Zeichens nach Endezeichen						1, 3, 4, 5, 4	
		Endezeichen nicht aufgetreten		MU			
3.14	OPX	OP- und AD_1 - Teil des Befehls unter der Adresse AD_{lmr} nach Indexregister					
3.15		Unzulässiger Code					

Codes für Verarbeitung und Erzeugung von Überlochinformation s. Befehl PCF (3.8)																
Lesen												Lochen				
kein Überloch			Überloch 11			Überloch 12			Überloch 11 und 12			AD _m	Überloch 11		Überloch 12	
(A)	ML	MU	(A)	ML	MU	(A)	ML	MU	(A)	ML	MU		(A) >= 0	(A) < 0	(A) >= 0	(A) < 0
+	0	0	+	1	0	+	0	1	+	1	1	0	0	0	0	0
-	0	0	+	0	0	-	0	1	+	0	1	1	1	0	0	0
+	0	0	-	0	0	+	0	1	-	0	1	2	0	1	0	0
+	1	0	+	0	0	+	1	1	+	0	1	3	1	1	0	0
-	0	0	-	1	0	+	0	0	+	1	0	4	0	0	1	0
-	0	0	+	0	0	+	0	0	+	0	0	5	1	0	1	0
+	0	0	-	0	0	+	0	0	+	0	0	6	0	1	1	0
-	1	0	-	0	0	+	1	0	+	0	0	7	1	1	1	0
+	0	0	+	1	0	-	0	0	-	1	0	8	0	0	0	1
+	0	0	+	0	0	-	0	0	+	0	0	9	1	0	0	1
+	0	0	+	0	0	+	0	0	-	0	0	10	0	1	0	1
+	1	0	+	0	0	-	1	0	-	0	0	11	1	1	0	1
+	0	1	+	1	1	+	0	0	+	1	0	12	0	0	1	1
-	0	1	+	0	1	-	0	0	+	0	0	13	1	0	1	1
+	0	1	-	0	1	+	0	0	-	0	0	14	0	1	1	1
+	1	1	+	0	1	+	1	0	+	0	0	15	1	1	1	1

ALC-Code	Symbol	Deutschland	Frankreich	Schweiz	Groß-	Dänemark	Schweden	Puerto Rico	Italien	Griechenland	Korrespondenz-
		Osterreich	Belgien		britannien	Norwegen	Finnland	Spanien	USA		deutsch
		1000	884	882	863	880	881	917	912	956	982
0.0	0	0	0	0	0	0	0	0	0	0	0
0.1	1	1	1	1	1	1	1	1	1	1	1
0.2	2	2	2	2	2	2	2	2	2	2	2
0.3	3	3	3	3	3	3	3	3	3	3	3
0.4	4	4	4	4	4	4	4	4	4	4	4
0.5	5	5	5	5	5	5	5	5	5	5	5
0.6	6	6	6	6	6	6	6	6	6	6	6
0.7	7	7	7	7	7	7	7	7	7	7	7
0.8	8	8	8	8	8	8	8	8	8	8	8
0.9	9	9	9	9	9	9	9	9	9	9	9
0.10	YBLK										
0.11		-	-	-	-	-	-	-	-	-	a
0.12	YPLS	+)	+	+	+)))	-	L
0.13	YMIN	-	-	-	-	-	-	-	-	-	b
0.14	YITS	o	o	o	o	o	o	o	o	o	c
0.15	YITSM	o	%	o	o	o	o	o	o	o	d
1.0	YSTR	*	*	*	*	*	*	*	*	*	e
1.1	YSTRM	±	%	±	±	Q	Ö	N	&	&	f
1.2	A	A	A	A	A	A	A	A	A	A	B
1.3	B	B	B	B	B	B	B	B	B	B	C
1.4	C	C	C	C	C	C	C	C	C	C	D
1.5	D	D	D	D	D	D	D	D	D	D	E
1.6	E	E	E	E	E	E	E	E	E	E	F
1.7	F	F	F	F	F	F	F	F	F	F	G
1.8	G	G	G	G	G	G	G	G	G	G	H
1.9	H	H	H	H	H	H	H	H	H	H	I
1.10	I	I	I	I	I	I	I	I	I	I	J
1.11	J	J	J	J	J	J	J	J	J	J	K
1.12	K	K	K	K	K	K	K	K	K	K	L
1.13	L	L	L	L	L	L	L	L	L	L	M
1.14	M	M	M	M	M	M	M	M	M	M	N
1.15	N	N	N	N	N	N	N	N	N	N	N
2.0	O	O	O	O	O	O	O	O	O	O	O
2.1	P	P	P	P	P	P	P	P	P	P	P
2.2	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
2.3	R	R	R	R	R	R	R	R	R	R	R
2.4	S	S	S	S	S	S	S	S	S	S	S
2.5	T	T	T	T	T	T	T	T	T	T	T
2.6	U	U	U	U	U	U	U	U	U	U	U
2.7	V	V	V	V	V	V	V	V	V	V	V
2.8	W	W	W	W	W	W	W	W	W	W	W
2.9	X	X	X	X	X	X	X	X	X	X	X
2.10	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
2.11	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
2.12	YNMB	#	;	#	:	■	■	:	:	Q	h
2.13	YPNT	i
2.14	YCM
2.15	YSEM	;	±	;	±	#	#	±	±	C	.

Befehlszelle 0.0.0

AD : Kommagrunderstattung KA (0-15). Diese kommt beim Einschalten mit Komma- und C-Taste nach Indexregister I₅.

Bit 6 = 1: Ausgabe von Nachkommastellen falls (D) = 0.

Bit 9 = 1: Auslösetasten zur Beendigung der Freigabe für Serialdrucker erlaubt.

Bit 11 = 1: Rücktaste erlaubt.

Bit 12 = 1: Dezimalpunkt statt Dezimalkomma

Befehlszelle 0.0.2: Anfangsadresse der Lochkartencode-Tabelle.

Befehlszelle 0.0.3: Anfangsadresse der 1. Lochstreifencode-Tabelle.

Befehlszelle 0.0.4: Anfangsadresse der 2. Lochstreifencode-Tabelle.

ALC	L. komb.	Code	ALC	L. komb.	Code	ALC	L. komb.	Code
0.0	0	0.1.0	2.0	11-6	0.2.6	4.0	0-3-8	0.9.3
0.1	1	0.0.1	2.1	11-7	0.2.7	4.1	0-2-8	0.9.2
0.2	2	0.0.2	2.2	11-8	0.10.0	4.2	8-9	0.8.8
0.3	3	0.0.3	2.3	11-9	0.2.8	4.3	0-8-9	0.9.8
0.4	4	0.0.4	2.4	0-2	0.1.2	4.4		
0.5	5	0.0.5	2.5	0-3	0.1.3	4.5		
0.6	6	0.0.6	2.6	0-4	0.1.4	4.6		
0.7	7	0.0.7	2.7	0-5	0.1.5	4.7		
0.8	8	0.8.0	2.8	0-6	0.1.6	4.8		
0.9	9	0.0.8	2.9	0-7	0.1.7	4.9		
0.10			2.10	0-8	0.9.0	4.10		
0.11			2.11	0-9	0.1.8	4.11		
0.12	12-6-8	0.12.6	2.12	3-8	0.8.3	4.12		
0.13	11	0.2.0	2.13	12-3-8	0.12.3	4.13		
0.14	12-4-8	0.12.4	2.14	0-3-8	0.9.3	4.14		
0.15	12-7-8	0.12.7	2.15	0-6-8	0.9.6	4.15		
1.0	11-4-8	0.10.4	3.0	2-8	0.8.2	5.0		
1.1	11-7-8	0.10.7	3.1	11-2-8	0.10.2	5.1		
1.2	12-1	0.4.1	3.2	0-7-8	0.9.7	5.2		
1.3	12-2	0.4.2	3.3	7-8	0.8.7	5.3		
1.4	12-3	0.4.3	3.4	6-8	0.8.6	5.4		
1.5	12-4	0.4.4	3.5	0-5-8	0.9.5	5.5		
1.6	12-5	0.4.5	3.6	12-5-8	0.12.5	5.6		
1.7	12-6	0.4.6	3.7	11-5-8	0.10.5	5.7		
1.8	12-7	0.4.7	3.8	11-3-8	0.10.3	5.8		
1.9	12-8	0.12.0	3.9	12	0.4.0	5.9		
1.10	12-9	0.4.8	3.10	11-6-8	0.10.6	5.10		
1.11	11-1	0.2.1	3.11	0-4-8	0.9.4	5.11		
1.12	11-2	0.2.2	3.12	4-8	0.8.4	5.12		
1.13	11-3	0.2.3	3.13	5-8	0.8.5	5.13		
1.14	11-4	0.2.4	3.14	0-1	0.1.1	5.14		
1.15	11-5	0.2.5	3.15	12-2-8	0.12.2	5.15		
6.8	0-1-8	0.9.1	6.11	12-1-8	0.12.1	6.12	11-1-8	0.10.1

OP	Symbol	Funktion	Merker		
0.0	NOP	Leerbefehl AD_{1mr} : beliebig			
0.1	ACC	$(E) \rightarrow SW AD_{1m}$ AD_r : Anzahl der Nachkommastellen falls Eingabe ohne Komma SW C, Stelle 13: Anzahl der Vorkommastellen Stelle 15: Anzahl der Nachkommastellen Stelle 12, 14: gelöscht			
0.2	MVH	$(SW AD_r) \rightarrow SW AD_{1m}$			
0.3	MV	$(SW AD_{1m}) \rightarrow SW AD_r$			
0.4	ADH	$(SW AD_{1m}) + (SW AD_r) \rightarrow SW AD_{1m}$			MC
0.5	AD	$(SW AD_r) + (SW AD_{1m}) \rightarrow SW AD_r$			MC
0.6	SBH	$(SW AD_{1m}) - (SW AD_r) \rightarrow SW AD_{1m}$			MC
0.7	SB	$(SW AD_r) - (SW AD_{1m}) \rightarrow SW AD_r$			MC
0.8	MLH	$(SW AD_{1m}) \times (SW AD_r) \rightarrow SW AD_{1m}, C$			MC
0.9	ML	$(SW AD_r) \times (SW AD_{1m}) \rightarrow SW AD_r, C$ Bei $(C) \neq 0$ ist $MC = 1$			MC
0.10	DVH	$(SW AD_{1m}) : (SW AD_r) \rightarrow SW AD_{1m}, C$			MC
0.11	DV	$(SW AD_r) : (SW AD_{1m}) \rightarrow SW AD_r, C$ Bei Überlauf im Zielspeicherwort erfolgt Abbruch der Operation und Merker $MC = 1$			MC
0.12	CPH	Vergleich $(SW AD_{1m})$ mit $(SW AD_r)$	ML	MU	
0.13	CP	Vergleich $(SW AD_r)$ mit $(SW AD_{1m})$	ML	MU	
0.14	CPZ	Vergleich $(SW AD_{1m})$ mit Null Merker entsprechend Vergleichsergebnis	ML	MU	
0.15	CLR	$AD_r = 0$: Löschen $(SW AD_{1m})$			
	SGNIN	$AD_r = 1$: Vorzeichenwechsel in $SW AD_{1m}$			

OP	Symbol	Funktion	Merker		
1.0	BR	Sprung nach AD_{1mr}			
1.1	BR1	Sprung wenn $M1 = 1$ nach AD_{1mr}			
1.2	BR2	Sprung wenn $M2 = 1$ nach AD_{1mr}			
1.3	BR3	Sprung wenn $M3 = 1$ nach AD_{1mr}			
1.4	BR4	Sprung wenn $M4 = 1$ nach AD_{1mr}			
1.5	BR5	Sprung wenn $M5 = 1$ nach AD_{1mr}			
1.6	BRL	Sprung wenn $ML = 1$ nach AD_{1mr}			
1.7	BRU	Sprung wenn $MU = 1$ nach AD_{1mr}			
1.8	BRC	Sprung wenn $MC = 1$ nach AD_{1mr}			
1.9	BXG	Sprung wenn $(I) > 1023$ nach AD_{1mr}			
1.10	BXU	Sprung wenn $(I) \neq 0$ nach AD_{1mr}			
1.11	BRS	Unterprogrammprung nach AD_{1mr} (Max. 5 Unterprogrammstufen)			
1.12	BRR	Rückprung aus Unterprogramm nach Rückkehradresse + AD_{1mr} modulo 2^{11}			
1.13	SST	Substitution des Befehls unter der Adresse AD_{1mr}			
1.14	ICA	AD_{1mr} von (AD_{1mr}) nach SW A vor das Komma			MC
1.15	CA	AD_{1mr} nach SW A vor das Komma			MC
2.0	SR	Rechtshift (SW AD_{1m}) um AD_r Stellen			MC
2.1	SL	Linkshift (SW AD_{1m}) um AD_r Stellen			MC
2.2	XF XFR	$AD_1 = 0, AD_m \neq 0$: Indexregistervorbefehl für einen Befehl je Anweisung $AD_m + 12$: Indexregistervorbefehl bis zum nächsten Vorbefehl gültig AD_m : Indexregister für Indizierung (0 - 3) AD_r : Indexregister für Operation (0 - 3)			

OP	Symbol	Funktion	Merker		
2.2	MVX MVXH	$AD_1 = 0, AD_m : 0 - 3, AD_r : 0 - 3, 5$ $AD_m + 4 : (Ireg. AD_r) \rightarrow Ireg. AD_m$ $AD_m + 8 : (Ireg. AD_m) \rightarrow Ireg. AD_r$			
	CPX	$AD_1 = 1, AD_m : 0 - 3, AD_r : 0 - 3$ Vergleich (Ireg. AD_m) mit (Ireg. AD_r)	ML	MU	
	CLX	$AD_1 = 3, AD_m : 0 - 3, AD_r : 0 - 3$ $2048 - (Ireg. AD_r) \rightarrow Ireg. AD_m$			
	ADX	$AD_1 = 4, AD_m : 0 - 3, AD_r : 0 - 3$ $(Ireg. AD_m) + (Ireg. AD_r) \rightarrow Ireg. AD_m$			MC
	SBX	$AD_1 = 5, AD_m : 0 - 3, AD_r : 0 - 3$ $(Ireg. AD_m) - (Ireg. AD_r) \rightarrow Ireg. AD_m$			MC
	SRX	$AD_1 = 6, AD_m : 0 - 3, AD_r : 0 - 11$ Rechtsshift (Ireg. AD_m) um AD_r Stellen			MC
	SLX	$AD_1 = 7, AD_m : 0 - 3, AD_r : 0 - 11$ Linksshift (Ireg. AD_m) um AD_r Stellen			MC
2.3	RND	Runden (SW AD_{1m}) mit Stelle AD_r (Absolute Addition von 5 in Stelle AD_r , danach werden diese und die rechts anschließenden Stellen auf 0 gesetzt.)			MC
2.4	CNT	Zählen (SW AD_{1m}) in Stelle AD_r (Absolute Addition von 1 in Stelle AD_r)			MC
2.5	DC1 bis DC4	$AD_1 + 0$: Transport(I) nach SW AD_m ab Stelle AD_r zuzüglich AD_1 Stellen links anschließend.			MC
	,PL	$AD_1 + 4$: zusätzlich Addition vom angesprochenen Teil des SW			MC

OP	Symbol	Funktion	Merker		
2.6	XR1 bis XR4	AD ₁ + 0: Transport (SW AD _m) ab Stelle AD _r zu- zätzlich AD ₁ Stellen links nach Indexregister I _{lmr}			MC
	, PL	AD ₁ + 4: zusätzlich Addition des ursprünglichen Indexregisterinhalts			MC
2.7	XM1 bis XM4	AD ₁ + 0: Transport (SW AD _m) ab Stelle AD _r zu- zätzlich AD ₁ Stellen links nach Indexregister I _{lm} und I _r = 0			MC
	, PL	AD ₁ + 4: zusätzlich Addition des ursprünglichen Indexregisterinhalts			MC
2.8	XL1 bis XL4	AD ₁ + 0: Transport (SW AD _m) ab Stelle AD _r zu- zätzlich AD ₁ Stellen links nach Indexregister I ₁ und I _{mr} = 0			MC
	, PL	AD ₁ + 4: zusätzlich Addition des ursprünglichen Indexregisterinhalts			MC
2.9	ICX	AD _{lmr} von (AD _{lmr}) nach Indexregister			
2.10	CX	AD _{lmr} nach Indexregister			
2.11	PO SX	AD ₁ = 0, AD _m = 0: Kugelkopfposition nach Indexregister AD _r = 2: Serialdrucker 1			
	BL	AD ₁ = 4: Vorbefehl Blockumschaltung AD _m = 0: Festspeicherblock AD _r			
2.12	EFK WT SM1 bis SM5 SML bis SMU , ZERO , ONE	AD _{lmr} = 0, 7, 15: Funktionstastenauswurf AD ₁ = 0: Warte auf Bedingung AD _{mr} AD ₁ ≠ 0: Setze Merker AD ₁ auf 1, wenn Bedin- gung AD _{mr} erfüllt ist. AD _{mr} = 0, 0: Merker AD ₁ auf 0 setzen AD _{mr} = 3, 12: Merker AD ₁ auf 1 setzen	ML	MU	
	SMWT , MTW	Setze Merker ML, MU, MC nach Beendigung AD _{lmr} = 0, 10, 0: Ausgabe Simultangerät 1	ML	MU	MC
	WTC WTF	AD _{lmr} = 0, 0, 0: Warte auf Taste C (Lampe rot) AD _{lmr} = 0, 0, 1: Warte auf Taste F (Lampe orange)			

AD _{mr}	Symbol	Taste	AD _{mr}	Symbol	Taste	AD _{mr}	Symbol	Bedingung	
1.1	MLFM		1	2.1	MRAR	8.0	MTW	<u>Operation beendet :</u> <u>Simultangerät 1 :</u> Serialdrucker 1	
1.2	MLFMB		2	2.2	MLAR				
1.3	MCM		1	2.3	MRSK				
1.4	MCMB		2	2.4	MLSK				
				2.5	MFTS				
1.6	MLFLU		1	2.6	MITS				
1.7	MLFLL		2	2.7	MMIN				
1.8	MNA			2.8	MD				
1.9	MSTO			2.9	MP				
				2.10	MPML				
				2.11	MPCT				
				2.12	MPD				
				2.13	MS				
				2.14	MW				
OP	Symbol	Funktion					Merker		
2.13	TAB	Tabulation nach Position AD _{mr} AD ₁ = 0: Serialdrucker 1							
	TW	AD ₁ = 2: Freigabe Serialdrucker 1 bis Position AD _{mr} - 1					ML	MU	MC
2.14	LF	Zeilenschaltung Bit 1 bis 7: Anzahl der Zeilen							
	,LEP,UP	AD ₁ = 2: Vorschubeinrichtung 1 an Serialdrucker 1							
	,LEP,LOW	AD ₁ = 3: Vorschubeinrichtung 2 an Serialdrucker 1							
	,PLT	AD ₁ = 6: Walze von Serialdrucker 1							
2.15	ALC	Ausgabe eines Zeichens Bit 1 bis 7: ALC-Code des Zeichens							
	,TW1	AD ₁ + 1: Serialdrucker 1							
	,RED	AD _m + 8: Druck in Rot							
3.0	TT	Drucke Tabelle (bis Endezeichen 3.15) AD _{lmr} : Anfangsadresse der Tabelle							
		Merker werden durch SMWT (2.12) gesetzt					ML	MU	MC
		8	YTAB		MU				
12 (Tab. nach Ausgangspos.)	YCAR			MC					
11, sonst	YTRK	ML							

OP	Symbol	Funktion	Merker		
3.1	BUFA ABUF ,MVSN	Transport zwischen ALC-Bereich und SW A AD ₁ = 0 : vom ALC-Bereich ab (I ₃) nach A AD ₁ = 1 : von A nach ALC-Bereich ab (I ₃) AD ₁ + 2 : Transport mit Vorzeichen AD _m : Anzahl der Nachkommastellen AD _r : Anzahl der Stellen			
	BUFX1 BUFX2 XBUF1 XBUF2	AD ₁ = 4, AD _r = 0 : Transport vom ALC-Bereich ab (I ₃) nach Indexregister AD _r + 0 : eine Sedezimalzahl AD _r + 1 : zwei Sedezimalzahlen AD ₁ = 4, AD _r = 2 : Transport Inhalt Indexregister in den ALC-Bereich ab (I ₃) AD _r + 0 : eine Sedezimalzahl AD _r + 1 : zwei Sedezimalzahlen			
	ALOUT 8 ,TW1	AD ₁ = 4 : Drucken Inhalt ALC-Bereich ab (I ₃) bis Endezeichen Bit 1 bis 7 : maximale Anzahl der auszudruckenden Zeichen AD _m + 8 : im 8-Bit-Code gespeichert AD ₁ + 1 : Druck auf Serialdrucker 1			
		Merker werden durch SMWT (2 ₁₂) gesetzt	ML	MU	MC
		Ende durch 6.8 Ende durch 6.12 (Tab. nach Ausgangspos.) Ende durch 6.11 Ende durch Erreichen der Anzahl	ML	MU	MC
3.2	ED	Drucken Inhalt Speicherwort D1 (ohne Vorbefehl (D1) in Rot falls negativ) AD ₁ : Anzahl der Nachkommastellen AD _{mr} : Druckposition des Kommas			
3.3	LC LCX ,LEP, UP ,LEP, LOW ,PLT	Zeilenzähler-Zugriff Bit 1 bis 7 : Zeilennummer AD _m + 0 : Zeilenzähler laden AD _m + 8 : Inhalt Zeilenzähler nach I ₁ , Vergleich mit Zeilennummer AD ₁ = 2 : Vorschubeinrichtung 1 an Serialdrucker 1 AD ₁ = 3 : Vorschubeinrichtung 2 an Serialdrucker 1 AD ₁ = 6 : Walze von Serialdrucker 1	ML	MU	

OP	Symbol	Funktion	Merker		
3.4	VP ,LEP,UP ,LEP,LOW ,PLT	Vertikalpositionierung auf Zeile Bit 1 bis 7: Zeilennummer AD ₁ = 2: Vorschubeinrichtung 1 an Serialdrucker 1 AD ₁ = 3: Vorschubeinrichtung 2 an Serialdrucker 1 AD ₁ = 6: Walze von Serialdrucker 1			
3.5		Nur bei vorhandenem Modul benutzen!			
3.6	EDF ,RED	Druckvorbefehl für TW (2.13) ALOUT8 (3.1) sowie ALIN8 (3.13) AD _m = 0: Vorbefehl für Serialdrucker 1 AD _m + 8: Druck in Rot			
	EDF ,RED	Druckvorbefehl für ALC (2.15) AD _m = 0: Vorbefehl für Serialdrucker 1 AD _m + 8: Druck in Rot AD _r : 0-15: Anzahl der Wiederholungen			
	EDF ,RED	Druckvorbefehl für TT (3.0) AD _m = 0: Druck auf Serialdrucker 1 AD _m + 8: Druck in Rot			
	EDF ,ZERO ,STAR ,BLACK ,REDL ,RED ,REDG ,MIN ,SGN ,ITS ,FTS	Druckvorbefehl für ED (3.2) AD _r : Mindestanzahl der auszugebenden Zeichen vor dem Komma AD _m = 0: Druck (D1), Serialdrucker 1 AD _m + 0: Vormullen AD _m + 1: Sicherungssteme AD _m + 0: Druck in Schwarz AD _m + 4: Druck in Rot falls (D) negativ AD _m + 8: Druck in Rot AD _m + 12: Druck in Rot falls (D) positiv AD ₁ = 0: ohne Kennzeichen AD ₁ = 1: Leertaste oder Minuszeichen AD ₁ = 3: Kennzeichen + oder - AD ₁ = 5: Kennzeichen ◊ oder ◊ AD ₁ = 7: Kennzeichen ✕ oder ✕			

OP	Symbol	Funktion	Merker		
3.7		Nur bei vorhandenem Modul benutzen!			
3.8		Nur bei vorhandenem Modul benutzen!			
3.9	MVV	Bit 1 bis 10: Adresse des Speicherwortes AD ₁ + 0: Transport Inhalt von hoher Adresse nach SW A			
	MVVH	AD ₁ + 4: Transport Inhalt SW A nach hoher Adresse			
3.10		Nur bei vorhandenem Modul benutzen!			
3.11	BPX	Laden des ALC-Bereichszeichers AD _{1m} : Adresse des Speicherwortes AD _r : Stellennummer (bewirkt AD _{1mr} + 0.4.0 → I ₃)			
3.12		Nur bei vorhandenem Modul benutzen!			
3.13	ALIN8	Eingabe über Serialdrucker 1 in ALC-Bereich ab (I ₃). Freigabe Position AD _{mr} - 1. AD ₁ = 2: im 8-Bit Code speichern	ML	MU	MC
		Tabulationstaste Endezeichen 6.8 Wagenaufzugstaste Endezeichen 6.12 Auslösetaste Endezeichen 6.11	ML	MU	MC
	ALPT RDPT	Nur bei vorhandenem Modul benutzen! Nur bei vorhandenem Modul benutzen!			
3.14	OPX	OP- und AD ₁ - Teil des Befehls unter der Adresse AD _{1mr} nach Indexregister			
3.15		Unzulässiger Code			

Codes für Verarbeitung und Erzeugung von Überlochinformationen :																
Lesen												Lochen				
kein Überloch			Überloch 11			Überloch 12			Überloch 11 und 12			AD _m	Überloch 11		Überloch 12	
(A)	ML	MU	(A)	ML	MU	(A)	ML	MU	(A)	ML	MU		(A) ≥ 0	(A) < 0	(A) ≥ 0	(A) < 0
+	0	0	+	1	0	+	0	1	+	1	1	0	0	0	0	0
-	0	0	+	0	0	-	0	1	+	0	1	1	1	0	0	0
+	0	0	-	0	0	+	0	1	-	0	1	2	0	1	0	0
+	1	0	+	0	0	+	1	1	+	0	1	3	1	1	0	0
-	0	0	-	1	0	+	0	0	+	1	0	4	0	0	1	0
-	0	0	+	0	0	+	0	0	+	0	0	5	1	0	1	0
+	0	0	-	0	0	+	0	0	+	0	0	6	0	1	1	0
-	1	0	-	0	0	+	1	0	+	0	0	7	1	1	1	0
+	0	0	+	1	0	-	0	0	-	1	0	8	0	0	0	1
+	0	0	+	0	0	-	0	0	+	0	0	9	1	0	0	1
+	0	0	+	0	0	+	0	0	-	0	0	10	0	1	0	1
+	1	0	+	0	0	-	1	0	-	0	0	11	1	1	0	1
+	0	1	+	1	1	+	0	0	+	1	0	12	0	0	1	1
-	0	1	+	0	1	-	0	0	+	0	0	13	1	0	1	1
+	0	1	-	0	1	+	0	0	-	0	0	14	0	1	1	1
+	1	1	+	0	1	+	1	0	+	0	0	15	1	1	1	1

ALC-Code	Symbol	Deutschland	Italien	Korrespon-
		Österreich	USA	denzdeutsch
		1000	912	982
0,0	0	0	0	0
0,1	1	1	1	1
0,2	2	2	2	2
0,3	3	3	3	3
0,4	4	4	4	4
0,5	5	5	5	5
0,6	6	6	6	6
0,7	7	7	7	7
0,8	8	8	8	8
0,9	9	9	9	9
0,10	YBLK	-	-	-
0,11		-	-	-
0,12	YPLS	+)	o
0,13	YMIN	-	-	b
0,14	YITS	o	%	c
0,15	YITSM	o	\$	d
1,0	YSTR	*	*	e
1,1	YSTRM	±	&	f
1,2	A	A	A	g
1,3	B	B	B	A
1,4	C	C	C	B
1,5	D	D	D	C
1,6	E	E	E	D
1,7	F	F	F	E
1,8	G	G	G	F
1,9	H	H	H	G
1,10	I	I	I	H
1,11	J	J	J	I
1,12	K	K	K	J
1,13	L	L	L	K
1,14	M	M	M	L
1,15	N	N	N	M
2,0	O	O	O	N
2,1	P	P	P	O
2,2	Q	Q	Q	P
2,3	R	R	R	Q
2,4	S	S	S	R
2,5	T	T	T	S
2,6	U	U	U	T
2,7	V	V	V	U
2,8	W	W	W	V
2,9	X	X	X	W
2,10	Y	Y	Y	X
2,11	Z	Z	Z	Y
2,12	YNMB	#	:	Z
2,13	YPNT	.	.	h
2,14	YCM	,	,	i
2,15	YSEM	;	±	i

ALC-Code	Symbol	Deutschland	Italien	Korrespon-
		Österreich	USA	denzdeutsch
		1000	912	982
3,0	YCOL	:	?	k
3,1	YEXM	! ?	! #	l
3,2	YQEM	! ?	! #	m
3,3	YQOM	" "	<	n
3,4	YEQL	" "	>	o
3,5	YULN	-	!	p
3,6	YPOP	("	q
3,7	YPCL)	"	r
3,8	YDLR	\$	¢	s
3,9	YCAN	&	°	t
3,10	YPD	%	°	u
3,11	YPCT	%	=	v
3,12	YPMI	%	=	w
3,13	YAPH	'	'	x
3,14	YBAR	/	/	y
3,15	YECC	/	/	z
4,0	YCMPT	,	,	,
4,1	YPRG	s	(s
4,2	YCA	@	+	+
4,3	YHS	!	!	-
4,4	YM	m	m	?
4,5	YL	l	l	!
4,6	YT	t	t	i
4,7		!	!	d
4,8		!	!	e
4,9		!	!	u
4,10		!	!	b
4,11	YDIA	!	!	u
4,12		!	!	b
4,13		!	!	u
4,14		!	!	b
4,15	YMCB	m ₃	()
5,0	YST	ST	!	!
5,1		!	!	!
5,2		!	!	!
5,3		!	!	!
5,4	YMSQ	m ₂	!	!
5,5	YKG	kg	!	!
5,6		!	!	!
5,7		!	!	!
5,8		!	!	!
5,9	YG	g	!	!
5,10	YINV	v	!	!
5,11		!	!	!
5,12	YBS	!	!	!
5,13		!	!	!
5,14	YDCM	,	,	!
5,15	YDPNT	.	.	!

Befehlszelle 0.0.0

AD : Kommagrunderstattung KA (0-15). Diese kommt beim Einschalten mit Komma- und C-Taste nach Indexregister I₅.

Bit 6 = 1: Ausgabe von Nachkommastellen falls (D) = 0.

Bit 9 = 1: Auslösetasten zur Beendigung der Freigabe für Serialdrucker erlaubt.

Bit 11 = 1: Rücktaste erlaubt.

Bit 12 = 1: Dezimalpunkt statt Dezimalkomma

Befehlszelle 0.0.2: Anfangsadresse der Lochkartencode-Tabelle.

Befehlszelle 0.0.3: Anfangsadresse der 1. Lochstreifencode-Tabelle.

Befehlszelle 0.0.4: Anfangsadresse der 2. Lochstreifencode-Tabelle.

ALC	L. komb.	Code	ALC	L. komb.	Code	ALC	L. komb.	Code
0.0	0	0.1.0	2.0	11-6	0.2.6	4.0	0-3-8	0.9.3
0.1	1	0.0.1	2.1	11-7	0.2.7	4.1	0-2-8	0.9.2
0.2	2	0.0.2	2.2	11-8	0.10.0	4.2	8-9	0.8.8
0.3	3	0.0.3	2.3	11-9	0.2.8	4.3	0-8-9	0.9.8
0.4	4	0.0.4	2.4	0-2	0.1.2	4.4		
0.5	5	0.0.5	2.5	0-3	0.1.3	4.5		
0.6	6	0.0.6	2.6	0-4	0.1.4	4.6		
0.7	7	0.0.7	2.7	0-5	0.1.5	4.7		
0.8	8	0.8.0	2.8	0-6	0.1.6	4.8		
0.9	9	0.0.8	2.9	0-7	0.1.7	4.9		
0.10			2.10	0-8	0.9.0	4.10		
0.11			2.11	0-9	0.1.8	4.11		
0.12	12-6-8	0.12.6	2.12	3-8	0.8.3	4.12		
0.13	11	0.2.0	2.13	12-3-8	0.12.3	4.13		
0.14	12-4-8	0.12.4	2.14	0-3-8	0.9.3	4.14		
0.15	12-7-8	0.12.7	2.15	0-6-8	0.9.6	4.15		
1.0	11-4-8	0.10.4	3.0	2-8	0.8.2	5.0		
1.1	11-7-8	0.10.7	3.1	11-2-8	0.10.2	5.1		
1.2	12-1	0.4.1	3.2	0-7-8	0.9.7	5.2		
1.3	12-2	0.4.2	3.3	7-8	0.8.7	5.3		
1.4	12-3	0.4.3	3.4	6-8	0.8.6	5.4		
1.5	12-4	0.4.4	3.5	0-5-8	0.9.5	5.5		
1.6	12-5	0.4.5	3.6	12-5-8	0.12.5	5.6		
1.7	12-6	0.4.6	3.7	11-5-8	0.10.5	5.7		
1.8	12-7	0.4.7	3.8	11-3-8	0.10.3	5.8		
1.9	12-8	0.12.0	3.9	12	0.4.0	5.9		
1.10	12-9	0.4.8	3.10	11-6-8	0.10.6	5.10		
1.11	11-1	0.2.1	3.11	0-4-8	0.9.4	5.11		
1.12	11-2	0.2.2	3.12	4-8	0.8.4	5.12		
1.13	11-3	0.2.3	3.13	5-8	0.8.5	5.13		
1.14	11-4	0.2.4	3.14	0-1	0.1.1	5.14		
1.15	11-5	0.2.5	3.15	12-2-8	0.12.2	5.15		
6.8	0-1-8	0.9.1	6.11	12-1-8	0.12.1	6.12	11-1-8	0.10.1

OP	Symbol	Funktion	Merker		
2.12	WT SM1 bis SMU ,MMAGB	AD ₁ = 0 : Warte auf Bedingung AD _{mr} AD ₁ = 0 : Setze Merker AD ₁ auf 1, wenn Bedingung AD _{mr} erfüllt ist AD _{mr} = 8,8 : Karte da in Schacht 2			
2.14	LF ,MAGB ,BACK	Zeilenschaltung Bit 1 bis 7 : Anzahl der Zeilen AD ₁ = 1 : Magnetkontenschacht 2 AD _m + 8 : in Einzugsrichtung			
3.3	LC LCX ,MAGB	Zeilenzähler-Zugriff Bit 1 bis 7 : Zeilennummer AD _m + 0 : Zeilenzähler laden AD _m + 8 : Inhalt Zeilenzähler nach I, Vergleich mit Zeilennummer AD ₁ = 1 : Magnetkontenschacht 2	ML	MU	
3.4	VP ,MAGB	Vertikalpositionierung auf Zeile Bit 1 bis 7 : Zeilennummer AD ₁ = 1 : Magnetkontenschacht 2			
	ALBG	AD _{1m} + 0,8 : Bit 1 bis 7 : Anfang ALC-Bereich			
3.10	MAGNB ,OUT ,IN ,MCLOW	AD ₁ + 1 : Magnetkontenschacht 2 AD ₁ = 0 : Auswurf ohne Schreiben AD ₁ = 2 : Einzug ohne Lesen Zeile AD _{mr} AD _m + 4 : Auswurf nach unten	ML	MU	MC
	MAGNB ,WRITE ,READ	Bit 1 - 6 : 0 - 31 = 1-32 Worte ab Anfang ALC- Bereich werden übertragen AD ₁ + 1 : Magnetkontenschacht 2 AD ₁ = 4 : Auswurf mit Schreiben AD ₁ = 6 : Einzug mit Lesen (Pos. max. Druckzeilenanzahl - 4) Keine Karte da (bei Einzug) Karte richtig ausgeworfen (bei Auswurf) Fehler erkannt (bei Lesen) Leerkarte (bei Lesen) (Nach "Fehler erkannt" wird lediglich Merker MU gesetzt, es erfolgt keine Kontrolllesung)	ML ML	MU MU	MC MC

OP	Symbol	Funktion	Merker		
2.12	WT SM1 bis SM5 SML und SMU ,MMAG	AD ₁ = 0 : Warte auf Bedingung AD _{mr} AD ₁ ≠ 0 : Setze Merker AD ₁ auf 1, wenn Bedingung AD _{mr} erfüllt ist AD _{mr} = 8.4 : Karte da (im Schacht)			
2.14	LF ,ST	Zeilenschaltung bit 1 bis 7 : Anzahl der Zeilen AD ₁ = 0 : Steinhilbereinzug			
3.3	LC LCX ,ST	Zeilenzähler-Zugriff Bit 1 bis 7 : Zeilennummer AD _m + 0 : Zeilenzähler laden AD _m + 8 : Inhalt Zeilenzähler nach I, Vergleich mit Zeilennummer AD ₁ = 0 : Steinhilbereinzug	ML	MU	
3.4	VP ,ST	Vertikalpositionierung auf Zeile Bit 1 bis 7 : Zeilennummer AD ₁ = 0 : Steinhilbereinzug			
3.10	STN ,OUT ,IN ,CUT	AD ₁ + 0 : Steinhilbereinzug AD ₁ = 0 : Auswurf AD ₁ = 2 : Einzug AD ₁ = 4 : Stanzen Zeilenmarkierung			

OP	Symbol	Funktion	Merker		
			ML	MU	
2.12	WT SM1 bis SM5 SML und SMU ,MMAG ,MMAGB	$AD_1 = 0$: Warte auf Bedingung AD_{mr} $AD_1 \neq 0$: Setze Merker AD_1 auf 1, wenn Bedingung AD_{mr} erfüllt ist $AD_{mr} = 8,4$: Karte da in Schacht 1 $AD_{mr} = 8,8$: Karte da in Schacht 2	ML	MU	
2.14	LF ,INTR ,INTRB ,OUT1	Zeilenschaltung Bit 1 bis 7: Anzahl der Zeilen $AD_1 = 0$: Intromat Schacht 1 $AD_1 = 1$: Intromat Schacht 2 $AD_{mr} = 3,0$: Auswurf darf nicht <u>unmittelbar</u> vor Stanzen Zeilen- markierung liegen			
3.3	LC LCX ,INTR ,INTRB	Zeilenzähler-Zugriff Bit 1 bis 7: Zeilennummer $AD_m + 0$: Zeilenzähler laden $AD_m + 8$: Inhalt Zeilenzähler nach I, Vergleich mit Zeilennummer $AD_1 = 0$: Intromat Schacht 1 $AD_1 = 1$: Intromat Schacht 2	ML	MU	
3.4	VP ,INTR ,INTRB	Vertikalpositionierung auf Zeile Bit 1 bis 7: Zeilennummer $AD_1 = 0$: Intromat Schacht 1 $AD_1 = 1$: Intromat Schacht 2			
3.10	INT INTB INTO ,IN ,CUT	$AD_1 + 0$: Intromat Schacht 1 $AD_1 + 1$: Intromat Schacht 2 $AD_{lmr} + 0,0,1$: Intromat Schacht 1 und 2 $AD_1 = 2$: Einzug $AD_1 = 4$: Stanzen Zeilenmarkierung (bei der letzten bedruckbaren Zeile des Kontos erfolgt keine Stanzung, sondern Extern-Fehler-Stop			
	SORTMX	$AD_{lmr} = 6,0,0$: Sortimat-Wert nach Indexregister			

OP	Symbol	Funktion	Merker		
2.11	PBG , RD , PN	$AD_1 = 0, AD_m + 8$: Anfang des PCT-Bereichs vorgeben Bit 1 bis 7: Anfangsspeicherwort $AD_1 + 0$: Eingabebereich, RD-Bereich (zum Einlesen von Lochkarten) $AD_1 + 2$: Ausgabebereich, PN-Bereich (zum Stanzen von Lochkarten) (Der PN-Bereichsanfang fällt intern mit dem ALC-Bereichsanfang zusammen)			
	PBP PBPX , RD , PN	$AD_1 = 1$: PCT-Bereichszeiger-Zugriff Bit 1 bis 7: Nummer $AD_m + 0$: Laden des PCT-Bereichszeigers mit Nummer $AD_m + 8$: Inhalt des PCT-Bereichszeigers nach Indexregister, Vergleich mit Nummer $AD_1 + 0$: Eingabebereichszeiger, RDP $AD_1 + 2$: Ausgabebereichszeiger, PNP	ML	MU	
	MVBLK	$AD_1 = 6$: Löschen des PCT-Ausgabebereichs ab Inhalt Ausgabebereichszeiger PNP Bit 1 bis 7: Anzahl der Spalten (entspricht der Ausgabe von Leerspalten auf der Lochkarte. Für eine Spalte werden zwei Kernspeicherstellen gelöscht.)			
2.12	SMWT , MTWS	Setze Merker ML, MU, MC nach Beendigung $AD_{1mr} = 0.10.1$: Ausgabe Simultangerät 2	ML	MU	MC
	WT SM1 bis SM5 SML und SMU , MPC	$AD_1 = 0$: Warte auf Bedingung AD_{mr} $AD_1 \neq 0$: Setze Merker AD_1 auf 1, wenn Bedingung AD_{mr} erfüllt ist. $AD_{mr} = 8.1$: Operation beendet: Simultangerät 2: Kartenlocher 1	ML	MU	

OP	Symbol	Funktion	Merker		
			ML	MU	MC
3.7	PBA, PC	$AD_1 = 0$: Transport numerischer Zeichen zwischen dem PCT-Bereich und SW A entsprechend Lochkartenvorbefehl PCF (3.8) AD_m : Anzahl der Nachkommastellen (0 - KGA) AD_r : Anzahl der Zeichen (1-15)	ML	MU	MC
		Nicht numerisches Zeichen (Abbruch)			MC
	DEC, PC CD8 WNI	$AD_1 = 1$: Decodierung des PCT-Eingabebereichsinhalts (RD-Bereich) ab (RDP) nach PC-Tabelle Bit 1 bis 7: Anzahl Bei Anzahl = 0: Beendigung nur durch Endezeichen (PC-Code für 3.15, 6.8, 6.11 oder 6.12) Bei Anzahl \neq 0: Beendigung durch Endezeichen oder Erreichen der Anzahl $AD_m + 8$: im 8-Bit-Code nach ALC-Bereich ab (I_3) bringen. Bei Beendigung durch Endezeichen wird dieses decodiert und gespeichert. $AD_1 + 0$: Bei Ende durch Anzahl kein Endezeichen speichern. $AD_1 + 2$: Bei Ende durch Anzahl Endezeichen 6.8 speichern.		MU	MC
		Endezeichen nicht aufgetreten Nicht definiertes Zeichen (Abbruch)		MU	MC
	RDX, PC XPN, PC	$AD_1 = 2$: Transport eines alphanumerischen Zeichens zwischen PCT-Bereich und Indexregister. Umcodierung nach PC-Tabelle $AD_r = 0$: ((RDP) decodiert nach I $AD_r = 1$: (I) codiert nach (PNP)			MC
		Nicht definiertes Zeichen			MC
3.8	PCH, RD	$AD_1 = 0$: auf Gerät 1 eine Karte lesen AD_{mr} : Anzahl der Spalten der Karte			
	PCH, PN	$AD_1 = 2$: auf Gerät 1 eine Karte stanzen Beim Nixdorf-Locher ist (I_2) Prüfzeichen. AD_{mr} : Anzahl der zu stanzenen Spalten ab Kartenanfang.			

OP	Symbol	Funktion	Merker		
			ML	MU	MC
3.8		Merker werden durch SMWT (2.12) gesetzt	ML	MU	MC
		Lochvorgang fehlerhaft durchgeführt Fehler bei Kontrolllesung festgestellt Nur Kontrolllesung durchgeführt	ML	MU	MC
	PCF ,RD ,PN	Lochkartenvorbefehl für PBA, PC (3.7) AD ₁ = 4: Transport von RD-Bereich nach A AD ₁ = 6: Transport von A nach PN-Bereich AD _m : Code für Überlochverarbeitung (vgl. Tabelle Blatt L 9) AD _r : Stelle in SW A (1 - 15), die dazugehörige Lochkartenspalte trägt die Überlochung. (Der Vorbefehl PCF fällt intern mit PTFS und EDFS zusammen)			
	COD, PC ,CD8 ,WNI	AD ₁ = 5: Codierung des ALC-Bereichinhalts ab (I ₃) nach PC-Tabelle Bit 1 bis 7: Anzahl Bei Anzahl = 0: Beendigung nur durch Endezeichen Bei Anzahl ≠ 0: Beendigung durch Endezeichen oder Erreichen der Anzahl AD _m + 8: vom 8-Bit-Code nach PCT-Ausgabebereich ab (PNP) bringen AD ₁ + 0: mit Endezeichen (PC-Code für 6.8, 6.11, 6.12) AD ₁ + 2: ohne Endezeichen		MU	
		Endezeichen nicht aufgetreten		MU	

OP	Symbol	Funktion	Merker		
2.11	PBG ,RD	AD ₁ = 0, AD _m + 8: Anfang des PCT-Bereichs vorgeben Bit 1 bis 7: Anfangsspeicherwort AD ₁ + 0: Eingabebereich, RD-Bereich (zum Ein- lesen von Lochstreifen)			
	PBP PBPX ,RD	AD ₁ = 1: PCT-Bereichs-Zugriff Bit 1 bis 7: Nummer AD _m + 0: Laden des PCT-Bereichs-Zeigers mit Nummer AD _m + 8: Inhalt des PCT-Bereichs-Zeigers nach Indexregister, Vergleich mit Nummer AD ₁ + 0: Eingabebereichs-Zeiger, RDP	ML	MU	
2.12	SMWT ,MTWS	Setze Merker ML, MU, MC nach Beedigung AD _{lmr} = 0.10.1: Ausgabe Simultangerät 2	ML	MU	MC
	WT SM1 bis SM5 SML und SMU ,MPT ,MFPT ,MWPT ,MHPT	AD ₁ = 0: Warte auf Bedingung AD _{mr} AD ₁ ≠ 0: Setze Merker AD ₁ auf 1, wenn Be- dingung AD _{mr} erfüllt ist. AD _{mr} = 8.1: Operation beendet: Simultangerät 2: Streifenlocher 1 AD _{mr} = 4.0: <u>Meldungen</u> : Lochstreifenende 1 AD _{mr} = 4.2: <u>Stanzfehler bei</u> : Streifenlocher 1 AD _{mr} = 4.4: <u>Kein Kartenführungsloch</u> : Streifenlocher 1	ML	MU	
2.15	ALC ,PT	Ausgabe eines Zeichens Bit 1 bis 7: ALC-Code des Zeichens AD ₁ = 4: Streifenlocher 1			
3.5	EDPT ,PT	Lochen Inhalt Speicherwort D2 (ohne Vorbefehl nach Tabelle 1) AD ₁ = 4: Streifenlocher 1 AD _r : Anzahl der Nachkommastellen			
	PTF ,TBL1 ,TBL2	AD _m = 2: Lochstreifenvorbefehle AD _m + 0: Tabelle 1 AD _m + 8: Tabelle 2			
	PTF	Vorbefehl für ALC (2.15) AD _r : Anzahl der Wiederholungen			
	PTF	Vorbefehl für EDPT (3.5) AD _r : Mindestanzahl der auszugebenden Zeichen vor dem Komma			

OP	Symbol	Funktion	Merker		
3.5	,ZERO ,STAR ,BLNK ,MIN ,SGN ,ITS ,FTS	AD _m + 0: Vornullen AD _m + 1: Sicherungssterne AD _m + 4: Leertasten AD ₁ = 0: ohne Kennzeichen AD ₁ = 1: Leertaste oder Minuszeichen AD ₁ = 3: Kennzeichen + oder - AD ₁ = 5: Kennzeichen ◊ oder ◊ AD ₁ = 7: Kennzeichen ✕ oder ✕			
	PTF ,PT	Vorbefehl für ALPT (3.13) AD ₁ = 4: Streifenlocher 1			
	PTF ,CD5 ,YZI ,YBU	Vorbefehl für RDPT (3.13) AD _m + 0: 6-, 7-, 8-Spur-Lochstreifen AD _m + 1: 5-Spur-Lochstreifen AD _r = 0: nach Umschaltzeichen ZI AD _r = 8: nach Umschaltzeichen BU			
3.7	PBA, PT	AD ₁ = 4: Transport vom PCT-Eingabebereich (RDP) nach SW A und I Umcodierung nach PT-Tabelle AD _m : Anzahl der Nachkommastellen AD _r : max. Anzahl Ziffern, Vornullen, Sicherungssterne, Leertasten Falls Komma, dann kommagerechter Transport ohne Beachtung von AD _m . In SW 4 (C): Stelle 13: Anzahl der Vorkommastellen Stelle 15: Anzahl der Nachkommastellen Stellen 12, 14: gelöscht Beendigung durch Zeichen verschieden von Ziffer, Komma, Stern, Leertaste vor erster Zahl, danach durch Zeichen ungleich Ziffer, Komma, nach Komma durch Zeichen ungleich Ziffer. Dieses Zeichen nach I bei - ◊ ✕ kommt - nach SW A sonst +		MU	MC
		Anzahl der Zeichen von AD _r verschieden. Nicht definiertes Zeichen (Abbruch)		MU	MC
	DEC, PT	AD ₁ = 5: Decodierung PCT-Eingabebereichinhalt ab (RDP) nach PT-Tabelle Bit 1 bis 7: Anzahl Bei Anzahl = 0: Beendigung nur durch Endezeichen (PT-Code für 3.15)		MU	MC

Blatt L 20
1. 9. 1970

Datenverarbeitungssystem 820
820/15 Liste der Befehle des
PT-Moduls mit GP-Modul

OP	Symbol	Funktion	Merker		
3.7	,CD8 ,WNI	Bei Anzahl ≠ 0: Beendigung durch Endezeichen oder Erreichen der Anzahl AD _m + 8: im 8-Bit-Code nach ALC-Bereich ab (I ₃) bringen. AD ₁ + 0: mit Decodierung des Zeichens nach Endezeichen AD ₁ + 2: ohne Decodierung des Zeichens nach Endezeichen (speichern 6,8)			
		Endezeichen nicht aufgetreten Nicht definiertes Zeichen (Abbruch)		MU	MC
	RDX ,PT	AD ₁ = 6: Transport eines Zeichens vom PCT- Eingabebereich nach I AD ₁ = 0: ((RDP)) decodiert nach I			MC
	Nicht definiertes Zeichen			MC	
3.13	ALPT ,CD8 ,WNI	AD ₁ = 4: Lochen auf Streifenlocher Inhalt ALC-Bereich ab (I ₃) Bit 1 bis 7: Anzahl Bei Anzahl = 0: Beendigung nur durch Endezeichen Bei Anzahl ≠ 0: Beendigung durch Endezeichen oder Erreichen der Anzahl AD _m + 8: im 8-Bit-Code gespeichert AD ₁ + 0: mit Lochen des Zeichens nach Endezeichen AD ₁ + 2: ohne Lochen des Zeichens nach Ende- zeichen			
		Merker werden durch SMWT (2.12) gesetzt	ML	MU	MC
		Endezeichen nicht aufgetreten		MU	
	RDPT ,WNI	AD ₁ = 5: Lesen mit Streifenleser in PCT-Eingabe- bereich ab Bereichsanfang Bit 1 bis 7: Anzahl Bei Anzahl = 0: Beendigung nur durch Endezeichen (PT-Code für 3.15) Bei Anzahl ≠ 0: Beendigung durch Endezeichen oder Erreichen der Anzahl AD _m + 0: Streifenleser 1 AD ₁ + 0: mit Lesen des Zeichens nach Endezeichen AD ₁ + 2: ohne Lesen des Zeichens nach Endezeichen			MU
		Endezeichen nicht aufgetreten		MU	

OP	Symbol	Funktion	Merker		
2.11	PBG ,RD ,PN	AD ₁ = 0, AD _m + 8 : Anfang des PCT-Bereichs vorgeben Bit 1 bis 7 : Anfangsspeicherwort AD ₁ + 0 : Eingabebereich, RD-Bereich (zum Einlesen von Cassette) AD ₁ + 2 : Ausgabe aus PN-Bereich (Cassette)			
	PBP PBPX ,RD ,PN	AD ₁ = 1 : PCT-Bereich-Zugriff Bit 1 bis 7 : Nummer AD _m + 0 : Laden des PCT-Bereichszeichers mit Nummer AD _m + 8 : Inhalt des PCT-Bereichszeichers nach Indexregister, Vergleich mit Nummer AD ₁ + 0 : Eingabebereichszeichers, RDP AD ₁ + 2 : Ausgabebereichszeichers, PNP	ML	MU	
2.12	SMWT ,MTWS	Setze Merker MU, MC nach Beendigung AD _{1mr} = 0.10.1 : Ausgabe Simultangerät 2	ML	MU	MC
	WT SM1 bis SM5 SML und SMU ,MPT	AD = 0 : Warte auf Bedingung AD _{mr} AD ₁ ≠ 0 : Setze Merker AD ₁ auf 1, wenn Bedingung AD _{mr} erfüllt ist. AD _{mr} = 8.1 : Operation beendet: Simultangerät 2: Cassette			
3.12	MGC	AD _{1mr} = 0.2.0 : Cassettenhauptbefehl (nur mit Folgebefehl anzuwenden)			
		<u>Folgebefehle:</u> AD ₁ + 0 = Cassette 1 AD ₁ + 2 = Cassette 2			
	0.0 MGR 0.0 MGR, 1 0.1 MGO	AD _{1mr} = 0.0.0 : Rückspulen AD _{1mr} = 0.0.1 : Reinigen des Kopfes AD _{1mr} = 0.0.0 : Deckel öffnen			
		Bit 1 bis 8 : Anzahl - 1 der zu übertragenden 8-Bit-Zeichen			
	0.2 MGRC 0.3 MGWC	Einen Satz : lesen Einen Satz schreiben			
	0.4 MGRFR 0.5 MGRBR	Vorlauf um vorgegebene Anzahl von Sätzen Rücklauf um vorgegebene Anzahl von Sätzen			

Blatt L 22
 1. 9. 1970

Datenverarbeitungssystem 820
 820/15 Liste der Befehle des
 TC-Moduls mit GP-Modul

OP	Symbol	Funktion	Merker		
		Merker werden durch SMWT (2.12) gesetzt Bandanfags-, Bandendemarke Lese- oder Schreibvorgang fehlerhaft durchgeführt Bandanfags-, Bandendemarke und Lese- oder Schreibvorgang fehlerhaft durchgeführt	ML	MU MU	MC MC MC

OP	Symbol	Funktion	Merker		
2.11	PBG , RD , PN	$AD_1 = 0, AD_m + 8$: Anfang des PCT-Bereich vorgeben Bit 1 bis 7: Anfangsspeicherwort $AD_1 + 0$: Eingabebereich. RD-Bereich (zum Einlesen von Lochkarten, Lochstreifen und Cassette) $AD_1 + 2$: Ausgabebereich, PN-Bereich (zum Stanzen von Lochkarten sowie Ausgabe auf Cassette) (Der PN-Bereichsanfang fällt intern mit dem ALC-Bereichsanfang zusammen)			
	PBP PBPX , RD , PN	$AD_1 = 1$: PCT-Bereichszeiger - Zugriff Bit 1 bis 7: Nummer $AD_m + 0$: Laden des PCT-Bereichszeigers mit Nummer $AD_m + 8$: Inhalt des PCT-Bereichszeigers nach Indexregister, Vergleich mit Nummer $AD_1 + 0$: Eingabebereichszeiger, RDP $AD_1 + 2$: Ausgabebereichszeiger, PNP	ML	MU	
	MVBLK	$AD_1 = 6$: Löschen des PCT-Ausgabebereichs ab Inhalt Ausgabebereichszeiger PNP Bit 1 bis 7: Anzahl der Spalten (entspricht der Ausgabe von Leerspalten auf der Lochkarte. Für eine Spalte werden zwei Stellen im Speicherwort gelöscht.)			
	BL , LIV	$AD_1 = 4$: Vorbefehl Blockumschaltung $AD_m = 1$: Programmblock im Magnetkernsp. AD_r			
	MVOX	$AD_1 = 2, AD_m : 0-3, AD_r : 0-3$ (Ireg. AD_m) = OP MC = AD_i (Ireg. AD_r) = AD_{1mr}	Ergibt einen neuen Befehl		
	$AD_m + 8$: Befehlsspeicherung: Der neue Befehl wird unter Adresse (I_3) in dem durch Vorbefehl Blockumschaltung bestimmten Programmblock im Magnetkernspeicher gespeichert.				
2.12	SMWT , MTWS	Setze Merker ML, MU, MC nach Beendigung $AD_{1mr} = 0.10.1$: Ausgabe Simultangerät 2	ML	MU	MC

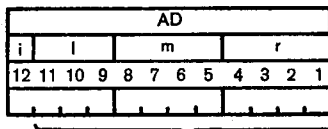
OP	Symbol	Funktion	Merker		
			ML	MU	
2.12	SM1 bis SMU ,MFPT ,MWPT ,MHPT	$AD_1 \neq 0$: Setze Merker AD_1 auf 1, wenn Bedingung AD_{mr} erfüllt ist. $AD_{mr} = 4.0$: <u>Meldungen</u> : Lochstreifenende 1 $AD_{mr} = 4.2$: <u>Stanzfehler bei</u> : Streifenlocher 1 $AD_{mr} = 4.4$: <u>Kein Kartenführungsloch</u> : Streifenlocher 1	ML	MU	
3.4	ALBG	$AD_{lm} + 0.8$: Bit 1 bis 7: Anfang ALC-Bereich			
3.9	EP	Als Vorbefehl für WT, SEP. 1. Betriebsprogramm MFA 1 AD_{lmr} = Adresse des letzten Speicherwortes im Arbeitsbereich für das Anwenderprogramm 2. Betriebsprogramm MFA 1 AD_{lmr} = Adresse des letzten Speicherwortes kleiner 123 im ersten Teil des Arbeitsbereichs für das Anwenderprogramm Transport des Speicherwortes nach A			
2.12	WT, SEP	$AD_{lmr} = 0.4.12$: Setzen Anfang des Befehlsbereiches Befehl muß unmittelbar auf Vorbefehl EP folgen.			
2.12	WT, CEP	$AD_{lmr} = 0.4.11$: Freigabe des Befehlsbereiches als Arbeitsbereich für das Anwenderprogramm			

Befehlsbeschreibung für die Serie 820

Die folgende Befehlsbeschreibung umfaßt die Befehle der Modelle 820/15/25/35.

Da Unterschiede zwischen den Modellen 820/15 und 820/25 bzw. 820/35 bestehen, ist festgesetzt, daß Aussagen in eckigen Klammern[] nur für das Modell 820/15, Aussagen in geschweiften Klammern{ } nur für die Modelle 820/25 und 820/35 gelten. Aussagen ohne Klammern treffen für alle Modelle zu.

NOP	Leerbefehl (no operation)	0.0
-----	---------------------------	-----



AD_{lmr}: beliebig

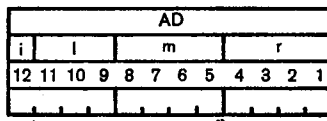
Wirkung:

Der Befehl "NOP Leerbefehl (0.0)" führt keine Operation aus. Er wird ohne Auswirkungen durchlaufen und reserviert beispielsweise Platz für weitere Befehle, die der Programmierer erst später in das Programm einfügen kann oder ersetzt bei Programmänderungen die Befehle, die nunmehr überflüssig sind und ausfallen.

Bemerkung:

Ist der Befehl "NOP Leerbefehl (0.0)" indiziert, so wird die Indizierungs-Anweisung eines zuvor gegebenen einfachen Indexvorbefehls gelöscht (vgl. Befehl "XF und XFR Indexvorbefehl (2.2)").

ACC	Eingabe (accept)	0.1
-----	------------------	-----



Anzahl der Nachkommastellen
(0 - KA)

Speicherwort {0.1 - 7.10}
D.1 - 7.10

Wirkung:

1. Die auf der Zehnertastatur eingegebene Zahl wird in das durch AD_lm benannte Speicherwort transportiert, wobei in die Vorzeichenstelle 0 des Speicherwortes der Code 0 für ein positives Vorzeichen kommt.
2. Die eingegebene Zahl wird vom Komma ab um die in AD_r angegebene Anzahl von Stellen nach rechts verschoben. Bei Tastaturen mit Komma-Taste gilt dies für die Eingabe ohne Komma. Bei einer Eingabe mit Komma hat dieses jedoch Vorrang und AD_r wird nicht beachtet.
3. In die Stellen 12 bis 15 des Speicherwortes 4 werden zur Kontrolle die Anzahl der eingetasteten Ziffern entsprechend ihrer Stellung im Speicherwort gespeichert, und zwar kommt die Anzahl der Nachkommastellen sedezimal in die Stelle 15, die Anzahl der Vorkommastellen sedezimal in die Stelle 13. Die Stellen 12 und 14 werden gelöscht, während alle weiteren Stellen von Speicherwort 4 unverändert bleiben.
4. Es gibt Tastaturen mit den Tasten 00 und 000. Die Betätigung einer solchen Taste hat die gleiche Wirkung wie das zwei- bzw. dreimalige Betätigen der Taste 0.
5. Die englische Tastatur ist mit den Zifferntasten 10, 11 und 12 ausgerüstet. Sie dienen dazu, eine der Zahlen 10, 11 oder 12 in eine Stelle des Speicherwortes zu bringen. Damit sind also auch Eingaben möglich, die keine Übernahme von Dezimalzahlen zur Folge haben.

ACC	Eingabe (accept)	0.1
-----	------------------	-----

Beispiel: (KA 5)

Über die Zehnertastatur sind folgende Ziffern der Reihe nach eingetastet worden:
1 2 3 4 5 6 7 8 9 00 7 0. Die Eingabe soll nach Speicherwort 17 erfolgen.

vorher: SW 4

-	0	0	0	0	0	0	0	0	0	0	0	6	4	2	5	7	4
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
0	0 1 0	1	1 1 3		ACC, 17.3
1					
2					

nachher: SW 17

+	1	2	3	4	5	6	7	8	9	0	0	7	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

SW 4

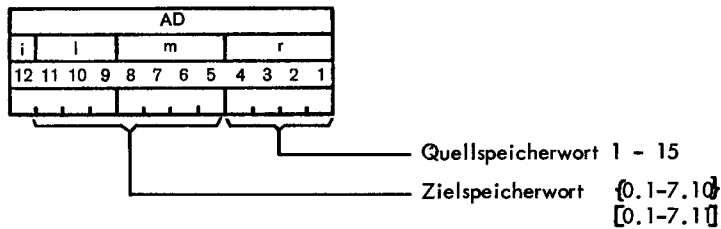
-	0	0	0	0	0	0	0	0	0	0	6	4	0	10	0	3
---	---	---	---	---	---	---	---	---	---	---	---	---	---	----	---	---

Im Speicherwort 17 steht also die Dezimalzahl 1234567890,070

Bemerkungen:

- Es kann auch eine Eingabe nach Speicherwort 4 befohlen werden. Dabei ist folgende Verfahrensweise zu beachten:
Das Speicherwort 4 wird als Zielspeicherwort gelöscht. Danach werden in die Stellen 13 und 15 die Anzahl der Vor- und Nachkommastellen eingeschrieben. Schließlich erfolgt die Übernahme der Eingabe in die angesprochenen Stellen, wobei die übrigen Stellen des Speicherwortes unverändert bleiben.
- Das Programm läuft jedesmal auf Externfehlerstop, wenn die Anzahl der eingetasteten Ziffern unzulässig ist. Nach Drücken der C-Taste ist die Eingabe gelöscht, und es kann erneut eingegeben werden. Die Betätigung einer beliebigen Auslösetaste bewirkt dann eine Wiederholung des Befehls "ACC Eingabe (0.1)".

MVH	Transport (move to high register)	0.2
-----	-----------------------------------	-----



Wirkung:

Der Inhalt des durch AD_r benannten Speicherwortes (Quellspeicherwort) wird einschließlich Vorzeichen in das durch $AD_l m$ benannte Speicherwort (Zielspeicherwort) übertragen. Der Inhalt des Quellspeicherwortes bleibt unverändert.

Beispiel:

Transport (SW 6) → SW 34 bei dezimalem Speicherwortinhalt.

vorher:

SW 6	+ 0 0 0 0 0 0 0 0 0 0 0 4 7 1 1 0
SW 34	- 0 0 0 0 0 0 0 0 0 7 3 6 5 2 4 1

BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
0	0 2	0	2 2 6		MVH, 34.6
1					

nachher:

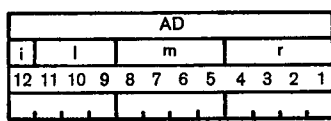
SW 6	+ 0 0 0 0 0 0 0 0 0 0 0 4 7 1 1 0
SW 34	+ 0 0 0 0 0 0 0 0 0 0 0 4 7 1 1 0

--	--	--

Bemerkungen:

1. Der Transport geschieht stellengerecht mit jeweils 4 Bit, d.h. es können auch sedezimale Stelleninhalte transportiert werden. Dies gilt für alle 16 Kernspeicherstellen, also einschließlich der Vorzeichenstelle.
2. Der Befehl ist auch zulässig, wenn für Quell- und Zielspeicherwort dasselbe Speicherwort genommen wird.
3. Man beachte die Vertauschung von Quell- und Zielspeicherwort im Befehl "MV Transport (0.3)".

MV	Transport (move to low register)	0.3
----	----------------------------------	-----



Zielspeicherwort 1 - 15
Quellspeicherwort ~~0.1 - 7.10~~
0.1 - 7.11

Wirkung:

Der Inhalt des durch AD_{lm} benannten Speicherwortes (Quellspeicherwort) wird einschließlich Vorzeichen in das durch AD_r benannte Speicherwort (Zielspeicherwort) übertragen. Der Inhalt des Quellspeicherwortes bleibt unverändert.

Beispiel:

Transport (SW 3) → SW 5 bei sedezimalem Speicherwortinhalt.

vorher: SW 3 7 0 0 0 0 0 0 0 0 0 0 0 4 7 13 1 10

SW 5 11 0 0 0 0 0 0 0 0 0 7 3 6 5 1 4 2

BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
0	0 3	0	0 3 5		MV, 3.5
1					

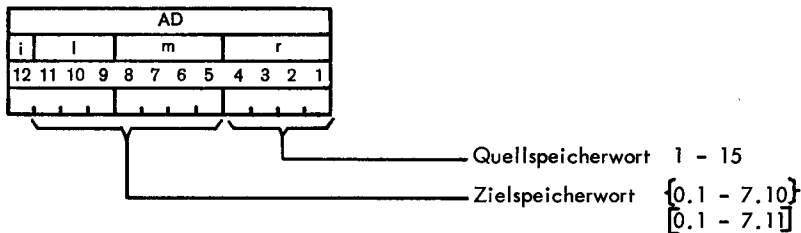
nachher: SW 3 7 0 0 0 0 0 0 0 0 0 0 0 4 7 13 1 10

SW 5 7 0 0 0 0 0 0 0 0 0 0 0 4 7 13 1 10

Bemerkung:

Für den Befehl "MV Transport (0.3)" gelten die Bemerkungen 1 und 2 des Befehls "MVH Transport (0.2)". Man beachte bei diesem Befehl aber die Vertauschung von Ziel- und Quellspeicherwort.

ADH	Addition (addition to high register)	0.4
-----	--------------------------------------	-----



Wirkung:

- Der Inhalt des durch AD_r benannten Speicherwortes (Quellspeicherwort) wird zum Inhalt des durch AD_l benannten Speicherwort (Zielspeicherwort) addiert. Das Ergebnis steht anschließend im Zielspeicherwort. Der Inhalt des Quellspeicherwortes bleibt unverändert.
- Findet bei der Addition ein Überlauf über Stelle 1 des Zielspeicherwortes statt, wird Merker $MC = 1$ andernfalls $MC = 0$ gesetzt.

Beispiel 1:

Addition (SW 17) + (SW 8) → SW 17

vorher:

SW 8	+ 0 0 0 0 0 0 0 0 0 0 0 1 6 0 0 0 0 0
SW 17	- 0 0 0 0 0 0 0 0 0 0 0 7 0 0 0 0 0 0

BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
0	0 4	0	1 1 8		ADH, 17. 8
1					

nachher:

SW 8	+ 0 0 0 0 0 0 0 0 0 0 0 1 6 0 0 0 0 0
(kein Überlauf) SW 17	+ 0 0 0 0 0 0 0 0 0 0 0 9 0 0 0 0 0 0

MC = 0

Bei der Addition wurde das negative Vorzeichen der Zahl in Speicherwort 17 berücksichtigt.

ADH	Addition (addition to high register)	0.4
-----	--------------------------------------	-----

Beispiel 2:

Addition (SW 10) + (SW 10) → SW 10

vorher: SW 10

+	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen				
0	0	4	0	0	10	10			ADH, 10, 10
1									

nachher: SW 10

+	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

MC = 1
(Überlauf)

Bemerkungen:

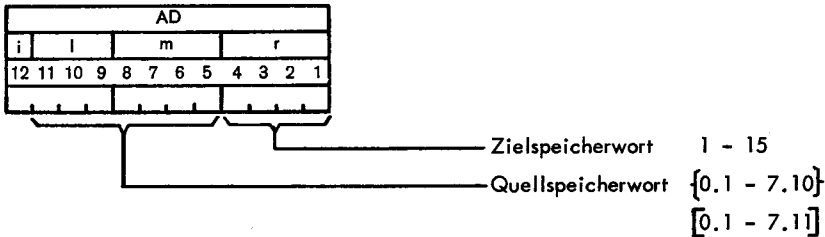
- Der Befehl ist auch zulässig, wenn für Quell- und Zielspeicherwort dasselbe Speicherwort genommen wird (vgl. Beispiel 2).
- Die Addition wird auch ausgeführt, wenn in einer Kernspeicherstelle eine Zahl größer als 9 steht. Eine solche Zahl verursacht einen Übertrag von 1 in die davorliegende Stelle. Fehler treten also dann auf, wenn die Summe in einer Stelle größer als 19 wird.

(es ist z.B. $\boxed{11} + \boxed{6} = \boxed{17}$ aber $\boxed{12} + \boxed{9} = \boxed{111}$).

Es wird also kaum Sinn haben, diesen Befehl auf nichtdezimale Speicherwortinhalte anzuwenden.

- Als Ergebnis der Addition kann -0 entstehen. Eine -0 wird rot gedruckt, wenn negative Zahlen rot gedruckt werden sollen. In allen anderen Fällen wird -0 wie +0 ausgewertet.
- Man beachte die Vertauschung von Quell- und Zielspeicherwort im Befehl "AD Addition (0.5)".

AD	Addition (addition to low register)	0.5
----	-------------------------------------	-----



Wirkung:

1. Der Inhalt des durch AD_m benannten Speicherwortes (Quellspeicherwort) wird zum Inhalt des durch AD_r benannten Speicherwortes (Zielspeicherwort) addiert. Das Ergebnis steht anschließend im Zielspeicherwort. Der Inhalt des Quellspeicherwortes bleibt unverändert.
2. Findet bei der Addition ein Überlauf über Stelle 1 des Zielspeicherwortes statt, wird Merker $MC = 1$ andernfalls $MC = 0$ gesetzt.

Beispiel:

Addition (SW 6) + (SW 18) → SW 6

vorher: SW 6

-	7	6	5	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

SW 18

-	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

BW-Adresse	Op. Teil	i	Adreßteil	Symb. Adr.	Bemerkungen
	0 0 5	0	1 2 6		AD, 18, 6
	1				

nachher: SW 6

-	1	9	7	1	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

MC = 1

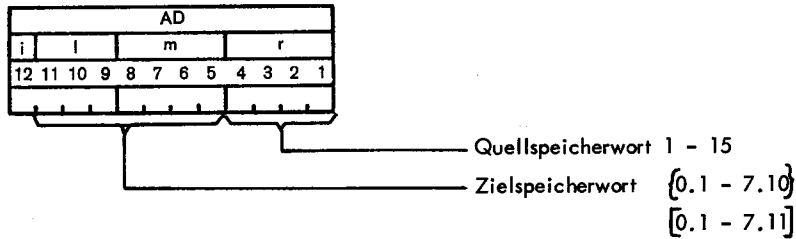
(Überlauf) SW 18

-	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Bemerkung:

Für den Befehl "AD Addition (0.5)" gelten die Bemerkungen 1 bis 3 des Befehls "ADH Addition (0.4)". Man beachte bei diesem Befehl jedoch die Vertauschung von Quell- und Zielspeicherwort.

SBH	Subtraktion (subtraction to high register)	0.6
-----	--	-----



Wirkung:

1. Der Inhalt des durch AD_r benannten Speicherwortes (Quellspeicherwort) wird vom Inhalt des durch $AD_l m$ benannten Speicherwortes (Zielspeicherwort) subtrahiert. Das Ergebnis steht anschließend im Zielspeicherwort. Der Inhalt des Quellspeicherwortes bleibt unverändert.
2. Findet bei der Subtraktion ein Überlauf über Stelle 1 des Zielspeicherwortes statt, wird Merker $MC = 1$ andernfalls $MC = 0$ gesetzt.

Beispiel 1:

Subtraktion (SW 20) - (SW 10) → SW 20.

vorher:	SW 20	+ 0 0 0 0 0 0 0 0 0 0 1 2 0 0 0 0 0 0
	SW 10	+ 0 0 0 0 0 0 0 0 0 0 0 8 0 0 0 0 0 0

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
	0 0 6 0		1 4 10		SBH, 20. 10
	1				

nachher:	SW 20	+ 0 0 0 0 0 0 0 0 0 0 0 4 0 0 0 0 0 0
MC = 0		
(kein Überlauf)	SW 10	+ 0 0 0 0 0 0 0 0 0 0 0 8 0 0 0 0 0 0

SBH	Subtraktion (subtraction to high register)	0.6
-----	--	-----

Beispiel 2:

Subtraktion (SW 5) - (SW 5) → SW 5.

vorher: SW 5

-	0	0	1	2	3	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
0	0 6	0	0 5 5		SBH, 5.5
1					

nachher: SW 5

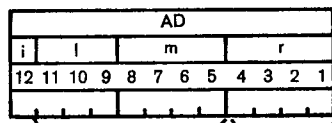
-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

MC = 0
(kein Überlauf)

Bemerkungen:

1. Der Befehl ist auch zulässig, wenn für Quell- und Zielspeicherwort dasselbe Speicherwort genommen wird (vgl. Beispiel 2).
2. Die Subtraktion wird auch ausgeführt, wenn in einer Kernspeicherstelle eine Zahl größer als 9 steht. Fehler treten aber auf, wenn eine solche Zahl von einer anderen subtrahiert wird, die um mehr als 10 kleiner ist (z.B. im Fall 3 - 14). Es wird also kaum Sinn haben, diesen Befehl auf sedezimale Speicherwortinhalt anzuwenden.
3. Als Ergebnis einer Subtraktion kann -0 entstehen (vgl. Beispiel 2). Eine -0 wird rot gedruckt, wenn negative Zahlen rot gedruckt werden sollen. In allen anderen Fällen wird -0 wie +0 ausgewertet.
4. Man beachte die Vertauschung von Quell- und Zielspeicherwort im Befehl "SB Subtraktion (0.7)".

SB	Subtraktion (subtraction to low register)	0.7
----	---	-----



Zielspeicherwort 1 - 15
 Quellspeicherwort $\left\{ \begin{matrix} 0.1 - 7.10 \\ 0.1 - 7.11 \end{matrix} \right\}$

Wirkung:

1. Der Inhalt des durch AD_m benannten Speicherwortes (Quellspeicherwort) wird vom Inhalt des durch AD_r benannten Speicherwortes (Zielspeicherwort) subtrahiert. Das Ergebnis steht anschließend im Zielspeicherwort. Der Inhalt des Quellspeicherwortes bleibt unverändert.
2. Findet bei der Subtraktion ein Überlauf über Stelle 1 des Zielspeicherwortes statt, wird Merker MC = 1 andernfalls MC = 0 gesetzt.

Beispiel:

Subtraktion (SW 10) - (SW 19) → SW 10

vorher:	SW 19	+ 0 0 0 0 0 0 0 0 0 0 7 0 0 0 0 0 0
	SW 10	- 0 0 0 0 0 0 0 0 1 8 0 0 0 0 0 0 0

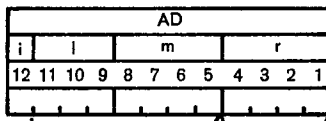
BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
	0 0 7	0	1 3 10		SB, 19.10

nachher:	SW 19	+ 0 0 0 0 0 0 0 0 0 0 7 0 0 0 0 0 0
	MC = 0	
	(Kein Überlauf) SW 10	- 0 0 0 0 0 0 0 0 2 5 0 0 0 0 0 0 0

Bemerkung:

Für den Befehl "SB Subtraktion (0.7)" gelten die Bemerkungen 1 bis 3 des Befehls "SBH Subtraktion (0.6)". Man beachte bei diesem Befehl jedoch die Vertauschung von Quell- und Zielspeicherwort.

MLH	Multiplikation (multiplication to high register)	0.8
-----	---	-----



Quellspeicherwort 1 - 15
 Zielspeicherwort $\{0.1 - 7.10\}$
 $[0.1 - 7.11]$

Wirkung:

1. Der Inhalt des Zielspeicherwortes wird mit dem Inhalt des Quellspeicherwortes multipliziert. Das Ergebnis steht anschließend kommagerecht und vorzeichenrichtig im Zielspeicherwort. Der Inhalt des Quellspeicherwortes bleibt unverändert.
2. Das Ergebnis einer Multiplikation hat doppelt soviel Stellen nach dem Komma wie die Faktoren. Im Zielspeicherwort werden aber nur so viele Stellen davon angegeben, wie in der KA festgelegt sind. Die überzähligen Nachkommastellen gehen verloren, wobei keine Rundung stattfindet.
3. Ist das Ergebnis der Multiplikation so groß, daß es im Zielspeicherwort allein nicht mehr dargestellt werden kann, so wird der Überlauf rechtsbündig mit gleichem Vorzeichen wie das Ergebnis in das Speicherwort C (= SW 4, Carry-Speicherwort) gestellt. Bei Überlauf wird MC = 1 andernfalls MC = 0 gesetzt.

Beispiel 1: (KA 3)

Multiplikation (SW 30) x (SW 9) → SW 30

vorher:	SW 9	+ 0 0 0 0 0 0 0 0 0 0 0 0 4 0 0 0 0 0
	SW 30	+ 0 0 0 0 0 0 0 0 0 0 0 0 0 2 5 0 0
	SW 4	- 0 0 0 0 0 0 0 0 0 0 0 3 5 7 9 6 1

BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
	0 0 8	0	1 14 9		MLH, 30.9
	1				

MLH	Multiplikation (multiplication to high register)	0.8
-----	---	-----

nachher: SW 9 + 0 0 0 0 0 0 0 0 0 0 0 4 0 0 0 0 0 0

SW 30 + 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0

MC = 0: kein Überlauf

SW 4 + 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Beispiel 2: (KA 5)

Multiplikation (SW 3) x (SW 24) → SW 24 (mit verschiedenen Vorzeichen)

vorher: SW 3 - 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0

SW 24 + 0 0 0 0 0 0 8 0 0 0 0 0 0 0 0 0 0 0

SW 4 +

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
	0 0 8 0		1 8 3		MLH, 24, 3
	1				

nachher: SW 3 - 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0

SW 24 - 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

MC = 1: Überlauf

SW 4 - 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

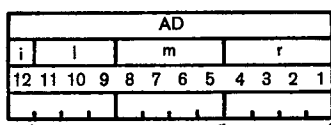
Bemerkungen:

1. Man beachte, daß der Befehl "MLH Multiplikation (0.8)" nicht zulässig ist, wenn für Quell- und Zielspeicherwort dasselbe Speicherwort genommen wird. Dieser Fehler wird vom Betriebsprogramm nicht erkannt.
2. Speicherwort C wird zur Rechnung benutzt und darf deshalb weder als Speicher noch als Quell- oder Zielspeicherwort verwendet werden. Findet kein Überlauf statt, ist der Inhalt von Speicherwort C nach der Operation gelöscht. In der Vorzeichenstelle steht jedoch das Vorzeichen des Ergebnisses.

MLH	Multiplikation (multiplication to high register)	0.8
-----	---	-----

3. Da die Multiplikation auf die Addition zurückgeführt wird, treten die beim Befehl "ADH Addition (0.4)" erwähnten Fehler auf, wenn der Inhalt der Kernspeicherstellen sedezimal ist. Es wird also kaum Sinn haben, die Multiplikation mit sedezimalen Speicherwortinhalten durchzuführen.

ML	Multiplikation (multiplication to low register)	0.9
----	--	-----



Zielspeicherwort 1 - 15
Quellspeicherwort {0.1 - 7.10}
 [0.1 - 7.11]

Wirkung:

1. Der Inhalt des Zielspeicherwortes wird mit dem Inhalt des Quellspeicherwortes multipliziert. Das Ergebnis steht anschließend kommagerecht und vorzeichenrichtig im Zielspeicherwort. Der Inhalt des Quellspeicherwortes bleibt unverändert.
2. Das Ergebnis einer Multiplikation hat doppelt soviel Stellen nach dem Komma wie die Faktoren. Im Zielspeicherwort werden aber nur so viele Stellen davon angegeben, wie in der KA festgelegt sind. Die überzähligen Nachkommastellen gehen verloren, wobei keine Rundung stattfindet.
3. Ist das Ergebnis der Multiplikation so groß, daß es im Zielspeicherwort allein nicht mehr dargestellt werden kann, so wird der Überlauf rechtsbündig mit gleichem Vorzeichen wie das Ergebnis in das Speicherwort C (= SW 4, Carry-Speicherwort) gestellt. Bei Überlauf wird MC = 1 andernfalls MC = 0 gesetzt.

Beispiel: (KA 6)

Multiplikation (SW 7) x (SW 20) → SW 7 (mit verschiedenen Vorzeichen)

vorher:	SW 7	+ 0 0 0 0 0 0 0 0 0 3 2 0 0 0 0 0 0
	SW 20	- 0 0 0 0 0 0 0 0 0 0 1 0 5 0 0 0 0
	SW 4	+ 0 0 0 0 0 0 0 1 2 3 4 0 0 0 0 7 8

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
	0 0 9	0	1 4 7		ML, 20.7
	1				
	2				

ML	Multiplikation (multiplication to low register)	0.9
----	--	-----

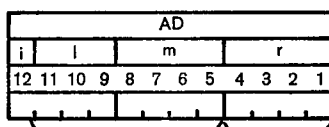
nachher:	SW 7	-	0	0	0	0	0	0	0	0	3	3	6	0	0	0	0	0
MC = 0																		
(kein Überlauf)	SW 20	-	0	0	0	0	0	0	0	0	0	1	0	5	0	0	0	0
	SW 4	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Im Speicherwort 4 steht also nach Ausführung des Befehls -0.

Bemerkungen:

1. Man beachte, daß der Befehl "ML Multiplikation (0.9)" nicht zulässig ist, wenn für Quell- und Zielspeicherwort dasselbe Speicherwort genommen wird. Dieser Fehler wird vom Betriebsprogramm nicht erkannt.
2. Speicherwort C wird zur Rechnung benutzt und darf deshalb weder als Speicher noch als Quell- oder Zielspeicherwort verwendet werden. Findet kein Überlauf statt, ist der Inhalt von Speicherwort C nach der Operation gelöscht. In der Vorzeichenstelle steht jedoch das Vorzeichen des Ergebnisses (vgl. Beispiel).
3. Da die Multiplikation auf die Addition zurückgeführt wird, treten die beim Befehl "ADH Addition (0.4)" erwähnten Fehler auf, wenn der Inhalt der Kernspeicherstellen sedezimal ist. Es wird also kaum Sinn haben, die Multiplikation mit sedezimalen Speicherwortinhalten durchzuführen.

DVH	Division (division to high register)	0.10
-----	---	------



Quellspeicherwort 1 - 15
Zielspeicherwort $\left[\begin{matrix} 0.1 - 7.10 \\ 0.1 - 7.11 \end{matrix} \right]$

Wirkung:

1. Der Inhalt des Zielspeicherwortes wird durch den Inhalt des Quellspeicherwortes dividiert. Das Ergebnis steht kommagerecht und vorzeichenrichtig im Zielspeicherwort. Der Inhalt des Quellspeicherwortes bleibt unverändert.
2. Ein Überlauf im Zielspeicherwort hat den Abbruch der Operation zur Folge. In diesem Fall wird der Merker MC = 1 andernfalls MC = 0 gesetzt.
3. Der Divisionsrest steht im Speicherwort C (SW 4, Carry-Speicherwort) mit doppelt soviel Nachkommastellen wie die KA festlegt. Das Vorzeichen vom Divisionsrest ist gleich dem Vorzeichen des Zielspeicherwortes vor der Operation.

Beispiel 1: (KA 4)

Division (SW 32) : (SW 5) → SW 32 ohne Rest.

vorher:

SW 32	+ 0 0 0 0 0 0 0 0 0 0 2 2 5 0 0 0 0
SW 5	+ 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
	0 0 10	0	2 0 5		DVH, 32.5
	1				

nachher:

SW 32	+ 0 0 0 0 0 0 0 0 0 0 4 5 0 0 0 0 0
MC = 0	
SW 5	+ 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0
SW 4	+ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

DVH	Division (division to high register)	0.10
-----	--------------------------------------	------

Beispiel 2: (KA 4)

Division (SW 20) : (SW 5) → SW 20 mit Rest.

vorher: SW 20

-	0	0	0	0	0	0	0	0	0	0	1	5	7	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

SW 5

-	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
	0 0 10	0	1 4 5		DVH, 20,5
	1				

nachher: SW 20

+	0	0	0	0	0	0	0	0	0	0	1	3	0	8	3	3
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

MC = 0 SW 5

-	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

SW 4

-	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Bemerkungen:

1. Man beachte, daß der Befehl "DVH Division (0.10)" nicht zulässig ist, wenn für Quell- und Zielspeicherwort dasselbe Speicherwort genommen wird. Dieser Fehler wird vom Betriebsprogramm nicht erkannt.
2. Speicherwort C wird zur Rechnung benutzt und darf deshalb weder als Speicher noch als Quell- oder Zielspeicherwort verwendet werden. Bleibt kein Rest, so ist der Inhalt von Speicherwort C nach der Ausführung des Befehls gelöscht. In der Vorzeichenstelle steht jedoch das Vorzeichen des Zielspeicherwortes vor der Operation.
3. Da die Division auf die Subtraktion zurückgeführt wird, treten die beim Befehl "SBH Subtraktion (0.6)" erwähnten Fehler auf, wenn der Inhalt der Kernspeicherstellen sedezimal ist. Es wird also kaum Sinn haben, die Division mit sedezimalen Speicherwortinhalten durchzuführen.
4. Man beachte die Vertauschung von Quell- und Zielspeicherwort im Befehl "DV Division (0.11)".

DV	Division (division to low register)	0.11
----	-------------------------------------	------

Beispiel 2: (KA 6)

Division (SW 9) : (SW 23) → SW 9 mit Überlauf im Zielspeicherwort und Abbruch der Operation.

vorher:

SW 9	+ 0 0 1 5 0 0 0 0 0 0 0 0 0 0 0 0
SW 23	+ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
0	0 11	0	1 7 9		DV, 23.9
1					

nachher:

SW 9	+ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 9
SW 23	+ 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
SW 4	+ 0 0 0 0 0 0 0 0 0 0 0 1 4 0 0 0

MC = 1

Im Zielspeicherwort und im Speicherwort 4 stehen keine verwertbaren Zahlen.

Bemerkung:

Für den Befehl "DV Division (0.11)" gelten die Bemerkungen 1 bis 3 des Befehls "DVH Division (0.10)". Man beachte bei diesem Befehl jedoch die Vertauschung von Quell- und Zielspeicherwort.

	Division	
--	----------	--

Nach einem einstelligen gekoppelten Linksshift des Carry- und des Dividenden-Speicherworts erfolgt eine wiederholte Subtraktion des Divisors vom Speicherwort, bis das Vorzeichen im Speicherwort wechselt. Die Anzahl der Subtraktionsschritte wird in der rechten Stelle des Dividenden-Speicherworts gezählt. Dieser Vorgang wird fünfzehnmal wiederholt. Das ergibt dann folgendes Bild nach Ausführung eines jeden der fünfzehn Schritte:

	Division	
--	----------	--

	<u>Dividenden-Speicherwort</u>	<u>Carry-Speicherwort</u>	<u>Divisoren-Speicherwort</u>
1. Schritt Anfang	+000,0000000000	+000,0378000000	+180,0000000000
1. Schritt Ende	+000,	+000,0378	+180,
2. Schritt Anfang	+000,	+000,378	+180,
2. Schritt Ende	+000,	+000,378	+180,
3. Schritt Anfang	+000,	+003,78	+180,
3. Schritt Ende	+000,	+003,78	+180,
4. Schritt Anfang	+000,	+037,8	+180,
4. Schritt Ende	+000,	+378,	+180,
5. Schritt Anfang	+000,	+018,	+180,
5. Schritt Ende	+000,	+180,	+180,
6. Schritt Anfang	+000,	+21	+180,
6. Schritt Ende	+000,	+210	+180,
7. Schritt Anfang	+000,	+210	+180,
7. Schritt Ende	+000,	+2100	+180,
8. Schritt Anfang	+000,	+2100	+180,
8. Schritt Ende	+000,	+21000	+180,
9. Schritt Anfang	+000,	+21000	+180,
9. Schritt Ende	+000,	+210000	+180,
10. Schritt Anfang	+000,	+210000	+180,
10. Schritt Ende	+000,	+2100000	+180,
11. Schritt Anfang	+000,	+2100000	+180,
11. Schritt Ende	+000,	+21000000	+180,
12. Schritt Anfang	+000,	+21000000	+180,
12. Schritt Ende	+000,	+210000000	+180,
13. Schritt Anfang	+000,	+210000000	+180,
13. Schritt Ende	+000,	+2100000000	+180,
14. Schritt Anfang	+000,	+2100000000	+180,
14. Schritt Ende	+000,	+21000000000	+180,
15. Schritt Anfang	+000,0210000000	+000,	+180,
15. Schritt Ende	+000,0210000000	+000,	+180,

Man beachte, daß die unterstrichene Stelle im C-Speicherwort nie ungleich Null wird.



