System- und Programmierhandbuch

TA10

INHALTSVERZEICHNIS

1.	**Einführung**	3
1.1	Zweck des Handbuches	3
1.2	Aufbau des Handbuches	3
2.	**Objektbeschreibung**	3
2.1	Modell TA 10/1	3
2.1.1	Ein-/Ausgabeeinheit	4
2.1.2	Verarbeitungseinheit	6
2.1.3	Zusatzeinrichtungen	6
2.2.	Modell TA 10/2	7
2.2.1	Ein-/Ausgabeeinheit	7
2.2.2	Verarbeitungseinheit	9
2.2.3	Zusatzeinrichtungen	11
2.3	Modell TA 10/3	13
2.3.1	Ein-/Ausgabeeinheit	14
2.3.2	Programmeingabe-Tastatur	14
2.3.3.	Verarbeitungseinheit	16
2.3.4	Magnetbandeinheit	16
2.3.5	Zusatzeinrichtungen	17
3.	**Funktionseinheiten**	18
3.1	Programm	18
3.2	Leitwerk	18
3.3.	Rechner und Register	20
3.3.1	Rechner	20
3.3.2	Register	20
3.4	Wagensteuerung	21
3.4.1	Schreibwerk	21
3.5	Alphasymbol - Codierer	22
3.6	Blockschaltbild	24
4.	**Programmiertechniken**	25
4.1	Mit Dioden gelötete Programmplatten	25
4.2	Stecktafeln	26
4.3	Programmplattenprüfgerät	26
4.4.	PROM	27
4.5	Arbeitsroutine für die TA 10/3-Programmerstellung	29
5.	**Programmformalismus**	31
5.1	Programmablaufplan	31
5.2	Befehlsblätter	32
6.	**TA 10 – Befehlsliste (deutsch und englisch)**	33
6.1	Transfer to variable register	35
6.2	Bring from variable register	38
6.3	Add	41
6.4	Multiply	46
6.5	Wait for key	48
6.6	Mark if key	49
6.7	Mark by value-test	50

6.8	Jump unconditional	52
6.9	Jump if Mark 1	53
6.10	Jump if Mark 2	55
6.11	Jump if Mark 3	56
6.12	Jump if Mark 4	57
6.13	Jump if negative	58
6.14	Jump if not zero	59
6.15	Jump if selector	60
6.16	No operation	61
6.17	Tabulate	62
6.18	Tabulate, free keyboard	64
6.19	Output preparation	65
6.20	Output	70
6.21	Output, free keyboard	74
6.22	Constant	75
6.23	Skip for form	78
6.24	End	79
7.	**Lochstreifen- (Karten-)Ausgabe**	80
7.1	Organisation der Daten auf dem Datenträger (allgemein)	80
7.2	Objektbeschreibung	81
7.3	Funktionseinheiten	84
7.3.1	Codierer	84
7.3.2	Kennzeichenmatrix	85
7.3.3	Schlüsselschalter	89
7.3.4	Datensicherung	90
7.3.5	Mögliche Fehlerfälle	90
7.4	Peripheriebefehle	91
7.4.1	Punch with 0 sign	92
7.4.2	Punch with 1 sign	95
7.4.3	Punch with 2 sign	96
7.4.4	Skip on record medium	98
7.4.5	Record gap	100
7.4.6	Punch constant	102
8.	**Codelist TA 10**	103

1. EINFÜHRUNG

1.1 Zweck des Handbuches

Der Zweck des Handbuches ist es, dem Programmierer der Modelle TA 10 über den Aufbau und die Wirkungsweise der Modelle Auskunft zu geben und ihm als Nachschlagwerk zu dienen.

1.2 Aufbau des Handbuches

Dem Zweck des Handbuches entsprechend ist es aufgeteilt in die Beschreibung von

- Grundmaschine
  und
- Peripherie

Innerhalb dieser Hauptteile gliedert es sich in:

- Objektbeschreibung
- Funktionsbeschreibung
- Befehlsbeschreibung

2. OBJEKTBESCHREIBUNG

2.1 Modell TA 10/1 (Fakturierautomat)

Grundausstattung dieses, in Kofferbauweise ausgeführten Modelles sind folgende, mit Verbindungskabel zusammengeschlossene Einheiten (Abb. 1):