2. Dividend:

+	0	0	3	1	4	1	6	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Divisor:

+	1	8	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Genau wie im ersten Beispiel erfolgt zuerst ein gekoppelter Linksshift um 12 Stellen. Das ergibt:

Carry-Speicher-
wort

+	0	0	0	0	0	3	1	4	1	6	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Dividenden-
Speicherwort

+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

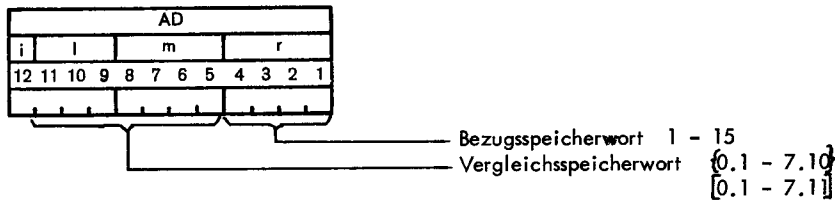
Die weiteren Schritte ergeben das Bild auf der folgenden Seite.

	Division	
--	----------	--

	<u>Dividenden-Speicherwort</u>	<u>Carry-Speicherwort</u>	<u>Divisoren-Speicherwort</u>
1. Schritt Anfang	+ 000, 000000000000	+ 000, 031416000000	+ 180, 000000000000
1. Schritt Ende	+ 000,	+ 000, 031416	+ 180,
2. Schritt Anfang	+ 000,	+ 000, 31416	+ 180,
2. Schritt Ende	+ 000,	+ 000, 31416	+ 180,
3. Schritt Anfang	+ 000,	+ 003, 1416	+ 180,
3. Schritt Ende	+ 000,	+ 003, 1416	+ 180,
4. Schritt Anfang	+ 000,	+ 031, 416	+ 180,
4. Schritt Ende	+ 000,	+ 031, 416	+ 180,
5. Schritt Anfang	+ 000,	+ 134, 16	+ 180,
5. Schritt Ende	+ 000,	+ 134, 16	+ 180,
6. Schritt Anfang	+ 000,	+ 341, 6	+ 180,

Hier wird der Divisionsvorgang abgebrochen, da eine Stelle ungleich Null vorn aus dem C-Speicherwort geschoben wird. Dem Anwenderprogramm wird das über den Merker MC, der auf 1 gesetzt wird, mitgeteilt.

CPH	Vergleich (compare high register)	0.12
-----	-----------------------------------	------



Wirkung:

Die Inhalte der durch AD_{lm} und AD_r benannten Speicherworte werden verglichen. Das Ergebnis des Vergleichs wird durch Setzen der Merker entsprechend folgender Tabelle zum Ausdruck gebracht:

Vergleichsspeicherwort	Bezugsspeicherwort	ML	MU
(SW AD_{lm}) < (SW AD_r)		1	1
(SW AD_{lm}) > (SW AD_r)		0	1
(SW AD_{lm}) = (SW AD_r)		0	0

Beispiel:

Vergleich von (SW 21) mit (SW 3).

vorher: SW 21 - 0 0 0 0 0 0 0 0 0 0 3 5 7 8 4 2 0

SW 3 + 0 0 0 0 0 0 0 0 0 0 1 3 9 7 0 6

BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
0	0 12	0	1 5 3		CPH, 21, 3
1					

nachher: SW 21 - 0 0 0 0 0 0 0 0 0 0 3 5 7 8 4 2 0

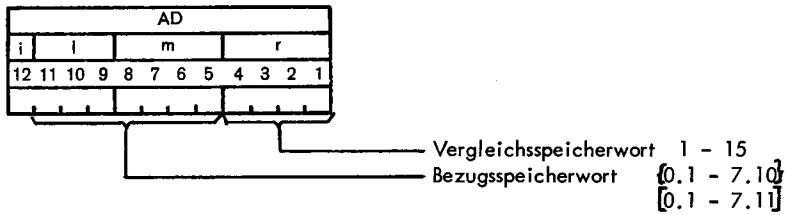
ML = 1 SW 3 + 0 0 0 0 0 0 0 0 0 0 1 3 9 7 0 6
 MU = 1

CPH	Vergleich (compare high register)	0.12
-----	-----------------------------------	------

Bemerkungen:

1. Die Inhalte der zu vergleichenden Speicherworte werden nicht verändert.
2. Beim Vergleich werden +0 und -0 nicht unterschieden.
3. Der Vergleich wird auch durchgeführt, wenn die Speicherworte sedezimale Inhalte haben. Die Abfrage von ML ist dann jedoch wenig sinnvoll.
4. Der Befehl ist auch zulässig, wenn für Bezugs- und Vergleichsspeicherworte dasselbe Speicherwort genommen wird. Es werden dann die Merker $ML = 0$ und $MU = 0$ gesetzt.
5. Man beachte die Vertauschung von Bezugs- und Vergleichsspeicherwort im Befehl "CP Vergleich (0.13)".

CP	Vergleich (compare low register)	0.13
----	----------------------------------	------



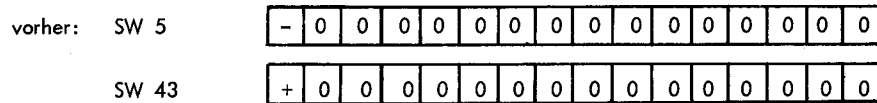
Wirkung:

Die Inhalte der durch AD_{l_m} und AD_r benannten Speicherworte werden verglichen. Das Ergebnis des Vergleichs wird durch Setzen der Merker entsprechend folgender Tabelle zum Ausdruck gebracht:

Vergleichsspeicherwort		Bezugsspeicherwort	ML	MU
(SW AD_r)	<	(SW AD_{l_m})	1	1
(SW AD_r)	>	(SW AD_{l_m})	0	1
(SW AD_r)	=	(SW AD_{l_m})	0	0

Beispiel:

Vergleich von (SW 5) mit (SW 43).



BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen			
0	0	13	0	2	11	5		CP, 43.5
1								

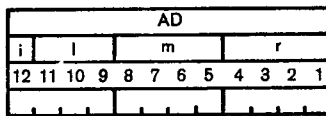
Nachher sind die Inhalte der Speicherworte 5 und 43 unverändert und die Merker $ML = 0$, $MU = 0$.

CP	Vergleich (compare low register)	0.13
----	----------------------------------	------

Bemerkung:

Für den Befehl "CP Vergleich (0.13)" gelten die Bemerkungen 1 bis 4 des Befehls "CPH Vergleich (0.12)". Man beachte bei diesem Befehl jedoch die Vertauschung von Bezugs- und Vergleichsspeicherwort.

CPZ	Vergleich mit Null (Compare to zero)	0.14
-----	--------------------------------------	------



Vergleichsspeicherwort $\{0.1 - 7.10\}$
 $[0.1 - 7.11]$

Wirkung:

Der Inhalt des durch AD_{Im} benannten Speicherwortes wird mit 0 verglichen. Das Ergebnis des Vergleichs wird durch Setzen der Merker entsprechend folgender Tabelle zum Ausdruck gebracht:

Vergleichsregister				ML	MU
$(SW AD_{Im}) <$	\wedge	0		1	1
$(SW AD_{Im}) >$	\vee	0		0	1
$(SW AD_{Im}) =$	\equiv	0		0	0

Beispiel:

Vergleich von (SW 22) mit Null.

vorher: SW 22

+	0	0	0	0	0	0	0	0	0	0	0	3	8	7	5	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

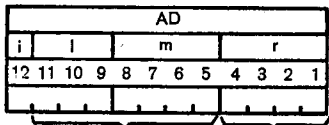
BW-Adresse	Op. Teil	i	Adreßteil			Symb. Adr.	Bemerkungen
	0 0 14	0	1	6	0		CPZ, 22.
	1						

nachher: Inhalt von Speicherwort 22 unverändert, ML = 0, MU = 1.

Bemerkungen:

1. Die Angaben in AD_r sind bedeutungslos.
2. Der Inhalt des Vergleichsspeicherwortes wird nicht verändert.
3. Eine -0 im Vergleichsspeicherwort wird wie +0 behandelt, d.h. ML = 0, MU = 0.
4. Der Vergleich wird auch durchgeführt, wenn die Speicherworte sedezimale Inhalte haben. Die Abfrage von ML ist dann jedoch wenig sinnvoll.

{CLRAL}	{Löschen ALC-Bereich (clear ALC-buffer)}	0.15
---------	---	------



$AD_r = 15$
Anzahl der zu löschenden Speicherworte
(0.0 - 7.10)

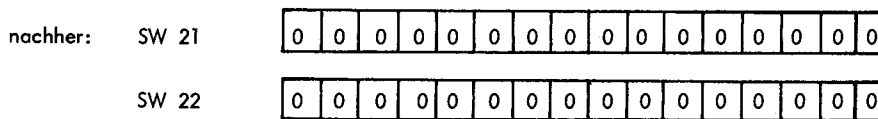
Wirkung:

1. Der Befehl "ALBG Anfang ALC-Bereich (3.4)" gibt das erste Speicherwort des ALC-Bereiches vor. Dieser Bereich dient zur Speicherung alphanumerischer Zeichen und als Datenbereich für Magnetkontokarten.
2. Im Befehl "CLRAL Löschen ALC-Bereich (0.15)" werden die ersten AD_l Speicherworte des ALC-Bereichs gelöscht, d.h. in alle Stellen der Speicherworte kommt die Ziffer 0.

Beispiel:

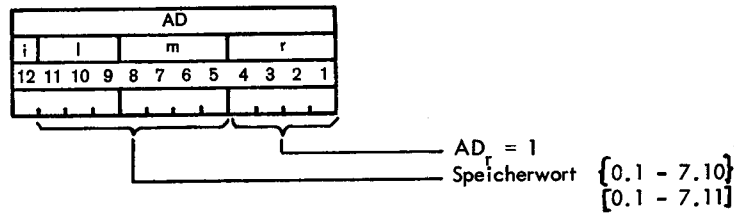
Der ALC-Bereich soll mit Speicherwort 21 beginnen. Die drei ersten Speicherworte des Bereiches sind zu löschen.

BW-Adresse	Op. Teil	Adreßteil	Symb. Adr.	Bemerkungen
0	3 4 0 0	9 5		ALBG, 21
1	0 15 0 0	3 15		CLRAL, 3.
2				



Der Befehl CLRAL darf bei dem Modell 820/15 auf keinen Fall angewendet werden!
Die Angaben $AD_r = 2$ bis 15 sind nicht abgesichert und führen zu Fehlern, die nicht erkannt werden können.

SGNIN	Vorzeichenwechsel (sign inversion)	0.15
-------	------------------------------------	------



Wirkung:

Das Vorzeichen vom Inhalt des durch AD_{Im} benannten Speicherwortes wechselt.

Beispiel:

Vorzeichenwechsel in SW 24:

vorher: SW 24 + 0 0 0 0 0 0 1 4 13 9 0 7 12 8 11 6

BW-Adresse	Op. Teil	i	Adreßteil	Symb. Adr.	Bemerkungen
	0 0 15	0	1 8 1		SGNIN, 24.
	1				

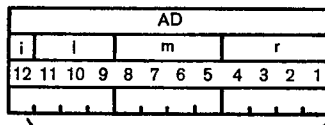
nachher: SW 24 - 0 0 0 0 0 0 1 4 13 9 0 7 12 8 11 6

Bemerkung:

Bei Vorzeichenabfragen entspricht eine gerade Zahl in Kernspeicherstelle 0 einem positiven, eine ungerade Zahl einem negativen Vorzeichen. Der Vorzeichenwechsel wird dadurch erreicht, daß nur Bit 1 vom Inhalt der Vorzeichenstelle geändert wird.

Beispiel: Vorher Stelle 0: OLLL mit der Bedeutung -,
 Nachher Stelle 0: OLLO mit der Bedeutung +.

BR	Unbedingter Sprung (branch)	1.0
----	-----------------------------	-----



} Zieladresse (0.0.0 - 7.15.15)

Wirkung:

Das Programm wird mit dem Befehl fortgesetzt, dessen Adresse unter AD_{lmr} angegeben ist.

Beispiel:

Das Programm soll mit dem Befehl unter Adresse 3.12.4 fortgesetzt werden.

BW-Adresse			Op. Teil		Adreßteil				Symb. Adr.	Bemerkungen
0	1	0	0	3	12	4		BR, ZWEIG		
1										

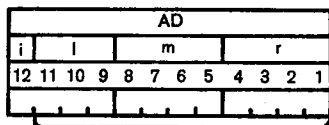
Nächster Befehl, der ausgeführt wird:

BW-Adresse			Op. Teil		Adreßteil				Symb. Adr.	Bemerkungen
3	12	4						ZWEIG		
		5								

Bemerkungen:

1. Der Befehl wird auch ausgeführt, wenn im Block 0 des Anwenderprogramms eine Zieladresse von 0.0.0 bis 0.0.4 angegeben wird.
2. Man beachte den Befehl "BL Blockumschaltung (2.11)".

	Sprung wenn Merker n auf 1 (branch if Mn = 1)	1.2 bis 1.8
--	--	-------------



Zieladresse (0.0.0 - 7.15.15)

Wirkung:

1. Der OP-Teil

BR2	1.2	bezieht sich auf Merker M2
BR3	1.3	bezieht sich auf Merker M3
BR4	1.4	bezieht sich auf Merker M4
BR5	1.5	bezieht sich auf Merker M5
BRL	1.6	bezieht sich auf Merker M6 = ML
BRU	1.7	bezieht sich auf Merker M7 = MU
BRC	1.8	bezieht sich auf Merker M8 = MC

2. Steht der entsprechende Merker auf 0, so wird das Programm mit dem nächsten Befehl fortgesetzt.

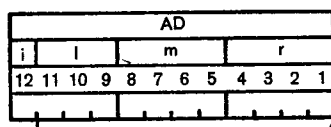
3. Steht der Merker auf 1, so wird das Programm mit dem Befehl fortgesetzt, dessen Adresse unter AD_{lmr} angegeben ist.

Bemerkungen:

1. Die Stellung des Merkers wird durch diesen Befehl nicht verändert.

2. Man beachte den Befehl " BL Blockumschaltung (2.11) ".

BXG	Sprung wenn $(I) > 1023$ (branch if (I) greater 1023)	1.9
-----	--	-----



Zieladresse (0.0.0 - 7.15.15)

Wirkung:

1. Ist der Inhalt des Indexregisters kleiner oder gleich 1023, so wird das Programm mit dem nächsten Befehl fortgesetzt.
2. Ist der Inhalt des Indexregisters größer als 1023, so wird das Programm mit dem Befehl fortgesetzt, dessen Adresse unter AD_{mr} angegeben ist.

Beispiel:

Das Programm soll mit dem Befehl unter Adresse 1.0.5 fortgesetzt werden, wenn der Inhalt des Indexregisters I_1 größer als 1023 ist.

vorher: I_1

4	3	5
---	---	---

 entspricht $(I_1) = 1077$

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
	0	2	2	0 0 0 1	XF, 1
	1	1	9	0 1 0 5	BXG, WEIT

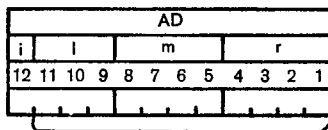
Nächster Befehl, der ausgeführt wird:

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
1	0	5		WEIT	

Bemerkungen:

1. Die Bedingung $(I) > 1023$ ist gleichbedeutend mit der Abfrage, ob im Indexregister Bit 11 = 1 gesetzt ist. Der Inhalt des Indexregisters bleibt dabei unverändert.
2. Man beachte die Befehle " XF und XFR Indexvorbefehle (2.2) " (vgl. Beispiel) und "BL Blockumschaltung (2.11) ".

BXU	Sprung wenn (I) \neq 0 (branch if (I) unequal 0)	1.10
-----	---	------



Zieladresse (0.0.0 - 7.15.15)

Wirkung:

1. Ist der Inhalt des Indexregisters gleich 0, so wird das Programm mit dem nächsten Befehl fortgesetzt.
2. Ist der Inhalt des Indexregisters von 0 verschieden, so wird das Programm mit dem Befehl fortgesetzt, dessen Adresse unter AD_{lmr} angegeben ist.

Beispiel:

Das Programm soll mit dem Befehl unter Adresse 3.15.4 fortgesetzt werden, wenn der Inhalt des Indexregisters I₂ von 0 verschieden ist.

vorher: I₂

0	0	0
---	---	---

 entspricht (I₂) = 0

BW-Adresse		Op. Teil		I	Adreßteil			Symb. Adr.	Bemerkungen
1	0	12	2	2	0	0	0	2	XF, 2
		13	1	10	0	3	15	4	BXU, RECH
1	0	14							
		15							

nachher: I₂

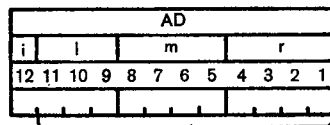
0	0	0
---	---	---

Das Programm wird mit dem Befehl unter Adresse 1.0.14 fortgesetzt.

Bemerkungen:

1. Der Inhalt des Indexregisters wird durch diesen Befehl nicht verändert.
2. Man beachte die Befehle "XF und XFR Indexvorbefehle (2.2)" (vgl. Beispiel) und "BL Blockumschaltung (2.11)".

BRS	Unterprogramm sprung (branch to subroutine)	1.11
-----	--	------



Anfangsadresse des Unterprogramms
(0.0.0 - 7.15.15)

Wirkung:

1. Das Programm fährt mit dem Befehl fort, dessen Adresse unter AD_{lmr} angegeben ist. Die Absprungadresse, d.h. die Adresse des Befehls "BRS Unterprogramm sprung (1.11)" wird bis zum Befehl "BRR Rücksprung (1.12)" gespeichert.

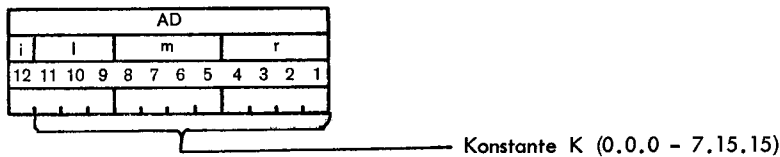
Im einzelnen geschieht folgendes:

2. Der laufende Befehlszähler enthält die Adresse des Befehls "BRS Unterprogramm sprung (1.11)", die dort gespeichert bleibt. Der Befehlszähler für die darunterliegende Programmstufe wird mit der in AD_{lmr} angegebenen Anfangsadresse des Unterprogramms geladen. Das Programm arbeitet mit dem zuletzt genannten Befehlszähler weiter.
3. Der Unterprogrammstufenzähler (Indexregister I₆) enthält die um 4 erhöhte Stufe des gerade laufenden Unterprogramms (4 bis 9), wobei das Unterprogramm 0. Stufe dem Hauptprogramm entspricht. Bei einem Unterprogramm sprung wird der Inhalt des Unterprogrammstufenzählers um 1 erhöht.

Bemerkungen:

1. Es sind maximal 5 echte Unterprogrammstufen zulässig. Wird eine 6. Stufe befohlen, so läuft das Programm auf Internfehlerstopp (vgl. Beschreibung "Programm, Aufbau", Kapitel 2.3).
2. Man beachte den Befehl "BL Blockumschaltung (2.11)".
3. Weitere Ausführungen und Beispiele zu diesem Befehl findet man im Anschluß an den Befehl "BRR Rücksprung (1.12)".

BRR	Rücksprung (branch return)	1.12
-----	----------------------------	------



Wirkung:

1. Das Programm fährt mit dem Befehl fort, der unter der zuletzt gespeicherten Absprungadresse + K + 1 steht.

Im einzelnen geschieht folgendes:

2. Der laufende Befehlszähler enthält die Adresse des Befehls "BRR Rücksprung (1.12)" und bleibt unverändert. Der Befehlszähler für die darüberliegende Programmstufe, der die Adresse des letzten Befehls "BRS Unterprogramm sprung (1.11)" enthält, wird um die in $AD_{l_{mr}}$ stehende Konstante K+1 erhöht. Der Befehl, dessen Adresse in dem zuletzt genannten Befehlszähler steht, wird als nächster ausgeführt; das Programm arbeitet mit diesem Befehlszähler weiter.
3. Der Inhalt des Unterprogrammstufenzählers (Indexregister I_6) wird um 1 vermindert.

Bemerkungen:

1. Die Konstante K in $AD_{l_{mr}}$ bewirkt, daß die Rücksprungadresse um K erhöht wird. Soll sie um K erniedrigt werden (das entspricht einer negativen Konstanten K), so muß in $AD_{l_{mr}}$ das Komplement von K zu 2^{11} eingesetzt werden.
2. Wird im Hauptprogramm (Unterprogramm 0. Stufe) der Befehl "BRR Rücksprung (1.12)" gegeben, so läuft das Programm auf Internfehlerstopp (vgl. Beschreibung "Programm, Aufbau", Kapitel 2.3).
3. Der Rücksprung erfolgt immer in den Block, in dem der Befehl "BRS Unterprogramm sprung (1.11)", gegeben wurde; ein Vorbefehl "BL Blockumschaltung (2.11)" hat also auf den Befehl "BRR Rücksprung (1.12)" keinen Einfluß und wird durch diesen nicht gelöscht.

	Unterprogramme	
--	----------------	--

Beispiel:

Die Befehle "BRS Unterprogramm sprung (1.11)" und BRR Rucksprung (1.12)" bei 2 Unterprogrammstufen im Festspeicherblock 0.

Im Hauptprogramm stehe auf Adresse 2.3.0 der Sprung in das 1. Unterprogramm, das bei Adresse 4.0.8 beginnen soll:

BW-Adresse			Op. Teil		I	Adreßteil			Symb. Adr.	Bemerkungen
2	3	0	1	11	0	4	0	8		BRS, ROUT1
		1								

Nach Ausführung des Befehls haben die Befehlszähler folgende Inhalte:

- Befehlszähler 0

2	3	0
---	---	---

 enthält die Absprungadresse in das 1. Unterprogramm
- Befehlszähler 1

4	0	8
---	---	---

 Befehlszähler für 1. Unterprogramm
- Unterprogrammstufenzähler

0	0	5
---	---	---

 enthält Unterprogrammstufe +4

Im 1. Unterprogramm stehe auf Adresse 4.1.9 der Sprung in das 2. Unterprogramm, das bei Adresse 5.3.2 beginnen soll:

BW-Adresse			Op. Teil		I	Adreßteil			Symb. Adr.	Bemerkungen
4	0	8							ROUT1	Unterprogramm 1
4	1	9	1	11	0	5	3	2		BRS, ROUT2

Nach Ausführung des Befehls haben die Befehlszähler folgende Inhalte:

- Befehlszähler 0

2	3	0
---	---	---

 enthält die Absprungadresse in das 1. Unterprogramm
- Befehlszähler 1

4	1	9
---	---	---

 enthält die Absprungadresse in das 2. Unterprogramm

	Unterprogramme	
--	----------------	--

Befehlszähler 2

5	3	2
---	---	---

 Befehlszähler für 2. Unterprogramm

Unterprogramm-
stufenzähler

0	0	6
---	---	---

 enthält Unterprogrammstufe +4

Da sich beide Unterprogramme im Festspeicherblock 0 befinden sollen, sind die zu den Adressen gehörenden Blocknummern für jeden Befehlszähler 0.

Das 2. Unterprogramm laufe bis zur Adresse 6.1.12. Dann ist

Befehlszähler 0

2	3	0
---	---	---

 enthält die Absprungadresse in das 1. Unterprogramm

Befehlszähler 1

4	1	9
---	---	---

 enthält die Absprungadresse in das 2. Unterprogramm

Befehlszähler 2

6	1	12
---	---	----

 Befehlszähler für 2. Unterprogramm

Unterprogramm-
stufenzähler

0	0	6
---	---	---

 enthält Unterprogrammstufe +4

Unter der Adresse 6.1.12 stehe ein Befehl "BRR Rücksprung (1.12)":

BW-Adresse			Op. Teil			Adreßteil			Symb. Adr.	Bemerkungen
5	3	2							ROUT2	Unterprogramm 2
6	1	12	1	12	0	0	0	0		BRR
		13								

Nach Ausführung des Befehls haben die Befehlszähler folgende Inhalte:

Befehlszähler 0

2	3	0
---	---	---

 enthält die Absprungadresse in das 1. Unterprogramm

Befehlszähler 1

4	1	10
---	---	----

 Befehlszähler für 1. Unterprogramm

Befehlszähler 2

6	1	12
---	---	----

 bleibt unverändert

Unterprogramm-
stufenzähler

0	0	5
---	---	---

 enthält Unterprogrammstufe +4

	Unterprogramme	
--	----------------	--

Das 1. Unterprogramm läuft weiter bis zur Adresse 4.2.14 und findet dort einen Befehl "BRR Rücksprung (1.12)":

BW-Adresse			Op. Teil			I			Adreßteil			Symb. Adr.	Bemerkungen
4	1	9	1	11	0	5	3	2					BRS, ROUT2
													Unterprogramm 1
4	2	14	1	12	0	0	0	3					BRR, 3
		15											

Nach Ausführung des Befehls haben die Befehlszähler folgende Inhalte:

Befehlszähler 0	<table border="1"><tr><td>2</td><td>3</td><td>4</td></tr></table>	2	3	4	Befehlszähler für Hauptprogramm
2	3	4			
Befehlszähler 1	<table border="1"><tr><td>4</td><td>2</td><td>14</td></tr></table>	4	2	14	bleibt unverändert
4	2	14			
Befehlszähler 2	<table border="1"><tr><td>6</td><td>1</td><td>12</td></tr></table>	6	1	12	bleibt unverändert
6	1	12			
Unterprogramm- stufenzähler	<table border="1"><tr><td>0</td><td>0</td><td>4</td></tr></table>	0	0	4	enthält Unterprogrammstufe +4
0	0	4			

Das Hauptprogramm fährt mit dem Befehl unter Adresse 2.3.4 fort:

BW-Adresse			Op. Teil			I			Adreßteil			Symb. Adr.	Bemerkungen
2	3	0	1	11	0	4	0	8					BRS, ROUT1
		1											
		2											
		3											
2	3	4											
		5											

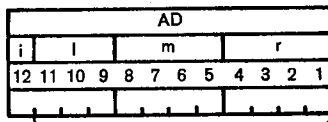
Bemerkung:

Die Inhalte der Befehlszähler und die dazugehörigen Blocknummern können über die Befehle "INCX und XINC Transport Befehlszähler (2.2)" verarbeitet werden.

Der Unterprogrammstufenzähler ist als Indexregister I_6 nur über die Befehle "MVX und MVXH Indexregistertransport (2.2)" zugänglich.

Dieses trifft allerdings nur bei den Modellen 820/25 und 820/35 zu.

SST	Substitution	1.13
-----	--------------	------



Befehlsadresse (0.0.0 - 7.15.15)

Wirkung:

Es wird der Befehl ausgeführt, dessen Adresse in AD_{l_mr} steht.

Beispiel:

Ausdrucken verschiedener Tabellen:

vorher: l₁

0	0	3
---	---	---

BW-Adresse			Op. Teil		l	Adreßteil			Symb. Adr.	Bemerkungen
1	3	0	2	2	0	0	1	0		XF, 1.
		1	1	13	1	2	8	10		SST, XI, DRUCK
		2								
		3								

Ab Adresse 2.8.10 im gleichen Block stehen die Befehle

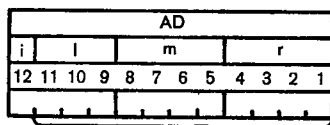
BW-Adresse			Op. Teil		l	Adreßteil			Symb. Adr.	Bemerkungen
2	8	10							DRUCK	
		11								
		12								
		13	3	0	0	6	0	11		TT, 6.0.11
		14								

Es wird der Befehl unter Adresse 2.8.13 ausgeführt, d.h. die ab 6.0.11 gespeicherte Tabelle ausgedruckt. Das Programm fährt mit dem Befehl unter Adresse 1.3.2 fort.

Bemerkungen:

1. Es kann jeder Befehl substituiert werden.
2. Man beachte den Befehl "BL Blockumschaltung (2.11)".

ICA	Indirekte Konstante nach SW A (indirect constant to A)	1.14
-----	---	------



Befehlsadresse (0.0.0 - 7.15.15)

Wirkung:

1. Es wird der Befehl aufgesucht, dessen Adresse in AD_{l_{mr}} steht. Die im AD_{l_{mr}}-Teil dieses Befehls sedezimal dargestellte Zahl wird als ganze Zahl mit positivem Vorzeichen nach Speicherwort 3 gebracht und dort dezimal dargestellt.
2. Entsteht bei großer KA ein Überlauf über Stelle 1 von Speicherwort 3 hinaus, so wird der Merker MC = 1 gesetzt; der Überlauf geht verloren. Gibt es keinen Überlauf ist MC = 0.

Beispiel: (KA 5)

vorher: I₁

3	12	3
---	----	---

SW 3

-	0	0	0	0	0	0	0	0	0	1	0	2	3	4	0	5
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

BW-Adresse			Op. Teil		l	Adreßteil			Symb. Adr.	Bemerkungen
1	15	15	2	2	0	0	1	0		XF, 1.
2	0	0	1	14	1	0	2	0		ICA, XI, 2.
		1								

Es kommt der AD_{l_{mr}}-Teil des folgenden Befehls nach Speicherwort 3:

BW-Adresse			Op. Teil		l	Adreßteil			Symb. Adr.	Bemerkungen
3	14	3	0	8	0	0	7	15		

nachher: SW 3

+	0	0	0	0	0	0	0	0	1	2	7	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

MC = 0

Das Programm fährt mit dem Befehl unter Adresse 2.0.1 fort.

ICA	Indirekte Konstante nach SW A (indirect constant to A)	1.14
-----	---	------

Bemerkungen:

1. Der Befehl "ICA Indirekte Konstante nach A (1.14) ist nur indiziert von Bedeutung (vgl. Beispiel).
2. Man beachte den Befehl "BL Blockumschaltung (2.11)".

CA	Konstante nach A (constant to A)	1.15
----	----------------------------------	------

AD											
i	l			m			r				
12	11	10	9	8	7	6	5	4	3	2	1

Konstante (0.0.0 - 7.15.15)

Wirkung:

1. Die im AD_{l,m,r}-Teil sedezimal dargestellte Zahl wird als ganze Zahl mit positivem Vorzeichen nach Speicherwort 3 gebracht und dort dezimal dargestellt.
2. Entsteht bei großer KA ein Überlauf über Stelle 1 von Speicherwort 3 hinaus, so wird der Merker MC = 1 gesetzt; der Überlauf geht verloren. Gibt es keinen Überlauf ist MC = 0.

Beispiel: (KA 6)

Die Zahl 100 soll nach Speicherwort A gebracht werden.

vorher: SW 3

-	0	0	0	0	0	0	0	0	0	0	3	4	5	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

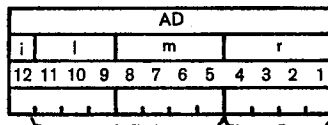
BW-Adresse	Op. Teil			l	Adreßteil			Symb. Adr.	Bemerkungen
	0	1	15	0	0	6	4		CA, 100
	1								

nachher: SW 3

+	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

MC = 0

SR	Rechtsshift (shift right)	2.0
----	---------------------------	-----



Anzahl der Kernspeicherstellen, um die geschoben werden soll (0 - 15)

Speicherwort $\{0.1 - 7.10\}$
 $[0.1 - 7.11]$

Wirkung:

1. Der Inhalt des durch AD_{im} benannten Speicherwortes wird um die in AD_r angegebene Stellenzahl nach rechts verschoben. Die nach rechts über Stelle 15 hinausgeschobenen Stelleninhalte gehen verloren, von links werden ab Stelle 1 Nullen nachgezogen. Die Vorzeichenstelle 0 des Speicherwortes wird nicht verändert.
2. Der Merker MC wird auf 1 gesetzt, wenn während des Shifts ein von 0 verschiedener Stelleninhalt nach rechts über Stelle 15 hinausgeschoben wurde. Andernfalls wird $MC = 0$ gesetzt.

Beispiel:

Der Inhalt von Speicherwort 20 soll um 4 Stellen nach rechts geschoben werden.

vorher: SW 20

-	0	0	0	0	0	0	0	0	0	0	0	14	7	5	13	0	0
---	---	---	---	---	---	---	---	---	---	---	---	----	---	---	----	---	---

BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
	0 2 0 0	0	1 4 4		SR, 20.4
	1				

nachher: SW 20

-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	7
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----	---

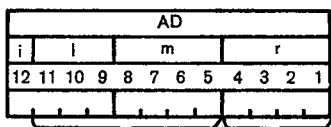
MC = 1

SR	Rechtsshift (shift right)	2.0
----	---------------------------	-----

Bemerkungen:

1. Das Verschieben eines dezimalen Speicherwortinhaltes um n Stellen nach rechts ist gleichbedeutend mit einer Division durch 10^n .
2. Es kann sowohl ein dezimaler als auch sedezimaler Speicherwortinhalt verschoben werden (vgl. Beispiel). Beim Verschieben eines sedezimalen Speicherwortinhaltes beachte man jedoch, daß der Inhalt der Stelle 0 unverändert bleibt.

SL	Linksshift (shift left)	2.1
----	-------------------------	-----



Anzahl der Kernspeicherstellen, um die geschoben werden soll (0 - 15)

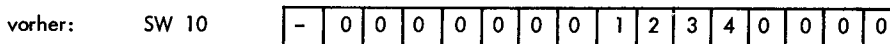
Speicherwort $\{0.1 - 7.10\}$
 $[0.1 - 7.1]$

Wirkung:

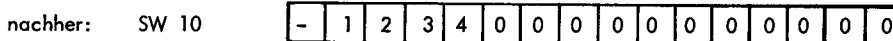
1. Der Inhalt des durch AD_{im} benannten Speicherwortes wird um die in AD_r angegebene Stellenzahl nach links verschoben. Die nach links über Stelle 1 hinausgeschobenen Stelleninhalte gehen verloren, von rechts werden ab Stelle 15 Nullen nachgezogen. Die Vorzeichenstelle 0 des Speicherwortes wird nicht verändert.
2. Der Merker MC wird auf 1 gesetzt, wenn während des Shifts ein von 0 verschiedener Stelleninhalt nach links über Stelle 1 hinausgeschoben wurde. Andernfalls wird $MC = 0$ gesetzt.

Beispiel:

Der Inhalt von Speicherwort 10 soll um 7 Stellen nach links geschoben werden.



BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
	0	2 1	0 0 10 7		SL, 10.7
	1				

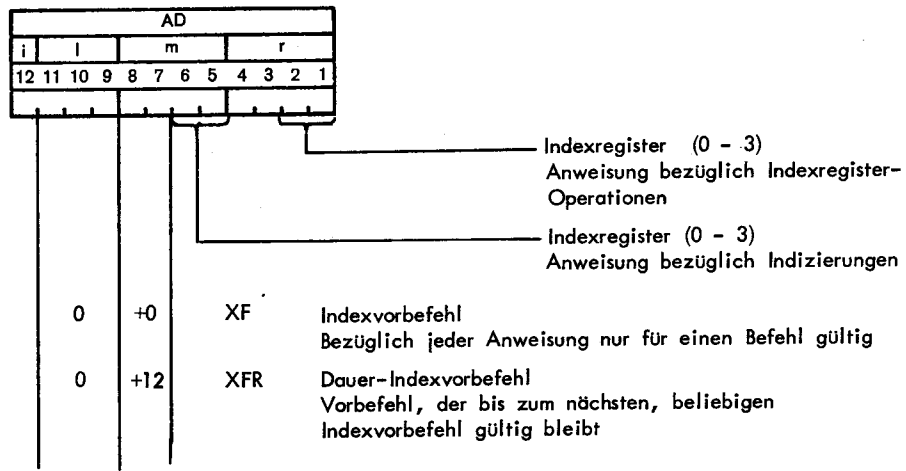


MC = 0

Bemerkungen:

1. Das Verschieben eines dezimalen Speicherwortinhalts um n Stellen nach links ist gleichbedeutend mit einer Multiplikation mit 10^n .
2. Es kann sowohl ein dezimaler als auch sedezimaler Speicherwortinhalt verschoben werden. Beim Verschieben eines sedezimalen Speicherwortinhaltes beachte man jedoch, daß der Inhalt der Stelle 0 unverändert bleibt.

	Indexvorbefehl (indexregister following)	2.2
--	--	-----



Wirkung:

1. Ein Indexregistervorbefehl, im folgenden kurz Indexvorbefehl genannt, enthält zwei Anweisungen. Er legt sowohl fest, mit welchem Indexregister (AD_m Bit 5 und Bit 6) eine Indizierung durchzuführen ist, als auch auf welches Indexregister (AD_r Bit 1 und Bit 2) eine Indexregister-Operation sich beziehen soll. Unter Indexregister-Operation ist dabei jede Benutzung eines Indexregisters zu verstehen, die keine Indizierung ist.
2. Der einfache Indexvorbefehl gilt bezüglich jeder der beiden Anweisungen für genau einen Befehl, der jedoch nicht unmittelbar auf den Vorbefehl folgen muß. Sobald eine Anweisung des Vorbefehls angewandt wurde, ist sie anschließend aufgehoben, doch bleibt die Bedeutung des Vorbefehls für die andere Anweisung weiter gültig. Liegt keine Anweisung vor, so wird I₀ benutzt.
3. Der Dauer-Indexvorbefehl gilt solange, bis ein neuer Indexvorbefehl gegeben wird. Das kann auch ein einfacher Indexvorbefehl sein.

	Indexvorbefehl (indexregister following)	2.2
--	--	-----

Beispiel 1:

Indizierte Addition zweier Speicherwortinhalte unter Verwendung von I₁.

vorher: I₁

1	2	0
---	---	---

 entspricht (I₁) = 18 · 2⁴

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
	0 2 2	0	0 1 0		XF, 1.
	1 0 4	1	0 10 5		ADH, XI, 10.5
	2				
	3				

nachher: I₁

1	2	0
---	---	---

Es wird die Addition (SW 28) + (SW 5) → SW 28 durchgeführt. Anschließend ist die Bedeutung des Indexvorbefehls aufgehoben, d.h. bei einem weiteren indizierten Befehl würde die Indizierung wieder mit Indexregister I₀ vorgenommen werden.

Beispiel 2:

Konstante 17 nach Indexregister I₂ links, mitte bringen.

vorher: I₂

3	5	3
---	---	---

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
	0 2 2	0	0 0 2		XF, 2
	1 2 10	0	1 1 0		CX, 17.
	2				

nachher: I₂

1	1	0
---	---	---

 entspricht (I₂) = 17 · 2⁴

Anschließend ist die Bedeutung des Indexvorbefehls aufgehoben, d.h. eine weitere Indexregisteroperation würde wieder mit Indexregister I₀ durchgeführt werden.

	Indexvorbefehl (indexregister following)	2.2
--	--	-----

Beispiel 3:

Zusammenfassung der Beispiele 1 und 2:

Zuerst indizierte Addition zweier Speicherwortinhalte unter Verwendung von I_1 , dann 17 nach I_2 links, mitte bringen.

vorher: I_1

1	2	0
---	---	---

 entspricht (I_1) = $18 \cdot 2^4$

BW-Adresse	Op. Teil			I	Adreßteil			Symb. Adr.	Bemerkungen
	0	2	2	0	0	1	2		XF, 1,2
	1	0	4	1	0	10	5		ADH, XI, 10,5
	2	2	10	0	1	1	0		CX, 17.
	3								

nachher: I_1

1	2	0
---	---	---

 entspricht (I_1) = $18 \cdot 2^4$

I_2

1	1	0
---	---	---

 entspricht (I_2) = $17 \cdot 2^4$

Da nur die ausgenutzte Anweisung des Vorbefehls gelöscht wird, braucht nur ein Indexvorbefehl mit beiden Anweisungen gegeben zu werden.

	Transport Indexregister (move indexregister)	2.2
--	---	-----

AD											
i	l						m			r	
12	11	10	9	8	7	6	5	4	3	2	1
		0								0	
									y		
											z
										+8	
									y		
											z

MVX

Transport Indexregister

Zielindexregister I_y (10 - 13)Quellindexregister I_z $\{10-17\}$ $[10-13, 15]$

MVXH

Transport Indexregister

Quellindexregister I_y (10 - 13)Zielindexregister I_z $\{10-17\}$ $[10-13, 15]$ Wirkung:

1. Der Inhalt des Quellindexregisters wird in das Zielindexregister übertragen. Der Inhalt des Quellindexregisters bleibt unverändert.

2. Bei den Modellen 820/25 und 820/35 gibt es 8 Indexregister ($I_0 - I_7$), von denen I_0 bis I_3 zur freien Verfügung stehen und I_4 bis I_7 nur über den Befehl "MVX oder MVXH Transport Indexregister (2.2)" angesprochen werden können.

3. Bei dem Modell 820/15 gibt es 5 Indexregister ($I_0 - I_3, I_5$), von denen I_0 bis I_3 zur freien Verfügung stehen und I_5 nur über den Befehl "MVX oder MVXH Transport Indexregister (2.2)" angesprochen werden können.

4. Die Funktion der Indexregister ist auf Seite P 7 (Programm, Aufbau) aufgeführt.

	Transport Indexregister (move indexregister)	2.2
--	---	-----

Beispiel für Modell 820/25 und 820/35:

Der Inhalt des Unterprogrammstufenzählers soll nach Indexregister I₁ gebracht werden.

BW-Adresse		Op. Teil		I	Adreßteil			Symb. Adr.	Bemerkungen
	0	2	2	0	0	5	6		MVX, 1.6
	1								

nachher: I₁

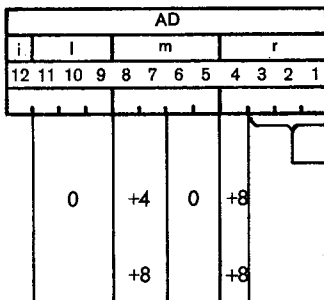
0	0	7
---	---	---

Das entspricht der Unterprogrammstufe 3, d.h. der Befehl wurde im 3. Unterprogramm gegeben.

Bemerkung:

Man beachte, daß der Unterprogrammstufenzähler immer die um 4 erhöhte Stufe des gerade laufenden Unterprogramms enthält (vgl. Beispiel), wobei das Unterprogramm 0. Stufe dem Hauptprogramm entspricht.

	Transport Befehlszähler (instruction counter, indexregister)	2.2
--	---	-----



Numer der Befehlszähler (0 - 5)

INCX Inhalt Befehlszähler nach I₀
 Blocknummer nach I₁

XINC Inhalt I₀ nach Befehlszähler
 Inhalt I₁ ergibt Blocknummer

Wirkung:

1. Befehl INCX: Der Inhalt des angesprochenen Befehlszählers wird in das Indexregister I₀ übertragen, die dazugehörige Blocknummer kommt nach Indexregister I₁.
2. Befehl XINC: Der Inhalt des Indexregisters I₀ wird in den in Bit 1 bis Bit 3 angegebenen Befehlszähler übertragen, der Inhalt des Indexregisters I₁ ergibt die dazugehörige Blocknummer.
3. Die Befehlszähler 0 bis 5 gelten für die 0. bis 5. Unterprogrammstufe, wobei die 0. Unterprogrammstufe dem Hauptprogramm entspricht.

Beispiel:

Der Inhalt des Befehlszählers 3 für die 3. Unterprogrammstufe ist gesucht.

BW-Adresse	Op. Teil	Adreßteil	Symb. Adr.	Bemerkungen
0	2 2	0 0 4 11		INCX, 3
1				

nachher: I₀

5	13	10
---	----	----

 Befehlsadresse

 I₁

0	0	2
---	---	---

 Blocknummer

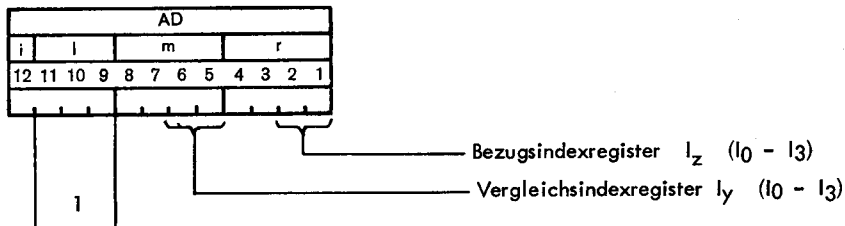
Im Befehlszähler 3 steht also die Adresse 5.13.10 vom Festspeicherblock 2.

	Transport Befehlszähler (instruction counter, indexregister)	2.2
--	---	-----

Bemerkungen:

1. Bei $AD_r = 14$ und $AD_i = 15$ läuft das Programm auf Internfehlerstopp (vgl. Beschreibung "Programm, Aufbau", Kapitel 2.3).
2. Bit 5 und Bit 6 des AD-Teiles sind für die Ausführung des Befehls bedeutungslos.
3. Die Befehle INCX und XINC sind bei dem Modell 820/15 nicht möglich. Falls sie angewendet werden, besteht die Gefahr, andere interne Zähler zu verändern, ohne daß das Programm auf interne Fehlerstopp läuft.

CPX	Indexregistervergleich (compare indexregister)	2.2
-----	---	-----



Wirkung:

Der Inhalt des durch AD_m Bit 5 und 6 benannten Indexregisters I_y wird mit dem Inhalt des durch AD_i Bit 1 und 2 benannten Indexregisters I_z verglichen. Das Ergebnis des Vergleichs wird durch Setzen der Merker entsprechend folgender Tabelle zum Ausdruck gebracht.

Vergleichsindexregister		Bezugsindexregister	ML	MU
(I_y)	\equiv	(I_z)	0	0
(I_y)	$>$	(I_z)	0	1
(I_y)	$<$	(I_z)	1	1

Beispiel:

Vergleich (I_1) mit (I_2) .

vorher: I_1

0	1	2
---	---	---

 entspricht $(I_1) = 18$
 I_2

0	0	15
---	---	----

 entspricht $(I_2) = 15$

BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen				
	0	2	2	0	1	1	2		
	1								CPX, 1.2

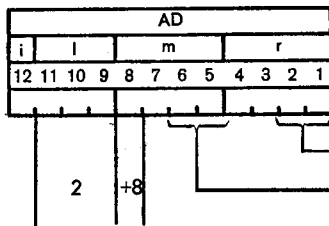
nachher: Inhalte der Indexregister I_1 und I_2 unverändert, $ML = 0$, $MU = 1$.

CPX	Indexregistervergleich (compare indexregister)	2.2
-----	---	-----

Bemerkungen:

1. Die Inhalte der zu vergleichenden Indexregister werden nicht verändert.
2. Der Befehl ist auch zulässig, wenn für Bezugs- und Vergleichsindexregister dasselbe Indexregister genommen wird. Es werden dann die Merker $ML = 0$, $MU = 0$ gesetzt.
3. Bit 3 und 4 sowie Bit 7 und 8 des AD-Teils sind für die Ausführung des Befehls bedeutungslos.

MVOX	Befehlsspeicherung (move order)	2.2
------	---------------------------------	-----



$I_3 \hat{=} \text{Adresse}$; aber nicht erhöht!!
um $\{0,5,0\}$ [0,4,0]

Wirkung:

1. Es wird ein Befehl zusammengestellt und im Programm-Kernspeicher gespeichert.
2. Der Inhalt des durch AD_m Bit 5 und 6 benannten Indexregisters ergibt den OP-Teil des Befehles.
Der Inhalt des durch AD_r Bit 1 und 2 benannten Indexregisters ergibt den AD_{lmr} -Teil des Befehls.
Der AD_i -Teil des Befehls wird gleich MC gesetzt.
3. Ein vorangegangener Blockumschaltbefehl legt den Kernspeicherblock fest, in dem der Befehl gespeichert werden soll. Der Befehl wird dort unter der Adresse gespeichert, die im Indexregister I_3 steht.
4. Die Wirkung des Blockumschaltbefehls ist anschließend aufgehoben, der Inhalt des Indexregisters I_3 ist um 1 erhöht.

Beispiel:

Der Befehl (SW 6) : (SW 17) \rightarrow SW 6 soll im Kernspeicherblock 1 unter der Adresse 3.7.4 gespeichert werden.

vorher: I_1

0	0	11
---	---	----

MC = 0 I_0

1	1	6
---	---	---

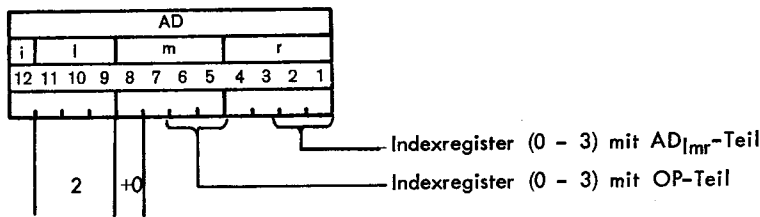
BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
0	3 11	0 3 2 4			BPX, 3.2.4
1	2 11	0 4 1 1			BL, LIV, 1
2	2 2 0	2 9 0			MVOX, 1.0
3					

MVOX	Befehlsspeicherung (move order)	2.2
------	---------------------------------	-----

Bemerkungen:

1. Ist zuvor kein Blockumschaltbefehl gegeben worden, so erfolgt die Abspeicherung im Kernspeicherblock 0. Bei einem Blockumschaltbefehl nach einem Festspeicherblock erfolgt die Abspeicherung im Kernspeicherblock mit der gleichen Nummer. Wird ein nicht vorhandener Kernspeicher angesprochen, so bleibt das Programm stehen. Ein zu kleiner Kernspeicher wird jedoch vom Betriebsprogramm nicht erkannt.
2. Für den OP-Teil sind nur die letzten 6 Bit des betreffenden Indexregisters maßgebend. Der Rest des Indexregisters bleibt unberücksichtigt. Der Merker MC und die angesprochenen Indexregister werden nicht verändert.
3. Bit 3 und 4 sowie Bit 7 des AD-Teils sind für die Ausführung des Befehls bedeutungslos.
4. Dieser Befehl darf nur bei dem vorhandenen "Modul GP" angewendet werden, da sonst der Ablauf des Programmes gestört wird. Es erfolgt kein interner Fehlerstop.

SSTX	Substitution mit Indexregistern (substitution with indexregister)	2.2
------	--	-----



Wirkung:

1. Es wird ein Befehl zusammengestellt und ausgeführt.
2. Der Inhalt des durch AD_m Bit 5 und 6 benannten Indexregisters ergibt den OP-Teil des Befehls.
Der Inhalt des durch AD_r Bit 1 und 2 benannten Indexregisters ergibt den AD_{lmr} -Teil des Befehls.
Der AD_i -Teil des Befehls wird gleich MC gesetzt.

Beispiel:

vorher:	I_0	<table border="1"><tr><td>0</td><td>3</td><td>5</td></tr></table>	0	3	5
0	3	5			
MC = 1	I_1	<table border="1"><tr><td>0</td><td>0</td><td>2</td></tr></table>	0	0	2
0	0	2			
	I_2	<table border="1"><tr><td>4</td><td>11</td><td>14</td></tr></table>	4	11	14
4	11	14			
	I_3	<table border="1"><tr><td>0</td><td>2</td><td>9</td></tr></table>	0	2	9
0	2	9			

entspricht OP-Teil von ICX

BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
	0 2 2	0	0 1 2		XF, 1, 2
	1 2 2	1	2 3 0		SSTX, XI, 3.
	2				

Es wird der Befehl SSTX, 3.2 ausgeführt, der seinerseits die Ausführung von ICX, XI, 4.11.14 bewirkt, d.h. der AD_{lmr} -Teil des Befehls unter Adresse 4.15.3 kommt nach Indexregister I_2 . Der Merker MC ist unverändert.

Besonderheit:

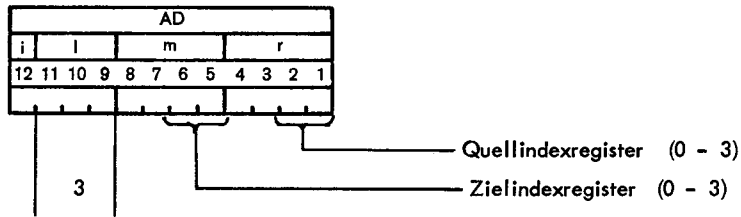
Dieser Befehl ist bei den Modellen 820/25/35 mit dem Betriebsprogramm SKZA2 nicht möglich.

SSTX	Substitution mit Indexregistern (substitution with indexregister)	2.2
------	--	-----

Bemerkung:

1. Für den Befehl "SSTX Substitution mit Indexregistern(2.2)" gelten die Bemerkungen 2 und 3 des Befehls "MVOX Befehlsspeicherung (2.2)".
2. Dieser Befehl gilt nicht für das Modell 820/15. Falls er dennoch angewendet wird, erfolgt kein interner Fehlerstop. Das Programm wird jedoch im weiteren Ablauf gestört.

CLX	Indexregister Komplementierung (complementation of indexregister)	2.2
-----	--	-----



Wirkung:

Der Inhalt des durch AD_r Bit 1 und 2 benannten Indexregisters (Quellindexregister) wird bezüglich $2^{11} = 2048$ komplementiert und in das durch AD_m Bit 5 und 6 benannte Indexregister (Zielindexregister) transportiert. Der Inhalt des Quellindexregisters bleibt unverändert.

Beispiel:

Der Inhalt des Indexregisters I_3 soll komplementiert und nach Indexregister I_2 gebracht werden.

vorher: I_3

3	2	13
---	---	----

 entspricht $(I_3) = 813$

BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
0	2 2	0	3 2 3		CLX, 2.3
1					

nachher: I_2

4	13	3
---	----	---

 entspricht $(I_2) = 1235 = 2048 - 813$
 I_3

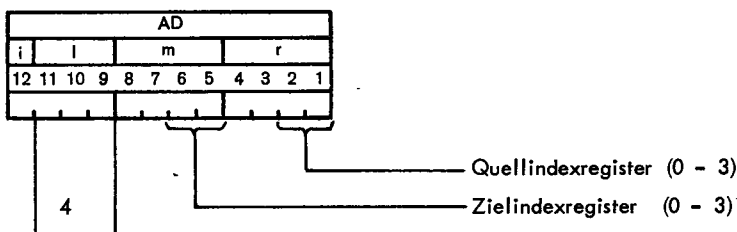
3	2	13
---	---	----

 entspricht $(I_3) = 813$

Bemerkungen:

- Der Befehl ist auch zulässig, wenn für Quell- und Zielindexregister dasselbe Indexregister genommen wird.
- Bit 3 und 4 sowie Bit 7 und 8 des AD-Teiles sind für die Ausführung des Befehls bedeutungslos.

ADX	Indexregisteraddition (addition indexregister)	2.2
-----	---	-----



Wirkung:

1. Der Inhalt des Quellindexregisters wird zum Inhalt des Zielindexregisters addiert. Das Ergebnis steht anschließend im Zielindexregister. Der Inhalt des Quellindexregisters bleibt unverändert.
2. Findet im Zielindexregister ein Überlauf über Bit 11 statt, wird Merker MC = 1 andernfalls MC = 0 gesetzt. Im Zielindexregister steht das Ergebnis der Addition modulo $2^{11} = 2048$.

Beispiel:

Indexregisteraddition $(I_3 + I_0) \rightarrow I_3$ mit Überlauf
 vorher: I_0

7	0	8
---	---	---

 entspricht $(I_0) = 1800$
 I_3

1	4	6
---	---	---

 entspricht $(I_3) = 326$

BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
0	2 2	0	4 3	0	ADX, 3.0
1					

nachher: I_0

7	0	8
---	---	---

 entspricht $(I_0) = 1800$
 MC = 1 I_3

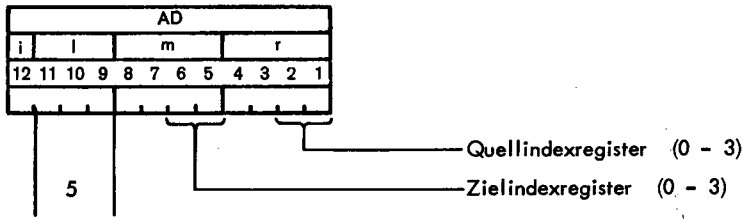
0	4	14
---	---	----

 entspricht $(I_3) = 78 = 2126 - 2048$

Bemerkungen:

1. Der Befehl ist auch zulässig, wenn für Quell- und Zielindexregister dasselbe Indexregister genommen wird.
2. Bit 3 und 4 sowie Bit 7 und 8 des AD-Teiles sind für die Ausführung des Befehles bedeutungslos.

SBX	Indexregistersubtraktion (subtraction indexregister)	2.2
-----	---	-----



Wirkung:

1. Der Inhalt des Quellindexregisters wird vom Inhalt des Zielindexregisters subtrahiert. Das Ergebnis steht anschließend im Zielindexregister. Der Inhalt des Quellindexregisters bleibt unverändert.
2. War der Inhalt des Quellindexregisters größer als der Inhalt des Zielindexregisters, wird Merker MC = 1 andernfalls MC = 0 gesetzt. Im Zielindexregister steht das Ergebnis der Subtraktion modulo $2^{11} = 2048$.

Beispiel:

Indexregistersubtraktion $(I_1) - (I_2) \rightarrow I_1$ mit $(I_2) > (I_1)$.

vorher: I_1

0	8	5
---	---	---

 entspricht $(I_1) = 133$
 I_2

0	9	0
---	---	---

 entspricht $(I_2) = 144$

BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen				
	0	2	2	0	5	1	2		
	1								SBX, 1,2

nachher: I_1

7	15	5
---	----	---

 entspricht $(I_1) = 2037 = -11 + 2048$
 MC = 1 I_2

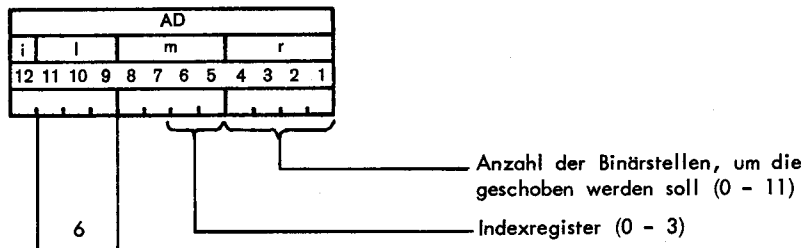
0	9	0
---	---	---

 entspricht $(I_2) = 144$

Bemerkungen:

1. Der Befehl ist auch zulässig, wenn für Quell- und Zielindexregister dasselbe Indexregister genommen wird.
2. Bit 3 und 4 sowie Bit 7 und 8 des AD-Teiles sind für die Ausführung des Befehles bedeutungslos.

SRX	Indexregisterrechtsshift (shift right indexregister)	2.2
-----	---	-----



Wirkung:

1. Der Inhalt des durch AD_m Bit 5 und 6 benannten Indexregisters wird um die in AD_r angegebene Anzahl von Binärstellen nach rechts verschoben. Die nach rechts hinausgeschobenen Stelleninhalte gehen verloren, von links werden ab Bit 11 Nullen nachgezogen.
2. Der Merker MC wird auf 1 gesetzt, wenn während des Shifts ein von 0 verschiedener Stelleninhalt nach rechts hinausgeschoben wurde. Andernfalls wird $MC = 0$ gesetzt.

Beispiel:

Rechtsshift in Indexregister I_2 um 3 Binärstellen.

vorher: I_2

3	11	2
---	----	---

 entspricht binär OLL LOLL OOLO

BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
0	2 2	0	6 2 3		SRX, 2.3
1					

nachher: I_2

0	7	6
---	---	---

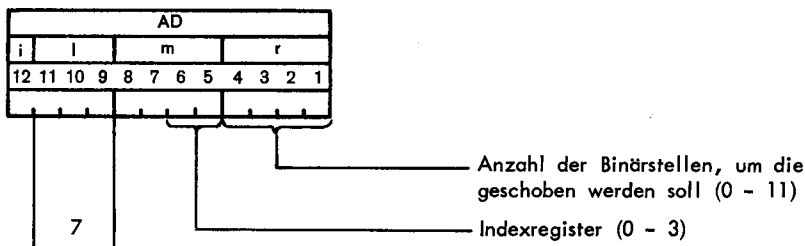
 entspricht binär OOO OLLL OLLO

MC = 1

Bemerkungen:

1. Das Verschieben eines binären Indexregisterinhalts um n Stellen nach rechts ist gleichbedeutend mit einer Division durch 2^n .
2. Ein Shift um mehr als 11 Stellen wird ausgeführt, doch ist er wenig sinnvoll, da nur das Indexregister gelöscht wird.
3. Bit 7 und 8 des AD-Teils sind für die Ausführung des Befehls bedeutungslos.

SLX	Indexregisterlinksshift (shift left indexregister)	2.2
-----	---	-----



Wirkung:

1. Der Inhalt des durch AD_m Bit 5 und 6 benannten Indexregisters wird um die in AD_r angegebene Anzahl von Binärstellen nach links verschoben. Die nach links über Bit 11 hinausgeschobenen Stelleninhalte gehen verloren, von rechts werden Nullen nachgezogen.
2. Der Merker MC wird auf 1 gesetzt, wenn während des Shifts ein von 0 verschiedener Stelleninhalt nach links über Bit 11 hinausgeschoben wurde. Andernfalls wird $MC = 0$ gesetzt.

Beispiel:

Linksshift in Indexregister I_1 um 6 Binärstellen.

vorher: I_1

0	1	14
---	---	----

 entspricht binär 000 000L LLL0

BW-Adresse		Op. Teil		l	Adreßteil			Symb. Adr.	Bemerkungen
0	2	2	0	7	1	6		SLX, 1.6	
1									

nachher: I_1

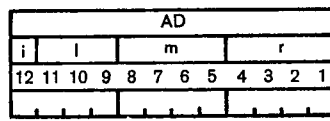
7	8	0
---	---	---

 entspricht binär LLL L000 0000
MC = 0

Bemerkungen:

1. Das Verschieben eines binären Indexregisterinhalts um n Stellen nach links ist gleichbedeutend mit einer Multiplikation mit 2^n .
2. Ein Shift um mehr als 11 Stellen wird ausgeführt, doch ist er wenig sinnvoll, da nur das Indexregister gelöscht wird.
3. Bit 7 und 8 des AD-Teils sind für die Ausführung des Befehls bedeutungslos.

RND	Runden (round)	2.3
-----	----------------	-----



Stelle, mit der gerundet werden soll (1 - 15)
Speicherwort, dessen Inhalt gerundet werden soll $\{0.1 - 7.10\}$ $[0.1 - 7.11]$

Wirkung:

Unabhängig vom Vorzeichen wird auf die in AD_r angegebene Stelle des durch AD_l_m benannten Speicherwortes eine 5 addiert. Dies bedeutet:
Ist der Inhalt des durch AD_l_m benannten Speicherwortes positiv oder Null, so wird auf die in AD_r angegebene Stelle eine 5 addiert, ist der Inhalt des Speicherwortes negativ, so wird eine 5 subtrahiert.
Findet dadurch ein Überlauf über Stelle 1 des Speicherwortes statt, wird $MC = 1$ andernfalls $MC = 0$ gesetzt. Danach werden die in AD_r angegebene Stelle und alle rechts anschließenden Kernspeicherstellen gelöscht.

Beispiel:

Im Speicherwort 22 soll mit Stelle 11 gerundet werden.

vorher: SW 22

0	0	0	0	0	0	0	0	0	1	2	7	3	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
	0 2 3	0	1 6 11		RND, 22.11
	1				

nachher: SW 22

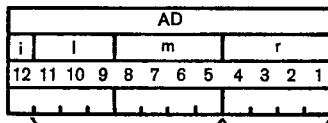
0	0	0	0	0	0	0	0	0	0	1	3	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

MC = 0

Bemerkungen:

1. Wenn $AD_r = 0$ ist, läuft das Programm auf Internfehlerstop (vgl. Beschreibung "Programm, Aufbau", Kapitel 2.3)
2. Für den Fall, daß der Inhalt der Kernspeicherstellen sedezimal ist, beachte man die Bemerkung 2 vom Befehl "ADH Addition (0.4)".

CNT	Zählen (count)	2.4
-----	----------------	-----



Stelle, in der gezählt werden soll (1 - 15)

Speicherwort, in dem gezählt werden soll $\{0.1 - 7.10\}$ $[0.1 - 7.11]$

Wirkung:

Unabhängig vom Vorzeichen wird auf die in AD_r angegebene Stelle des durch AD_{lm} benannten Speicherwortes eine 1 addiert. Dies bedeutet:
 Ist der Inhalt des in AD_{lm} benannten Speicherwortes positiv oder Null, so wird auf die durch AD_r gekennzeichnete Stelle eine 1 addiert, ist der Inhalt des Speicherwortes negativ, so wird eine 1 subtrahiert.
 Findet dadurch ein Überlauf über Stelle 1 des Speicherwortes statt, wird $MC = 1$ andernfalls $MC = 0$ gesetzt.

Beispiel:

Im Speicherwort 18 soll auf Stelle 3 eine 1 absolut addiert werden.

vorher: SW 18

-	9	9	9	1	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

BW-Adresse	Op. Teil			l	Adreßteil			Symb. Adr.	Bemerkungen
	0	2	4	0	1	2	3		CNT, 18.3
	1								

nachher: SW 18

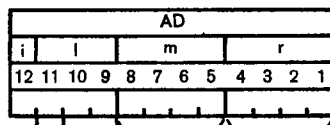
-	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

MC = 1

Bemerkungen:

1. Wenn $AD_r = 0$ ist, läuft das Programm auf Internfehlerstop (vgl. Beschreibung "Programm, Aufbau", Kapitel 2.3).
2. Für den Fall, daß der Inhalt der Kernspeicherstelle sedezimal ist, beachte man die Bemerkung 2 vom Befehl "ADH Addition (0.4)".

	Inhalt Indexregister nach Speicherwort (decimal conversion)	2.5
--	--	-----



		Stelle im Speicherwort 0 - 15	
		Speicherwort 1 - 15	
0	0	DC1	Transport nach der in AD_r angegebenen Stelle
	+1	DC2	zuzüglich 1 Stelle links
	+2	DC3	zuzüglich 2 Stellen links
	+3	DC4	zuzüglich 3 Stellen links
4	0	DC1, PL	Addition zum Inhalt der in AD_r angegebenen Stelle
	+1	DC2, PL	zuzüglich 1 Stelle links
	+2	DC3, PL	zuzüglich 2 Stellen links
	+3	DC4, PL	zuzüglich 3 Stellen links

Wirkung:

1. Bei Bit 11 = 0: Der Inhalt des Indexregisters wird in eine Dezimalzahl umgewandelt und in das durch AD_m benannte Speicherwort von der in AD_r angegebenen Stelle ab nach links übertragen. Bit 9 und Bit 10 geben an, wieviele Stellen zusätzlich zu der in AD_r angegebenen Stelle nach links hin überschrieben werden sollen. Alle weiteren Stellen des Speicherwortes bleiben unverändert.
Benötigt der Inhalt des Indexregisters dezimal mehr Stellen als im Befehl zugelassen sind, so erfolgt der Transport nur nach den angegebenen Stellen und die Überzähligen gehen verloren. In diesem Fall wird der Merker $MC = 1$ andernfalls $MC = 0$ gesetzt.

2. Bei Bit 11 = 1: Der Inhalt des Indexregisters wird dezimal zu den angegebenen Kernspeicherstellen addiert. Die übrigen Stellen bleiben wieder unverändert. Entsteht bei der Addition ein Überlauf über die zugelassenen Stellen hinaus, wird Merker $MC = 1$ gesetzt, die davorliegenden Kernspeicherstellen bleiben unverändert. Andernfalls ist $MC = 0$.

	Inhalt Indexregister nach Speicherwort (decimal conversion)	2.5
--	--	-----

Beispiel 1:

Transport Inhalt Indexregister I_1 nach Speicherwort 6 Stellen 2 bis 4.

vorher: I_1

0	1	15
---	---	----

 entspricht (I_1) = 31

SW 6

9	4	11	7	5	10														
---	---	----	---	---	----	--	--	--	--	--	--	--	--	--	--	--	--	--	--

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
0	2 2	0	0 0 1		XF, 1
1	2 5	0	2 6 4		DC3, 6.4
2					

nachher: SW 6

9	0	3	1	5	10														
---	---	---	---	---	----	--	--	--	--	--	--	--	--	--	--	--	--	--	--

MC = 0 (kein Überlauf)

Der Inhalt des Indexregisters kann vollständig in 2 Kernspeicherstellen gespeichert werden. Die nicht benutzte dritte Stelle vorn ist deshalb gelöscht.

Beispiel 2:

Addition Inhalt Indexregister I_2 zum Inhalt Speicherwort 11 Stellen 13 und 14.

vorher: I_2

0	4	5
---	---	---

 entspricht (I_2) = 69

SW 11

												3	12	8	7	1
--	--	--	--	--	--	--	--	--	--	--	--	---	----	---	---	---

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
0	2 2	0	0 0 2		XF, 2
1	2 5	0	5 11 14		DC2, PL, 11.14
2					

nachher: SW 11

												3	12	5	6	1
--	--	--	--	--	--	--	--	--	--	--	--	---	----	---	---	---

MC = 1 (Überlauf)

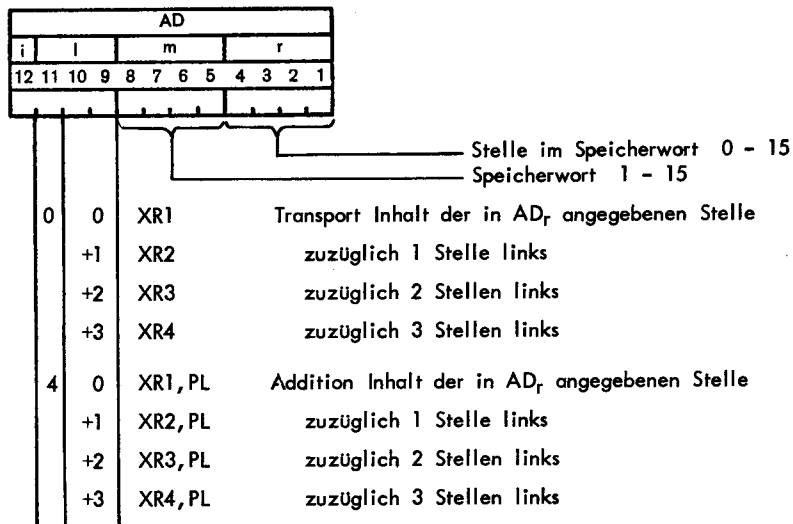
	Inhalt Indexregister nach Speicherwort (decimal conversion)	2.5
--	--	-----

Durch die Addition entsteht die 3-stellige Dezimalzahl 156, von der nur die beiden letzten Stellen gespeichert werden. Die davorliegende Kernspeicherstelle bleibt unverändert.

Bemerkungen:

1. Bei der Addition des Indexregisterinhalts zu einem sedezimalen Stelleninhalt beachte man Bemerkung 2 zum Befehl "ADH Addition (0.4)".
2. Man beachte den Indexvorbefehl (vgl. Beispiele).

	Transport nach Indexregister lmr (indexregister conversion to right)	2.6
--	---	-----



Wirkung:

1. Aus dem in AD_m angegebenen Speicherwort wird der Inhalt der in AD_r angegebenen Stelle zuzüglich der in Bit 9 und Bit 10 angegebenen Anzahl von Stellen links davon entnommen. Dadurch entsteht eine 1-, 2-, 3- oder 4-stellige Dezimalzahl, die in eine Binärzahl umgewandelt wird.
2. Bei Bit 11 = 0: Die Binärzahl wird in ein Indexregister gebracht. Ist sie größer als die Kapazität des Indexregisters, so gehen die nach links überzähligen Binärstellen verloren. In diesem Falle ist MC = 1, sonst MC = 0.
3. Bei Bit 11 = 1: Die Binärzahl wird zum Inhalt des Indexregisters addiert. Das Einstellen der Summe in das Indexregister sowie das Setzen des Merkers MC geschieht wie im Falle Bit 11 = 0.

	Transport nach Indexregister l _{mr} (indexregister conversion to right)	2.6
--	---	-----

Beispiel 1:

Transport Inhalt Speicherwort 12 Stellen 0 und 1 nach I_{1mr} von Indexregister I₁ bei sedezimalem Speicherwortinhalt.

vorher: I₁

5	13	8
---	----	---

SW 12

1	13	4																	
---	----	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

BW-Adresse			Op. Teil			Adreßteil			Symb. Adr.	Bemerkungen
		0	2	2	0	0	0	1		XF, 1
		1	2	6	0	1	12	1		XR2, 12.1
		2								

nachher: I₁

0	1	7
---	---	---

 entspricht (I₁) = 23

MC = 0 (kein Überlauf)

Die Zahl 13 in Stelle 1 wurde also richtig umgewandelt.

Beispiel 2:

Addition Inhalt Speicherwort 9 Stellen 7 bis 10 zum Inhalt I_{1mr} von Indexregister I₃.

vorher: I₃

0	12	6
---	----	---

 entspricht (I₃) = 198

SW 9

						3	1	9	6	2	4								
--	--	--	--	--	--	---	---	---	---	---	---	--	--	--	--	--	--	--	--

BW-Adresse			Op. Teil			Adreßteil			Symb. Adr.	Bemerkungen
		0	2	2	0	0	0	3		XF, 3
		1	2	6	0	7	9	10		XR4, PL, 9.10
		2								

nachher: I₃

0	7	0
---	---	---

 entspricht (I₃) = 112 = 2160 - 2048

MC = 1 (Überlauf im Indexregister)

Blatt B 78

Datenverarbeitungssystem 820

1. 9. 1970

Beschreibung der Befehle

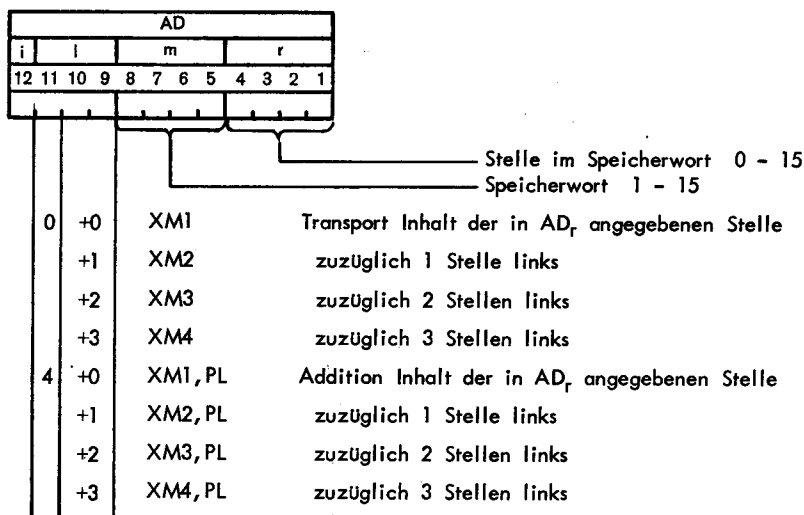
	Transport nach Indexregister I_{mr} (indexregister conversion to right)	2.6
--	--	-----

Durch die Addition entsteht eine 12-stellige Binärzahl, wovon Bit 12 verloren geht, da im Indexregister nur die letzten 11 Bit gespeichert werden können.

Bemerkungen:

1. Der Befehl wird auch bei sedezimalem Speicherwortinhalt richtig ausgeführt.
2. Man beachte den Indexvorbefehl (vgl. Beispiel).

	Transport nach Indexregister Im (indexregister conversion to midth)	2.7
--	--	-----



Wirkung:

1. Aus dem in AD_m angegebenen Speicherwort wird der Inhalt der in AD_r angegebenen Stelle zuzüglich der in Bit 9 und Bit 10 angegebenen Anzahl von Stellen links davon entnommen. Dadurch entsteht eine 1-, 2-, 3- oder 4-stellige Dezimalzahl, die in eine Binärzahl umgewandelt und um 4 Binärstellen nach links verschoben wird, wobei von rechts Nullen nachgezogen werden. Dieses entspricht einer Multiplikation von $2^4 = 16$. Dadurch entsteht eine neue Binärzahl.
2. Bei Bit 11 = 0: Die Binärzahl wird in ein Indexregister gebracht. Ist sie größer als die Kapazität des Indexregisters, so gehen die nach links überzähligen Binärstellen verloren. In diesem Falle ist MC = 1, sonst MC = 0.
3. Bei Bit 11 = 1: Die Binärzahl wird zum Inhalt des Indexregisters addiert. Das Einstellen der Summe in das Indexregister sowie das Setzen des Merkers MC geschieht wie im Falle Bit 11 = 0.

	Transport nach Indexregister I_m (indexregister conversion to midth)	2.7
--	---	-----

Beispiel 1:

Transport Inhalt Speicherwort 8 Stellen 3 bis 6 nach I_m von Indexregister I_1 .

vorher: I_1

5	14	2
---	----	---

SW 8

		6	2	0	4	3	1								
--	--	---	---	---	---	---	---	--	--	--	--	--	--	--	--

BW-Adresse	Op. Teil			I	Adreßteil			Symb. Adr.	Bemerkungen
	0	2	2	0	0	0	1		XF, 1
	1	2	7	0	3	8	6		XM4, 8.6
	2								

nachher: I_1

7	11	0
---	----	---

 entspricht $(I_1) = 123 \cdot 2^4$

MC = 1 (Überlauf im Indexregister)

Beispiel 2:

Addition Inhalt Speicherwort 14 Stellen 11 bis 13 zum Inhalt I_m von Indexregister I_2 .

vorher: I_2

1	3	14
---	---	----

 entspricht $(I_2) = 318$

SW 14

										5	1	0	2	8
--	--	--	--	--	--	--	--	--	--	---	---	---	---	---

BW-Adresse	Op. Teil			I	Adreßteil			Symb. Adr.	Bemerkungen
	0	2	2	0	0	0	2		XF, 2
	1	2	7	0	6	14	13		XM3, PL, 14.13
	2								

nachher: I_2

7	9	14
---	---	----

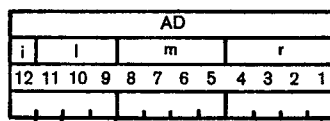
 entspricht $(I_2) = 1950 = 318 + 102 \cdot 2^4$

MC = 0 (kein Überlauf)

Bemerkungen:

1. Der Befehl wird auch bei sedezimalem Speicherwortinhalt richtig ausgeführt.
2. Man beachte den Indexvorbefehl (vgl. Beispiele).

	Transport nach Indexregister I_j (indexregister conversion to left)	2.8
--	--	-----



			Stelle im Speicherwort 0 - 15
			Speicherwort 1 - 15
0	+0	XL1	Transport Inhalt der in AD_r angegebenen Stelle
	+1	XL2	zuzüglich 1 Stelle links
	+2	XL3	zuzüglich 2 Stellen links
	+3	XL4	zuzüglich 3 Stellen links
4	+0	XL1, PL	Addition Inhalt der in AD_r angegebenen Stelle
	+1	XL2, PL	zuzüglich 1 Stelle links
	+2	XL3, PL	zuzüglich 2 Stellen links
	+3	XL4, PL	zuzüglich 3 Stellen links

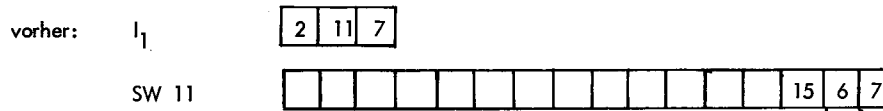
Wirkung:

1. Aus dem in AD_m angegebenen Speicherwort wird der Inhalt der in AD_r angegebenen Stelle zuzüglich der in Bit 9 und Bit 10 angegebenen Anzahl von Stellen links davon entnommen. Dadurch entsteht eine 1-, 2-, 3- oder 4-stellige Dezimalzahl, die in eine Binärzahl umgewandelt und um 8 Binärstellen nach links verschoben wird, wobei von rechts Nullen nachgezogen werden. Dieses entspricht einer Multiplikation mit $2^8 = 256$. Dadurch entsteht eine neue Binärzahl.
2. Bei Bit 11 = 0: Die Binärzahl wird in ein Indexregister gebracht. Ist sie größer als die Kapazität des Indexregisters, so gehen die nach links überzähligen Binärstellen verloren. In diesem Fall ist $MC = 1$, sonst $MC = 0$.
3. Bei Bit 11 = 1: Die Binärzahl wird zum Inhalt des Indexregisters addiert. Das Einstellen der Summe in das Indexregister sowie das Setzen des Merkers MC geschieht wie im Fall Bit 11 = 0.

	Transport nach Indexregister I ₁ (indexregister conversion to left)	2.8
--	---	-----

Beispiel 1:

Transport Inhalt Speicherwort 11 Stelle 14 nach I₁ von Indexregister I₁.



BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
0	2 2	0	0 0 1		XF, 1
1	2 8	0	0 11 14		XL1, 11.14
2					

nachher: I₁

6	0	0
---	---	---

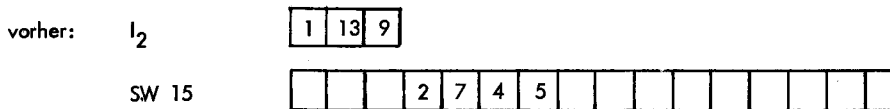
 entspricht $(I_1) = 6 \cdot 2^8$

MC = 0 (kein Überlauf)

Die Zahl 6 kann vollständig in I₁ dargestellt werden.

Beispiel 2:

Addition Inhalt Speicherwort 15 Stellen 4 und 5 zum Inhalt I₁ von Indexregister I₂.



BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
0	2 2	0	0 0 2		XF, 2
1	2 8	0	5 15 5		XL2, PL, 15.5
2					

nachher: I₂

3	13	9
---	----	---

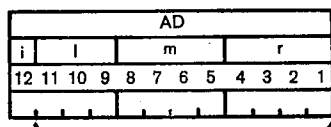
MC = 1 (Überlauf im Indexregister)

Der Überlauf geht verloren, I_{mr} bleibt unverändert.

Bemerkungen:

1. Der Befehl wird auch bei sedezimalem Speicherwortinhalt richtig ausgeführt.
2. Man beachte den Indexvorbefehl (vgl. Beispiele).

ICX	Indirekte Konstante nach Indexregister (indirect constant to indexregister)	2.9
-----	--	-----



— Befehlsadresse (0.0.0 - 7.15.15)

Wirkung:

Es wird der Befehl aufgesucht, dessen Adresse in AD_{l_{mr}} angegeben ist. Die im AD_{l_{mr}}-Teil dieses Befehles sedezimal dargestellte Zahl wird in ein Indexregister gebracht.

Beispiel:

Der Befehl "ICX Indirekte Konstante nach I (2.9)" wird indiziert gegeben. Es soll dabei die Indizierung mit Indexregister I₂ durchgeführt werden und der Transport nach Indexregister I₁ erfolgen.

vorher: I₁

2	5	13
---	---	----

I₂

3	12	9
---	----	---

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
0	2 2	0	0 2 1		XF, 2.1
1	2 9	1	0 0 0		ICX, XI
2					

Von dem folgenden Befehl wird der AD_{l_{mr}}-Teil nach I₁ gebracht.

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
3	12 9 0 4	1	1 15 2		ADH, XI, 31.2

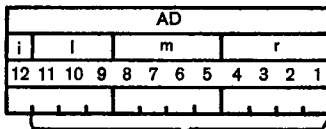
nachher: I₁

1	15	2
---	----	---

Bemerkungen:

1. Der Befehl "ICX Indirekte Konstante nach I (2.9)" ist nur indiziert von Bedeutung (vgl. Beispiel).
2. Man beachte die Befehle "XF und XFR Indexvorbefehle (2.2)" und "BL Blockumschaltung (2.11)".

CX	Konstante nach Indexregister (constant to indexregister)	2.10
----	---	------



Konstante (0.0.0 - 7.15.15)

Wirkung:

Die im AD_{l_{mr}}-Teil sedezimal dargestellte Zahl wird in ein Indexregister gebracht.

Beispiel:

Der Inhalt von Indexregister I₂ soll um 17 erhöht werden. Dazu wird der Befehl "CX Konstante nach I (2.10)" indiziert gegeben.

vorher: I₂

0	11	7
---	----	---

 entspricht (I₂) = 183

BW-Adresse	Op. Teil			I	Adreßteil			Symb. Adr.	Bemerkungen
	0	2	2	0	0	2	2		XF, 2.2
	1	2	10	1	0	1	1		CX, XI, 17
	2								

nachher: I₂

0	12	8
---	----	---

 entspricht (I₂) = 200

Im Indexvorbefehl wurde in AD_m und AD_r das Indexregister I₂ angegeben. Die Indizierung bewirkt die Addition der Konstanten 17 zum vorherigen Indexregisterinhalt, der Transport erfolgt wieder nach Indexregister I₂.

Bemerkung:

Wird der Befehl "CX Konstante nach I (2.10)" indiziert gegeben, so erfolgt eine Addition der Konstanten in AD_{l_{mr}} zum Inhalt des Indexregisters (vgl. Beispiel). Diese Rechnung geschieht modulo 2¹¹, so daß mit einer geeigneten Konstanten auch eine Subtraktion im Indexregister durchgeführt werden kann.

	Kugelkopfposition nach Indexregister (position to indexregister)	2.11
--	---	------

AD											
i	l	m						r			
12	11	10	9	8	7	6	5	4	3	2	1
0		0						2			
								3			

POSX Kugelkopf vom Drucker 1
{POSXS Kugelkopf vom Drucker 2}

Wirkung:

Die Position des Kugelkopfes des {jeweiligen} Druckers wird in ein Indexregister gebracht.

Beispiel für Modell 820/25 und 820/35:

Die Position des Kugelkopfes vom Drucker 2 soll nach Indexregister I₁ gebracht werden. Der Kugelkopf stehe in Position 50.

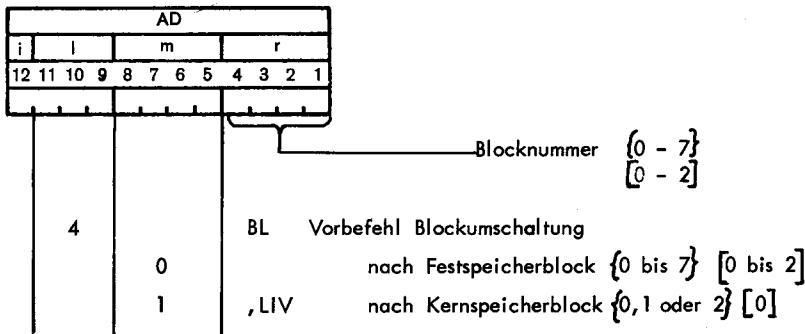
BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
0	2 2	0	0 0 1		XF, 1
1	2 11	0	0 0 3		POSXS
2					

nachher: I₁ 0 3 2 entspricht (I₁) = 50

Bemerkungen:

1. Tritt durch Simultanarbeit der Fall ein, daß die Position des Kugelkopfes abgefragt wird, während ein Ausdruck oder eine Tabulation stattfindet, so wartet das Betriebsprogramm mit der Abfrage solange, bis die Tätigkeit des Druckers beendet ist. Es braucht also kein gesonderter Wartebefehl gegeben zu werden.
2. Die erste Kugelkopfposition links ist die Position 0, die letzte Kugelkopfposition rechts ist Position 130.
3. Es ist zu beachten, daß beim Modell 820/15 kein zweiter Drucker angeschlossen werden kann. Der Befehl POSXS darf also nicht angewendet werden!

BL	Blockumschaltung (block)	2.11
----	--------------------------	------

Wirkung:

- Der Befehl Blockumschaltung ist ein Vorbefehl und wirkt auf den nächstfolgenden Befehl, in dessen AD_{l_{mr}}-Teil eine Befehlsadresse steht. Außerdem wirkt er bei dem Befehl "MVOX Befehlsspeicherung (2.2)". Nach der Ausführung eines solchen Befehles ist der Vorbefehl Blockumschaltung aufgehoben, auch wenn seine Bedeutung nicht ausgenutzt wurde (z.B. bei bedingtem Sprungbefehl).

- Der Befehl Blockumschaltung wirkt bei folgenden Befehlen:

BR	Unbedingter Sprung	(1.0)
BR1 bis BRC	Sprung, wenn Merker auf 1	(1.1 bis 1.8)
BXG	Sprung, wenn (l) größer 1023	(1.9)
BXU	Sprung, wenn (l) ungleich 0	(1.10)

Wird der Befehl Blockumschaltung vor diesen Befehlen gegeben, und ist die jeweilige Bedingung erfüllt, so fährt das Programm in dem Block fort, der durch den Befehl Blockumschaltung vorgeschrieben wurde. Ist die Bedingung nicht erfüllt, so bleibt das Programm in dem Block, in dem der Sprungbefehl steht. Nach jedem dieser Befehle ist der Befehl Blockumschaltung aufgehoben.

BRS	Unterprogrammprung	(1.11)
-----	--------------------	--------

Steht der Befehl Blockumschaltung vor dem Unterprogrammprung, so wird das Unterprogramm in dem Block gesucht, der durch den Befehl Blockumschaltung bestimmt wurde. Nach Ausführung des Unterprogrammes fährt das Programm in dem Block fort, in dem der Befehl "BRS Unterprogrammprung (1.11)" gegeben wurde.

BL	Blockumschaltung (block)	2.11
----	--------------------------	------

Unabhängig davon können im Laufe des Unterprogrammes weitere Blockumschaltungen durchgeführt werden. Ein Befehl Blockumschaltung hat auf den Rücksprungbefehl keine Wirkung und wird durch diesen also auch nicht gelöscht.

SST Substitution (1.13)

Steht der Befehl Blockumschaltung vor dem Befehl Substitution, so wird ein Befehl aus dem Block substituiert, der durch den Befehl Blockumschaltung bestimmt wurde. Für den substituierten Befehl gilt der Befehl Blockumschaltung nicht mehr (vgl. Beispiel). Das Programm läuft anschließend im ursprünglichen Block weiter.

ICA Indirekte Konstante nach A (1.14)
ICX Indirekte Konstante nach I (2.9)
TT Drucken Tabelle (3.0)
OPX OP- und AD_i-Teil nach I (3.14)

Ein Befehl Blockumschaltung vor diesen Befehlen bewirkt, daß die Adressen jeweils in dem Block aufgesucht werden, der durch den Befehl Blockumschaltung bestimmt wurde. Das Programm fährt im ursprünglichen Block fort.

Beispiel:

Die Wirkung des Befehles Blockumschaltung auf den Befehl Substitution:

Block 0

BW-Adresse		Op. Teil			I	Adreßteil			Symb. Adr.	Bemerkungen
5	13	3	2	11	0	4	0	2		BL, 2
		4	1	13	0	1	11	15		SST, ALPHA

Der folgende Befehl wird substituiert:

Block 2

BW-Adresse		Op. Teil			I	Adreßteil			Symb. Adr.	Bemerkungen
1	11	15	3	0	0	5	13	6	ALPHA	TT, TABEL
										Block 0
5	13	6							TABEL	

BL	Blockumschaltung (block)	2.11
----	--------------------------	------

Das Programm fährt nach Ausdruck der Tabelle im Block 0 mit dem Befehl unter Adresse 5.13.5 fort.

Bemerkungen:

1. Läuft ein Programm linear, z.B. von Block 1 nach Block 2, so muß der vorletzte Befehl in Block 1 der Befehl Blockumschaltung und der letzte Befehl ein Sprungbefehl auf den ersten Befehl in Block 2 sein. In allen Blöcken außer Block 0 stehen die Befehlsadressen 0.0.0 bis 7.15.15 zur Verfügung.

2. Mit den Befehlen "INCX und XINC Transport Befehlszähler (2.2)" kann die Blocknummer einer Programmstufe in ein Indexregister gebracht bzw. neu geladen werden. Dabei entsprechen dem Inhalt 0.0.0 bis 0.0.7 von Indexregister I₁ die Festspeicherblöcke 0 bis 7 und dem Inhalt 0.1.0 bis 0.1.2 die Kernspeicherblöcke 0 bis 2.

3. Bit 4 und Bit 6 bis Bit 8 des AD-Teiles sind für die Ausführung des Befehles bedeutungslos. Wird ein Befehl Blockumschaltung mit einer zulässigen Blocknummer gegeben, doch ist das Chassis mit dem angegebenen Block nicht bestückt, so stoppt das Betriebsprogramm bei der Ausführung der Blockumschaltung.

Bei einer Blockumschaltung in einen Kernspeicherblock mit größerer Blocknummer als 2 wird folgendermaßen umgeschaltet:

- bei Block 3 nach Kernspeicherblock 1
- bei Block 4 nach Kernspeicherblock 2
- bei Block 5 nach Kernspeicherblock 1
- bei Block 6 nach Kernspeicherblock 2
- bei Block 7 nach Kernspeicherblock 2.

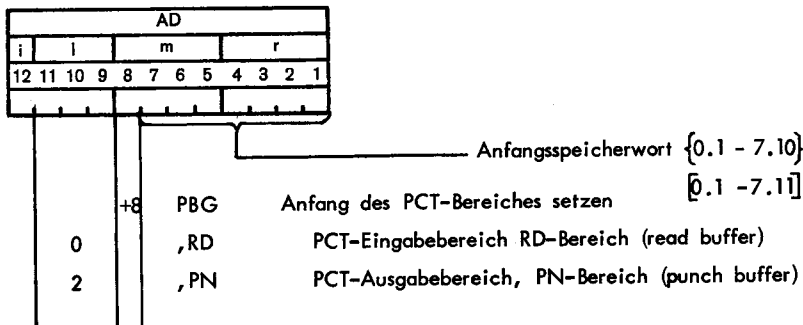
4. Das Modell 820/35 kann maximal 8 Festspeicherblöcke (0 - 7) und 3 Kernspeicherblöcke (0 - 2) enthalten. Werden nicht mehr als 506 Speicherworte benutzt, können bis zu 6000 Befehls Worte im Kernspeicher abgestellt werden. Bei Verwendung bis zu 1024 Speicherworten sind 4096 Befehls Worte im Kernspeicher möglich.

5. Das Modell 820/25 kann maximal einen Kernspeicherblock (0) und 4 Festspeicherblöcke (0 - 3) oder 6 Festspeicherblöcke ohne Kernspeicher enthalten.

BL	Blockumschaltung (block)	2.11
----	--------------------------	------

6. Das Modell 820/15 kann bis zu 3 Festspeicherblöcke (0 - 2) und einen Kernspeicherblock (0) enthalten. Der Kernspeicherblock 0 kann nur in Verbindung mit dem GP-Modul angesprochen werden.
7. Der Anruf nicht vorhandener Blöcke ist zu vermeiden!
8. Die Codierung im AD_m ist abhängig vom Einschaltmodus:
- Start des Programms im Kernspeicherblock
- 0 $\hat{=}$ Kernspeicherblock
 - 1 $\hat{=}$ Festspeicherblock
- Start des Programms im Festspeicherblock
- 0 $\hat{=}$ Festspeicherblock
 - 1 $\hat{=}$ Kernspeicherblock

PBG	Anfang PCT-Bereich setzen (PCT-buffer begin)	2.11
-----	---	------



Wirkung:

1. Das durch Bit 1 bis Bit 7 des AD-Teils benannte Speicherwort wird das erste Speicherwort des jeweiligen PCT-Bereiches.
2. Der PCT-Eingabebereich dient zum schnellen Einlesen von Lochkarten und Lochstreifen. Der PCT-Ausgabebereich enthält die Information, die bei der Ausgabe auf die Lochkarte übertragen wird.
3. Der Eingabebereich dient zur Ein- und Ausgabe von Daten auf Magnetband-cassette.

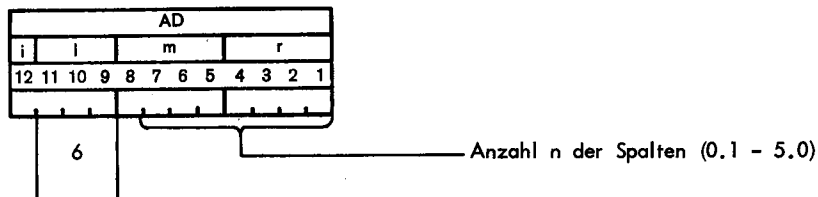
Bemerkungen:

1. Der Anfang des PCT-Ausgabebereiches, der Anfang des ALC-Bereiches und der Magnetkontenvorbefehl werden intern in der gleichen Speicherzelle gemerkt. Man beachte also, daß sich die Befehle "PBG,PN (2.11)" und "ALBG (3.4)" gegenseitig aufheben.
2. Wird nacheinander in denselben Bereich eingelesen bzw. aus demselben Bereich ausgegeben, so braucht der Anfang des betreffenden Bereiches nur einmal vorgegeben zu werden.
3. Liegt der Anfang eines PCT-Bereiches in den Speicherworten 1 bis 4, so ist darauf zu achten, daß diese Speicherworte nach dem Einlesen bzw. vor dem Ausgeben durch andere Operationen nicht verändert werden.

PBG	Anfang PCT-Bereich setzen (PCT-buffer begin)	2.11
-----	---	------

4. Der PCT-Ausgabebereich dient nur zur Ausgabe auf Lochkarten. Da das Betriebsprogramm bei der Ausgabe simultan arbeitet, ist darauf zu achten, daß der Inhalt des Ausgabebereiches erst nach Beendigung des Stanzens verändert werden darf (vgl. Warte- und Merkersetzbefehl 2.12).
5. Der Befehl PBG,RD/PN kann nur bei Verwendung entsprechender Moduln angewendet werden:
- Eingabe/Ausgabebereich für Magnetbandcassette Modul TC
Eingabe/Ausgabebereich für Lochstreifen oder Lochkarte Modul GP

MVBLK	Löschen PCT-Ausgabebereich (move blank)	2.11
-------	--	------

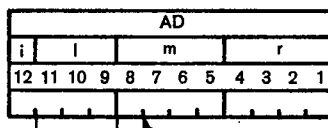
Wirkung:

1. Es werden im PCT-Ausgabebereich von der durch den Ausgabebereichszeiger bestimmten Stelle ab 2 n Kernspeicherstellen gelöscht. Das entspricht beim Befehl "PCH,PN Lochkarte stanzen (3.8)" der Ausgabe von n Leerspalten auf der Lochkarte.
2. Nach Ausführung des Befehles ist der Ausgabebereichszeiger um 2 n weitergestellt.

Bemerkungen:

1. Für den Befehl "MVBLK Löschen Ausgabebereich (2.11)" gelten die gleichen Bemerkungen wie für den Befehl "PBP PCT-Bereichszeiger laden (2.11)".
2. Beim Modell 820/15 ist dieser Befehl nur in Verbindung mit dem GP- und PC-Modul anwendbar.

PBP	PCT-Bereichzeiger laden (load PCT-buffer pointer)	2.11
-----	--	------



└──────────────────────────┘ Nummer (0.1 - 7.15)

	+0	PBP	Laden des PCT-Bereichzeigers
1		,RD	Eingabebereichzeiger RDP (read buffer pointer)
3		,PN	Ausgabebereichzeiger PNP (punch buffer pointer)

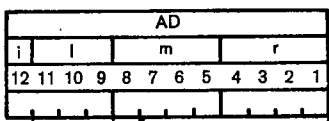
Wirkung:

Der jeweilige Bereichzeiger wird mit der in AD_{mr} Bit 1 bis Bit 7 angegebenen Nummer geladen.

Bemerkungen:

1. Bei der Verarbeitung von Lochkarten enthalten die Bereichszeiger die Nummer der Lochkartenspalte, die der augenblicklichen Stelle im Bereich entspricht.
 2. Bei der Verarbeitung von Lochstreifen und Magnetband steht im Eingabebereichszeiger die Nummer des gelesenen Zeichens.
 3. Die PCT-Bereichszeiger stehen nach einem Transport zwischen dem Bereich und einem anderen Speicher (ALC-Bereich, Speicherwort A oder Indexregister) auf der Stelle, die dem zuletzt verarbeiteten Zeichen folgt.
4. Es ist zu beachten, daß die entsprechenden Moduln vorhanden sind (siehe Befehl "PBG,2.11" 820/15).

PBPX	Inhalt PCT-Bereichszeiger nach I (PCT-buffer pointer to indexregister)	2.11
------	---	------



Nummer n (0.1 - 7.15)

	+8	PBPX	Inhalt PCT-Bereichszeiger nach Indexregister
1		,RD	Eingabebereichszeiger RDP (read buffer pointer)
3		,PN	Ausgabebereichszeiger PNP (punch buffer pointer)

Wirkung:

Der Inhalt des jeweiligen Bereichszeigers wird in ein Indexregister gebracht und mit der in AD_{mr} Bit 1 bis Bit 7 angegebenen Nummer n verglichen.

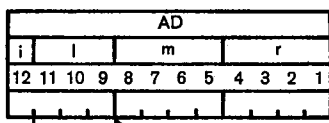
Die Merker werden auf Grund des Vergleiches folgendermaßen gesetzt:

			ML	MU
(Bereichszeiger)	<	n	1	1
(Bereichszeiger)	>	n	0	1
(Bereichszeiger)	=	n	0	0

Bemerkung:

- Für den Befehl "PBPX Inhalt PCT-Bereichszeiger nach I (2.11)" gelten die gleichen Bemerkungen wie für den Befehl "PBP PCT-Bereichszeiger laden (2.11)".
- Dieser Befehl kann nur bei den entsprechenden Moduln angewendet werden
siehe Befehl "PBG, (2.11)" (820/15).

	Warte- und Merkersetzbefehl	2.12
--	-----------------------------	------



Codierung der Bedingung

0	WT	Warte bis Bedingung erfüllt
1	SM1	Setze Merker M1 = 1, wenn Bedingung erfüllt
2	SM2	Setze Merker M2 = 1, wenn Bedingung erfüllt
3	SM3	Setze Merker M3 = 1, wenn Bedingung erfüllt
4	SM4	Setze Merker M4 = 1, wenn Bedingung erfüllt
5	SM5	Setze Merker M5 = 1, wenn Bedingung erfüllt
6	SML	Setze Merker ML = 1, wenn Bedingung erfüllt
7	SMU	Setze Merker MU = 1, wenn Bedingung erfüllt

Wirkung:

1. Ist $AD_i = 0$, so wird das Programm erst dann mit dem nächsten Befehl fortgesetzt, wenn die in AD_{mr} codierte Bedingung erfüllt ist.
2. Ist $AD_i \neq 0$, so wird der durch AD_i angegebene Merker auf 1 gesetzt, wenn die in AD_{mr} codierte Bedingung erfüllt ist, auf 0 gesetzt, wenn die in AD_{mr} codierte Bedingung nicht erfüllt ist.
Nach Abfrage der Bedingung und Setzen der Merker wird das Programm ohne Wartezeit fortgesetzt.

Bemerkung:

Ein unzulässiger Code in AD_{mr} wird vom Betriebsprogramm nicht erkannt, d.h. es findet kein Fehlerstopp statt. Man beachte, daß ein solcher Code unter Umständen eine interne Bedingung abfragt, die stets oder auch nie erfüllt sein kann.

	Warte- und Merkersetzbehl	2.12
--	---------------------------	------

AD													
i	l	m						r					
12	11	10	9	8	7	6	5	4	3	2	1		
	0				0						0	WTC	Warte auf Taste C Gleichzeitig leuchtet die Kontrolllampe rot
	0				0						1	WTF	Warte auf Taste F Gleichzeitig leuchtet die Kontrolllampe orange
	≠ 0											SM1 bis SM5 SML und SMU	Der in AD _i angegebene Merker wird
					0						0	,ZERO	auf 0 gesetzt,
					3						12	,ONE	auf 1 gesetzt.

Wirkung:

- Das Programm fährt erst dann mit dem nächsten Befehl fort, wenn die C- bzw. F-Taste an der Zehnertastatur betätigt wurde. Während des Wartens auf die Taste leuchtet die dazugehörige Kontrolllampe rot bzw. orange auf, nach Betätigen der Taste erlischt sie. Ferner sind während des Wartens die Zehnertastatur sowie die Auslösetasten gesperrt.
- Durch die Codierung 0.0 bzw. 3.12 kann der in AD_i angegebene Merker unbeding auf 0 bzw. 1 gesetzt werden.

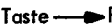





Bemerkungen:

- Die C-Taste dient im allgemeinen zur Korrektur von Externfehlern, die als Folge falscher manueller Eingaben auftreten. Sie befindet sich links neben der Zehnertastatur.
- Die F-Taste dient im allgemeinen zur Fortsetzung des Programmes nach der Behebung eines Fehlers bei einem externen Datenträger-Gerät. Sie ist die oberste Taste der am weitesten rechts gelegenen Tastenreihe.

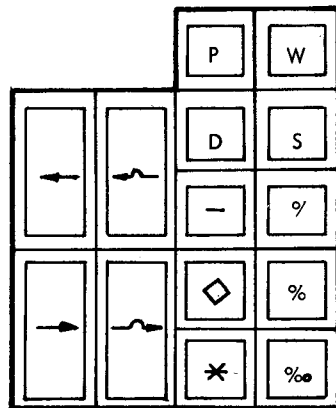
	Codierungen der Tastatur	2.12
--	--------------------------	------

AD												
i				l				m				r
12	11	10	9	8	7	6	5	4	3	2	1	

Auslösetasten:

MRAR	Taste 	Pfeil rechts (right arrow)
MLAR	Taste 	Pfeil links (left arrow)
MRSK	Taste 	Übersprung rechts (right skip)
MLSK	Taste 	Übersprung links (left skip)
MFTS	Taste 	Endsumme (final total sign)
MITS	Taste 	Zwischensumme (intermediate total sign)
MMIN	Taste -	Minus
MD	Taste D	Datum
MP	Taste P	Programmwahl
MPML	Taste %•	Promille
MPCT	Taste %	Prozent
MPD	Taste %'	Prodez
MS	Taste S	Symbol
MW	Taste W	Wortänderung

	Tastatur	2.12
--	----------	------

Wirkung:

- Die Auslösetasten sind mechanisch nicht arretierend und frei programmierbar. Das Betriebsprogramm übernimmt folgende Steuerung:

Beim Drücken einer der Tasten erfolgt
 Sperrung der Zehnertastatur (außer C-Taste),
 Sperrung der Auslösetasten,
 Festhalten der gedrückten Taste.

Diese Wirkung bleibt solange bestehen, bis entweder vom Anwenderprogramm die gedrückte Auslösetaste durch den Warte- und Merkersetzbefehl abgefragt oder die C-Taste betätigt wurde.

- Läuft das Programm bei der Abarbeitung eines Befehles auf einen Fehlerstopp, so erfolgt

Sperrung der Zehnertastatur (außer C-Taste),
 Sperrung der Auslösetasten.









Bei einem Externfehlerstopp leuchtet die rote Lampe, bei einem Datenträgerfehlerstopp die orange Lampe, bei einem Internfehlerstopp die rote und die orange Lampe. Beim Internfehlerstopp ist keine Korrektur möglich, beim Externfehlerstopp wird nach Betätigung der C-Taste, beim Datenträgerfehlerstopp nach Betätigen der F-Taste der Befehl erneut ausgeführt.

- Beim Befehl "ACC Eingabe (0.1)" läuft das Programm auf Externfehlerstopp, wenn die Anzahl der eingetasteten Ziffern unzulässig ist. Nach Drücken der C-Taste ist die Eingabe gelöscht, und es kann erneut eingegeben werden. Die Betätigung einer beliebigen Auslösetaste bewirkt dann eine Wiederholung des Befehls "ACC Eingabe (0.1)".

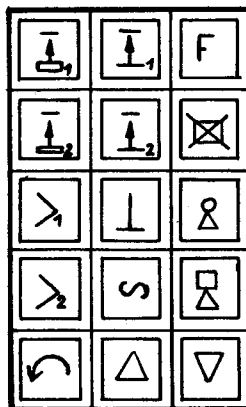
	Codierung der Tastatur	2.12
--	------------------------	------

AD											
j			l			m			r		
12	11	10	9	8	7	6	5	4	3	2	1

Funktionstasten:

- MLFM Taste  1 Zeilenschaltung
Magnetkontenschacht 1
(line feed magn.)
- MLFMB Taste  2 Zeilenschaltung
Magnetkontenschacht 2
(line feed magn. back)
- MCM Taste  1 Übertrag Magnet-
kontenschacht 1
(carry magn.)
- MCMB Taste  2 Übertrag Magnet-
kontenschacht 2
(carry magn. back)
- MLFLU Taste  1 Zeilenschaltung
Leporello 1
(line feed upper leporello)
- MLFLL Taste  2 Zeilenschaltung
Leporello 2
(line feed lower leporello)
- MNA Taste  Neues Konto
(new account)
- MSTO Taste  Storno

	Tastatur	2.12
--	----------	------

Wirkung:

- Die Funktionstasten haben die Codierungen 1.1 bis 1.4 und 1.6 bis 1.9.
Sie sind

in beliebiger Kombination einrastbar,
gegenseitig ausrastbar,
durch Programmbefehl ausrastbar,
frei programmierbar.

- Der Befehl "EFK Funktionstastenauswurf (2.12)" löst die Arretierung sämtlicher Funktionstasten, so daß alle zuvor eingerasteten Funktionstasten in die Ruhestellung zurückgehen.

Bemerkung:

Die rechts neben den Funktionstasten befindliche Tastenreihe enthält als oberste Taste die F-Taste. Die anschließenden Tasten sowie die beiden unterhalb der Funktionstasten gelegenen Tasten sind für den Betrieb als freiprogrammierbare Monitor-Maschine reserviert.

	Warte- und Merkersetzbehl	2.12
--	---------------------------	------

AD												
i	l				m				r			
12	11	10	9	8	7	6	5	4	3	2	1	

Codierung von Bedingungen:

Operation beendet bei Simultangerät 1:
MTW Drucker 1

Operation beendet bei Simultangerät 2:
{MTWS Drucker 2}
MPC Lochkartenstanzer 1
{MPCS Lochkartenstanzer 2}
MPT Lochstreifenstanzer 1
{MPTS Lochstreifenstanzer 2}
MPT Cassette {1-4} [1-2]

Karte da im

8	4	{MMAG Magnetkontenschacht 1 (vorn)} [Kontenschacht 1 der automatischen Einzugs- einrichtung (auch bei SKZA2)]
8	8	MMAGB Magnetkontenschacht 2 (hinten) [Kontenschacht 1 des automatischen Kontenein- zuges (auch bei SKZA2)]

Kein Papier bei

8	12	MPLU Vorschub 1 an Drucker 1
8	13	MPLL Vorschub 2 an Drucker 1
8	14	{MPLSU Vorschub 1 an Drucker 2}
8	15	{MPLSL Vorschub 2 an Drucker 2}

Ende des Lochstreifens bei

4	0	MFPT Lochstreifenstanzer 1
4	1	{MFPTS Lochstreifenstanzer 2}

Stanzfehler bei

4	2	MWPT Lochstreifenstanzer 1
4	3	{MWPTS Lochstreifenstanzer 2}

Streifenkarte nicht in Grundstellung

4	4	MHPT Lochstreifenstanzer 1
4	5	{MHPTS Lochstreifenstanzer 2}

	Warte- und Merkersetzbefehl	2.12
--	-----------------------------	------

Wirkung:

1. Die Ausgabe auf dem Drucker, dem Lochkartenstanzer und dem Lochstreifenstanzer erfolgt simultan zum weiteren Befehlsablauf. Die Bedingungen 8.0 bzw. 8.1 sind genau dann erfüllt, wenn auf dem jeweiligen Simultangerät keine Ausgabe erfolgt.
2. Um zu erreichen, daß eine Magnetkontokarte aus dem Schacht genommen wird, kann mit MMAG bzw. MMAGB abgefragt werden, ob sich die Karte noch im Schacht befindet. Beim Einzug einer Karte ist diese Abfrage nicht erforderlich, da sie automatisch intern erfolgt.
3. Die Bedingung MPLU bzw. MPLL bzw. MPLSU bzw. MPLSL ist erfüllt, wenn bei dem angegebenen Vorschub kein Papier mehr vorhanden ist.
4. Für jeden Lochstreifenstanzer gibt es die Meldung MFPT bzw. MFPTS, die das Ende des Lochstreifens anzeigt. Es steht dann nur noch eine bestimmte (einstellbare) Länge des Streifens zur Verfügung.
5. Lochstreifenstanzer können auch für die Verarbeitung von Lochstreifenkarten benutzt werden. Hierzu ist die Bedingung MHPT bzw. MHPTS solange erfüllt, wie sich eine Karte nicht in Grundstellung befindet. Die Grundstellung ist dadurch gegeben, daß sich das Führungsloch der Karte an der Abfühlstation des Lochstreifenstanzers befindet.
6. Beim Modell 820/15 dürfen der 2. Drucker, der 2. Lochstreifenstanzer und der 2. Lochkartenstanzer nicht angesprochen werden. Es kann nur ein Magnetkontenschacht angeschlossen werden. Dieser ist als Schacht 2 zu programmieren.
7. Beim Modell 820/25 oder 820/35 kann die automatische Konteneinzugsrichtung durch den Befehl 2.12 "MMAG" oder "MMAGB" angewählt werden (bei SKZA2).
8. Die Befehle "Operation beendet bei Simultangerät 2", "Karte im Schacht da", "Kein Papier bei", "Ende des Lochstreifens", "Stanzfehler und Lochstreifenkarte nicht in Grundstellung", sind beim Modell 820/15 nur bei Verwendung des Moduls GP möglich.

	Warte- und Merkersetzbefehl	2.12
--	-----------------------------	------

AD												
i	l	m						r				
12	11	10	9	8	7	6	5	4	3	2	1	
0												
						10				0		
						10				1		

SMWT

Setze Merker ML, MU, MC nach Beendigung der Ausgabe und Eingabe

,MTW

Simultangerät 1:
Drucker 1

{,MTWS

Simultangerät 2:
Drucker 2 oder

,MPC

Lochkartenstanzer 1 oder

{,MPCS

Lochkartenstanzer 2 oder

,MPT

Lochstreifenstanzer 1 oder Cassette

{,MPTS

{1-4} [1-2]
Lochstreifenstanzer 2}

Wirkung:

1. Die Ausgabe auf den aufgeführten externen Geräten erfolgt simultan zum weiteren Befehlsablauf. Die dazugehörigen Ausgabebefehle, die durch Stellung der Merker Kontrollinformationen liefern, bewirken selbst noch keine Veränderung der Merker ML, MU, MC. Deren Stellung ist nach Beendigung der Ausgabe intern zwischengespeichert.
2. Im Befehl "SMWT Setze Merker nach Beendigung Simultan-Ausgabe (2.12)" wird auf das Ende der Ausgabe des angegebenen Simultangerätes gewartet. Im Anschluß daran erfolgt das Setzen der Merker entsprechend der zuletzt gespeicherten Stellung.
3. Beim Modell 820/15 kann der Befehl "Setze Merker nach Beendigung der Ausgabe und Eingabe auf Simultangerät 2" nur in Verbindung mit dem Modul GP verwendet werden. Es ist weiterhin darauf zu achten, daß das Modell 820/15 keine Zweitgeräte mit Ausnahme der Magnetband-Cassette zuläßt. Da jedoch der gleiche Code "2.12 0.10.1" wie bei Erstgeräten gilt, wird kein Fehler gemeldet.

	Warte- und Merkersehbefehl	2.12
--	----------------------------	------

Bemerkungen:

1. Es gibt zwei externe Geräte, die zueinander und zu internen Operationen simultan arbeiten können. Das erste Simultangerät ist Drucker 1, das zweite Simultangerät kann wahlweise {Drucker 2}, Lochkartenstanzer 1, {Lochkartenstanzer 2}, Lochstreifenstanzer 1, {Lochstreifenstanzer 2} oder die Magnetbandcassette ~~{1-4}~~ ~~[1-2]~~ sein.
2. Die zwischengespeicherte Merkerstellung bleibt solange unverändert, bis auf demselben Simultangerät eine erneute Ausgabe erfolgt.
3. Durch den Befehl "SMWT Setze Merker nach Beendigung Simultan-Ausgabe (2.12)" und die Tatsache, daß die simultan ablaufenden Ausgabe-Befehle die Stellung der Merker nicht verändern, besteht die Möglichkeit der weitgehenden Ausnutzung der Simultanarbeit.

WT	Setze Lösche	Anfang des Befehlsbereichs	2.12
----	-----------------	----------------------------	------

AD											
i	l	m				r					
12	11	10	9	8	7	6	5	4	3	2	1
	0					4					11
	0					4					12

WT, CEP Freigabe des Befehlsbereichs als Datenfeld

WT, SEP Setzen des Anfang des Befehlsbereichs
(Folgebefehl von EP 3.9)

Wirkung:

1. Diese Befehle sind nur bei dem Modell 820/15 anwendbar. Sie legen eine Grenze zwischen Datenbereich und Befehlsbereich. Da bei diesem Modell nur der Einsatz eines Kernspeichers möglich ist, müssen Befehle und Daten gemeinsam untergebracht werden.

Das Betriebsprogramm MFA 1 ermöglicht eine Befehls-Speicherung im Kernspeicher Typ 160 bis 165, wogegen das Betriebsprogramm MFAGS 1 nur eine Befehls-Speicherung im Kernspeicher 166 und 164 zuläßt.

2. Der Befehl WT, SEP 2.12 0.4.12.
Diese Befehlskombination ist nur mit dem Befehl "EP 3.9" als Hauptbefehl wirksam. Der Befehl WT, SEP ist ein Folgebefehl, der unmittelbar folgen muß.

Im Hauptbefehl EP wird das letzte Speicherwort des Datenbereiches angegeben. Im folgenden Befehl (WT, SEP) wird nur die Aussage wiederholt, eine Grenze zu setzen. Dies ist notwendig, da der Befehl 3.9 EP ohne Folgebefehl noch eine andere Bedeutung hat.

Diese Grenze wird vom Betriebsprogramm solange erhalten, bis eine neue Grenze gesetzt oder die Grenze aufgehoben wird.

Erst nach den Befehlen "EP" und "WT, SEP" kann ein Sprung in den Befehlsbereich des Kernspeichers erfolgen.

Wird die gesetzte Grenze vom Datenbereich überschritten, so erfolgt ein interner Fehlerstopp. Dadurch wird vermieden, daß Befehls Worte gelöscht oder überschrieben werden.

Beim Kernspeichertyp 160 bis 165 liegt der Datenbereich nur im vorderen Teil, und der Befehlsbereich im hinteren Teil des Kernspeichers. Es ist zu be-

WT	Setze Lösche	Anfang des Befehlsbereichs	2.12
----	-----------------	----------------------------	------

achten, daß ein Befehl 5 Kernspeicherstellen einnimmt. Ab der vorgegebenen Grenze werden die Befehls Worte mit 0.0.0 beginnend fortlaufend vom Betriebsprogramm nummeriert.

Bei den Kernspeichertypen 164 und 166 wird der Befehlsbereich vom Datenbereich eingeschlossen. Die Operationsteile liegen in 6-Bit tiefen Zellen und breiten sich ab der externen Adresse 123 von rechts nach links aus, während die Operandenteile in 12-Bit tiefen Zellen liegen und sich ab der externen Adresse 123 von links nach rechts ausbreiten.

WT	Setze Lösche	Anfang des Befehlsbereichs	2.12
----	-----------------	----------------------------	------

Im Betriebsprogramm MFA GST belegen 16 Befehls worte 4 Speicher worte.

Wobei die Speicher worte sich wie folgt aufteilen:

- 1 Speicher wort der unteren Klasse also 123 und kleiner.
- 3 Speicher worte der oberen Klasse also 507 und kleiner.

Die Befehle 2.12 0.4.11 und 2.12 0.4.12 können nur in Verbindung mit dem Modul GP verwendet werden. Die Definition kann nur im Fest Speicher erfolgen.

Beispiel

Es werden von einem Kern Speicher-Typ 161 (= 128 Speicher worte), 50 Speicher worte für den Datenbereich und die restlichen Felder für freie Befehle verwendet.

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
	0 3 9	0	0 3 1		EP, 49
	1 2 12	0	0 4 12		WT, SEP
	2				

Das letzte Speicher wort im Datenbereich ist also Speicher wort 49. Das erste Speicher wort im Befehlsbereich ist folglich Speicher wort 50. In Speicher wort 50, Kern Speicher stelle 0 bis 4 befindet sich also der erste Befehl mit der Adresse 0.0.0.

WT	Setze Lösche Anfang des Befehlsbereichs	2.12
----	--	------

Der Befehl MVH,70.5 "Transportiere (SW 5) nach SW 70 " würde einen internen Fehlerstopp in diesem Beispiel erzeugen.

2. Der Befehl "WT, CEP = 2.12. 0.4.11"

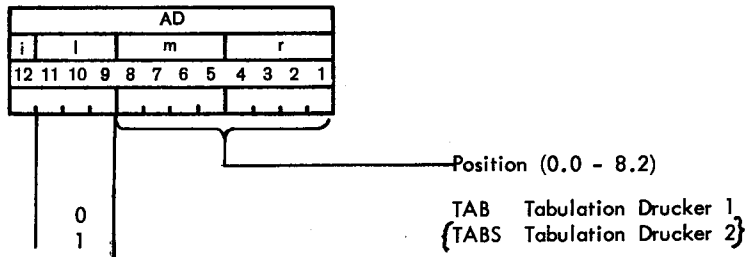
2.1. Wirkung:

Diese Befehlskombination dient zur Aufhebung der gesetzten Grenzen zwischen Datenbereich und Befehlsbereich. Es ist kein Vorbefehl erforderlich. Nach diesem Befehl können wieder alle verfügbaren Speicherworte als Speicherwort angesprochen werden.

2.2. Bemerkung:

Der Befehl wird in den meisten Fällen vor einer Laderoutine im neuen Befehlsbereich verwendet.

TAB	Tabulation Drucker	2.13
-----	--------------------	------



Wirkung:

Der Kugelkopf des (jeweiligen) Druckers tabuliert nach der in AD_{mr} angegebenen Position.

Beispiel:

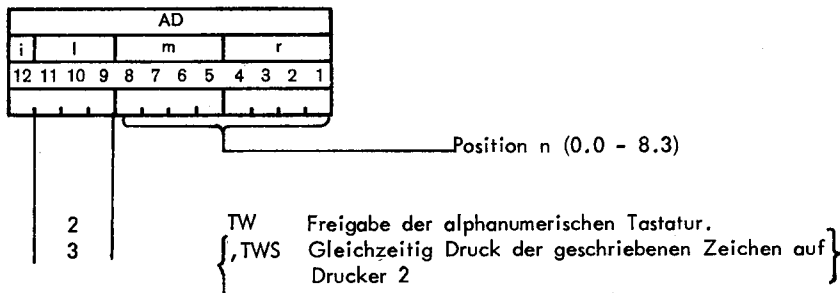
Tabulation Drucker 1 nach Position 67.

BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
	0 2 13	0	0 4 3		TAB, 67

Bemerkungen:

1. Der Befehl "TAB (bzw. TABS) Tabulation (2.13)" wird simultan zum weiteren Befehlsablauf ausgeführt. Vor einer neuen Ausgabe auf dem benutzten Simultangerät wartet intern das Betriebsprogramm auf die Beendigung der Operation.
2. In Position 0 ist die erste Schreibstelle links, in Position 130 die letzte Schreibstelle rechts.
3. Bei einer Position größer als 8.2 kann die Tabulation nicht durchgeführt werden, d.h. der Tabulationsvorgang kommt zu keinem Abschluß.
4. Es ist unbedingt zu vermeiden, beim Modell 820/15 den zweiten Drucker anzusprechen, da das Betriebsprogramm diesen Fehler nicht erkennt und falsche Interpretationen auftreten können.

TW	Freigabe der alphanumerischen Tastatur (type writer)	2.13
----	---	------

Wirkung:

- Drucker 1 ist bis Position $n - 1$ zum Schreiben freigegeben. Ist die Position des Schreibkopfes größer oder gleich Position n , so erfolgt keine Freigabe der Schreibmaschine. Ab Position n können nur noch die Tabulationstaste \Rightarrow , die Wagenaufzugtaste \Leftarrow oder evtl. eine der Auslösetasten zur Beendigung der Befehlsausführung betätigt werden.
- Der AD-Teil vom Befehl unter Adresse 0.0.0 im Anwenderprogrammblock 0 gibt an, ob die Rücktaste benutzt werden kann. Bei jeder Betätigung der Rücktaste geht der Schreibkopf um eine Schreibstelle zurück. Ein Zurückstellen vor die Ausgangsposition der Schreibmaschinenfreigabe ist jedoch nicht möglich. Wenn die Rücktaste erlaubt ist, kann sie auch noch in Position n benutzt werden, um das in Position $n-1$ geschriebene Zeichen zu korrigieren.
- Die Freigabe der alphanumerischen Tastatur wird durch eine der folgenden Tasten beendet, wobei gleichzeitig die Merker gesetzt werden. (Die Beendigung durch eine der Auslösetasten ist nur dann möglich, wenn dies durch entsprechende Belegung des AD-Teils vom Befehl unter Adresse 0.0.0 im Anwenderprogrammblock 0 zugelassen wurde.)

Taste	ML	MU	MC
Tabulationstaste	0	1	0
Wagenaufzugtaste	0	0	1
Auslösetasten (Codes 2.1 - 2.14)	1	0	0

Die Beendigung durch Tabulationstaste oder Auslösetaste bewirkt keine Bewegung des Kugelkopfes. Nur bei Betätigen der Wagenaufzugtaste findet eine Tabulation in die Ausgangsposition statt (jedoch keine Zeilenschaltung).

TW	Freigabe der alphanumerischen Tastatur (type writer)	2.13
----	---	------

4. Die geschriebenen Zeichen können gleichzeitig auf dem Drucker 2 mitgedruckt werden, wobei deren Ausgangsposition eine andere als die von Drucker 1 sein kann. Die Betätigung der Rücktaste wirkt sich analog auf dem Drucker 2 aus, und die Beendigung durch Wagenaufzugstaste hat ebenfalls eine Tabulation des Schreibkopfes in die Ausgangsposition von Drucker 2 zur Folge.

Beispiel: für Modell 820/25 und 820/35:

Der Kugelkopf vom Serialdrucker 1 befindet sich auf der Position 21, der vom Serialdrucker 2 auf der Position 67. Es ist folgender Befehl gegeben worden:

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
	0 2 13	0	3 1 11		TW, TWS, 27
	1				

Danach ist Drucker 1 zum Schreiben freigegeben. Es wird von Hand geschrieben

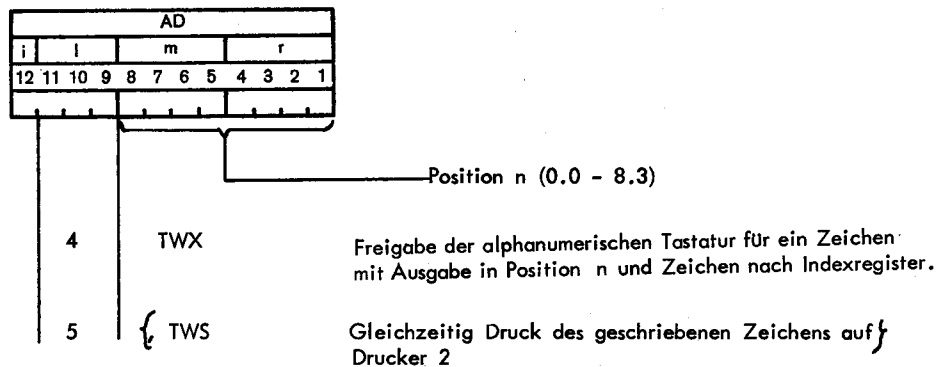
ANLAGE

was gleichzeitig auch auf dem Drucker 2 ausgedruckt wird. Anschließend ist der Kugelkopf vom Drucker 1 in Position 27, der Kugelkopf vom Drucker 2 in Position 73. Es kann kein Zeichen mehr geschrieben werden. Nach Drücken der Wagenaufzugtaste tabulieren die Kugelköpfe in die Positionen 21 bzw. 67, und es werden die Merker ML = 0, MU = 0 und MC = 1 gesetzt.

Bemerkungen:

1. In Position 0 ist die erste Schreibstelle links, in Position 130 die letzte Schreibstelle rechts. Danach geschriebene Zeichen werden alle in Position 130 ausgedruckt.
2. Man beachte, daß das Programm erst dann mit dem nächsten Befehl fortfährt, wenn die Freigabe der alphanumerischen Tastatur beendet wurde.
3. Beim Befehl "TW Freigabe alphanumerischer Tastatur (2.13)" findet keine Simultanarbeit statt. Nach Beendigung der Befehlsausführung sind die Merker entsprechend obiger Tabelle gesetzt, ohne daß zuvor der Befehl "SMWT Setze Merker nach Beendigung Simultan-Ausgabe (2.12)" gegeben werden muß.
4. Falls der Zusatz " TWS " mit gleichzeitigem Druck auf dem Drucker 2 bei Modell 820/15 gegeben worden ist, so wird dieser Zusatz als nicht vorhanden angesehen. Das Programm läuft ungestört weiter.

TWX	Ein Zeichen ausdrucken und nach Indexregister (type writer sign to indexregister)	2.13
-----	---	------

Wirkung:

1. Drucker 1 ist zum Schreiben von einem Zeichen freigegeben, wenn sich der Kugelkopf in einer Position befindet, die kleiner als die in AD angegebene Position n ist. Ist die Position des Kugelkopfes größer oder gleich n, kann kein Zeichen geschrieben werden, es sind lediglich die Wagenaufzugtaste \leftarrow , die Tabulationstaste \Rightarrow und evtl. eine der Auslösetasten erlaubt.
2. Die Freigabe der alphanumerischen Tastatur ist nach Betätigung einer Taste beendet. Bei entsprechender Belegung des AD-Teils vom Befehl unter Adresse 0.0.0 im Anwenderprogrammblock 0 kann dies auch durch Betätigen einer der Auslösetasten an der Zehnertastatur bewirkt werden.
3. Ein eingetastetes Zeichen wird vom Drucker 1 ausgegeben. Anschließend rückt der Kugelkopf nach rechts auf die nächste Schreibposition. Es sind alle Zeichen, Tasten sowie die Leertaste zulässig, und es kann zum Schreiben von Sonderzeichen zusätzlich die Umschalttaste bedient werden. Der AD-Teil vom Befehl unter Adresse 0.0.0 im Anwenderprogrammblock 0 gibt an, ob die Rücktaste benutzt werden kann. Ihre Betätigung hat jedoch keine Bewegung des Kugelkopfes zur Folge.
4. Das geschriebene Zeichen kann gleichzeitig auf dem Drucker 2 ausgegeben werden, wobei die Position des Kugelkopfes vom Drucker 2 eine andere als die des Kugelkopfes vom Drucker 1 sein kann.


TWX	Ein Zeichen ausdrucken und nach Indexregister (type writer sign to indexregister)	2.13
-----	---	------

5. Das Indexregister und die Merker erhalten folgende Werte:

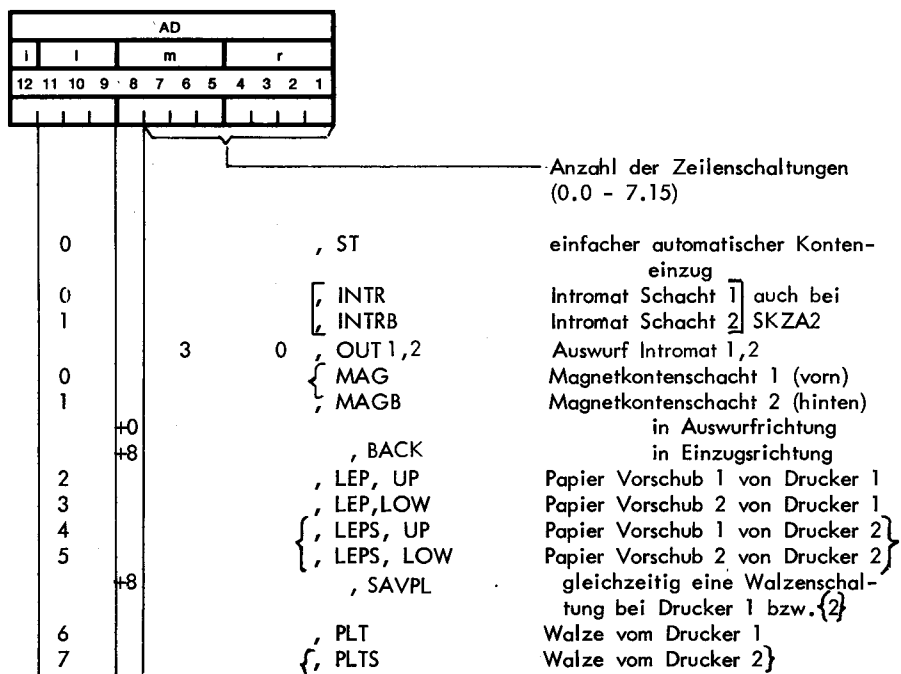
Betätigte Taste	I	ML	MU	MC
Zeichen-Taste	ALC-Code	0	0	0
Rücktaste	5.12	0	0	0
Tabulationstaste	6.8	0	1	0
Wagenaufzugtaste	6.12	0	0	1
Auslösetasten (Codes 2.1 - 2.14)	6.11	1	0	0

Nach dem Drücken einer Zeichentaste und dem Ausdruck des Zeichens rückt der Kugelkopf auf die nächste Schreibstelle weiter. Bei allen anderen Tasten findet keine Bewegung des Kugelkopfes statt.

Bemerkungen:

1. In Position 0 ist die erste Schreibstelle links, in Position 130 die letzte Schreibstelle rechts.
2. Man beachte, daß das Programm erst dann mit dem nächsten Befehl fortführt, wenn die Freigabe der Tastatur beendet wurde.
3. Beim Befehl "TWX, Ein Zeichen nach I (2.13)" findet keine Simultanarbeit statt. Nach Beendigung des Befehls sind die Merker entsprechend der obigen Tabelle gesetzt, ohne daß zuvor der Befehl "SMWT Setze Merker nach Beendigung Simultan-Ausgabe (2.12)" gegeben werden muß.
4. Man beachte, daß beim Schreiben des Zeichens "-" auf dem Drucker der ALC-Code 0.11 in das Indexregister gebracht wird.
5. Die Taste Zeilenschaltung  am Serialdrucker ist stillgelegt. Ihre Betätigung bringt keinen Code in das Indexregister und beendet auch nicht die Freigabe der Schreibmaschine bzw. die Ausführung des Befehls.
6. Dieser Befehl kann beim Modell 820/15 nicht angewendet werden, da er im Befehlsvorrat nicht enthalten ist.

LF	Zeilenschaltung (line feed)	2.14
----	-----------------------------	------



Wirkung:

1. Das jeweilige Formular wird um die in AD_{mr} Bit 1 bis Bit 7 angegebene Anzahl von Zeilen weitertransportiert. Bei Magnetkontokarten ist auch ein Rückwärtstransport (Transport in Einzugsrichtung) möglich.
 2. Um die Schreibmaschinenwalze zu schonen, ist es erforderlich, bei einigen Leporello-Zeilenschaltungen zusätzlich eine Weiterschaltung der Druckerwalze vorzunehmen.
- Dies ist bei Modell 820/15 nicht möglich.
3. Bei Modell 820/15 ist nur ein Magnetkontenschacht anschließbar. Er wird von der Programmierung als Schacht 2 angesprochen.

LF	Zeilenschaltung (line feed)	2.14
----	-----------------------------	------

4. Die Zeilenzähler sind nach Beendigung des Befehls auf dem neuen Stand. Beim Betriebssystem SKZA2 wird der Zeilenzähler der Walze bei Walzenschaltung nicht erhöht. Wird in Magnetkontenschächten über die 0-Stellung hinaus zurückgeschaltet, so steht das Komplement der überzähligen Anzahl von Zeilenschaltungen zu $2^{11} - 2048$ im jeweiligen Zeilenzähler.

Beispiel:

Das Formular in der Papiervorschubeinrichtung 2 vom Drucker 1 soll um 3 Zeilen weitertransportiert werden. Gleichzeitig ist die Walze um eine Zeile weiterzuschalten.

vorher:

0	1	9
---	---	---

 Zeilenzähler Leporello 2 von Drucker 1

BW-Adresse	Op. Teil	1	Adreßteil	Symb. Adr.	Bemerkungen				
	0	2	14	0	3	8	3		LF, LEP, LOW, SAVPL, 3

nachher:

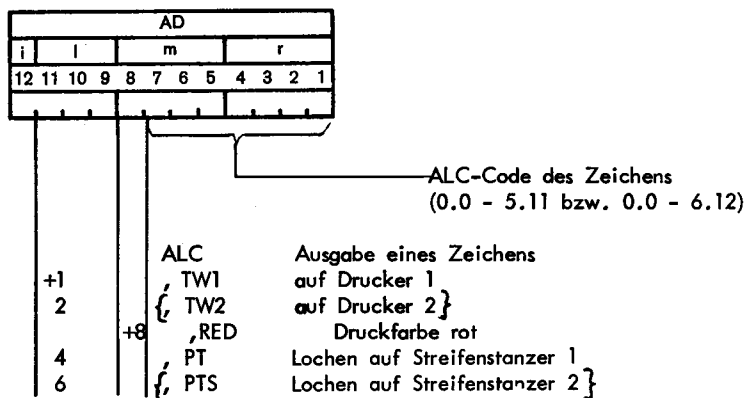
0	1	12
---	---	----

 Zeilenzähler Leporello 2 von Drucker 1

Bemerkungen:

- Im Anwenderprogramm unmittelbar aufeinanderfolgende Zeilenschaltungs- und Formularpositionierungsbefehle werden gleichzeitig ausgeführt.
- Der Befehl "LF Zeilenschaltung (2.14)" ist nicht zum Einzug von Karten im Magnetkontenschacht zu verwenden. Auch darf er nicht innerhalb der Gültigkeit eines Vorbefehls "BL Blockumschaltung (2.11)" gegeben werden. Andernfalls ist dieser gelöscht.
- Bei Modell 820/15 ist kein zweiter Drucker anschließbar. Wird jedoch eine Zeilenschaltung für dieses Gerät codiert, so wird diese auf dem ersten Gerät ausgeführt.
- Erfolgen mehrere Ausgabebefehle auf dem gleichen Formularträger hintereinander, so wird nur der erste Befehl akzeptiert.
- Es ist bei folgenden Formularträgern darauf zu achten, daß diese nur mit den entsprechenden Moduln arbeiten.
einfacher automatischer Konteneinzug Modul ST
doppelter automatischer Konteneinzug Modul IN
einfacher Magnetkontenschacht Modul MC

ALC	Ausgabe eines Zeichens (alphanumeric character)	2.15
-----	--	------



Wirkung:

1. Ausgabe auf dem Drucker: Das Zeichen, dessen ALC-Code in AD Bit 1 bis Bit 7 angegeben ist, wird an der gegenwärtigen Kugelkopf^{mr}position ausgedruckt. Danach rückt der Kugelkopf nach rechts auf die nächste Schreibstelle.
2. Ohne Druckvorbefehl erfolgt die Ausgabe genau eines Zeichens. Die Druckfarbe rot ist entweder im Hauptbefehl oder in einem entsprechenden Druckvorbefehl anzugeben (nicht in beiden). Ferner ist es durch einen Druckvorbefehl möglich, eine n-malige Wiederholung der Ausgabe des Zeichens zu erreichen ($n \hat{=} 0$ bis 15).
3. Der Zusatz "TW2" darf nicht mit angegeben werden. Wird "TW1, TW2" codiert, so wird TW2 in diesem Falle als nicht programmiert angesehen.
4. Ausgabe auf Lochstreifenstanzer: Der in AD Bit 1 bis Bit 7 angegebene ALC-Code des Zeichens wird in den Lochstreifen^{mr}code umgewandelt und auf dem angegebenen Lochstreifenstanzer ausgegeben. Danach wird der Lochstreifen auf die nächste Position transportiert.
5. Ohne Lochstreifen^{mr}vorbefehl erfolgt die Umcodierung nach der ersten Lochstreifen^{mr}codetabelle, und es wird genau ein Zeichen ausgegeben. Durch einen entsprechenden Lochstreifen^{mr}vorbefehl kann eine Umcodierung nach der zweiten Lochstreifen^{mr}codetabelle oder eine n-malige Wiederholung der Ausgabe des Zeichens erreicht werden ($n \hat{=} 0$ bis 15).
6. Der Zusatz "PT" beim Modell 820/15 darf nur in Verbindung mit dem Modul "PT" angewendet werden.

ALC	Ausgabe eines Zeichens (alphanumeric character)	2.15
-----	---	------

7. Da kein zweiter Lochstreifenstanzer anschließbar ist, darf in AD₁ auf keinen Fall 6 oder 7 codiert werden.

ALC	Ausgabe eines Zeichens (alphanumeric character)	2.15
-----	--	------

Beispiel:

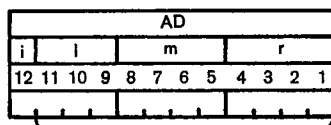
Das Unterstreichungszeichen "-" soll 10-mal hintereinander auf Drucker 1 und 2 ausgegeben werden, wobei für Drucker 2 die Druckfarbe rot zu wählen ist.

BW-Adresse		Op. Teil			Adreßteil				Symb. Adr.	Bemerkungen
	0	3	6	0	0	10	9		EDFS, RED, 9	
	1	3	6	0	0	0	9		EDF, 9	
	2	2	15	0	3	3	5		ALC, TW1, TW2, YULN	
	3									

Bemerkungen:

- Bei den ALC-Codes 0.11 und 0.13 wird auf dem Drucker das Zeichen "-", bei 2.14 und 4.0 das Zeichen "-" und bei 3.15 ein Leerschritt ausgegeben. Die Lochung auf den Streifenlochern ist abhängig vom jeweiligen Lochstreifencode.
- Für die Ausgabe auf den Serialdrucker dürfen nur Zeichen codiert werden, die in der Schreibmaschinentabelle vorhanden sind, d.h. Zeichen mit einem Code von 0.0 bis 5.11 (auch 5.12 ist nicht zulässig). Andernfalls kann ein undefinierter Ausdruck erfolgen. Für die Ausgabe auf den Streifenlochern sind auch Zeichen mit einem Code von 5.12 bis 7.15 zulässig. Bei Codes über 6.12 ist jedoch zu beachten, daß diese später nach dem Einlesen nicht automatisch decodiert werden können.
- Die zur Umcodierung des ALC-Codes in den Lochstreifencode erforderlichen Codetabellen müssen sich im Anwenderprogrammblock 0 befinden. Die Anfangsadresse von Tabelle 1 ist im AD-Teil des Befehls unter Adresse 0.0.3, die von Tabelle 2 im AD-Teil des Befehls unter Adresse 0.0.4 im Anwenderprogrammblock 0 anzugeben (vgl. Beschreibung "Programm, Aufbau", Kapitel 2.2 und 4.2).
- Ist AD₁ = 0, so erfolgt keine Ausgabe. Der Befehl wirkt dann wie ein Leerbefehl.

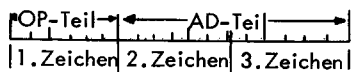
TT	Drucken Tabelle (type table)	3.0
----	------------------------------	-----



Befehlsadresse (0.0.0 - 7.15.15)

Wirkung:

- Der Befehl unter der in AD_{lmr} stehenden Adresse ist ein 18-Bit-Wort. Werden von links nach rechts je 6 Bit zusammengefaßt, so entstehen drei 6-Bit-Worte,



denen nach dem ALC-Code drei Zeichen entsprechen.

- Von der in AD_{lmr} angegebenen Adresse ab werden nach dem unter Punkt 1 beschriebenen Verfahren die Zeichen fortlaufend von links nach rechts aufgesucht und ausgedruckt (ohne Druckvorbefehl auf Drucker 1 in schwarz). Nach der Ausgabe der drei Zeichen einer Adresse erfolgt analog die Ausgabe der Zeichen im Befehlsword unter der nächsten Adresse. Die Operation wird solange fortgesetzt, bis das Endezeichen 3.15 auftritt. Diesem Zeichen entspricht keine Ausgabe, der Druckvorgang ist beendet.
- Das auf das Zeichen 3.15 folgende Zeichen wird ebenfalls aufgesucht. In Abhängigkeit von seinen letzten 4 Bit werden die folgenden Merkerstellungen intern zwischengespeichert:

		ML	MU	MC
8	YTAB	0	1	0
12	YCAR	0	0	1
11, sonst	YTRK	1	0	0

Im Befehl "SMWT Setze Merker nach Ende Simultan-Ausgabe (2.12)" wird auf das Ende des Ausdrucks gewartet. Im Anschluß daran erfolgt das Setzen der Merker entsprechend obiger Tabelle.

TT	Drucken Tabelle (type table)	3.0
----	------------------------------	-----

- Haben die letzten 4 Bit des Zeichens nach 3.15 den Wert 12, so tabuliert der Kugelkopf nach Beendigung des Drucks auf die Ausgangsposition zurück. In allen anderen Fällen bleibt der Kugelkopf nach dem zuletzt geschriebenen Zeichen stehen.
- Das Programm fährt mit dem auf den Befehl "TT Drucken Tabelle (3.0)" folgenden Befehl fort. (Eine Ausnahme hiervon bilden nur die Fälle, in denen der Befehl 3.0 durch Substitution entstanden ist. Man vergleiche hierzu die Befehle "SST Substitution(1.13)" und "SSTX Substitution mit Indexregistern (2.2)".)

Beispiel:

Es soll 'KONTO!' von Position 23 ab auf dem Drucker 1: in schwarz ausgedruckt werden und der Kugelkopf nach dem Zeichen ":" stehenbleiben. Dazu wird nach dem Endezeichen das Zeichen 0.11 gespeichert.

BW-Adresse			Op. Teil			Adreßteil			Symb. Adr.	Bemerkungen
4	2	10	2	13	0	0	1	7		TAB,23
		11	3	0	0	7	13	4		TT,TEXT
		12								
		13								

Die Tabelle steht ab Adresse 7.13.4 im gleichen Block:

BW-Adresse			Op. Teil			Adreßteil			Symb. Adr.	Bemerkungen
7	13	4	1	12	1	0	1	15	TEXT	KONTO/YCOL/&
		5	2	5	1	0	3	0		/YECC//YTRK/
		6	3	15	0	2	12	0		
		7								

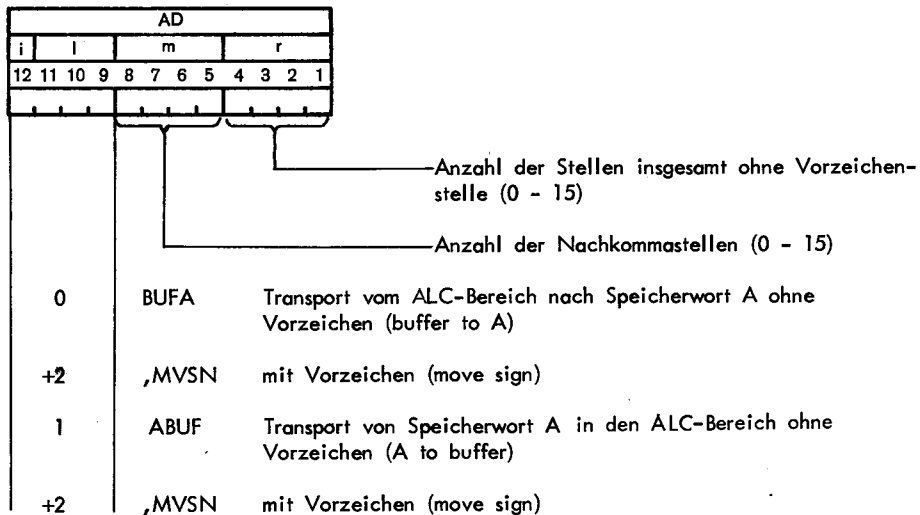
Der Kugelkopf vom Drucker 1 bleibt anschließend in Position 29 stehen. Die Merkerstellung ML = 1, MU = 0, MC = 0 ist intern zwischengespeichert. Die Merker selbst sind jedoch unverändert. Das Programm fährt mit dem Befehl unter Adresse 4.2.12 fort.

TT	Drucken Tabelle (type table)	3.0
----	------------------------------	-----

Bemerkungen:

1. Man beachte, daß in jedem Fall nach dem Zeichen 3.15 die darauffolgende 6-Bit-Kombination untersucht wird, um die Merkerstellung dementsprechend zwischenspeichern und eventuell eine Tabulation durchzuführen. Es muß also immer eine solche Kombination vorhanden sein, d.h. das Endezeichen 3.15 sollte z.B. nicht in Bit 1 bis Bit 6 des letzten Befehls eines Textblockes stehen, da dann die folgenden 6 Bit des nächsten Befehlswortes mit interpretiert werden.
2. Der Befehl "TT Drucken Tabelle (3.0)" wird simultan zum weiteren Befehlsablauf ausgeführt. Vor einer neuen Ausgabe auf dem benutzten Simultangerät wartet intern das Betriebsprogramm auf die Beendigung der Operation.
3. Ist für jeden Drucker ein Druckvorbefehl gegeben worden, so wird nur auf Drucker 2 gedruckt und anschließend der dazugehörige Druckvorbefehl aufgehoben. Damit auch auf Drucker 1 gedruckt wird, ist ein zweiter Druckbefehl zu geben, der jedoch simultan zum ersten ausgeführt wird.
4. Man beachte den Befehl "BL Blockumschaltung (2.11)".

	ALC-Bereich und Speicherwort A	3.1
--	--------------------------------	-----

Wirkung:

- Es findet ein Transport von dezimalen Stelleninhalten zwischen dem ALC-Bereich und Speicherwort A statt. Die Anfangsstelle im ALC-Bereich legt der Inhalt des dazugehörigen Bereichszeigers I_3 fest, der nach Ausführung des Befehls um die Anzahl der transportierten Stellen (einschließlich Vorzeichen) erhöht ist und folglich im Bereich auf die folgende der zuletzt transportierten Stelle zeigt.
- Beim Transport ohne Vorzeichen aus dem ALC-Bereich nach Speicherwort A erhält das Speicherwort ein positives Vorzeichen, d.h. in die Vorzeichenstelle kommt die Ziffer 0.
- Beim Transport mit Vorzeichen ist folgendes zu beachten:

Bei der Übertragung vom Bereich nach Speicherwort A kommt der Inhalt der zuerst transportierten Stelle in die Vorzeichenstelle von A, ohne verändert zu werden.

Bei der Übertragung von Speicherwort A in den Bereich kommt zuerst der Inhalt der Vorzeichenstelle von A in die durch den Bereichszeiger I_3 festgelegte Stelle im Bereich, ohne verändert zu werden. Daran anschließend erfolgt der Transport der weiteren Stelleninhalte.

	ALC-Bereich und Speicherwort A	3.1
--	--------------------------------	-----

Beispiel 1: (KA 5)

Transport von 4 Stellen aus dem ALC-Bereich nach Speicherwort A, ohne Vorzeichen und mit 1 Nachkommastelle.

vorher: I_3 1 7 1 1
 {SW} 18 | | | | | | | | 9 | 10 | 7 | 1 1 | 2 | 1 4 | 3
 [SW] 19 | | | | | | | | | | | | | | |

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
0	3 1 0	0	0 1 4		BUFA, 1.4
1					

nachher: I_3 1 7 1 5
 SW 3 0 0 0 0 0 0 0 0 7 1 1 2 1 4 0 0 0 0

Der Inhalt von Speicherwort A wird positiv, da ohne Vorzeichen übertragen wurde.

Beispiel 2: (KA 6)

Transport von 7 Stellen aus Speicherwort A in den ALC-Bereich, mit Vorzeichen und 3 Nachkommastellen.

vorher: I_3 2 6 1 2
 SW 3 4 0 0 0 8 9 1 5 7 1 1 4 2 5 1 1 3 0 0

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
0	3 1 0	3 3 7			ABUF, MVS, 3.7
1					

nachher: I_3 2 7 4
 [SW 34] {SW 33} | | | | | | | | | | | | | | 4 1 5 7 1
 [SW 35] {SW 34} 1 4 2 5 1 1 | | | | | | | | |

Der Inhalt der Vorzeichenstelle von Speicherwort A wird unverändert als erste Sedezimalzahl in den ALC-Bereich gebracht.

	ALC-Bereich und Speicherwort A	3.1
--	--------------------------------	-----

Bemerkungen:

1. Der ALC-Bereich dient zur Speicherung alphanumerischer Zeichen und als Datenbereich für Magnetkontokarten. Der Transport zwischen ALC-Bereich und Speicherwort A wird deshalb vorwiegend für die Verarbeitung von dezimal dargestellten Daten im Zusammenhang mit Magnetkontokarten Verwendung finden.
2. Man beachte, daß der Bereichszeiger I_3 in I_{1m} immer die interne Speicherwortnummer enthält, die aus der normalen Speicherwortnummer durch Addition von $\{0.5\}$ bzw. $[0.4]$ entsteht.
3. Bei Modell 820/15 enthält der Bereichszeiger in I_3 in I_{1m} die interne Speicherwortnummer, die aus der normalen Speicherwortnummer durch Addition von 0.4 entstanden ist.

	ALC-Bereich und Indexregister	3.1
--	-------------------------------	-----

AD												
i				m				r				
12	11	10	9	8	7	6	5	4	3	2	1	
4				0								
								0				
								1				
4				0								
								2				
								3				

Transport von ALC-Bereich nach Indexregister (buffer to indexregister)
 eine Sedezimalzahl nach I_r ($I_m = 0$)
 zwei Sedezimalzahlen nach I_m und I_r ($I_l = 0$)
 Transport vom Indexregister nach ALC-Bereich (indexregister to buffer)
 eine Sedezimalzahl aus I_r
 zwei Sedezimalzahlen aus I_m und I_r

Wirkung:

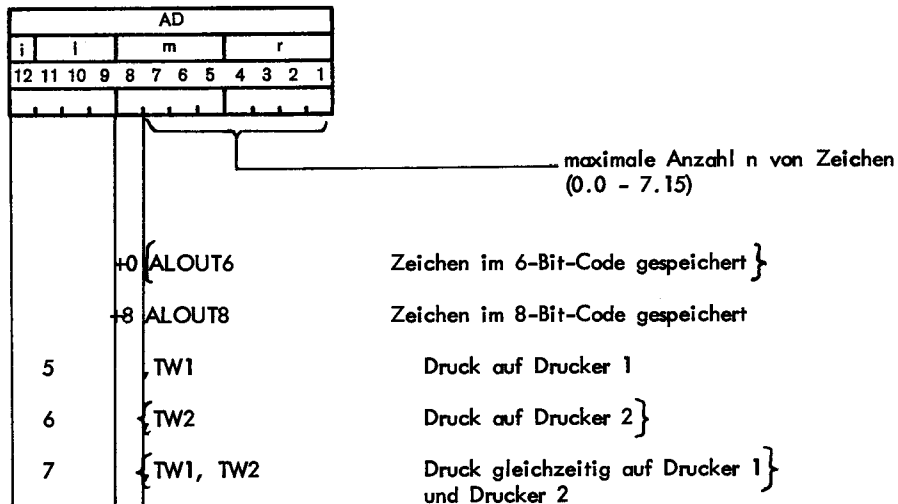
1. Es findet ein Transport von sedezimalen Stelleninhalten zwischen dem ALC-Bereich und einem Indexregister statt. Die Stellen im Bereich, die der Sedezimalzahl entsprechen, sind durch den Inhalt des Bereichszeigers I_3 festgelegt. Beim Transport von zwei Zahlen entspricht die erste dem Inhalt von I_m , die zweite dem Inhalt von I_r .
2. Der Bereichszeiger I_3 ist nach Ausführung des Befehls um die Anzahl der transportierten Zahlen weitergestellt und zeigt folglich im Bereich auf die der zuletzt transportierten Stelle folgende Stelle.

	ALC-Bereich und Indexregister	3.1
--	-------------------------------	-----

Bemerkungen:

1. Man beachte, daß der Bereichszeiger I_3 in I_{im} immer die interne Speicherwortnummer enthält, die aus der normalen Speicherwortnummer durch Addition von $\{0.5\}$ bzw. $[0.4]$ entsteht (vgl. Beispiele und den Befehl "BPX Laden ALC-Bereichszeiger (3.11)").
2. Man beachte die Befehle "XF und XFR Indexvorbefehl (2.2)" bei der Wahl des Indexregisters (vgl. Beispiele).

ALOUT	Drucken Inhalt ALC-Bereich (alphanumeric output)	3.1
-------	---	-----

Wirkung:

1. Der Inhalt des ALC-Bereichs wird fortlaufend ab der gegenwärtigen Kugelkopfposition auf dem Drucker ausgegeben. Den Anfangspunkt des ALC-Bereichs legt der Inhalt des Bereichszeigers in I_3 fest.
 2. Es kann wahlweise ein Druck auf Drucker 1, auf Drucker 2 oder auf beiden Druckern gleichzeitig erfolgen, wobei die Druckfarbe durch Vorbefehle einzeln angegeben werden kann.
 3. Bei dem Modell 820/15 darf nur eine Ausgabe auf dem ersten Drucker codiert werden. Bei einem Befehl, in dem der zweite Drucker angegeben ist, wird dieser Fehler nicht erkannt. Wird ein Druck auf Drucker 1 und 2 befohlen, so wird nur der Drucker 1 angesprochen, das Programm läuft unbehindert weiter.
 4. Der Ausdruck wird beendet, wenn unter den ersten n Zeichen das Umschaltzeichen 3.15 oder eines der Endezeichen (Codes 6.8) auftritt. Kommt keines dieser Zeichen vor, so werden genau n Zeichen ausgegeben.
- Im 6-Bit-Code bedeutet das Zeichen 3.15 eine Umschaltung auf den 8-Bit-Code des Endezeichens. Der Inhalt der beiden folgenden Kernspeicherstellen wird als Endezeichen interpretiert.

ALOUT	Drucken Inhalt ALC-Bereich (alphanumeric output)	3.1
-------	---	-----

Bei der Speicherung im 8-Bit-Code entfällt im allgemeinen das Zeichen 3.15 vor dem Endezeichen, da eine Umschaltung nicht erforderlich ist. Tritt es trotzdem auf, so wird das folgende Zeichen als Endezeichen interpretiert.

Je nachdem, ob der Ausdruck durch Auftreten eines Endezeichens oder durch Erreichen der Anzahl n beendet wurde, werden die folgenden Merkerstellungen intern zwischengespeichert.

Beendigung, Bit 1 bis 4 des Endezeichens	ML	MU	MC
Ende durch Endezeichen, 8	0	1	0
Ende durch Endezeichen, 12	0	0	1
Ende durch Endezeichen, 11, sonst	1	0	0
Ende durch Erreichen der Anzahl n	0	0	0

Im Befehl "SMWT Setze Merker nach Ende Simultan-Ausgabe (2.12)" wird auf das Ende der Ausgabe gewartet. Im Anschluß daran erfolgt das Setzen der Merker entsprechend obiger Tabelle.

- Haben die letzten 4 Bit des Endezeichens den Wert 12, so tabuliert der Kugelkopf des angesprochenen Druckers nach Beendigung des Drucks auf die Ausgangsposition zurück. In allen anderen Fällen bleibt er nach dem zuletzt geschriebenen Zeichen stehen.
- Der Bereichszeiger I_3 zählt während des Ausdrucks mit. Nach Beendigung durch ein Endezeichen zeigt er auf die dem Endezeichen folgende Kernspeicherstelle, nach Beendigung durch Erreichen der Anzahl n zeigt er im Bereich auf die dem zuletzt ausgedruckten Zeichen folgende Kernspeicherstelle.

Beispiel:

Aus dem ALC-Bereich soll der ab Speicherwort 22 Stelle 14 im 6-Bit-Code gespeicherte Text auf Drucker 2 ausgedruckt werden. Der Kugelkopf vom Drucker 2 befinde sich in Position 76.

Zunächst wird der Bereichszeiger I_3 des ALC-Bereichs geladen.

BW-Adresse	Op. Teil	Adreßteil	Symb. Adr.	Bemerkungen
0	3 11 0	1 6 14		BPX, 22, 14
1				

nachher: I_3 1 11 14 entspricht SW 22 Stelle 14

ALOUT	Drucken Inhalt ALC-Bereich (alphanumeric output)	3.1
-------	---	-----

Danach folgt der Druckbefehl:

vorher: I₃ 1 11 14

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
0	3 1	1	0 6 0 7		ALOUT6, TW2,7
1					

nachher: I₃ 1 12 11

SW 22 [] [] [] [] [] [] [] [] [] [] [] [] 9 2

SW 23 10 9 2 5 5 9 14 15 12 6 8 [] [] [] []

Auf Drucker 2 wird ausgedruckt SYSTEM und zwar in schwarz, wenn zuvor kein Druckvorbefehl gegeben worden ist. Die Merker ML, MU, MC werden durch diesen Befehl noch nicht verändert.

Die darauffolgenden Befehle dürfen I₃ und die angesprochenen Stellen der Speicherorte 22 und 23 nicht verändern. Würde auch kein Ausgabebefehl für das 2. Si-multangerät gegeben, so ist nach dem Befehl

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
0	2 12	1	0 0 10 1		SMWT, MTWS
1					

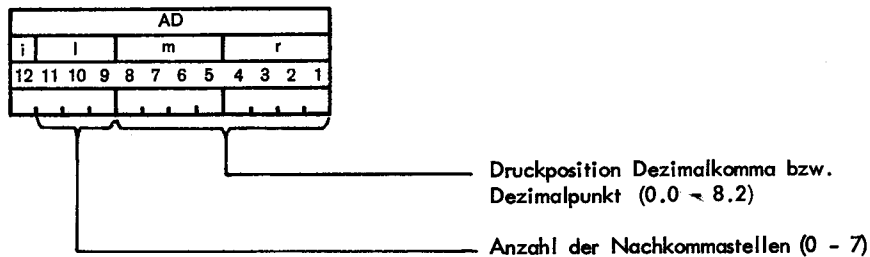
der Ausdruck auf dem Drucker beendet. Der Kugelkopf befindet sich in Position 82, und die Merker sind ML = 0, MU = 1, MC = 0.

ALOUT	Drucken Inhalt ALC-Bereich (alphanumeric output)	3.1
-------	---	-----

Bemerkungen:

1. Der Befehl "ALOUT Drucken Inhalt ALC-Bereich (3.1)" wird simultan zum weiteren Befehlsablauf ausgeführt. Vor einer neuen Ausgabe auf dem benutzten Simultange-
rät wartet intern das Betriebsprogramm auf die Beendigung der Operation. Man be-
achte jedoch, daß der angesprochene Teil des ALC-Bereichs und der Bereichszei-
ger I_3 erst nach Beendigung der Ausgabe verändert werden dürfen (vgl. "Warte- und
Merkersetzbefehl (2.12)").
2. Ist der Code im 6-Bit-Code gespeichert, so steht der Bereichszeiger nur bei jedem
zweiten Zeichen richtig. Darauf ist besonders dann zu achten, wenn der Aus-
druck durch Erreichen der Anzahl n beendet wurde.
3. Man beachte, daß der Bereichszeiger I_3 in I_m immer die interne Speicherwort-
nummer enthält, die aus der normalen Speicherwortnummern durch Addition von
[0.4] bzw. { 0.5 } entsteht (vgl. Beispiel und den Befehl "BPX Laden ALC-Ber-
eichszeiger (3.11)").
4. Eine Ausgabe der Zeichen im 6-Bit-Code ist beim Modell 820/15 nicht möglich.
Hier aknn nur im 8-Bit-Code gespeichert werden. Wird jedoch eine Ausgabe im
6-Bit-Code codiert, so stoppt das Programm auf diesem Befehl. Es erfolgt jedoch
keine interne Fehlermeldung.

ED	Druckbefehl (edition)	3.2
----	-----------------------	-----

Wirkung:

- Der Inhalt des im Druckvorbefehl angegebenen Druckspeicherwortes wird auf dem zugeordneten Drucker ausgegeben. Ohne einen vorangegangenen Befehl "EDF oder EDFS Druckvorbefehl (3.6)" erfolgt Ausdruck Inhalt von D1 auf Drucker 1, ohne Vorkommastellen falls (D1) absolut kleiner 1, in schwarz, wenn $(D1) \geq 0$, in rot, wenn $(D1) \leq 0$, ohne anschließendes Kennzeichen. Der Inhalt des Druckspeicherwortes bleibt unverändert.

Es ist zu beachten, daß beim Modell 820/15 nur eine Ausgabe auf dem Drucker 1, also auch nur aus dem Druckspeicherwort D1 erfolgen kann.

- Ist zuvor ein Druckvorbefehl mit $AD_{lmr} \neq 0$ gegeben worden, so erfolgt die Ausgabe entsprechend den Anweisungen des Druckvorbefehls.
- Die Anzahl der auszugebenden Nachkommastellen ist in AD_i anzugeben. Von $AD_i = 0$ bis $AD_i = 6$ wird die codierte Anzahl von Nachkommastellen, bei $AD_i = 7$ die volle, durch die KA festgelegte Anzahl von Nachkommastellen ausgegeben.
- Die Druckposition des Dezimalkommas bzw. Dezimalpunktes ist in AD_m anzugeben. Bei Ausgabe von $AD_i = 0$ Nachkommastellen findet jedoch kein Druck vom Komma oder Punkt statt. Durch entsprechende Belegung des AD-Teils vom Befehl unter Adresse 0.0.0 im Anwenderprogrammblock 0 ist festzulegen, ob bei Ausgabe von Nachkommastellen zuvor ein Dezimalkomma oder ein Dezimalpunkt gedruckt werden soll.
- Ist der Inhalt des Druckspeicherwortes gleich 0 und gilt kein Druckvorbefehl, so werden bei $AD_i \neq 0$ nur dann Nachkommastellen ausgegeben, wenn dies durch entsprechende Belegung des AD-Teils vom Befehl unter Adresse 0.0.0 im Anwenderprogrammblock 0 angeordnet wurde.

ED	Druckbefehl (edition)	3.2
----	-----------------------	-----

Beispiel: (KA 9)

Der AD-Teil vom Befehl unter Adresse 0.0.0 im Anwenderprogrammblock 0 ist folgendermaßen belegt:
Block 0

BW-Adresse			Op. Teil			Adreßteil			Symb. Adr.	Bemerkungen
0	0	0	0	0	1	0	2	9		
		1								

Auf Drucker 2 ist der Inhalt des dazugehörigen Druckspeicherwortes D2 mit 9 Nachkommastellen und Zwischensummenzeichen auszugeben. Der Dezimalpunkt soll dabei in Position 63 stehen.

vorher: SW 2 + 0 0 0 1 2 3 4 7 0 0 0 0 0 0 0

BW-Adresse			Op. Teil			Adreßteil			Symb. Adr.	Bemerkungen
		0	3	6	0	5	14	1		EDFS, REDG, ITS, 1
		1	3	2	0	7	3	15		ED, 7..63
		2								

nachher: SW 2 + 0 0 0 1 2 3 4 7 0 0 0 0 0 0 0

Es wird auf Drucker 2 ausgegeben

123.47000000 ◊

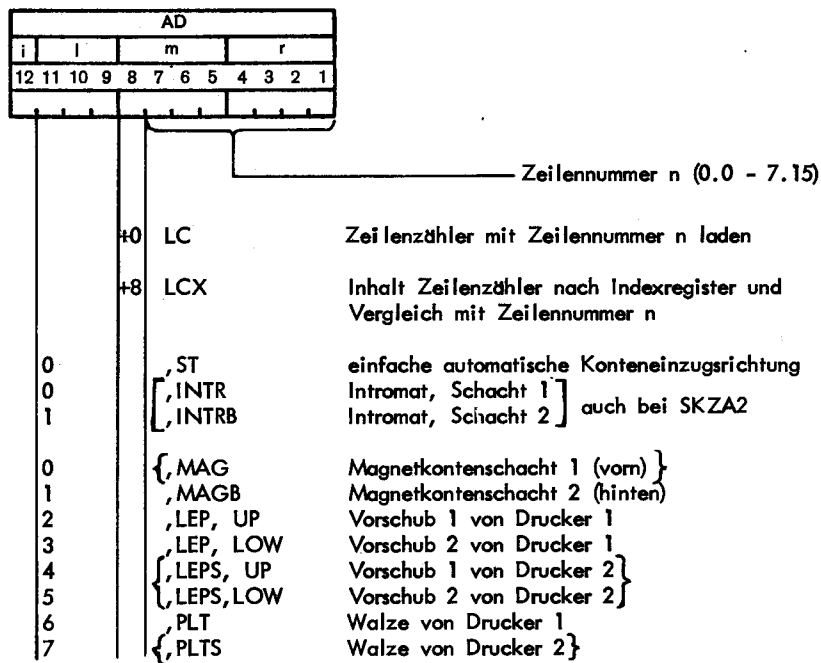
in rot, da die Zahl positiv ist und "REDG" gefordert war (vgl. "EDF und EDFS Druckvorbefehle (3.6)"). Der Kugelkopf vom Drucker 2 steht anschließend auf Position 74.

ED	Druckbefehl (edition)	3.2
----	-----------------------	-----

Bemerkungen:

1. Man beachte, daß im Fall $(D) = 0$ keine Ausgabe auf dem Drucker erfolgt, wenn $AD_1 = 0$ ist und kein Druckvorbefehl mit $AD_r \neq 0$ gegeben wurde.
2. Bei einer Position größer als 8.2 kann die Tabulation nicht durchgeführt werden, d.h. der Tabulationsvorgang kommt zu keinem Abschluß. Dies muß unbedingt vermieden werden.
3. Ist die in AD_1 angegebene Anzahl von Nachkommastellen größer als die Kommagrundausstattung KA , so erfolgen undefinierte Ausgaben.
4. Der Befehl "ED, Drucken Inhalt Druckspeicherwort (3.2)" wird simultan zum weiteren Befehlsablauf ausgeführt. Vor einer neuen Ausgabe auf dem benutzten Simultangerät sowie vor einer Veränderung des angesprochenen Druckspeicherwortes wartet intern das Betriebsprogramm auf die Beendigung der Operation. Dazu ist kein Wartebefehl erforderlich.
5. } Ist für jeden Drucker ein Druckvorbefehl gegeben worden, so wird nur auf Drucker 2 gedruckt und anschließend der dazugehörige Druckvorbefehl aufgehoben. Damit auch auf Drucker 1 gedruckt wird, ist ein zweiter Druckvorbefehl zu geben, der jedoch simultan zum ersten ausgeführt wird.
6. Ist der Inhalt des Druckspeicherwortes sedezimal, so werden für die Zahlen 10 bis 15 die Zeichen ausgegeben, die den ALC-Codes 0.10 bis 0.15 zugeordnet sind.
7. Bei simultanem Ablauf des Druckbefehls sollte die KA in I_5 nicht verändert werden.

	Zeilenzähler-Zugriff (line counter)	3.3
--	-------------------------------------	-----



Wirkung:

1. Ist im AD-Teil Bit 8 = 0 gesetzt, so wird der in AD_i codierte Zeilenzähler mit der in Bit 1 bis Bit 7 angegebenen Zeilennummer n geladen.
2. Ist im AD-Teil Bit 8 = 1 gesetzt, so wird der Inhalt des in AD_i codierten Zeilenzählers in ein Indexregister gebracht und mit der in Bit 1 bis Bit 7 angegebenen Zeilennummer n verglichen. Das Ergebnis des Vergleichs wird durch Setzen der Merker ML und MU entsprechend folgender Tabelle zum Ausdruck gebracht:

	ML	MU
(Zeilenzähler) < n	1	1
(Zeilenzähler) > n	0	1
(Zeilenzähler) = n	0	0

	Zeilenzähler-Zugriff (line counter)	3.3
--	-------------------------------------	-----

Beispiel:

Der Inhalt des Zeilenzählers vom 2. (hinteren) Magnetkontenschacht soll in das Indexregister I_1 gebracht und mit der Zeilennummer 26 verglichen werden. Der Inhalt des Zeilenzählers sei 31.

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
	0 2 2	0	0 0 1		XF, 1
	1 3 3	0	1 9 10		LCX, MAGB, 26
	2				

nachher: I_1

0	1	15
---	---	----

 entspricht $(I_1) = 31$

Außerdem sind die Merker $ML = 0$ und $MU = 1$ gesetzt.

Bemerkungen:

1. Vor jedem Einzug einer Magnetkontokarte muß der Zeilenzähler des betreffenden Schachtes mit einer Zahl m geladen werden, damit die Karte auf die richtige Zeile eingezogen wird. Diese Zahl m berechnet sich aus der gesamten Kartenhöhe und der Höhe des Kartenkopfes jeweils in Zeilen gemessen nach der Formel:

$$m = \text{Kartenhöhe} - \text{Kopfhöhe} + 13$$

Unter dieser Voraussetzung trägt die erste beschreibbare Zeile von oben die Nummer 0.

Beispiel: Die Magnetkontokarte hat eine Höhe von 11,5", d.h. 69 Zeilen + 3 mm. Der Kartenkopf hat die vorgeschriebene Mindesthöhe von 14 + 3 mm. Dann ist

$$m = 69 - 14 + 13 = 68,$$

d.h. vor dem Einzug einer Magnetkontokarte ist der Zeilenzähler mit der Nummer 68 zu laden. Wird anschließend ein Einzug auf Zeile 0 befohlen, so kommt die erste beschreibbare Zeile, d.h. die 15. Zeile von oben, auf die Höhe des Schreibkopfes.

2. Man beachte den Befehl "XF bzw. XFR Indexvorbefehl (2.2)".
3. Es ist zu beachten, daß die automatische Kontoeinzugseinrichtung ebenfalls einen Zeilenzähler hat. Ebenso der Intromat. Dieser Zeilenzähler darf allerdings nicht zum Einzug eines Kontos verwendet werden, da der Einzug automatisch

	Zeilenzähler-Zugriff (line counter)	3.3
--	-------------------------------------	-----

beim Einlegen der Karte in den Schacht erfolgt. Der Zeilenzähler dient lediglich zur vertikalen Formularpositionierung des Kontos. Bei den Modellen 820/25/35 können die Befehle für die automatische Konteneinzugseinrichtung nur angewendet werden, wenn das Betriebsprogramm SKZA2 vorhanden ist.

Werden Magnetkontenbefehle verwendet, so ist das Betriebsprogramm MSKZ2 unbedingt erforderlich. Die Befehle für den Intromaten sind nicht anwendbar.

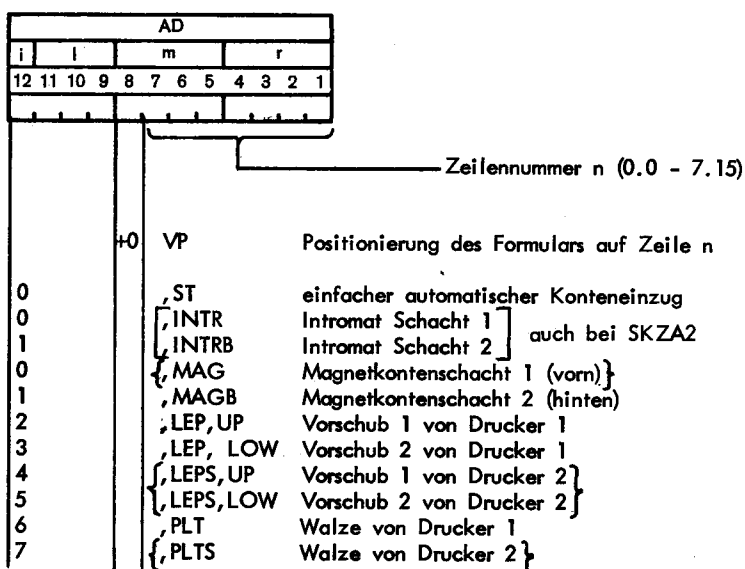
Bei Modell 820/15 müssen folgende Moduln bei Anwendung der einzelnen Zeilenzähler vorhanden sein:

- . Intromat Schacht 1 und 2 Modul IN
- . einfacher automatischer Konteneinzug Modul ST
- . Magnetkontenschacht einfach Modul MC

Es ist unbedingt zu vermeiden, den zweiten Drucker (Walze, Vorschubeinrichtung) anzusprechen, da für diese Geräte kein Zeilenzähler aufgestellt worden ist. Wird jedoch ein nicht vorhandener Zeilenzähler angesprochen, so werden andere interne Speicherstellen verändert. Der weitere Programmablauf kann dadurch erheblich gestört werden. Das Betriebsprogramm erkennt diese Fehler nicht.

Von der Programmierung muß der Schacht 2 der Magnetkonteneinrichtung angesprochen werden.

VP	Vertikalpositionierung (vertical position)	3,4
----	--	-----

Wirkung:

- Das jeweilige Formular wird auf die angegebene Zeilennummer transportiert und der dazugehörige Zeilenzähler auf den neuen Stand gebracht. Ein Rückwärtstransport kann jedoch nur in den Magnetkontenschächten stattfinden.
- Bei der Positionierung von Magnetkontokarten bleibt der Merker MC unverändert.
- Ist bei anderen Formularträgern als Magnetkontenschächten der Inhalt des Zeilenzählers größer als die codierte Zeilennummer, so gilt für die verschiedenen Betriebsprogramme folgendes:

MSKZ1 + MSKZ2:

Es erfolgt keine Formularbewegung, aber es wird Merker MC = 1 gesetzt (andernfalls MC = 0).

MSKZ1 + SKZA2:

Es erfolgt keine Formularbewegung, Merker MC wird nicht verändert.

MFA1 bzw. MFAGS1:

Fehllauf des Programmes ohne Fehlermeldung. Deshalb ist eine codierte Zeilennummer kleiner als Inhalt des Zeilenzählers unbedingt zu vermeiden.

VP	Vertikalpositionierung (vertical position)	3.4
----	--	-----

BW-Adresse	Op. Teil			Adreßteil			Symb. Adr.	Bemerkungen
0	3	4	0	5	4	5		VP, LEPS, LOW, 69
1	3	3	0	5	0	0		LC, LEPS, LOW
2								

Nach Ausführung des Befehls VP, LEPS, LOW, 69 ist MC = 0 und der Inhalt des Zeilenzählers 69. Durch den darauffolgenden Befehl LC, LEPS, LOW wird der Zeilenzähler mit der Zeilennummer 0 geladen.

Bemerkungen:

1. Der Befehl "VP Vertikalpositionierung (3.4)" ist keinesfalls zum Einzug von Magnetkontokarten zu verwenden. Auch darf er nicht innerhalb der Gültigkeit eines Vorbefehls "BL Blockumschaltung (2.11)" gegeben werden. Andernfalls ist dieser gelöscht.
2. Bei Magnetkontokarten hat die erste beschreibbare Zeile von oben immer die Ziffer 0.
3. Aufeinanderfolgende Zeilenschaltung und Vertikalpositionierung von gleichen und verschiedenen Formularen werden gleichzeitig ausgeführt. Ein solcher Formulartransport an einem Drucker kann simultan zu einer davon unabhängigen Ausgabe am anderen Simultangerät ablaufen (z.B. Druck auf dem anderen Drucker, sowie Lochstreifen- oder Lochkartenausgabe) falls Formulartransport auf dem 1. Drucker. Dazu muß jedoch die andere Ausgabe vor dem Formulartransport befohlen werden, da Befehle nach Formulartransporten erst nach deren Beendigung ausgeführt werden.
4. Bei den Modellen 820/15 dürfen Zeilenschaltungen und Vertikalpositionierungen für den gleichen Formularträger nicht hintereinander folgen. Aufeinanderfolgende Zeilenschaltungen und Vertikalpositionierungen von verschiedenen Formularträgern werden gleichzeitig ausgeführt. Ansonsten gelten die Angaben unter Punkt 3.
5. Es ist weiterhin darauf zu achten, daß die Befehle "VP (3.4) und "ALBG" (3.4) nicht unmittelbar aufeinanderfolgen. In diesem Falle würde der Befehl "ALBG" als Leerbefehl angesehen werden.
6. Wird ein nicht anschließbarer Formularträger angesprochen, zum Beispiel der Magnetkontenschacht 1, Vorschub 1 und 2 am Drucker 2 oder die Walze von Drucker 2, so läuft das Programm fehl. Es tritt keine Fehlermeldung ein.

VP	Vertikalpositionierung (vertical position)	3.4
----	--	-----

7. Falls die folgenden Formularträger angesprochen werden, müssen die angeführten Moduln vorhanden sein:

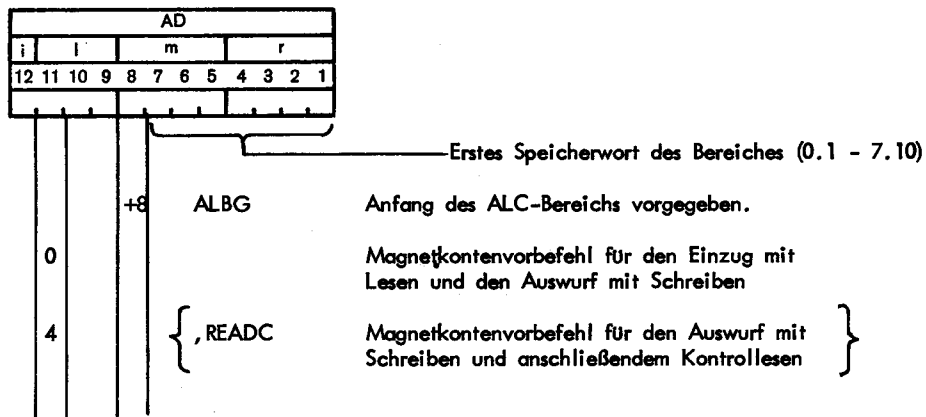
automatische Konteneinzugseinrichtung	Modul ST
Intromat	Modul IN
Magnetkonteneinrichtung	Modul MC

Sind die Moduln nicht vorhanden, und es wird ein entsprechender Formularträger angesprochen, so ist die Wirkung gleich der unter Punkt 6 beschriebenen Möglichkeit.

8. Es ist nur ein Magnetkontenschacht anschließbar. Von der Programmierung her muß er als Schacht 2 angesprochen werden.

9. Bei den Modellen 820/25/35 kann nur dann ein Intromat oder einfacher Konteneinzug angeschlossen werden, wenn das Betriebsprogramm SKZA2 vorhanden ist. Bei der Magnetkonteneinrichtung muß das Betriebsprogramm MSKZ2 vorhanden sein.

ALBG	ALC-Bereichsanfang (ALC-buffer begin)	3.4
------	---------------------------------------	-----



Wirkung:

1. Das in AD Bit 1 bis Bit 7 angegebene Speicherwort wird zum ersten SW des ALC-Bereichs. Dieser Bereich dient als Ausgabe- und Eingabe-Bereich für Magnetkontokarten.
2. Vor einem Magnetkonten-Auswurf mit Schreiben muß der Anfang des Bereichs definiert sein, dessen Inhalt auf den Magnetstreifen übertragen wird. Im Vorbefehl ist außerdem festzulegen, ob die aufgeschriebene Information anschließend zur Kontrolle gelesen und mit dem Bereichsinhalt verglichen werden soll.

[Eine Kontrolllesung beim Modell 820/15 ist nicht möglich.]
3. Vor einem Magnetkonteneinzug mit Lesen muß der Anfang des Bereiches definiert sein, in den die gelesene Information übertragen werden soll.

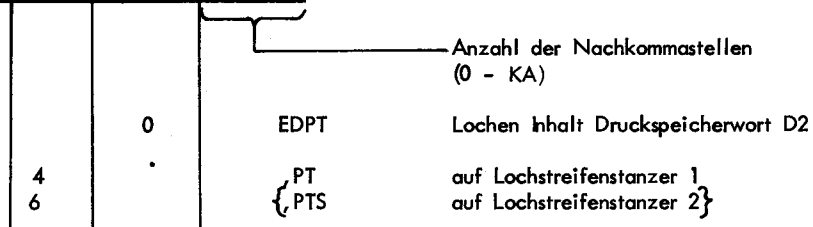
ALBG	ALC-Bereichsanfang (ALC-buffer begin)	3.4
------	---------------------------------------	-----

Bemerkungen:

1. Der ALC-Bereich dient außerdem zur Speicherung alphanumerischer Zeichen im ALC-Code. Hierzu ist der durch den Befehl "ALBG ALC-Bereichsanfang (3.4)" definierte Bereich jedoch bedeutungslos, da für diese Verarbeitung nur die Stellung des Bereichszeigers I_3 benutzt wird.
2. Der Inhalt des zum ALC-Bereich gehörenden Bereichszeigers I_3 wird durch den Befehl "ALBG ALC-Bereichsanfang (3.4)" nicht verändert.
3. Durch den Befehl "CLRAL Löschen ALC-Bereich (0.15)" kann eine vorzugebende Anzahl von Speicherworten des ALC-Bereichs beginnend mit Bereichsanfang gelöscht werden.
4. Ein Magnetknoten-Einzug mit Lesen darf nicht innerhalb der Gültigkeit eines Befehls "ALBG, READC Vorbefehl für Auswurf mit Schreiben und Kontrolllesen (3.4)" gegeben werden. Andernfalls wird die Einzugshöhe fälschlicherweise aus den Stellen 2 und 3 des zuerst gelesenen Speicherwortes entnommen.
5. Der Anfang des ALC-Bereichs, der Magnetknoten-vorbefehl und der Anfang des PCT-Ausgabebereichs werden intern in der gleichen Speicherzelle gemerkt. Man beachte also, daß sich die Befehle "ALBG (3.4)" und der Befehl "PBG,PN (2.11)" gegenseitig aufheben.
6. Die Befehle "VP", und "ALBG" dürfen nicht aufeinander folgen, da sonst der Befehl ALBG ohne Wirkung bleibt.

EDPT	Stanzen Inhalt Druckspeicherwort (edition punch tape)	3.5
------	--	-----

AD											
j			l			m			r		
12	11	10	9	8	7	6	5	4	3	2	1



Wirkung:

1. Der Inhalt von Druckspeicherwort D2 wird auf dem gewählten Lochstreifenstanzer im Lochstreifencode ausgegeben. Ohne einen vorangegangenen Befehl "PTF Lochstreifenvorbefehl (3.5)" erfolgt die Ausgabe in dem in Tabelle 1 angegebenen Lochstreifencode, ohne Vorkommastellen falls (D2) absolut kleiner 1, ohne anschließendes Kennzeichen.
Der Inhalt von Druckspeicherwort D2 bleibt unverändert.
2. Ist zuvor ein Lochstreifenvorbefehl gegeben worden, so erfolgt die Ausgabe entsprechend den Anweisungen des Vorbefehls.
3. Die Anzahl der auszugebenden Nachkommastellen ist in AD anzugeben. Ist jedoch der Inhalt des Druckspeicherwortes gleich 0 und gilt kein Lochstreifenvorbefehl, so werden bei $AD \neq 0$ nur dann Nachkommastellen ausgegeben, wenn dies durch entsprechende Belegung des AD-Teils vom Befehl unter Adresse 0.0.0 im Anwenderprogrammblock 0 angeordnet wurde. Bei Ausgabe von $AD = 0$ Nachkommastellen findet auch keine Ausgabe von Dezimalpunkt oder -komma statt.
4. Durch entsprechende Belegung des AD-Teils vom Befehl unter Adresse 0.0.0 im Anwenderprogrammblock 0 ist festzulegen, ob bei Ausgabe von Nachkommastellen zuvor ein Dezimalpunkt (PT-Code 5.15) oder ein Dezimalkomma (PT-Code 5.14) gelocht werden soll.
5. Eine Ausgabe auf Gerät 2 ist nicht möglich, da beim Modell 820/15 keine Zweitgeräte angeschlossen werden können. Es ist also unbedingt zu vermeiden, den Zusatz PTS zu benutzen.

EDPT	Stanzen Inhalt Druckspeicherwort (edition punch tape)	3.5
------	--	-----

Beispiel: (KA 5)

Der AD-Teil vom Befehl unter Adresse 0.0.0 im Anwenderprogrammblock 0 ist folgendermaßen belegt:

Block 0

BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
	0 0 0	0	0 2 5		
	1				

Der Inhalt von Druckspeicherwort D2 soll mit 7 Stellen vor dem Komma auf dem Lochstreifenstanzer 1 ausgelocht werden, wobei das Zahlenfeld nach vorn eventuell mit Sicherungsternen auszufüllen ist.

vorher: SW 2

1	0	0	0	0	0	0	0	0	2	1	8	5	0	3	4	7
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
	0 3 5	0	5 11 7		PTF, STAR, ITS, TBL2,7
	1 3 5	0	4 0 3		EDPT, PT, 3
	2				

nachher: SW2

1	0	0	0	0	0	0	0	0	2	1	8	5	0	3	4	7
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Es werden folgende Zeichen der Reihe nach ausgelocht:

*** 2 1 8, 5 0 3 ◊

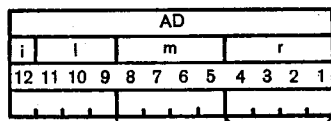
Die Umcodierung in den Lochstreifencode erfolgte nach Tabelle 2, wobei für das Dezimalkomma der PT-Code von 5.14 ausgegeben wurde.

EDPT	Stanze Inhalt Druckspeicherwort D2 (edition punch tape)	3.5
------	--	-----

Bemerkungen:

1. Man beachte, daß im Fall $(D2) = 0$ keine Ausgabe auf dem Streifenlocher erfolgt, wenn $AD_r = 0$ ist und kein Lochstreifenvorbefehl mit $AD_r \neq 0$ gegeben wurde.
2. Ist die in AD_r angegebene Anzahl von Nachkommastellen größer als die Komma-grundausrüstung KA , so erfolgen undefinierte Ausgaben.
3. Der Befehl "EDPT Stanze Inhalt Druckspeicherwort (3.5)" wird simultan zum weite-ren Befehlsablauf ausgeführt. Vor einer neuen Ausgabe auf Simultangerät ⁽²⁾ bzw. 1 sowie vor einer Veränderung des Druckspeicherworts D2 wartet intern das Betriebsprogramm auf die Beendigung der Operation. Dazu ist kein Wartebefehl erforderlich.
4. Ist der Inhalt von Druckspeicherwort D2 dezimal, so werden für die Zahlen 10 bis 15 die Zeichen ausgegeben, die den ALC-Codes 0.10 bis 0.15 zugeordnet sind.
5. Die Umcodierung des ALC-Codes in den Lochstreifencode geschieht nach Codetabel-len, die sich im Anwenderprogrammblock 0 befinden müssen. Die Anfangsadresse von Tabelle 1 ist im AD-Teil des Befehls unter Adresse 0.0.3, die von Tabelle 2 im AD-Teil des Befehls unter Adresse 0.0.4 im Festspeicherblock 0 anzugeben.

PTF	Lochstreifenvorbefehle (punch tape following)	3.5
-----	--	-----



Anzahl der Wiederholungen n (0-15)

2	PTF	Lochstreifenvorbefehl für "ALC Ausgabe eines Zeichens (2.15)"
+0	,TBL1	Codierung nach PT-Tabelle 1
+8	,TBL2	Codierung nach PT-Tabelle 2

Wirkung:

- Der Befehl "PTF Lochstreifenvorbefehl (3.5)" wirkt auf einen nachfolgenden Befehl "ALC Ausgabe eines Zeichens (2.15)". Nach der Ausführung dieses Befehls ist der Vorbefehl gelöscht. Ein nicht gegebener oder gelöschter Vorbefehl wirkt wie ein Vorbefehl mit $AD_{lmr} = 0.2.0$.
- Das im Hauptbefehl codierte Zeichen wird n+1-mal hintereinander ausgegeben. Die Codierung des ALC-Codes in den Lochstreifencode kann wahlweise nach PT-Tabelle 1 oder PT-Tabelle 2 erfolgen.

Bemerkungen:

- Für den Befehl "ALC Ausgabe eines Zeichens (2.15)" sind nur Bit 6 und Bit 8 sowie der AD-Teil des Vorbefehls von Bedeutung. Alle übrigen Anweisungen haben auf den Hauptbefehl keinen Einfluß.
- Die Vorbefehle "PTF Lochstreifenvorbefehl (3.5)", {"EDFS Druckvorbefehl für Drucker 2 (3.6)"} und "PCF Lochkartenvorbefehl (3.8)" werden intern in derselben Speicherzelle gemerkt. Man beachte also, daß sich diese Befehle gegenseitig aufheben und ersetzen.

PTF	Lochstreifenvorbefehl (punch tape following)	3.5
-----	---	-----

AD											
i	l	m						r			
12	11	10	9	8	7	6	5	4	3	2	1

Mindestanzahl der auszugebenden Zeichen
vor dem Komma (0 - 15)

	2	PTF	Lochstreifenvorbefehl für "EDPT Stanzen Inhalt Druckspeicherwort (3.5)"
	+0	,TBL1	Codierung nach Tabelle 1
	+8	,TBL2	Codierung nach Tabelle 2
	+0	,ZERO	Vornullen
	+1	,STAR	Sicherungssterne
	+4	,BLNK	Leertasten
0			ohne anschließendes Kennzeichen
1		,MIN	Leertaste, wenn (D2) ≥ 0 Kennzeichen -, wenn (D2) < 0
3		,SGN	Kennzeichen +, wenn (D2) ≥ 0 Kennzeichen -, wenn (D2) < 0
5		,ITS	Kennzeichen ◊, wenn (D2) ≥ 0 Kennzeichen ⊘, wenn (D2) < 0
7		,FTS	Kennzeichen *, wenn (D2) ≥ 0 Kennzeichen †, wenn (D2) < 0

Wirkung:

1. Der Befehl "PTF Lochstreifenvorbefehl (3.5)" wirkt auf einen nachfolgenden Befehl "EDPT Stanzen Inhalt Druckspeicherwort D2 (3.5)". Nach der Ausführung dieses Befehls ist der Vorbefehl gelöscht. Ein nicht gegebener oder gelöschter Vorbefehl wirkt wie ein Vorbefehl mit AD_lm_r = 0.2.0.
2. Der Vorbefehl enthält Anweisungen bezüglich der Ausgabe von Vorkommastellen, Sicherungszeichen und anschließendem Kennzeichen. Ferner legt er fest, ob die Codierung des ALC-Codes in den Lochstreifencode nach PT-Tabelle 1 oder PT-Tabelle 2 erfolgen soll.

Blatt B 148

Datenverarbeitungssystem 820

1. 9. 1970

Beschreibung der Befehle

PTF	Lochstreifenvorbefehl (punch tape following)	3.5
-----	---	-----

Bemerkung:

Die Vorbefehle "PTF Lochstreifenvorbefehl (3.5)", {"EDFS Druckvorbefehl für Drucker 2 (3.6)"} und "PCF Lochkartenvorbefehl (3.8)" werden intern in derselben Speicherzelle gemerkt. Man beachte also, daß sich diese Befehle gegenseitig aufheben und ersetzen.

PTF	Lochstreifenvorbefehl (punch tape following)	3.5
-----	---	-----

AD											
i			l			m			r		
12	11	10	9	8	7	6	5	4	3	2	1
.

2	PTF	Lochstreifenvorbefehl für "PBA, PT Bereich nach SW A (3.7)" "DEC, PT Decodierung (3.7)" "RDX, PT Bereich nach I (3.7)"
+0	,TBL1	Decodierung nach PT-Tabelle 1
+8	,TBL2	Decodierung nach PT-Tabelle 2

Wirkung:

- Der Befehl "PTF Lochstreifenvorbefehl (3.5)" wirkt auf einen der nachfolgenden, oben aufgeführten Befehle. Nach der Ausführung dieses Befehls ist der Vorbefehl gelöscht. Ein nicht gegebener oder gelöschter Vorbefehl wirkt wie ein Vorbefehl mit $AD_{imr} = 0.2.0$.
- Der Vorbefehl legt fest, ob die im Hauptbefehl durchzuführende Decodierung des PT-Codes in den ALC-Code nach PT-Tabelle 1 oder PT-Tabelle 2 erfolgen soll.

Bemerkungen:

- Für die aufgeführten Befehle sind nur Bit 6 und Bit 8 des Vorbefehls von Bedeutung. Alle übrigen Anweisungen haben auf die Hauptbefehle keinen Einfluß.
- Die Vorbefehle "PTF Lochstreifenvorbefehl (3.5)", {"EDFS Druckvorbefehl für Drucker 2 (3.6)"} und "PCF Lochkartenvorbefehl (3.8)" werden intern in derselben Speicherzelle gemerkt. Man beachte also, daß sich diese Befehle gegenseitig aufheben und ersetzen.

PTF	Lochstreifenvorbefehl (punch tape following)	3.5
-----	---	-----

AD															
i				l				m				r			
12	11	10	9	8	7	6	5	4	3	2	1				

	2	PTF	Lochstreifenvorbefehl für "ALPT Lochen aus Bereich (3.13)"
	+0	,TBL1	Codierung nach PT-Tabelle 1
	+8	,TBL2	Codierung nach PT-Tabelle 2
4		,PT	Ausgabe auf Lochstreifenstanzer 1
6		{PTS	Ausgabe auf Lochstreifenstanzer 2 }

Wirkung:

- Der Befehl "PTF Lochstreifenvorbefehl (3.5)" wirkt auf einen nachfolgenden Befehl "ALPT Stanzen aus Bereich (3.13)". Nach der Ausführung dieses Befehls ist der Vorbefehl gelöscht. Ein nicht gegebener oder gelöschter Vorbefehl wirkt wie ein Vorbefehl mit $AD_{lmr} = 0.2.0$.
- Der Vorbefehl legt fest, ob die im Hauptbefehl durchzuführende Codierung des ALC-Codes in den Lochstreifencode nach PT-Tabelle 1 oder PT-Tabelle 2 erfolgen soll. Ferner ist durch den Vorbefehl anzugeben, ob Lochstreifenstanzer 1 [oder Lochstreifenstanzer 2] zur Ausgabe benutzt werden soll.
[Beim Modell 820/15 ist der Zusatz "PTS" $AD_{l=6}$ nicht zu verwenden.]

Bemerkungen:

- Für den Befehl "ALPT Stanzen aus ALC-Bereich (3.13)" sind nur Bit 6, Bit 8, Bit 10 und Bit 11 des Vorbefehls von Bedeutung. Alle übrigen Anweisungen haben auf den Hauptbefehl keinen Einfluß.
- Die Vorbefehle "PTF Lochstreifenvorbefehl (3.5)", {"EDFS Druckvorbefehl für Drucker 2 (3.6)} und "PCF Lochkartenvorbefehl (3.8)" werden intern in derselben Speicherzelle gemerkt. Man beachte also, daß sich diese Befehle gegenseitig aufheben und ersetzen.

PTF	Lochstreifenvorbefehl (punch tape following)	3.5
-----	---	-----

AD												
i	l	m				r						
12	11	10	9	8	7	6	5	4	3	2	1	

2	PTF	Lochstreifenvorbefehl für "RDPT Lesen nach Bereich (3.13)"
+0	,TBL1	Endezeichen aus PT-Tabelle 1
+8	,TBL2	Endezeichen aus PT-Tabelle 2
+0		6-, 7-, 8-Spur Lochstreifen
+1	,CD5	5-Spur Lochstreifen
	+0 ,YZI	nach Umschaltzeichen ZI
	+8 ,YBU	nach Umschaltzeichen BU

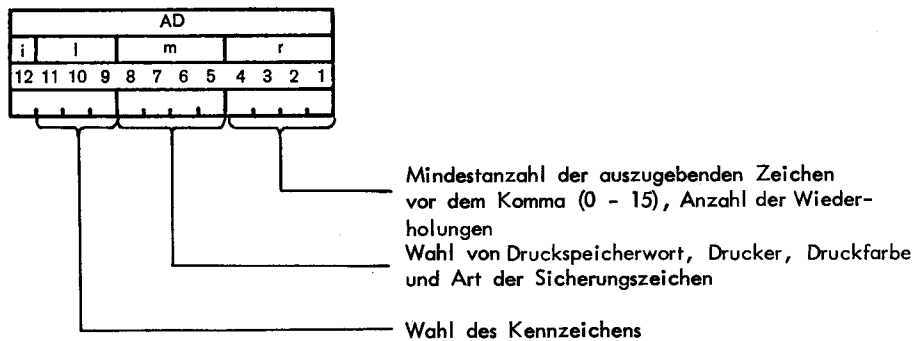
Wirkung:

- Der Befehl "PTF Lochstreifenvorbefehl (3.5)" wirkt auf einen nachfolgenden Befehl "RDPT Lesen nach Bereich (3.13)". Nach der Ausführung dieses Befehls ist der Vorbefehl gelöscht. Ein nicht gegebener oder gelöschter Vorbefehl wirkt wie ein Vorbefehl mit $AD_{lmr} = 0.2.0$.
- Der Vorbefehl legt fest, ob das Endezeichen, das im Hauptbefehl den Abbruch des Lesens bewirkt, aus PT-Tabelle 1 oder PT-Tabelle 2 zu entnehmen ist. Ferner ist anzugeben, ob es sich um 5-Spur Lochstreifen handelt, sowie ob in diesem Fall das erste gelesene Zeichen als Ziffernzeichen oder als Buchstabenzeichen interpretiert werden soll.

Bemerkungen:

- Für den Befehl "RDPT Lesen nach Bereich (3.13)" sind nur Bit 4 bis Bit 6 und Bit 8 des Vorbefehls von Bedeutung. Alle übrigen Anweisungen haben auf den Hauptbefehl keinen Einfluß.
- Die Vorbefehle "PTF Lochstreifenvorbefehl (3.5)", {"EDFS Druckvorbefehl für Drucker 2 (3.6)"}, {"PCF Lochkartenvorbefehl (3.8)"} werden intern in derselben Speicherzelle gemerkt. Man beachte also, daß sich diese Befehle gegenseitig aufheben und ersetzen.

	Druckvorbefehle (edition following)	3.6
--	-------------------------------------	-----

Wirkung:

- Die Druckvorbefehle enthalten Anweisungen, die in den Druckbefehlen aus Platzgründen nicht gegeben werden können. Jedem Drucker ist ein Druckvorbefehl zugeordnet. Nach dem Ausdruck auf einem Drucker ist der dazugehörige Druckvorbefehl aufgehoben, ein eventuell für den anderen Drucker gegebener Druckvorbefehl bleibt jedoch bestehen. Ferner setzt ein neuer Druckvorbefehl einen früheren außer Kraft.

- Die Wirkung der Druckvorbefehle auf die einzelnen Druckbefehle ist unterschiedlich und von der Art des folgenden Druckbefehls abhängig:

TW und TWX: Freigabe der alphanumerischen Tastatur (2.13)

Die Druckvorbefehle bestimmen für jeden Drucker einzeln die Druckfarbe, in der die manuell eingegebenen Zeichen gedruckt werden (ohne Vorbefehl in schwarz).

ALC: Ausgabe eines Zeichens (2.15)

Drucker 1 (oder 2) ist im Hauptbefehl anzugeben. Ein entsprechender Druckvorbefehl bewirkt, daß das Zeichen auf dem zugeordneten Drucker (AD + 1)- mal hintereinander ausgegeben wird (ohne Vorbefehl 1-mal). Bei gleichzeitigem Ausdruck auf beiden Druckern können in zwei Druckvorbefehlen unterschiedliche Anzahlen festgelegt werden. Die Druckfarbe rot ist entweder im Hauptbefehl oder in Vorbefehlen anzugeben (jedoch nicht in beiden!). Bei unterschiedlichen Druckfarben müssen evtl. zwei Druckvorbefehle gegeben werden.

	Druckvorbefehle (edition following)	3.6
--	-------------------------------------	-----

TT: Drucken Tabelle (3.0)

Im Druckvorbefehl wird festgelegt, auf welchem Drucker und in welcher Farbe die Tabelle gedruckt werden soll (ohne Vorbefehl Druck auf Drucker 1 in schwarz).

Ist für jeden Drucker ein Druckvorbefehl gegeben worden, so wird auf Drucker 2 gedruckt und anschließend der dazugehörige Druckvorbefehl aufgehoben. Damit auch auf Drucker 1 gedruckt wird, ist ein zweiter Druckbefehl zu geben, der jedoch simultan zum ersten ausgeführt wird.

{ALOUT6} und ALOUT8: Drucken Inhalt ALC-Bereich (3.1)

Die Druckvorbefehle geben für jeden Drucker die Druckfarbe an, in der die Zeichen gedruckt werden (ohne Vorbefehl in schwarz).

ED: Druckbefehl (3.2)

Für diesen Befehl gelten alle Informationen des Druckvorbefehls bezüglich der Ausgabe von Vorkommastellen, der Wahl des Druckspeicherworts, des Druckers, der Druckfarbe, der Sicherungszeichen und der Kennzeichen (ohne Vorbefehl 0 Vorkommastellen, Druckspeicherwort D1, schwarz wenn $(D1) \geq 0$, rot wenn $(D1) < 0$, kein Kennzeichen).

{ALIN6} und ALIN8: Alphanumerische Eingabe (3.13)

Die Druckvorbefehle bestimmen für jeden Drucker einzeln die Druckfarbe, in der die manuell eingegebenen Zeichen gedruckt werden (ohne Vorbefehl in schwarz).

[Beim Modell 820/15 ist nur Drucker 1 anschließbar.]

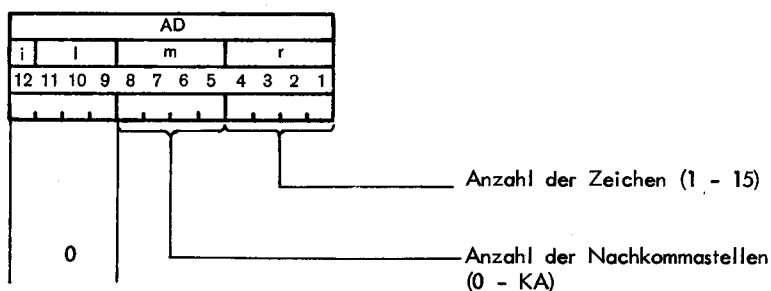
	Druckvorbefehle (edition following)	3,6
--	-------------------------------------	-----

AD											
i	l	m				r					
12	11	10	9	8	7	6	5	4	3	2	1

Mindestanzahl der auszugebenden Zeichen vor dem Komma (0 - 15)

	0	EDF	Druckvorbefehl für Drucker 1 Drucken Inhalt Druckspeicherwort D1
	2	{ EDFS	Druckvorbefehl für Drucker 2 Drucken Inhalt Druckspeicherwort D2 }
	+0	,ZERO	Vomullen
	+1	,STAR	Sicherungssterne
	+0	,BLACK	Druck in schwarz
	+4	,REDL	Druck in schwarz, wenn $(D) \geq 0$ Druck in rot, wenn $(D) < 0$
	+8	,RED	Druck in rot
	+12	,REDG	Druck in rot, wenn $(D) \geq 0$ Druck in schwarz, wenn $(D) < 0$
	0		ohne anschließendes Kennzeichen ,FTS
	1	,MIN	Leertaste, wenn $(D) \geq 0$ Kennzeichen -, wenn $(D) < 0$
	3	,SGN	Kennzeichen +, wenn $(D) \geq 0$ Kennzeichen -, wenn $(D) < 0$
	5	,ITS	Kennzeichen \diamond , wenn $(D) \geq 0$ Kennzeichen $\underline{\diamond}$, wenn $(D) < 0$
	7		Kennzeichen \ast , wenn $(D) \geq 0$ Kennzeichen $\underline{\ast}$, wenn $(D) < 0$

PBA, PC	PCT-Bereich und Speicherwort A (punch buffer and A)	3.7
---------	--	-----



Wirkung:

1. Es findet ein Transport von numerischen Zeichen zwischen den PCT-Bereichen und Speicherwort A statt, wobei die Umcodierung nach dem Lochkartencode 1 aus 10 erfolgt. Die Richtung des Transports (vom Eingabebereich nach A bzw. von Speicherwort A in den Ausgabebereich) und die Befehlsbehandlung von Überlochinformationen legt der Befehl "PCF Lochkartenvorbefehl (3.8)" fest. Die Anfangsstelle im Ein- bzw. Ausgabebereich, ab der der Transport erfolgt, bestimmt der Inhalt des dazugehörigen Bereichszeigers, der nach Ausführung des Befehls um die Anzahl der transportierten Zeichen erhöht ist.

2. In AD_r steht die Anzahl der Zeichen, die transportiert werden sollen.

Beim Transport vom Eingabebereich nach Speicherwort A kommt davon die in AD_m angegebene Anzahl von Zeichen hinter das Komma, während die restlichen $AD_r - AD_m$ Zeichen vor dem Komma stehen. Die übrigen Stellen des Speicherwortes werden gelöscht, das Vorzeichen ist durch die Überlochverarbeitung festgelegt.

Beim Transport von Speicherwort A in den Ausgabebereich wird analog die in AD_m angegebene Anzahl von Zeichen hinter dem Komma, die restlichen $AD_r - AD_m$ Zeichen vor dem Komma dem Speicherwort entnommen. Der Inhalt von Speicherwort A bleibt unverändert.

Der Transport geschieht in allen Fällen im Speicherwort A von links nach rechts.

PBA, PC	PCT-Bereich und Speicherwort A (punch buffer and A)	3.7
---------	--	-----

3. Beim Transport vom Eingabebereich nach Speicherwort A werden die Merker ML, MU und MC folgendermaßen gesetzt:

Die Stellung der Merker ML und MU ist durch die Verarbeitung der Überlochinformation festgelegt. Ein falsches 11er-Überloch hat ML = 1 ein falsches 12er-Überloch hat MU = 1 zur Folge. Andernfalls ist ML = 0 bzw. MU = 0.

Tritt ein nichtnumerisches Zeichen auf, wird der Merker MC = 1 gesetzt (andernfalls MC = 0) und der Transport abgebrochen. Im Speicherwort A stehen nur die vor dem unzulässigen Zeichen transportierten Zeichen. Der Bereichszeiger zeigt auf das unzulässige Zeichen.

PBA, PC	PCT-Bereich und Speicherwort A (punch buffer and A)	3.7
---------	--	-----

4. Beim Transport von Speicherwort A in den Ausgabebereich bleiben die Merker ML und MU unverändert. Es wird der Merker MC = 1 gesetzt und die Operation abgebrochen, wenn in einer zu transportierenden Kernspeicherstelle ein nichtnumerisches Zeichen steht. Im Bereich befinden sich nur die vor dem unzulässigen Zeichen transportierten Zeichen. Der Bereichszeiger zeigt auf das dem zuletzt transportierten Zeichen folgende Zeichen. Sind beim Transport nur numerische Zeichen aufgetreten, so ist der Merker MC = 0 gesetzt.

Beispiel: (KA 5)

Der PCT-Ausgabebereichszeiger steht auf Spalte 41. Es sollen 5 Vorkommastellen und 3 Nachkommastellen von Speicherwort A in den Ausgabebereich transportiert werden. Ferner ist Überloch 11 zu setzen und Überloch 12 genau dann, wenn das Vorzeichen von Speicherwort A positiv ist. Die Überlöcher sollen in der Spalte stehen, in der die erste Nachkommastelle gelocht wird.

vorher: SW 3 - 0 0 0 0 0 0 1 0 3 9 1 2 5 7 0 4

BW-Adresse	Op. Teil	1	Adreßteil	Symb. Adr.	Bemerkungen
	0 3 8	0	6 7 11		PCF, PN, 7.11
	1 3 7	0	0 3 8		PBA, PC, 3.8
	2				

nachher: SW 3 - 0 0 0 0 0 0 1 0 3 9 1 2 5 7 0 4

MC = 0

entspricht den Lochkartenspalten 41 42 43 44 45 46 47 48
mit Ü11 = 1, Ü12 = 0 in Spalte 46.

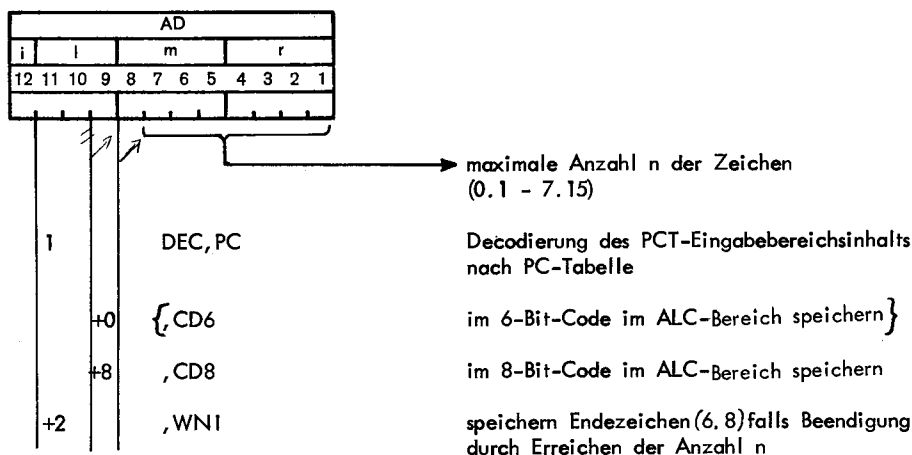
Nach Ausführung des Transports steht der Ausgabebereichszeiger auf Spalte 49 (MC = 0 da nur numerische Zeichen auftraten).

PBA, PC	PCT-Bereich und Speicherwort A (punch buffer and A)	3.7
---------	--	-----

Bemerkungen:

1. Das Betriebsprogramm stellt nicht fest, wenn durch eine falsche Befehlsfolge der festgelegte Bereich überschritten wird. Desgleichen entstehen undefinierbare Ergebnisse, wenn $AD_r = 0$ oder $AD_m > KA$ gesetzt werden.
2. Numerische Daten werden auf der Lochkarte im 1 aus 10 Code dargestellt, der intern erzeugt und verarbeitet wird. Bei nur numerischer Verarbeitung von Lochkarten ist also keine Lochkartencode-Tabelle erforderlich.
3. Der Befehl PBA.PC kann nur angewendet werden, wenn das Modul PC vorhanden ist. (Modell 820/15).

DEC, PC	Decodierung des PCT-Bereichsinhalts nach PC-Tabelle	3.7
---------	---	-----



Wirkung:

1. Der Inhalt des PCT-Eingabebereichs wird anhand der Lochkartencode-Tabelle (PC-Tabelle) in den ALC-Code umgewandelt und im ALC-Bereich gespeichert. Die Anfangsstelle im PCT-Bereich legt der dazugehörige Eingabebereichszeiger, die Anfangsstelle im ALC-Bereich legt der Bereichszeiger I_3 fest.
 2. Im ALC-Bereich können die Zeichen {im 6-Bit- oder} im 8-Bit-Code gespeichert werden. {Beim 6-Bit-Code wird vor einem Endezeichen 6.8, 6.11, 6.12 das Umschaltzeichen 3.15 eingeschoben, um die Umschaltung auf den 8-Bit-Code zu bewirken.}
 3. Beim Modell 820/15 können die Daten nur im 8 Bit-Code im ALC-Bereich stehen. Deshalb ist der Abruf "CD6" nicht möglich.
 4. Auf der Lochkarte gibt es die drei Endezeichen 6.8, 6.11, 6.12, die dort im entsprechenden PC-Code dargestellt werden. Ein vorheriges Umschaltzeichen 3.15 ist nicht erforderlich.
 5. Bei $n \neq 0$: Tritt unter den ersten n decodierten Zeichen eines der Zeichen 3.15, 6.8, 6.11, 6.12 auf, so wird die Operation nach der Abspeicherung dieses Zeichens beendet. Andernfalls findet nach der Decodierung von n Zeichen ein Abbruch statt. In diesem Fall kann gewählt werden, ob im Anschluß zusätzlich das Endezeichen 6.8 abgespeichert werden soll.
- Bei $n = 0$: Die Operation wird durch das Auftreten eines der Zeichen 3.15, 6.8, 6.11, 6.12 beendet.

DEC, PC	Decodierung des PCT-Pufferinhalts nach PC-Tabelle	3.7
---------	---	-----

6. Die Bereichszeiger werden während der Decodierung weitergestellt. Nach Ausführung des Befehls zeigt der PCT-Bereichszeiger auf das dem zuletzt decodierten Zeichen folgende Zeichen und der ALC-Bereichszeiger auf die nächste freie Stelle im ALC-Bereich.
7. Es wird der Merker MU = 1 gesetzt, wenn die Decodierung durch Erreichen der Anzahl n beendet wurde. Kam jedoch ein Endezeichen vor, so ist MU = 0.
- Ein Zeichen ist nicht definiert, wenn es im Fall des 6-Bit-Codes unter den ersten 64, im Fall des 8-Bit-Codes unter den ersten 109 Zeichen der PC-Tabelle nicht auftritt. Bei einem nicht definierten Zeichen wird der Merker MC = 1 gesetzt (andernfalls MC = 0) und die Operation abgebrochen. Der PCT-Bereichszeiger zeigt auf das dem unzulässigen Zeichen folgende Zeichen, der ALC-Bereichszeiger auf die dem zuletzt gespeicherten, definierten Zeichen folgende Kernspeicherstelle.

Beispiel :

Es sollen maximal 24 Zeichen des PCT-Eingabebereichs nach der PC-Tabelle decodiert und im 8-Bit-Code im ALC-Bereich gespeichert werden. Der PCT-Bereichszeiger steht auf 15, der ALC-Bereichszeiger auf

vorher: I_3 2 5 13 entspricht ~~SW 32~~ Stelle 13
~~SW 33~~

BW-Adresse			Op. Teil		Adreßteil				Symb. Adr.	Bemerkungen
0	1	0	3	7	0	3	9	8		DEC, PC, CD8, WNI, 24

Unter den ersten 24 Zeichen trat kein Endezeichen auf. Anschließend steht der PCT-Bereichszeiger auf 39 und der ALC-Bereichszeiger auf

nachher: I_3 2 8 15 entspricht ~~SW 33~~ Stelle 15
~~SW 38~~

Im ALC-Bereich ist

SW 33	SW 38	x	x	x	x	x	x	x	x	x	x	x	x	x	x	6	8
------------------	------------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

und die Merker sind MU = 1, MC = 0.

DEC, PC	Decodierung des PCT-Bereichsinhalts (nach PC-Tabelle)	3.7
---------	--	-----

Bemerkungen:

1. Das Betriebsprogramm stellt nicht fest, wenn durch eine falsche Befehlsfolge der festgelegte Bereich überschritten wird.
2. Man beachte, daß der Bereichszeiger I_3 in I_{1m} immer die interne Speicherwortnummer enthält, die aus der normalen Speicherwortadresse durch Addition von $\{0.5\}$ bzw. $[0.4]$ entsteht.
3. Wird der Text im 6-Bit-Code gespeichert, so steht der Puffzeiger I_3 nur bei jedem zweiten Zeichen richtig. Darauf ist besonders dann zu achten, wenn die Operation durch Erreichen der Anzahl n beendet wurde.
4. Der Anfang der PC-Tabelle ist im AD_{1m}^1 -Teil des Befehls unter Adresse 0.0.2 im Festspeicherblock 0 anzugeben. Die PC-Tabelle umfaßt 109 Zeichen und muß sich im Festspeicherblock 0 befinden.
5. Beim Modell 820/15 kann dieser Befehl nur angewendet werden, wenn der Modul PC vorhanden ist.

	PCT-Bereich und Indexregister	3.7
--	-------------------------------	-----

AD											
i			l			m			r		
12	11	10	9	8	7	6	5	4	3	2	1
2			0			0					
									1		

RDX, PC Transport vom PCT-Eingabebereich
nach Indexregister
(read buffer to indexregister)

XPB, PC Transport vom Indexregister
nach PCT-Ausgabebereich
(indexregister to punch buffer)

Wirkung:

1. Es wird ein alphanumerisches Zeichen aus dem PCT-Eingabebereich in ein Indexregister bzw. aus einem Indexregister in den PCT-Ausgabebereich transportiert, wobei die Umcodierung nach der Lochkartencode-Tabelle (PC-Tabelle) erfolgt. Die Stelle im Ein- bzw. Ausgabebereich bestimmt der Inhalt des dazugehörigen Bereichszeigers, der nach Ausführung des Befehls um 1 erhöht ist.
2. Beim Transport vom Eingabebereich in ein Indexregister wird das Zeichen zur Decodierung mit 109 Zeichen der Lochkartencode-Tabelle verglichen, um den ALC-Code des Zeichens zu erhalten. Ist keine Übereinstimmung mit einem Zeichen der Tabelle festzustellen, wird Merker MC = 1 andernfalls MC = 0 gesetzt. Der Inhalt des Indexregisters ist im Falle eines nicht definierten Zeichens unverändert und ein gegebener Indexvorbefehl nicht gelöscht.

Beim Transport vom Indexregister in den Ausgabebereich bleiben die Merkerstellungen unverändert.

	PCT-Bereich und Indexregister	3.7
--	-------------------------------	-----

Beispiel:

Der im Indexregister I_1 stehende ALC-Code eines Zeichens soll nach der Lochkartencode-Tabelle umgewandelt und im PCT-Ausgabebereich (Bereichsanfang 23) gespeichert werden. Der Ausgabebereichszeiger steht auf 61.

vorher: I_1 0 3 4 entspricht ALC-Code "="

BW-Adresse	Op. Teil		I	Adreßteil				Symb. Adr.	Bemerkungen
0	2	2	0	0	0	0	1		XF, 1
1	3	7	0	2	0	0	1		XPN, PC

nachher: I_1 0 3 4

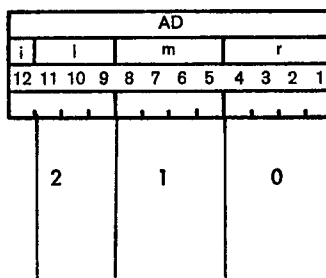
SW 30 8 6

Der Ausgabebereichszeiger steht anschließend auf 62.

Bemerkungen:

1. Das Betriebsprogramm stellt nicht fest, wenn durch eine falsche Befehlsfolge der festgelegte Bereich überschritten wird.
2. Der Anfang der PC-Tabelle ist im AD_{Imr} -Teil des Befehls unter Adresse 0.0.2 im Anwenderprogrammblock 0 anzugeben. Die PC-Tabelle umfaßt 109 Zeichen und muß sich im Anwenderprogrammblock 0 befinden.
3. Man beachte den Befehl "XF und XFR Indexvorbefehle (2.2)".

XBUF	ALC-Bereich und Indexregister (indexregister to buffer)	3.7
------	--	-----



XBUF Transport vom Indexregister
in den ALC-Bereich

Wirkung:

1. Der Befehl XBUF bewirkt einen Transport aus $I_{l,m,r}$ in die durch den Stand des ALC-Bereichszeigers definierte Kernspeicherstelle. Falls diese Kernspeicherstelle nur 4 Bit tief ist, wird nur der Inhalt von I_l dort abgestellt. Wird ein Kernspeicher vom Typ 164 mit 12 Bit tiefen Kernspeicherstellen als Datenspeicher verwendet, so werden alle 11 Bit des Indexregisters in einer Kernspeicherstelle gespeichert. Bei Kernspeichern vom Typ 166 mit 6 Bit tiefen Kernspeicherstellen werden nur Bit 1-6 des Indexregisters gespeichert.

2. Der ALC-Bereichszeiger ist nach Ausführung dieses Befehls um 1 erhöht.

Bemerkung:

1. Man beachte die unterschiedliche Wirkung von XBUF gegenüber XBUF1 und XBUF2.

2. Der Befehl XBUF kann für die Aufbereitung eines Ausgabebereichs zum Stanzen von Lochkarten in einem wählbaren Code verwendet werden.

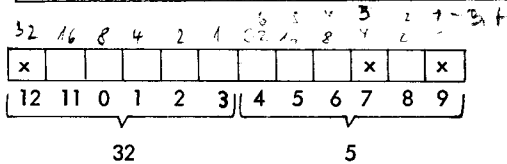
3. Beim NIXDORF-Lochkartenstanzer und beim IBM 024/026 dürfen aus elektro-mechanischen Gründen höchstens 5 Löcher pro Kartenspalte gestanzt werden.

XBUF	ALC-Bereich und Indexregister	3.7
------	-------------------------------	-----

Beispiel:

Aufbereitung des Ausgabebereichs zum Stanzen von Lochkarten in einem frei wählbaren Code. In die 12 Bit der 42. Spalte soll der Lochkarten-Code 12-7-9 abgestellt werden. Speicherwort 20 soll das erste Speicherwort des Ausgabebereichs sein.

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
0	3 11	0	1 9 2		BPX,25.2
1	2 2	0	0 12 2		XFR,2
2	2 10	0	0 2 0		CX,32
3	3 7	0	2 1 0		XBUF
4	2 10	0	0 0 5		CX,5
5	3 7	0	2 1 0		XBUF



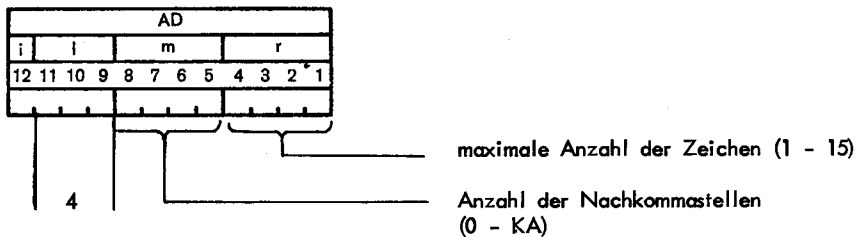
Spalte 42

Nach Ausführung dieser Befehlsfolge zeigt der ALC-Bereichszeiger in Speicherwort 25 auf Stelle 4.

Bemerkung:

Dieser Befehl ist im Befehlsvorrat des Modells 820/15 nicht enthalten.

PBA, PT	PCT-Bereich nach Speicherwort A vom PT-Code (punch buffer and A)	3.7
---------	--	-----

Wirkung:

1. Es findet ein Transport von numerischen Zeichen aus dem PCT-Eingabebereich nach Speicherwort A statt, wobei die Umcodierung nach der Lochstreifen-code-Tabelle (PT-Tabelle) erfolgt. Die Anfangsstelle im Bereich bestimmt der Inhalt des dazugehörigen Eingabebereichszeigers, der nach Ausführung des Befehls um die Anzahl der verarbeiteten Zeichen erhöht ist.
2. Maximal wird nach A die in AD_r angegebene Anzahl von Ziffern transportiert. Eine frühere Beendigung der Operation findet in folgenden Fällen statt:
 - a) Es trat ein Zeichen verschieden von Ziffern, Komma, Stern oder Leertaste auf.
 - b) Nach einer Ziffer trat ein Zeichen verschieden von Ziffern und Komma auf.
 - c) Nach einem Komma trat ein Zeichen verschieden von Ziffern auf.

Der ALC-Code des Zeichens, das die Beendigung bewirkt, wird in einem Indexregister gespeichert.

3. Das Speicherwort A wird gelöscht und erhält ein positives Vorzeichen. Anschließend werden die Zeichen aus dem Eingabebereich nach Speicherwort A gebracht. Einem Sicherungstern oder einer Leertaste entspricht dabei der Wert 0; ein Komma und ein die Beendigung bewirkendes Zeichen werden nicht im Speicherwort gespeichert.

Von den transportierten Zeichen kommt die durch AD_m angegebene Anzahl im Speicherwort A hinter das Komma, wenn unter den m Zeichen kein Komma-Zeichen auftrat; andernfalls hat dieses Vorrang und AD_m wird nicht beachtet.

Wird die Beendigung durch eines der Zeichen -, ∅ oder * bewirkt, so erhält Speicherwort A ein negatives Vorzeichen.

PBA, PT	PCT-Bereich nach Speicherwort A vom PT-Code (punch buffer and A)	3.7
---------	--	-----

4. In die Stellen 12 bis 15 des Speicherwortes 4 werden zur Kontrolle die Anzahl der transportierten Ziffern entsprechend ihrer Stellung im Speicherwort A gespeichert, und zwar kommt die Anzahl der Nachkommastellen sedezimal in die Stelle 15, die Anzahl der Vorkommastellen (einschließlich Leertasten und Sicherungsterne) sedezimal in die Stelle 13. Die Stellen 12 und 14 werden gelöscht, während alle weiteren Stellen von Speicherwort 4 unverändert bleiben.
5. Die Merker MU und MC werden folgendermaßen gesetzt:

War die Anzahl der transportierten Ziffern gleich AD_r , so ist $MU = 0$,
war sie kleiner als AD_r , ist $MU = 1$.

Jedes aus dem Bereich gelesene Zeichen wird zur Decodierung mit maximal 109 Zeichen der Lochstreifencode-Tabelle verglichen, um den ALC-Code des Zeichens zu erhalten. Ist keine Übereinstimmung mit einem Zeichen der Tabelle festzustellen, wird Merker $MC = 1$ (andernfalls $MC = 0$) gesetzt. Der Inhalt des Indexregisters ist im Falle eines nicht definierten Zeichens unverändert und ein gegebener Indexvorbefehl nicht gelöscht. Im Speicherwort A stehen linksbündig die vor dem unzulässigen Zeichen transportierten Ziffern, der Eingabebereichs-zeiger zeigt auf das unzulässige Zeichen.

Werden nach einem Komma mehr Nachkommastellen gelesen als die KA zuläßt, so gehen die überzähligen Ziffern verloren. Sind von 0 verschiedene Ziffern verlorengegangen, wird Merker $MC = 1$ andernfalls $MC = 0$ gesetzt.
6. Durch den Befehl "PTF Lochstreifenvorbefehl (3.5)" ist festzulegen, ob die Umcodierung des Lochstreifencodes in den ALC-Code nach Tabelle 1 oder Tabelle 2 erfolgen soll. Ohne Vorbefehl wird nach Tabelle 1 decodiert.
7. Die Abfrage nach Ziffern, Komma, Stern und Leertaste erfolgt nach der Umcodierung in den ALC-Code. Es wird dabei für Komma 4.0, für Stern 1.0 und für Leertaste 0.10 genommen. Es ist also zu beachten, daß die PT-Codes für normales Komma und Dezimalkomma unterschiedlich sein müssen. Im Gegensatz zur Eingabe-Verarbeitung erfolgt bei der Ausgabe von Dezimalzahlen wahlweise das Lochen des PT-Codes von 5.14 oder 5.15, je nachdem ob beim Druck ein Dezimalkomma oder ein Dezimalpunkt geschrieben werden soll. Es ist also unter 4.0, 5.14 und 5.15 derselbe Lochstreifencode anzugeben.
8. Mit diesem Befehl können auch Daten, die im Eingabebereich der Magnetband-cassette liegen, abgerufen werden. Es ist dabei die Aufteilung der Lochstreifen-Code-Tabelle zu beachten.
9. Bei dem Modell 820/15 muß das Modul "PT" vorhanden sein. Ansonsten wird dieser Befehl nicht ausgeführt und das Programm läuft auf Intern-Fehlerstop.

PBA, PT	PCT-Bereich nach Speicherwort A vom PT-Code (punch buffer and A)	3.7
---------	--	-----

Beispiel: (KA 5)

Der im Beispiel zum Befehl "EDPT Stanzen Inhalt Druckspeicherwort (3,5)" gestanzte Lochstreifen wurde eingelesen. Das dort angegebene Zahlenfeld steht im Bereich ab Zeichen Nummer 12 und soll nach Speicherwort A gebracht werden, wobei nach Tabelle 2 zu decodieren ist.

BW-Adresse			Op. Teil			Adreßteil			Symb. Adr.	Bemerkungen
		0	3	5	0	0	10	0		PTF, TBL2
		1	2	2	0	0	0	2		XF, 2
		2	2	11	0	1	0	12		PBP, RD, 12
		3	3	7	0	4	4	15		PBA, PT, 4.15

Es treten folgende Zeichen der Reihe nach auf * * * * 218, 503 ◊

nachher: 1_2

0	0	15
---	---	----

 entspricht ALC-Code von ◊

MU = 1 SW 3

1	0	0	0	0	0	0	0	0	2	1	8	5	0	3	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

MC = 0

													0	7	0	3
--	--	--	--	--	--	--	--	--	--	--	--	--	---	---	---	---

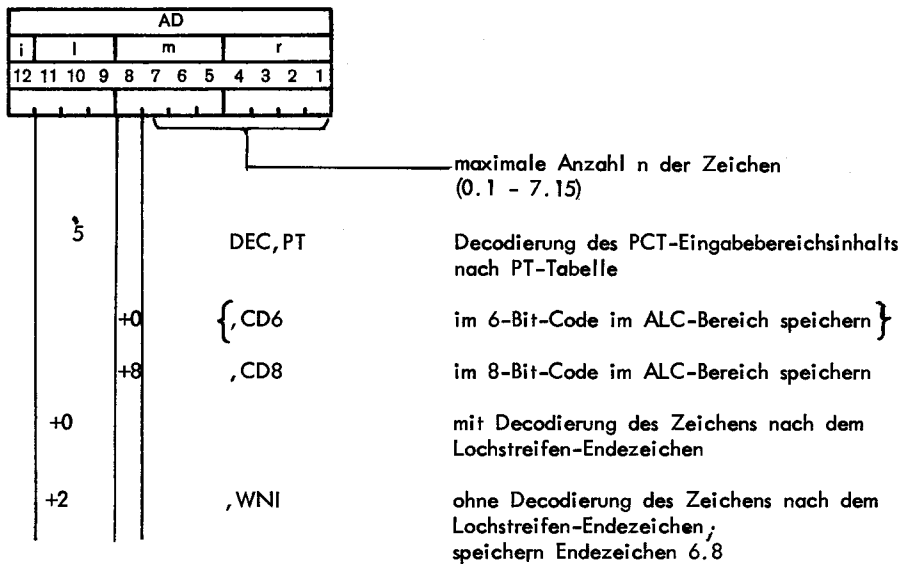
Anschließend steht die Zahl 24 im PCT-Eingabebereichszeiger.

PBA, PT	PCT-Bereich nach Speicherwort A vom PT-Code (punch buffer and A)	3.7
---------	---	-----

Bemerkungen:

1. Das Betriebsprogramm stellt nicht fest, wenn durch eine falsche Befehlsfolge der festgelegte Bereich überschritten wird. Desgleichen entstehen undefinierte Ergebnisse, wenn $AD_r = 0$ oder $AD_m > KA$ gesetzt werden.
2. Jede Lochstreifencode-Tabelle umfaßt 109 Zeichen und muß sich im Festspeicherblock 0 befinden. Die Anfangsadresse von Tabelle 1 ist in AD_{lmr} des Befehls unter Adresse 0.0.3, die von Tabelle 2 ist in AD_{lmr} des Befehls unter Adresse 0.0.4 im Anwenderprogrammblock 0 anzugeben.
3. Man beachte den Befehl "XF und XFR Indexvorbefehle (2.2)".

DEC, PT	Decodierung des PCT-Bereichsinhalts nach PT-Tabelle (decoding)	3.7
---------	--	-----

Wirkung:

- Der Inhalt des PCT-Eingabebereichs wird anhand einer Lochstreifencode-Tabelle (PT-Tabelle) in den ALC-Code umgewandelt und im ALC-Bereich gespeichert. Die Anfangsstelle im PCT-Bereich legt der dazugehörige Eingabebereichszeiger, die Anfangsstelle im ALC-Bereich legt der Bereichszeiger I_3 fest.
- Durch den Befehl "PTF Lochstreifenvorbefehl (3.5)" ist festzulegen, ob die Decodierung des Lochstreifencodes in den ALC-Code nach Tabelle 1 oder Tabelle 2 erfolgen soll. Ohne Vorbefehl wird nach Tabelle 1 decodiert.
- Im ALC-Bereich können die Zeichen im 6-Bit- oder im 8-Bit-Code gespeichert werden. (Beim 6-Bit-Code wird vor einem Endezeichen 6.8, 6.11, 6.12 das Umschaltzeichen 3.15 gespeichert, um die Umschaltung auf den 8-Bit-Code des Endezeichens zu bewirken. Beim 8-Bit-Code entfällt dieses Zeichen.)
- Beim Modell 820/15 können die Daten nur im 8-Bit-Code gespeichert sein. Deshalb ist der Zusatz "CD6" nicht im Befehlsvorrat enthalten.

DEC.PT	Decodierung des PCT-Bereichinhalts nach PT-Tabelle	3.7
--------	--	-----

5. Auf dem Lochstreifen gibt es ein Endezeichen, dem im ALC-Code das Ende- und Umschaltzeichen 3.15 entspricht. Eine Unterscheidung kann durch ein Zeichen nach dem Endezeichen vorgenommen werden, wofür die Codes 6.8, 6.11, 6.12 zur Verfügung stehen.
6. Bei $n \neq 0$: Tritt unter den ersten n decodierten Zeichen das Endezeichen des Lochstreifens auf (d.h. PT-Code für 3.15), so stoppt die Operation. Es kann gewählt werden, ob das dem Endezeichen folgende Zeichen ebenfalls decodiert und gespeichert wird oder unabhängig davon der Code 6.8 gespeichert werden soll. Tritt das Endezeichen nicht auf, so bricht die Operation nach der Decodierung des n -ten Zeichens ab. Es wird kein weiteres Zeichen mehr gespeichert.

Bei $n = 0$: Die Operation wird nur durch das Auftreten des Zeichens 3.15 beendet.

7. Die Bereichszeiger werden während der Decodierung weitergestellt. Nach Ausführung des Befehls zeigt der PCT-Bereichszeiger auf das dem zuletzt decodierten Zeichen folgende Zeichen und der ALC-Bereichszeiger auf die nächste freie Stelle im ALC-Bereich.
8. Es wird der Merker $MU = 1$ gesetzt, wenn n Zeichen decodiert wurden und das Endezeichen dabei nicht auftrat. Kam das Endezeichen unter den ersten n Zeichen vor, so ist $MU = 0$.

Ein Zeichen ist nicht definiert, wenn es im Fall des 6-Bit-Codes unter den ersten 64, im Fall des 8-Bit-Codes unter den ersten 109 Zeichen der PT-Tabelle nicht auftritt. Bei einem nicht definierten Zeichen wird der Merker $MC = 1$ gesetzt (andernfalls $MC = 0$) und die Operation abgebrochen. Der PCT-Bereichszeiger zeigt auf das dem unzulässigen Zeichen folgende Zeichen, der ALC-Bereichszeiger auf die dem zuletzt gespeicherten definierten Zeichen folgende Kernspeicherstelle.

DEC, PT	Decodierung des PCT-Bereichsinhalts nach PT-Tabelle	3.7
---------	---	-----

Beispiel:

Es sollen maximal 20 Zeichen des PCT-Eingabebereichs nach Lochstreifencode-Tabelle 1 decodiert und im 6-Bit-Code im ALC-Bereich gespeichert werden. Der PCT-Bereichszeiger steht auf 73, der ALC-Bereichszeiger auf

I₃ 2 7 14 entspricht SW 34, Stelle 14

BW-Adresse	Op. Teil	l	Adreßteil				Symb. Adr.	Bemerkungen	
	0	3	5	0	0	2	0		PTF, TBL1
	1	3	7	0	7	1	4		DEC, PT, CD6, WNI, 20
	2								

Als 9. Zeichen trat das Endezeichen auf. Anschließend steht der PCT-Bereichs-zeiger auf 82 und der ALC-Bereichszeiger auf

I₃ 2 8 14 entspricht SW 35, Stelle 14

Im ALC-Bereich ist

SW 34 x x

SW 35 x x x x x x x x x x 15 12 6 8

und die Merker sind MU = 0, MC = 0.

DEC, PT	Decodierung des PCT-Bereichsinhalts nach PT-Tabelle	3.7
---------	--	-----

Bemerkungen:

1. Das Betriebsprogramm stellt nicht fest, wenn durch eine falsche Befehlsfolge der festgelegte Bereich überschritten wird.

2. Man beachte, daß der Bereichszeiger I_3 in I_{lm} immer die interne Speicherwortnummer enthält, die aus der Speicherwortnummer durch Addition von $\{0.3\}$ bzw. $\{0.4\}$ entsteht.

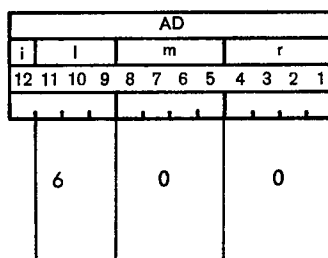
$\left\{ \begin{array}{l} 3. \text{ Wird der Text im 6-Bit-Code gespeichert, so steht der Bereichszeiger } I_3 \\ \text{ nur bei jedem zweiten Zeichen richtig. Darauf ist besonders dann zu} \\ \text{ achten, wenn die Operation durch Erreichen der Anzahl } n \text{ beendet wurde.} \end{array} \right\}$

4. Jede Lochstreifencode-Tabelle umfaßt 109 Zeichen und muß sich im Anwenderprogrammblock 0 befinden. Die Anfangsadresse von Tabelle 1 ist im $AD_{l_{mr}}$ -Teil des Befehls unter Adresse 0.0.3, die von Tabelle 2 im $AD_{l_{mr}}$ -Teil des Befehls unter Adresse 0.0.4 im Anwenderprogrammblock 0 anzugeben.

5. Dieser Befehl dient auch zur Decodierung von Daten aus dem Ein-/Ausgabebereich der Magnetbandcassette. Es ist dabei die Aufbereitung der Lochstreifencodetabelle zu beachten.

$\left[\begin{array}{l} 6. \text{ Beim Modell 820/15 ist dieser Befehl nur in Verbindung mit dem Modul PT möglich.} \end{array} \right]$

RDX, PT	PCT-Bereich und Indexregister (read buffer to indexregister)	3.7
---------	---	-----



RDX, PT

Transport mit PT-Decodierung vom PCT-Eingabebereich nach Indexregister

Wirkung:

1. Es wird ein alphanumerisches Zeichen aus dem PCT-Eingabebereich in ein Indexregister transportiert, wobei eine Decodierung nach einer Lochstreifen-code-Tabelle (PT-Tabelle) erfolgt. Die Stelle im Eingabebereich bestimmt der Inhalt des dazu gehörigen Eingabezeigers, der nach Ausführung des Befehls um 1 erhöht ist.
 2. Durch den Befehl "PTF Lochstreifen-vorbefehl (3.5)" ist festzulegen, ob die Decodierung des Lochstreifencodes in den ALC-Code nach Tabelle 1 oder Tabelle 2 erfolgen soll. Ohne Vorbefehl wird nach Tabelle 1 decodiert.
 3. Zur Decodierung wird das Zeichen mit 109 Zeichen der Lochstreifen-code-Tabelle verglichen, um den ALC-Code des Zeichens zu erhalten. Ist keine Übereinstimmung mit einem Code der Tabelle festzustellen, wird Merker MC = 1 (andernfalls MC = 0) gesetzt. Der Inhalt des Indexregisters ist im Falle eines nicht definierten Zeichens unverändert und ein gegebener Indexvorbefehl nicht gelöscht.
 4. Dieser Befehl kann auch zum Transport vom Cassetten Ein-/Ausgabebereich nach I verwendet werden. Es ist hierbei die Lochstreifen-code-Tabelle zu verwenden.
- [5. Beim Modell 820/15 sind die Moduln TC und PT bei Anwendung dieses Befehls notwendig, sobald mit Cassette gearbeitet wird. Ansonsten genügt der Modul PT.]

RDX,PT	PCT-Bereich und Indexregister (read buffer to indexregister)	3.7
--------	---	-----

Beispiel:

Aus dem PCT-Bereich ist ein Zeichen nach PT-Tabelle 2 zu decodieren und nach Indexregister I_3 zu bringen. Der Eingabebereichszeiger enthält die Zahl 93.

BW-Adresse	Op. Teil			Adreßteil				Symb. Adr.	Bemerkungen
	0	3	5	0	0	10	0		PTF, TBL2
	1	2	2	0	0	0	3		XF, 3
	2	3	7	0	6	0	0		RDX, PT
	3								

Im Bereich steht der PT-Code des Zeichens %. Anschließend ist

I_3

0	3	11
---	---	----

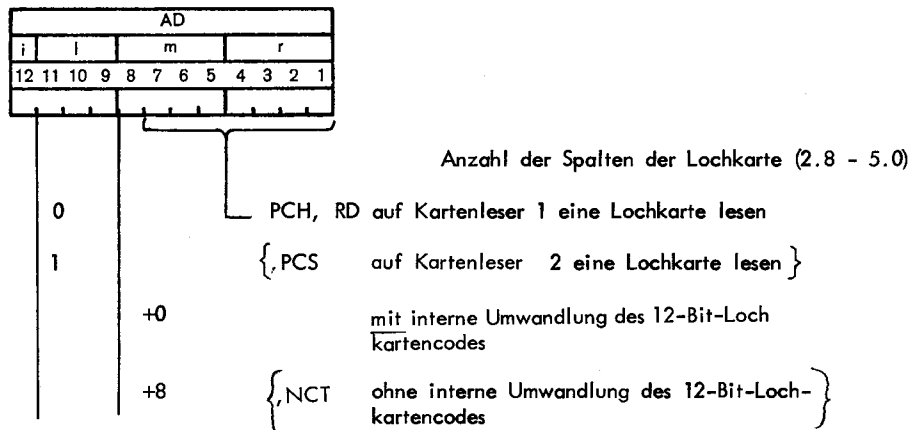
, dies entspricht dem ALC-Code des Zeichens %.

Der Eingabebereichszeiger steht auf 94 (MC = 0).

Bemerkungen:

1. Das Betriebsprogramm stellt nicht fest, wenn durch eine falsche Befehlsfolge der Bereich überschritten wird.
2. Jede Lochstreifencode-Tabelle umfaßt 109 Zeichen und muß sich im Anwenderprogrammblock 0 befinden. Die Anfangsadresse von Tabelle 1 ist im AD_{Imr} - Teil des Befehls unter Adresse 0.0.3 die von Tabelle 2 im AD_{Imr} - Teil des Befehls unter Adresse 0.0.4 im Anwenderprogrammblock 0 anzugeben.
3. Man beachte den Befehl "XF und XFR Indexvorbefehle (2.2)".

PCH, RD	Lochkarte lesen (read)	3.8
---------	------------------------	-----

Wirkung:

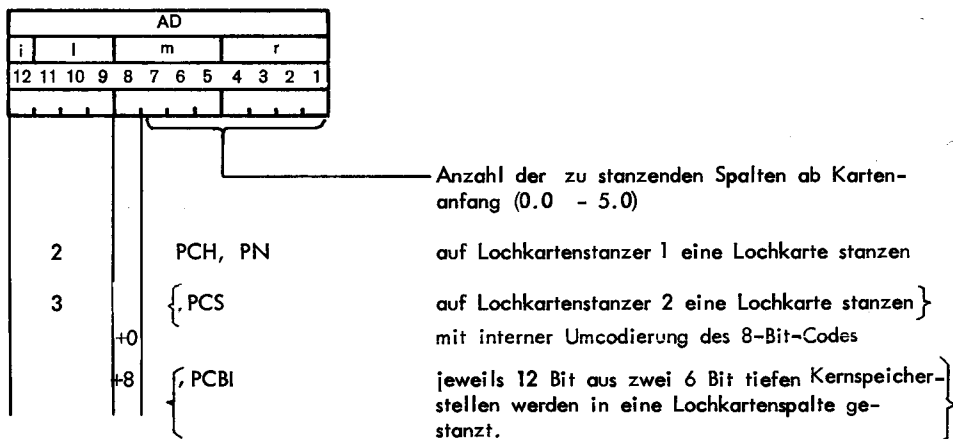
1. Auf dem gewählten Kartenleser wird eine vollständige Lochkarte gelesen. Die Gesamtzahl der Spalten, die die Karte besitzt, ist in AD Bit 1 bis Bit 7 anzugeben. Das Lesen geschieht mit einem fotoelektrischen^{mr} Leser, der 40- bis 80- spaltige Lochkarten der genormten Breite liest.
2. Die gelesene Information der Lochkarte wird im PCT-Eingabebereich, beginnend beim Bereichsanfang, gespeichert. Hierbei sind 2 Fälle zu unterscheiden:
 1. Bit 8 ist im Adressteil nicht gesetzt. Der 12-Bit-Lochkartencode wird intern in einen 8-Bit-Code umgewandelt, so daß jeder Lochkartenspalte zwei Kernspeicherstellen im Bereich entsprechen. Wegen der internen Umcodierung ist es nicht möglich, jeden beliebigen Lochkartencode zu lesen. Zulässig ist unter anderem der IBM-Code. Tritt in einer Spalte ein nicht lesbarer Code auf, so läuft das Programm auf Externfehlerstop. Nach Drücken der C-Taste wird intern mit dem gleichen Befehl eine neue Karte eingezogen.
 2. Bit 8 ist im Adressteil gesetzt. Der 12-Bit Lochkartencode wird in jeweils zwei 6 Bit tiefe Kernspeicherstellen pro Kartenspalte abgestellt. Hiermit ist die Möglichkeit gegeben, Lochkarten mit frei wählbaren Codes zu verwenden.

PCH, RD	Lochkarten lesen (read)	3.8
---------	-------------------------	-----

Bemerkungen:

1. Der Anfang des PCI-Eingabebereiches ist durch den Befehl "PBG, RD Bereichsanfang setzen (2.11)" zu definieren. Wird nacheinander in denselben Bereich eingelesen, so braucht der Befehl "PBG, RD (2.11)" nur einmal gegeben zu werden.
2. Das Programm läuft auf Datenträgerfehlerstopp, wenn sich im Leser keine Karte befindet, wenn sich eine Karte verklemmt hat oder wenn der Leser abgeschaltet ist. Nach Drücken der F-Taste wird erneut versucht, eine Karte zu lesen.
3. Beim Modell 820/15 kann der Befehl nur beim vorhandene n Modul PC ausgeführt werden.
4. Es ist hier grundsätzlich der Zusatz "PCS" nicht anwendbar, da kein zweiter Lochkartenleser angeschlossen werden kann.

PCH, PN	Lochkarte stanzen (punch)	3.8
---------	---------------------------	-----

Wirkung:

- Auf dem gewählten Lochkartenstanzer wird eine vollständige, 80-spaltige Lochkarte gestanzt. Die Anzahl der ab Kartenanfang zu lochenden Spalten ist in AD_{mr} Bit 1 bis Bit 7 anzugeben. Als Gerät kann entweder der IBM Kartenlocher 024/026 oder der NIXDORF-Lochkartenstanzer 091/092 angeschlossen werden.
- Die auszugebende Information wird dem PCT Ausgabebereich, beginnend beim Bereichsanfang, entnommen. Hierbei sind 2 Fälle zu unterscheiden.
 - Bit 8 ist im Adressteil nicht gesetzt. Jedes Zeichen ist in jeweils 2 Kernspeicherstellen im Bereich dargestellt. Vor der Ausgabe erfolgt eine interne Umcodierung des 8-Bit-Codes in den 12-Bit-Code der Lochkarte.
 - Bit 8 ist im Adressteil gesetzt. Die 12 Bit von zwei 6 Bit tiefen Kernspeicherstellen werden in eine Lochkartenspalte gestanzt. Hiermit ist die Möglichkeit gegeben, einen frei wählbaren Lochkartencode zu verwenden.
- Bei Anschluß des NIXDORF-Lochkartenstanzer findet während des Stanzens einer Lochkarte die Kontrolllesung der zuvor gestanzten Karte statt. Das Kontrollverfahren läuft folgendermaßen ab:
Bei jedem Stanzen einer Lochkarte muß das Prüfzeichen zur Kontrolllesung der zuvor gestanzten Karte im Indexregister I₂ stehen. Nach Beendigung der Operation ist in I₂ das neue Prüfzeichen der soeben gestanzten Karte gespeichert.

PCH,PN	Lochkarte stanzen (punch)	3.8
--------	---------------------------	-----

Aus der zur Kontrolle gelesenen Information wird das Prüfzeichen berechnet und mit dem in I_2 gespeicherten Prüfzeichen verglichen. Tritt dabei ein Unterschied auf, so folgt die Meldung "Fehler bei Kontrolllesung festgestellt".

4. Der Befehl "PCH,PN Lochkarte stanzen (3.8)" verändert nicht die Merker ML, MU, MC. Es werden lediglich die folgenden Merkerstellungen intern zwischengespeichert:

Wenn der Stanzvorgang fehlerhaft abgelaufen ist, läuft das Programm auf Datenträgerfehlerstopp. Nach Drücken der F-Taste fährt das Programm mit dem nächsten Befehl fort, und es ist $ML = 1$ zwischengespeichert. Andernfalls ist $ML = 0$.

Bei Anschluß des IBM Kartenlochers werden $MU = 0$ und $MC = 0$ gespeichert. Bei Anschluß des Nixdorf-Kartenlochers haben MU und MC folgende Funktionen: Wurde bei der Kontrolllesung ein Fehler festgestellt, so ist $MU = 1$ andernfalls $MU = 0$.

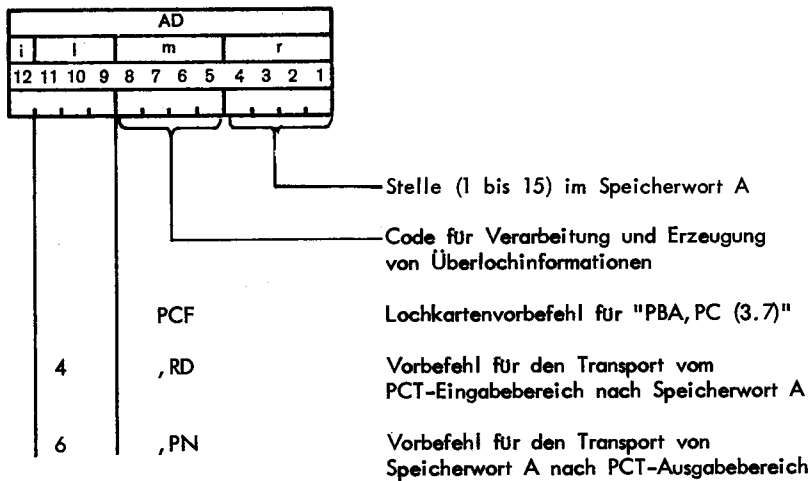
Lag eine Karte in Lesestation aber keine Karte in Stanzstation, so wird nur eine Kontrolllesung durchgeführt und die nächste Karte vorgelegt. In diesem Fall ist $MC = 1$, sonst $MC = 0$ gespeichert.

Im Befehl "SMWT Setze Merker nach Ende Simultan-Ausgabe (2.12)" wird auf das Ende der Ausgabe gewartet (Simultangerät 2). Im Anschluß daran erfolgt das Setzen der Merker entsprechend obiger Aufstellung.

Bemerkungen:

1. Ist der NIXDORF-Lochkartenstanzer angeschlossen und in AD_{mr} Bit 1 bis Bit 7 die Anzahl 0 codiert, so wird eine Leerkarte ausgegeben und die zuvor gestanzte Karte zur Kontrolle gelesen. In diesem Fall ist ebenfalls $MC = 1$ zwischengespeichert.
2. Der Befehl "PCH,PN Lochkarte stanzen (3.8)" wird simultan zum weiteren Befehlsablauf ausgeführt. Vor einer neuen Ausgabe auf Simultangerät 2 wartet intern das Betriebsprogramm auf die Beendigung der Operation. Man beachte jedoch, daß die Speicherworte des PCT-Ausgabebereichs erst nach Beendigung der Ausgabe verändert werden dürfen (vgl. "Warte- und Merkersetzbefehl (2.12)". Im Gegensatz dazu kann der Anfang des PCT-Ausgabebereichs sofort neu definiert werden.

PCF	Lochkartenvorbefehl	3.8
-----	---------------------	-----

Wirkung:

- Der Lochkartenvorbefehl legt fest, welche Verarbeitungsart beim nächsten Befehl "PBA, PC Transport zwischen PCT-Bereich und A (3.7)" durchzuführen ist. Nach der Ausführung dieses Befehls ist der Vorbefehl gelöscht.
- Beim Vorbefehl PCF, RD, d.h. $AD_l = 4$, erfolgt der Transport vom PCT - Eingabebereich nach Speicherwort A. Beim Vorbefehl PCF, PN, d.h. $AD_l = 6$, erfolgt der Transport von Speicherwort A in den PCT-Ausgabebereich.
- Jeder übertragenen Stelle von Speicherwort A entspricht beim numerischen Verarbeiten eine Spalte auf der Lochkarte, die im Bereich in zwei Kernspeicherstellen dargestellt ist. Werte werden im 11 aus 10 Code gelesen und gelocht, d.h., jeder Dezimalziffer entspricht eine Lochung in genau einer der Lochzeilen 0 bis 9. Lochungen in den Zeilen 11 und 12 heißen Überlöcher und dienen bei numerischen Daten zur Darstellung des Vorzeichens einer Dezimalzahl.
- Der in AD angegebene Stelle von Speicherwort A entspricht beim Transport eine bestimmte Spalte der eingelesenen bzw. auszugebenden Lochkarte.

PCF	Lochkartenvorbefehl	3.8
-----	---------------------	-----

Beim Transport vom PCT-Eingabebereich nach Speicherwort A werden in Abhängigkeit von den Überlöchern 11 und 12 dieser Spalte das Vorzeichen des Speicherwortes und die Merker ML, MU gesetzt.

Beim Transport von Speicherwort A in den PCT-Ausgabebereich werden in Abhängigkeit vom Vorzeichen des Speicherwortes die Überlöcher 11 und 12 in dieser Spalte gesetzt.

5. In der anschließenden Tabelle sind die Möglichkeiten der Verarbeitung und Erzeugung von Überlöchern zusammengestellt. Der Code für die gewünschte Art der Überlochverarbeitung ist im AD_m-Teil des Befehls "PCF Lochkartenvorbefehl (3.8)" anzugeben.

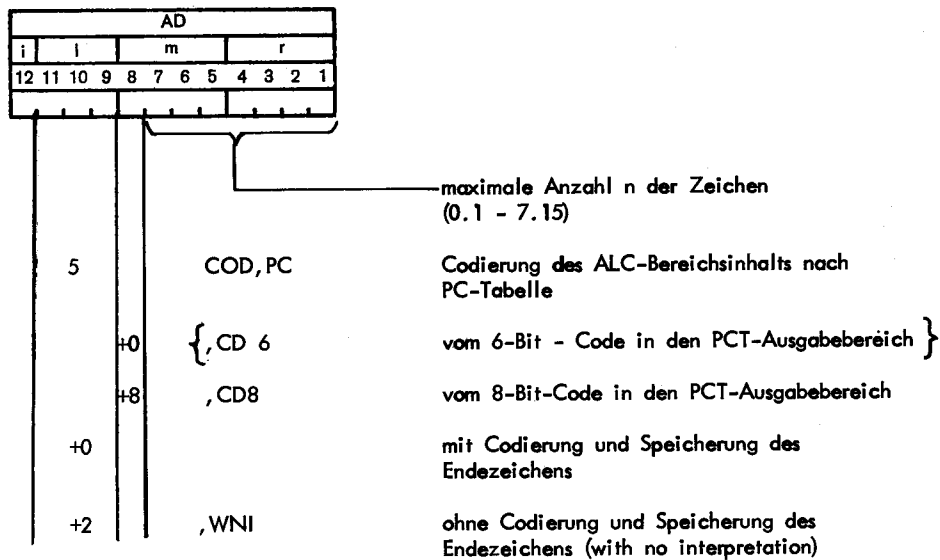
Lesen												Stanzen				
kein Überloch			Überloch 11			Überloch 12			Überloch 11 und 12			AD _m	Überloch 11		Überloch 12	
(A)	ML	MU	(A)	ML	MU	(A)	ML	MU	(A)	ML	MU		(A) ⁰	(A) ¹	(A) ⁰	(A) ¹
+	0	0	+	1	0	+	0	1	+	1	1	0	0	0	0	0
-	0	0	+	0	0	-	0	1	+	0	1	1	1	0	0	0
+	0	0	-	0	0	+	0	1	-	0	1	2	0	1	0	0
+	1	0	+	0	0	+	1	1	+	0	1	3	1	1	0	0
-	0	0	-	1	0	+	0	0	+	1	0	4	0	0	1	0
-	0	0	+	0	0	+	0	0	+	0	0	5	1	0	1	0
+	0	0	-	0	0	+	0	0	+	0	0	6	0	1	1	0
-	1	0	-	0	0	+	1	0	+	0	0	7	1	1	1	0
+	0	0	+	1	0	-	0	0	-	1	0	8	0	0	0	1
+	0	0	+	0	0	-	0	0	+	0	0	9	1	0	0	1
+	0	0	+	0	0	+	0	0	-	0	0	10	0	1	0	1
+	1	0	+	0	0	-	1	0	-	0	0	11	1	1	0	1
+	0	1	+	1	1	+	0	0	+	1	0	12	0	0	1	1
-	0	1	+	0	1	-	0	0	+	0	0	13	1	0	1	1
+	0	1	-	0	1	+	0	0	-	0	0	14	0	1	1	1
+	1	1	+	0	1	+	1	0	+	0	0	15	1	1	1	1

PCF	Lochkartenvorbefehl	3.8
-----	---------------------	-----

Bemerkungen:

1. Steht in AD_r die Kernspeicherstelle 0, erfolgt keine Überlochauswertung bzw. -erzeugung.
2. Die Vorbefehle "PCF Lochkartenvorbefehl (3.8)", "PTF Lochstreifen-vorbefehl (3.5)" (und "EDFS Druckvorbefehl für Drucker 2 (3.6)") werden intern in derselben Speicherzelle gemerkt. Man beachte also, daß sich diese Befehle gegenseitig aufheben und ersetzen.
3. Beim Modell 820/15 muß bei Anwendung dieses Befehls das Modul PC vorhanden sein.

COD, PC	Codierung des ALC-Bereichsinhalts nach PC-Tabelle (coding)	3.8
---------	--	-----



Wirkung:

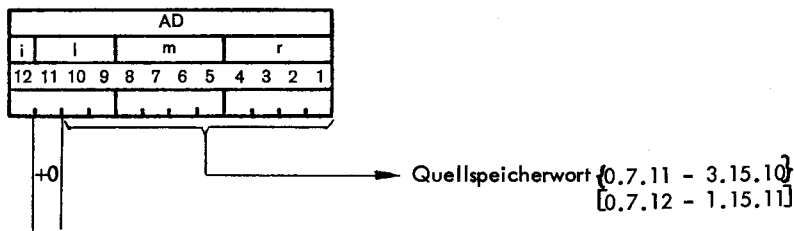
- Der Inhalt des ALC-Bereichs wird anhand der Lochkartencode-Tabelle (PC-Tabelle) in den Lochkartencode umgewandelt und im PCT-Ausgabebereich gespeichert. Die Anfangsstelle im ALC-Bereich legt der Bereichszeiger I_3 , die Anfangsstelle im PCT-Bereich legt der dazugehörige Ausgabebereichszeiger fest.
- Die Codierung wird beendet, wenn das Umschaltzeichen 3.15 oder eines der Endezeichen 6.8, 6.11, 6.12 auftritt. Ist $n \neq 0$, so werden maximal n Zeichen codiert, wenn keines der zuvor genannten Zeichen auftritt. Das Umschaltzeichen wird nicht codiert und gespeichert. Das Endezeichen wird nur dann codiert und gespeichert, wenn im AD_1 -Teil Bit 10= 0 gesetzt ist.
- Waren die Zeichen im 6-Bit-Code gespeichert, so bedeutet das Zeichen 3.15 eine Umschaltung auf den 8-Bit-Code des Endezeichens. Der Inhalt der beiden folgenden Kernspeicherstellen wird demnach als Endezeichen interpretiert.

COD,PC	Codierung des ALC-Bereichsinhalts nach PC-Tabelle	3.8
--------	--	-----

Bemerkungen:

1. Das Betriebsprogramm stellt nicht fest, wenn durch eine falsche Befehlsfolge der festgelegte Bereich überschritten wird.
2. Man beachte, daß der Bereichszeiger I_3 in I_{1m} immer die interne Speicherwortnummer enthält, die aus der Speicherwortnummer durch Addition von $[0.5]$ bzw. $[0.4]$ entsteht.
3. Wird der Text im 6-Bit-Code gespeichert, so steht der Bereichszeiger I_3 nur bei jedem zweiten Zeichen richtig. Darauf ist besonders dann zu achten, wenn die Operation durch Erreichen der Anzahl n beendet wurde.
4. Der Anfang der PC-Tabelle ist im $AD_{l_{mr}}$ -Teil des Befehls unter Adresse 0.0.2 im Anwenderprogrammblock 0 anzugeben. Die PC-Tabelle umfaßt 109 Zeichen und muß sich im Anwenderprogrammblock 0 befinden.
5. Bei Modell 820/15 ist der Zusatz "CD6" nicht erlaubt, da keine 6-Bit-Speicherung im ALC-Bereich möglich ist. Es ist unbedingt zu vermeiden, diese Codierung anzugeben. Der Befehl COD,PC darf nur in Verbindung mit dem PC-Modul benutzt werden.

MVV	Transport hoher Speicherworte (move very high register)	3.9
-----	--	-----



Wirkung:

Der Inhalt des durch AD_{lmr} Bit 1 bis Bit 10 benannten Speicherwortes (Quellspeicherwort) wird einschließlich Vorzeichen in das Speicherwort A übertragen. Der Inhalt des Quellspeicherwortes bleibt unverändert.

Beispiel:

Transport (SW 165) → SW 3 bei dezimalem Speicherwortinhalt

vorher: SW 165 + 0 0 0 0 0 0 0 0 0 0 7 3 6 5 1 4 2

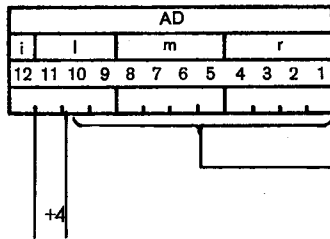
BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
	0 3 9	0	0 10 5		MVV, 165
	1				

nachher: SW 165 + 0 0 0 0 0 0 0 0 0 0 7 3 6 5 1 4 2
 SW 3 + 0 0 0 0 0 0 0 0 0 0 7 3 6 5 1 4 2

Bemerkungen:

1. Der Transport geschieht stellengerecht mit jeweils 4 Bit, d.h. es können auch sedezimale Stelleninhalte transportiert werden. Dies gilt für alle 16 Kernspeicherstellen, also einschließlich der Vorzeichenstelle.
2. Der Befehl "MVV Transport hoher Speicherworte (3.9)" ist auch zulässig für den Transport von Speicherworten {0.1 - 7.10} [0.1 - 7.1] nach Speicherwort 3, doch ist wegen des geringen Zeitaufwandes hierzu der Befehl "MV Transport (0.3)" besser geeignet.

MVVH	Transport nach hohem Speicherwort (move to very high register)	3.9
------	---	-----



Zielspeicherwort {0.7.11 - 3.15.10}
[0.7.12 - 1.15.11]

Wirkung:

Der Inhalt von Speicherwort A wird einschließlich Vorzeichen in das durch AD_{lmr} Bit 1 bis Bit 10 benannte Speicherwort (Zielspeicherwort) übertragen. Der Inhalt von Speicherwort 3 bleibt unverändert.

Beispiel:

Transport (SW 3) → SW 213 bei sedezimalem Speicherwortinhalt.

vorher: SW 3 7 0 0 0 0 0 0 0 0 0 0 0 0 4 13 7 10 15

BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
	0 3 9 0		4 13 5		MVVH, 213
	1				

nachher: SW 213 7 0 0 0 0 0 0 0 0 0 0 0 0 4 13 7 10 15

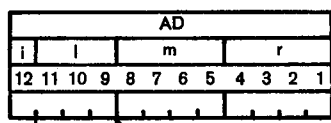
SW 3 7 0 0 0 0 0 0 0 0 0 0 0 0 4 13 7 10 15

MVVH	Transport nach hohem Speicherwort (move to very high register)	3.9
------	---	-----

Bemerkungen:

1. Der Transport geschieht stellengerecht mit jeweils 4 Bit, d.h. es können auch sedezimale Stelleninhalte transportiert werden. Dies gilt für alle 16 Kernspeicherstellen, also einschließlich der Vorzeichenstelle.
2. Der Befehl "MVVH Transport nach hohem Speicherwort (3.9)" ist auch zulässig für den Transport von Speicherwort 3 nach Speicherwort $\{0.1 - 7.10\}$ $[0.1 - 7.11]$, doch ist wegen des geringeren Zeitaufwandes hierzu der Befehl "MVH Transport (0.2)" besser geeignet.

EP	Vorbefehl Setzen des Anfanges vom Befehlsbereich	3,9
----	--	-----



0

EP letztes Speicherwort im Datenbereich
beim Betriebsprogramm MFA 1
(0.1 - 15.10)

beim Betriebsprogramm MFAGS1
(0.1 - 7.10)

Wirkung:

1. Dieser Vorbefehl darf nur in Verbindung mit dem Folgebefehl "WT,SEP,2.12.0.4.12" benutzt werden. Im Vorbefehl wird die Grenze zwischen Datenbereich und Befehlsbereich angegeben. Dieser Befehl ist nur bei dem Modell 820/15 anwendbar. Hier ist es möglich, in einem Kernspeicher Befehlsorte und Datenbereich unterzubringen. Im AD_{mr} wird das letzte Speicherwort angegeben, daß im Datenbereich liegt. Je nach Betriebsprogramm belegt der Befehlsbereich einen bestimmten Platz im Kernspeicher.

2. Betriebsprogramm MFA1

Der Datenbereich liegt von 0.0 bis zu dem im Befehl EP (3.9) angegebenen Speicherwort; danach kommt der Befehlsbereich. Ein Befehl nimmt 5 KSPST (Kernspeicherstellen) in Anspruch.

3. Das Betriebsprogramm MFAGS1

Der Datenbereich liegt von 0.0 bis zum angegebenen Speicherwort kleiner 7.11. Die Angabe errechnet sich aus:

16 Befehlsorte entsprechen einem Speicherwort der niedrigen Klasse (0.0 bis 7.11) (nur Operationsteile) und 3 Speicherworten der hohen Klasse ab 507 und abwärts. Vom Speicherwort 124 an beginnt die zweite Hälfte des Datenbereichs bis zu einer weiteren Grenze die sich automatisch aus der ersten Grenze errechnet, also nicht angegeben werden braucht.

Blatt B 190
1. 9. 1970

Datenverarbeitungssystem 820
Beschreibung der Befehle

EP	Vorbefehl Setzen des Anfanges vom Befehlsbereich	3.9
----	--	-----

Bemerkung:

1. Durch den Befehl 3,9 EP wird der Inhalt des Speicherwortes A zerstört (siehe Wirkung des Befehles MVV).
2. Als Beispiel für das Festlegen der Grenze siehe unter Befehl "2.12.0.4.12 WT, SEP".
3. Falls die angegebene Grenze im späteren Programmablauf von der Datenseite her überschritten wird, erfolgt interner Fehlerstopp.

STN	Kontenbefehle	3.10
-----	---------------	------

AD															
i				l				m				r			
12	11	10	9	8	7	6	5	4	3	2	1				
0				0				0							
2				0				0							
4				0				0							

STN, OUT Kontenauswurf, einfache automatische Konteneinzugseinrichtung

STN, IN Konteneinzug, einfache automatische Konteneinzugseinrichtung

STN, CUT Schlitzung bei einfacher automatischer Konteneinzugseinrichtung

Wirkung:

1. Bei $AD_1 = 0$ erfolgt ein Kontenauswurf.
2. Bei $AD_1 = 2$ erfolgt ein Konteneinzug bis zur ersten Schlitzung vom unteren Kontenrand aus gesehen. Das Betriebsprogramm wartet intern so lange, bis eine Karte im Schacht ist. Dieses gilt nur für das Modell 820/15.
3. Bei den Modellen 820/25 und 820/35 gibt es keinen Einzugsbefehl. Das Konto wird automatisch eingezogen, falls ein Konto in den Schacht eingeführt wird.
4. Es ist weiterhin darauf zu achten, daß das Betriebsprogramm SKZA2 vorhanden sein muß.
5. Bei dem Modell 820/15 muß das Modul ST vorhanden sein.
6. Bei $AD_1 = 4$ erfolgt eine Schlitzung. Diese Schlitzung soll vor jedem Kontenauswurf erfolgen, damit bei einem späteren Konteneinzug das Konto automatisch auf die nächste freie Zeile eingezogen wird.

Bemerkung:

Der Zeilenzähler wird bei einem Konteneinzug bzw. -auswurf nicht angesprochen. Er dient nur zur Vertikalpositionierung eines im Schacht befindlichen Kontos.

INT	Kontenbefehle	3.10
-----	---------------	------

AD											
i				m				r			
12	11	10	9	8	7	6	5	4	3	2	1

INT Intramat Schacht 1
 INTB Intramat Schacht 2
 INTO Intramat Schacht 1 und 2
 ,IN Konteneinzug
 ,CUT Schlitzung

Wirkung:

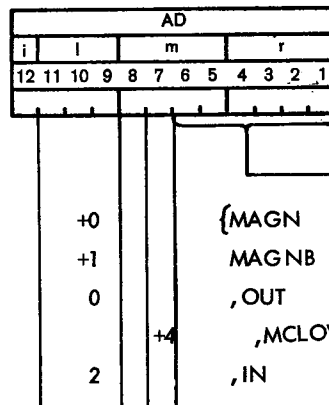
- Der Intramat ist beim Modell 820/15 nur in Verbindung mit dem "Modul IN" anschlieÙbar. Bei $AD_i +0$ wird der Schacht 1 angesprochen. Bei $AD_i +1$ wird der Schacht 2 angesprochen. Bei $AD_{imr} +0.0.1$ werden beide Schächte angesprochen.
- Bei $AD_i = 2$ erfolgt ein Einzug bis zur letzten Schlitzstelle.
Das Betriebsprogramm wartet intern so lange, bis eine Karte im betreffenden Schacht ist.
- Bei $AD_i = 4$ erfolgt eine Schlitzung im angegebenen Schacht. Die Schlitzung soll vor jedem Kontenauswurf erfolgen, damit bei einem späteren Konteneinzug das Konto automatisch auf die nächste freie Zeile eingezogen wird.

Bemerkung:

Der Zeilenzähler wird bei diesen Befehlen nicht angesprochen und braucht auch deshalb nicht geladen zu werden. Er dient nur zur Vertikalpositionierung innerhalb des Schachtes.

Dieser Befehl ist bei den Modellen 820/25/35 nur in Verbindung mit dem Betriebssystem SKZA2 möglich.

MAGN	Magnetkontenhauptbefehle	3.10
------	--------------------------	------



Zeilennummer (0.0 - 7.15)

+0	{MAGN	Magnetkontenschacht 1 (vorn)}
+1	MAGNB	Magnetkontenschacht 2 (hinten)
0	,OUT	Auswurf ohne Schreiben
	,MCLOW	Auswurf nach unten
2	,IN	Einzug ohne Lesen auf Zeile AD _{mr}

Wirkung:

- Bei {AD_l = 0 bzw.} AD_l = 1 erfolgt im {jeweiligen} Magnetkontenschacht der Auswurf einer Magnetkontokarte ohne Beschreiben des Magnetstreifens. Es kann wahlweise auch ein Auswurf nach unten in ein Ablagefach angeordnet werden.
- Bei {AD_l = 2 bzw.} AD_l = 3 erfolgt im {jeweiligen} Magnetkontenschacht der Einzug einer Magnetkontokarte ohne Lesen der Information des Magnetstreifens. Das Betriebsprogramm wartet dazu intern, bis eine Karte im betreffenden Schacht steckt. Nach dem Einzug befindet sich die in AD_{mr} Bit 1 bis Bit 7 angegebene Zeile auf der Höhe des Kugelkopfes.
- Die Merker ML, MU, MC werden folgendermaßen gesetzt:
Beim Auswurf ist ML = 1, wenn die Karte nach oben über die Lichtschranke gesprungen ist, beim Einzug besagt ML = 1, daß sich die Karte nach erfolgtem Einzug nicht mehr im Schacht befindet, andernfalls ist ML = 0.

Die Merker MU und MC sind nach Ausführung des Befehles auf 0 gesetzt.

MAGN	Magnetkontenhauptbefehle	3,10
------	--------------------------	------

Beispiel:

In Schacht 2 soll eine Magnetkontokarte ohne Lesen auf Zeile 20 eingezogen werden. Die Höhe der Karte beträgt 69 Zeilen + 3,5 mm, wovon der Kartenkopf 14 Zeilen einnimmt.

BW-Adresse	Op. Teil			l	Adreßteil			Symb. Adr.	Bemerkungen
	0	3	3	0	1	4	4		LC, MAGB, 4, 4
	1	3	10	0	3	1	4		MAGNB, IN, 20
	2								

Der Zeilenzähler steht anschließend auf 20, die Merker sind ML = 0, MU = 0, MC = 0. Es befindet sich Zeile 20, d.h. die 35. Zeile vom oberen Kartenrand auf der Höhe des Kugelkopfes.

Bemerkungen:

- Vor jedem Einzugs einer Magnetkontokarte muß der Zeilenzähler des betreffenden Schachtes mit einer Zahl m geladen werden, damit die Karte auf die richtige Zeile eingezogen wird. Diese Zahl m berechnet sich aus der gesamten Kartenhöhe und der Höhe des Kartenkopfes jeweils in Zeilen gemessen nach der Formel

$$m = \text{Kartenhöhe} - \text{Kopfhöhe} + 13.$$

Unter dieser Voraussetzung trägt die erste beschreibbare Zeile von oben die Nummer 0.

- Kommt nach einem Einzugsbefehl die Meldung ML = 1, so ist vor einem erneuten Einzugsbefehl der Zeilenzähler des betreffenden Schachtes neu zu laden.

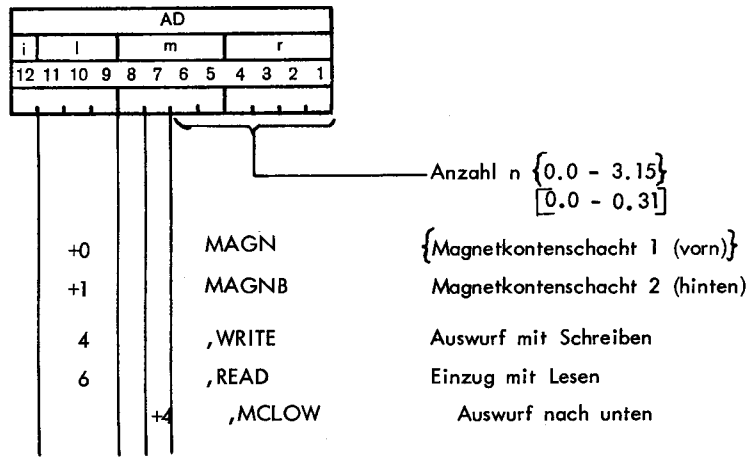
3 3 1 1 1 4

3 10 1 1 1

(3)

1 7

MAGN	Magnetkontenhauptbefehle	3.10
------	--------------------------	------



Wirkung:

- Bei $\{AD_1 = 4 \text{ bzw. } AD_1 = 5\}$ erfolgt im {jeweiligen} Magnetkontenschicht das Beschreiben des Magnetstreifens der Magnetkontokarte während des Auswurfs der Karte. Dabei wird auf den Magnetstreifen von unten nach oben der Inhalt der ersten $n+1$ Speicherworte des ALC-Bereichs beginnend mit Bereichsanfang übertragen.
- Bei $\{AD_1 = 6 \text{ bzw. } AD_1 = 7\}$ erfolgt im {jeweiligen} Magnetkontenschicht der Einzug einer Magnetkontokarte mit Lesen der Information des Magnetstreifens. Dabei werden vom Magnetstreifen von unten nach oben die ersten $n+1$ Gruppen zu je 16 Zeichen gelesen und in die ersten $n+1$ Speicherworte des ALC-Bereiches beginnend mit Bereichsanfang übertragen.

Der Inhalt der beiden ersten Kernspeicherstellen des ALC-Bereiches gibt sedezimal die Nummer der Zeile an, auf die die Karte nach dem Lesen gefahren wird.

MAGN	Magnetkontenhauptbefehle	3.10
------	--------------------------	------

3. Beim Modell 820/15 können zwischen 1 und 32 Gruppen aufgesprochen bzw. gelesen werden.
Eine Kontrolllesung ist beim Auswurf mit Schreiben nicht möglich.
4. Der Anfang des ALC-Bereiches ist durch den Befehl "ALBG ALC-Bereichs-anfang setzen (3.4)" vorzugeben.
5. Er kann für den Auswurf mit Schreiben das anschließende Kontrolllesen festlegen. In diesem Falle wird die Magnetkontokarte nach dem Beschreiben wieder eingezogen, die aufgeschriebene Information gelesen und mit dem Bereichsinhalt stellenweise verglichen. Erst wenn die Kontrolle keinen Fehler festgestellt hat, erfolgt der endgültige Auswurf der Karte.
6. Es kann wahlweise ein Auswurf nach unten in ein Ablagefach angeordnet werden.
7. Beim Auswurf ist $ML = 1$, wenn die Karte nach oben über die Lichtschranke gesprungen ist, beim Einzug besagt $ML = 1$, daß sich die Karte nach erfolgtem Einzug nicht mehr im Schacht befindet, andernfalls ist $ML = 0$.

Beim normalen Auswurf mit Schreiben sind nach Ausführung des Befehles die Merker $MU = 0$, $MC = 0$. Beim Auswurf mit Schreiben und Kontrolllesen sowie beim Einzug mit Lesen werden die Merker MU , MC folgendermaßen gesetzt:

	MU	MC
Richtig gelesen	0	0
Falsch gelesen	1	0
Leerkarte	1	1

	Magnetkontenhauptbefehle	3.10
--	--------------------------	------

8. Der Einzug einer Magnetkontokarte erfolgt erst dann, wenn sich im angesprochenen Schacht eine Karte befindet. Hierzu ist kein Wartebefehl erforderlich.

Verarbeitung von Kontrolldaten:

1. Beim Beschreiben des Magnetstreifens der Magnetkontokarte werden aus den aufzuschreibenden Speicherwortinhalten Kontrolldaten berechnet und ebenfalls aufgeschrieben. Beim Lesen der Information eines Magnetstreifens werden diese Kontrolldaten erneut berechnet und mit den aufgeschriebenen verglichen. Im Falle eines Unterschiedes wird noch bis zu zweimal nach unterschiedlichen Verfahren versucht, den Magnetstreifen der Karte zu lesen. Tritt dann immer noch ein Fehler auf, so kommt die Meldung "Falsch gelesen".
2. Die berechneten Kontrolldaten werden in den Speicherworten 1 bis 5 zwischengespeichert, wobei die Belegung von der Anzahl der aufzuschreibenden bzw. einzulesenden Speicherworte abhängig ist:

SW 1 wird immer vollständig benutzt.

Beginnend mit SW 2 Stelle 0 wird für jedes zu verarbeitende Speicherwort eine Stelle belegt. Die restlichen Speicherwortstellen bleiben unverändert. Es wird also:

SW 2 immer benutzt,
SW 3 nur bei Verarbeitung von mehr als 16,
SW 4 nur bei Verarbeitung von mehr als 32,
SW 5 nur bei Verarbeitung von mehr als 48 Speicherwortinhalten
3. Beim Einlesen der Information einer Magnetkontokarte werden die Daten geprüft und festgestellte Fehler in den Speicherworten 1 bis 5 folgendermaßen angezeigt:

Der Inhalt der Stelle p im Speicherwort 1 ist genau dann von 0 verschieden, wenn in einer Stelle p der gelesenen Speicherworte ein Fehler festgestellt wurde.

	Magnetkontenhauptbefehle	3.10
--	--------------------------	------

Da eine Magnetkontokarte höchsten: den Inhalt von 64 Speicherworten speicher kann, steht jedem dieser Speicherworte eine Stelle der Speicherworte 2 bis 5 für Kontrolldaten zur Verfügung. Nach dem Einlesen ist der Inhalt der q-ten Stelle (beginnend mit SW 2 Stelle 0) genau dann von 0 verschieden, wenn im q-ten gelesenen Speicherwort ein Fehler erkannt wurde.

Bemerkungen:

1. Der Magnetstreifen der Magnetkontokarte ist in 4 Informations- und 2 Taktspuren aufgeteilt. Beim normalen Betrieb wird mit beiden Taktspuren gearbeitet. Ist beim Einzug mit Lesen ein Lesefehler festgestellt worden, so wird der Lesevorgang noch bis zu zweimal wiederholt, wobei jeweils nur mit einer Taktspur gearbeitet wird. Erst nach der dritten fehlerhaften Lesung erfolgt die Meldung "falsch gelesen".
2. Man beachte, daß der Vorbefehl "ALBG ALC-Bereichsanfang setzen (3.4)" nach Ausführung eines Magnetkontenhauptbefehles nicht gelöscht ist. {Ferner darf der Vorbefehl "ALBG,READC (3.4)" für den Auswurf mit Schreiben und Kontrolllesen keinesfalls vor einem Befehl "MAG,READ" oder "MAGB,READ" gegeben werden,}[außerdem darf er beim Modell 820/15 grundsätzlich nicht benutzt werden.]

SORTMX	Sortimatwert nach Indexregister	3.10
--------	---------------------------------	------

AD											
i			l			m			r		
12	11	10	9	8	7	6	5	4	3	2	1
						6					0

Wirkung:

1. Ein am Intromat angebrachter Sortimat tastet Codierungen von Kontokarten ab, die in Form von Löchern am unteren Rand der Kontokarte gestanzt sind.
2. Durch den Befehl SORTMX Sortimatwert nach I (3.10)¹¹ wird der festgestellte Code einer Karte in das Indexregister gebracht, und zwar auf folgende Weise:
3. Den Bit-Positionen 1 bis 4 des Indexregisters entsprechen auf der Karte die Löcher 1 bis 4 von links nach rechts. Das am weitesten links gelegene Loch setzt Bit 1 = 1, das nächste rechts davon setzt Bit 2 = 1, usw.
4. Dadurch kann I_m die Werte von 0 (ungestanztes Konto) bis 15 (Konto mit allen vier möglichen Stanzungen) annehmen.
5. Nach I_m gelangt der Wert 0.1, wenn ein Konto eingelaufen ist, ist dagegen kein Konto eingezogen worden (unabhängig von der Erfüllung der Bedingung 8.4 bzw. 8.8), so ist I_m = 0.0

Die Befehlsfolge

SORTMX
CX,8..-16,XI
BXG,FALSCH

ermöglicht es, dies zu erkennen und mit der bereinigten Sortimat-Wertigkeit (0 bis 15) weiterzuarbeiten.

	Listgerät	3.10
--	-----------	------

AD											
i				m				r			
12	11	10	9	8	7	6	5	4	3	2	1
0				8				0			
+0				+0				+0			

MCORG

,SELZ Setze Merker ML wenn Trommel frei

Wirkung:

1. Befindet sich keine Karte in der Trommel, so wird Merker ML = 1 gesetzt. Die Abfrage erfolgt durch mehrere Lichtschranken.

Bemerkung:

Es ist vor einem Einzugsbefehl zu beachten, daß sich nicht mehr als 1 Karte in der Trommel befindet. In diesem Falle wird die Abfrage "Trommel frei?" angewendet.

	Listgerät	3.10
--	-----------	------

AD															
i				l				m				r			
12	11	10	9	8	7	6	5	4	3	2	1				

MCORG
,SEL1 Ablagefach 1
,SEL2 Ablagefach 2
,SEL3 Ablagefach 3

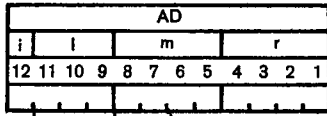
Wirkung:

1. Es wird die jeweilige Weiche geöffnet, wenn die zugehörige und nächstfolgende Lichtschranke nicht unterbrochen ist. Im AD_r wird die jeweilige Weiche für das Ablagefach codiert (1 - 3). Es erfolgt eine Ablage in das angegebene Fach. Dieser Befehl läuft simultan zum weiteren Befehlsablauf.
2. Die Merker werden folgendermaßen gesetzt:
Wenn Karte klemmt: Merker MU = 0, Merker ML = 0, Merker MC = 1, andernfalls ist Merker MC = 0.

Bemerkung:

Die Größe der Magnetkontenkarte ist in diesem Falle nicht maßgebend.

	Listgerät	3.10
--	-----------	------



+1

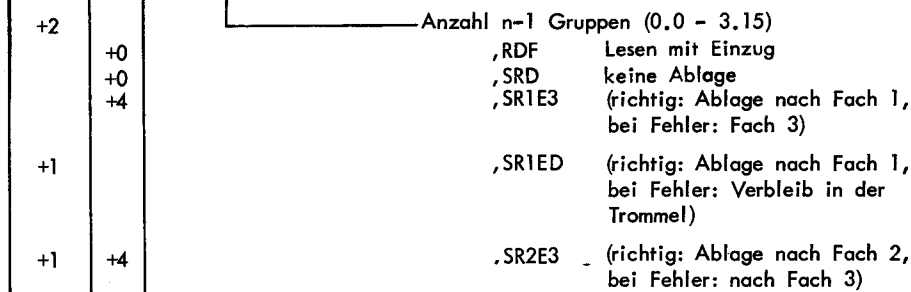
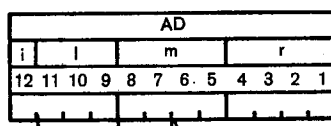
Anzahl n-1 Gruppen (0.0 - 3.15)
,RDO Lesen ohne neuen Einzug (keine Ablage)

Wirkung:

1. Die Karte, die sich in der Trommel befindet, wird gelesen, es erfolgt keine Ablage. Es wird die Anzahl n Gruppen eingelesen. Dabei werden vom Magnetstreifen von unten nach oben die Gruppen gelesen und die ersten n Gruppen in den ALC-Bereich beginnend bei Bereichsanfang übertragen.
2. Der Anfang des ALC-Bereiches wird durch den Befehl "ALBG(3,4)" vorgegeben. Sind Lesefehler entstanden, so wird der Lesevorgang bis zu zweimal wiederholt. Die Merker setzen sich folgendermaßen:

MU	MC	
0	0	Richtig gelesen
1	0	Falsch gelesen
1	1	Leerkarte

	Listgerät	3.10
--	-----------	------



Wirkung:

1. Es erfolgt ein Einzug einer Kontenkarte in die Trommel. Anschließend werden die angegebenen Gruppen n+1 in den ALC-Bereich ab Bereichsanfang eingelesen. Ist beim Lesen ein Lesefehler aufgetreten, so läuft die Karte einmal in der Trommel herum und wird erneut gelesen und verglichen. Danach erfolgt die Entscheidung über den Verbleib der Karte; außerdem werden die Merker entsprechend gesetzt.

2. Bei $AD_l = 2$ erfolgt ein Einzug mit Lesen. Danach verbleibt die Karte unbedingt in der Trommel.

Bei $AD_l = 2, AD_m = +4$ erfolgt ein Einzug mit Lesen. Anschließend wird die Karte bei Fehler nach Ablagefach 3 transportiert, ansonsten nach Fach 1.

Bei $AD_l = 3$ erfolgt ein Einzug mit Lesen. Anschließend wird die Karte nach Fach 1 abgelegt, falls kein Fehler auftrat, ansonsten verbleibt das Konto in der Trommel.

Bei $AD_l = 3, AD_m = +4$ erfolgt ein Einzug mit Lesen. Anschließend wird bei einem Fehler die Karte nach Fach 3 abgelegt, ansonsten nach Fach 2.

	Listgerät	3.10
--	-----------	------

3. Die Merker setzen sich nach Ausführung des Befehles folgendermaßen:

ML	MU	MC	
0	1	1	Leerkarte
0	0	0	Karte richtig gelesen
0	1	0	Karte falsch gelesen
0	0	1	Karte klemmt (Motor aus)

	Listgerät	3.10
--	-----------	------

AD											
i	l				m				r		
12	11	10	9	8	7	6	5	4	3	2	1

4		,WRO	Anzahl n-1 der zu übertragenden Speicherworte (0.0 - 3.15)
+0	+0	,SWD	Schreiben ohne Einzug (keine Ablage)
+0	+4	,SW1ED	(richtig: Fach 1, falsch: Verbleib in der Trommel)
+1	+0	,SW2ED	(richtig: Fach 2, falsch: Verbleib in der Trommel)

Wirkung:

1. Es wird der Inhalt des ALC-Bereichs auf das Magnetkonto aufgesprochen. Das Konto muß sich in der Trommel befinden, das kann nur durch einen Befehl "Konteneinzug mit Lesen" geschehen.

Das Konto wird von unten nach oben beschrieben, beginnend bei ALC-Bereichsanfang.

2. Durch den Vorbefehl "ALBG 3.4" wird der Bereichsanfang definiert und entschieden, ob anschließend eine Kontrolllesung vorgenommen werden soll. Bei Kontrolllesung wird die Magnetkontenkarte einmal um die Trommel transportiert.

	Listgerät	3.10
--	-----------	------

3. Die Merker setzen sich folgendermaßen:

ML	MU	MC	
0	0	0	richtig geschrieben
0	1	0	falsch geschrieben
0	0	1	Karte klemmt, Motor aus
1	0	0	keine Karte in der Trommel bei Aufruf

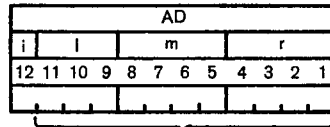
4. Ist ein Fehler aufgetreten, so verbleibt die Karte in der Trommel.

Bei $AD_1 = 4$ verbleibt die Karte grundsätzlich nach Ausführung des Befehles in der Trommel.

Bei $AD_1 = 4$, AD_m wird die Karte in das Fach abgelegt, falls keine Fehler aufgetreten sind, ansonsten verbleibt die Karte in der Trommel.

Bei $AD_1 = 5$ wird die Karte in das Fach abgelegt, falls keine Fehler aufgetreten sind, ansonsten verbleibt die Karte in der Trommel.

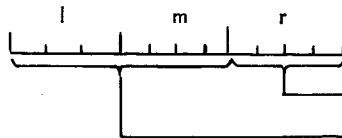
BPX	ALC-Bereichszeiger I ₃ laden (buffer pointer indexregister)	3.11
-----	---	------



Konstante (0.0.0 - 7.15.15)

Wirkung:

- Die in AD_{l_{mr}} sedezimal dargestellte Konstante wird um {0.5.0 (das entspricht 80)} [0.4.0 (das entspricht 64)] erhöht und nach Indexregister I₃ gebracht. Die Addition von 0.4.0 geschieht modulo 2¹¹, so daß also I₃ ohne Indexvorbefehl beliebig geladen werden kann.
- Das Indexregister I₃ ist in einer Sonderfunktion der Bereichszeiger für den ALC-Bereich. Dazu ist der Inhalt des Indexregisters folgendermaßen aufgebaut:



Kernspeicherstelle (0 - 15)

interne Speicherwortnummer (0.6 - 7.15)

- Die Speicherworte 0.1 bis {7.10} [7.11] sind intern {0.6} [0.5] bis 7.15 numeriert. Deshalb steht in l_m die interne Speicherwortnummer, die aus der normalen Speicherwortnummer durch Addition von {0.5} [0.4] entsteht. In l_r ist die Stelle des Speicherwortes anzugeben, auf die der Bereichszeiger zu setzen ist.

Beispiel:

Der ALC-Bereichszeiger soll in Speicherwort 13 auf Stelle 14 gesetzt werden.

BW-Adresse	Op. Teil			l	Adreßteil			Symb. Adr.	Bemerkungen
	0	3	11	0	0	13	14		BPX, 13, 14
	1								

nachher: I₃ $\left\{ \begin{array}{|c|c|c|} \hline 1 & 2 & 14 \\ \hline \end{array} \right\}$ entspricht (I₃) = {302} [286]

$\left[\begin{array}{|c|c|c|} \hline 1 & 1 & 14 \\ \hline \end{array} \right]$

BPX	ALC-Bereichszeiger I ₃ laden (buffer pointer indexregister)	3.11
-----	---	------

Bemerkungen:

1. Zur Ausgabe des ALC-Bereichsinhalts auf Drucker oder Streifenstanzer wird der Bereichszeiger I₃ benutzt. Da diese Ausgaben simultan zum weiteren Befehlsablauf ausgeführt werden, darf der Inhalt von I₃ also erst nach Beendigung der Ausgabe verändert werden.
2. Sind im ALC-Bereich Zeichen im 6-Bit-Code gespeichert, so steht der Bereichszeiger nur bei jedem zweiten Zeichen richtig.

MGC	Magnetbandcassettenbefehle	3.12
-----	----------------------------	------

AD											
i	l	m				r					
12	11	10	9	8	7	6	5	4	3	2	1
0		2				0					

Cassettenhauptbefehl (nur mit Folgebefehl anwendbar)

Bemerkung:

Der Befehl "3.12 0.2.0 MGC" darf nur mit einem der aufgeführten Folgebefehle angewendet werden. Die Folgebefehle belegen den OP-Teil und den AD-Teil. Folgt ein unzulässiger Folgebefehl, so erfolgt interner Fehlerstopp.

Es ist zu beachten, daß beim Modell 820/15 max. 2 Cassettengeräte angeschlossen werden können, wobei nur 1 Cassette immer arbeiten kann; außerdem müssen die Moduln TC und GP vorhanden sein.

Bei den Modellen 820/25/35 können max. 4 Cassettengeräte angeschlossen werden; außerdem muß das Betriebsprogramm MSKZ3CASS vorhanden sein.

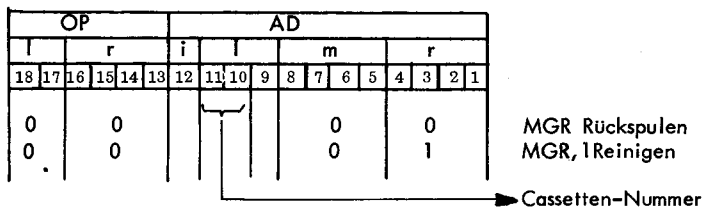
Folgebefehle

In sämtlichen Folgebefehlen muß die Cassetten-Nummer angegeben werden. Die Cassetten-Nummer wird in Bit 10 bis 11 dargestellt.

Bit 10 = 0	Bit 11 = 0	≙ Cassette 1	AD _l = +0
Bit 10 = 1	Bit 11 = 0	≙ Cassette 2	AD _l = +2
Bit 10 = 0	Bit 11 = 1	≙ Cassette 3	AD _l = +4
Bit 10 = 1	Bit 11 = 1	≙ Cassette 4	AD _l = +6

Es darf nur ein Folgebefehl dem Hauptbefehl folgen.

MGC	Magnetbandcassettenbefehle	3.12
MGR	Rückspulen (Folgebefehl)	0.0
bzw. MGR,1	Reinigen (Folgebefehl)	0.0



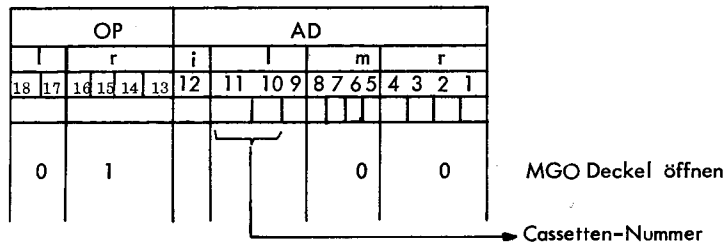
Wirkung MGR Rückspulen:

1. Die angesprochene Cassette nimmt einen Bandrücklauf vor bis zum physikalischen Bandanfang. Der Inhalt des Magnetbandes wird nicht gelesen oder verändert. Der Deckel der Cassette muß während des Ablaufes geschlossen sein. Ansonsten erfolgt Internfehlerstop und die gelbe Lampe leuchtet auf. Durch Schließen des Deckels und Betätigung der F-Taste wird der Ablauf erneut vorgenommen. Wird eine nicht angeschlossene Cassette angesprochen, so erfolgt die gleiche Fehlermeldung. Die maximale Rückspulzeit liegt bei 20 sec.
2. Dieser Befehl läuft simultan zum anderen Befehlsablauf ab. Wird eine Cassette angesprochen, so wartet das Betriebsprogramm intern so lange, bis der Befehl ausgeführt ist.
3. Durch den Befehl "2.12 0.10.1 SMWT, MTWS" wird erreicht, daß gewartet wird, bis der Bandanfang erreicht ist. Anschließend wird MU auf "1" gesetzt (Bandanfangsmarke überschritten).

Wirkung MGR,1 Reinigen:

1. Der Reinigungsbefehl läßt das Band etwa 20 sec. lang vor- und anschließend die gleiche Zeit zurücklaufen.
2. Ansonsten gelten die Angaben unter dem Befehl MGR = Rückspulen.
3. Bei der Anwendung dieses Befehls muß eine Cassette mit einem Reinigungsband verwendet werden.

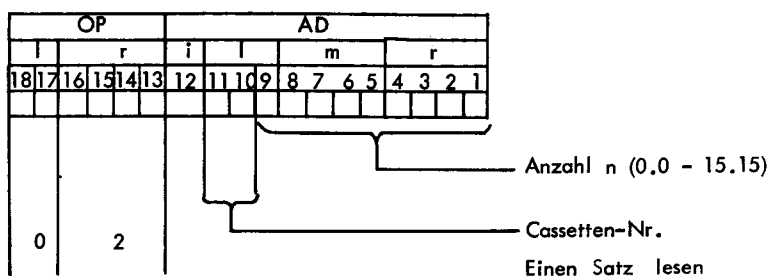
MGC	Magnetbandcassettenbefehle	3.12
MGO	Deckel öffnen	0.1



Wirkung:

1. Bei diesem Befehl wird der Deckel der Cassette geöffnet. Die Cassette kann jetzt entnommen, gewendet oder gegen eine andere ausgetauscht werden. Der Deckel kann nur durch Programmbefehl geöffnet werden. Kann der Deckel nicht geöffnet werden, so erfolgt Fehlermeldung durch Merker MC "1", sofern SMWT, MTWS programmiert wurde.
2. Ist der Deckel bei einem anderen Arbeitsablauf mit dem Cassetten-Gerät geöffnet, so erfolgt eine Fehlermeldung vom Betriebsprogramm.
3. Im AD_l-Teil wird die Cassetten-Nummer codiert.
4. Der Deckel muß manuell geschlossen werden.

MGC	Magnetbandcassettenbefehle	3.12
MGRC	Einen Satz lesen	0.2

Wirkung:

- Es wird ein Satz in den Eingabebereich eingelesen. Ein Satz kann maximal 256 Zeichen haben (+ 1 LPC-Prüfzeichen). Aus Sicherheitsgründen dürfen nicht weniger als 3 Zeichen eingelesen werden. Eingelesen wird die angegebene Anzahl n + 1 Zeichen + 1 Prüfzeichen. Das Prüfzeichen wird also in die anzugebende Anzahl nicht mit einbezogen, es belegt jedoch beim Einlesen auch einen Platz im Bereich.

Beispiel 1:

Ein Satz besteht aus fünf 8-Bit-Informationen und 1 Prüfzeichen = 6 Zeichen. Soll dieser Satz eingelesen werden, so wird der Befehl MGRC,4 gegeben (0.2 0.0.4). Es werden jetzt 6 Zeichen zu je 8 Bit in den Bereich eingelesen.

Beispiel 2:

Die codierte Anzahl 0 bedeutet: 1 Datenzeichen + 1 LPC-Prüfzeichen lesen.

- Die Zeichen werden im Eingabebereich im 8-Bit-Code dargestellt. Das bedeutet, daß ein Zeichen 2 Kernspeicherstellen im Bereich belegt. Auf dem Magnetband werden die Zeichen seriell im 8-Bit-Code zuzüglich eines Prüfbits und einer Zeichenlücke (in der Größe von 2 Bit) gespeichert.
- Die Anzahl der Zeichen in einem Satz ist nicht fest. Es ist kein Endezeichen erforderlich, da ein Satz vom nächsten durch die Satzlücke (in der Größe von etwa 12 mm) getrennt wird.

MGC	Magnetbandcassettenbefehle	3.12
MGRC	Einen Satz lesen	0. 2

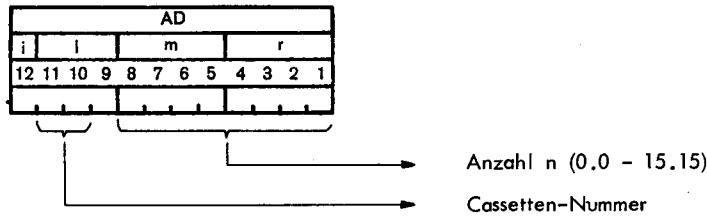
4. Der Lesevorgang wird beendet bei:
1. erreichter Anzahl und Satzende
 2. erreichter Anzahl, kein Satzende
 3. bei Satzende, Anzahl noch nicht eingelesen
 4. keine Daten vorhanden
- 4.1 Falls die angegebene Anzahl eingelesen ist und das Satzende erreicht ist, weiterhin kein Fehler beim Lesen entstanden ist, so ist der Befehl ordnungsgemäß ausgeführt. Wenn der Befehl "SMWT,MTWS" folgt, wird der Merker MC "0" gesetzt.
- 4.2 Falls die angegebene Anzahl eingelesen worden ist, jedoch das Satzende noch nicht erreicht worden ist, so wird dies in jedem Falle als Fehler gemeldet, da kein Prüfzeichen erkannt worden ist. Es ist deshalb zu vermeiden, weniger Zeichen einzulesen, als im Satz vorhanden sind. In diesem Falle wird eine Kontrolllesung vorgenommen (siehe Punkt 5). Durch den Befehl "SMWT, MTWS" wird hier der Merker MC "1" gesetzt.
- 4.3 Wenn das Satzende erkannt worden ist (Satzlücke ist aufgetreten), jedoch die angegebene Anzahl noch nicht eingelesen wurde, so wird bei Satzende der Lesevorgang abgebrochen. Das Satzende wird durch die Satzlücke von 12 mm gekennzeichnet. Sind ansonsten keine Lesefehler entstanden, so ist, falls der Befehl "SMWT, MTWS" folgt, der Merker MC "0" gesetzt.
- 4.4 Treten bei einem Lesebefehl keine Zeichen auf, so wird etwa 35 sec. nach Daten gesucht. Sind danach keine Daten aufgetreten, so wird der Lesebefehl beendet und der Merker MC "1" gesetzt.
5. Falls ein Lesefehler auftritt, wird der Satz bis zum Ende in den Eingabebereich eingelesen. Anschließend erfolgt ein Rücklauf zum Satz-anfang. Der Satz wird erneut eingelesen. Dieser Vorgang kann sich bis zu 9 mal wiederholen, falls erneut Fehler auftreten. Diese Kontrollen nimmt das Betriebsprogramm vor. Ist nach dem Lesebefehl der Befehl "SMWT,MTWS" codiert, so wird der Merker MC "1" gesetzt, wenn der 10. Leseversuch fehlerhaft war.

MGC	Magnetbandcassettenbefehle	3.12
MGRC	Einen Satz lesen	0. 2

6. Die Werte werden immer ab Bereichsanfang eingelesen. Der Eingabebereichszeiger läuft intern mit. Er steht nach Beendigung des Vorganges ein Zeichen weiter.
7. Da dieser Befehl simultan zum weiteren Befehlsablauf behandelt wird, ist darauf zu achten, daß während des Lesevorganges nicht der Eingabebereich oder der Eingabebereichszeiger verändert wird. In diesem Falle sollte ein Wartebefehl codiert werden.
8. Wird die gleiche Cassette von einem anderen Befehl angesprochen, so wartet das Betriebsprogramm intern so lange, bis der laufende Befehl abgearbeitet ist. Hierzu ist kein Wartebefehl erforderlich.
9. Im Befehl "SMWT, MTWS" werden nach Beendigung der Simultanarbeit die Merker wie folgt gesetzt:

MU	MC	Meldung
1	0	Bandendemarke aufgetreten
0	1	Falsch gelesen, Bereichsüberschreitung
1	1	Falsch gelesen und Bandendemarke aufgetreten

MGC	Magnetbandcassettenbefehle	3.12
MGWC	Einen Satz aufschreiben	0.3



Wirkung:

1. Es wird ein Satz aus dem Bereich ausgegeben. Ein Satz kann maximal 256 Zeichen haben (+ 1 LPC-Prüfzeichen). Aus Sicherheitsgründen dürfen nicht weniger als 3 Zeichen ausgegeben werden. Ausgegeben wird die codierte Anzahl n + 1 Zeichen + 1 Prüfzeichen. Das Prüfzeichen wird also in die anzugebende Anzahl nicht mit einbezogen, es wird automatisch vom Betriebsprogramm zusätzlich ausgegeben. Es steht als letztes Zeichen eines Satzes, hat jedoch nicht die Funktion eines Endezeichens.
2. Die Zeichen werden im Bereich im 8-Bit-Code dargestellt (8Bit = ein alpha-numerisches oder zwei numerische Zeichen). Diese 8 Bit beanspruchen also 2 Kernspeicherstellen. Auf dem Magnetband werden diese 8 Bit-Zeichen zuzüglich eines Prüfbits und einer Zeichenlücke (in der Größe von 2 Bit) gespeichert. Das Prüfzeichen am Ende eines Satzes hat den gleichen Aufbau. Nach dem Prüfzeichen wird eine Satzlücke (gap) von etwa 12 mm Länge gebildet.
3. Die Länge eines Satzes wird fest angegeben. Ein Endezeichen (3.15, 6.8, 6.11, 6.12) bewirkt nicht das Beenden des Schreibvorganges. Da der Datenblock vorn und hinten mit einem Gap versehen ist, wird kein besonderes Endezeichen benötigt. Es wird grundsätzlich die angegebene Anzahl aufgeschrieben.

MGC	Magnetbandcassettenbefehle	3.12
MGWC	Einen Satz schreiben	0. 3

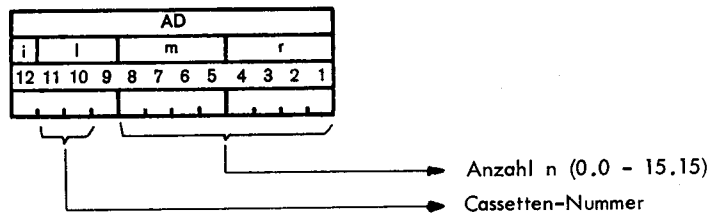
4. Nach dem Schreibvorgang nimmt das Betriebsprogramm eine Kontrolllesung vor. Die aufgeschriebenen Zeichen und Prüfzeichen werden mit dem Inhalt des Ausgabebereiches verglichen, nachdem das Band bis an den Satzanfang zurückgespult worden ist.

Wird hierbei ein Fehler festgestellt, so wird zum Satzanfang zurückgespult und der Satz erneut aufgeschrieben. Anschließend erfolgt eine erneute Kontrolllesung. Das Betriebsprogramm kann den Vorgang bis zu 7 mal wiederholen. Bleiben alle Versuche erfolglos, so wird, falls der Befehl "SMWT, MTWS" codiert wird, der Merker MC = "1" gesetzt. Nach dem 3. und 6. Versuch wird die Satzücke vor dem Satzanfang um das ca. 7-fache vergrößert (Mehrfachgap).

5. Der Bereichszeiger zählt während des Schreibvorganges mit (0.1 bis $n + 1$).
6. Dieser Befehl läuft simultan zum folgenden Befehlsablauf ab. Deshalb ist darauf zu achten, daß während des Schreibvorganges der Bereich oder der Bereichszeiger nicht verändert wird.
7. Wird die gleiche Cassette von einem anderen Befehl angesprochen, so wartet das Betriebsprogramm intern so lange, bis der laufende Befehl abgearbeitet ist.
8. Im Befehl "SMWT, MTWS" werden nach der Beendigung der Simultanarbeit die Merker wie folgt gesetzt:

MU	MC	Meldung
1	0	Bandendemarke aufgetreten
0	1	Falsch geschrieben oder Schreibsperr
1	1	Falsch geschrieben und Bandendemarke aufgetreten

MGC	Magnetbandcassettenbefehle	3.12
MGRFR	Vorlauf	0. 4



Wirkung:

1. Der Befehl bewirkt das Vorlaufen des Bandes um $n + 1$ Sätze (maximal 256 Sätze). Während des Vorlaufes wird kontrolliert, ob eine Satzlücke auftritt. In diesem Falle wird der interne Zähler für die Anzahl der Blöcke um 1 vermindert.
2. Eine vorzeitige Beendigung des Befehls kann aus folgenden Gründen auftreten:

Das Band soll um eine bestimmte Anzahl von Sätzen vorgesetzt werden. Es tritt während der Ausführung des Befehls die Bandendemarke auf:

 - a) Das Band spult weiter, die angegebene Anzahl wird vor dem physikalischen Ende erreicht: MU = 1 (Bandendemarke aufgetreten)
MC = 0 (Befehl richtig ausgeführt)
 - b) Das Band spult weiter, die angegebene Anzahl wird nicht erreicht. Das Band spult bis zum physikalischen Ende vor: Die Operation wird abgebrochen: MU = 1 (Bandendemarke aufgetreten)
MC = 1 (Befehl wurde vorzeitig abgebrochen)

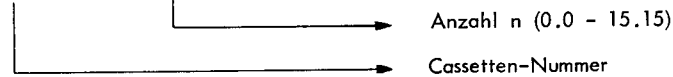
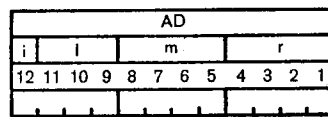
Die Merker werden nur gesetzt, wenn der Befehl "SMWT, MTWS" codiert wurde.

MGC	Magnetbandcassettenbefehle	3.12
MGRFR	Vorlauf	0.4

3. Dieser Befehl wird ebenfalls vorzeitig beendet, wenn eine Satz-
lücke auftritt, die größer als 240 cm ist. Das bedeutet, daß die
Cassetten-Einheit nach einem Satzende etwa 35 sec. nach einem
neuen Satzanfang sucht. Falls keine Zeichen auftreten, wird der
Befehl abgebrochen. Im Befehl "SMWT,MTWS" wird in diesem Falle
der Merker MC = "1" gesetzt (Fehler aufgetreten).
4. Dieser Befehl liest zwar Daten auf dem Magnetband, jedoch erfolgt
keine Veränderung des Eingabebereiches.
5. Der Befehl läuft simultan zum folgenden Befehlsablauf ab. Wird die
Cassette von einem anderen Befehl angesprochen, so wartet das Be-
triebsprogramm intern so lange, bis der Befehl abgelaufen ist.
6. Im Befehl "SMWT,MTWS" werden nach Beendigung der Simultanar-
beit die Merker wie folgt gesetzt:

MU	MC	Meldung
0	0	Vorgang richtig
1	0	Bandendemarke aufgetreten
0	1	Satzlücke länger als 240 cm
1	1	Bandendemarke aufgetreten und Satz- lücke länger als 240 cm

MGC	Magnetbandcassettenbefehle	3.12
MGRBR	Rücklauf	0. 5



Wirkung:

1. Der Befehl bewirkt das Rücklaufen des Bandes um $n + 1$ Sätze (maximal 256 Sätze). Während des Rücklaufes wird kontrolliert, ob eine Satzücke auftritt. In diesem Falle wird der interne Zähler für die Anzahl der Sätze um 1 vermindert.
2. Eine vorzeitige Beendigung des Befehls kann aus folgenden Gründen auftreten:

Das Band soll um eine bestimmte Anzahl von Sätzen zurückgesetzt werden. Es tritt während der Ausführung des Befehls die Bandanfangsmarke auf:

- a) Das Band spult weiter, die angegebene Anzahl wird vor dem physikalischen Anfang erreicht:
 - MU = 1 (Bandanfangsmarke aufgetreten)
 - MC = 0 (Befehl richtig ausgeführt)
- b) Das Band spult weiter, die angegebene Anzahl wird nicht erreicht. Das Band spult bis zum physikalischen Anfang zurück. Die Operation wird abgebrochen:
 - MU = 1 (Bandanfangsmarke aufgetreten)
 - MC = 1 (Befehl wurde vorzeitig abgebrochen)

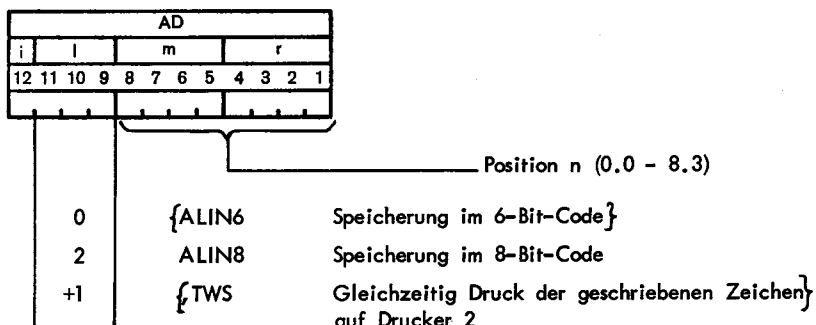
Die Merker werden nur gesetzt, wenn der Befehl "SMWT,MTWS" codiert wurde.

MGC	Magnetbandcassettenbefehle	3.12
MGRBR	Rücklauf	0. 5

3. Dieser Befehl wird ebenfalls vorzeitig beendet, wenn eine Satz-
 lücke auftritt, die größer ist als 240 cm. Das bedeutet, daß die
 Cassetten-Einheit nach einem Satzanfang das Satzende des vorher-
 gehenden Satzes etwa 35 sec. lang sucht. Im Befehl "SMWT,MTWS"
 wird in diesem Falle der Merker MC = "1" gesetzt.
4. Dieser Befehl liest zwar Daten auf dem Magnetband, jedoch erfolgt
 keine Veränderung des Eingabebereiches.
5. Der Befehl läuft simultan zum folgenden Befehlsablauf ab. Wird
 die Cassette von einem anderen Befehl angesprochen, so wartet das
 Betriebsprogramm intern so lange, bis der Befehl abgelaufen ist.
6. Im Befehl "SMWT,MTWS" werden nach Beendigung der Simultan-
 arbeit die Merker wie folgt gesetzt:

MU	MC	Meldung
0	0	Vorgang richtig
1	0	Bandanfangsmarke aufgetreten
0	1	Satzlücke länger als 240 cm
1	1	Bandanfangsmarke aufgetreten und Satzlücke länger als 240 cm

ALIN	Eingabe über alphanumerische Tastatur in ALC-Bereich (alphanumeric input)	3.13
------	---	------



Wirkung:

1. Drucker 1 ist bis Position n-1 zum Drucken freigegeben. Ist die Position des Kugelkopfes größer oder gleich Position n, so erfolgt keine Freigabe des Druckers. Ab Position n können nur noch die Tabulationstaste ⇨, die Wagenaufzugstaste ⇐ oder evtl. eine der Auslösetasten zur Beendigung der Befehlsausführung betätigt werden.
2. Der ALC-Code der geschriebenen Zeichen wird fortlaufend im ALC-Bereich gespeichert, wobei der Inhalt des Bereichszeigers I₃ den Anfangspunkt im Bereich festlegt.

{Speicherung im 6-Bit-Code: Es können nur Zeichen mit einem ALC-Code kleiner als 3.15 gespeichert werden. Bei jedem Zeichen mit einem Code über 3.15 läuft das Programm auf Externfehlerstopp, wobei der Drucker gesperrt ist. Das unzulässige Zeichen wird nicht gespeichert. Nach Drücken der C-Taste rückt der Kugelkopf um eine Stelle nach links, und die Eingabe kann fortgesetzt werden.
 Nach Beendigung der Druckfreigabe wird der Code 3.15 als zusätzliches Zeichen vor dem Endezeichen in den Bereich gebracht werden, um die Umschaltung auf den 8-Bit-Code des Endezeichens zu bewirken. In dem Fall, daß der Code 3.15 nur 2 Bit in der letzten Kernspeicherstelle belegt, werden die restlichen 2 Bit auf Null gesetzt.

	Eingabe über alphanumerische Tastatur in ALC-Bereich (alphanumeric input)	3.13
--	---	------

Speicherung im 8-Bit-Code: Es können alle Zeichen des Druckers geschrieben und gespeichert werden, denn für jedes Zeichen stehen die 8 Bit von zwei Kernspeicherstellen zur Verfügung. Im Gegensatz zur Abspeicherung im 6-Bit-Code entfällt bei Beendigung der Druckfreigabe das Einfügen des Codes 3.15, da eine Umschaltung für das Endezeichen nicht erforderlich ist.

3. Der AD₁-Teil vom Befehl unter Adresse 0.0.0 im Anwenderprogrammblock 0 gibt an, ob die Rücktaste benutzt werden kann. Bei jeder Betätigung der Rücktaste geht der Kugelkopf um eine Schreibstelle zurück, und der Bereichszeiger wird zurückgestellt. Ein Zurückstellen vor die Ausgangsposition der Druckerfreigabe ist jedoch nicht möglich. Wenn die Rücktaste erlaubt ist, kann sie auch noch in Position n benutzt werden, um das in Position n-1 geschriebene Zeichen zu korrigieren.
4. Die Freigabe der alphanumerischen Tastatur wird durch eine der folgenden Tasten beendet, wobei die dazugehörigen Endezeichen gespeichert und die Merker ML, MU, MC entsprechend der folgenden Tabelle gesetzt werden. (Die Beendigung durch eine der Auslösetasten ist nur dann möglich, wenn dies durch entsprechende Belegung des AD-Teils vom Befehl unter Adresse 0.0.0 im Anwenderprogrammblock 0 zugelassen wurde.)

Taste	Endezeichen	ML	MU	MC
Tabulationstaste	6.8	0	1	0
Wagenaufzugstaste	6.12	0	0	1
Auslösetasten	6.11	1	0	0

Die Beendigung durch Tabulationstaste oder Auslösetaste bewirkt keine Bewegung des Kugelkopfes. Nur bei Betätigen der Wagenaufzugstaste findet eine Tabulation in die Ausgangsposition statt (jedoch keine Zeilenschaltung).

ALIN	Eingabe über alphanumerische Tastatur in ALC-Bereich (alphanumeric input)	3.13
------	---	------

5. Die geschriebenen Zeichen können gleichzeitig auf dem Drucker 2 gedruckt werden, wobei deren Ausgangsposition eine andere als die von Drucker 1 sein kann. Die Betätigung der Rücktaste wirkt sich analog auf Drucker 2 aus, und die Beendigung durch Wagenautzugstaste hat ebenfalls eine Tabulation des Kugelkopfes in die Ausgangsposition von Drucker 2 zur Folge.
6. Beim Modell 820/15 kann kein zweiter Drucker angeschlossen werden. Wird jedoch ein zusätzlicher Ausdruck auf diesen Drucker codiert, so wird der Zusatz "TWS" als nicht gegeben betrachtet.
7. Die Druckfarbe kann {für jeden Drucker getrennt} durch einen entsprechenden Druckvorbefehl festgelegt werden. Ohne Druckvorbefehl erfolgt der Ausdruck in schwarz.
8. Der Bereichszeiger I₃ wird bei der Abspeicherung weitergestellt. Nach Beendigung der Eingabe zeigt er auf die dem Endezeichen folgende Kernspeicherstelle im ALC-Bereich.

Beispiel:

Im ALC-Bereich soll ab Speicherwort 22 Stelle 14 im 6-Bit-Code fortlaufend Text gespeichert werden. Der Kugelkopf vom Drucker 1 befindet sich in Position 76.

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
	0 3 11	0	1 6 14		BPX, 22.14
	1				

nachher: I₃

{ 1 11 14 }

entspricht Speicherwort 22, Stelle 14

ALIN	Eingabe über alphanumerische Tastatur in ALC-Bereich (alphanumeric input)	3.13
------	---	------

Danach wird der Eingabebefehl gegeben:

BW-Adresse			Op. Teil			Adreßteil			Symb. Adr.	Bemerkungen
		0	3	13	0	0	5	2		ALIN6,82
		1								

Die Tastatur und der Drucker 1 sind freigegeben. Es wird SYSTEM eingetastet und auf dem Drucker 1 gedruckt (ohne Druckvorbefehl in schwarz). Nachdem anschließend die Tabulationstaste gedrückt wurde, ist die Befehlsausführung beendet und

nachher: I_3

1	12	11
---	----	----

 entspricht Speicherwort 23, Stelle 11

SW 22

																		9	2
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	---

SW 23

10	9	2	5	5	9	14	15	12	6	8									
----	---	---	---	---	---	----	----	----	---	---	--	--	--	--	--	--	--	--	--

Ferner sind die Merker ML = 0, MU = 1, MC = 0.

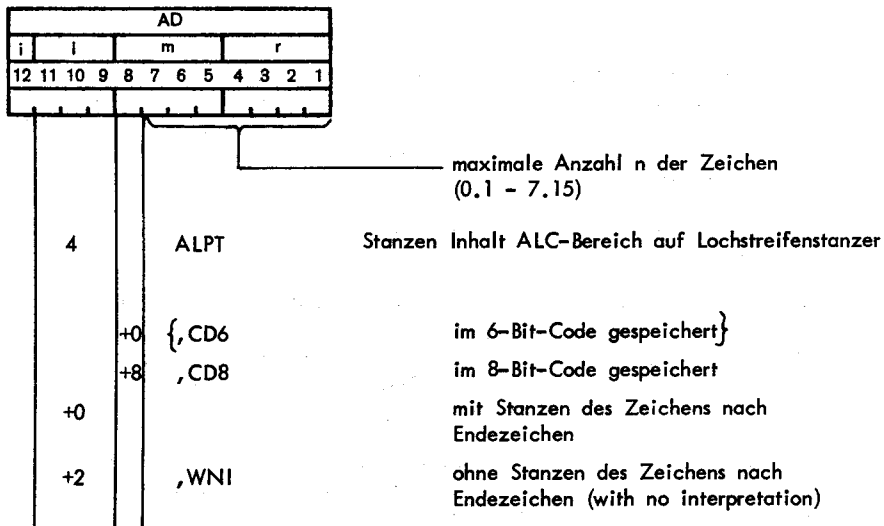
Bemerkungen:

1. In Position 0 ist die erste Druckstelle links, in Position 130 die letzte Druckstelle rechts. Danach gedruckte Zeichen werden alle in Position 130 ausgedruckt.
2. Man beachte, daß das Programm erst dann mit dem nächsten Befehl fortfährt, wenn die Freigabe des Druckers beendet wurde.

ALIN	Eingabe über alphanumerische Tastatur in ALC-Bereich (alphanumeric input)	3.13
------	---	------

3. Beim Befehl {"ALIN6 bzw.} ALIN8 Eingabe in Bereich (3.13)" findet keine Simultanarbeit statt. Nach Beendigung der Befehlsausführung sind die Merker entsprechend obiger Tabelle gesetzt, ohne daß zuvor der Befehl "SMWT Setze Merker nach Beendigung Simultan-Ausgabe (2.12)" gegeben werden muß.
4. Man beachte, daß beim Drucken des Zeichens "-" auf dem Drucker der ALC-Code 0.11 im ALC-Bereich gespeichert wird.
5. Die Taste Zeilenschaltung ~~✱~~ an der alphanumerischen Tastatur bewirkt unabhängig vom Programm eine Weiterschaltung der Druckerwalze. Sie wirkt weder auf die Magnetkontenschächte noch auf die Leporello-Formulartransporte. Die Betätigung der Taste zählt nicht als Eingabe eines Zeichens, d.h. es wird kein Code gespeichert.
6. Man beachte, daß der Bereichszeiger I_3 in I_{1m} immer die interne Speicherwortnummer enthält, die aus der normalen Speicherwortnummer durch Addition von {0.5} [0.4] entsteht.
7. Es ist zu beachten, daß beim Modell 820/15 kein Befehl ALIN6 vorhanden ist. Wird dieser Befehl jedoch angewendet, so wird der weitere Programmablauf gestört. Es erfolgt kein interner Fehlerstopp.

ALPT	Stanzen Inhalt ALC-Bereich auf Lochstreifenstanzer (alphanumeric punch tape)	3.13
------	--	------

Wirkung:

- Der Inhalt des ALC-Bereiches wird fortlaufend auf einem Lochstreifenstanzer ausgegeben, wobei zuvor eine Umcodierung des ALC-Codes in den Lochstreifencode anhand einer Lochstreifencode-Tabelle (PT-Tabelle) erfolgt. Den Anfangspunkt im ALC-Bereich legt der Inhalt des Bereichszeichers $\{3\}$ fest.
- Durch den Befehl "PTF Lochstreifenvorbefehl (3.5)" kann festgelegt werden, ob die Umcodierung in den Lochstreifencode nach Tabelle 1 oder Tabelle 2 und die Ausgabe auf Streifenlocher 1 {oder Streifenlocher 2} erfolgen soll. Ohne Vorbefehl wird nach Tabelle 1 umcodiert und auf Lochstreifenstanzer 1 ausgegeben.

ALPT	Stanzen Inhalt ALC-Bereich auf Lochstreifenstanzer (alphanumeric punch tape)	3.13
------	--	------

3. Die Ausgabe wird beendet, wenn das Umschaltzeichen 3.15 oder eines der Endezeichen (Zeichen mit einem Code größer oder gleich 6.8) auftritt. Ist $n \neq 0$, so werden maximal n Zeichen ausgegeben. Bei Beendigung durch Auftreten eines Umschalt- oder Endezeichens wird in jedem Falle zuerst das Lochstreifen-Endezeichen (PT-Code für 3.15) gelocht. Falls gefordert, folgt im Anschluß daran noch die Ausgabe des gespeicherten Endezeichens im Lochstreifencode.

4. Waren die Zeichen im 6-Bit-Code gespeichert, so bedeutet das Zeichen 3.15 eine Umschaltung auf den 8-Bit-Code des Endezeichens. Der Inhalt der beiden folgenden Kernspeicherstellen wird demnach als Endezeichen interpretiert.

Bei der Speicherung im 8-Bit-Code entfällt im allgemeinen das Zeichen 3.15 vor dem Endezeichen, da eine Umschaltung nicht erforderlich ist. Tritt es trotzdem auf, so wird das folgende Zeichen als Endezeichen interpretiert.

5. Je nachdem, ob die Ausgabe durch Auftreten eines Endezeichens oder durch Erreichen der Anzahl n beendet wurde, werden die folgenden Merkerstellungen intern zwischengespeichert:

Beendigung	ML	MU	MC
Ende durch Endezeichen	0	0	0
Ende durch Erreichen der Anzahl n	0	1	0

Im Befehl "SMWT Setze Merker nach Ende Simultan-Ausgabe (2.12)" wird auf das Ende der Ausgabe gewartet. Im Anschluß daran erfolgt das Setzen der Merker entsprechend obiger Tabelle.

ALPT	Stanzen inhalt ALC-Bereich auf Lochstreifen- stanzer (alphanumeric punch tape)	3.13
------	---	------

6. Der Bereichszeiger I_3 zählt während der Ausgabe mit. Nach Beendigung durch Endezeichen zeigt er auf die dem Endezeichen folgende Kernspeicherstelle; nach Beendigung durch Erreichen der Anzahl n zeigt er im Bereich auf die dem zuletzt ausgegebenen Zeichen folgende Kernspeicherstelle.
7. Beim Modell 820/15 darf nicht der Zusatz "CD6" verwendet werden, da die Daten nicht im 6-Bit-Code gespeichert werden können.
8. Weiterhin darf auch im Vorbefehl die Angabe "Ausgabe auf Lochstreifenstanzer 2" nicht erfolgen.

ALPT	Lochen Inhalt ALC-Bereich auf Lochstreifenstanzer (alphanumeric punch tape)	3.13
------	---	------

Beispiel:

Aus dem ALC-Bereich sollen die ab Speicherwort 34 Stelle 12 gespeicherten Zeichen auf Lochstreifenstanzer 2 ausgegeben werden. Die Umcodierung in den Lochstreifencode soll nach Tabelle 1 erfolgen.
Zunächst wird der Bereichszeiger I_3 des ALC-Bereiches geladen:

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
	0	3	11 0	2 2 12	BPX, 34.12
	1				

nachher: I_3

2	7	12
---	---	----

 entspricht Speicherwort 34, Stelle 12

Danach folgen Lochstreifenvorbefehl und Lochstreifenhauptbefehl:

vorher: SW 34

														x	x	x	x
--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	---	---	---

 SW 35

x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	6	8	
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
	0	3	5 0	6 2 0	PTF, PTS, TBL1
	1	3	13 0	4 9 11	ALPT, CD8, 27
	2				

nachher: I_3

2	8	14
---	---	----

 entspricht Speicherwort 35, Stelle 14

Auf Lochstreifenstanzer 2 werden nacheinander 10 Zeichen in dem in Tabelle 1 angegebenen Lochstreifencode ausgegeben. Als 9. Zeichen trat das Endezeichen 6,8 auf, welches den Abbruch bewirkte. Die beiden zuletzt ausgegebenen Zeichen sind das Lochstreifen-Endezeichen (PT-Code für 3,15) und das Interpretierungszeichen (PT-Code für 6,8). Die Merker ML, MU, MC werden durch diesen Befehl noch nicht verändert. Die darauffolgenden Befehle dürfen I_3 und die angesprochenen Stellen der Speicherworte 34 und 35 nicht verändern.

ALPT	Lochen Inhalt ALC-Bereich auf Lochstreifenstanzer (alphanumeric punch tape)	3.13
------	---	------

Wurde auch kein Ausgabebefehl für das 2. Simultangerät gegeben, so ist nach dem Befehl

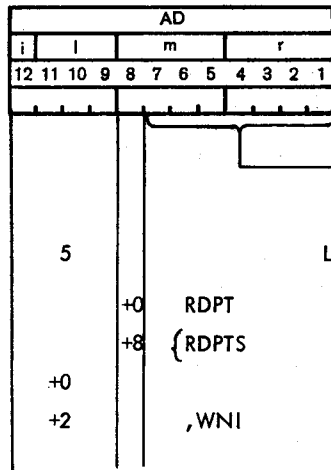
BW-Adresse		Op. Teil			Adreßteil			Symb. Adr.	Bemerkungen
		0	2	12	0	0	10	1	SMWT, MPTS
		-							

die Ausgabe auf Lochstreifenstanzer 2 beendet und die Merker sind $ML = 0$, $MU = 0$, $MC = 0$.

Bemerkungen:

- Der Befehl "ALPT Stanzen Inhalt ALC-Bereich (3.13)" wird simultan zum weiteren Befehlsablauf ausgeführt. Vor einer neuen Ausgabe auf Simultangerät 2 wartet intern das Betriebsprogramm auf die Beendigung der Operation. Man beachte jedoch, daß der angesprochene Teil des ALC-Bereiches und der Bereichszeiger I_3 erst nach Beendigung der Ausgabe verändert werden dürfen (vgl. "Warte- und Merkersetzbefehl (2.12)").
- Ist der Text im 6-Bit-Code gespeichert, so steht der Bereichszeiger nur bei jedem zweiten Zeichen richtig. Darauf ist besonders dann zu achten, wenn die Ausgabe durch Erreichen der Anzahl n beendet wurde.
- Man beachte, daß der Bereichszeiger I_3 in I_{1m} immer die interne Speicherwortnummer enthält, die aus der normalen Speicherwortnummer durch Addition von $\{0.3\}$ $[0.4]$ entsteht (vgl. Beispiel und den Befehl "BPX Laden ALC-Bereichszeiger (3.11)").
- Die zur Umcodierung des ALC-Codes in den Lochstreifencode erforderlichen Codetabellen müssen sich im Anwenderprogrammblock 0 befinden. Die Anfangsadresse von Tabelle 1 ist im AD-Teil des Befehls unter Adresse 0.0.3, die von Tabelle 2 im AD-Teil des Befehls unter Adresse 0.0.4 im Anwenderprogrammblock 0 anzugeben.

RDPT	Lesen mit Lochstreifenleser in Bereich (read punch tape)	3.13
------	---	------



maximale Anzahl n der Zeichen
(0.1 - 7.15)

Lesen in PCT-Eingabebereich ab Bereichsanfang

+0 RDPT mit Lochstreifenleser 1
+8 {RDPTS mit Lochstreifenleser 2}

+0 mit Lesen des Zeichens nach Endezeichen
+2 ,WNI ohne Lesen des Zeichens nach Endezeichen
(with no interpretation)

Wirkung:

1. Auf dem gewählten Lochstreifenleser werden fortlaufend Zeichen gelesen und im PCT-Eingabebereich beginnend mit Bereichsanfang im Eingabecode des Lochstreifens gespeichert. Jedem gelesenen Zeichen entsprechen dabei zwei Kernspeicherstellen im Bereich.
2. Die Eingabe wird beendet, wenn das Lochstreifen-Endezeichen (PT-Code für 3.15) auftritt. Ist $n \neq 0$, so werden maximal n Zeichen eingelesen. Bei Beendigung durch Auftreten des Endezeichens kann gewählt werden, ob das folgende Zeichen ebenfalls noch gelesen und gespeichert werden soll.
3. Durch den Befehl "PTF Lochstreifenvorbefehl (3.5)" ist festzulegen, ob der Code des Endezeichens aus Tabelle 1 oder Tabelle 2 zu entnehmen ist. Ferner ist bei Verarbeitung von 5-Kanal-Lochstreifen festzulegen, ob das erste Zeichen als Buchstaben- oder Ziffernzeichen zu interpretieren ist.

RDPT	Lesen mit Lochstreifenleser in Bereich (read punch tape)	3.13
------	---	------

4. Bei 5-Kanal-Lochstreifen erfolgt während des Einlesens eine Umcodierung des 5-Bit-Codes in einen 6-Bit-Code, der in Abhängigkeit von den Umschaltzeichen BU und ZI erzeugt wird. Diese Zeichen werden nicht gespeichert und zählen auch nicht bei der Anzahl gelesener Zeichen mit. Das Endezeichen muß sowohl ein BU- als auch ein ZI-Zeichen sein.
5. Es wird der Merker $MU = 1$ gesetzt, wenn $n \neq 0$ Zeichen gelesen wurden, ohne daß das Endezeichen auftrat, andernfalls ist $MU = 0$.
6. Der Eingabebereichszeiger des PCT-Eingabebereiches zählt während des Einlesens mit. Nach Beendigung zeigt er auf die dem letzten Zeichen zugeordnete Stelle.

Beispiel:

Mit Lochstreifenleser 1 sollen maximal 100 Zeichen in den ab Speicherwort 17 beginnenden PCT-Eingabebereich eingelesen werden.

BW-Adresse	Op. Teil			Adreßteil				Symb. Adr.	Bemerkungen
	0	2	11	0	0	9	1		PBG, RD, 17
	1	3	5	0	0	10	0		PTF, TBL2
	2	3	13	0	7	6	4		RDPT, WNI, 100

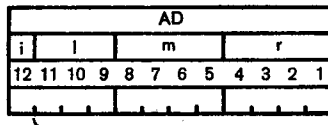
Als 82. Zeichen tritt das Endezeichen auf, d.h. der in Tabelle 2 für 3.15 angegebene Lochstreifencode. Die Zeichen sind von Speicherwort 17 Stelle 0 bis einschließlich Speicherwort 27 Stelle 3 gespeichert. Der Eingabebereichszeiger steht anschließend auf 82, und es ist $MU = 0$.

RDPT	Lesen mit Lochstreifenleser in Bereich (read punch tape)	3.13
------	---	------

Bemerkungen:

1. Wird nacheinander in denselben Bereich eingelesen, so braucht der Befehl "PBG, RD Bereichsanfang setzen (2.11)" nur einmal gegeben zu werden.
2. Das Programm läuft auf Datenträgerfehlerstopp, wenn sich kein Lochstreifen im Leser befindet oder wenn der Lesevorgang fehlerhaft durchgeführt wurde. Nach Drücken der F-Taste wird der Lesevorgang wiederholt.
3. Es ist beim Modell 820/15 darauf zu achten, daß der Lochstreifenleser 2 nicht angesprochen werden darf. Der Befehl "RDPTS" ist im Befehlsvorrat dieses Modells nicht vorhanden.

OPX	OP- und AD _i -Teil nach Indexregister (operation code to indexregister)	3.14
-----	---	------



Befehlsadresse (0.0.0 - 7.15.15)

Wirkung:

Der OP- und AD_i-Teil des Befehls unter der Adresse AD_{lmr} werden in ein Indexregister gebracht. Dabei entsprechen

- Bit 1 des Indexregisters dem AD_i-Teil und
- Bit 2 bis Bit 7 des Indexregisters dem OP-Teil.

Die restlichen Bit 8 bis Bit 11 des Indexregisters werden gelöscht.

Beispiel:

Der Befehl "OPX OP-Teil und AD_i-Teil nach I (3.14)" wird indiziert gegeben:

vorher: I₂

0	1	6
---	---	---

BW-Adresse			Op. Teil		l	Adreßteil			Symb. Adr.	Bemerkungen
		0	2	2	0	0	2	1		XF, 2.1
		1	3	14	1	3	10	2		OPX, XI, ZIEL

Unter der Adresse 3.11.8 steht der Befehl:

BW-Adresse			Op. Teil		l	Adreßteil			Symb. Adr.	Bemerkungen
3	10	2							ZIEL	
		6								
		7								
3	11	8	0	5	1	1	1	3		AD, XI, 17.3

nachher: I₁

0	0	11
---	---	----

OP	AD _i	Symb	Konst	OP	AD _i	Symb	Konst	OP	AD _i	Symb	Konst
0.0	0	Z0	0	1.8	0	Z48	48	3.0	0	F96	96
0.0	1	Z1	1	1.8	1	Z49	49	3.0	1	F97	97
0.1	0	Z2	2	1.9	0	F50	50	3.1	0	F98	98
0.1	1	Z3	3	1.9	1	F51	51	3.1	1	F99	99
0.2	0	Z4	4	1.10	0	F52	52	3.2	0	H100	100
0.2	1	Z5	5	1.10	1	F53	53	3.2	1	H101	101
0.3	0	Z6	6	1.11	0	F54	54	3.3	0	H102	102
0.3	1	Z7	7	1.11	1	F55	55	3.3	1	H103	103
0.4	0	Z8	8	1.12	0	F56	56	3.4	0	H104	104
0.4	1	Z9	9	1.12	1	F57	57	3.4	1	H105	105
0.5	0	Z10	10	1.13	0	F58	58	3.5	0	H106	106
0.5	1	Z11	11	1.13	1	F59	59	3.5	1	H107	107
0.6	0	Z12	12	1.14	0	F60	60	3.6	0	H108	108
0.6	1	Z13	13	1.14	1	F61	61	3.6	1	H109	109
0.7	0	Z14	14	1.15	0	F62	62	3.7	0	H110	110
0.7	1	Z15	15	1.15	1	F63	63	3.7	1	H111	111
0.8	0	Z16	16	2.0	0	F64	64	3.8	0	H112	112
0.8	1	Z17	17	2.0	1	F65	65	3.8	1	H113	113
0.9	0	Z18	18	2.1	0	F66	66	3.9	0	H114	114
0.9	1	Z19	19	2.1	1	F67	67	3.9	1	H115	115
0.10	0	Z20	20	2.2	0	F68	68	3.10	0	H116	116
0.10	1	Z21	21	2.2	1	F69	69	3.10	1	H117	117
0.11	0	Z22	22	2.3	0	F70	70	3.11	0	H118	118
0.11	1	Z23	23	2.3	1	F71	71	3.11	1	H119	119
0.12	0	Z24	24	2.4	0	F72	72	3.12	0	H120	120
0.12	1	Z25	25	2.4	1	F73	73	3.12	1	H121	121
0.13	0	Z26	26	2.5	0	F74	74	3.13	0	H122	122
0.13	1	Z27	27	2.5	1	F75	75	3.13	1	H123	123
0.14	0	Z28	28	2.6	0	F76	76	3.14	0	H124	124
0.14	1	Z29	29	2.6	1	F77	77	3.14	1	H125	125
0.15	0	Z30	30	2.7	0	F78	78	3.15	0	H126	126
0.15	1	Z31	31	2.7	1	F79	79	3.15	1	H127	127
1.0	0	Z32	32	2.8	0	F80	80				
1.0	1	Z33	33	2.8	1	F81	81				
1.1	0	Z34	34	2.9	0	F82	82				
1.1	1	Z35	35	2.9	1	F83	83				
1.2	0	Z36	36	2.10	0	F84	84				
1.2	1	Z37	37	2.10	1	F85	85				
1.3	0	Z38	38	2.11	0	F86	86				
1.3	1	Z39	39	2.11	1	F87	87				
1.4	0	Z40	40	2.12	0	F88	88				
1.4	1	Z41	41	2.12	1	F89	89				
1.5	0	Z42	42	2.13	0	F90	90				
1.5	1	Z43	43	2.13	1	F91	91				
1.6	0	Z44	44	2.14	0	F92	92				
1.6	1	Z45	45	2.14	1	F93	93				
1.7	0	Z46	46	2.15	0	F94	94				
1.7	1	Z47	47	2.15	1	F95	95				

	SYSTEMBESCHREIBUNG	
--	--------------------	--

1. Einleitung

2. Allgemeine Übersicht
 - 2.1. Aufbau
 - 2.1.1. Zentraleinheit
 - 2.1.2. Randeinheiten
 - 2.1.2.1. Randeinheiten für Modell 820/15
 - 2.1.2.2. Randeinheiten für die Modelle 820/25 und 820/35
 - 2.2. Programmierung
 - 2.2.1. Programmiersprachen
 - 2.2.2. Anwenderprogramm
 - 2.2.3. Betriebsprogramme
 - 2.2.3.1. Betriebsprogramme für das Modell 820/15
 - 2.2.3.2. Betriebsprogramme für die Modelle 820/25 und 820/35
 - 2.2.4. Testprogramme
 - 2.2.5. Assembler
 - 2.2.6. Dienstprogramme

3. Aufbau und Funktion der Bauteile
 - 3.1. Rechner mit Standard Ein/Ausgabe-Einheit 154
 - 3.1.1. Rechner
 - 3.1.2. Standard Ein/Ausgabe-Einheit 154
 - 3.2. Zentralspeicher
 - 3.2.1. Allgemeines
 - 3.2.2. Festspeicher
 - 3.2.2.1. Stäbchenfestspeicher
 - 3.2.2.2. Ringkernspeicher
 - 3.2.2.3. Speicheransteuerung
 - 3.2.2.4. Adressierung beim 177er Festspeicher
 - 3.2.2.5. Einsatz der Festspeicher
 - 3.2.2.6. Einteilung des Befehlswortes im Anwenderprogramm
 - 3.2.3. Magnetkernspeicher
 - 3.2.3.1. Allgemeines
 - 3.2.3.2. Einteilung des Magnetkernspeichers
 - 3.2.3.3. Speicherworte mit numerischem Inhalt
 - 3.2.3.4. Speicherworte mit alphanumerischem Inhalt
 - 3.2.3.5. Speichertypen
 - 3.2.3.6. Aufbau, interne und externe Adressierung beim Modell 820/15
 - 3.2.3.7. Aufbau, interne und externe Adressierung bei den Modellen 820/25 und 820/35

	SYSTEMBESCHREIBUNG	
--	--------------------	--

- 3.3. Ein/Ausgabe-Einheit
- 3.3.1. Allgemeines
- 3.3.2. Arten der Ein/Ausgabe-Einheiten
- 3.4. Chassis
- 3.4.1. Allgemeines
- 3.4.2. Chassis-Arten
- 3.4.3. Chassis für Modell 820/15
- 3.4.4. Chassis für Modell 820/25
- 3.4.4.1. Chassis 207
- 3.4.4.2. Chassis 208
- 3.4.5. Chassis für Modell 820/35
- 3.5. Netzteil 109
- 3.6. Umschalt- und Verbindungsplatten
- 3.6.1. Modell 820/15
- 3.6.2. Modell 820/25 und 820/35
- 3.7.1. Kabel
- 3.8. Numerische Tastatur 017
- 3.8.1. Funktion und Aufbau
- 3.8.2. Arbeitsweise
- 3.8.3. Internationale Zehnerblochtastatur
- 3.8.3.1. Zifferntasten 0-9
- 3.8.3.2. C-Taste
- 3.8.4. Das Einschalten der Anlage
- 3.8.5. F-Taste
- 3.8.6. Anzeigelampen
- 3.8.6.1. Grüne Lampe
- 3.8.6.2. Rote Lampe
- 3.8.6.3. Orange Lampe
- 3.8.6.4. Gelbe Lampe
- 3.8.6.5. Weiße Lampe
- 3.8.6.6. Rote und orange Lampe
- 3.9. Alphanumerische Tastatur
- 3.9.1. Allgemeines
- 3.9.2. Sonderfunktion von verschiedenen Tasten
- 3.10. Drucker
- 3.10.1. Bestandteile
- 3.10.2. Arbeitsweise
- 3.10.3. Positionierung
- 3.10.4. Papiertransport
- 3.10.4.1. Walze
- 3.10.4.2. Vorschubeinrichtung

	SYSTEMBESCHREIBUNG	
--	--------------------	--

- 4. Systemkonfiguration
- 4.1. Übersicht
- 4.2. Modell 820/15
 - 4.2.1. Kleinste Konfiguration
 - 4.2.2. Größte Konfiguration
 - 4.2.2.1. Fakturier-und Abrechnungs-Computer
 - 4.2.2.2. Magnetkonten-Computer in Verbindung mit Magnetbandcassetten-Einheiten und mit Magnetkernspeicher 165
 - 4.2.2.3. Magnetkonten-Computer in Verbindung mit Lochkarten-und Lochstreifen-Rändeinheiten sowie Magnetkernspeicher 166
- 4.3. Modell 820/25
 - 4.3.1. Vorbemerkung
 - 4.3.2. Chassis 207, maximale Konfiguration als Magnetkonten-Computer, fest programmierbar
 - 4.3.3. Chassis 208, kleinste Konfiguration als Fakturier-und Abrechnungs-Computer, frei programmierbar
 - 4.3.3.1. Monitor-Anlage 820/25
- 4.4. Modell 820/35
 - 4.4.1. Übersicht
 - 4.4.2. Maximale Ausstattung als Magnetkonten-Computer
 - 4.4.3. Monitor-Anlage 820/35
 - 4.4.4. Assembler-Monitor-Anlage
- 4.5. Weitere Rändeinheiten

	SYSTEMBESCHREIBUNG	
--	--------------------	--

1. Einleitung

Die NIXDORF Computer Serie 820 werden in verschiedenen Anwendungsbereichen eingesetzt, und zwar als:

- Fakturier-, Abrechnungs- und Magnetkonten-Computer,
- Computer zur Datenerfassung und Datenfernübertragung,
- Prozeßrechner,
- Bakkalaureus, d.h. als Computer im rechnerunterstützten Unterricht.

Die vorliegende Broschüre befaßt sich ausschließlich mit dem Einsatz der NIXDORF Computer Serie 820 als Fakturier-, Abrechnungs- und Magnetkonten-Computer.

Es gibt folgende Modellbezeichnungen:

- Modell 820/15,
- Modell 820/25,
- Modell 820/35.

Alle Anlagen dieser Serie lassen sich als Fakturier- und Abrechnungs-Computer oder als Magnetkonten-Computer einsetzen.

Die Modelle der Serie 820 unterscheiden sich aufgrund der Größe der Chassis und der Betriebsprogrammarten. Daraus resultieren die unterschiedlichen Ausbaumöglichkeiten des Zentralspeichers und der Randeinheiten.

Die Grundausstattung besteht aus:

- dem Rechner (bei allen Anlagen derselbe Rechnertyp),
- einer mit dem Rechner verbundenen Standard Ein/Ausgabe-Einheit,
- einer numerischen Eingabe-Tastatur,
- einer alphanumerischen Eingabe-Tastatur,
- dem Serialdrucker.

Der Ausbau einer Anlage auf die einem Problem zugeschnittene Größe erfolgt durch Auswahl von:

- Chassis,
- Magnetkernspeicher,
- Festspeicher,
- Betriebsprogramm,
- Ein/Ausgabe-Einheiten,
- Randeinheiten.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Alle Modelle der Serie 820 lassen sich mit unterschiedlichen Magnetkernspeichern ausstatten. Die minimale Konfiguration besteht aus 16 und reicht bis zu 1024 Speicherworten. Das entspricht einer maximalen freien Kapazität von 8 K Bytes. Festspeicher gibt es ab einer Kapazität von 2048 Befehlsworten, Magnetkernspeicher ab 16 Speicherworten. In allen Festspeichern zusammen lassen sich bis zu 16000, in allen Magnetkernspeichern zusammen bis zu 6000 Befehle unterbringen.

Neben den Randeinheiten zur direkten manuellen Daten-Ein/Ausgabe, die zur Grundausstattung gehören, stehen für die Anlagen der NIXDORF-Serie 820 auch Randeinheiten zur maschinellen Daten-Ein/Ausgabe zur Verfügung:

- automatischer Konteneinzug,
- Magnetkonten-Einheiten, mit manuellem oder mit automatischem Konteneinzug,
- Lochkarten-Stanzer/Leser,
- Lochstreifen-Stanzer/Leser,
- Magnetbandcassetten-Einheit.

Der Anschluß dieser Randeinheiten erfolgt über Ein/Ausgabe-Einheiten.

Bei der Serie 820 ist der modulare Aufbau von Hard- und Software bis ins kleinste Detail konsequent verwirklicht. Aufgrund des wohldurchdachten Baukastenprinzips und eines günstigen Preis-/Leistungsverhältnisses eignen sich die Computer der Serie 820 zum Einsatz in Betrieben jeder Größe, sowie in Behörden und Verwaltungen.

Für das NIXDORF System 820 gibt es folgende Software:

- Herstellersoftware,
 - . Betriebsprogramme,
 - . Testprogramme,
 - . Dienstprogramme,
 - . Assembler,
- Anwendersoftware,
 - . im Assembler-Code vorhandene Lösungen von Anwendungsproblemen.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

2. Allgemeine Übersicht

2.1. Aufbau

Alle Modelle sind nach dem gleichen Prinzip aufgebaut und bestehen aus:

- der Zentraleinheit und
- den Randeinheiten.

2.1.1. Zentraleinheit

Die Zentraleinheit besteht aus:

- Rechner,
- Zentralspeicher,
- Steereinheiten.

Sämtliche Funktionseinheiten arbeiten mit eigener Taktung, und zwar asynchron.

Der Rechner ist ein 12-Bit-Parallelrechner und arbeitet in Verbindung mit verschiedenen Registern und mit einem eigenen Steuerwerk (Steuerlogik).

Der Zentralspeicher ist aufgliedert in:

- Festspeicher,
- Magnetkernspeicher.

Zu den Steereinheiten gehören sowohl ein Teil des Betriebsprogramms als auch die Ein/Ausgabe-Einheiten. Im Betriebsprogramm sind die Steuerfunktionen enthalten. Eine Ein/Ausgabe-Einheit ist serienmäßig an jedem Rechner fest angebaut und dient zum Anschluß folgender standardmäßiger Randeinheiten:

- Numerische Tastatur,
- Alphanumerische Tastatur,
- Serialdrucker.

Die Ansteuerung dieser Randeinheiten ist in jedem Betriebsprogramm enthalten.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

2.1.2. Randeinheiten

An alle Modelle der Serie 820 lassen sich außer den standardmäßigen Randeinheiten (Tastaturen und ein Serialdrucker) weitere Randeinheiten anschließen.

2.1.2.1. Randeinheiten für Modell 820/15

Bei der 820/15 können, wenn keine Magnetkonten-Einheit angeschlossen ist, zu den standardmäßigen Randeinheiten hinzukommen:

- ein automatischer Konteneinzug,
- ein Lochkartenstanzer,
- ein Lochkartenleser,
- ein Lochstreifenstanzer oder ein Lochstreifenkartenstanzer,
- ein Lochstreifenleser oder ein Lochstreifenkartenleser,
- ein oder zwei Magnetbandcassetten-Einheiten.

Ist eine Magnetkonten-Einheit (nur ein Einzugsschacht) für das Modell 820/15 vorgesehen, dann lassen sich außer dieser noch folgende Randeinheiten anschließen:

- ein Lochkartenstanzer,
- ein Lochkartenleser,
- ein Lochstreifenstanzer oder ein Lochstreifenkartenstanzer, oder ein bzw. zwei Magnetbandcassetten-Einheiten.
- ein Lochstreifenleser oder ein Lochstreifenkartenleser.

2.1.2.2. Randeinheiten für die Modelle 820/25 und 820/35

Beim Modell 820/25 bestehen ähnliche Anschlußmöglichkeiten wie beim Modell 820/35. Die Anzahl der anschließbaren Randeinheiten hängt ab von:

- der Anzahl der Ein/Ausgabe-Plätze im Chassis und
- vom jeweils vorhandenen Betriebsprogramm.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Außer der Grundausstattung gibt es Anschlußmöglichkeiten für:

- ein zweiter Serialdrucker
- ein oder zwei Lochstreifenstanzer oder Lochstreifenkartenstanzer,
- ein oder zwei Lochstreifenleser oder Lochstreifenkartenleser,
- ein oder zwei Lochkartenstanzer,
- ein oder zwei Lochkartenleser,
- ein bis vier Magnetbandcassetten-Einheiten.

An den Serialdrucker lassen sich entweder ein automatischer Konteneinzug (ein oder zwei Einzugschächte) oder eine Magnetkonten-Einheit (zwei Einzugschächte), anschließen.

2.2. Programmierung

2.2.1. Programm-Sprachen

Die Programmierung ist einheitlich. Lediglich für das Modell 820/15 gelten bezüglich des Befehlsreportoirs und der unterschiedlichen Funktionen einiger Befehle gewisse Besonderheiten, die den Listen der Befehle für Modell 820/15 zu entnehmen sind. Die Programmierung kann sowohl in der Assembler-Sprache als auch in der Maschinensprache erfolgen. Das Programmieren in der Assembler-Sprache ist einfach und erfordert wesentlich weniger Aufwand als das Programmieren in der sedezimalen Maschinensprache. Sämtliche Angaben, die für einen vollständigen Befehl notwendig sind, können durch mnemotechnische Symbole angegeben werden. Alle Assemblerbefehle sind klar gegliedert und lassen sich in übersichtlichen und einfach zu handhabenden Formularen eintragen.

Ein Übersetzerprogramm (Assembler) wandelt die Symbolsprache (Quellprogramm) in die Maschinensprache (Objektprogramm) um.

Da in der Assembler-Sprache die Adressen symbolisch angegeben werden, lassen sich Programmteile oder einzelne Befehle an beliebigen Stellen des Programms einfügen und ersparen dem Programmierer langwierige und umständliche Adreßrechnungen.

Nur die Assembler-Sprache ermöglicht den sinnvollen Aufbau einer Programmbibliothek.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

2.2.2. Anwenderprogramm

Der Anwender kann die Befehle in der Maschinen-Sprache seinen Erfordernissen entsprechend fest verdrahten oder frei programmieren.

Bei der ersten Speicherart werden Festspeicher (Read-only-memory, "Nur-Lese-Speicher") benutzt, die den Datentransport nur in einer Richtung gestatten.

Magnetkernspeicher lassen im Gegensatz zum Festspeicher den Datenübertrag in beiden Richtungen (Lesen und Schreiben) zu. Das bedeutet, daß sich Befehle mit Hilfe eines Ladeprogramms in den Kernspeicher laden lassen.

Da bei NIXDORF beide Speicherarten gleichzeitig eingesetzt werden können, spricht man von fest- und/oder freiprogrammierten Anlagen.

2.2.3. Betriebsprogramme

Die Betriebsprogramme sind auf Festspeichern untergebracht und steuern die Arbeitsweise einer Anlage. Durch sie sind die Befehle des Anwenderprogramms festgelegt.

Man spricht auch von einer mikroprogrammierten Anlage, d.h. mit Hilfe einfachster Befehle (Mikrobefehle), die der Rechner unmittelbar interpretiert, wird eine komfortable Maschinensprache realisiert, die eine gewisse Ausrichtung auf einen Anwenderkreis besitzen kann.

2.2.3.1. Betriebsprogramm für das Modell 820/15

Für das Modell 820/15 gibt es zwei Betriebsprogramme:

- das MFA1, das bei Magnetkernspeichern mit 16, 32, 64, 128 oder 256 Speicherworten in Frage kommt, und
- das MFAGS1, das in Verbindung mit dem Magnetkernspeicher mit 512 Speicherworten einzusetzen ist.

Für die Steuerung von Randeinheiten werden Ergänzungen zum Betriebsprogramm benötigt. Diese werden mit Moduln bezeichnet.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Sie sind auf vorgeschriebenen Plätzen auf dem zweiten Programmträger des Festspeichers, der auch den ersten Programmträger mit dem Betriebsprogramm enthält, zu verdrahten. Dieser zweite Programmträger enthält sowohl Befehle des Betriebs- als auch des Anwenderprogramms.

Die nachfolgende Aufstellung zeigt die Speicherbelegung der vorhandenen Moduln:

Befehlswordadressen für das Anwenderprogramm	Modul	Anzahl Befehle
8. 0. 8 - 9. 9.11	MC	404
8. 9. 2 - 8.12. 3	ST	50
8. 9. 2 - 8.14.10	IN	89
9. 9.12 -11.10. 7	PC	524
10.14. 9 -12.15. 9	PT	529
11.10.12 -12. 7. 5	LR	202
12.15.10 -14.11.10	TC	449
15. 5. 8 -15. 9.15	GP	72
15.10. 0 -15.15.15	TW	96

Die Moduln überlappen sich teilweise, da zur Ansteuerung mancher Randeinheiten Befehle benötigt werden, die in den gleichen Programm-routinen enthalten sind.

Alle Befehlswordadressen, die nicht von Moduln belegt sind, stehen dem Anwenderprogramm zur Verfügung; es ist jedoch zweckmäßig, die letzten (freien) 16 Befehlswordadressen für nachträgliche Erweiterungen des Betriebsprogrammes nicht zu belegen.

Der TW-Modul muß immer vorhanden sein.

Der GP-Modul ist immer dann erforderlich,

- wenn Befehle im Magnetkernspeicher vorgesehen sind oder
- wenn eines der nachstehend aufgeführten Moduln erforderlich ist:
MC, PC, PT, LR, TC.

Die Moduln ST und IN kommen ohne den GP-Modul aus.

	SYSTEMBESCHREIBUNG	
--	---------------------------	--

2.2.3.2. Betriebsprogramme für die Modelle 820/25 und 820/35

In der folgenden Übersicht werden die Betriebsprogramme mit ihren Besonderheiten aufgeführt:

820/25 und 820/35	Grundausstattung	Automatischer Konteneinzug	Magnetknoten-Einheit	2. Drucker	Lochkartenstanzer	Lochkartenleser	Lochstreifenstanzer	Lochstreifenleser	Magnetband-Cassette	Listgerät 720
MSK Z1 + MSK Z2/LU	x		x	x						
MSK Z1 + MSK Z2/LU + MSK Z3	x	x	x	x	x	x	x	x		
MSK Z1 + MSK Z2/LU + MSK Z3/CASS	x	x	x	x	x	x	x	x	x	
MSK Z1 + MSK Z2/LU + MSK Z3/LIST	x	x	x	x	x	x	x	x		x
MSK Z1 + MSK Z2/LU + MSK Z3/LIST E	x	x	x	x	x	x	x	x		x
MSK Z1 + SK ZA2	x	x		x	x	x	x	x		
MSK Z1 + SK ZA2 + MSK Z3/CASS	x	x		x	x	x	x	x	x	
MSK Z1 + SK ZA2 + MSK Z3/LIST	x	x		x	x	x	x	x		x
MSK Z1 + SK ZA2 + MSK Z3/LIST E	x	x		x	x	x	x	x		x
* MSK Z1 + SACK2	x	x		x	x		x		x	
SLS1 + MSK Z2/LU	x		x							x
SLS1 + MSK Z2/LU + MSK Z3	x	x			x	x	x	x		x
SLS1 + MSK Z2/LU + MSK Z3/CASS	x	x			x	x	x	x	x	x
SLS1 + SK ZA2	x	x			x	x	x	x		x
SLS1 + SK ZA2 + MSK Z3/CASS	x	x			x	x	x	x	x	x
* SLS1 + SACK2	x	x			x	x		x		x
nur 820/35	MSK Z4 MSK Z5 - Assembler Die Bestückung mit MSK Z4 und MSK Z5 ist nur in Verbindung mit MSK Z6 möglich.									
	MSK Z6 - Testprogramme } - Dienstprogramme } Monitor									

* nur 2 Magnetband-Cassetten möglich

	SYSTEMBESCHREIBUNG	
--	--------------------	--

2.2.4. Testprogramme

Diese Programme unterstützen das Austesten von Anwenderprogrammen, Durch sie ist es z.B. möglich, das Anwenderprogramm an bestimmten Stellen zu stoppen oder im Einzelschritt zu durchlaufen.

2.2.5. Assembler

Wie schon hingewiesen wurde (2.2.1.), dient der Assembler zum Umwandeln von Programmen, die im symbolischen Code geschrieben sind.

2.2.6. Dienstprogramme

Mit den Dienstprogrammen (Utilities) wurde eine Reihe zusammengehöriger Programme geschaffen, die häufig auftretende Probleme auf einheitliche Weise lösen.

Sie erleichtern das Programmieren und ermöglichen dem Anwender ein wirkungsvolles Arbeiten mit den Anlagen der Serie 820.

Es gibt grundsätzlich zwei Arten von Dienstprogrammen, und zwar:

- Dienstprogramme, die im Betriebsprogramm der Monitorversion (MSKZ 6) zusammengefaßt sind (siehe Datenverarbeitungssystem 820, MONITOR), und
- Dienstprogramme, die im Anwenderprogramm stehen.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Für die zweite Gruppe bietet NIXDORF folgende Dienstprogramme in Assemblerkarten an (siehe auch Datenverarbeitungssystem 820, DIENSTPROGRAMME):

DIENST-PROGRAMME	VERARBEITUNGS-RICHTUNG		NAME	
Kernspeicher ↔ Tastatur	--	IN/OUT	----	UTTW
Kernspeicher → Serialdrucker				
Kernspeicher ↔ Lochkarte	IN	IN/OUT	UTPCI	UTPCI/UTPCO
Kernspeicher ↔ Lochstreifen	IN	IN/OUT	UTPTI	UTPTI/UTPTO
Kernspeicher ↔ Magnetkonto	IN	IN/OUT	UTMCI	UTMCI/UTMCO
Kernspeicher ↔ Magnetband-Cassette	IN	IN/OUT	UTTCI	UTTCI/UTTCO

	SYSTEMBESCHREIBUNG	
--	--------------------	--

3. Funktionseinheiten

3.1. Rechner mit der standardmäßigen Ein/Ausgabe-Einheit 154

3.1.1. Rechner

Der Rechner ist aus integrierten Schaltkreisen aufgebaut (3. Computer-Generation). Er besteht aus einem 6-Bit-Steuerwerk, einem 12-Bit-Parallelrechnerwerk und mehreren Registern. Das Steuerwerk regelt abhängig von Operationsteil eines Befehles den Ablauf innerhalb des Rechners. Das Rechnerwerk arbeitet 12-Bit-parallel und führt arithmetische und logische Operationen durch. Die Register des Rechners halten Zwischenergebnisse fest, die für das Steuerwerk und das Betriebsprogramm wichtig sind.

3.1.2. Ein/Ausgabe-Einheit des Rechners

An jedem Rechner ist standardmäßig eine Ein/Ausgabe-Einheit angeschlossen. Die Ein/Ausgabe-Einheit besteht aus Ein- und Ausgabekanälen, um die Datenein- und -ausgabe für die standardmäßigen Randeinheiten zu steuern. Zu den Standard-Randeinheiten zählen die beiden Tastaturen und der Serialdrucker. Darüber hinaus läßt sich an dieser Ein/Ausgabe-Einheit der automatische Konteneinzug anschließen.

3.2. Zentralspeicher

3.2.1. Allgemeines

NIXDORF setzt als Zentralspeicher zwei Speicherarten ein, und zwar:

- Festspeicher und
- Magnetkernspeicher.

Der Begriff "Festspeicher" weist schon mit dem Namen darauf hin, daß die Informationen fest verdrahtet (gefädelt) sind. Folglich lassen sich diese Informationen nur lesen (read-only-memory = "Nur-Lese-Speicher").

	SYSTEMBESCHREIBUNG	
--	--------------------	--

NIXDORF verwendet zwei Arten von Festspeichern:

- Stäbchenspeicher und
- Ringkernspeicher.

Die Ringkernspeicher werden überwiegend für Anwenderprogramme eingesetzt, während Stäbchenspeicher sowohl das Anwender- als auch das Betriebsprogramm aufnehmen. Es ist vorgesehen, die Produktion der Ringkernspeicher auslaufen zu lassen, sobald eine ausreichende Produktionskapazität von Stäbchenspeichern aufgebaut ist.

Im Gegensatz zu den Festspeichern läßt sich im Magnetkernspeicher der Inhalt durch Programmbefehle oder Eingaben verändern (z. B. Löschen und Schreiben).

Magnetkernspeicher nehmen auf:

- Arbeitsbereiche für
 - . Betriebsprogramm und
 - . Anwenderprogramm und
- Befehlsbereiche.

3.2.2. Festspeicher

3.2.2.1. Stäbchenspeicher

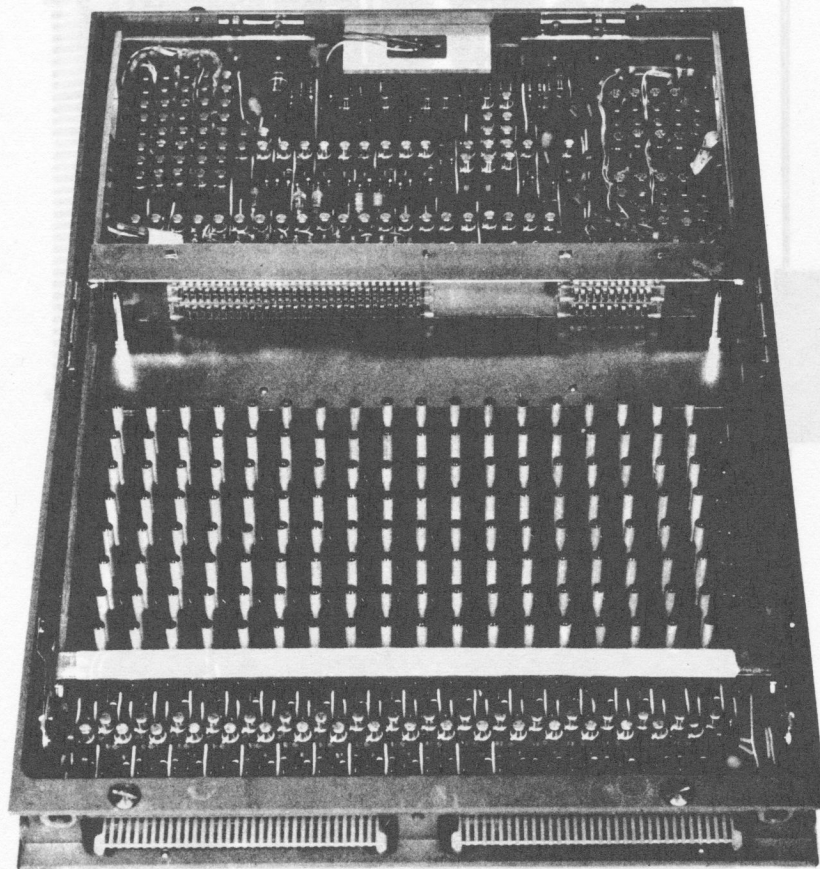
Der Stäbchenspeicher hat die Typenbezeichnung 177 und nimmt jeweils zwei Programmträger Typ 380 bzw. 381 auf.

Der 177er-Speicher enthält Ferritkerne (Stäbchen) mit Primärwicklungen und Sekundärwicklungen zu 200 Windungen. Die Stäbchen sind fest mit dem Rahmen verbunden. Sie stellen kleine Transformatoren dar, die in acht Zeilen und 18 Spalten angeordnet sind (144 Ferritkerne).

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Die folgenden zwei Abbildungen zeigen den Aufbau des Festspeichers 177 und des Programmträgers 381.

Festspeicher 177



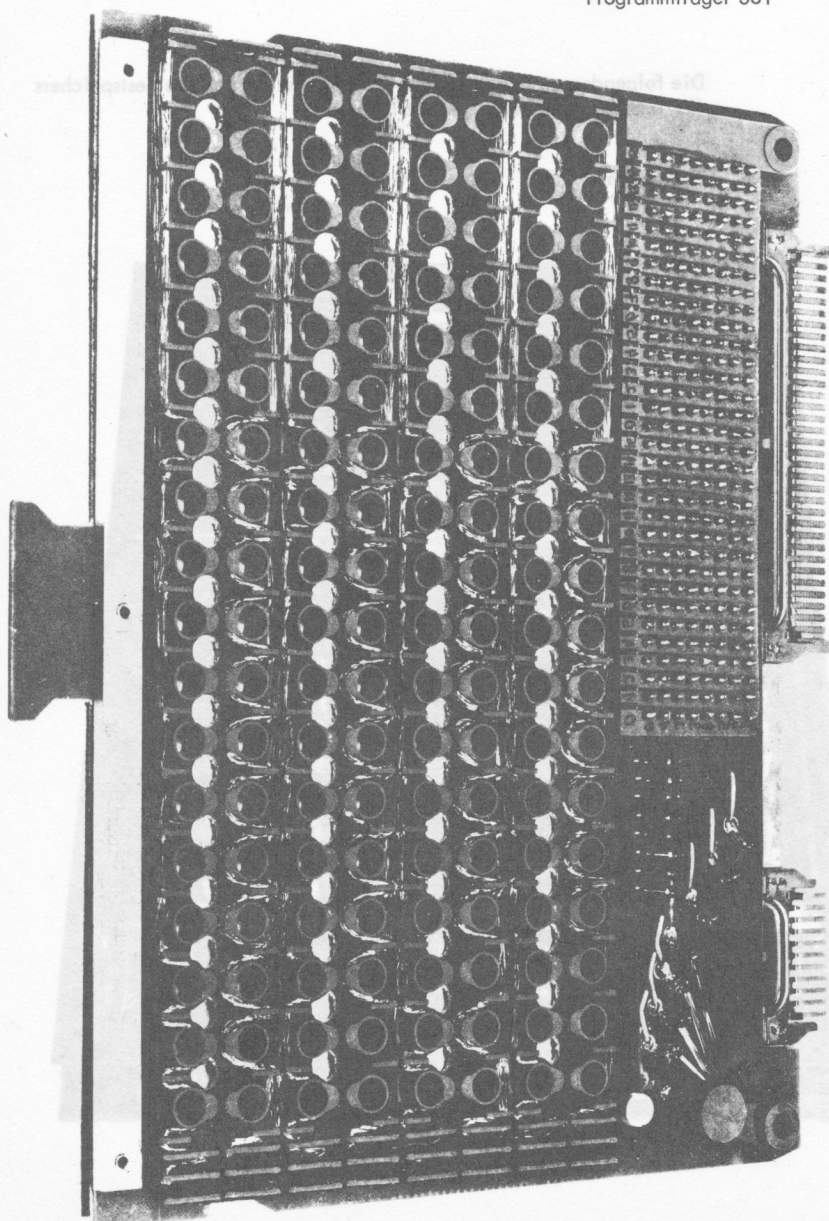
Blatt S18
1. 3. 1971

Datenverarbeitungssystem 820

NIXDORF
COMPUTER

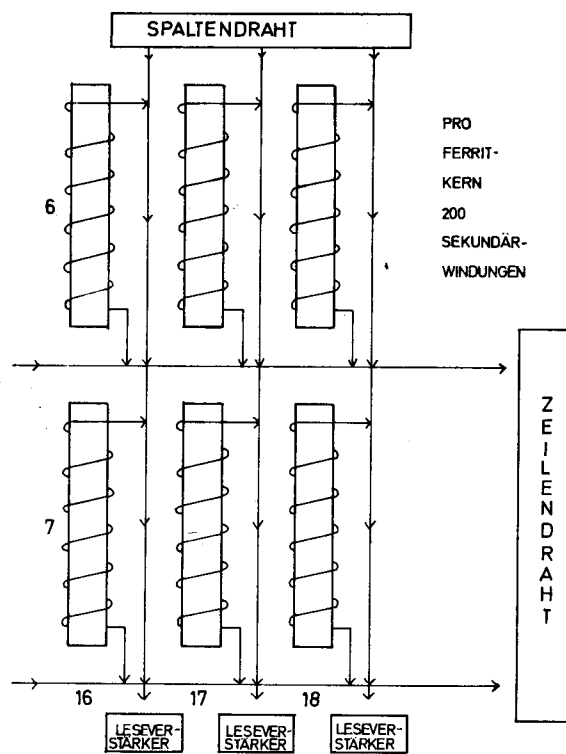
	SYSTEMBESCHREIBUNG	
--	--------------------	--

Programmträger 381



	SYSTEMBESCHREIBUNG	
--	---------------------------	--

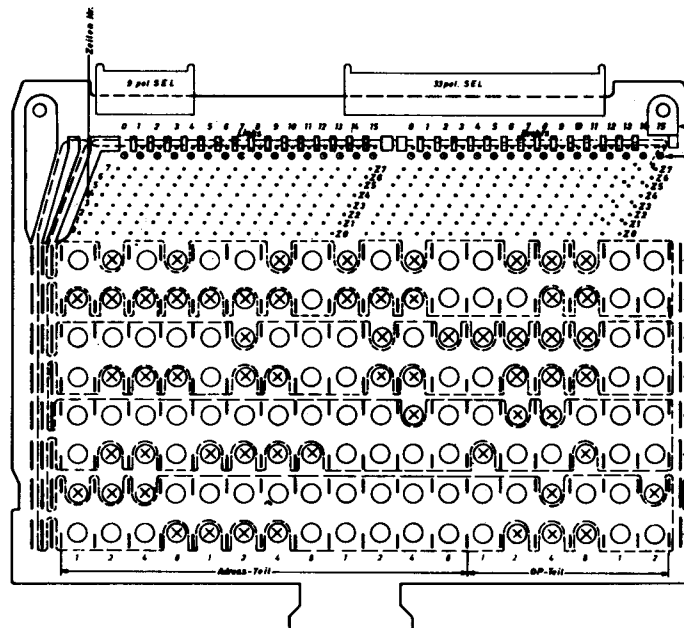
Die nachstehende Schemazeichnung zeigt sechs Ferritkerne mit den Sekundärwicklungen, die durch beide Programmträger hindurchreichen. Die Programmträger 380 enthalten die Primärwicklung und bestimmen aufgrund der jeweiligen Führung des Drahtes um den Ferritkern, wie das Magnetfeld aufgebaut wird, d.h. ob ein Bit auf Eins oder Null gesetzt wird.



pro Spalte 1 Leseverstärker = 18 Leseverstärker

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Die folgende Zeichnung zeigt schematisch, wie Befehls-
worte in Gruppen von acht Zeilen zu je 18 Bit verdrahtet (gefädelt) werden.



Die Speicherkapazität eines Programmträgers 380 beträgt 2048 Befehls-
worte, so daß sich in einem Festspeicher 177 maximal 4096 Befehls-
worte speichern lassen.

3.2.2.2. Ringkernspeicher

Die Ringkernspeicher haben die Typenbezeichnung 17/20 und bieten
eine Speicherkapazität von 2048 Befehls-
worten an.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

3.2.2.3. Speichermanagement der 177er Speicher

Die 177er Festspeicher haben außer den Ferritstäbchen folgende Einrichtungen:

- Adreßauswahl zur Auffindung des gewünschten Befehlswortes,
- Einrichtung zur Steuerung des zeitlichen Ablaufes: "Taktung".

3.2.2.4. Adressierung bei 177er Festspeicher

Die Adressen der Befehlsorte setzen sich technisch gesehen aus dem ausgewählten Programmtraht (Primärwindung) und dem ausgewählten Zeilendraht (Sekundärwindung) zusammen. Es gibt bis zu 256 Programmtrahte pro Programmträger. Bei der Auswahl (eins aus 256) eines der 256 möglichen Programmtrahte, werden gleichzeitig acht Befehlsorte angesprochen, da jeder einzelne Programmtraht durch alle acht Zeilen der Stäbchenordnung läuft. Aus diesen acht Befehlsorten wird der gewünschte Befehl durch Auswahl eines der acht Zeilendrahte angesprochen (eins aus acht). Die dadurch ausgewählten Lesesignale kommen über die 18 Spaltendrahte zu den 18 Leseverstärkern. Die verstärkten Signale werden an den Rechner weitergeleitet.

Innerhalb eines Programmblockes laufen die Adressen von null bis 2047. Deshalb hat gegebenenfalls eine Blockumschaltung auf den jeweiligen Chassisplatz bzw. Programmträger zu erfolgen, ehe der Zugriff auf das gesuchte Befehlsort stattfinden kann.

Der Speicherplatz des Programmblockes (Nummer des Speicherplatzes im Chassis bzw. Nummer des Programmträgers) entspricht der Blocknummer. Jeder Befehlsblock befindet sich auf einem bestimmten Einschub im Chassis und beim 177er Speicher auf einem der beiden Programmträger.

3.2.2.5. Einsatz der Festspeicher

Die Festspeicher nehmen sowohl das Betriebsprogramm als auch das Anwenderprogramm auf. Für das Betriebsprogramm sind nur noch 177er Speicher zugelassen. In einem Chassisplatz stehen entweder 2048 oder 4096 Befehle.

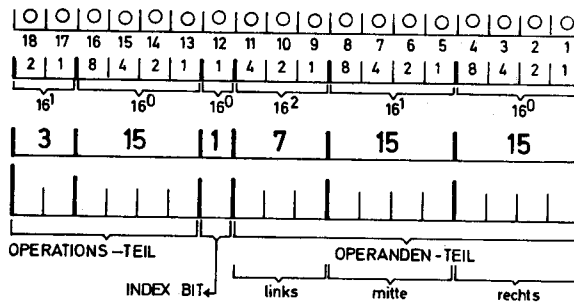


Das Anwenderprogramm läßt sich auf beiden Festspeicherarten unterbringen. Beim 177er Speicher sind max. zwei Blöcke = 4096 Befehle möglich. Der Einschub mit einem 17/20-Festspeicher enthält 2048 Befehlswoorte. In beiden Fällen, d.h. beim Anwender- und Betriebsprogramm besteht das Befehlswoort aus 18 Bit, wobei ein Befehlswoort folgende Informationen speichert:

- einen Befehl oder
- zwei numerische Konstanten oder
- drei alphanumerische Konstanten im 6-Bit-Code.

3.2.2.6. Einteilung des Befehlswoortes im Anwenderprogramm

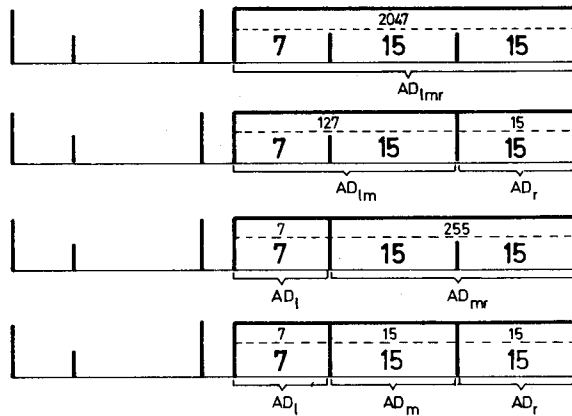
Nimmt das Befehlswoort einen Befehl des Anwenderprogrammes auf, so sind die 18 Bit folgendermaßen eingeteilt:



Die Bit-Kombination im Operationsteil gibt die Art des Befehls an und wird auch als Operations-Code bezeichnet (abgekürzt OP).

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Der Operandenteil setzt sich gegebenenfalls aus verschiedenen Abschnitten zusammen, die als links, mitte, rechts bezeichnet sind. Da im Operandenteil sehr oft Adressen stehen, wird er auch als Adreßteil bezeichnet (abgekürzt AD). Dementsprechend gibt es folgende Möglichkeiten, Adreßteile anzugeben:



	SYSTEMBESCHREIBUNG	
--	--------------------	--

Das Indexbit trennt den Operationsteil vom Adreßteil und gibt an, ob der Adreßteil zu indizieren ist oder nicht.

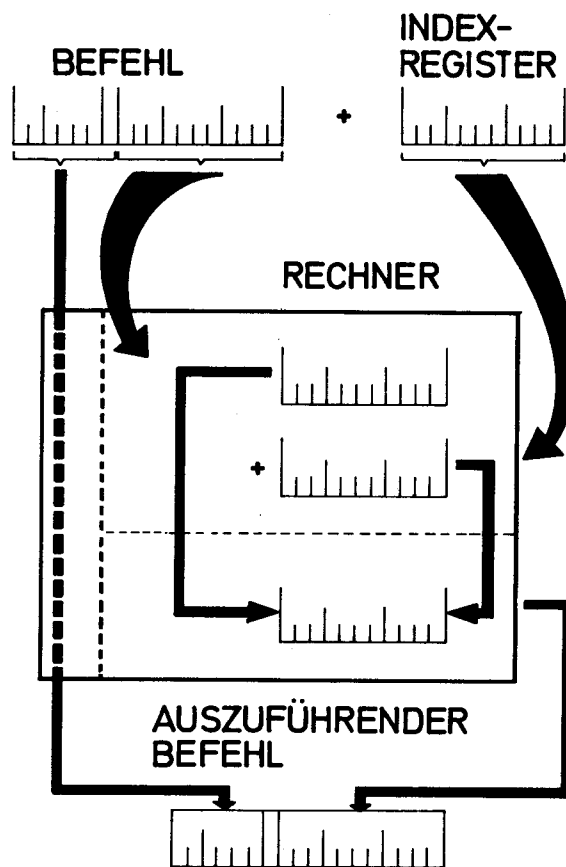
Indizieren bedeutet, daß auf den binären Inhalt des Adreßteils links, mitte, rechts der binäre Inhalt des Indexregisters aufaddiert wird. Die Summe ergibt den neuen Adreßteil, wobei ein Überlauf über Bit 11 (über 2047) hinaus für den Programmierer verlorengeht. Die Summe steht in einem Register des Rechners und wird als der auszuführende Adreßteil des Befehls angesehen.

Die Indexregister sind spezielle Register, die sich sowohl durch das Betriebsprogramm als auch durch das Anwenderprogramm ansprechen lassen.

Die Modelle der Serie 820 haben eine unterschiedliche Anzahl von Indexregistern, die in einem reservierten Bereich des Magnetkernspeichers (im Arbeitsbereich des Betriebsprogramms) stehen. Das Modell 820/15 arbeitet extern mit fünf Indexregistern, während die Modelle 820/25 und 820/35 jeweils acht Indexregister ansprechen.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Folgende Abbildung zeigt die schematische Darstellung für den Vorgang des Indizierens.



	SYSTEMBESCHREIBUNG	
--	--------------------	--

3.2.3. Magnetkernspeicher

3.2.3.1. Allgemeines

Die Magnetkernspeicher haben sechs Typenbezeichnungen, die sich auf den Aufbau und auf die Kapazität des Speichers beziehen.

Im Gegensatz zum Festspeicher läßt sich der Inhalt des Magnetkernspeichers durch Programmbefehle verändern, da die Magnetkerne aus Ferrit (Eisenoxyd) bestehen, deren Magnetfeld auch beim Ausschalten des Stromes erhalten bleibt (Permanent-Magnet). Die Magnetringe sind matrixförmig angeordnet. Durch die Magnetringe führen mehrere Drähte. Zwei davon sind so angeordnet, daß sich ihre Spannungen je nach Stromrichtung ergänzen oder aufheben. Nur wenn sich die Spannungen aus beiden Richtungen addieren, ist der Strom in der Lage, die Magnetfeldlinien umzukehren (umzupolen).

Ein Lesedraht stellt den Zustand des Magnetkernes fest und gibt den Inhalt an den Rechner weiter (Lesen). Durch das Lesen wird das Magnetfeld des Magnetkernes umgepolt, so daß es notwendig ist, den ursprünglichen Zustand wieder herzustellen. Infolgedessen sind beim Lesen folgende Zeiten zu unterscheiden:

- Zugriffszeit = Zeit für den Datenübertrag vom Magnetkernspeicher zum Rechner (Lesen)
- Zykluszeit = Zeit für den Datenübertrag vom Magnetkernspeicher + Zeit für Regenerieren des alten Zustandes. Dementsprechend besteht die Zykluszeit aus der Lesephase und der Regenerationsphase.

3.2.3.2. Einteilung des Magnetkernspeichers

Magnetkernspeicher sind in Speicherworte unterteilt, die aus 16 Stellen bestehen. Je nach Speichertyp sind diese Stellen entweder 12 Bit, 6 Bit oder 4 Bit tief. Je nach Belegung der Speicherworte werden die Stellenkapazitäten verschieden ausgenutzt.

- 12 Bit für den Arbeitsbereich des Betriebsprogramms.
 - . Die Einteilung des Arbeitsbereiches ist abhängig vom jeweiligen Betriebsprogramm.
- 4 Bit für den Arbeitsbereich des Anwenderprogramms.
 - . Belegung mit numerischen Zeichen,
 - . Belegung mit alphanumerischen Zeichen im 8-Bit-Code,
 - . Belegung mit alphanumerischen Zeichen im 6-Bit-Code.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

- 6 Bit und 12 Bit beim Befehlsbereich für das Anwenderprogramm.
- . Befehle im Magnetkernspeicher sind je nach Speichertyp bzw. Betriebsprogramm unterschiedlich gespeichert.

Einteilung der Kernspeicher

Bei 820/15 (Typ 160 bis 163 und 165)

Arbeitsbereich	Arbeitsbereich	Befehlsbereich
BP	AP	AP

BP = Betriebs-
Programm
AP = Anwender-
Programm

Bei 820/15 (Typ 166)

Arbeitsbereich	Arbeitsbereich	Befehlsbereich	Arbeitsbereich
BP	AP	AP	AP

BP = Betriebs-
Programm
AP = Anwender-
Programm

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Bei 820/25 und 820/35 (Typ 160 bis 163, 165,166)
Einsatz als Datenspeicher (Speicherworte)

Arbeitsbereich	Arbeitsbereich
BP	AP

BP = Betriebs-
Programm
AP = Anwender-
Programm

Bei 820/25 und 820/35 (Typ 166)
Einsatz als Programmspeicher (Befehlswords)

Befehlsbereich
AP

BP = Betriebs-
Programm
AP = Anwender-
Programm

	SYSTEMBESCHREIBUNG	
--	--------------------	--

3.2.3.3. Speicherworte mit numerischem Inhalt

Beim numerischen Inhalt nehmen die Stellen 1 bis 15 Dezimalziffern von 0 bis 9 in binärer Darstellung auf, während Speicherstelle 0 das Vorzeichen enthält.

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

Das Vorzeichen richtet sich nach der jeweiligen Bitkombination. Eine ungerade Wertigkeit ergibt ein negatives und eine gerade Wertigkeit ein positives Vorzeichen.

3.2.3.4. Speicherworte mit alphanumerischem Inhalt

ALC-Tabelle

Alle druckbaren Zeichen werden bei NIXDORF in sedezipalmer Verschlüsselung in einer sogenannten ALC-Tabelle zusammengefaßt (ALC = alphanumeric character).

Für folgende Kugelkopftypen gibt es spezielle ALC-Tabellen:

- Typ 1000 (Deutschland, Österreich)
- Typ 884 (Frankreich, Belgien)
- Typ 882 (Schweiz)
- Typ 863 (Großbritannien)
- Typ 880 (Dänemark, Norwegen)
- Typ 881 (Schweden, Finnland)
- Typ 917 (Puerto Rico, Spanien)
- Typ 912 (Italien, USA)
- Typ 956 (Griechenland)
- Typ 982 (Korrespondenzdeutsch, PICA 1)

Zu jedem Zeichen gehören der ALC-Code, die symbolische Bezeichnung und die Benennung des Zeichens. Der ALC-Code reicht von 0.0 bis 5.15.

	SYSTEMBESCHREIBUNG
--	--------------------

ALC-Tabelle

ALC-Code	Symbol	ALC-Code	Symbol
00	0	30	YCOL
01	1	31	YEDM
02	2	32	YEDM
03	3	33	YEDM
04	4	34	YEDM
05	5	35	YEDM
06	6	36	YEDM
07	7	37	YEDM
08	8	38	YEDM
09	9	39	YEDM
010	YALU	310	YEDM
011	YALS	311	YEDM
012	YALN	312	YEDM
013	YALM	313	YEDM
014	YALH	314	YEDM
015	YALB	315	YEDM
10	YALR	40	YEDM
11	YALG	41	YEDM
12	YALF	42	YEDM
13	YALE	43	YEDM
14	YALD	44	YEDM
15	YALC	45	YEDM
16	YALB	46	YEDM
17	YALA	47	YEDM
18	YALZ	48	YEDM
19	YALY	49	YEDM
20	YALX	50	YEDM
21	YALW	51	YEDM
22	YALV	52	YEDM
23	YALU	53	YEDM
24	YALT	54	YEDM
25	YALS	55	YEDM
26	YALR	56	YEDM
27	YALQ	57	YEDM
28	YALP	58	YEDM
29	YALO	59	YEDM
30	YALN	60	YEDM
31	YALM	61	YEDM
32	YALL	62	YEDM
33	YALK	63	YEDM
34	YALJ	64	YEDM
35	YALI	65	YEDM
36	YALH	66	YEDM
37	YALG	67	YEDM
38	YALF	68	YEDM
39	YALE	69	YEDM
40	YALD	70	YEDM
41	YALC	71	YEDM
42	YALB	72	YEDM
43	YALA	73	YEDM
44	YALZ	74	YEDM
45	YALY	75	YEDM
46	YALX	76	YEDM
47	YALW	77	YEDM
48	YALV	78	YEDM
49	YALU	79	YEDM
50	YALT	80	YEDM
51	YALS	81	YEDM
52	YALR	82	YEDM
53	YALQ	83	YEDM
54	YALP	84	YEDM
55	YALO	85	YEDM
56	YALN	86	YEDM
57	YALM	87	YEDM
58	YALL	88	YEDM
59	YALK	89	YEDM
60	YALJ	90	YEDM
61	YALI	91	YEDM
62	YALH	92	YEDM
63	YALG	93	YEDM
64	YALF	94	YEDM
65	YALE	95	YEDM
66	YALD	96	YEDM
67	YALC	97	YEDM
68	YALB	98	YEDM
69	YALA	99	YEDM
70	YALZ	100	YEDM

SYSTEMBESCHREIBUNG

Für das Modell 820/15 stehen bis jetzt nur die Tabellen für die Kugelköpfe 1000, 982 und 912 zur Verfügung (unter Verwendung der entsprechenden Kugelköpfe und der Moduln TW 1000 bzw. TW 982 bzw. 912).

ALC-Code	Symbol	Deutschland Österreich		Italien USA	Korrespondenzbereich
		1000	912		
0.0	0	0	0	0	0
0.1	1	1	1	1	1
0.2	2	2	2	2	2
0.3	3	3	3	3	3
0.4	4	4	4	4	4
0.5	5	5	5	5	5
0.6	6	6	6	6	6
0.7	7	7	7	7	7
0.8	8	8	8	8	8
0.9	9	9	9	9	9
0.10	YBLK	-	-	-	a
0.11		-	-	-	b
0.12	YPLS	+	+	+	c
0.13	YMIN	-	-	-	d
0.14	YITS	o	o	o	e
0.15	YITSM	o	o	o	f
1.0	YSTR	+	+	+	g
1.1	YSTRM	+	+	+	h
1.2	A	A	A	A	A
1.3	B	B	B	B	B
1.4	C	C	C	C	C
1.5	D	D	D	D	D
1.6	E	E	E	E	E
1.7	F	F	F	F	F
1.8	G	G	G	G	G
1.9	H	H	H	H	H
1.10	I	I	I	I	I
1.11	J	J	J	J	J
1.12	K	K	K	K	K
1.13	L	L	L	L	L
1.14	M	M	M	M	M
1.15	N	N	N	N	N
2.0	O	O	O	O	O
2.1	P	P	P	P	P
2.2	Q	Q	Q	Q	Q
2.3	R	R	R	R	R
2.4	S	S	S	S	S
2.5	T	T	T	T	T
2.6	U	U	U	U	U
2.7	V	V	V	V	V
2.8	W	W	W	W	W
2.9	X	X	X	X	X
2.10	Y	Y	Y	Y	Y
2.11	Z	Z	Z	Z	Z
2.12	YNMB	.	.	.	h
2.13	YPNT	.	.	.	i
2.14	YCM	.	.	.	j
2.15	YSEM	.	.	.	k

ALC-Code	Symbol	Deutschland Österreich		Italien USA	Korrespondenzbereich
		1000	912		
3.0	YCOL	:	:	:	?
3.1	YEXM	!	!	!	k
3.2	YQEM	?	?	?	l
3.3	YQOM	"	"	"	m
3.4	YEQL	<	<	<	n
3.5	YULN	>	>	>	o
3.6	YPOP	(((p
3.7	YPCL)))	q
3.8	YDLR	\$	\$	\$	r
3.9	YCAN	&	&	&	s
3.10	YPD	%	%	%	t
3.11	YPCT	%	%	%	u
3.12	YPMI	%	%	%	v
3.13	YAPH	%	%	%	w
3.14	YBAR	/	/	/	x
3.15	YECC	/	/	/	y
4.0	YCMPT	,	,	,	z
4.1	YPRG	(((+
4.2	YCA	@	@	@	-
4.3	YHS	#	#	#	.
4.4	YM	m	m	m	?
4.5	YL	l	l	l	i
4.6	YT	t	t	t	o
4.7					b
4.8					u
4.9					b
4.10					B
4.11	YDIA				/
4.12					U
4.13					Ä
4.14					3
4.15	YMCB)
5.0	YST	ST	ST	ST	:
5.1					;
5.2					-
5.3					2
5.4	YMSQ	m ²	m ²	m ²	%
5.5	YKG	kg	kg	kg	&
5.6					"
5.7					X
5.8					O
5.9	YG	g	g	g	Ü
5.10	YINV	v	v	v	/
5.11					
5.12	YBS	.	.	.	
5.13					
5.14	YDCM	.	.	.	
5.15	YDPNT	.	.	.	

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Zeichen im 8-Bit-Code

Im 8-Bit-Code belegen alphanumerische Zeichen jeweils zwei Speicherstellen á 4 Bit, so daß ein Speicherwort bis zu acht alphanumerische Zeichen aufnimmt. Ein Endezeichen schließt variabel lange Informationen ab. Die Endezeichen haben unterschiedliche Codierungen und lassen sich durch verschiedene Tasten erzeugen.

ENDEZEICHEN-CODES	
TASTE	Codierung
TAB-Taste (alphanumerische Eingabetastatur)	6.8
AUSLÖSETASTE (numerische Eingabetastatur)	6.11 = 6 ⁻
WZ-Taste (alphanumerische Eingabetastatur)	6.12 = 6 ⁺

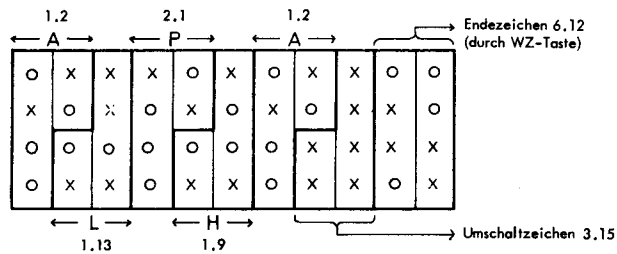
Mit speziellen Befehlen lassen sich alphanumerische Zeichen im 8-Bit-Code ab jeder beliebigen Stelle eines Speicherwortes unterbringen. Die Verschlüsselung der einzugebenden Zeichen erfolgt mit Hilfe des ALC-Codes.

0 0	0 x	0 0	0 x	0 0	0 x	8 4 2 1 Wertigkeit
0 0	0 x	0 0	0 0	0 0	x 0	
0 x	0 0	x 0	0 0	0 x	x 0	
x 0	x x	0 x	x x	x 0	0 0	
A	L	P	H	A	↑	Endezeichen (durch TAB-Taste)
1.2	1.13	2.1	1.9	1.2	6.8	← ALC-Code

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Zeichen im 6-Bit-Code

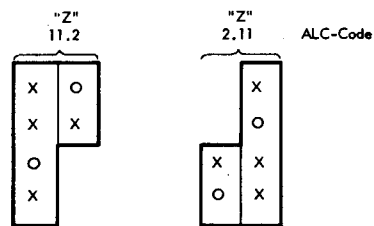
Im 6-Bit-Code belegt ein Zeichen in verzahnter Form 1 1/2 Speicherstellen. Das Betriebsprogramm setzt vor ein Endezeichen (vgl. vorstehende Tabelle) automatisch das Umschaltzeichen mit dem sechszimalen Code 3.15.



1 2 | 5 | 13 | 1 | 6 | 9 | 2 | 13 | 15 | 6 | 12 |

Die Speicherung der 6-Bit-Zeichen erfolgt entsprechend dem Code in Gruppen von zwei und vier Bits, wobei abwechselnd die 2-Bit-Gruppe (Wertigkeit 16⁴) oder die 4-Bit-Gruppe (Wertigkeit 16⁶) vorn steht.

Das Zeichen "Z" (2.11) lässt sich auf folgende zwei Arten darstellen:



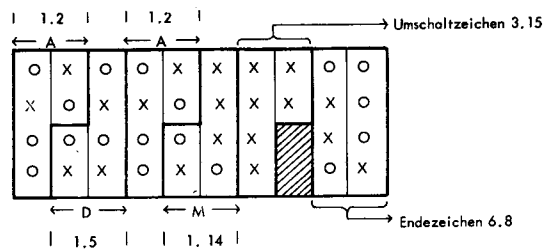
11.2 2.11

11 11

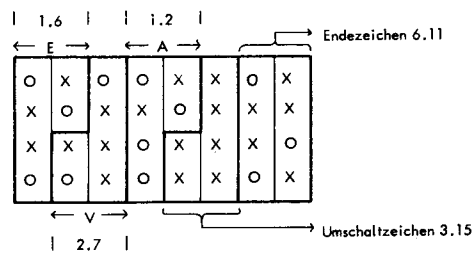
	SYSTEMBESCHREIBUNG	
--	--------------------	--

Immer dann, wenn eine Zeichenkette mit einer vollständig belegten Speicherstelle abgeschlossen ist, sind zwei Bits zwischen dem Endezeichen und dem Umschaltzeichen bedeutungslos, da das Umschaltzeichen nur sechs der vorhandenen 8 Bit belegt.

Beispiel für gerade Zeichenzahl der Informationskette "ADAM"



Beispiel für ungerade Zeichenzahl der Informationskette "EVA"



Beim Modell 820/15 ist eine 6-Bit-Code Darstellung nicht möglich.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

3.2.3.5. Speichertypen

Es gibt folgende Magnetkernspeicher:

Typ	160	161	162	163	165	166
Anzahl Speicherworte	16	128	64	32	256	512

Die Adressierung der Speicherworte unterscheidet sich nach der internen und der externen Adressierung. Das Betriebsprogramm benutzt die interne Adressierung.

In diesem Falle entspricht der Anzahl der Adressen auch die Anzahl der Speicherworte. Je nach Art des Betriebsprogrammes belegt der Arbeitsbereich des Betriebsprogrammes entweder die ersten vier oder fünf internen Adressen des Magnetkernspeichers.

Infolgedessen beginnt die externe Adressierung (Speicherwort 0) nicht am Anfang des Magnetkernspeichers, sondern auf der internen Adresse vier oder fünf.

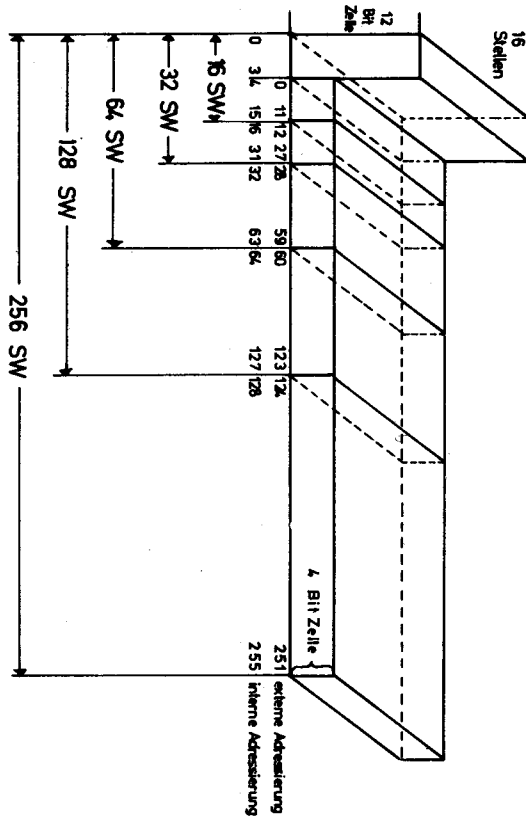
3.2.3.6. Aufbau, interne und externe Adressen beim Modell 820/15

Das Modell 820/15 arbeitet in Verbindung mit den Betriebsprogrammen MFA1 oder MFAGS1. Diese beiden Betriebsprogramme unterscheiden sich von allen übrigen Betriebsprogrammen darin, wie sie Befehle im Magnetkernspeicher ansprechen. Befehle lassen sich beim Modell 820/15 nur dann im Magnetkernspeicher unterbringen, wenn der Modul GP (general part) zur Verfügung steht.

SYSTEMBESCHREIBUNG

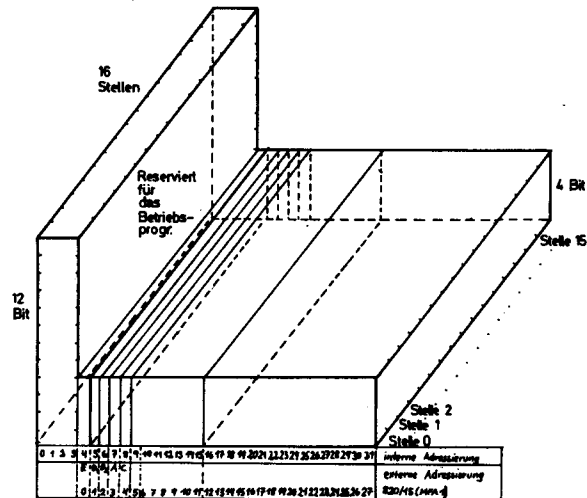
Befehlsspeicherung mit dem MFA1

Das MFA1 nutzt die Magnetkernspeicher 160, 163, 162, 161 und 165 folgendermaßen aus:



	SYSTEMBESCHREIBUNG	
--	---------------------------	--

820/15
Schema der Adressierung für Speicher 160 und 163



Die interne Adressierung reserviert die ersten vier Speicherworte, die aus 12-Bit-Zellen bestehen, als Arbeitsbereich für das Betriebsprogramm. Dadurch beginnt die externe Adressierung hardwaremäßig auf dem fünften Speicherwort, d.h. die interne Adresse vier bezeichnet dasselbe Speicherwort wie die externe Adresse Null. Befehle lassen sich erst ab der externen Adresse fünf speichern, wobei ein Speicherwort 3 1/5 Befehle enthält.

Damit belegen 16 Befehle fünf Speicherworte.

Das MFA1 erlaubt es, die Magnetkernspeicher 160 bis 165 jeweils in einen Arbeits- und einen Befehlsbereich einzuteilen. Es sind zwei Befehle im Anwenderprogramm notwendig, um die Grenze zwischen dem Arbeits- und Befehlsbereich festzulegen. Der Arbeitsbereich belegt immer die Speicherworte mit den unteren Adressen. Daran schließt sich der Befehlsbereich bis zum Ende des Magnetkernspeichers an.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Die Trennung zwischen beiden Bereichen heißt der Anfangspunkt (entry-point). Der Anfangspunkt bezieht sich auf die höchste Adresse des Arbeitsbereiches. Der Programmbereich beginnt immer mit der Speicherstelle Null eines Speicherwortes. Die 64 Bit des Speicherwortes nehmen 3 1/5 Befehle auf, d.h. die 12 Bit des Operandenteils belegen drei Speicherstellen. Der Operationsteil belegt zwei Speicherstellen, wobei von den 8 Bit nur 6 ausgenutzt werden.

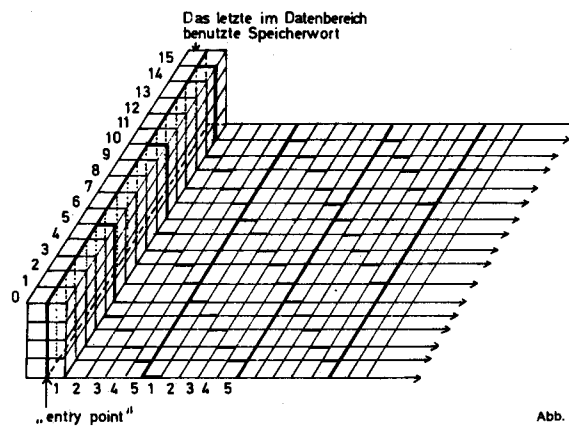


Abb. 18

Die nachstehende Tabelle führt für jeden Speichertyp die Anzahl der Speicherworte und den externen Adreßbereich auf.

Aufbau der Magnetkernspeicher Typ 160 bis 165

Typ	Anzahl SW	externe Adressierung	Maximaler Befehlsbereich	
			Anzahl BW	durch BW belegte SW
160	16	0 bis 11	22	7
163	32	0 bis 27	73	23
162	64	0 bis 59	176	55
161	128	0 bis 123	380	119
165	256	0 bis 251	790	247

SW = Speicherworte
BW = Befehlsworte

	SYSTEMBESCHREIBUNG	
--	---------------------------	--

Möglichkeiten der Aufteilung des Magnetkernspeichers in Speicherworte und Befehlswords
(Magnetkernspeicher Nr. 160 bis 163, 165)

Von der Gesamtzahl der Speicherworte eines Speichers gehen vier Speicherworte (interne Adressen-Nr. 0 bis 3) für den Arbeitsbereich des Betriebsprogrammes ab.

	Externe Adressen der zur Verfügung stehenden Speicherworte	Anzahl der Speicherworte		Anzahl der speicherbaren Befehle	Höchste Adresse der Befehle
		für Daten	für Befehle		
Magnetkernspeicher Nr. 160 (16 SW)	0-11	12	0	0	-
	0-10	11	1	3	0.0. 2
	0-9	10	2	6	0.0. 5
	0-8	9	3	9	0.0. 8
	0-7	8	4	12	0.0.11
	0-6	7	5	16	0.0.15
	0-5	6	6	19	0.1. 2
	0-4	5	7	22	0.1. 5
	0-3	4	8	25	0.1. 8
	0-2	3	9	28	0.1.11
0-1	2	10	32	0.1.15	
Magnetkernspeicher Nr. 163 (32 SW)	0-27	28	0	0	-
	0-26	27	1	3	0.0. 2
	0-25	26	2	6	0.0. 5
	0-24	25	3	9	0.0. 8
	0-23	24	4	12	0.0.11
	0-22	23	5	16	0.0.15

	0-6	7	21	67	0.4. 2
	0-5	6	22	70	0.4. 5
	0-4	5	23	73	0.4. 8
0-3	4	24	76	0.4.11	
0-2	3	25	80	0.4.15	
0-1	2	28	83	0.5. 2	
Magnetkernspeicher Nr. 162 (64 SW)	0-59	60	0	0	-
	0-58	59	1	3	0.0. 2
	0-57	58	2	6	0.0. 5
	0-56	57	3	9	0.0. 8
	0-55	56	4	12	0.0.11

	0-5	6	54	172	0.10.11
	0-4	5	55	176	0.10.15
	0-3	4	56	179	0.11. 2
	0-2	3	57	182	0.11. 5
0-1	2	58	185	0.11. 8	

	SYSTEMBESCHREIBUNG	
--	---------------------------	--

	Externe Adressen der zur Verfügung stehenden Speicherworte	Anzahl der Speicherworte		Anzahl der speicherbaren Befehle	Höchste Adresse der Befehle
		für Daten	für Befehle		
Magnetkernspeicher Nr. 161 (128 SW)	0 - 123	124	0	0	—
	0 - 122	123	1	3	0.0. 2
	0 - 121	122	2	6	0.0. 5
	0 - 120	121	3	9	0.0. 8
	0 - 119	120	4	12	0.0.11

	0 - 5	6	118	377	1.7. 8
	0 - 4	5	119	380	1.7.11
	0 - 3	4	120	384	1.7.15
	0 - 2	3	121	387	1.8. 1
	0 - 1	2	122	390	1.8. 5
Magnetkernspeicher Nr. 165 (256 SW)	0 - 251	252	0	0	—
	0 - 250	251	1	3	0.0. 2
	0 - 249	250	2	6	0.0. 5
	0 - 248	249	3	9	0.0. 8
	0 - 247	248	4	12	0.0.11
	0 - 246	247	5	16	0.0.15

	0 - 6	7	245	784	3.0.15
	0 - 5	6	246	787	3.1. 2
	0 - 4	5	247	790	3.1. 5
	0 - 3	4	248	793	3.1. 8
0 - 2	3	249	796	3.1.11	
0 - 1	2	250	800	3.1.15	

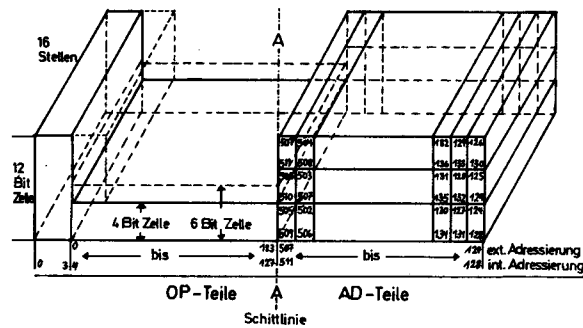
	SYSTEMBESCHREIBUNG	
--	--------------------	--

Befehlspeicherung mit dem MFAGS1

Das MFAGS1 nutzt die 512 Speicherworte des Magnetkernspeichers 166 folgendermaßen aus:

820/15 (MFAGS1)

Schema der Adressierung für Speicher 166



Die ersten vier Speicherworte sind für das Betriebsprogramm reserviert. Mit dem fünften hardwaremäßigen Speicherwort (interne Adresse 4) beginnt die externe Adressierung (Adresse Null). Der Arbeitsbereich des Betriebsprogramms besteht aus 12-Bit-Zellen. Daran schließen sich die Speicherworte mit den externen Adressen von Null bis 511 an. Die unteren externen Adressen von Null bis 123 stellen zwar 6-Bit-Zellen zur Verfügung, aber das MFAGS1 nutzt diese 6 Bits nur dann voll aus, wenn Befehle gespeichert sind. Alphanumerische Zeichen belegen bei den unteren Adressen nur 4 Bit der 6-Bit-Zelle.

	SYSTEMBESCHREIBUNG	
--	---------------------------	--

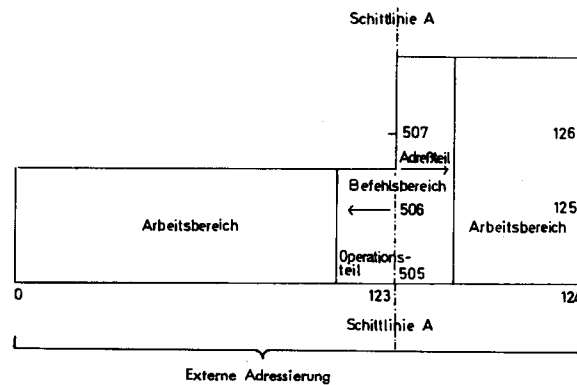
An den Block mit den 6-Bit-Zellen schließen sich 128 Zellen an, die wie der Arbeitsbereich des Betriebsprogramms aus 12-Bit-Zellen bestehen. Das MFAGS1 interpretiert diesen Block mit 384 Speicherworten (Adressen 124 bis 507) folgendermaßen:

- Jede 12-Bit-Zelle ist in 3 Gruppen zu je 4 Bit eingeteilt.
- Die Adressierung der Speicherworte 124 bis 511 geht räumlich (hardwaremäßig) gesehen folgendermaßen vor sich:
 - Innerhalb einer 12-Bit-Zelle von unten nach oben.
 - Innerhalb des Blocks der 12-Bit-Zellen von (rechts) außen zur Mitte hin.
 (vgl. obige Zeichnung Schnittlinie A)

Der Befehlsbereich beginnt an der Schnittlinie A und dehnt sich gleichmäßig um jeweils eine Zelle nach rechts und nach links aus. Die 6-Bit-Zellen der unteren Speicherworte von 123 an abwärts nehmen jeweils den Operationscode für 16 Befehle auf. Je drei Speicherworte von 511 an abwärts nehmen jeweils die zugehörigen 16 Operandenteile der Befehle auf.

Auf diese Weise entstehen:

- ein zusammenhängender Befehlsbereich,
- zwei getrennte Arbeitsbereiche, die den Befehlsbereich einschließen.



	SYSTEMBESCHREIBUNG	
--	--------------------	--

In einen 166er Magnetkernspeicher lassen sich maximal 1904 Befehle speichern, wobei die Speicherworte mit den Adressen von 123 bis 5 und 507 bis 151 den Befehlsbereich darstellen. Damit bleiben die Speicherworte 0 bis 4 und 124 bis 150 als Arbeitsbereich übrig.

Möglichkeiten der Aufteilung des Magnetkernspeichers 166 in Speicherworte und Befehlsworte:

Von der Gesamtanzahl der Speicherworte (512) gehen vier Speicherworte (interne Adressen 0 bis 3) für den Arbeitsbereich des Betriebsprogramms ab. Die verbleibenden 508 Speicherworte (externe Adressen 0 bis 507) gliedern sich in:

124 Speicherworte unten (interne Adressen 0 bis 123) mit Stellen zu je 6 Bit und

384 Speicherworte oben (interne Adressen 124 bis 507) mit Stellen zu je 4 Bit.

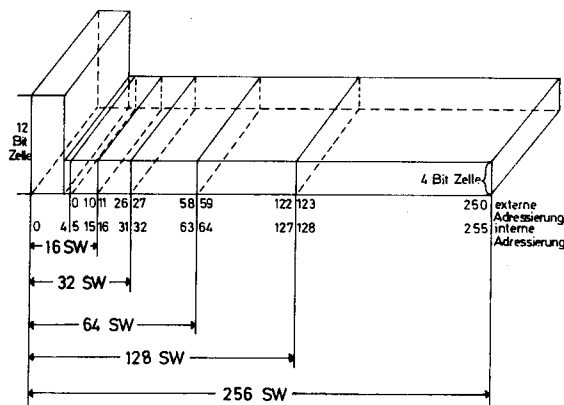
	Für Daten verwendete Speicherworte				Gesamt-Anzahl	Für Bef. Gesamt-Anzahl d. Reg.	Anzahl der Befehle	Höchste Adresse der Befehle
	unten (externe) Adresse	Anzahl	oben (externe) Adresse	Anzahl				
0 - 123	124		124 - 507	384	508	0	0	
0 - 122	123		124 - 504	381	504	4	16	0.0.15
0 - 121	122		124 - 501	378	500	8	32	0.1.15
0 - 120	121		124 - 498	375	496	12	48	0.2.15
0 - 119	120		124 - 495	372	492	16	64	0.3.15
0 - 118	119		124 - 492	369	488	20	80	0.4.15
0 - 117	118		124 - 489	366	484	24	96	0.5.15
0 - 116	117		124 - 486	363	480	28	112	0.6.15
:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:
0 - 8	9		124 - 162	39	48	480	1840	7.2.15
0 - 7	8		124 - 159	36	44	464	1856	7.3.15
0 - 6	7		124 - 156	33	40	468	1872	7.4.15
0 - 5	6		124 - 153	30	36	472	1888	7.5.15
0 - 4	5		124 - 150	27	32	476	1904	7.6.15
0 - 3	4		124 - 147	24	28	480	1920	7.7.15
0 - 2	3		124 - 144	21	24	484	1936	7.8.15
0 - 1	2		124 - 141	18	20	488	1952	7.9.15

	SYSTEMBESCHREIBUNG	
--	--------------------	--

3.2.3.7. Aufbau, interne und externe Adressierung bei den Modellen 820/25 und 820/35

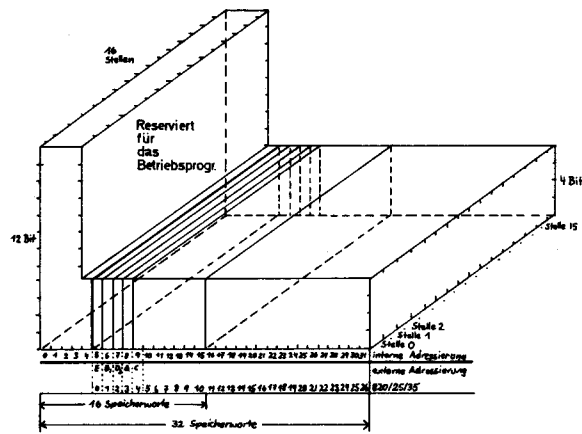
Bei den Modellen 820/25 und 820/35 lassen sich Befehle im Magnetkernspeicher nur im Typ 166 unterbringen. Bei allen Typen (160 bis 166) belegt das Betriebsprogramm die internen Adressen 0 bis 4, so daß die externe Adresse 0 mit dem internen Speicherwort 5 beginnt. Die nachstehenden Zeichnungen zeigen die interne und externe Adressierung.

820/25 und 820/35
Schema der Adressierung für Speicher 160 bis 163 und 165

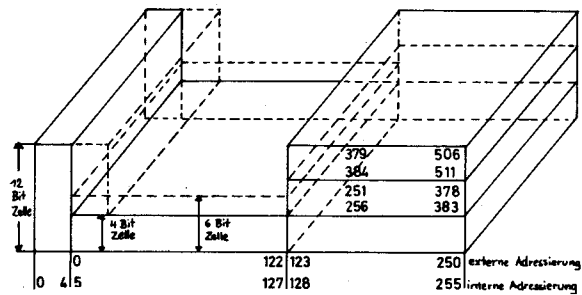


SYSTEMBESCHREIBUNG

820/25 und 820/35
Schema der Adressierung für Speicher 160 und 163



820/25 und 820/35
Schema der Adressierung für Speicher 166



	SYSTEMBESCHREIBUNG	
--	--------------------	--

3.3. Ein/Ausgabe-Einheit

3.3.1. Allgemeines

Die Ein/Ausgabe-Einheiten stellen die hardwaremäßigen Verbindungen zwischen der Zentraleinheit und den Randeinheiten her.

3.3.2. Arten der Ein/Ausgabe-Einheiten

Folgende Ein/Ausgabe-Einheiten sind für die unterschiedlichen Randeinheiten zu verwenden:

Ein/Ausgabe Typ	Randeinheiten		
Rechner Ein/Ausgabe (standardmäßig)	numerische und alphanumerische Tastatur, Serialdrucker, einfache und doppelte Vorschubeinrichtung für Endlos-Papier, automatischer Konteneinzug,		
184	<table style="border: none;"> <tr> <td style="border: none;"> <ul style="list-style-type: none"> - 1 Lochkartenstanzer - 1 Lochkartenleser - 1 Lochstreifenstanzer oder - 1 Lochstreifenkartenstanzer - 1 Lochstreifenleser oder - 1 Lochstreifenkartenleser </td> <td style="border: none; vertical-align: middle; padding-left: 10px;">} oder 2 Drucker</td> </tr> </table>	<ul style="list-style-type: none"> - 1 Lochkartenstanzer - 1 Lochkartenleser - 1 Lochstreifenstanzer oder - 1 Lochstreifenkartenstanzer - 1 Lochstreifenleser oder - 1 Lochstreifenkartenleser 	} oder 2 Drucker
<ul style="list-style-type: none"> - 1 Lochkartenstanzer - 1 Lochkartenleser - 1 Lochstreifenstanzer oder - 1 Lochstreifenkartenstanzer - 1 Lochstreifenleser oder - 1 Lochstreifenkartenleser 	} oder 2 Drucker		
186	Magnetkonten-Einheit		
310	2 Magnetbandcassetten-Einheiten		

	SYSTEMBESCHREIBUNG	
--	--------------------	--

3.4. Chassis

3.4.1. Allgemeines

Das Chassis nimmt die einzelnen Bauteile auf und verbindet sie miteinander. In den Verdrahtungsplänen für die Chassisrückwände werden die Plätze von links nach rechts durchnummeriert. Verwendet man diese Numerierung grundsätzlich, so erkennt man, wenn das Chassis von der Vorderseite betrachtet wird, von der es ja bestückt wird, daß sich die Numerierungsrichtung geändert hat, und zwar von "links nach rechts" in von "rechts nach links".

Der Rechneinschub beansprucht drei Chassisplätze, alle anderen Einschübe entsprechen einem Platz innerhalb der Numerierung.

3.4.2. Chassis-Arten

Für die Modelle der Serie 820 gibt es vier verschiedene Chassisgrößen, die folgendermaßen bezeichnet werden:

Modell	Chassis	Größe
820/15	205	einstöckig
820/25	207 oder 208	einstöckig
820/35	540	doppelstöckig

3.4.3. Chassis für Modell 820/15

Chassis 205

AP	AP/ BP	KSP	Rechner mit Standard Ein/Ausgabe- Einheit			E/A	E/A	frei	frei
10	9	8	7	6	5	4	3	2	1

AP = Anwenderprogramm, BP = Betriebsprogramm,
KSP = Kernspeicher, E/A = Ein/Ausgabe-Einheit.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Das Chassis läßt sich bestücken mit:

- Platz 10: Festspeicher 177
für das Anwenderprogramm (maximal 4096 Befehle, wahlfrei)
- Platz 9: Festspeicher 177
für gemischte Anwender- und Betriebsprogramme. Der erste Programmträger 380 nimmt 2048 Befehle des Betriebsprogramms (MFA1 oder MFAGS1) auf.
Auf dem zweiten Programmträger können maximal 1952 Befehle des Anwenderprogramms untergebracht werden, da das Betriebsprogramm mit dem TW Modul die Befehlsadressen 7.10.0 bis 7.15.15 belegt.
Die Befehle 0.0.0 bis 0.0.7 sind generell für das Anwenderprogramm reserviert. Die Befehlsadressen 0.0.0 bis 0.0.4 enthalten immer die Parameterangaben zur Kommafestlegung und Ein/Ausgabe-Vorschriften sowie die Anfangsadressen von Codetabellen für Randeinheiten. Die Befehlsadressen 0.0.5 bis 0.0.7 müssen Befehle des Anwenderprogramms enthalten.
Die Befehle 0.0.8 bis 7.15.15 nehmen entweder die Moduln entsprechend den anzuschließenden Randeinheiten auf oder stehen für das Anwenderprogramm zur Verfügung.
- Platz 8: Magnetkernspeicher,
und zwar mit
- Betriebsprogramm MFA1 für die Typen 160, 163, 162, 161 und 165;
- Betriebsprogramm MFAGS1 für Typ 166.
Sofern das MFAGS1 zusammen mit Speichertypen 160 bis 163 und 165 eingesetzt ist, lassen sich keine Befehle des Anwenderprogramms im Magnetkernspeicher ansteuern.
- Platz 7:
6 und 5 : Rechner 154 und Standard-Ein/Ausgabe-Einheit
- Platz 4: Ein/Ausgabe-Einheit,
und zwar wahlweise Ein/Ausgabe-Einheit 184 oder Ein/Ausgabe-Einheit 310
- Platz 3: Ein/Ausgabe-Einheit,
und zwar beim
- Magnetkonten-Computer immer Ein/Ausgabe-Einheit 186
- Fakturier- und Abrechnungs-Computer Ein/Ausgabe-Einheit 310.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Platz 1
und 2: zur Zeit noch frei, evtl. vorgesehen für späteren on-line-Anschluß.

Die Plätze 3 und 4 können höchstens eine Ein/Ausgabe-Einheit 310 zum Anschluß von maximal zwei Magnetbandcassetten-Einheiten aufnehmen.

3.4.4. Chassis für Modell 820/25

Für das Modell 820/25 besteht die Auswahlmöglichkeit zwischen Chassis 207 und 208. Chassis 207 ist das standardmäßige Chassis, bei dem nur fest verdrahtete Befehle möglich sind. Demgegenüber läßt das Chassis 208 Anwenderprogrammbeefehle sowohl im Fest- als auch im Magnetkernspeicher zu.

3.4.4.1. Chassis 207

Das Chassis läßt sich folgendermaßen bestücken:

AP	AP	BP/ AP	BP	KSP	U- oder V- Platte	Rechner mit Standard E/A			E/A	E/A	E/A
12	11	10	9	8	7	6	5	4	3	2	1

Dabei ergeben sich folgende Kombinationsmöglichkeiten hinsichtlich des Anwender- und des Betriebsprogrammes in Festspeichern, der Daten im Magnetkernspeicher sowie der Verbindungs- bzw. Umschaltplatten.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Übersicht über die Plätze 7 bis 12 des Chassis 207

Platz 12 AP	Platz 11 AP	Platz 10 AP/BP	Platz 9 BP	Platz 8 KSP	Platz 7 V-Platte oder U-Platte	max. Bestückung
AP BL 0 AP BL 1	frei	frei	MSKZ 1 MSKZ 2/SKZ 2	16-512 SW	V-Platte 401	4096 BW/AP 4096 BW/BP 512 SW
AP BL 0 AP BL 1	frei	AP BL 2 AP BL 3	MSKZ 1 MSKZ 2	16-512 SW	U-Platte 402 Stecker 420/06	8192 BW/AP 4096 BW/BP 512 SW
AP BL 0 AP BL 1	frei	AP BL 2 MSKZ 3	MSKZ 1 MSKZ 2	16-512 SW	U-Platte 402 Stecker 420/06	6144 BW/AP 6144 BW/BP 512 SW
AP BL 0 AP BL 1	AP BL 2 AP BL 3	AP BL 4 AP BL 5	MSKZ 1 MSKZ 2/SKZ 2	16-512 SW	U-Platte 402 Stecker 420/09	12288 BW/AP 4096 BW/BP 512 SW
AP BL 0 AP BL 1	AP BL 2 AP BL 3	AP BL 4 MSKZ 3*)	MSKZ 1 MSKZ 2	16-512 SW	U-Platte 402 Stecker 420/09	10240 BW/AP 6144 BW/BP 512 SW
AP BL 0 Ringkernspeicher	AP BL 1	AP BL 2 AP BL 3	MSKZ 1 MSKZ 2	16-512 SW	U-Platte 402 Stecker 420/10	8192 BW/AP 4096 BW/BP 512 SW
AP BL 0 Ringkernspeicher	AP BL 1	AP BL 2 MSKZ 3	MSKZ 1 MSKZ 2	16-512 SW	U-Platte 402 Stecker 420/10	6144 BW/AP 6144 BW/BP 512 SW

*) MSKZ 3 läuft nicht mit SKZ 2, sondern nur mit MSKZ 2

AP = Anwenderprogramm

BP = Betriebsprogramm

KSP = Kernspeicher

V-Platte = Verbindungsplatte

MSKZ, SKZ = Betriebsprogramm

U-Platte = Umschaltplatte

BL = Block

SW = Speicherwort

BW = Befehlswort

	SYSTEMBESCHREIBUNG	
--	---------------------------	--

Die nachstehende Tabelle führt die Variationsmöglichkeiten für Ein/Ausgabe-Einheiten ohne Magnetknotenverarbeitung auf:

Platz 3	Platz 2	Platz 1	Zusätzlicher Anschluß zur Standardausstattung
E/A 184 1. Geräte - LKS - LKL - LSS/LSKS - LSL/LSKL	E/A 184 2. Geräte - LKS - LKL - LSS/LSKS - LSL/LSKL	E/A 184 2. Drucker	- 2 LKS - 2 LKL - 2 LSS oder 2 LSKS - 2 LSL oder 2 LSKL - 1 AKE - 2 Drucker
E/A 184 1. Geräte - LKS - LKL - LSS/LSKS - LSL/LSKL	E/A 184 2. Geräte - LKS - LKL - LSS/LSKS - LSL/LSKL	E/A 310 2 MBC	- 2 LKS - 2 LKL - 2 LSS oder 2 LSKS - 2 LSL oder 2 LSKL - 2 MBC - 1 AKE
E/A 184 1. Geräte - LKS - LKL - LSS/LSKS - LSL/LSKL	E/A 184 2. Drucker	E/A 310 2 MBC	- 1 LKS - 1 LKL - 1 LSS oder 1 LSKS - 1 LSL oder 1 LSKL - 2 MBC - 2 Drucker - 1 AKE
E/A 184 1. Geräte - LKS - LKL - LSS/LSKS - LSL/LSKL	E/A 310 2 MBC	E/A 310 2 MBC	- 1 LKS - 1 LKL - 1 LSS oder 1 LSKS - 1 LSL oder 1 LSKL - 4 MBC - 1 AKE
E/A 184 2. Drucker	E/A 310 2 MBC	E/A 310 2 MBC	- 4 MBC - 2 Drucker - 1 AKE

E/A = Ein/Ausgabe-Einheit	LSL = Lochstreifenleser
LKS = Lochkartenstanzer	LSKL = Lochstreifenkartenleser
LKL = Lochkartenleser	MBC = Magnetbandcassetten-Einheit
LSS = Lochstreifenstanzer	AKE = automatischer Konteneinzug
LSKS = Lochstreifenkartenstanzer	

1. 3. 1971

	SYSTEMBESCHREIBUNG	
--	---------------------------	--

Die nachstehende Tabelle führt die Variationsmöglichkeiten für Ein/Ausgabe-Einheiten mit Magnetkontenverarbeitung auf:

Platz 3	Platz 2	Platz 1	Zusätzlicher Anschluß zur Standardausstattung
E/A 184 1. Geräte - LKS - LKL - LSS/LSKS - LSL/LSKL	E/A 186 Magnetkonten-Einheit	E/A 184 2. Geräte - LKS - LKL - LSS/LSKS - LSL/LSKL	- 1 MKE - 2 LKS - 2 LKL - 2 LSS oder 2 LSKS - 2 LSL oder 2 LSKL
E/A 184 1. Geräte - LKS - LKL - LSS/LSKS - LSL/LSKL		E/A 184 2. Drucker	- 1 MKE - 1 LKS - 1 LKL - 1 LSS oder 1 LSKS - 1 LSL oder 1 LSKL - 2. Drucker
E/A 184 1. Geräte - LKS - LKL - LSS/LSKS - LSL/LSKL		E/A 310 2 MBC	- 1 MKE - 1 LKS - 1 LKL - 1 LSS oder 1 LSKS - 1 LSL oder 1 LSKL - 2 MBC
E/A 310 2 MBC		E/A 310 2 MBC	- 1 MKE - 4 MBC
E/A 184 2. Drucker		E/A 310 2 MBC	- 1 MKE - 2 MBC - 2. Drucker

E/A = Ein/Ausgabe-Einheit
 MKE = Magnetkonten-Einheit
 LKS = Lochkartenstanzer
 LKL = Lochkartenleser
 LSS = Lochstreifenstanzer
 LSKS = Lochstreifenkartenstanzer
 LSL = Lochstreifenleser
 LSKL = Lochstreifenkartenleser
 MBC = Magnetbandcassetten-Einheit

	SYSTEMBESCHREIBUNG	
--	--------------------	--

3.4.4.2. Chassis 208

Das Chassis läßt sich folgendermaßen bestücken:

AP	AP/ BP	BP	KSP	KSP	U- oder V- Platte	Rechner mit Standard Ein/Ausgabe- Einheit			E/A	E/A	E/A
12	11	10	9	8	7	6	5	4	3	2	1

Bedeutung der Abkürzungen

BP = Betriebsprogramm

AP = Anwenderprogramm

KSP = Kernspeicher

E/A = Ein/Ausgabe-Einheit

Dabei ergeben sich folgende Kombinationsmöglichkeiten hinsichtlich des Anwender- und des Betriebsprogrammes in Festspeichern, des Anwenderprogrammes und der Daten in Magnetkernspeichern sowie der Verbindungs- bzw. Umschaltplatten.

	SYSTEMBESCHREIBUNG	
--	---------------------------	--

Übersicht über die Plätze 7 bis 12 des Chassis 208

Platz 12 AP/BP 177	Platz 11 AP/BP 177	Platz 10 BP 177	Platz 9 KSP 166	Platz 8 KSP 160-166	Platz 7 V-Platte oder U-Platte	max. Bestückung
AP BL 1 Ringkernspeicher	AP BL 0	MSKZ 1 MSKZ 2	frei	16-512 SW	V-Platte 401	4096 BW/AP 4096 BW/BP 512 SW
AP BL 0 AP BL 1	frei	MSKZ 1 MSKZ 2/SKZ 2	frei	16-512 SW	V-Platte 401	4096 BW/AP 4096 BW/BP 512 SW
AP BL 0 AP BL 1	AP BL 2 MSKZ 3	MSKZ 1 MSKZ 2	frei	16-512 SW	U-Platte 402 Stecker 420/06	6144 BW/AP 6144 BW/BP 512 SW
AP BL 0 AP BL 1	AP BL 2 AP BL 3	MSKZ 1 MSKZ 2	frei	16-512 SW	U-Platte 402 Stecker 420/06	8192 BW/AP 4096 BW/BP 512 SW
frei MSKZ 6	MSKZ 3 AP BL 0	MSKZ 1 MSKZ 2	2048 BW	16-512 SW	U-Platte 402 Stecker 420/07	4096 BW/AP 8192 BW/BP 512 SW
frei MSKZ 6	AP BL 0 AP BL 1	MSKZ 1 SKZ 2	2048 BW	16-512 SW	U-Platte 402 Stecker 420/07	6144 BW/AP 6144 BW/BP 512 SW

Test-Ves.

- AP - Anwenderprogramm
- BP - Betriebsprogramm
- KSP - Kernspeicher
- V-Platte - Verbindungsplatte
- U-Platte - Umschaltplatte
- BL - Block
- SW - Speicherwort
- BW - Befehlswort

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Die nachstehende Tabelle führt die Variationsmöglichkeiten für Ein/Ausgabe-Einheiten ohne Magnetkontenverarbeitung auf:

Platz 3	Platz 2	Platz 1	Zusätzlicher Anschluß zur Standardausstattung
E/A 184 1. Geräte - LKS - LKL - LSS/LSKS - LSL/LSKL	E/A 184 2. Geräte - LKS - LKL - LSS/LSKS - LSL/LSKL	E/A 184 2. Drucker	- 2 LKS - 2 LKL - 2 LSS oder 2 LSKS - 2 LSL oder 2 LSKL - 1 AKE - 2. Drucker
E/A 184 1. Geräte - LKS - LKL - LSS/LSKS - LSL/LSKL	E/A 184 2. Geräte - LKS - LKL - LSS/LSKS - LSL/LSKL	E/A 310 2 MBC	- 2 LKS - 2 LKL - 2 LSS oder 2 LSKS - 2 LSL oder 2 LSKL - 2 MBC - 1 AKE
E/A 184 1. Geräte - LKS - LKL - LSS/LSKS - LSL/LSKL	E/A 184 2. Drucker	E/A 310 2 MBC	- 1 LKS - 1 LKL - 1 LSS oder 1 LSKS - 1 LSL oder 1 LSKL - 2 MBC - 2. Drucker - 1 AKE
E/A 184 1. Geräte - LKS - LKL - LSS/LSKS - LSL/LSKL	E/A 310 2 MBC	E/A 310 2 MBC	- 1 LKS - 1 LKL - 1 LSS oder 1 LSKS - 1 LSL oder 1 LSKL - 4 MBC - 1 AKE
E/A 184 2. Drucker	E/A 310 2 MBC	E/A 310 2 MBC	- 4 MBC - 2. Drucker - 1 AKE

<p>E/A = Ein/Ausgabe-Einheit LKS = Lochkartenstanzer LKL = Lochkartenleser LSS = Lochstreifenstanzer LSKS = Lochstreifenkartenstanzer</p>	<p>LSL = Lochstreifenleser LSKL = Lochstreifenkartenleser MBC = Magnetbandcassetten-Einheit AKE = automatischer Konteneinzug</p>
---	--

	SYSTEMBESCHREIBUNG	
--	---------------------------	--

Die nachstehende Tabelle führt die Variationsmöglichkeiten für Ein/Ausgabe-Einheiten mit Magnetkontenverarbeitung auf:

Platz 3	Platz 2	Platz 1	Zusätzlicher Anschluß zur Standardausstattung
E/A 184 1. Geräte - LKS - LKL - LSS/LSKS - LSL/LSKL	E/A 186 Magnetkonten-Einheit (MKE)	E/A 184 2. Geräte - LKS - LKL - LSS/LSKS - LSL/LSKL	- 1 MKE - 2 LKS - 2 LKL - 2 LSS oder 2 LSKS - 2 LSL oder 2 LSKL
E/A 184 1. Geräte - LKS - LKL - LSS/LSKS - LSL/LSKL		E/A 184 2. Drucker	- 1 MKE - 1 LKS - 1 LKL - 1 LSS oder 1 LSKS - 1 LSL oder 1 LSKL - 2. Drucker
E/A 184 1. Geräte - LKS - LKL - LSS/LSKS - LSL/LSKL		E/A 310 2 MBC	- 1 MKE - 1 LKS - 1 LKL - 1 LSS oder 1 LSKS - 1 LSL oder 1 LSKL - 2 MBC
E/A 310 2 MBC		E/A 310 2 MBC	- 1 MKE - 4 MBC
E/A 184 2. Drucker		E/A 310 2 MBC	- 1 MKE - 2 MBC - 2. Drucker

E/A = Ein/Ausgabe-Einheit
 MKE = Magnetkonten-Einheit
 LKS = Lochkartenstanzer
 LKL = Lochkartenleser
 LSS = Lochstreifenstanzer
 LSKS = Lochstreifenkartenstanzer
 LSL = Lochstreifenleser
 LSKL = Lochstreifenkartenleser
 MBC = Magnetbandcassetten-Einheit

	SYSTEMBESCHREIBUNG	
--	---------------------------	--

3.4.5. Chassis 540

BP 177 MSKZ 5 MSKZ 6)	BP 177 MSKZ 3 MSKZ 4	BP 177 MSKZ 1 MSKZ 2/ SKZ 2 **)	Umschalt- platte 402 Stecker 01	Rechner mit Standard E/A 154			E/A	E/A	E/A	E/A	E/A
12	11	10	9	8	7	6	5	4	3	2	1
KSP 160-163, 165, 166 **)	KSP 166 AP BL 2 oder SW 512-1023	KSP 166 AP BL 0	KSP 166 AP BL 1	AP	AP	AP	AP	AP	AP	AP	AP
SW 0-511				BL 6 BL 7	BL 4 BL 5	BL 2 BL 3	BL 0 BL 1				
20	19	18	17	16	15	14	13				

*) MSKZ6 läuft nur in Verbindung mit 2 Magnetkernspeichern 166 auf den Plätzen 19 und 20.

***) Magnetkernspeicher 164 lassen sich ebenfalls ansteuern.

****) MSKZ3 läuft nicht mit SKZA2, sondern nur mit MSKZ2.

Dabei ergeben sich folgende Möglichkeiten hinsichtlich der Bestückung mit Ein/Ausgabe-Einheiten:

Platz 5	Platz 4	Platz 3	Platz 2	Platz 1
E/A 184 1. Geräte - LKS - LKL - LSS/LSKS - LSL/LSKL	frei (ohne Magnet- kontenverarbei- tung) E/A 186 MKE (mit Magnet- kontenverarbei- tung)	E/A 184 2. Drucker	E/A 184 2. Geräte - LKS - LKL - LSS/LSKS - LSL/LSKL E/A 310 MBC 3 + 4	E A 310 MBC 1 + 2

LKS = Lochkartenstanzer
LKL = Lochkartenleser
LSS = Lochstreifenstanzer
LSKS = Lochstreifenkartenstanzer
LSL = Lochstreifenleser
LSKL = Lochstreifenkartenleser
MKE = Magnetkonten-Einheit

MBC = Magnetbandcassetten-
Einheit
E/A = Ein/Ausgabe-Einheit
BP = Betriebsprogramm
AP = Anwenderprogramm
MSKZ/SKZ = Betriebsprogramm

	SYSTEMBESCHREIBUNG	
--	--------------------	--

3.5. Netzteil 109

Für alle Modelle wird einheitlich das Netzteil 109 benötigt. Das Netzteil erzeugt aus der Netzspannung von 220 V die für den Betrieb der Zentraleinheit und den Randeinheiten erforderlichen Spannungen. Magnetische Stabilisatoren (Konstanthalter) im Netzteil gleichen Schwankungen der Netzspannung im Bereich von -15% bis +10% aus. Beim Absinken der Netzspannung unter -15% oder bei Ausfall der Netzspannung sorgt eine besondere Schaltung dafür, daß begonnene Operationen - außer E/A-Operation - zu Ende geführt werden, (s.a. 3.8.4.).

3.6. Umschalt- und Verbindungsplatten

3.6.1. Modell 820/15

Für das Chassis 205 sind keine Umschalt- bzw. Verbindungsplatten notwendig.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

3.6.2. Modelle 820/25 und 820/35

Bei den Kapazitäten an Betriebs- und Anwenderprogrammen der Modelle 820/25 und 820/35 ist es erforderlich, die verschiedenen Chassis mit Umschalt- bzw. Verbindungsplatten zu bestücken, damit die belegten Plätze überhaupt angesteuert werden können.

Für die unterschiedlichen Chassis sind folgende Platten zu verwenden (V = Verbindungsplatte, U = Umschaltplatte):

Chassis 208	Chassis 207	Chassis 540
V 401	V 401	U 402 Stecker 420/01
U 402 Stecker 420/06	U 402 Stecker 420/06	
U 402 Stecker 420/07	U 402 Stecker 420/07	
	U 402 Stecker 420/09	
	U 402 Stecker 420/10	

Der Einsatz der Umschalt- bzw. Verbindungsplatten ist aus Kapitel "3.4. Chassis" zu ersehen.

	SYSTEMBESCHREIBUNG	
--	---------------------------	--

3.7. Kabel

Kabel verbinden die Randeinheiten mit den Ein/Ausgabe-Einheiten. Die anzuschließenden Kabelarten sind abhängig von den Randeinheiten und den Ein/Ausgabe-Einheiten. Die nachstehende Tabelle zeigt die Auswahlmöglichkeiten auf:

Kabelübersicht

Baugr. Nr.	Bezeichnung	Verwendbar für:						erforderlich bei:									
		Fakturier- und Abrechn.-Comp.			Magnetkonten-Computer			Lochkarten-leser	Lochkarten-stanzer	Lochstreifenleser	Lochstreifenstanzer	2. Drucker	Papier-vorschub	automat. Kontenanz.	MB. Cas.	seitl.-Einheit	Magnetk.-tenschacht
		820/15	25	35	820/15	25	35										
276	Kabel	X	X	X		X	X						X	X			
277	Kabel		X	X		X	X					X					
283	Kabel	X	X	X	X	X	X	X	X								
284	Kabel	X	X	X	X	X	X	X	X								
285	Kabel	X	X	X	X	X	X										
289	Kabel	X	X	X	X	X	X								X		
293	Kabel	X	X	X	X	X	X	X	545 X	X	X						
294	Kabel	X	X	X	X	X	X			X	X						
295	Kabel	X	X	X	X	X	X	X	X	X	X						
296	Kabel	X	X	X	X	X	X	X	X	X	X						
297	Kabel	X	X	X	X	X	X					X					
298	Kabel				X	X	X									X	
299	Kabel	X	X	X	X	X	X	X	545 X								

	SYSTEMBESCHREIBUNG	
--	--------------------	--

3.8. Numerische Tastatur 017

3.8.1. Funktion und Aufbau

Die Tastatur 017 läßt sich mit einem Kabel 276 oder 297 an die Standard-Ein/Ausgabe-Einheit des Rechners anschließen und hat folgende Funktionen:

- Eingabe rein numerische Werte (Ziffern 0 bis 9),
- Manuelle Steuerung des Anwenderprogramms,
- Einschalten der Anlage, und zwar im Zusammenwirken mit der alpha-numerischen Tastatur,

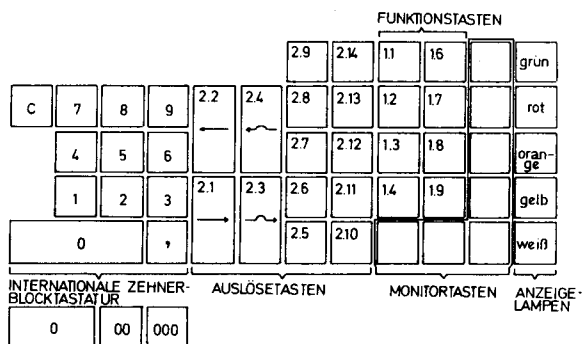
Auf der Tastatur befinden sich Tasten und Anzeigelampen. Die folgende Abbildung zeigt die Anordnung der Tasten und Lampen. Jede Taste wird mit einer Schlüsselzahl (Tastencode) gekennzeichnet.

Die numerische Eingabetastatur wird in folgende Gruppen eingeteilt:

- internationale Zehnerblocktastatur
 - . Tasten 1-9,
 - . C-Taste (Korrekturtaste),
 - . Kommataste,
 - . Tasten 0 und 00,
 - . statt der Kommataste wahlweise die Taste 000,
- 14 Auslösetasten (2.1-2.14),
- 8 Funktionstasten (1.1-1.9),
- F-Taste,
- 7 Monitor-Tasten (bei Modell 820/15 bedeutungslos),
- 5 Anzeigelampen.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Numerische Eingabetastatur



3.8.2. Arbeitsweise

Die Tastatur 017 arbeitet mit kontaktlosen Tasten, d.h. mit Tasten, die Impulse auf magnetischem Wege und nicht mit Hilfe von mechanischen Kontakten erzeugen. Kontaktlose Tasten haben folgende Vorteile: Sie verschmutzen oder oxydieren nicht und verbrennen nicht. Infolgedessen arbeiten die elektromagnetischen Tasten weniger stör anfällig, praktisch sogar störungsfrei. Impulse werden von der Taste ausgelöst und gelangen über die Codierschaltung als "Codewort" in die Ein/Ausgabe-Einheit des Rechners. Jede Taste und der zugehörige Impuls wird mit Hilfe einer dezimalen Schlüsselzahl (Tastencode) gekennzeichnet. Befehle unterscheiden die Tasten anhand des Schlüsselcodes. Ein Teil der Tasten (2.1 bis 2.12) wird unter bestimmten Voraussetzungen soweit vom Betriebs- als auch vom Anwenderprogramm gesperrt. Andere Tasten (1.1 bis 1.9) sind mechanisch einrastbar.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Die Zifferntasten lassen sich nicht durch das Anwenderprogramm ansprechen.

Das Betriebsprogramm fragt mit Hilfe der Uhr im Rechner (5 ms) in regelmäßigen Abständen ab, ob eine Eingabe über die Tastatur vorliegt.

3.8.3. Internationale Zehnerblock-Tastatur

3.8.3.1. Zifferntasten 0-9

Die Ziffern- und Nullentasten dienen ausschließlich der manuellen Dateneingabe numerischer Werte. Eingetastete Ziffern werden im Eingabewort des Magnetkernspeichers abgestellt. Bei richtiger Programmierung im Anwenderprogramm läßt sich die Eingabe kontrolliert-überlappt absichern. Über die "kontrolliert-überlappte-Eingabe" ist es möglich, den nächsten Wert bereits einzugeben, während das Programm den vorherigen verarbeitet.

Da das Anwenderprogramm die Zifferntasten nicht anspricht, ist die Beendigung der manuellen Eingabe mit Hilfe einer Auslöse- und Funktionstaste anzuzeigen. Allein die Auslösetasten (2.1 bis 2.14) führen zur überlappt-kontrollierten Eingabe, während die übrigen Funktionstasten zwar eine überlappte, aber keine kontrollierte Eingabe bewirken.

3.8.3.2. C - Taste

Die C-Taste heißt auch Korrekturtaste und hat folgende Aufgaben:

- Löschen von eingetasteten Werten im Eingabewort, wenn die rote Lampe nicht brennt.
- Löschen der roten Lampe und von eingetasteten Werten im Eingabewort.
- Einschalten der Rechenanlage.

Das Einschalten geht immer nur in Verbindung mit dem Einschalter an der alphanumerischen Tastatur.

Ein besonderer Wartebefehl für die C-Taste bewirkt, daß sich die Anlage im Wartezustand befindet und daß gleichzeitig die rote Anzeigelampe brennt, und zwar so lange, bis die C-Taste betätigt wird.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Löschen von Werten im Eingabewort, wenn die rote Lampe nicht brennt
Sofern die Bedienung, bevor sie die Auslösetaste hat, merkt, daß sie sich vertippt hat, kann sie die bisherige Eingabe ohne weiteres mit einem Druck auf die C-Taste löschen und die Eingabe anschließend wiederholen.

Löschen der roten Lampe und von Werten im Eingabewort

In diesem Fall hat der spezielle Übertragungsbefehl für das Eingabewort bereits die letzte Dateneingabe im Eingabewort angesprochen und mit Hilfe des Betriebsprogramms oder einer Irrtumsschutzprogrammierung im Anwenderprogramm festgestellt, daß entweder eine Kapazitätsüberschreitung im Eingabewort bzw. bei den zulässigen Nachkommastellen vorliegt. Diese Art von Fehler blockiert die Zehnertastatur. Nach einem Druck auf die C-Taste läßt sich der Fehler korrigieren. Dabei wird die rote Lampe gelöscht. Außerdem ist die Tastatur nicht mehr gesperrt. Die Bedienung kann den Wert neu eingeben.

3.8.4.

Das Einschalten der Anlage

Hierbei gibt es in Verbindung mit der Alphatastatur zwei Möglichkeiten:

- Einschalten auf den Programmanfang.
Dazu sind die C-Taste und die 3-Nullen-Tasten gleichzeitig zu drücken.
- Einschalten nach Programmunterbrechung auf den nächstfolgenden Befehl.

Nach einer Programmunterbrechung läßt sich die Arbeit mit einem Druck auf die C-Taste (allein) genau dort fortsetzen, wo das Programm angehalten hat.

Bei einer Programmunterbrechung, sei es durch Fehler (z.B. Stromausfall) oder durch absichtliches manuelles Abschalten (z.B. zur Pause), führt die Anlage, mit Ausnahme von Ein/Ausgabe-Operationen, einen begonnenen Befehl bis zum Ende durch, weil das Netzteil mit seinen Kondensatoren dafür ausreichenden Strom speichert. Das Betriebsprogramm hält nach dem Abschalten der Anlage die Adresse des Befehls fest, der als nächster auszuführen ist. Drückt die Bedienung nach einem Programmstop die C-Taste, in Verbindung mit dem Einschalter, dann führt die Anlage den Befehl aus, den sich das Betriebsprogramm laut Adreßregister im Rechner abrufft. Das bedeutet, daß sich ein Programm nach einer Unterbrechung mit dem richtigen Befehl fortsetzen läßt und

	SYSTEMBESCHREIBUNG	
--	--------------------	--

daß die Arbeit nicht von Anfang an wiederholt werden muß.

3.8.5. F - Taste

Die Fehler- oder F-Taste befindet sich auf der Tastatur rechts oben (vgl. Abb. Seite S 62). Sie hat eine ähnliche Funktion wie die C-Taste, ist aber statt mit der roten mit der gelben Anzeigelampe gekoppelt. Die gelbe Lampe brennt, um einen Programmstop anzuzeigen, der auftritt, wenn eine der folgenden Bedingungen erfüllt ist.

- Das Betriebsprogramm erkennt einen Fehler während einer maschinellen Daten-Ein/Ausgabe.

- Für das Anwenderprogramm gibt es den Befehl "Warte auf F-Taste". Bei einem Maschinenfehler läßt sich die gelbe Lampe mit einem Druck auf die F-Taste nur dann löschen, wenn der Fehler zuvor behoben wurde.

Bei einem Wartebefehl auf die F-Taste, geht das Programm mit dem nächsten Befehl weiter, sobald die F-Taste betätigt wird.

3.8.6. Anzeigelampen

3.8.6.1. Grüne Lampe

Die ohne Unterbrechungen leuchtende grüne Lampe zeigt den Betriebszustand der Maschine an.

Das Blinken der grünen Lampe zeigt u.a. an, daß nach Drücken der C-Taste das Programm an der Stelle der Unterbrechung fortgesetzt werden kann.

3.8.6.2. Rote Lampe

Sofern laut Anwenderprogramm die rote Lampe mit einem Befehl aufleuchtet, kann nach Drücken der C-Taste das Anwenderprogramm den auf den Wartebefehl folgenden Befehl ausführen.

Sofern das Betriebsprogramm eine unzulässige manuelle oder Kartenlesungs-Eingabe festgestellt hat, kann nach Drücken der C-Taste eine neue Eingabe erfolgen.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

3.8.6.3. Orange Lampe

Vom Anwenderprogramm wurde die orange Lampe mit einem Befehl gesetzt. Nach Drücken der F-Taste wird das Anwenderprogramm mit dem auf den Wartebefehl folgenden Befehl fortgesetzt. Externfehlerstop, z.B. Fehler beim Serialdrucker oder Lochkarte klemmt. Nach Behebung des Fehlers kann nach Drücken der F-Taste das Programm fortgesetzt werden.

3.8.6.4. Gelbe Lampe

Die leuchtende gelbe Lampe zeigt den Programmablauf in der Testebene an.

3.8.6.5. Weiße Lampe

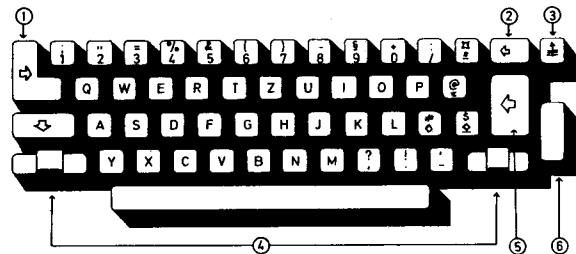
Die leuchtende weiße Lampe zeigt an, daß in der Monitorebene eine Eingabe über die Tastatur für den Monitor-Betrieb möglich ist.

3.8.6.6. Rote und orange Lampe

Internfehlerstop. Das Betriebsprogramm hat einen unzulässigen Befehlscode festgestellt. Eine Korrektur mit anschließender Wiederholung des Befehls, der zum Internfehlerstop führte, ist nicht möglich. Das Anwenderprogramm kann nach Ausschalten und erneutem Einschalten der Maschinen von Anfang an neu gestartet werden. In der Monitorebene entfällt die Notwendigkeit des Ausschaltens und Wiedereinschaltens der Maschine; nach Drücken der entsprechenden Monitor-taste wird der Programmablauf neu gestartet.

SYSTEMBESCHREIBUNG

3.9. Alphanumerische Tastatur



- 1 = Tabulationstaste (TAB-Taste) 5 = Wagenaufzugstaste (WZ-Taste)
2 = Rücktaste 6 = Einschalter
3 = manuelle Walzenschaltung
4 = Umschalttaste

3.9.1. Allgemeines

Die alphanumerische Tastatur dient zur manuellen Eingabe von Buchstaben, Ziffern und Zeichen in den Magnetkernspeicher. Diese Tastatur ist mit dem Serialdrucker verbunden. Es gibt bei der Eingabe verschiedene Möglichkeiten:

- Eingabe und Drucken ohne Speicherung,
- Eingabe und Drucken mit Speichern im 8-Bit-Code,
 - bei den Modellen 820/25 und 820/35 auch im 6-Bit-Code.
- Eingabe und Drucken eines alphanumerischen Wertes und Speichern im 8-Bit-Code,
 - bei den Modellen 820/25 und 820/35 auch im 6-Bit-Code.

Die Tastatur ist während des Programmablaufes mit Ausnahme nach Eingabe-Befehlen für die alphanumerische Tastatur, gesperrt.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

3.9.2. Sonderfunktionen von verschiedenen Tasten

- Bei der alphanumerischen Eingabetastatur gibt es normalerweise nur Großbuchstaben. In diesem Fall darf die Umschalttaste bei Buchstaben nicht benutzt werden. Es gibt aber auch spezielle Kugelköpfe mit Korrespondenzschriften, die große und kleine Buchstaben aufweisen.
- Die Tabulations-Taste (TAB-Taste) und die Wagenaufzugtaste (WZ-Taste) beenden eine vom Anwenderprogramm erteilte Tastatur-Freigabe. Beim Druck der TAB-Taste bleibt der Kugelkopf nach der letzten Druckposition stehen, während bei der WZ-Taste der Kugelkopf auf die Ausgangsposition zurückkehrt.
- Die Rücktaste bewirkt eine Positionierung auf die derzeitige Position minus 1. Sie ist nur innerhalb einer Tastatur-Freigabe wirksam, wenn nicht durch das Anwenderprogramm ein entsprechender Befehl diese Wirkung auch außerhalb der Freigabe ermöglicht.
- Die manuelle Walzenschaltung bewirkt bei angeschalteter Maschine, ohne Rücksicht auf die Tastatur-Sperre fortlaufende Walzenschaltungen. Die Vorschubeinrichtung wird hierdurch nicht angesprochen.
- Durch die Betätigung der Einschalttaste wird die Maschine mit dem Stromnetz verbunden. Beim Einschalten ist folgendes zu beachten:
 - Einschalten auf den Programmablauf
Dazu sind die C-Taste und die 3-Nullen-Taste bzw. Kommataste gleichzeitig zu drücken.
 - Einschalten nach Programmunterbrechung läßt sich die Arbeit mit einem Druck auf die C-Taste genau dort fortsetzen, wo das Programm angehalten hat.

3.10. Drucker

3.10.1. Bestandteile

Der Drucker verarbeitet je nach Ausstattung Endlospapier und/oder Einzelformulare und druckt zeichenweise (Serial-Drucker). Die Hauptbestandteile des Druckers sind:

- | | |
|--|-----------------------|
| - Kugelkopf | - Druckwalze |
| - elektronische Positionierungseinrichtung | - Vorschubeinrichtung |
| | - Mechanik |

	SYSTEMBESCHREIBUNG	
--	--------------------	--

3.10.2. Arbeitsweise

Der Drucker arbeitet

- gleichzeitig mit einer manuellen Dateneingabe über die alphanumerische Tastatur,
- aufgrund von Befehlen des Anwenderprogrammes.

Jedes Druckzeichen wird mit Hilfe von sieben Magneten angewählt. Sechs Magnete stellen das Zeichen dar, während der 7. Magnet generell die Buchstaben- bzw. Zifferauswahl übernimmt. Der 7. Magnet dreht den Kugelkopf entweder auf die Hälfte mit den Ziffern und Zeichen oder auf die Hälfte der Buchstaben. Der Serialdrucker arbeitet mit der maximalen Geschwindigkeit von 15,6 Zeichen/sec.

Für alle sieben Magnete sind Rückmelder vorhanden, die dem Betriebsprogramm melden, ob ein Magnet angesprochen wurde oder nicht. Das Betriebsprogramm vergleicht den Code des zu druckenden Zeichens mit dem Code der Rückmeldung (Echomeldung), stoppt bei ungleichem Ergebnis das Programm und zeigt mit der roten Lampe den Fehler an. Außerdem sind im Fehlerfall beide Tastaturen gesperrt. Durch Ausschalten der Anlage und Wiedereinschalten mit der C-Taste läßt sich innerhalb des Betriebsprogramms der Druckbefehl für das letzte Zeichen wiederholen.

3.10.3. Positionieren des Kugelkopfes

Eine elektronische Positionierungsscheibe steuert die Stellung des Kopfes auf die jeweilige Druckposition.

Es gibt maximal 131 Druckstellen, die mit Druckstelle Null (ganz links) bis Druckstelle 130 (ganz rechts) bezeichnet werden.

Die Positionierungsscheibe bewegt sich gleichzeitig mit dem Kugelkopf. Elf Fotozellen erkennen anhand einer Hell/Dunkelkombination aus schwarzen und weißen Flächen, auf welcher Druckposition der Kugelkopf steht.

SYSTEMBESCHREIBUNG

ELEKTRONISCHE
POSITIONIERUNGSSCHEIBE

Es gibt die Positionierung im Einzelschritt und die "fliegende Positionierung". Die fliegende Positionierung geht über mehrere Druckpositionen und läuft wesentlich schneller als die Positionierung im Einzelschritt ab.

Nach jedem Positionierungsbefehl vergleicht das Betriebsprogramm den Sollwert gegen den Istwert des Kugelkopfes.

Ist der Sollwert kleiner als der Istwert, dann findet eine Positionierung nach links statt. Die Sollposition wird überlaufen und von links nach rechts angelaufen. Der Kugelkopf stoppt zwei Positionen vor der Sollposition (Zielposition) und geht dann im Einzelschritt auf die echte Sollposition weiter.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Ist der Sollwert größer als der Istwert, dann findet eine fliegende Positionierung nach rechts statt. Das Betriebsprogramm gibt in diesem Fall zunächst die Wagenaufzugssperre frei und positioniert wie vorstehend beschrieben nach rechts.

3.10.4. Papiertransport

An den Drucker lassen sich verschiedene Einrichtungen zum Papiertransport anschließen, und zwar:

- Walze einfach.
- Stachelwalze.
- Walze und einfacher Formularvorschub (Leporello 1).
- Walze und doppelter Formularvorschub (Leporello 2).

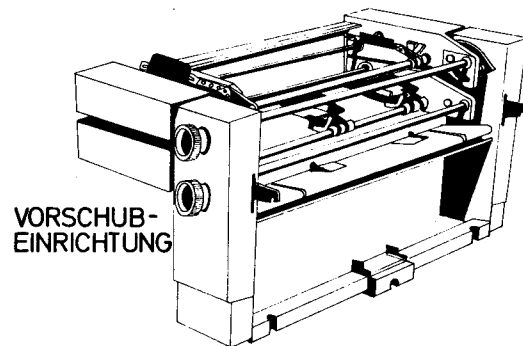
3.10.4.1. Walze

Jede Maschine ist mit einer Schreibwalze ausgestattet. Die Walze dient zur Formularführung und als Unterlage für den Kugelkopf. Wird die Walze als Formularträger eingesetzt, so kann zur Verarbeitung von Einzelformularen ein Konteneinzug für das Formular vorhanden sein. Bei Endlosformularen greifen Transportstacheln in die Transportlochungen ein, um das Papier vorwärtszubewegen (Leporello).

	SYSTEMBESCHREIBUNG	
--	--------------------	--

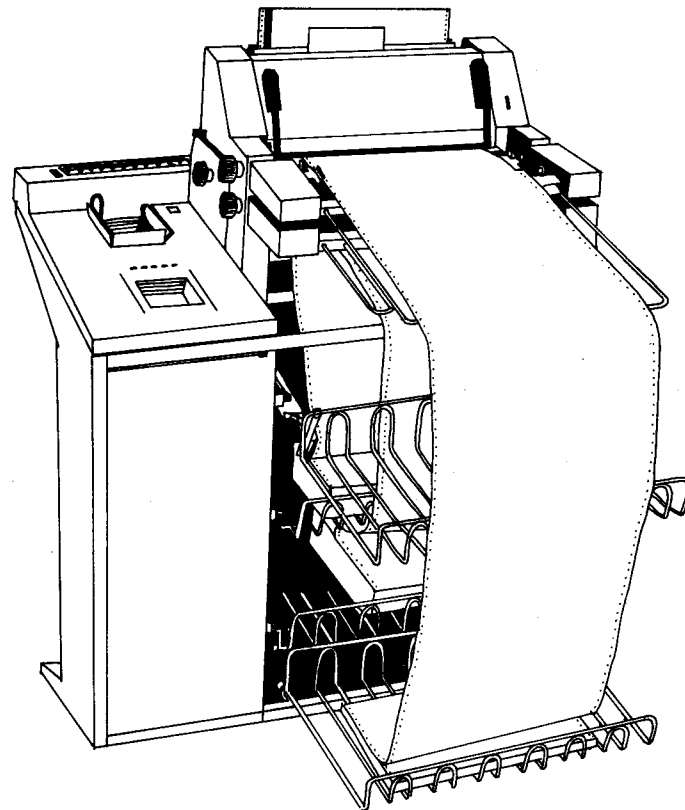
3.10.4.2. Vorschubeinrichtung

An die Serialdrucker 026 und 028 läßt sich eine einfache und doppelte Vorschubeinrichtung zur Verarbeitung von Endlospapier anschließen. Der Vorteil gegenüber der Stachelwalze ist, daß jetzt die Möglichkeit besteht, zwei unabhängig voneinander laufende Formulare zu transportieren.



	SYSTEMBESCHREIBUNG	
--	--------------------	--

Magnetknoten-Computer mit Vorschubeinrichtung



	SYSTEMBESCHREIBUNG	
--	--------------------	--

4. Beispiele für Systemkonfigurationen

4.1. Übersicht

Jedes der drei Modelle 820/15, 820/25 und 820/35 stellt ein in sich geschlossenes System dar. Trotzdem vollzieht sich der Übergang von einem Modell zum anderen sowohl Hard- als auch softwaremäßig nahtlos. Dadurch, daß bei einer Erweiterung der Anlagen über die maximale Konfiguration hinaus, lediglich die Bauelemente der bisherigen Anlage in ein größeres Chassis eingesetzt werden müssen, ist eine Erweiterung von der kleinsten Konfiguration bis zur größten Ausstattung einfach durchzuführen.

Das NIXDORF Computer System der Serie 820 läßt ein breites Angebot von Konfigurationen zu, wobei aufgrund der Typenbezeichnung drei Versionen zu unterscheiden sind:

- Fakturier-, Abrechnungs- und Magnetkonten-Computer (820/15, 820/25 und 820/35),
- Monitor-Computer (820/25, 820/35),
- Assembler-Monitor-Computer (820/35).

Monitor-Anlagen dienen zum Testen von Programmen.

Assembler-Monitor-Anlagen bieten folgende Einsatzmöglichkeiten:

- Testen von Programmen, wie die Monitor-Anlage.
(über das Betriebsprogramm MSKZ6)
- Umwandeln von Programmen, deren Befehle in der Assembler-Sprache geschrieben sind,
in die Maschinen-Sprache.
(Betriebsprogramm MSKZ4 und MSKZ5)

Für das Modell 820/15 gibt es spezielle Dienstprogramme (Utilities im Anwenderprogramm), die das Austesten von Anwenderprogrammen in beschränktem Umfang ermöglichen. Sind die Utilities nicht vorhanden, so können Programme für Modelle 820/15 über Monitor-Computer 820/25 oder 820/35 getestet werden.

Assemblieren ist für die Anlagen 820/15 und 820/25 nur über Assembler-Monitor-Anlagen 820/35 möglich.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

4.2. Modell 820/15

4.2.1. Kleinste Konfiguration

Die kleinste Konfiguration belegt nur die Chassisplätze 5 bis 9

Chassis 205

frei	AP/ BP	KSP	Rechner mit Standard E/A			frei	frei	frei	frei
10	9	8	7	6	5	4	3	2	1

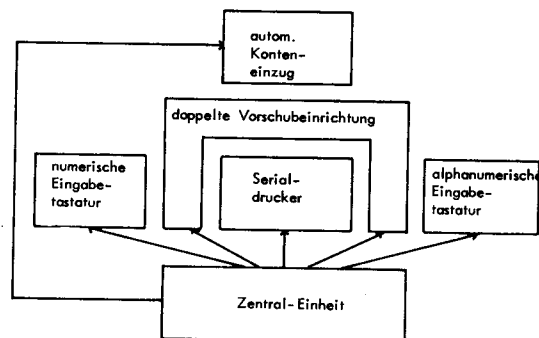
AP = Anwenderprogramm
BP = Betriebsprogramm
KSP = Kernspeicher

Diese Chassisbestückung erlaubt folgende Ausstattung:

- die beiden Tastaturen,
- Serialdrucker mit doppeltem Papiervorschub,
- automatischer Konteneinzug.

Die Anzahl der Speicherworte bzw. Befehle (fest und frei) richtet sich unter anderem nach der Größe des Magnetkernspeichers. Es sind maximal 1952 fest verdrahtete Befehle für das Anwenderprogramm möglich.

Für Befehle im Magnetkernspeicher ist das Betriebsprogramm um den GP-Modul zu ergänzen. Die Befehle für den automatischen Konteneinzug stehen im ST-Modul.



	SYSTEMBESCHREIBUNG	
--	--------------------	--

4.2.2. Maximale Konfigurationsbeispiele

4.2.2.1. Fakturier- und Abrechnungs-Computer in Verbindung mit Magnetbandcassetten-Einheit, mit Lochkarten- und Lochstreifen-Randeinheiten, mit Magnetkernspeicher 165 und Betriebsprogramm MFA1

Chassis 205

AP	AP/ MFA1	KSP	Rechner mit Standard E/A			E/A	E/A	frei	frei
177		165				184	310		
10	9	8	7	6	5	4	3	2	1

AP = Anwenderprogramm

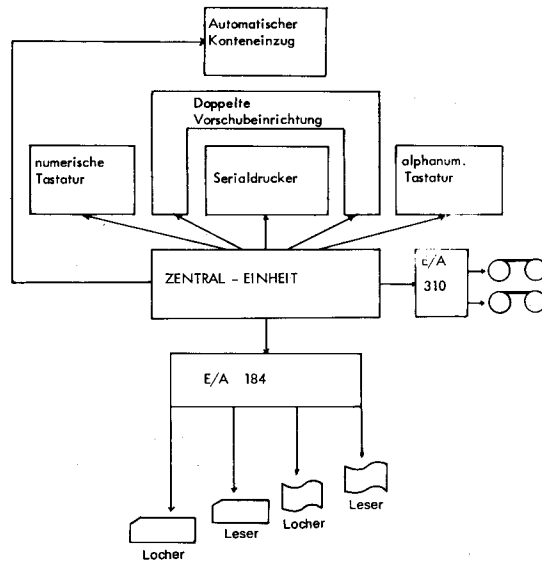
MFA1= Betriebsprogramm

KSP = Kernspeicher

Nachstehende Konfiguration erlaubt folgende Ausstattung:

- Die beiden Tastaturen
- Serialdrucker mit doppelter Vorschubeinrichtung,
- automatischer Konteneinzug,
- 2 Magnetbandcassetten-Einheiten,
- 1 Lochstreifenstanzer oder 1 Lochstreifenkartenstanzer,
- 1 Lochstreifenleser oder 1 Lochstreifenkartenleser,
- 1 Lochkartenstanzer,
- 1 Lochkartenleser,
- rund 5000 Befehle für das Anwenderprogramm (etwa 4500 Befehle im Festspeicher und 500 Befehle im Magnetkernspeicher),
- rund 120 Speicherworte zur Datenspeicherung von etwa 950 alpha-numerischen Zeichen (oder 1900 numerische Zeichen).

	SYSTEMBESCHREIBUNG	
--	--------------------	--



An die Ein/Ausgabe-Einheit 184 lassen sich alle vier oder auch nur eine beliebige Auswahl der vier Randeinheiten anschließen. Zu den Moduln gehören in jedem Fall noch die Code-Tabellen.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

4.2.2.2. Magnetkonten-Computer in Verbindung mit Magnetbandcassetten-Einheiten, mit Magnetkernspeicher 165 und Betriebsprogramm MFA1

Chassis 205

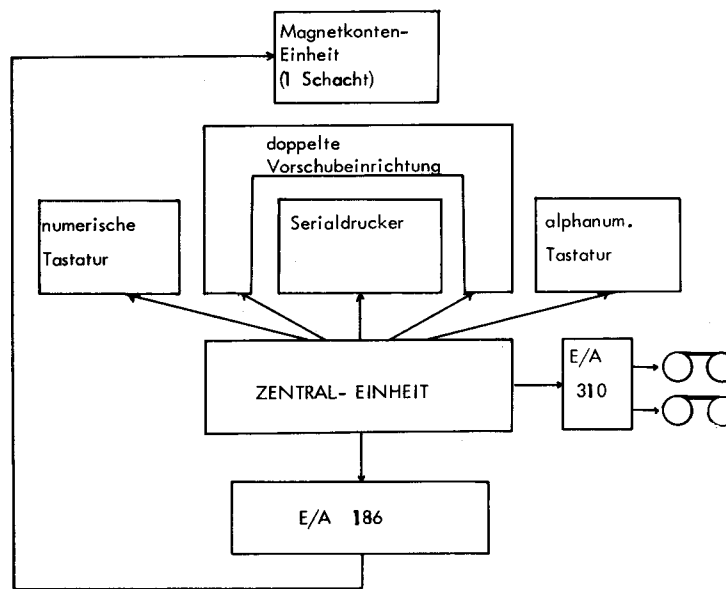
AP	AP/ MFA1	KSP	Rechner mit Standard E/A			E/A	E/A	frei	frei
177	177	165				310	186		
10	9	8	7	6	5	4	3	2	1

AP = Anwenderprogramm
MFA1 = Betriebsprogramm
KSP = Kernspeicher

Diese Konfiguration erlaubt folgende Ausstattung:

- Die beiden Tastaturen,
- Serialdrucker mit doppelter Vorschubeinrichtung,
- Magnetkonten-Einheit (ein Schacht),
- 2 Magnetbandcassetten-Einheiten,
- rund 5600 Befehle für das Anwenderprogramm (etwa 5100 Befehle im Festspeicher und 500 Befehle im Magnetkernspeicher),
- rund 120 Speicherworte zur Datenspeicherung von etwa 950 alpha-numerischen Zeichen (oder ca. 1900 numerische Zeichen).

SYSTEMBESCHREIBUNG



4.2.2.3. Magnetkonten-Computer in Verbindung mit Lochkarten- und Lochstreifen-Randeinheiten, mit Magnetkernspeicher 166 und Betriebsprogramm MFAGS1

Chassis 205

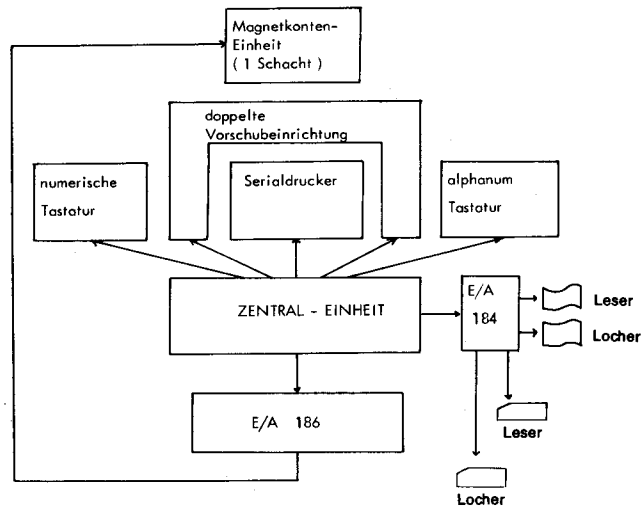
AP	AP/ MFA- GS1	KSP	Rechner mit Standard E/A			E/A	E/A	frei	frei
177	177	166				184	186		
10	9	8	7	6	5	4	3	2	1

AP = Anwenderprogramm, MFAGS1 = Betriebsprogramm, KSP = Kernspeicher

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Diese Konfiguration erlaubt folgende Ausstattung:

- Die beiden Tastaturen,
 - Serialdrucker mit doppelter Vorschubeinrichtung,
 - Magnetknoten-Einheit (1 Schacht),
 - 1 Lochstreifenstanzer oder 1 Lochstreifenkartenstanzer,
 - 1 Lochstreifenleser oder 1 Lochstreifenkartenleser,
 - 1 Lochkartenstanzer,
 - 1 Lochkartenleser,
 - rund 5000 Befehle für das Anwenderprogramm (etwa 4000 Befehle im Festspeicher und 1000 Befehle im Magnetkernspeicher),
 - rund 250 Speicherwort zur Datenspeicherung von etwa 2000 alpha-numerischen Zeichen (oder 4000 numerische Zeichen).
- Zu den Moduln gehören in jedem Fall noch die Code-Tabellen.



	SYSTEMBESCHREIBUNG	
--	--------------------	--

4.3. Modell 820/25

4.3.1. Vorbemerkung

Für das Modell gibt es die beiden Chasistypen 207 (Standard-Chassis), das nur festprogrammierte Befehle zuläßt, während das Chassis 208 Befehle des Anwenderprogrammes im Fest- und Magnetkernspeicher zuläßt.
Die Monitoversion 820/25 setzt das Chassis 208 voraus.

4.3.2. Chassis 207, maximale Konfiguration als Magnetknoten-Computer, fest programmiertes Anwenderprogramm

Chassis 207

AP	AP	AP/BP	BP	KSP	U	Rechner mit Standard E/A			E/A	E/A	E/A
177	177	177	177	186	402				184	186	310
AP BL 0	AP BL 2	MSKZ 3	MSKZ 1	SW/	Stecker						
AP BL 1	AP BL 3	AP BL 4	MSKZ 2	AP	420/10						
12	11	10	9	8	7	6	5	4	3	2	1

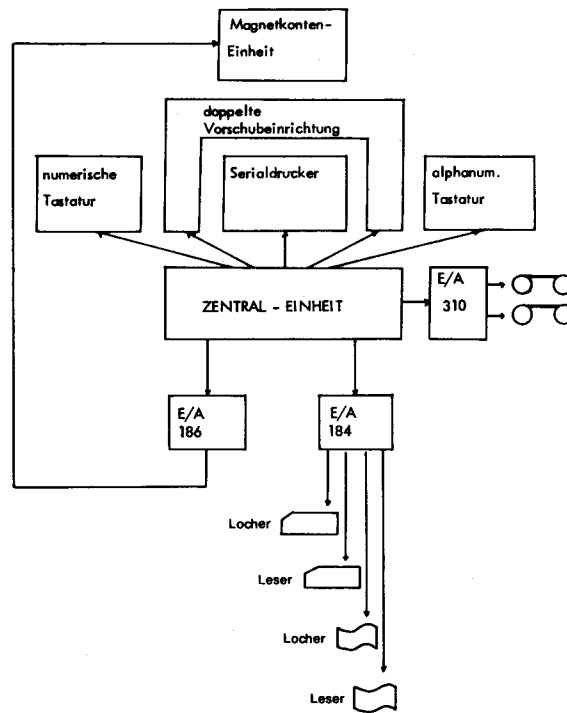
- AP = Anwenderprogramm
- BL = Block
- BP = Betriebsprogramm
- MSKZ = Betriebsprogramm
- KSP = Kernspeicher
- SW = Speicherwort
- U = Umschalt-Platte

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Diese Konfiguration erlaubt folgende Ausstattung:

- Die beiden Tastaturen,
- Serialdrucker,
- Magnetknoten-Einheit (2 Schächte),
- 1 Lochstreifenstanzer oder 1 Lochstreifenkartenstanzer,
- 1 Lochstreifenleser oder 1 Lochstreifenkartenleser,
- 1 Lochkartenstanzer,
- 1 Lochkartenleser,
- 2 Magnetbandcassetten-Einheiten,
- 6144 Befehle, Betriebsprogramm (MSKZ1, MSKZ2, MSKZ3),
- 10240 Befehle, Anwenderprogramm (fest),
- 512 Speicherworte.

SYSTEMBESCHREIBUNG



	SYSTEMBESCHREIBUNG	
--	--------------------	--

4.3.3. Chassis 208, kleinste Konfiguration als Fakturier- und Abrechnungs-Computer, frei programmiertes Anwenderprogramm

Chassis 208

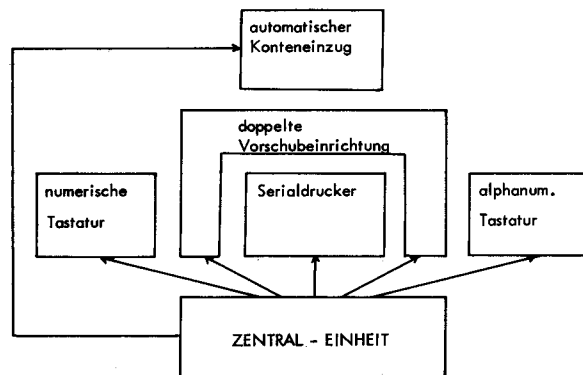
frei	frei	BP 177	KSP 166	KSP 160- 166	U 402	Rechner mit Standard E/A			frei	frei	frei
		MSKZ 1 SKZ 2	BW/ AP BL 0	SW/ AP AP 0-511	Stecker 420/07						
12	11	10	9	8	7	6	5	4	3	2	1

BP = Betriebsprogramm
 MSKZ1, SKZ2 = Betriebsprogramm
 KSP = Kernspeicher
 BW = Befehlsword
 AP = Anwenderprogramm
 BL = Block
 SW = Speicherwort
 U = Umschalt-Platte

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Diese Konfiguration erlaubt folgende Ausstattung:

- Die beiden Tastaturen,
- Serialdrucker mit automatischem Konteneinzug,
- Anwenderprogramm mit 2048 Befehlen (frei programmiert),
- Speicherworte je nach Magnetkernspeichertyp auf Platz 8.



	SYSTEMBESCHREIBUNG	
--	--------------------	--

4.3.3.1. Monitor-Anlage 820/25

Als Monitor-Anlage ist das Chassis 208 folgendermaßen zu bestücken:

Chassis 208

AP/BP	AP/BP	BP	KSP	KSP	U	Rechner mit Standard E/A			BELIEBIG			
177	177	177	166	160-166	402				E/A	E/A	E/A	
MSKZ 6 frei	MSKZ 3 AP BLD	MSKZ 1 SKZ 2	BW/AP 2048	SW/AP 0-511	Stecker 420/07							
12	11	10	9	8	7	6	5	4	3	2	1	

- AP = Anwenderprogramm
- BP = Betriebsprogramm
- MSKZ, SKZ = Betriebsprogramm
- KSP = Kernspeicher
- BW = Befehlsword
- SW = Speicherword
- U = Umschalt-Platte

Diese Konfiguration ermöglicht das Austesten von 2048 Befehlen im Magnetkernspeicher.

	SYSTEMBESCHREIBUNG	
--	--------------------	--

4.4. Modell 820/35

4.4.1. Übersicht

Für das Modell 820/35 gibt es folgende Variationsmöglichkeiten:

- Fakturier- und Abrechnungs-Computer,
- Magnetkonten-Computer,
- Monitor-Anlage,
- Assembler-Monitor-Anlage.

4.4.2. Maximale Ausstattung als Magnetkonten-Computer

Chassis 540

	BP	BP	U	Rechner mit Standard E/A			E/A	E/A	E/A	E/A	E/A
frei	177	177	402				184	188	184	184	310
	MSKZ 3 frei	MSKZ 1 MSKZ 2	Stecker 420/01				Erst- geräte	MKE	Zweit- drucker	Zweit- geräte	MBC 1 + 2
12	11	10	9	8	7	6	5	4	3	2	1
KSP 160-163, 165, 166 SW/AP 0-511		KSP 166 AP BL 2		KSP 166 AP BL 0		KSP 166 AP BL 1		AP 177 BL 6 BL 7	AP 177 BL 4 BL 5	AP 177 BL 2 BL 3	AP 177 BL 0 BL 1
20		19		18		17		16	15	14	13

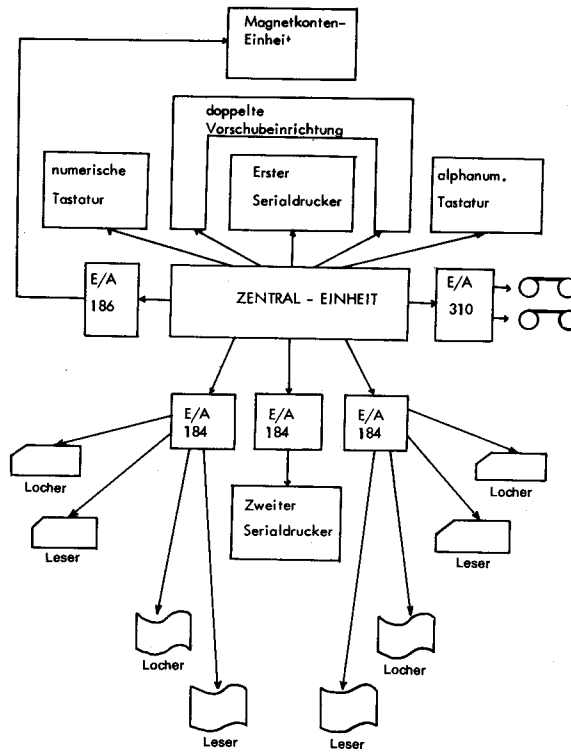
- | | |
|-----------------------------------|-----------------------|
| BP = Betriebsprogramm | KSP = Kernspeicher |
| U = Umschalt-Platte | AP = Anwenderprogramm |
| MKE = Magnetkonten-Einheit | SW = Speicherwort |
| MBC = Magnetbandcassetten-Einheit | BL = Block |

	SYSTEMBESCHREIBUNG	
--	--------------------	--

Diese Konfiguration erlaubt folgende Ausstattung:

- 2 Tastaturen,
- 2 Drucker,
- 1 Magnetkarten-Einheit,
- 2 Lochstreifenstanzer oder 2 Lochstreifenkartenstanzer,
- 2 Lochstreifenleser oder 2 Lochstreifenkartenleser,
- 2 Lochkartenstanzer,
- 2 Lochkartenleser,
- 2 Magnetbandcassetten-Einheiten,
- 22 528 Befehle für das Anwenderprogramm (davon 6 144 frei programmierbar),
- 512 Speicherworte.

	SYSTEMBESCHREIBUNG	
--	--------------------	--



	SYSTEMBESCHREIBUNG	
--	--------------------	--

4.4.3. Monitor-Anlage 820/35

Als Monitor-Anlage ändert sich die vorstehende Chassis-Bestückung nur auf Platz 12, der einen 177er Festspeicher mit dem Programmträger für das Betriebsprogramm MSKZ6 aufnimmt (2048 Befehle für das Betriebsprogramm).

4.4.4. Assembler-Monitor-Anlage

Als Assembler-Monitor-Anlage erfolgt zur Bestückung des Chassis 540 in Abschnitt 4.4.2. auf

Platz 11 und 12 folgende Erweiterung:

Platz 11: MSKZ4 (2048 Befehle für das Betriebsprogramm)

Platz 12: MSKZ5 und MSKZ6 (4096 Befehle für das Betriebsprogramm)

Diese maximale Konfiguration bietet folgende Befehlskapazität:

- 12288 Befehle, Betriebsprogramm (fest),
- 22528 Befehle, Anwenderprogramm (frei und fest).

Mit dieser kompletten Auslastung des Modells 820/35 steht ein Computer zur Verfügung, der aufgrund seiner Speicherkapazität eigentlich nicht mehr zu den Computern der Mittleren Datenverarbeitung zählt.

NIXDORF
COMPUTER

Der NIXDORF-Hochleistungsdrucker 20817
für das Datenverarbeitungssystem 820

	Inhaltsverzeichnis	
--	--------------------	--

Inhaltsverzeichnis

1. Einleitung
2. Der technische Aufbau
 - 2.1. Der Drucker
 - 2.2. Das Druckwerk
 - 2.3. Der Zeichenvorrat
 - 2.4. Die Zeilenlänge
 - 2.5. Druck- und Rücklaufgeschwindigkeit
 - 2.6. Das Farbband
 - 2.7. Durchschlagsfähigkeit
 - 2.8. Formulartransport
 - 2.9. Akustischer Alarm
 - 2.10. Bedienungselemente
 - 2.10.1. EIN-AUS-Schalter
 - 2.10.2. Grundstellung Formular
 - 2.10.3. START-Schalter
 - 2.10.4. Formular-Ende-Unterdrückung
 - 2.10.5. Die obere Leuchtanzeige
 - 2.10.6. HDWE ALARM
 - 2.10.7. Papier-Ende
 - 2.11. Manuelle Bedienungselemente
 - 2.11.1. Formular-Einzugs-Hebel
 - 2.11.2. Formular-Walzenfreilauf
 - 2.11.3. Formular-Stärke-Einstellung
3. Anlegen des Vorschublochstreifens
4. Bedienungsanleitung
5. Programmierung
 - 5.1. Übertragen von Zeichen
 - 5.2. USASCII-Code-Tabelle
 - 5.3. Beschreibung des Funktionscodes
 - 5.3.1. Druckauslösung
 - 5.3.2. Formularanfang
 - 5.3.3. Vertikalposition
 - 5.3.4. Zeilenvorschub
 - 5.3.5. Programmierter Alarm
 - 5.3.6. Verlängerte Schrift
 - 5.3.7. Druckbereich löschen
 - 5.4. Befehlsliste / Liste der Befehle / Beschreibung der Befehle

	Einleitung	
--	------------	--

1. Einleitung

Der NIXDORF-Hochleistungsdrucker ist ein Zweitdrucker. Durch seine hohe Ausgabegeschwindigkeit wird er zur idealen Ergänzung von Anlagen, mit denen Daten aus Datenträgern wie Lochkarte, Lochstreifen, Magnetkonto, Magnetbandcassette, Magnetplatte usw. eingelesen werden und zeitsparend auf Endlosformulare ausgegeben werden sollen.

Der NIXDORF-Hochleistungsdrucker kann an die Modelle 820/25, 820/35 und 880/65 angeschlossen werden.

	Der technische Aufbau	
--	-----------------------	--

2. Der technische Aufbau

Die folgenden Punkte geben eine Übersicht über den technischen Aufbau des Druckers sowie eine kurze Bedienungsanleitung.

2.1 Der Drucker

Der NIXDORF-Hochleistungsdrucker ist ein Serial-Matrix-Drucker (Nadel-Drucker) mit einer Schreibbreite von max. 132 Zeichen. Dabei handelt es sich um eine geschlossene Einheit, die verschiedene Baugruppen enthält.

Mechanische Baugruppen

z. B.
Druckwerk,
Farbband-Transport,
Formular-Transport,
akustischer Alarm,
Bedienungselemente
usw.

Elektronische Baugruppen

z. B.
Stromversorgung,
Kontroll-Logik,
Zeichen-Muster-Generator,
Einzel-Zeilen-Ausgabebereich
(Druckbereich),
usw.

2.2 Das Druckwerk

Das Druckwerk besteht aus einem Druckkopf und 7 Drucksolenoids mit daran angebrachten Drucknadeln. Alle diese Teile sind auf einem beweglichen Druckschlitten montiert, der sie horizontal vor der Druckwalze hin und her transportieren kann.

Die 7 Drucknadeln werden durch den Druckkopf in einer Senkrechten übereinander ausgerichtet. Werden die Solenoids von der Steuer-Elektronik aktiviert, so schießen die betreffenden Drucknadeln hervor und bilden über ein vor dem Papier vorbeigeführten Farbband punktförmige Anschläge auf dem Papier ab. Die Solenoids werden, je nach Zeichen, bis zu 5 mal pro Zeichen aktiviert. Dabei bewegt sich der Druckschlitten von links nach rechts.

Ein einheitlicher Zeichenabstand wird durch eine Synchronisation des Druckschlittens mit einem optischen Leser erreicht. Dieser Leser tastet während des Druckvorgangs einen Positionstreifen ab. Durch die Lage des Positionstreifens sind auch die seitlichen Begrenzungen für das 1. bzw. 132. Zeichen festgelegt.

	Der technische Aufbau	
--	-----------------------	--

Bei etwaiger Fehlfunktion (Überschreiten der Begrenzungen durch den Druckschlitten) wird die Anlage über einen Sicherheitsschalter automatisch abgeschaltet, ein akustisches Alarmzeichen gegeben und die Hardware-Anzeige zum Aufleuchten gebracht. Nach dem Beheben des Fehlers kann die Arbeit wieder aufgenommen werden. Bei wiederholten Störungen ist ein Kundendienstmechaniker hinzuzuziehen.

2.3. Der Zeichenvorrat

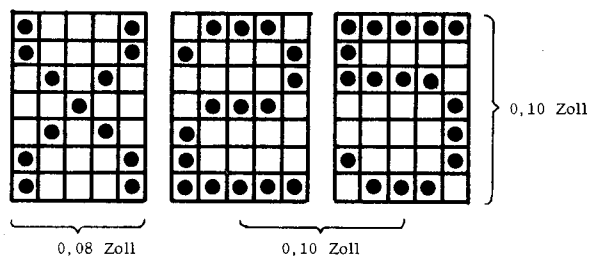
Der Zeichenvorrat des NIXDORF-Hochleistungsdruckers besteht aus 60 verschiedenen Zeichen im USASCII-Code.

- 10 Ziffern (0-9)
- 26 Großbuchstaben (A-Z)
- 20 Sonderzeichen

Die Liste der Zeichen, sowie ihre Zuordnung zum ALC-Code und zur Code-Tabelle im Anwenderprogramm ist der Tabelle unter 5.2. zu entnehmen.

Jedes Zeichen wird aus einzelnen Punkten zu einem Raster (Matrix) zusammengesetzt. Der Raster besteht bei normaler Schrift aus max. 5 x 7 Punkten.

Beispiel:



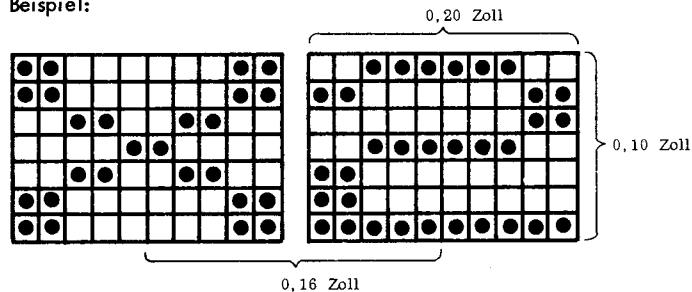
	Der technische Aufbau	
--	-----------------------	--

Der horizontale Abstand der Zeichen beträgt zwischen zwei Zeichenmitten 0,10 Zoll. Das entspricht der üblichen Teilung von 10 Zeichen je Zoll.

Der vertikale Abstand der Zeichen beträgt, bedingt durch die Höhe einer Zeilenschaltung, zwischen zwei Zeichenmitten 0,166 Zoll, was der üblichen Teilung von 6 Zeilen pro Zoll entspricht.

Der Hochleistungsdrucker bietet die Möglichkeit, bestimmte Texte durch verlängerte Schrift hervorzuheben. In diesem Falle wird die Zeichenbreite verdoppelt. Das geschieht, indem jede senkrechte Nadelanschlag-Kombination doppelt ausgeführt wird, wodurch sich das Raster auf maximal 10 x 7 Punkte je Zeichen erweitert.

Beispiel:



2.4. Die Zeilenlänge

Der Hochleistungsdrucker kann in einer Zeile bei normaler Schrift 132 Zeichen, bei verlängerter Schrift 66 Zeichen ausgeben.

Die Anzahl der auszugebenden Zeichen wird vom Anwenderprogramm bestimmt. Sie kann innerhalb der oben angegebenen Grenzen beliebig sein. Verkürzte Zeilen bringen Zeitersparnis.

	Der technische Aufbau	
--	-----------------------	--

2.5. Druck- und Rücklaufgeschwindigkeit

Die Druckgeschwindigkeit (Lauf des Druckschlittens von links nach rechts) ist konstant und beträgt etwa 41 cm/sec., was einer Ausgabe von etwa 165 Zeichen/sec. entspricht.

Die Rücklaufgeschwindigkeit des Druckschlittens (von rechts nach links) beträgt etwa das 4-fache der Druckgeschwindigkeit.

Aus diesen Geschwindigkeiten und der Zeilenlänge ergibt sich beispielsweise für den Druck einer vollen DIN-A-4 Zeile einschließlich Druckwerksrücklauf die Zeit von ca. 0,6 Sekunden (ca. 6000 Zeilen/Stunde).

2.6. Das Farbband

Der NIXDORF-Hochleistungsdrucker verwendet ein spezielles schwarzes Farbband von 1 Zoll Breite. Es wird leicht schräg vor dem Papier vorbeigeführt. Dadurch ergibt sich ein hoher Ausnutzungsgrad und eine lange Lebensdauer des Farbbandes.

2.7. Durchschlagsfähigkeit

Trotz der hohen Horizontalgeschwindigkeit des Druckschlittens wird ein Nadel-Anschlag erzielt, der für 5 Nutzen ausreicht.

2.8. Formulartransport

Der NIXDORF-Hochleistungsdrucker bedruckt Endlosformulare mit der üblichen Transportlochung von 2 Löchern je Zoll.

Er transportiert das Papier mit einer vertikalen Formulartransport-Einrichtung, die aus zwei Vorschubraupen (Formulartraktor) besteht. Der Abstand der beiden Vorschubraupen ist stufenlos verstellbar, so daß verschieden breite Formulare von mindestens 4 Zoll (10,8 cm) bis höchstens 14 1/2 Zoll (36,54 cm) Breite eingespannt werden können.

Die Höhe einer Zeile ist 1/6 Zoll. Die Vorschubgeschwindigkeit beträgt 24 Zeilen je Sekunde.

	Der technische Aufbau	
--	-----------------------	--

Eine reibungslos ablaufende Papierablage (Stapelung) hängt weitgehend von der Beschaffenheit der Formulare und der Umgebung ab. Die günstigsten Bedingungen liegen in folgenden Bereichen:

Raumtemperatur: 10° bis 38 ° C
Relative Luftfeuchtigkeit: 25% bis 62%.

Trotzdem könnengelegentliche Eingriffe der Bedienungskraft zur Ordnung der Formularstapelung erforderlich sein, insbesondere, wenn die oben genannten Grenzwerte in der einen oder anderen Richtung überschritten werden.

Es gibt 4 unterschiedliche Arten des Papiervorschubs:

- Automatische Schaltung einer Zeile im Anschluß an die Ausgabe einer Druckzeile (s. 5.3.1.)
- Zeilenschaltung ohne Druckausgabe (s. 5.3.4.)
- Vorschub des Papiers auf die erste Zeile des folgenden Formulars (s. 5.3.2.)
- Vorschub des Papiers zur nächsten festgelegten Vertikalposition des gleichen Formulars (5.3.3.)

Um die beiden letztgenannten Vorschubarten zu ermöglichen, wird ein Vorschublochband eingesetzt. Dabei handelt es sich um einen Standard-Lochstreifen, bei dem Kanal 7 dem Vorschub zum Anfang des nächsten Formulars und Kanal 5 dem Vorschub zur nächsten Vertikalposition zugeordnet sind. Das Anfertigen und Einsetzen des Vorschublochbandes wird unter Punkt 3. beschrieben.

2.9. Akustischer Alarm

Beim Hochleistungsdrucker werden Fehler durch ein akustisches Zeichen (2 Sekunden Summton) gemeldet.

Dieser Alarm kann auf 3 verschiedene Arten ausgelöst werden:

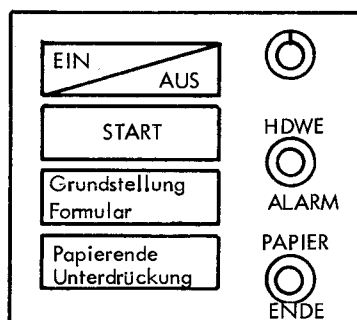
	Der technische Aufbau	
--	-----------------------	--

- Programmierter Alarm
Durch den Funktionscode BE kann innerhalb des Anwenderprogramms der akustische Alarm programmiert werden. Dadurch wird die Bedienungskraft auf Fehler aufmerksam gemacht.
- Papier-Ende oder Fehlzuführung
In diesem Falle wird der Summton von der Hardware ausgelöst.
- Überschreitung der seitlichen Begrenzungen
Der Summton wird von der Hardware ausgelöst und macht auf eine Überschreitung der seitlichen Begrenzungen durch den Druckschlitten aufmerksam. Ferner wird das dadurch hervorgerufene Abschalten der Anlage angezeigt.

In jedem Fall muß der Fehler von der Bedienungskraft beseitigt werden.

2.10. Bedienungselemente

Am Hochleistungsdrucker befindet sich eine Bedienungs-Schalttafel, die 4 Schalter und 3 Leuchtanzeigen enthält.



2.10.1. EIN-AUS-Schalter

Der Schalter leuchtet, wenn der Drucker eingeschaltet ist.

	Der technische Aufbau	
--	-----------------------	--

2.10.2. Grundstellung Formular

Dieser Schalter dient dem manuellen Vorschub zum nächsten Formularanfang oder der genauen Ausrichtung des Vorschublochbandes beim Einlegen neuer Formulare.

2.10.3. START-Schalter

Nach Drücken dieses Schalters ist der Drucker zur Aufnahme und Ausgabe von Daten bereit. Bei Druck des START-Schalters wird der Druckbereich gelöscht.

2.10.4. Papierende-Unterdrückung

Mit diesem Schalter kann die Bedienungskraft die Papierende-Meldung überwählen, so daß begonnene Formulare noch zu Ende verarbeitet werden können.

2.10.5. Die obere Leuchtanzeige ist für spezielle Zwecke bisher freigehalten.

2.10.6. HDWE ALARM. Diese Leuchtanzeige zeigt an, daß der Druckschlitten den linken oder rechten Begrenzungsschalter überschritten hat.

2.10.7. PAPIER ENDE. Die Leuchtanzeige leuchtet bei Papier-Ende oder Papier-Fehlzuführung auf.

2.11. Manuelle Bedienungselemente

Zusätzlich zu den Schaltern der Bedienungsschalttafel hat der Hochleistungsdrucker folgende Elemente zur manuellen Bedienung:

2.11.1. Formular-Einzugs-Hebel

Ein Hebel an der rechten Seite der Maschinenverkleidung gestattet eine schnelle Zuführung neuer Formulare um die Walze und in die Papiervorschubraupen.

Blatt HW 12

Datenverarbeitungssystem 820

1.9. 1971

Hochleistungsdrucker

	Der technische Aufbau	
--	-----------------------	--

2.11.2. Formular-Walzenfreilauf

Der Knopf an der linken Seite des Gehäuses dient zur manuellen Vor- und Rückpositionierung des Formulars.

2.11.3. Formular-Stärke-Einstellung

Am Druckwerk befindet sich eine Skala mit Bezugspunkten. Je nach Stärke des Formulars bzw. Formularsatzes kann der Abstand zwischen Walze und Druckkopf eingestellt werden.

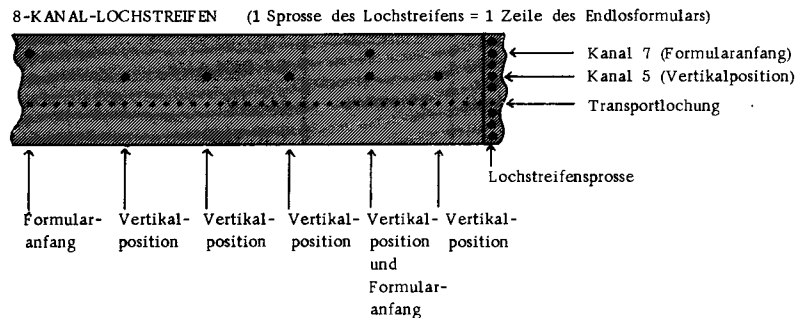
	Anlegen des Vorschublochstreifens	
--	-----------------------------------	--

3. Anlegen des Vorschublochstreifens

Der Vorschublochstreifen, der zur Vertikalpositionierung des Endlosformulars benutzt wird, ist ein schwarzer, lichtundurchlässiger 8-Kanal-Lochstreifen mit der üblichen Transportlochung zwischen dem 3. und 4. Kanal. Jede Sprosse des Lochstreifens entspricht einer Zeile des zu bedruckenden Formulars.

Der Lochstreifenleser des Hochleistungsdruckers liest nur die Kanäle 7 und 5. Eine Lochung in Kanal 7 bedeutet Anfang des zu bedruckenden Formulars. Auf Druck der Taste Grundstellung Formular oder eine entsprechende Angabe im Anwenderprogramm wird der Vorschublochstreifen und das Papier soweit vorgeschoben, bis eine Lochung in Kanal 7 erkannt wird.

Eine Lochung in Kanal 5 bedeutet eine Vertikalpositionierung. Auf eine entsprechende Angabe im Anwenderprogramm wird der Vorschublochstreifen solange weitergeschoben, bis eine Lochung in Kanal 5 erkannt wird.



Dieser Vorschublochstreifen kann mit dem auf den folgenden Codierformularen codierten Programm auf einer Anlage 820/35 selbst hergestellt werden.

Blatt HW 14

1.9. 1971

NIXDORF SERIE 820**MASCHINEN-SPRACHE**

Programm: Programm zur Erstellung von Vorschublochstreifen									Blocknr.	Blatt	1
Programmierer									Firma		Datum
BW-Adresse			Op. Teil		I	Adreßteil		Symb. Adr.	Bemerkungen		
0	00	00	0	00	0	0	00	00		KGA 0	
		1									
		2									
		3	0	00	0	0	02	10		Anfangsadresse LS-Tabelle	
		4									
		5	2	14	0	2	08	05	ANFANG	LF, LEP, UP, SAVPL, 5	
		6	2	12	0	0	02	01	EINGAB	WT, MRAR	
		7	0	01	0	0	01	00		ACC, 1.	
		8	2	02	0	0	00	01		XF, 1	
		9	2	06	0	3	01	15		XR4, 3.1.15	
		10	2	10	0	0	09	03		CX, 147	
		11	2	02	0	1	00	01		CPX, 1	
		12	1	06	0	0	00	15		BRL, ABFRA2	
		13	2	12	0	0	00	00	ROLAMP	WTC	
		14	1	00	0	0	00	06		BR, EINGAB	
		15	2	10	0	0	15	15	ABFRA2	CX, 255	
0	01	00	2	02	0	1	01	00		CPX, 1.	
		1	1	06	0	0	01	03		BRL, DRUZH	
		2	1	00	0	0	00	13		BR, ROLAMP	
		3	3	02	0	0	00	10	DRUZH	ED, 10	
		4	1	15	0	0	09	04		CA, 148	
		5	0	06	0	0	01	03		SBH, 1.3	
		6	2	06	0	3	01	15		XR4, 3.1.15	
		7	2	14	0	2	08	02		LF, LEP, UP, SAVPL, 2	
		8	3	11	0	0	05	00		BPX, 5.	
		9	2	13	0	0	00	01		TAB, 1	
		10	3	13	0	0	08	00		ALIN6, 128	
		11	2	14	0	2	08	01		LF, LEP, UP, SAVPL, 1	
		12	2	13	0	0	00	01		TAB, 1	
		13	3	13	1	0	01	06		ALIN6, X1, 22	
		14	3	11	0	0	05	00		BPX, 5.	
		15	3	13	0	4	07	15		ALPT, CD6, 127	

PR 4

Programm: Programm zur Erstellung von Vorschublochstreifen								Blocknr.	Blatt 2	
Programmierer								Firma		Datum
BW-Adresse		Op. Teil	I	Adreßteil			Symb. Adr.	Bemerkungen		
0	2	00	2	12	0	0	08	01	WT, MPT	
		1	3	11	0	1	01	02	BPX, 1.1.2	
		2	3	13	1	4	01	05	ALPT, CD6, XI, 21	
		3	2	10	0	3	14	06	CX, 3.14.6	
		4	2	15	0	4	00	00	ENDLOS ALC, PT, 0	
		5	2	10	1	0	00	01	CX, XI, 1	
		6	1	09	0	0	02	08	BXG, ROTELA	
		7	1	00	0	0	02	04	BR, ENDLOS	
		8	2	12	0	0	00	00	ROTELA WTC	
		9	1	00	0	0	00	05	BR, ANFANG	
		10	0	00	1	1	08	00	LSTAB Lochstreifencode-Tabelle (0 = nur Trans- portloch.)	
		11	0	00	1	1	13	00	(1 = Formularanfang + Vertikalposition)	
		12	0	00	1	1	04	00	(2 = Formularanfang)	
		13	0	00	1	1	01	00	(3 = Vertikalposition)	
		14								
		15								
		0								
		1								
		2								
		3								
		4								
		5								
		6								
		7								
		8								
		9								
		10								
		11								
		12								
		13								
		14								
		15								

Anlegen des Vorschublochstreifens

Nach Eingabe des Programms in den Kernspeicher kann über die Zehner-tastatur und Taste → eine Zahl zwischen 148 und 254 (Anzahl der zu stanzenen Sprossen des Lochstreifens) eingegeben werden, die angibt, für wieviele Zeilen der Vorschublochstreifen ausgelegt werden soll. Dabei entspricht eine Sprosse des Lochstreifens einer Zeile des Formulars. Nach dem Ausdruck der eingegebenen Zahl erfolgt eine Freigabe der alpha-numerischen Tastatur. Über diese wird nun für jede Zeile des zu bedruckenden Formulars eine Ziffer eingegeben.

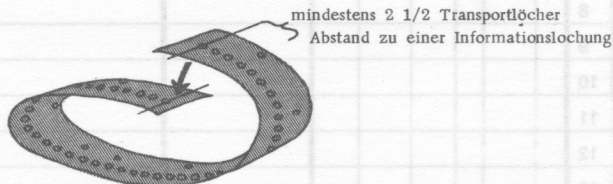
- 0 = Keine Lochung (nur Transportlochung)
- 1 = Formularanfang und Vertikalpositionierung
- 2 = Formularanfang
- 3 = Vertikalpositionierung

Andere Eingaben führen zu Stanzfehlern.

Die Schreibmaschinenfreigabe wird nach der ersten eingegebenen Zeile und bei Erreichen der Anzahl mit der WZ-Taste beendet.

Anschließend wird über den angeschlossenen Lochstreifenstanzer der Lochstreifen ausgegeben. Es muß unbedingt darauf geachtet werden, daß schwarzer, lichtundurchlässiger Lochstreifen verwandt wird, da sonst Fehler beim Lesen auftreten.

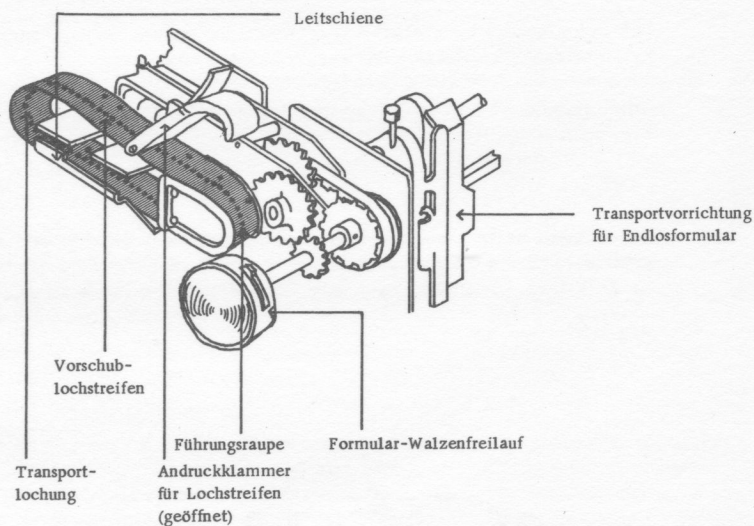
Dieser Lochstreifen wird $2\frac{1}{2}$ Transportlöcher lang überlappend zusammengeklebt. Dabei muß darauf geachtet werden, daß der Streifen mindestens 15 Zoll (148 Transportlöcher) lang ist und daß die beiden Schnittkanten mindestens $2\frac{1}{2}$ Transportlöcher von einer Informationslochung entfernt sind.



	Anlegen des Vorschublochstreifens	
--	-----------------------------------	--

Der auf diese Weise entstandene Endloslochstreifen wird wie folgt eingelegt.

- obere linke Verkleidung der Anlage abnehmen.
- Klammer nach oben klappen
- Lochstreifen mit der Transportlochung in die Führungsraupe und die Leitschiene einlegen
- Klammer schließen
- Lochstreifen durch Druck der Taste Grundstellung Formular bis zur Formularanfangslochung vorlaufen lassen.
- Verkleidung schließen.



	Bedienungsanleitung	
--	---------------------	--

4. Bedienungsanleitung

Voraussetzung zur Inbetriebnahme des Hochleistungsdruckers als Zweitdrucker an die Modelle 820/25, 820/35 und 880/65 sind:

- Bestückung der Anlage mit dem Betriebsprogramm MSKZ1/HD anstelle des MSKZ1. Dieses Betriebsprogramm ist mit dem MSKZ2 bzw. MSKZ2/LU oder mit dem SKZA2 zu kombinieren.
- Anschluß des Hochleistungsdruckers an die E/A-Einheit 333 (Chassisplatz 2. Serialdrucker) über ein Kabel Nr. 452. Die Stromversorgung erfolgt über ein eigenes Netzkabel.

Sind diese Voraussetzungen erfüllt, und ist der Computer eingeschaltet, so wird der Hochleistungsdrucker wie folgt in Betrieb genommen.

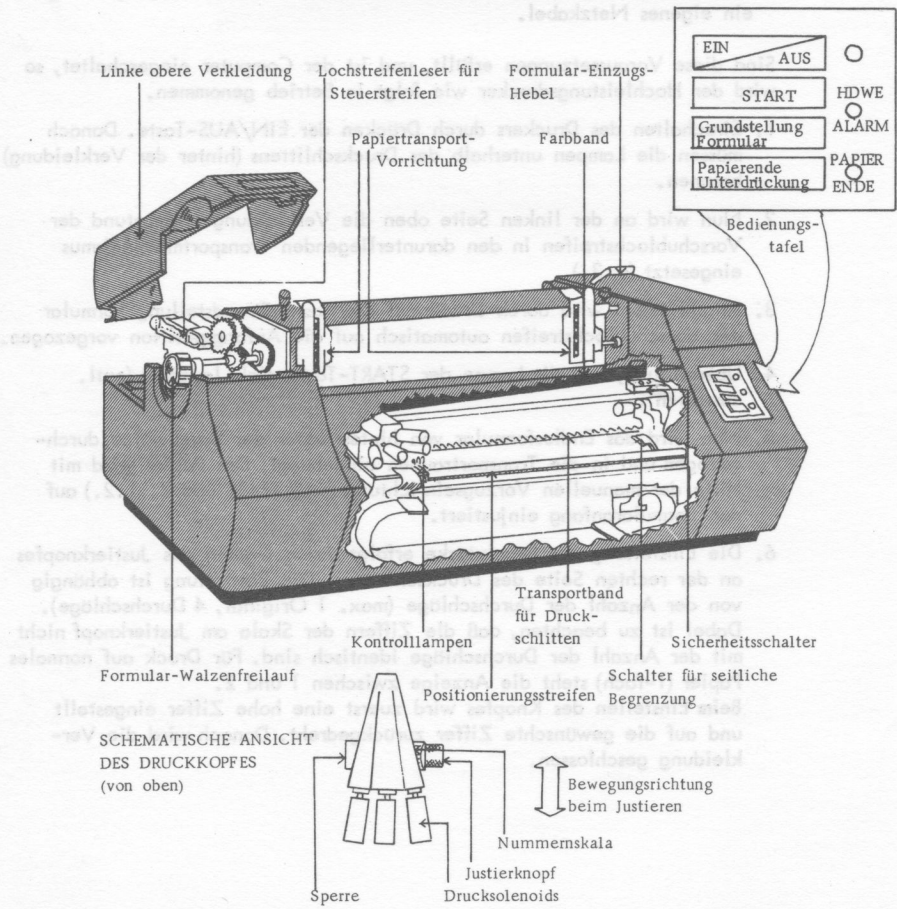
1. Einschalten des Druckers durch Drücken der EIN/AUS-Taste. Danach müssen die Lampen unterhalb des Druckschlittens (hinter der Verkleidung) brennen.
2. Nun wird an der linken Seite oben die Verkleidung entfernt und der Vorschubblockstreifen in den darunterliegenden Transportmechanismus eingesetzt (s. 3.).
3. Anschließend wird durch Druck auf die Taste Grundstellung Formular der Vorschubblockstreifen automatisch auf die Anfangsposition vorgezogen.
4. Überprüfung, daß die Lampe der START-Taste nicht leuchtet (evtl. ausrasten).
5. Nun wird das Endlosformular von hinten unter der Druckwalze durchgezogen und in die Transportraupen eingehängt. Das Papier wird mit Hilfe der manuellen Vorzugseinrichtung (s. 2.11.1. und 2.11.2.) auf den Formularanfang einjustiert.
6. Die Einstellung der Druckstärke erfolgt durch Drehen des Justierknopfes an der rechten Seite des Druckschlittens. Die Einstellung ist abhängig von der Anzahl der Durchschläge (max. 1 Original, 4 Durchschläge). Dabei ist zu beachten, daß die Ziffern der Skala am Justierknopf nicht mit der Anzahl der Durchschläge identisch sind. Für Druck auf normales Papier (1-fach) steht die Anzeige zwischen 1 und 2. Beim Einstellen des Knopfes wird zuerst eine hohe Ziffer eingestellt und auf die gewünschte Ziffer zurückgedreht. Danach wird die Verkleidung geschlossen.

	Bedienungsanleitung	
--	---------------------	--

7. Nach Drücken der START-Taste ist der Drucker arbeitsbereit. Der Hochleistungsdrucker kann Daten empfangen und drucken.

Während der Arbeit des Druckers sollte hin und wieder die Druckqualität überprüft und der Druck gegebenenfalls nachgestellt werden. Dazu werden die unter Punkt 6. angeführten Arbeiten erneut durchgeführt.

Abgeschaltet wird der Drucker durch Ausrasten der START-Taste und anschließendes Drücken der Taste EIN/AUS



	Programmierung	
--	----------------	--

5. Programmierung

Die Ausgabe der Zeichen, die vom Druckwerk des NIXDORF-Druckers auf das Formular gedruckt werden sollen, erfolgt nicht in einzelnen, durch einzelne Ausgabebefehle bedingten Schritten, sondern, ähnlich wie bei der Lochkartenausgabe bei der Serie 820, über einen besonderen Bereich, hier über den Druckbereich.

Der Druckbereich kann bei normaler Schrift 132 Zeichen, bei verlängerter Schrift 66 Zeichen aufnehmen.

Der Druckbereich ist ein Schiebe-Register innerhalb der Elektronik des NIXDORF-Hochleistungsdruckers, es liegt also nicht in der Zentraleinheit des Computers. Daher braucht im Kernspeicher kein Speicherplatz für den Druckbereich des Hochleistungsdruckers freigehalten werden.

Alle Zeichen, die in einer Druckzeile hintereinander ausgedruckt werden sollen (max. 132 Zeichen bei normaler Schrift), werden einschließlich der dazwischenliegenden Leerschritte zunächst in den Druckbereich übertragen.

Von dort werden sie, bei Auslösung des Druckvorgangs, in einem Zuge auf das Formular ausgegeben.

Nach Beendigung des Druckvorgangs ist der Druckbereich automatisch gelöscht und zur Aufnahme weiterer Zeichen bereit. Der Vorgang des Druckens läuft simultan zum Anwenderprogramm des Computers ab. Neue Übertragungen in den Druckbereich stoppen jedoch den Programmablauf bis zur Fertigmeldung des Druckers.

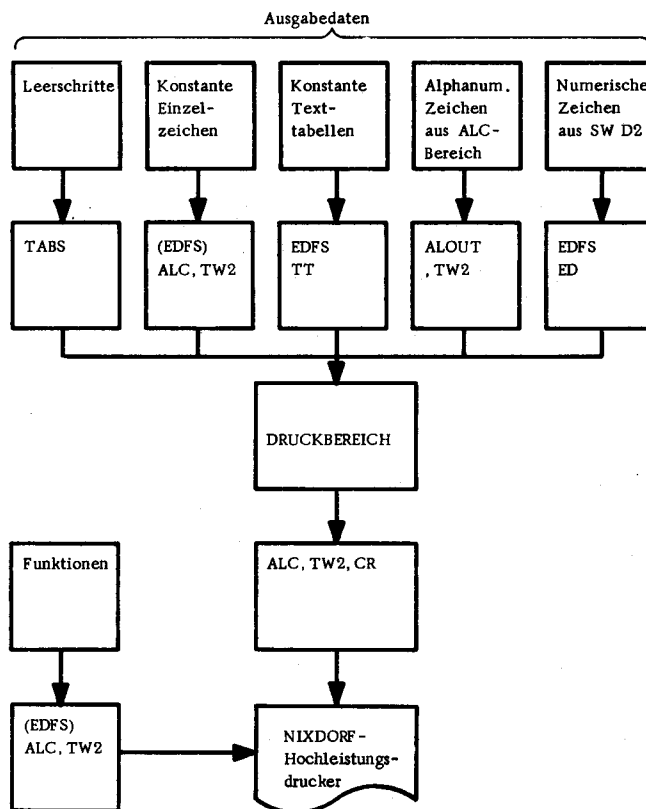
Der Druckbereich wird auch zu Beginn der Arbeit beim Betätigen des START-Schalters gelöscht und kann außerdem durch einen bestimmten Funktionscode (s.5.3.7.) gelöscht werden.

Da immer nur geschlossene Druckzeilen komplett ausgegeben werden können, muß den mit auszugebenden Leerschritten besondere Aufmerksamkeit geschenkt werden. Die nicht zu bedruckenden Stellen innerhalb einer Zeile müssen vom Programmierer mit Leerschritten aufgefüllt werden. Die Befehle für den Hochleistungsdrucker wurde so geschaffen, daß das Einfügen von Blanks für den Programmierer weitgehend erleichtert wird.

	Programmierung	
--	----------------	--

5.1. Übertragen von Zeichen

Die Übertragung der Zeichen kann auf verschiedene Arten erfolgen. Die folgende Schemazeichnung gibt eine Übersicht.



	Programmierung	
--	----------------	--

5.2. USASCII-Code-Tabelle

Die Elektronik des Hochleistungsdruckers verarbeitet die Zeichen im USASCII-Code, einem 7-Bit-Code.

Daher werden alle Zeichen beim Übertragen in den Druckbereich anhand einer Code-Tabelle vom ALC-Code in den USASCII-Code umcodiert. Diese Code-Tabelle belegt 109 Adressen im Anwenderprogrammblock 0. Ihre Anfangsadresse ist im ADImr-Teil des Befehls unter Adresse 0.0.1 im Anwenderprogrammblock 0 anzugeben.

	Programmierung	
--	----------------	--

USASCII-Code-Tabelle

ALC-Code	Zeichen	AD lmr	ALC-Code	Zeichen	AD lmr
0. 0	0	0. 3. 0	3. 0	:	0. 3.10
0. 1	1	0. 3. 1	3. 1	!	0. 2. 1
0. 2	2	0. 3. 2	3. 2	?	0. 3.15
0. 3	3	0. 3. 3	3. 3	"	0. 2. 2
0. 4	4	0. 3. 4	3. 4	=	0. 3.13
0. 5	5	0. 3. 5	3. 5	-	
0. 6	6	0. 3. 6	3. 6	(0. 2. 8
0. 7	7	0. 3. 7	3. 7)	0. 2. 9
0. 8	8	0. 3. 8	3. 8	\$	0. 2. 4
0. 9	9	0. 3. 9	3. 9	&	0. 2. 6
0.10		0. 2. 0	3.10	%	
0.11	-	0. 2.13	3.11	%	0. 2. 5
0.12	+	0. 2.11	3.12	%	
0.13	-	0. 2.13	3.13	'	0. 2. 7
0.14	◇	0. 3. 2	3.14	/	0. 2.15
0.15	◇	0. 3. 4	3.15		
1. 0	*	0. 2.10	4. 0	,	0. 2.12
1. 1	*	0. 5. 11	4. 1	\$	
1. 2	A	0. 4. 1	4. 2	@	0. 4. 0
1. 3	B	0. 4. 2	.		
1. 4	C	0. 4. 3	.		
1. 5	D	0. 4. 4	.		
1. 6	E	0. 4. 5	.		
1. 7	F	0. 4. 6	5.14	YDCM	0. 2.12
1. 8	G	0. 4. 7	5.15	YDPNT	0. 2.14
1. 9	H	0. 4. 8			
1.10	I	0. 4. 9	6. 0	CR	0. 0.13
1.11	J	0. 4.10	6. 1	TF	0. 0.12
1.12	K	0. 4.11	6. 2	VF	0. 0.11
1.13	L	0. 4.12	6. 3	LN	0. 0.10
1.14	M	0. 4.13	6. 4	BE	0. 0. 7
1.15	N	0. 4.14	6. 5	EL	0. 0.14
			6. 6	DE	0. 7.15
2. 0	O	0. 4.15	.		
2. 1	P	0. 5. 0	.		
2. 2	Q	0. 5. 1	6.12		
2. 3	R	0. 5. 2			
2. 4	S	0. 5. 3			
2. 5	T	0. 5. 4			
2. 6	U	0. 5. 5			
2. 7	V	0. 5. 6			
2. 8	W	0. 5. 7			
2. 9	X	0. 5. 8			
2.10	Y	0. 5. 9			
2.11	Z	0. 5.10			
2.12	#	0. 2. 3			
2.13	.	0. 2.14			
2.14	,	0. 2.12			
2.15	;	0. 3.11			

Die USASCII-Codes, die den ALC-Codes 6.0 bis 6.6 entsprechen, sind unter Punkt 5.3 näher erläutert.

Zeichen, denen in dieser Tabelle noch kein USASCII-Code zugeordnet wurde, können gegenwärtig auf dem Hochleistungsdrucker nicht ausgegeben werden. Es empfiehlt sich, diese Plätze, sowie die Lücken ab ALC-Code 4.3 mit dem USASCII-Code für Leerschritt (0.2.0.) zu belegen.

v 6 15 00

	Programmierung	
--	----------------	--

5.3. Beschreibung der Funktionscodes

In der Tabelle unter 5.2. sind ab ALC-Code 6.0 insgesamt 7 Funktionscodes aufgeführt.

Alle diese Funktions-Codes können über den Befehl ALC, TW2 gegeben werden. Sie werden nicht in den Druckbereich übertragen, sondern lösen direkt eine Funktion aus.

5.3.1. Druckauslösung (CR = Carriage return) ALC-Code 6.0

Die Druckauslösung kann auf zwei unterschiedliche Arten erfolgen:

- Der Druck kann ausgelöst werden, indem nach dem vollständigen Übertragen der Zeile im Anwenderprogramm der Befehl ALC, TW2, CR (2.15 2.6.0) gegeben wird. Der Code "CR" bewirkt ein Drucken der Zeile, einen Rücklauf des Druckschlittens und eine Zeilenschaltung. Wird "CR" bei gelöschtem Druckbereich gegeben, so erfolgen undefinierte Ausgaben.
- Wurden 132 Zeichen in den Druckbereich übertragen und der Code "CR" nicht gegeben, so wird die Zeile automatisch ausgedruckt und eine Zeilenschaltung durchgeführt.

In beiden Fällen ist nach Druckende der Druckbereich gelöscht.

5.3.2. Formularanfang (TF=Top of form) ALC-Code 6.1

Dieser Funktionscode bewirkt einen Formularvorschub bis zur nächsten festgelegten Vertikalposition. Das Formular wird solange vorgeschoben, bis in Kanal 5 des Vorschub-Lochstreifens vom Streifenleser eine Lochung erkannt wird.

5.3.3. Vertikalpositionierung (VF = Vertical format) ALC-Code 6.2

Dieser Funktionscode bewirkt einen Zeilenvorschub bis zur nächsten Vertikalposition. Das Formular wird solange vorgeschoben, bis in Kanal 5 des Vorschublochbandes vom Streifenleser eine Lochung erkannt wird.

5.3.4. Zeilenschaltung (LN=Line feed) ALC-Code 6.3

Der Funktionscode LN bewirkt eine Zeilenschaltung um eine Zeile.

	Programmierung	
--	----------------	--

5.3.5. Programmierter Alarm (BE = Bell) ALC-Code 6.4

Dieser Funktionscode veranlaßt einen etwa 2 Sekunden andauernden Ton über einen eingebauten Summer.

5.3.6. Verlängerte Schrift (EL = Elongated characters) ALC-Code 6.5

Dieser Funktionscode, vor oder während des Übertragens von Zeichen in den Druckbereich gegeben, bewirkt, daß die gesamte Zeile in verlängerter Schrift gedruckt wird. Die Zeichen werden, in der Waagerechten auf die doppelte Breite gedehnt. Dabei können in einer Zeile nur 66 Zeichen ausgegeben werden.

5.3.7. Druckbereich löschen (DE = Delete) ALC-Code 6.6

Bei Angabe dieses Funktionscodes wird der Druckbereich gelöscht.

5.4. Befehlsliste

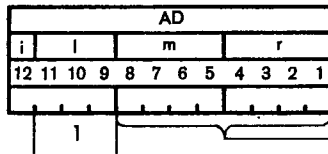
Für die Programmierung des NIXDORF-Hochleistungsdruckers wurden einige Befehle, die im Befehlsvorrat der Serie 820 schon vorhanden sind und bisher zum Ansprechen des 2. Serialdruckers benutzt wurden, in ihrer Wirkungsweise mehr oder weniger verändert.

Die folgenden Seiten beschreiben diese Befehle und bringen Anwendungsbeispiele.

OP	Symbol	Funktion	Merker		
			ML	MU	MC
2.13	TABS	Übertragen von Leerschritten in den Druckbereich AD ₁ = 1 : Hochleistungsdrucker AD _{mr} : Anzahl der zu übertragenden Leerschritte			
2.15	ALC	Übertragen eines Zeichens in den Druckbereich AD ₁ = 2 : Hochleistungsdrucker Bit 1 bis 7: ALC-Code des Zeichens			
	,TW2	Drucksteuerung durch Funktionszeichen AD ₁ = 2 : Hochleistungsdrucker			
	,CR	AD _{mr} = 6.0 : Druckauslösung, Wagenrücklauf, Zeilenschaltung			
	,TF	AD _{mr} = 6.1 : Vorschub auf Formularanfang			
	,VF	AD _{mr} = 6.2 : Vorschub auf nächste Vertikalposition			
	,LN	AD _{mr} = 6.3 : Eine Zeilenschaltung			
	,BE	AD _{mr} = 6.4 : Zwei Sekunden Summton			
	,EL	AD _{mr} = 6.5 : Verlängerte Schrift			
	,DE	AD _{mr} = 6.6 : Druckbereich löschen			
3.0	TT	Übertragen einer Texttabelle in den Druckbereich (bis Endezeichen 3.15) AD _{1mr} : Anfangsadresse der Texttabelle			
		Merker werden durch SMWT (2.12) gesetzt	ML	MU	MC
		8 YTAB	0	1	0
		12 YCAR	0	0	1
	11, sonstiges YTRK	1	0	0	
3.1	ALOUT	Übertragen alphanumerischer Zeichen aus dem ALC-Bereich in den Druckbereich			
	,TW2	AD ₁ = 6 : Hochleistungsdrucker Bit 1 - 7 : Anzahl der zu übertragenden Zeichen einschließlich etwaiger auffüllender Leerschritte (ab I ₃)			
	ALOUT 6	AD _m = + 0 : Zeichen im 6-Bit-Code gespeichert			
	ALOUT 8	AD _m = + 8 : Zeichen im 8-Bit-Code gespeichert			

OP	Symbol	Funktion	Merker		
			ML	MU	MC
		Beendigung Bit 1 - 4 des Endezeichens			
		Ende durch Endezeichen, 8	0	1	0
		Ende durch Endezeichen, 12	0	0	1
		Ende durch Endezeichen, 11, sonstiges	1	0	0
		Ende durch Erreichen der Anzahl n	0	0	0
3.2	ED	Übertragen numerischer Inhalte aus Druckspeicherwort D2 in den Druckbereich AD ₁ : Anzahl n der zu übertragenden Nachkommastellen AD _{mr} : Anzahl v der zu übertragenden Zeichen vor dem Komma, einschließlich etwaiger Leerschritte, Vornullen oder Sicherungsterne			
3.6	EDFS ,ZERO ,STAR ,MIN ,SGN ,ITS ,FTS	AD _m = 2 : Hochleistungsdrucker Vorbefehl für ALC (2.15) AD _r : Anzahl der Wiederholungen (0 - 15) Vorbefehl für TT (3.0) AD _{1mr} = 0.2.0 : Hochleistungsdrucker Vorbefehl für ED (3.2) AD _m + 0 : Vornullen AD _m + 1 : Sicherungsterne AD _r = . : Anzahl der Vorkommastellen, die mit Vornullen bzw. Sicherungsternen aufzufüllen sind. AD ₁ = 0 : Übertragen ohne Kennzeichen hinter der Zahl AD ₁ = 1 : Leerschritt, wenn (D2) ≥ 0 Kennzeichen -, wenn (D2) < 0 AD ₁ = 3 : Kennzeichen +, wenn (D2) ≥ 0 Kennzeichen -, wenn (D2) < 0 AD ₁ = 5 : Kennzeichen ◊, wenn (D2) ≥ 0 Kennzeichen ◊, wenn (D2) ≥ 0 AD ₁ = 7 : Kennzeichen ✱, wenn (D2) ≥ 0 Kennzeichen ✱, wenn (D2) < 0			

TABS	Übertragen von Leerschritten in den Druckbereich	2.13
------	--	------



Anzahl der Leerschritte (0.0 - 8.4)

Wirkung:

Die im AD_{mr} angegebene Anzahl von Leerschritten wird in den Druckbereich übertragen.

Beispiel:

In der 95. Position einer Zeile soll das Zeichen "A" gedruckt werden und dann das Formular um eine Zeile weiterschaltet werden. Der Druckbereich ist gelöscht.

BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
	0	2 13	0 1 5 14		TABS, 94
	1	2 15	0 2 1 2		ALC, TW2, A
	2	2 15	0 2 6 0		ALC, TW2, CR

Bemerkungen:

1. Der Befehl "TABS Übertragen von Leerschritten in den Druckbereich (2.13)" wird simultan zum weiteren Befehlsablauf ausgeführt. Vor einem neuen Übertragen in den Druckbereich bzw. vor einer anderen Ausgabe auf dem Simultangerät 2 wartet intern das Betriebsprogramm auf die Beendigung des Übertragens.
2. Ein Druckvorbefehl ist vor dem Befehl TABS nicht erforderlich, da der Hochleistungsdrucker im Hauptbefehl angegeben wird (AD_l = 1). Ein etwa gegebener Druckvorbefehl wird durch diesen Hauptbefehl auch nicht gelöscht.

ALC	Drucksteuerung durch Funktionscodes	2.15
-----	-------------------------------------	------

AD											
i	l	m				r					
12	11	10	9	8	7	6	5	4	3	2	1

2

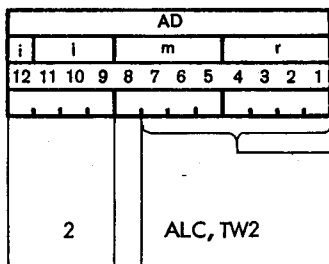
- 6.0 } ALC-Code 6.0 = Funktionscode CR
,CR } (Druckauslösung, Wagenrücklauf und Zeilenschaltung)
- 6.1 } ALC-Code 6.1 = Funktionscode TF
,TF } (Formularvorschub auf 1. Formularzeile)
- 6.2 } ALC-Code 6.2 = Funktionscode VF
,VF } (Formularvorschub auf nächste Vertikalposition)
- 6.3 } ALC-Code 6.3 = Funktionscode LN
,LN } (Eine Zeilenschaltung)
- 6.4 } ALC-Code 6.4 = Funktionscode BE
,BE } (Programmierter Alarm: 2 Sekunden Summton)
- 6.5 } ALC-Code 6.5 = Funktionscode EL
,EL } (Verlängerte Schrift)
- 6.6 } ALC-Code 6.6 = Funktionscode DE
,DE } (Löschen des Druckbereichs)

ALC, TW2

Wirkung:

Alle Funktionscodes werden in Verbindung mit dem Befehl ALC, TW 2 gegeben. Sie werden nicht in den Druckbereich übertragen, sondern lösen direkt eine Funktion aus.

ALC	Übertragen eines Zeichens in den Druckbereich	2.15
-----	---	------



ALC-Code des zu übertragenden Zeichens (0.0 - 6.12)

Übertragen eines Zeichens in den Druckbereich

Wirkung:

1. Das Zeichen, dessen ALC-Code in AD_{mr} Bit 1 bis Bit 7 angegeben ist, wird in den Druckbereich übertragen.
2. Beim Übertragen erfolgt eine Umcodierung anhand einer USASCII-Code-Tabelle. Diese Tabelle belegt 109 Adressen im Anwenderprogrammblock 0. Ihre Anfangsadresse ist im AD_{mr}-Teil des Befehls unter Adresse 0.0.1 im Anwenderprogrammblock 0 anzugeben.
3. Dem Übertragungsbefehl kann ein Druckvorbefehl EDFs (3.6) vorangehen. Durch diesen ist es möglich, eine bis zu 15-malige Wiederholung des Übertragens des Zeichens zu erreichen. Nach dem Übertragen ist der Vorbefehl gelöscht.

Beispiel:

Das Zeichen "-" (Bindestrich, Minus) soll 10-mal hintereinander in den Druckbereich übertragen werden.

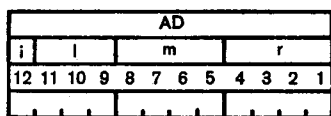
BW-Adresse	Op. Teil	i	Adreßteil	Symb. Adr.	Bemerkungen
	0 3 6 0	0 0 2 9			EDFS,9
	1 2 15 0	2 0 13			ALC, TW2, YMIN

ALC	Übertragen eines Zeichens in den Druckbereich	2.15
-----	---	------

Bemerkungen:

1. Ist $AD_1 = 0$, so erfolgt kein Übertragen. Der Befehl wirkt dann wie ein Leerbefehl.
2. Es dürfen nur Zeichen mit einem ALC-Code von 0.0 bis 6.12 codiert werden. Andernfalls können undefinierte Übertragungen erfolgen.
3. Da nicht alle Zeichen des ALC-Codes im USASCII-Code vorkommen, dagegen zusätzliche Sonder-Codes vorhanden sind, beachte man den Vorschlag für eine USASCII-Code-Tabelle unter Punkt 5.2.

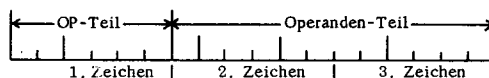
TT	Übertragen einer Texttabelle in den Druckbereich	3.0
----	--	-----



Anfangsadresse der Texttabelle
(0.0.0 - 7.15.15)

Wirkung:

- Der Befehl unter der in AD_{lmr} angegebenen Adresse ist ein 18-Bit-Wort. Werden von links nach rechts je 6 Bit zusammengefaßt, so entstehen drei 6-Bit-Worte, denen nach dem ALC-Code drei Zeichen entsprechen:



- Von der in AD_{lmr} angegebenen Adresse an werden nach dem unter Punkt 1 beschriebenen Verfahren die Zeichen fortlaufend von links nach rechts aufgesucht und in den Druckbereich übertragen. Nach dem Übertragen der drei Zeichen einer Adresse erfolgt sinngemäß das Übertragen der Zeichen im Befehlsword unter der nächsten Adresse. Dieser Vorgang wird solange fortgesetzt, bis das Endezeichen 3.15 auftritt. Dieses Zeichen wird nicht mehr übertragen, das Übertragen in den Druckbereich ist beendet.
- Das auf das Zeichen 3.15 folgende Zeichen wird ebenfalls aufgesucht. In Abhängigkeit von seinen letzten 4 Bit werden die folgenden Merkerstellungen intern zwischengespeichert:

		ML	MU	MC
8	YTAB	0	1	0
12	YCAR	0	0	1
11, sonstiges	YTRK	1	0	0

Im Befehl "SMWT Setze Merker nach Ende Simultan-Ausgabe (2.12)" wird auf das Ende des Übertragens gewartet. Im Anschluß daran erfolgt das Setzen der Merker entsprechend obiger Tabelle.

TT	Übertragen einer Texttabelle in den Druckbereich	3.0
----	--	-----

- Beim Übertragen erfolgt eine Umcodierung anhand einer USASCII-Code-Tabelle. Diese Tabelle belegt 109 Adressen im Anwenderprogrammblock 0. Ihre Anfangsadresse ist im AD_{Imr}-Teil des Befehls unter Adresse 0.0.1 im Anwenderprogrammblock 0 anzugeben.
- Dem Übertragungsbefehl muß ein Druckvorbefehl EDFs (3.6) vorangehen. Dieser ist nach dem Übertragen gelöscht.
- Nach Beendigung der Übertragung steht im Indexregister I₃ die Anzahl der durch den Befehl TT (3.0) in den Druckbereich übertragenden Zeichen. Da hier Simultarbeit stattfindet, beachte man die Bemerkung 3.
- Das Programm fährt mit dem auf den Befehl TT folgenden Befehl fort. (Eine Ausnahme hiervon bilden nur die Fälle, in denen der Befehl 3.0 durch Substitution entstanden ist. Man vergleiche hierzu die Befehle "SST Substitution (1.13)" und "SSTX Substitution mit Indexregistern (2.2)".)

Beispiel:

Es soll der Text KONTO: in den Druckbereich übertragen werden. Nach dem Endezeichen soll das Zeichen 0.11 gespeichert sein.

BW-Adresse			Op. Teil			I			Adreßteil			Symb. Adr.	Bemerkungen
4	2	0	3	6	0	0	2	0					EDFS
		1	3	0	0	7	12	0					TT, TEXT

Die Tabelle steht ab Adresse 7.12.0 im gleichen Block:

BW-Adresse			Op. Teil			I			Adreßteil			Symb. Adr.	Bemerkungen
7	12	0	1	12	1	0	1	15	TEXT				*KONTO/YCOL/&
		1	2	5	1	0	3	0					/YECC//YTRK/*
		2	3	15	0	2	12	0					

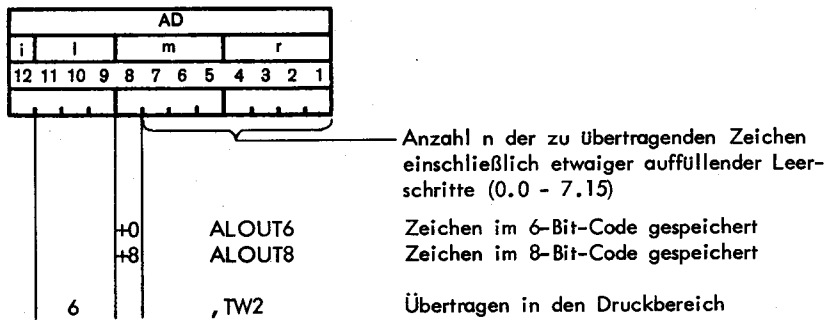
Es werden 6 Zeichen (KONTO:) in den Druckbereich übertragen. Danach steht im Indexregister I₃ die Zahl 6. Die Merkerstellung ML = 1, MU = 0, MC = 0 ist intern zwischengespeichert. Die Merker selbst sind jedoch unverändert. Das Programm fährt mit dem Befehl unter Adresse 4.2.13 fort.

TT	Übertragen einer Texttabelle in den Druckbereich	3.0
----	--	-----

Bemerkungen:

1. Man beachte, daß in jedem Fall nach dem Zeichen 3.15 die darauffolgende 6-Bit-Kombination untersucht wird, um die Merkerstellung dementsprechend zwischenspeichern. Es muß also immer eine solche Kombination vorhanden sein, d.h., das Endezeichen 3.15 sollte z.B. nicht in Bit 1 bis Bit 6 des letzten Befehls eines Blockes stehen.
2. Der Befehl "TT Übertragen einer Texttabelle in den Druckbereich (3.0)" wird simultan zum weiteren Befehlsablauf ausgeführt. Vor einer neuen Übertragung in den Druckbereich bzw. vor einer anderen Ausgabe auf dem Simultangerät 2 wartet intern das Betriebsprogramm auf die Beendigung des Übertragens.
3. Da bei diesem Übertragungsbefehl die Anzahl der in den Druckbereich übertragenen Zeichen nicht aus der Codierung hervorgeht, sondern, z.B. beim indizierten Aufsuchen von Texttabellen, unterschiedlich groß sein kann, wird vom Betriebsprogramm die Anzahl der übertragenen Zeichen im Indexregister I3 gespeichert. Der vorherige Inhalt von I3 ist grundsätzlich zerstört. Da der Zeitpunkt der Beendigung des simultanen Übertragens und somit des Index-Ladens unbestimmt ist, sollte der Zugriff zu I3 oder anderweitige Verwendung des I3 nicht vor Beendigung der Simultanarbeit erfolgen. Dies kann durch die Wartebefehle "WT, MTWS 2.12 0.8.1" oder "SMWT, MTWS 2.12 0.10.1" erreicht werden.
4. Man beachte den Befehl "BL Blockumschaltung (2.11)".

ALOUT	Übertragen alphanumerischer Inhalte aus dem ALC-Bereich in den Druckbereich	3.1
-------	---	-----



Wirkung:

1. Der Inhalt des ALC-Bereiches wird in den Druckbereich übertragen. Den Anfangspunkt im ALC-Bereich legt der ALC-Bereichszeiger (I3) fest.
2. Beim Übertragen erfolgt eine Umcodierung anhand einer USASCII-Code-Tabelle. Diese Tabelle belegt 109 Adressen im Anwenderprogrammblock 0. Ihre Anfangsadresse ist im AD_{lmr}-Teil des Befehls unter Adresse 0.0.1 im Anwenderprogrammblock 0 anzugeben.
3. Das Übertragen von Zeichen aus dem ALC-Bereich in den Druckbereich wird beendet, sobald n Zeichen übertragen wurden.
Tritt unter den ersten n Zeichen das Umschaltzeichen 3.15 oder eines der Endezeichen (Zeichen mit einem Code größer oder gleich 6.8) auf, so wird dieses Zeichen nicht übertragen. Es bewirkt jedoch, daß keine weiteren Zeichen dem ALC-Bereich entnommen werden, aber, daß die noch bis zur Anzahl n fehlenden Zeichen als Leerschritte übertragen werden.
4. War der Text im 6-Bit-Code gespeichert, so bedeutet das Zeichen 3.15 eine Umschaltung auf den 8-Bit-Code des Ende-Zeichens. Der Inhalt der beiden folgenden Speicherstellen wird demnach als Endezeichen interpretiert.
Bei der Speicherung im 8-Bit-Code entfällt im allgemeinen das Zeichen 3.15 vor dem Endezeichen, da eine Umschaltung nicht erforderlich ist. Tritt es trotzdem auf, so wird das folgende Zeichen als Endezeichen interpretiert.

ALOUT	Übertragen alphanumerischer Inhalte aus dem ALC-Bereich in den Druckbereich	3.1
-------	---	-----

Je nachdem, ob das Übertragen aus dem ALC-Bereich durch Auftreten eines Endezeichens oder durch Erreichen der Anzahl n beendet wurde, werden die folgenden Merkerstellungen intern zwischengespeichert:

Beendigung, Bit 1 bis 4 des Endezeichens	ML	MU	MC
Ende durch Endezeichen, 8	0	1	0
Ende durch Endezeichen, 12	0	0	1
Ende durch Endezeichen, 11, sonstiges	1	0	0
Ende durch Erreichen der Anzahl n	0	0	0

Im Befehl "SMWT Setze Merker nach Ende Simultan-Ausgabe (2.12)" wird auf das Ende des Übertragens gewartet. Im Anschluß daran erfolgt das Setzen der Merker entsprechend obiger Tabelle.

5. Der Bereichszeiger I₃ zählt während des Übertragens mit. Nach Beendigung durch ein Endezeichen zeigt er auf die dem Endezeichen folgende Speicherstelle, nach Beendigung durch Erreichen der Anzahl n zeigt er im Bereich auf die dem zuletzt übertragenen Zeichen folgende Speicherstelle.

Beispiel:

Aus dem ALC-Bereich soll der ab Speicherwort 22, Stelle 6, im 8-Bit-Code gespeicherte Text in den Druckbereich übertragen werden. Insgesamt sollen 20 Zeichen in den Druckbereich gelangen. Tritt ein Endezeichen bereits vor Erreichen dieser Anzahl auf, so ist die Differenz mit Leerschritten aufzufüllen.

Zunächst wird der Bereichszeiger I₃ des ALC-Bereiches geladen:

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
	0 3 11	0	1 6 6		BPX, 22., 6

nachher: I₃

1	11	6
---	----	---

 entspricht SW 22, Stelle 6

Es folgt der Übertragungsbefehl:

BW-Adresse	Op. Teil	I	Adreßteil	Symb. Adr.	Bemerkungen
	0 3 1 0	6	9 4		ALOUT8, TW2, 20

ALOUT	Übertragen alphanumerischer Inhalte aus dem ALC-Bereich in den Druckbereich	3.1
-------	---	-----

nachher: I_3

1	12	10
---	----	----

 entspricht SW 23, Stelle 10

SW 22

						2	1	1	2	1	5	1	6	2	3
--	--	--	--	--	--	---	---	---	---	---	---	---	---	---	---

SW 23

1	3	2	0	2	3	1	5	6	11						
---	---	---	---	---	---	---	---	---	----	--	--	--	--	--	--

Es werden insgesamt 20 Zeichen in den Druckbereich übertragen, nämlich

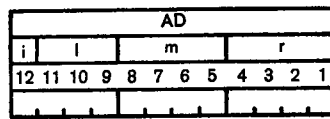
PADERBORN▲▲▲▲▲▲▲▲▲▲

Das Übertragen aus dem ALC-Bereich wurde durch das Auftreten des Endezeichens 6.11 beendet, deshalb ist die Merkerstellung $ML = 1$, $MU = 0$, $MC = 0$ intern zwischengespeichert.

Bemerkungen:

- Der Befehl "ALOUT Übertragen alphanumerischer Inhalte aus dem ALC-Bereich in den Druckbereich (3.1)" wird simultan zum weiteren Befehlsablauf ausgeführt. Vor einem neuen Übertragen in den Druckbereich bzw. vor einer anderen Ausgabe auf dem Simultangerät 2 wartet intern das Betriebsprogramm auf die Beendigung des Übertragens.
Man beachte jedoch, daß der angesprochene Teil des ALC-Bereichs und der Bereichszeiger I_3 erst nach Beendigung des Übertragens verändert werden dürfen (vgl. "Warte- und Merkersehbefehl (2.12)").
- Ist der Text im 6-Bit-Code gespeichert, so steht der ALC-Bereichszeiger nur bei jedem zweiten Zeichen richtig. Darauf ist besonders dann zu achten, wenn das Übertragen durch Erreichen der Anzahl n beendet wurde.
- Man beachte, daß der Bereichszeiger I_3 in I_{1m} immer die interne Speicherwortnummer enthält, die aus der normalen Speicherwortnummer durch Addition von 0.5 entsteht (vgl. Beispiel und den Befehl "BPX Laden ALC-Bereichszeiger (3.11)").

ED	Übertragen numerischer Inhalt aus D2 in den Druckbereich	3.2
----	---	-----



Anzahl v der zu übertragenden Zeichen vor dem Komma, einschließlich etwaiger Leerschritte, Vornullen oder Sicherungssterne (0.0 - 8.4)

Anzahl n der zu übertragenden Nachkommastellen (0 - 7)

Wirkung:

1. Der Inhalt des Druckspeicherwortes D2 wird in den Druckbereich übertragen. Der Inhalt des Druckspeicherwortes bleibt unverändert.
2. Dem Übertragungsbefehl muß ein Druckvorbefehl EDF5 (3.6) vorangehen. Das Übertragen erfolgt entsprechend den Anweisungen des Druckvorbefehls, der nach dem Übertragen gelöscht ist.
3. Die Anzahl v der zu übertragenden Zeichen vor dem Komma ist in AD_m anzugeben. Sind im SW D2 weniger als v Zeichen $\neq 0$ vor dem Komma enthalten, so wird vor diesen Zeichen bis zur Anzahl v mit Leerschritten aufgefüllt. Wurden im Druckvorbefehl jedoch Vornullen oder Sicherungssterne verlangt, so vermindern diese die Anzahl der auffüllenden Leerschritte.
4. Die Anzahl n der zu übertragenden Nachkommastellen ist in AD_1 anzugeben. Von $AD_1 = 0$ bis $AD_1 = 6$ wird die codierte Anzahl von Nachkommastellen, bei $AD_1 = 7$ die volle, durch die KA festgelegte Anzahl von Nachkommastellen übertragen.
5. Ist $AD_1 \neq 0$, so wird nach den Vorkommastellen ein Dezimalkomma bzw. Dezimalpunkt übertragen. Die Auswahl wird durch entsprechende Belegung des AD-Teils vom Befehl unter Adresse 0.0.0 im Anwenderprogrammblock 0 getroffen.
6. Ist der Inhalt des Druckspeicherwortes D2 = 0 und wurden im Vorbefehl keine Vornullen oder Sicherungssterne verlangt, so werden bei $AD_1 \neq 0$ nur dann Dezimalkomma bzw. -punkt und Nachkommastellen übertragen, wenn dies durch entsprechende Belegung des AD-Teils vom Befehl unter Adresse 0.0.0 im Anwenderprogrammblock 0 angeordnet wurde, andernfalls werden $v+n+1$ Leerschritte übertragen.

ED	Übertragen numerischer Inhalte aus D2 in den Druckbereich	3.2
----	---	-----

7. Beim Übertragen erfolgt eine Umcodierung anhand einer USASCII-Code-Tabelle. Diese Tabelle belegt 109 Adressen im Anwenderprogrammblock 0. Ihre Anfangsadresse ist im AD_{l_{mr}}-Teil des Befehls unter Adresse 0.0.1 im Anwenderprogrammblock 0 anzugeben.

Beispiel: (KA 5)

Der AD-Teil vom Befehl unter Adresse 0.0.0 im Anwenderprogrammblock 0 ist folgendermaßen belegt:

BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
0 0 0	0 0 1	0 2 5			

Der Inhalt des Druckspeicherwortes D2 ist mit 7 Vor- und 3 Nachkommastellen zusätzlich Zwischensummenzeichen in den Druckbereich zu übertragen. Stellengerecht soll dabei ein Dezimalpunkt übertragen werden. Bis zur 4. Vorkommastelle soll im Bedarfsfalle mit Vornullen aufgefüllt werden.

vorher: SW 2

+ 0 0 0 0 0 0 0 0 1 2 3 4 5 0 0 0

BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
	0 3 6 0	5 2 4			EDFS, ITS, ZERO, 4
	1 3 2 0	3 0 7			ED, 3., 7

nachher: SW 2

+ 0 0 0 0 0 0 0 0 1 2 3 4 5 0 0 0

Es werden insgesamt 12 Zeichen in den Druckbereich übertragen, nämlich

▲▲▲0123.450◇

Bemerkungen:

- Man beachte, daß im Fall (D2) = 0 nur AD_{l_{mr}} Leerschritte übertragen werden, wenn AD_l = 0 ist und kein Druckvorbefehl mit AD_r ≠ 0 gegeben wurde.
- Ist die in AD_l angegebene Anzahl von Nachkommastellen größer als die KA, so erfolgen undefinierte Übertragungen.

ED	Übertragen numerischer Inhalte aus D2 in den Druckbereich	3.2
----	---	-----

3. Der Befehl ED (3.2) wird simultan zum weiteren Befehlsablauf ausgeführt. Vor einem neuen Übertragen in den Druckbereich bzw. vor einer anderen Ausgabe auf dem Simultangerät 2 wartet intern das Betriebsprogramm auf die Beendigung des Übertragens, genauso vor einer Veränderung des Druckspeicherwortes D2. Dazu ist kein Wartebefehl erforderlich. Allerdings darf bei simultanem Ablauf des Druckbefehls die KA in I₅ nicht verändert werden.
4. Ist der Inhalt des Druckspeicherwortes D2 dezimal, so werden für die Zahlen 10 bis 15 die Zeichen übertragen, die den ALC-Codes 0.10 bis 0.15 entsprechen.

EDFS	Druckvorbefehl	3.6
------	----------------	-----

AD											
i	l		m				r				
12	11	10	9	8	7	6	5	4	3	2	1
						2					

EDFS Druckvorbefehl für Übertragen
in den Druckbereich des Hochleistungs-
druckers

Wirkung:

1. Der Druckvorbefehl EDFS enthält Anweisungen, die in den einzelnen Übertragungsbefehlen aus Platzgründen nicht angegeben werden können. Nach einem Übertragen in den Druckbereich ist der Vorbefehl EDFS aufgehoben, ein etwa für den 1. Drucker gegebener Druckvorbefehl EDF bleibt jedoch bestehen. Ferner setzt ein neuer Druckvorbefehl EDFS einen früheren Druckvorbefehl EDFS außer Kraft.
2. Die Wirkung des Druckvorbefehls EDFS auf die einzelnen Übertragungsbefehle ist unterschiedlich und hängt von der Art des folgenden Übertragungsbefehles ab:

EDFS vor "ALC, TW2 (2.15) Übertragen eines Zeichens in den Druckbereich":

Der Hochleistungsdrucker wird im Hauptbefehl angegeben ($AD_l = 2$). Ein Druckvorbefehl EDFS ist nur erforderlich, wenn das im Hauptbefehl codierte Zeichen mehrmals nacheinander in den Druckbereich übertragen werden soll. In diesem Falle ist die Anzahl der Wiederholungen im AD_r des EDFS anzugeben.

EDFS vor "TT (3.0) Übertragen einer Texttabelle in den Druckbereich":

Um eine Texttabelle in den Druckbereich zu übertragen, ist ein vorangehender Druckvorbefehl EDFS unbedingt erforderlich ($AD_{lmr} = 0.2.0$), da im Hauptbefehl das Gerät nicht angegeben ist.

EDFS vor "ALOUT6, TW2 bzw. ALOUT8, TW2 (3.1) Übertragen alphanumerischer Inhalte aus ALC-Bereich in den Druckbereich":

Vor diesem Übertragungsbefehl ist ein Druckvorbefehl EDFS nicht erforderlich, da der Hochleistungsdrucker im Hauptbefehl angegeben wird ($AD_l = 6$). Der Einheitlichkeit halber wird jedoch ein etwa gegebener Druckvorbefehl EDFS auch durch diesen Hauptbefehl, jedoch ohne weitere Wirkung, gelöscht.

EDFS	Druckvorbefehl	3.6
------	----------------	-----

AD											
i	l	m						r			
12	11	10	9	8	7	6	5	4	3	2	1
						2					
						+0					
						+1					
0											
1											
3											
5											
7											

EDFS Druckvorbefehl
vor "ED (3.2) Übertragen numerischer
Inhalte aus D2 in den Druckbereich"

,ZERO Vornullen
,STAR Sicherungssterne

Anzahl der Vorkommastellen, die mit
Vornullen oder Sicherungssternen aufzu-
füllen sind

Übertragen ohne Kennzeichen hinter der
Zahl

,MIN Leerschritt, wenn $(D2) \geq 0$
Kennzeichen-, wenn $(D2) < 0$

,SGN Kennzeichen+, wenn $(D2) \geq 0$
Kennzeichen-, wenn $(D2) < 0$

,ITS Kennzeichen \diamond , wenn $(D2) \geq 0$ *
Kennzeichen \oslash , wenn $(D2) < 0$ *

,FTS Kennzeichen*, wenn $(D2) \geq 0$
Kennzeichen \ddagger , wenn $(D2) < 0$ *

EDFS vor "ED (3.2) Übertragen numerischer Inhalte aus D2 in den Druckbereich:

Vor diesem Hauptbefehl ist ein vorangehender Druckvorbefehl EDFS unbedingt erforder-
lich, da im Hauptbefehl das Gerät nicht angegeben ist.

Für diesen Hauptbefehl gelten alle Informationen des Druckvorbefehls bezüglich des
Übertragens von Vorkommastellen, der Sicherungszeichen und der Kennzeichen
hinter der Zahl gemäß obiger Übersicht.

* Diese Zeichen können zur Zeit vom Hochleistungsdrucker noch nicht ausgegeben
werden. Es empfiehlt sich, die Symbole ITS und FTS zu vermeiden und anstelle der
Zeichen \diamond , \oslash und \ddagger den Code für Leerschritt in die Codetabelle einzusetzen.

NIXDORF
COMPUTER

Der NIXDORF-Hochleistungsdrucker 20817
für das Datenverarbeitungssystem 820
in Verbindung mit dem MSKZ1/HD-KEYBOL

Ausgabe vom 1.6.1972

NIXDORF
COMPUTER

Der NIXDORF-Hochleistungsdrucker 20817
für das Datenverarbeitungssystem 820

	Inhaltsverzeichnis	
--	--------------------	--

Inhaltsverzeichnis

1. Einleitung
2. Der technische Aufbau
 - 2.1. Der Drucker
 - 2.2. Das Druckwerk
 - 2.3. Der Zeichenvorrat
 - 2.4. Die Zeilenlänge
 - 2.5. Druck- und Rücklaufgeschwindigkeit
 - 2.6. Das Farbband
 - 2.7. Durchschlagsfähigkeit
 - 2.8. Formulartransport
 - 2.9. Akustischer Alarm
 - 2.10. Bedienungselemente
 - 2.10.1. EIN-AUS-Schalter
 - 2.10.2. Grundstellung Formular
 - 2.10.3. START-Schalter
 - 2.10.4. Formular-Ende-Unterdrückung
 - 2.10.5. Die obere Leuchtanzeige
 - 2.10.6. HDWE ALARM
 - 2.10.7. Papier-Ende
 - 2.11. Manuelle Bedienungselemente
 - 2.11.1. Formular-Einzugshebel
 - 2.11.2. Formular-Walzenfreilauf
 - 2.11.3. Formular-Stärke-Einstellung
3. Anlegen des Vorschublochstreifens
4. Bedienungsanleitung
5. Programmierung
 - 5.1. Übertragen von Zeichen
 - 5.2. USASCII-Code-Tabelle
 - 5.3. Beschreibung der Funktionscodes
 - 5.3.1. Formularanfang
 - 5.3.2. Vertikalposition
 - 5.3.3. Zeilenvorschub
 - 5.3.4. Druckauslösung
 - 5.3.5. Verlängerte Schrift
 - 5.4. Befehlsliste / Liste der Befehle / Beschreibung der Befehle

NIXDORF
COMPUTER

Datenverarbeitungssystem 820

Blatt HW 3

Hochleistungsdrucker

1.9. 1971

	Einleitung	
--	------------	--

1. Einleitung

Der NIXDORF-Hochleistungsdrucker ist ein Zweitdrucker. Durch seine hohe Ausgabegeschwindigkeit wird er zur idealen Ergänzung von Anlagen, mit denen Daten aus Datenträgern wie Lochkarte, Lochstreifen, Magnetkonto, Magnetbandcassette, Magnetplatte usw. eingelesen werden und zeitsparend auf Endlosformulare ausgegeben werden sollen.

Der NIXDORF-Hochleistungsdrucker kann an die Modelle 820/25, 820/35 und 880/65 angeschlossen werden.

	Der technische Aufbau	
--	-----------------------	--

2. Der technische Aufbau

Die folgenden Punkte geben eine Übersicht über den technischen Aufbau des Druckers sowie eine kurze Bedienungsanleitung.

2.1 Der Drucker

Der NIXDORF-Hochleistungsdrucker ist ein Serial-Matrix-Drucker (Nadel-Drucker) mit einer Schreibbreite von max. 130 Zeichen. Dabei handelt es sich um eine geschlossene Einheit, die verschiedene Baugruppen enthält.

Mechanische Baugruppen

z.B.
Druckwerk,
Farbband-Transport,
Formular-Transport,
akustischer Alarm,
Bedienungselemente
usw.

Elektronische Baugruppen

z.B.
Stromversorgung,
Kontroll-Logik,
Zeichen-Muster-Generator,
Einzel-Zeilen-Ausgabebereich
(Druckbereich),
usw.

2.2 Das Druckwerk

Das Druckwerk besteht aus einem Druckkopf und 7 Drucksolenoids mit daran angebrachten Drucknadeln. Alle diese Teile sind auf einem beweglichen Druckschlitten montiert, der sie horizontal vor der Druckwalze hin und her transportieren kann.

Die 7 Drucknadeln werden durch den Druckkopf in einer Senkrechten übereinander ausgerichtet. Werden die Solenoids von der Steuer-Elektronik aktiviert, so schießen die betreffenden Drucknadeln hervor und bilden über ein vor dem Papier vorbeigeführten Farbband punktförmige Anschläge auf dem Papier ab. Die Solenoids werden, je nach Zeichen, bis zu 5 mal pro Zeichen aktiviert. Dabei bewegt sich der Druckschlitten von links nach rechts.

Ein einheitlicher Zeichenabstand wird durch eine Synchronisation des Druckschlittens mit einem optischen Leser erreicht. Dieser Leser tastet während des Druckvorgangs einen Positionstreifen ab. Durch die Lage des Positionstreifens sind auch die seitlichen Begrenzungen für das 1. bzw. 130. Zeichen festgelegt.

	Der technische Aufbau	
--	-----------------------	--

Bei etwaiger Fehlfunktion (Überschreiten der Begrenzungen durch den Druckschlitten) wird die Anlage über einen Sicherheitsschalter automatisch abgeschaltet, ein akustisches Alarmzeichen gegeben und die Hardware-Anzeige zum Aufleuchten gebracht. Nach dem Beheben des Fehlers kann die Arbeit wieder aufgenommen werden. Bei wiederholten Störungen ist ein Kundendienstmechaniker hinzuzuziehen.

2.3. Der Zeichenvorrat

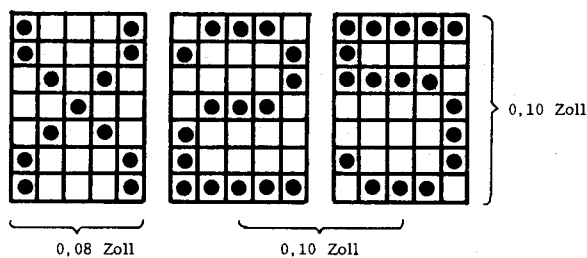
Der Zeichenvorrat des NIXDORF-Hochleistungsdruckers besteht aus 60 verschiedenen Zeichen im USASCII-Code.

- 10 Ziffern (0-9)
- 26 Großbuchstaben (A-Z)
- 20 Sonderzeichen

Die Liste der Zeichen, sowie ihre Zuordnung zum ALC-Code und zur Code-Tabelle im Anwenderprogramm ist der Tabelle unter 5.2. zu entnehmen.

Jedes Zeichen wird aus einzelnen Punkten zu einem Raster (Matrix) zusammengesetzt. Der Raster besteht bei normaler Schrift aus max. 5 x 7 Punkten.

Beispiel:

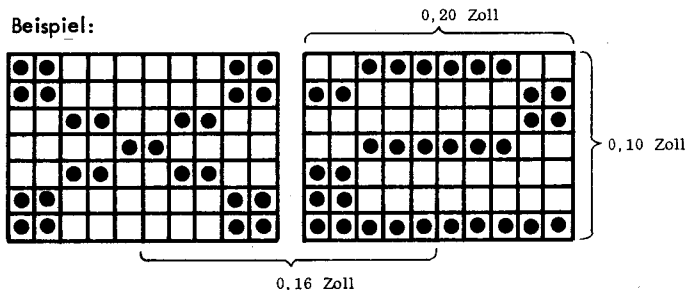


	Der technische Aufbau	
--	-----------------------	--

Der horizontale Abstand der Zeichen beträgt zwischen zwei Zeichenmitten 0,10 Zoll. Das entspricht der üblichen Teilung von 10 Zeichen je Zoll.

Der vertikale Abstand der Zeichen beträgt, bedingt durch die Höhe einer Zeilenschaltung, zwischen zwei Zeichenmitten 0,166 Zoll, was der üblichen Teilung von 6 Zeilen pro Zoll entspricht.

Der Hochleistungsdrucker bietet die Möglichkeit, bestimmte Texte durch verlängerte Schrift hervorzuheben. In diesem Falle wird die Zeichenbreite verdoppelt. Das geschieht, indem jede senkrechte Nadelanschlag-Kombination doppelt ausgeführt wird, wodurch sich das Raster auf maximal 10 x 7 Punkte je Zeichen erweitert.



2.4. Die Zeilenlänge

Der Hochleistungsdrucker kann in einer Zeile bei normaler Schrift 130 Zeichen, bei verlängerter Schrift 65 Zeichen ausgeben.

Die Anzahl der auszugebenden Zeichen wird vom Anwenderprogramm bestimmt. Sie kann innerhalb der oben angegebenen Grenzen beliebig sein. Verkürzte Zeilen bringen Zeitersparnis.

	Der technische Aufbau	
--	-----------------------	--

2.5. Druck- und Rücklaufgeschwindigkeit

Die Druckgeschwindigkeit (Lauf des Druckschlittens von links nach rechts) ist konstant und beträgt etwa 41 cm/sec., was einer Ausgabe von etwa 165 Zeichen/sec. entspricht.

Die Rücklaufgeschwindigkeit des Druckschlittens (von rechts nach links) beträgt etwa das 4-fache der Druckgeschwindigkeit.

Aus diesen Geschwindigkeiten und der Zeilenlänge ergibt sich beispielsweise für den Druck einer vollen DIN-A-4 Zeile einschließlich Druckwerksrücklauf die Zeit von ca. 0,6 Sekunden (ca. 6000 Zeilen/Stunde).

2.6. Das Farbband

Der NIXDORF-Hochleistungsdrucker verwendet ein spezielles schwarzes Farbband von 1 Zoll Breite. Es wird leicht schräg vor dem Papier vorbeigeführt. Dadurch ergibt sich ein hoher Ausnutzungsgrad und eine lange Lebensdauer des Farbbandes.

2.7. Durchschlagsfähigkeit

Trotz der hohen Horizontalgeschwindigkeit des Druckschlittens wird ein Nadel-Anschlag erzielt, der für 5 Nutzen ausreicht.

2.8. Formulartransport

Der NIXDORF-Hochleistungsdrucker bedruckt Endlosformulare mit der üblichen Transportlochung von 2 Löchern je Zoll.

Er transportiert das Papier mit einer vertikalen Formulartransport-Einrichtung, die aus zwei Vorschubraupen (Formulartraktor) besteht. Der Abstand der beiden Vorschubraupen ist stufenlos verstellbar, so daß verschieden breite Formulare von mindestens 4 Zoll (10,8 cm) bis höchstens 14 1/2 Zoll (36,54 cm) Breite eingespannt werden können.

Die Höhe einer Zeile ist 1/6 Zoll. Die Vorschubgeschwindigkeit beträgt 24 Zeilen je Sekunde.

	Der technische Aufbau	
--	-----------------------	--

Eine reibungslos ablaufende Papierablage (Stapelung) hängt weitgehend von der Beschaffenheit der Formulare und der Umgebung ab. Die günstigsten Bedingungen liegen in folgenden Bereichen:

Raumtemperatur: 10° bis 38° C
Relative Luftfeuchtigkeit: 25% bis 62%.

Trotzdem können gelegentliche Eingriffe der Bedienungskraft zur Ordnung der Formularstapelung erforderlich sein, insbesondere, wenn die oben genannten Grenzwerte in der einen oder anderen Richtung überschritten werden.

Es gibt 4 unterschiedliche Arten des Papiervorschubs:

- Automatische Schaltung einer Zeile im Anschluß an die Ausgabe einer Druckzeile
- Zeilenschaltung ohne Druckausgabe (nur wenn keine Datenzeichen im Druckpuffer stehen)
- Vorschub des Papiers auf die erste Zeile des folgenden Formulars
- Vorschub des Papiers zur nächsten festgelegten Vertikalposition des gleichen Formulars

Um die beiden letztgenannten Vorschubarten zu ermöglichen, wird ein Vorschublochband eingesetzt. Dabei handelt es sich um einen Standard-Lochstreifen, bei dem Kanal 7 dem Vorschub zum Anfang des nächsten Formulars und Kanal 5 dem Vorschub zur nächsten Vertikalposition zugeordnet sind. Das Anfertigen und Einsetzen des Vorschublochbandes wird unter Punkt 3. beschrieben.

2.9. Akustischer Alarm

Beim Hochleistungsdrucker werden Fehler durch ein akustisches Zeichen (2 Sekunden Summton) gemeldet.

Dieser Alarm kann auf 3 verschiedene Arten ausgelöst werden.

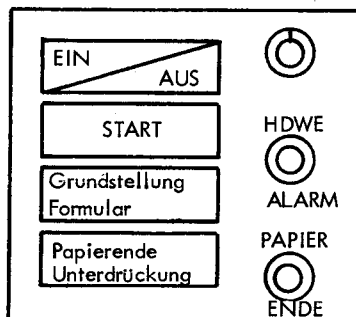
	Der technische Aufbau	
--	-----------------------	--

- Papier-Ende oder Fehlzuführung
In diesem Falle wird der Summton von der Hardware ausgelöst.
- Überschreitung der seitlichen Begrenzungen
Der Summton wird von der Hardware ausgelöst und macht auf eine Überschreitung der seitlichen Begrenzungen durch den Druckschlitten aufmerksam. Ferner wird das dadurch hervorgerufene Ausschalten der Anlage angezeigt.

In jedem Fall muß der Fehler von der Bedienungskraft beseitigt werden.

2.10. Bedienungselemente

Am Hochleistungsdrucker befindet sich eine Bedienungs-Schalttafel, die 4 Schalter und 3 Leuchtanzeigen enthält.



2.10.1. EIN-AUS-Schalter

Der Schalter leuchtet, wenn der Drucker eingeschaltet ist.

	Der technische Aufbau	
--	-----------------------	--

2.10.2. Grundstellung Formular

Dieser Schalter dient dem manuellen Vorschub zum nächsten Formularanfang oder der genauen Ausrichtung des Vorschublochbandes beim Einlegen neuer Formulare.

2.10.3. START-Schalter

Nach Drücken dieses Schalters ist der Drucker zur Aufnahme und Ausgabe von Daten bereit. Bei Druck des START-Schalters wird der Druckbereich gelöscht.

2.10.4. Papierende-Unterdrückung

Mit diesem Schalter kann die Bedienungskraft die Papierende-Meldung überwählen, so daß begonnene Formulare noch zu Ende verarbeitet werden können.

2.10.5. Die obere Leuchtanzeige ist für spezielle Zwecke bisher freigehalten.

2.10.6. HDWE ALARM. Diese Leuchtanzeige zeigt an, daß der Druckschlitten den linken oder rechten Begrenzungsschalter überschritten hat.

2.10.7. PAPIER ENDE. Die Leuchtanzeige leuchtet bei Papier-Ende oder Papier-Fehlzuführung auf.

2.11. Manuelle Bedienungselemente

Zusätzlich zu den Schaltern der Bedienungsschalttafel hat der Hochleistungsdrucker folgende Elemente zur manuellen Bedienung:

2.11.1. Formular-Einzugs-Hebel

Ein Hebel an der rechten Seite der Maschinenverkleidung gestattet eine schnelle Zuführung neuer Formulare um die Walze und in die Papiervorschubraupen.

Blatt HW 12

Datenverarbeitungssystem 820

1.9. 1971

Hochleistungsdrucker

	Der technische Aufbau	
--	-----------------------	--

2.11.2. Formular-Walzenfreilauf

Der Knopf an der linken Seite des Gehäuses dient zur manuellen Vor- und Rückpositionierung des Formulars.

2.11.3. Formular-Stärke-Einstellung

Am Druckwerk befindet sich eine Skala mit Bezugspunkten. Je nach Stärke des Formulars bzw. Formularsatzes kann der Abstand zwischen Walze und Druckkopf eingestellt werden.

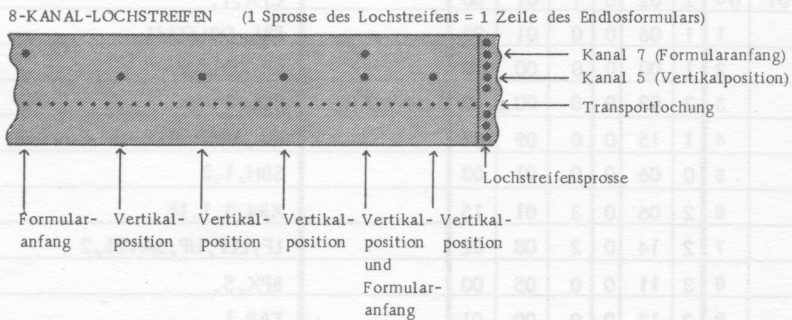
Anlegen des Vorschublochstreifens

3. Anlegen des Vorschublochstreifens

Der Vorschublochstreifen, der zur Vertikalpositionierung des Endlosformulars benutzt wird, ist ein schwarzer, lichtundurchlässiger 8-Kanal-Lochstreifen mit der üblichen Transportlochung zwischen dem 3. und 4. Kanal. Jede Sprosse des Lochstreifens entspricht einer Zeile des zu bedruckenden Formulars.

Der Lochstreifenleser des Hochleistungsdruckers liest nur die Kanäle 7 und 5. Eine Lochung in Kanal 7 bedeutet Anfang des zu bedruckenden Formulars. Auf Druck der Taste Grundstellung Formular oder eine entsprechende Angabe im Anwenderprogramm wird der Vorschublochstreifen und das Papier soweit vorgeschoben, bis eine Lochung in Kanal 7 erkannt wird.

Eine Lochung in Kanal 5 bedeutet eine Vertikalpositionierung. Auf eine entsprechende Angabe im Anwenderprogramm wird der Vorschublochstreifen solange weitergeschoben, bis eine Lochung in Kanal 5 erkannt wird.



Dieser Vorschublochstreifen kann mit dem auf den folgenden Codierformularen codierten Programm auf einer Anlage 820/35 selbst hergestellt werden.

NIXDORF SERIE 820**MASCHINEN-SPRACHE**

Programm: Programm zur Erstellung von Vorschublochstreifen									Blocknr.	Blatt	1
Programmierer									Firma		Datum
BW-Adresse			Op. Teil			Adreßteil			Symb. Adr.	Bemerkungen	
0	00	00	0	00	0	0	00	00		KGA 0	
		1									
		2									
		3	0	00	0	0	02	10		Anfangsadresse LS-Tabelle	
		4									
		5	2	14	0	2	08	05	ANFANG	LF, LEP, UP, SAVPL, 5	
		6	2	12	0	0	02	01	EINGAB	WT, MRAR	
		7	0	01	0	0	01	00		ACC, 1.	
		8	2	02	0	0	00	01		XF, 1	
		9	2	06	0	3	01	15		XR4, 3.1.15	
		10	2	10	0	0	09	03		CX, 147	
		11	2	02	0	1	00	01		CPX, 1	
		12	1	06	0	0	00	15		BRL, ABFRA2	
		13	2	12	0	0	00	00	ROLAMP	WTC	
		14	1	00	0	0	00	06		BR, EINGAB	
		15	2	10	0	0	15	15	ABFRA2	CX, 255	
0	01	00	2	02	0	1	01	00		CPX, 1.	
		1	1	06	0	0	01	03		BRL, DRUZH	
		2	1	00	0	0	00	13		BR, ROLAMP	
		3	3	02	0	0	00	10	DRUZH	ED, 10	
		4	1	15	0	0	09	04		CA, 148	
		5	0	06	0	0	01	03		SBH, 1.3	
		6	2	06	0	3	01	15		XR4, 3.1.15	
		7	2	14	0	2	08	02		LF, LEP, UP, SAVPL, 2	
		8	3	11	0	0	05	00		BPX, 5.	
		9	2	13	0	0	00	01		TAB, 1	
		10	3	13	0	0	08	00		ALIN6, 128	
		11	2	14	0	2	08	01		LF, LEP, UP, SAVPL, 1	
		12	2	13	0	0	00	01		TAB, 1	
		13	3	13	1	0	01	06		ALIN6, XI, 22	
		14	3	11	0	0	05	00		BPX, 5.	
		15	3	13	0	4	07	15		ALPT, CD6, 127	

Programm: Programm zur Erstellung von Vorschublochstreifen									Blocknr.	Blatt 2	
Programmierer									Firma		Datum
BW-Adresse			Op. Teil			Adreßteil			Symb. Adr.	Bemerkungen	
0	2	00	2	12	0	0	08	01		WT, MPT	
			1	3	11	0	1	01	02	BPX, 1, 1, 2	
			2	3	13	1	4	01	05	ALPT, CD6, XI, 21	
			3	2	10	0	3	14	06	CX, 3, 14, 6	
			4	2	15	0	4	00	00	ENDLOS ALC, PT, 0	
			5	2	10	1	0	00	01	CX, XI, 1	
			6	1	09	0	0	02	08	BXG, ROTELA	
			7	1	00	0	0	02	04	BR, ENDLOS	
			8	2	12	0	0	00	00	ROTELA WTC	
			9	1	00	0	0	00	05	BR, ANFANG	
			10	0	00	1	1	08	00	LSTAB Lochstreifencode-Tabelle (0 = nur Trans- portloch.)	
			11	0	00	1	1	13	00	(1 = Formularanfang + Vertikalposition)	
			12	0	00	1	1	04	00	(2 = Formularanfang)	
			13	0	00	1	1	01	00	(3 = Vertikalposition)	
			14								
			15								
			0								
			1								
			2								
			3								
			4								
			5								
			6								
			7								
			8								
			9								
			10								
			11								
			12								
			13								
			14								
			15								

	Anlegen des Vorschublochstreifens
--	-----------------------------------

Nach Eingabe des Programms in den Kernspeicher kann über die Zehner-tastatur und Taste → eine Zahl zwischen 148 und 254 (Anzahl der zu stanzenden Sprossen des Lochstreifens) eingegeben werden, die angibt, für wieviele Zeilen der Vorschublochstreifen ausgelegt werden soll. Dabei entspricht eine Sprosse des Lochstreifens einer Zeile des Formulars. Nach dem Ausdruck der eingegebenen Zahl erfolgt eine Freigabe der alpha-numerischen Tastatur. Über diese wird nun für jede Zeile des zu bedruk-kenden Formulars eine Ziffer eingegeben.

- 0 = Keine Lochung (nur Transportlochung)
- 1 = Formularanfang und Vertikalpositionierung
- 2 = Formularanfang
- 3 = Vertikalpositionierung

Andere Eingaben führen zu Stanzfehlern.

Die Schreibmaschinenfreigabe wird nach der ersten eingegebenen Zeile und bei Erreichen der Anzahl mit der WZ-Taste beendet.

Anschließend wird über den angeschlossenen Lochstreifenstanzer der Loch-streifen ausgegeben. Es muß unbedingt darauf geachtet werden, daß schwar-zer, lichtundurchlässiger Lochstreifen verwandt wird, da sonst Fehler beim Lesen auftreten.

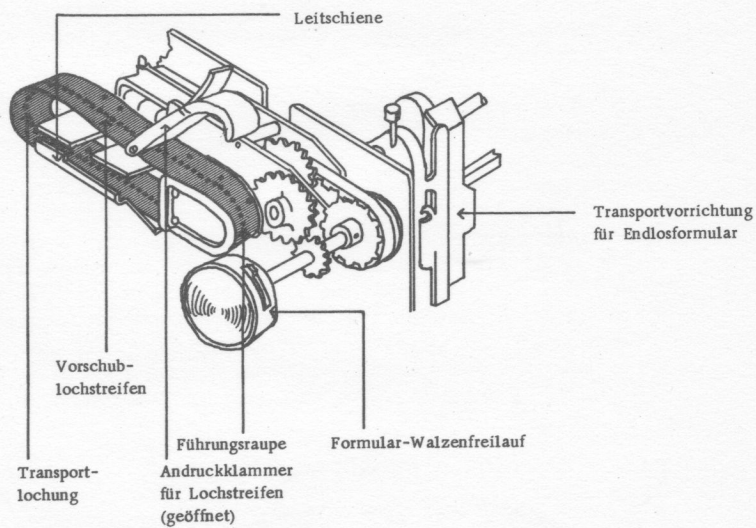
Dieser Lochstreifen wird $2\frac{1}{2}$ Transportlöcher lang überlappend zusammen-geklebt. Dabei muß darauf geachtet werden, daß der Streifen mindestens 15 Zoll (148 Transportlöcher) lang ist und daß die beiden Schnittkanten mindestens $2\frac{1}{2}$ Transportlöcher von einer Informationslochung entfernt sind.



	Anlegen des Vorschublochstreifens	
--	-----------------------------------	--

Der auf diese Weise entstandene Endloslochstreifen wird wie folgt eingelegt.

- obere linke Verkleidung der Anlage abnehmen.
- Klammer nach oben klappen
- Lochstreifen mit der Transportlochung in die Führungsraupe und die Leitschiene einlegen
- Klammer schließen
- Lochstreifen durch Druck der Taste Grundstellung Formular bis zur Formularanfangslochung vorlaufen lassen.
- Verkleidung schließen.



	Bedienungsanleitung	
--	---------------------	--

4. Bedienungsanleitung

Voraussetzung zur Inbetriebnahme des Hochleistungsdruckers als Zweitdrucker an die Modelle 820/25, 820/35 und 880/55 sind:

- Bestückung der Anlage mit dem Betriebsprogramm MSKZ1/HD-KEYBOL anstelle des MSKZ1. Dieses Betriebsprogramm ist mit dem MSKZ2 bzw. MSKZ2/LU oder mit dem SKZA2 zu kombinieren.
- Anschluß des Hochleistungsdruckers an die E/A-Einheit 333 (Chassisplatz 2. Serialdrucker) über ein Kabel Nr. 452. Die Stromversorgung erfolgt über ein eigenes Netzkabel.

Sind diese Voraussetzungen erfüllt, und ist der Computer eingeschaltet, so wird der Hochleistungsdrucker wie folgt in Betrieb genommen.

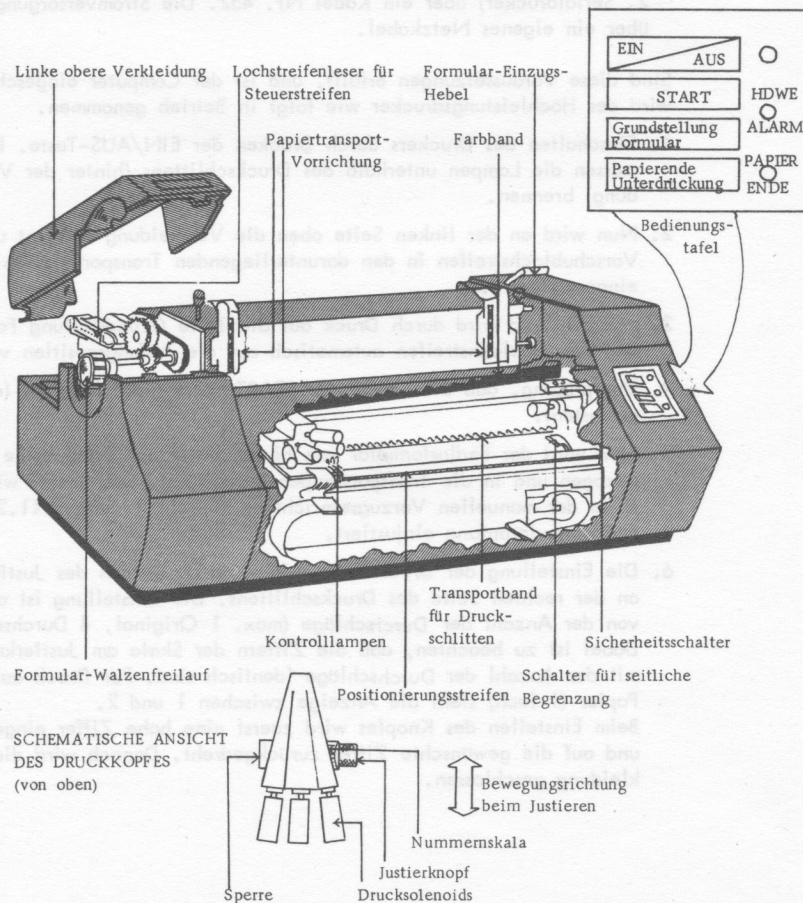
1. Einschalten des Druckers durch Drücken der EIN/AUS-Taste. Danach müssen die Lampen unterhalb des Druckschlittens (hinter der Verkleidung) brennen.
2. Nun wird an der linken Seite oben die Verkleidung entfernt und der Vorschublochstreifen in den darunterliegenden Transportmechanismus eingesetzt (s.3.).
3. Anschließend wird durch Druck auf die Taste Grundstellung Formular der Vorschublochstreifen automatisch auf die Anfangsposition vorgezogen.
4. Überprüfung, daß die Lampe der START-Taste nicht leuchtet (evtl. austrasten).
5. Nun wird das Endlosformular von hinten unter der Druckwalze durchgezogen und in die Transportraupen eingehängt. Das Papier wird mit Hilfe der manuellen Vorzugseinrichtung (s.2.11.1. und 2.11.2.) auf den Formulanfang einjustiert.
6. Die Einstellung der Druckstärke erfolgt durch Drehen des Justierknopfes an der rechten Seite des Druckschlittens. Die Einstellung ist abhängig von der Anzahl der Durchschläge (max. 1 Original, 4 Durchschläge). Dabei ist zu beachten, daß die Ziffern der Skala am Justierknopf nicht mit der Anzahl der Durchschläge identisch sind. Für Druck auf normales Papier (1-fach) steht die Anzeige zwischen 1 und 2. Beim Einstellen des Knopfes wird zuerst eine hohe Ziffer eingestellt und auf die gewünschte Ziffer zurückgedreht. Danach wird die Verkleidung geschlossen.

	Bedienungsanleitung	
--	---------------------	--

7. Nach Drücken der START-Taste ist der Drucker arbeitsbereit. Der Hochleistungsdrucker kann Daten empfangen und drucken.

Während der Arbeit des Druckers sollte hin und wieder die Druckqualität überprüft und der Druck gegebenenfalls nachgestellt werden. Dazu werden die unter Punkt 6. angeführten Arbeiten erneut durchgeführt.

Abgeschaltet wird der Drucker durch Ausrasten der START-Taste und anschließendes Drücken der Taste EIN/AUS



	Programmierung	
--	----------------	--

5. Programmierung

Die Ausgabe der Zeichen, die vom Druckwerk des NIXDORF-HD-Druckers auf das Formular gedruckt werden sollen, erfolgt nicht in einzelnen, durch einzelne Ausgabebefehle bedingten Schritten, sondern, ähnlich wie bei der Lochkartenausgabe bei der Serie 820, über einen besonderen Bereich, hier über den Druckbereich.

Der Druckbereich kann bei normaler Schrift 130 Zeichen, bei verlängerter Schrift 65 Zeichen aufnehmen.

Der Druckbereich ist ein Schiebe-Register innerhalb der Elektronik des NIXDORF-Hochleistungsdruckers, es liegt also nicht in der Zentraleinheit des Computers. Daher braucht im Kernspeicher kein Speicherraum für den Druckbereich des Hochleistungsdruckers freigehalten werden.

Alle Zeichen, die in einer Druckzeile hintereinander ausgedruckt werden sollen (max. 130 Zeichen bei normaler Schrift), werden einschließlich der dazwischenliegenden Leerschritte zunächst in den Druckbereich übertragen.

Von dort werden sie, bei Auslösung des Druckvorgangs, in einem Zuge auf das Formular ausgegeben.

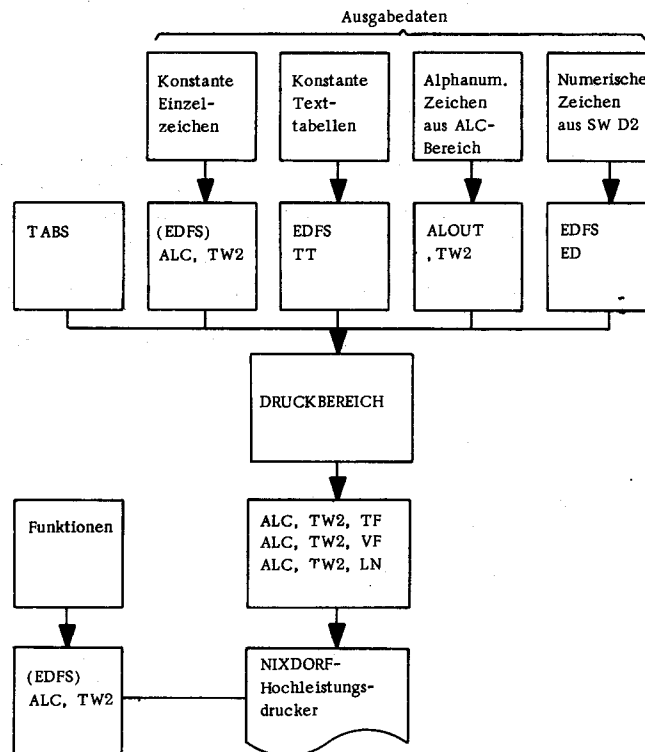
Nach Beendigung des Druckvorgangs ist der Druckbereich automatisch gelöscht und zur Aufnahme weiterer Zeichen bereit. Der Vorgang des Druckens läuft simultan zum Anwenderprogramm des Computers ab. Neue Übertragungen in den Druckbereich stoppen jedoch den Programmablauf bis zur Fertigmeldung des Druckers.

Der Druckbereich wird auch zu Beginn der Arbeit beim Betätigen des START-Schalters gelöscht.

	Programmierung	
--	----------------	--

5.1. Übertragen von Zeichen

Die Übertragung der Zeichen kann auf verschiedene Arten erfolgen. Die folgende Schemazeichnung gibt eine Übersicht.



	Programmierung	
--	----------------	--

5.2. USASCII-Code-Tabelle

Die Elektronik des Hochleistungsdruckers verarbeitet die Zeichen im USASCII-Code, einem 7-Bit-Code.

Daher werden alle Zeichen beim Übertragen in den Druckbereich anhand einer Code-Tabelle vom ALC-Code in den USASCII-Code umcodiert. Diese Code-Tabelle belegt 109 Adressen im Anwenderprogrammblock 0. Ihre Anfangsadresse ist im ADImr-Teil des Befehls unter Adresse 0.0.1 im Anwenderprogrammblock 0 anzugeben.

	Programmierung	
--	----------------	--

USASCII-Code-Tabelle

ALC-Code	Zeichen	AD lmr	ALC-Code	Zeichen	AD lmr
0. 0	0	0. 3. 0	3. 0	:	0. 3.10
0. 1	1	0. 3. 1	3. 1	!	0. 2. 1
0. 2	2	0. 3. 2	3. 2	?	0. 3.15
0. 3	3	0. 3. 3	3. 3	"	0. 2. 2
0. 4	4	0. 3. 4	3. 4	=	0. 3.13
0. 5	5	0. 3. 5	3. 5	-	
0. 6	6	0. 3. 6	3. 6	(0. 2. 8
0. 7	7	0. 3. 7	3. 7)	0. 2. 9
0. 8	8	0. 3. 8	3. 8	\$	0. 2. 4
0. 9	9	0. 3. 9	3. 9	&	0. 2. 6
0.10		0. 2. 0	3.10	9	
0.11	-	0. 2.13	3.11	%	0. 2. 5
0.12	+	0. 2.11	3.12	%	
0.13	-	0. 2.13	3.13	'	0. 2. 7
0.14	◇		3.14	/	0. 2.15
0.15	◇		3.15		
1. 0	*	0. 2.10	4. 0	,	0. 2.12
1. 1	#		4. 1	\$	
1. 2	A	0. 4. 1	4. 2	@	0. 4. 0
1. 3	B	0. 4. 2	.		
1. 4	C	0. 4. 3	.		
1. 5	D	0. 4. 4	.		
1. 6	E	0. 4. 5	.		
1. 7	F	0. 4. 6	5.14	YDCM	0. 2.12
1. 8	G	0. 4. 7	5.15	YDPNT	0. 2.14
1. 9	H	0. 4. 8			
1.10	I	0. 4. 9	6. 0	CR	0. 0.13
1.11	J	0. 4.10	6. 1	TF	0. 0.12
1.12	K	0. 4.11	6. 2	VF	0. 0.11
1.13	L	0. 4.12	6. 3	LN	0. 0.10
1.14	M	0. 4.13	6. 4	BE	0. 0. 7
1.15	N	0. 4.14	6. 5	EL	0. 0.14
2. 0	O	0. 4.15	6. 6	DE	0. 7.15
2. 1	P	0. 5. 0	.		
2. 2	Q	0. 5. 1	.		
2. 3	R	0. 5. 2	6.12		
2. 4	S	0. 5. 3			
2. 5	T	0. 5. 4			
2. 6	U	0. 5. 5			
2. 7	V	0. 5. 6			
2. 8	W	0. 5. 7			
2. 9	X	0. 5. 8			
2.10	Y	0. 5. 9			
2.11	Z	0. 5.10			
2.12	#	0. 2. 3			
2.13	.	0. 2.14			
2.14	,	0. 2.12			
2.15	;	0. 3.11			

Die USASCII-Codes, die den ALC-Codes 6.0 bis 6.6 entsprechen, sind unter Punkt 5.3 näher erläutert.

Zeichen, denen in dieser Tabelle noch kein USASCII-Code zugeordnet wurde, können gegenwärtig auf dem Hochleistungsdrucker nicht ausgegeben werden. Es empfiehlt sich, diese Plätze, sowie die Lücken ab ALC-Code 4.3 mit dem USASCII-Code für Leerschritt (0.2.0.) zu belegen.

	Programmierung	
--	----------------	--

5.3. Beschreibung der Funktionscodes

In der Tabelle unter 5.2. sind ab ALC-Code 6.0 insgesamt 4 Funktionscodes aufgeführt.

Alle diese Funktions-Codes können über den Befehl ALC, TW2 gegeben werden. Sie werden nicht in den Druckbereich übertragen, sondern lösen direkt eine Funktion aus.

5.3.1. Formularanfang (TF = Top of form) ALC-Code 6.1

Dieser Funktionscode bewirkt einen Formularvorschub bis zur nächsten festgelegten Vertikalposition. Das Formular wird solange vorgeschoben, bis in Kanal 5 des Vorschub-Lochstreifens vom Streifenleser eine Lochung erkannt wird.

5.3.2. Vertikalpositionierung (VF = Vertical format) ALC-Code 6.2

Dieser Funktionscode bewirkt einen Zeilenvorschub bis zur nächsten Vertikalposition. Das Formular wird solange vorgeschoben, bis in Kanal 5 des Vorschublochbandes vom Streifenleser eine Lochung erkannt wird.

5.3.3. Zeilenschaltung (LN = Line feed) ALC-Code 6.3

Der Funktionscode LN bewirkt eine Zeilenschaltung um eine Zeile.

5.3.4. Druckauslösung

Die Funktionscodes VF (Vertikalpositionierung), TF (Formularanfang) und LN (Zeilenschaltung) bewirken neben ihrer normalen Funktion automatisch eine Druckauslösung, wenn vorher im Druckpuffer Zeichen abgestellt wurden (Makrobefehle ED, TT, ALOUT, ALC). Ist der Druckpuffer leer oder nur mit Blanks aufgefüllt, bleibt der Druckschlitten in der Ausgangsposition. Nach einer Druckauslösung ist der Druckbereich gelöscht.

	Programmierung	
--	----------------	--

5.3.5. Verlängerte Schrift (EL = Elongated characters) ALC-Code 6.5

Dieser Funktionscode, vor oder während des Übertragens von Zeichen in den Druckbereich gegeben, bewirkt, daß die gesamte Zeile in verlängerter Schrift gedruckt wird. Die Zeichen werden, in der Waagerechten auf die doppelte Breite gedehnt. Dabei können in einer Zeile nur 65 Zeichen ausgegeben werden.

5.4. Befehlsliste

Für die Programmierung des NIXDORF-Hochleistungsdruckers wurden einige Befehle, die im Befehlsvorrat der Serie 820 schon vorhanden sind und bisher zum Ansprechen des 2. Serialdruckers benutzt wurden, in ihrer Wirkungsweise mehr oder weniger verändert.

Die folgenden Seiten beschreiben diese Befehle und bringen Anwendungsbeispiele.

Außer den in der Kurzliste aufgeführten Befehlen, deren Wirkungsweise verändert ist, sind bei den folgenden Befehlen Besonderheiten zu beachten.

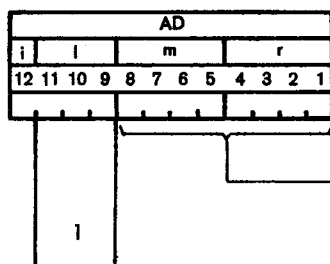
- POSX2 (2.2)
Der Befehl bewirkt, daß die Position des Druckzeigers nach 1 gebracht wird.
- TW (2.13)
Die Op-Code-Ergänzung ,TWS ist nicht zulässig.
- LF (2.14)
Die OP-Code-Ergänzungen ,LEPS,UP / ,LEPS,LOW sowie ,PLTS sind nicht zulässig.
- VP (3.4)
Die OP-Code-Ergänzungen ,LEPS,UP / ,LEPS,LOW sowie ,PLTS sind nicht zulässig.

OP	Symbol	Funktion	Merker		
2.13	TABS	Tabulation nach Position AD_{mr} $AD_1 = 1$: Hochleistungsdrucker AD_{mr} : Position			
2.15	ALC ,TW2	Übertragen eines Zeichens in den Druckbereich $AD_1 = 2$: Hochleistungsdrucker Bit 1 bis 7: ALC-Code des Zeichens			
	,TW2	Drucksteuerung durch Funktionszeichen $AD_1 = 2$: Hochleistungsdrucker			
	,TF	$AD_{mr} = 6.1$ } 1. Druckauslösung, wenn im Druckbereich Datenzeichen stehen 2. Vorschub auf Formularanfang			
	,VF	$AD_{mr} = 6.2$ } 1. Druckauslösung, wenn im Druckbereich Datenzeichen stehen 2. Vorschub auf nächste Vertikalposition			
	,LN	$AD_{mr} = 6.3$ } 1. Druckauslösung, wenn im Druckbereich Datenzeichen stehen 2. Eine Zeilenschaltung			
	,EL	$AD_{mr} = 6.5$: Verlängerte Schrift			
3.0	TT	Übertragen einer Tabelle in den Druckbereich (bis Endezeichen 3.15) AD_{1mr} : Anfangsadresse der Texttabelle			
		Merker werden durch SMWT (2.12) gesetzt	ML	MU	MC
		8 YTAB	0	1	0
		12 YCAR	0	0	1
		11, sonstiges YTRK	1	0	0
3.1	ALOUT	Übertragen alphanumerischer Zeichen aus dem ALC-Bereich in den Druckbereich			
	,TW2	$AD_1 = 6$: Hochleistungsdrucker Bit 1 - 7: max. Anzahl der auszudruckenden Zeichen			
	ALOUT 6	$AD_m = + 0$: Zeichen im 6-Bit-Code gespeichert			
	ALOUT 8	$AD_m = + 8$: Zeichen im 8-Bit-Code gespeichert			

Datenverarbeitungssystem 820
Liste der Befehle für den
Hochleistungsdrucker (nur mit MSKZ1/HD-Keybol)

OP	Symbol	Funktion	Merker		
			ML	MU	MC
		Beendigung Bit 1 - 4 des Endezeichens			
		Ende durch Endezeichen, 8	0	1	0
		Ende durch Endezeichen, 12	0	0	1
		Ende durch Endezeichen, 11, sonstiges	1	0	0
		Ende durch Erreichen der Anzahl n	0	0	0
3.2	ED	Übertragen numerischer Inhalte aus Druckspeicherwort D2 in den Druckbereich AD ₁ : Anzahl n der zu übertragenden Nachkommastellen AD _{mr} : Druckposition des Kommas			
3.6	EDFS	AD _m = 2: Hochleistungsdrucker			
		Vorbefehl für ALC (2.15) AD _r : Anzahl der Wiederholungen (0 - 15)			
		Vorbefehl für TT (3.0) AD _{1mr} = 0.2.0: Hochleistungsdrucker			
		Vorbefehl für ED (3.2)			
	,ZERO	AD _m + 0: Vornullen			
	,STAR	AD _m + 1: Sicherungssterne AD _r = : Anzahl der Vorkommastellen, die mit Vornullen bzw. Sicherungssternen aufzufüllen sind. AD ₁ = 0 : Übertragen ohne Kennzeichen hinter der Zahl			
	,MIN	AD ₁ = 1 : Leerschritt, wenn (D2) ≥ 0 Kennzeichen -, wenn (D2) < 0			
	,SGN	AD ₁ = 3 : Kennzeichen +, wenn (D2) ≥ 0 Kennzeichen -, wenn (D2) < 0			
	,ITS	AD ₁ = 5 : Kennzeichen ◊, wenn (D2) ≥ 0 Kennzeichen ◊, wenn (D2) < 0			
	,FTS	AD ₁ = 7 : Kennzeichen *, wenn (D2) ≥ 0 Kennzeichen *, wenn (D2) < 0			

TABS	Tabulation Hochleistungsdrucker	2.13
------	---------------------------------	------



Position 0.0 - 8.2

TABS Tabulation Hochleistungsdrucker

Wirkung:

Von der aktuellen Position des Druckbereichs-Zeigers bis zu der in AD_{mr} angegebenen Position -1 werden Leerschritte in den Druckbereich übertragen.

Beispiel: Tabulation Hochleistungsdrucker nach Position 67

BW-Adresse			Op. Teil		Adreßteil			Symb. Adr.	Bemerkungen
3	2	0	1	13	1	4	3	TABS, 67	

Der Druckbereich wird bis zur Position 66 mit \emptyset aufgefüllt. Der Druckbereichszeiger steht anschließend auf Pos. 67.

Bemerkungen:

1. Der Befehl TABS wird simultan zum weiteren Befehlsablauf ausgeführt. Vor einer neuen Ausgabe auf dem Hochleistungsdrucker wartet das Betriebssystem auf die Beendigung der Operation.
2. Position 0 ist die erste, Position 130 die letzte Schreibstelle.
3. Bei Ausgabe einer Position größer 8.2 (130) wird ein Internfehlerstop ausgelöst.
4. Eine Tabulation nach links ist nur nach einer Druckauslösung möglich.

ALC	Drucksteuerung durch Funktionscodes (Hochleistungsdrucker)	2.15
-----	---	------

AD											
i	l	m				r					
12	11	10	9	8	7	6	5	4	3	2	1

2

Hochleistungsdrucker

6.1, TF

- ALC-Code 6.1 = Funktionscode TF
1. Druckauslösung, wenn im Druckpuffer Datenzeichen abgestellt sind.
 2. Formularvorschub auf 1. Formularzeile.

6.2, VF

- ALC-Code 6.2 = Funktionscode VF
1. Druckauslösung, wenn im Druckpuffer Datenzeichen abgestellt sind.
 2. Formularvorschub auf die nächste Vertikalposition.

6.3, LN

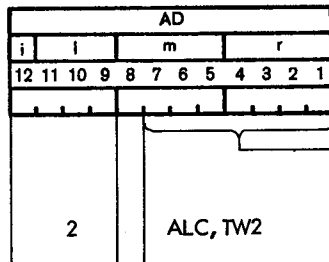
- ALC-Code 6.3 = Funktionscode LN
1. Druckauslösung, wenn im Druckpuffer Datenzeichen abgestellt sind.
 2. Eine Zeilenschaltung

6.5, EL

- ALC-Code 6.5 = Funktionscode EL
- Verlängerte Schrift, d.h. die Zeichen werden auf die doppelte Breite gedehnt
(dieser Code bewirkt keine Druckauslösung)

Die Funktionscodes werden nicht in den Druckbereich übertragen, sondern dienen nur zur Auslösung einer bestimmten Funktion.

ALC	Übertragen eines Zeichens in den Druckbereich	2.15
-----	---	------



ALC-Code des zu Übertragenden Zeichens (0.0 - 6.12)

Übertragen eines Zeichens in den Druckbereich

Wirkung:

1. Das Zeichen, dessen ALC-Code in AD_{mr} Bit 1 bis Bit 7 angegeben ist, wird in den Druckbereich übertragen.
2. Beim Übertragen erfolgt eine Umcodierung anhand einer USA \AA SCII-Code-Tabelle. Diese Tabelle belegt 109 Adressen im Anwenderprogrammblock 0. Ihre Anfangsadresse ist im AD_{lmr}-Teil des Befehls unter Adresse 0.0.1 im Anwenderprogrammblock 0 anzugeben.
3. Dem Übertragungsbefehl kann ein Druckvorbefehl EDFs (3.6) vorangehen. Durch diesen ist es möglich, eine bis zu 15-malige Wiederholung des Übertragens des Zeichens zu erreichen. Nach dem Übertragen ist der Vorbefehl gelöscht.

Beispiel:

Das Zeichen "-" (Bindestrich, Minus) soll 10-mal hintereinander in den Druckbereich übertragen werden.

BW-Adresse	Op. Teil	l	Adreßteil	Symb. Adr.	Bemerkungen
	0 3 6 0	0 2 9			EDFS, 9
	1 2 15 0	2 0 13			ALC, TW2, YMIN

Blatt HW/B 4

1.9. 1971

Datenverarbeitungssystem 820

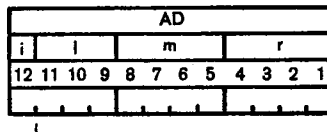
Hochleistungsdrucker, Beschreibung der Befehle

ALC	Übertragen eines Zeichens in den Druckbereich	2.15
-----	---	------

Bemerkungen:

1. Ist $AD_i = 0$, so erfolgt kein Übertragen. Der Befehl wirkt dann wie ein Leerbefehl.
2. Es dürfen nur Zeichen mit einem ALC-Code von 0.0 bis 6.12 codiert werden. Andernfalls können undefinierte Übertragungen erfolgen.
3. Da nicht alle Zeichen des ALC-Codes im USASCII-Code vorkommen, dagegen zusätzliche Sonder-Codes vorhanden sind, beachte man den Vorschlag für eine USASCII-Code-Tabelle unter Punkt 5.2.

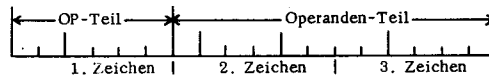
TT	Übertragen einer Texttabelle in den Druckbereich	3.0
----	--	-----



Anfangsadresse der Texttabelle
(0.0.0 - 7.15.15)

Wirkung:

- Der Befehl unter der in AD_{lmr} angegebenen Adresse ist ein 18-Bit-Wort. Werden von links nach rechts je 6 Bit zusammengefaßt, so entstehen drei 6-Bit-Worte, denen nach dem ALC-Code drei Zeichen entsprechen:



- Von der in AD_{lmr} angegebenen Adresse an werden nach dem unter Punkt 1 beschriebenen Verfahren die Zeichen fortlaufend von links nach rechts aufgesucht und in den Druckbereich übertragen. Nach dem Übertragen der drei Zeichen einer Adresse erfolgt sinngemäß das Übertragen der Zeichen im Befehlsword unter der nächsten Adresse. Dieser Vorgang wird solange fortgesetzt, bis das Endezeichen 3.15 auftritt. Dieses Zeichen wird nicht mehr übertragen, das Übertragen in den Druckbereich ist beendet.
- Das auf das Zeichen 3.15 folgende Zeichen wird ebenfalls aufgesucht. In Abhängigkeit von seinen letzten 4 Bit werden die folgenden Merkerstellungen intern zwischengespeichert:

		ML	MU	MC
8	YTAB	0	1	0
12	YCAR	0	0	1
11, sonstiges	YTRK	1	0	0

Im Befehl "SMWT Setze Merker nach Ende Simultan-Ausgabe (2.12)" wird auf das Ende des Übertragens gewartet. Im Anschluß daran erfolgt das Setzen der Merker entsprechend obiger Tabelle.

TT	Übertragen einer Texttabelle in den Druckbereich	3.0
----	--	-----

4. Beim Übertragen erfolgt eine Umcodierung anhand einer USASCII-Code-Tabelle. Diese Tabelle belegt 109 Adressen im Anwenderprogrammblock 0. Ihre Anfangsadresse ist im AD_lm_r-Teil des Befehls unter Adresse 0.0.1 im Anwenderprogrammblock 0 anzugeben.
5. Dem Übertragungsbefehl muß ein Druckvorbefehl EDFS (3.6) vorangehen. Dieser ist nach dem Übertragen gelöscht.
6. Das Programm fährt mit dem auf den Befehl TT folgenden Befehl fort. (Eine Ausnahme hiervon bilden nur die Fälle, in denen der Befehl 3.0 durch Substitution entstanden ist. Man vergleiche hierzu die Befehle "SST Substitution (1.13)" und "SSTX Substitution mit Indexregistern (2.2)".)

Beispiele:

Es soll der Text KONTO: in den Druckbereich übertragen werden. Nach dem Endezeichen soll das Zeichen 0.11 gespeichert sein.

BW-Adresse	Op.Teil	Adreßteil	Symb. Adr.	Bemerkungen
4 2 0	3 6 0 0 2 0			EDFS
	1 3 0 0 7 12 0			TT, TEXT

Die Tabelle steht ab Adresse 7.12.0 im gleichen Block:

BW-Adresse	Op.Teil	Adreßteil	Symb. Adr.	Bemerkungen
7 12 0	1 12 1 0 1 15		TEXT	*KONTO/YCOL/&
	1 2 5 1 0 3 0			/YECC//YTRK/*
	2 3 15 0 2 12 0			

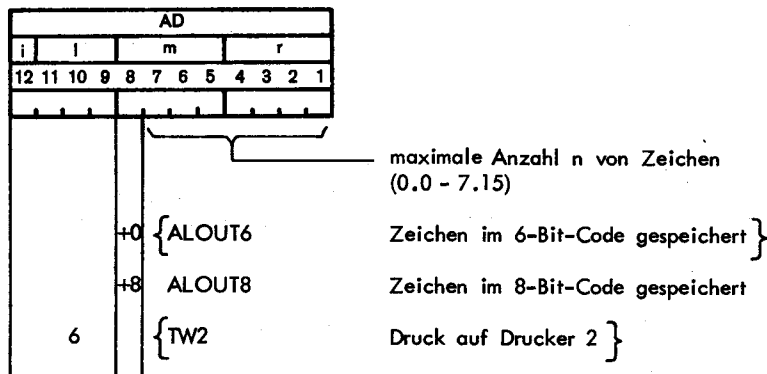
Es werden 6 Zeichen (KONTO:) in den Druckbereich übertragen. Die Merkerstellung ML = 1, MU = 0, MC = 0 ist intern zwischengespeichert. Die Merker selbst sind jedoch unverändert. Das Programm fährt mit dem Befehl unter Adresse 4.2.13 fort.

TT	Übertragen einer Texttabelle in den Druckbereich	3.0
----	--	-----

Bemerkungen:

1. Man beachte, daß in jedem Fall nach dem Zeichen 3.15 die darauffolgende 6-Bit-Kombination untersucht wird, um die Merkerstellung dementsprechend zwischenspeichern. Es muß also immer eine solche Kombination vorhanden sein, d.h., das Endezeichen 3.15 sollte z.B. nicht in Bit 1 bis Bit 6 des letzten Befehls eines Blockes stehen.
2. Der Befehl "TT Übertragen einer Texttabelle in den Druckbereich (3.0)" wird simultan zum weiteren Befehlsablauf ausgeführt. Vor einer neuen Übertragung in den Druckbereich bzw. von einer anderen Ausgabe auf dem Simultangerät 2 wartet intern das Betriebsprogramm auf die Beendigung des Übertragens.
3. Man beachte den Befehl "BL Blockumschaltung (2.11)".

ALOUT	Übertragen alphanumerischer Inhalte aus dem ALC-Bereich nach Druckbereich	3.1
-------	---	-----



Wirkung:

1. Der Inhalt des ALC-Bereichs wird fortlaufend ab der gegenwärtigen Schreibkopfposition in den Druckbereich übertragen. Den Anfangspunkt des ALC-Bereichs legt der Inhalt des Bereichszeigers in l_3 fest.
2. Beim Übertragen erfolgt eine Umcodierung anhand einer USASCII-Code-Tabelle. Diese Tabelle belegt 109 Adressen im Anwenderblock 0. Ihre Anfangsadresse ist im AD_{lmr} -Teil des Befehls unter Adresse 0.0.1 im Anwenderblock 0 anzugeben.
3. Das Übertragen wird beendet, wenn unter den ersten n Zeichen das Umschaltzeichen 3.15 oder eines der Endezeichen (Codes 6.8) auftritt. Kommt keines dieser Zeichen vor, so werden genau n Zeichen ausgegeben.

Im 6-Bit-Code bedeutet das Zeichen 3.15 eine Umschaltung auf den 8-Bit-Code des Endezeichens. Der Inhalt der beiden folgenden Kernspeicherstellen wird als Endezeichen interpretiert.

ALOUT	Übertragen alphanumerischer Inhalte aus dem ALC-Bereich nach Druckbereich	3.1
-------	---	-----

Bei der Speicherung im 8-Bit-Code entfällt im allgemeinen das Zeichen 3.15 vor dem Endezeichen, da eine Umschaltung nicht erforderlich ist. Tritt es trotzdem auf, so wird das folgende Zeichen als Endezeichen interpretiert.

Je nachdem, ob das Übertragen durch Auftreten eines Endezeichens oder durch Erreichen der Anzahl n beendet wurde, werden die folgenden Merkerstellungen intern zwischengespeichert.

Beendigung, Bit 1 bis 4 des Endezeichens	ML	MU	MC
Ende durch Endezeichen, 8	0	1	0
Ende durch Endezeichen, 12	0	0	1
Ende durch Endezeichen, 11, sonst	1	0	0
Ende durch Erreichen der Anzahl n	0	0	0

Im Befehl "SMWT Setze Merker nach Ende Simultan-Ausgabe (2.12)" wird auf das Ende des Übertragens gewartet. Im Anschluß daran erfolgt das Setzen der Merker entsprechend obiger Tabelle.

5. Der Bereichszeiger I_3 zählt während des Übertragens mit. Nach Beendigung durch ein Endezeichen zeigt er auf die dem Endezeichen folgende Kernspeicherstelle, nach Beendigung durch Erreichen der Anzahl n zeigt er im Bereich auf die dem zuletzt übertragenen Zeichen folgende Kernspeicherstelle.

Beispiel:

Aus dem ALC-Bereich soll der ab Speicherwort 22 Stelle 14 im 8-Bit-Code gespeicherte Text in den Druckbereich übertragen werden. Der Druckbereichszeiger vom Drucker 2 befindet sich in Position 76.

Zunächst wird der Bereichszeiger I_3 des ALC-Bereichs geladen.

BW-Adresse	Op. Teil	Adreßteil	Symb. Adr.	Bemerkungen
0	3	11 0 1 6 14		BPX, 22.14
1				

nachher: I_3

1	11	14
---	----	----

 entspricht SW 22 Stelle 14

ALOUT	Übertragen alphanumerischer Inhalte aus dem ALC-Bereich nach Druckbereich	3.1
-------	---	-----

Danach folgt der Druckbefehl:

vorher: I_3

1	11	14
---	----	----

BW-Adresse	Op. Teil	Adreßteil	Symb. Adr.	Bemerkungen
0	3 1 0	6 9 4		ALOUT8, TW2, 20
1				

nachher: I_3

1	12	11
---	----	----

SW 22

0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	4
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

SW 23

2	10	2	4	2	5	1	6	1	14	6	11	0	0	0	0
---	----	---	---	---	---	---	---	---	----	---	----	---	---	---	---

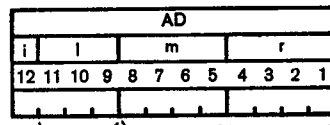
Es werden 6 Zeichen in den Druckbereich übertragen
nämlich
SYSTEM

Das Übertragen aus dem ALC-Bereich wurde durch das Endezeichen 6.11 beendet, deshalb ist die Merkerstellung ML = 1, MU = 0, MC = 0 intern zwischengespeichert.

Bemerkungen:

1. Der Befehl "ALOUT Übertragen Inhalt ALC-Bereich (3.1)" wird simultan zum weiteren Befehlsablauf ausgeführt. Vor einem neuen Übertragen in den Druckbereich wartet intern das Betriebsprogramm auf die Beendigung der Übertragung. Man beachte jedoch, daß der angesprochene Teil des ALC-Bereichs und der Bereichszeiger I_3 erst nach Beendigung des Übertragens verändert werden dürfen (vgl. "Warte- und Merkersetzbefehl (2.12)").
2. Ist der Code im 6-Bit-Code gespeichert, so steht der Bereichszeiger nur bei jedem zweiten Zeichen richtig. Darauf ist besonders dann zu achten, wenn das Übertragen durch Erreichen der Anzahl n beendet wurde.
3. Man beachte, daß der Bereichszeiger I_3 in I_m immer die interne Speicherwortnummer enthält, die aus der normalen Speicherwortnummer durch Addition von 0.5 entsteht (vgl. Beispiel und den Befehl "BPX Laden ALC-Bereichszeiger (3.11)").

ED	Übertragen numerischer Inhalte aus D2 in den Druckbereich	3.2
----	--	-----



Druckposition Dezimalkomma bzw. Dezimalpunkt
(0.0 - 8.2)
Anzahl n der zu übertragenden Nachkommastellen (0 - 7)

Wirkung:

1. Der Inhalt des Druckspeicherwortes D2 wird in den Druckbereich übertragen. Der Inhalt des Druckspeicherwortes bleibt unverändert.
2. Dem Übertragungsbefehl muß ein Druckvorbefehl EDFS (3.6) vorangehen. Das Übertragen erfolgt entsprechend den Anweisungen des Druckvorbefehls, der nach dem Übertragen gelöscht ist.
3. Die Druckposition des Kommas bzw. Punktes ist in AD_{mr} anzugeben. Bei Ausgabe von $AD_l = 0$ Nachkommastellen findet jedoch kein Druck von Komma oder Punkt statt. Durch entsprechende Belegung des AD-Teils vom Befehl unter Adr. 0.0.0 im Anwenderprogrammblock 0 ist festzulegen, ob bei Ausgabe von Nachkommastellen zuvor ein Dezimalkomma oder ein Dezimalpunkt in den Druckbereich übertragen werden soll.
4. Die Anzahl n der zu übertragenden Nachkommastellen ist in AD_l anzugeben. Von $AD_l = 0$ bis $AD_l = 6$ wird die codierte Anzahl von Nachkommastellen, bei $AD_l = 7$ die volle, durch die KA festgelegte Anzahl von Nachkommastellen übertragen.
5. Ist der Inhalt des Druckspeicherwortes $D2 = 0$ und gilt kein Druckvorbefehl, so werden bei $AD_l \neq 0$ nur dann Nachkommastellen übertragen, wenn die durch entsprechende Belegung des AD-Teils vom Befehl unter der Adresse 0.0.0 im Anwenderprogrammblock 0 angeordnet wurde.

ED	Druckbefehl (edition)	3.2
----	-----------------------	-----

Beispiel: (KA 9)

Der AD-Teil vom Befehl unter Adresse 0.0.0 im Anwenderprogrammblock 0 ist folgendermaßen belegt:

Block 0

BW-Adresse	Op.Teil	Adreßteil	Symb. Adr.	Bemerkungen
0 0 0	0 0 1	0 2 9		
	1			

In den Druckbereich ist der Inhalt des dazugehörigen Druckspeicherwortes D2 mit 9 Nachkommastellen und Zwischensummenzeichen zu übertragen. Der Dezimalpunkt soll dabei in Position 63 stehen.

vorher: SW 2 + 0 0 0 1 2 3 4 7 0 0 0 0 0 0 0

BW-Adresse	Op.Teil	Adreßteil	Symb. Adr.	Bemerkungen
0	3 6 0	5 2 1		EDFS,1
1	3 2 0	7 3 15		ED,7..63
2				

nachher: SW 2 + 0 0 0 1 2 3 4 7 0 0 0 0 0 0 0

Es wird in den Druckbereich übertragen:

123.47000000 ◊

ED	Druckbefehl (edition)	3.2
----	-----------------------	-----

Bemerkungen:

1. Man beachte, daß im Fall $(D) = 0$ keine Übertragung in den Druckbereich erfolgt, wenn $AD_1 = 0$ ist und kein Druckvorbefehl mit $AD_r \neq 0$ gegeben wurde.
2. Bei einer Position größer als 8.2 wird ein Internfehlerstop ausgelöst.
3. Ist die in AD_1 angegebene Anzahl von Nachkommastellen größer als die Kommagrundausstattung KA , so erfolgen undefinierte Ausgaben.
4. Der Befehl "ED, Drucken Inhalt Druckspeicherwort (3.2)" wird simultan zum weiteren Befehlsablauf ausgeführt. Vor einer neuen Übertragung in den Druckbereich sowie vor einer Veränderung des angesprochenen Druckspeicherwortes wartet intern das Betriebsprogramm auf die Beendigung der Operation. Dazu ist kein Wartebefehl erforderlich.
5. Ist der Inhalt des Druckspeicherwortes sedezimal, so werden für die Zahlen 10 bis 15 die Zeichen ausgegeben, die den ALC-Codes 0.10 bis 0.15 zugeordnet sind.
6. Bei simultanem Ablauf des Druckbefehls sollte die KA in I_5 nicht verändert werden.

EDFS	Druckvorbefehl	3.6
------	----------------	-----

AD																			
i	l	m				r													
12	11	10	9	8	7	6	5	4	3	2	1								

EDFS Druckvorbefehl für Übertragen
in den Druckbereich des Hochleistungs-
druckers

Wirkung:

1. Der Druckvorbefehl EDFs enthält Anweisungen, die in den einzelnen Übertragungsbefehlen aus Platzgründen nicht angegeben werden können. Nach einem Übertragen in den Druckbereich ist der Vorbefehl EDFs aufgehoben, ein etwa für den 1. Drucker gegebener Druckvorbefehl EDF bleibt jedoch bestehen. Ferner setzt ein neuer Druckvorbefehl EDFs einen früheren Druckvorbefehl EDFs außer Kraft.
2. Die Wirkung des Druckvorbefehls EDFs auf die einzelnen Übertragungsbefehle ist unterschiedlich und hängt von der Art des folgenden Übertragungsbefehles ab:

EDFS vor "ALC, TW2 (2.15) Übertragen eines Zeichens in den Druckbereich":

Der Hochleistungsdrucker wird im Hauptbefehl angegeben ($AD_1 = 2$). Ein Druckvorbefehl EDFs ist nur erforderlich, wenn das im Hauptbefehl codierte Zeichen mehrmals nacheinander in den Druckbereich übertragen werden soll. In diesem Falle ist die Anzahl der Wiederholungen im AD_r des EDFs anzugeben.

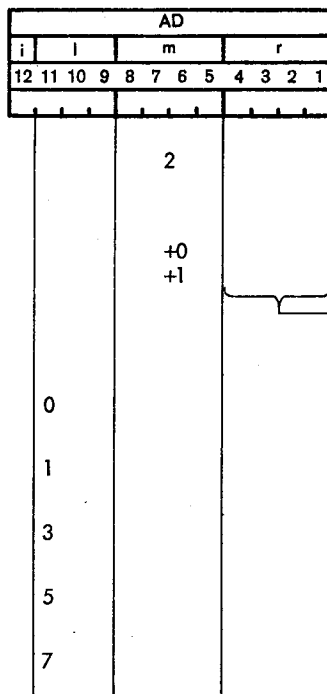
EDFS vor "TT (3.0) Übertragen einer Texttabelle in den Druckbereich":

Um eine Texttabelle in den Druckbereich zu übertragen, ist ein vorangehender Druckvorbefehl EDFs unbedingt erforderlich ($AD_{lmr} = 0.2.0$), da im Hauptbefehl das Gerät nicht angegeben ist.

EDFS vor "ALOUT6, TW2 bzw. ALOUT8, TW2 (3.1) Übertragen alphanumerischer Inhalte aus ALC-Bereich in den Druckbereich":

Vor diesem Übertragungsbefehl ist ein Druckvorbefehl EDFs nicht erforderlich, da der Hochleistungsdrucker im Hauptbefehl angegeben wird ($AD_1 = 6$). Der Einheitlichkeit halber wird jedoch ein etwa gegebener Druckvorbefehl EDFs auch durch diesen Hauptbefehl, jedoch ohne weitere Wirkung, gelöscht.

EDFS	Druckvorbefehl	3.6
------	----------------	-----



- EDFS Druckvorbefehl vor "ED (3.2) Übertragen numerischer Inhalte aus D2 in den Druckbereich"
- ,ZERO Vornullen
 - ,STAR Sicherungssysteme
- Anzahl der Vorkommastellen, die mit Vornullen oder Sicherungssystemen aufzufüllen sind
- Übertragen ohne Kennzeichen hinter der Zahl
- ,MIN Leerschritt, wenn $(D2) \geq 0$
Kennzeichen-, wenn $(D2) < 0$
 - ,SGN Kennzeichen+, wenn $(D2) \geq 0$
Kennzeichen-, wenn $(D2) < 0$
 - ,ITS Kennzeichen◇, wenn $(D2) \geq 0$ *
Kennzeichen◊, wenn $(D2) < 0$ *
 - ,FTS Kennzeichen*, wenn $(D2) \geq 0$
Kennzeichen*, wenn $(D2) < 0$ *

EDFS vor "ED (3.2) Übertragen numerischer Inhalte aus D2 in den Druckbereich:

Vor diesem Hauptbefehl ist ein vorangehender Druckvorbefehl EDFS unbedingt erforderlich, da im Hauptbefehl das Gerät nicht angegeben ist.

Für diesen Hauptbefehl gelten alle Informationen des Druckvorbefehls bezüglich des Übertragens von Vorkommastellen, der Sicherungszeichen und der Kennzeichen hinter der Zahl gemäß obiger Übersicht.

* Diese Zeichen können zur Zeit vom Hochleistungsdrucker noch nicht ausgegeben werden. Es empfiehlt sich, die Symbole ITS und FTS zu vermeiden und anstelle der Zeichen ◇, ◊ und * den Code für Leerschritt in die Codetabelle einzusetzen.