- kombinierte Ein- /Ausgabeeinheit
- Verarbeitungseinheit

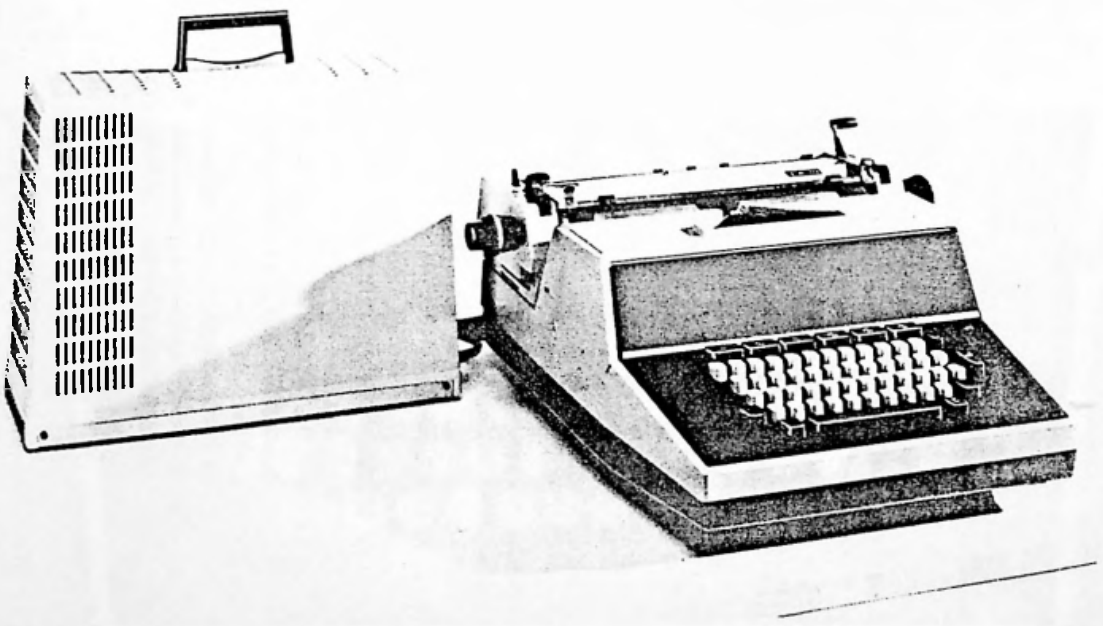


Abb. 1 TA 10/1 – mit Elektronikkoffer

2.1.1 **Ein- /Ausgabeeinheit** ist ein Typenhebel-Schreibwerk mit geschlossener Bodenwanne.

Die Bedienertastatur Abb. 2 ist eine Tastatur ähnlich der einer Schreibmaschine mit 6 zusätzlichen Steuertasten.

Steuertasten sind die Tasten ⑤ ⑥ ① ② ③ und ④

Sie werden im Programm u.a. für die Freigabe folgender Operationen eingesetzt:

- Eingaben
- Ausgaben
- Verrechnungen
- Registerwahl
- Festwertwahl
- Fehlbedienungsabsicherungen
- Fehlerkorrekturen
- Programmwahl
- Programmverzweigungen
- Datenträgerausgabe

Im Arbeitsmodus „Rechnen" erfolgt die Eingabe rein numerischer Daten über die Zifferntastenreihe. Die Buchstabentasten, Leertaste, Rücktaste und Umschalttasten sind in diesem Arbeitsmodus gesperrt.

Textschreiben (Großbuchstaben und Ziffern) ist nur an vom Programm her zugelassenen Positionen möglich. Im Arbeitsmodus „Text" ist die gesamte Tastatur entsperrt.

Mit der C-Taste (Korrekturtaste) Abb. 2 kann vor dem Betätigen einer Steuertaste das Tast-/Rechenregister (eingetasteter Wert) sowie eine fälschlicherweise ausgeführte Minustastenbetätigung gelöscht werden.

Mit der Fehlerbehebungstaste (⁄) Abb. 2 wird der im Peripheriefehlerfalle gestoppte Programmlauf nach dem Beheben des Fehlers wieder gestartet.

Die Minustaste Abb. 2 hat keine auslösende Funktion. Sie wirkt lediglich vorbereitend und muß vor der jeweiligen Steuertaste betätigt werden.

Die grüne Signallampe im rechten Tastenfeld Abb. 2 zeigt die „Betriebsbereitschaft" der Maschine an.

Das Modell TA 10/1 wird nur mit Wagengröße II ausgeliefert.

Im Arbeitsmodus „Rechnen" wird bei dieser Wagengröße eine Schreibgeschwindigkeit von **18 Zeichen / Sekunde** erreicht.

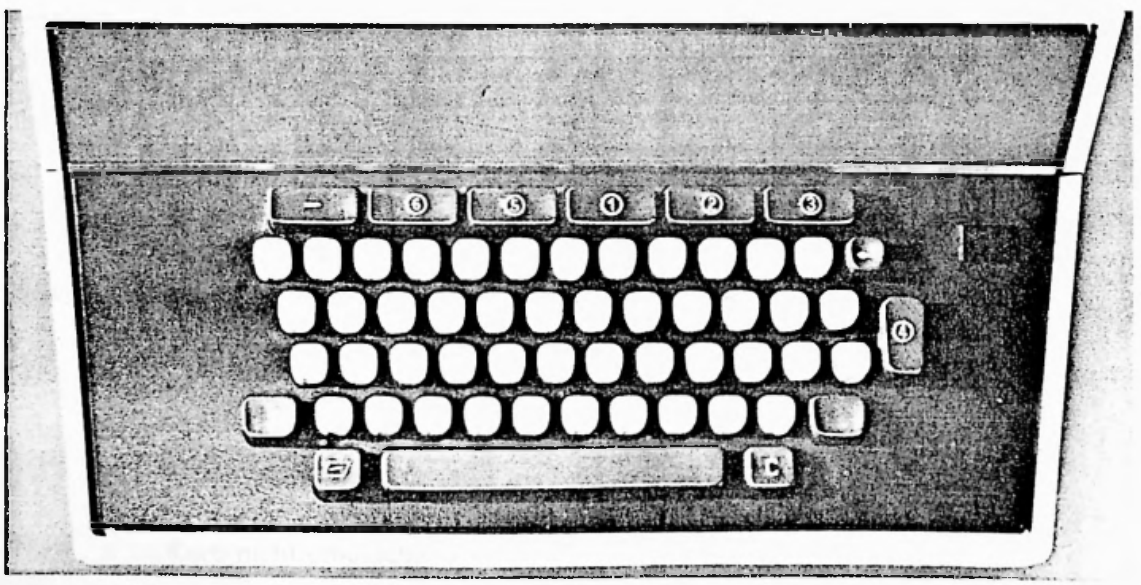


Abb. 2 TA 10/1 – Bedienungstastatur mit Steuertasten

Spezifikation der Ausgabe:

Walzenbreite:	339 mm
max. zulässige Papierbreite:	352 mm
Zeichenabstand	1/10 Zoll (2,54 mm)
Anschläge pro Zeile:	127
Schriftart:	Ro 587
Anzahl Typenhebel:	46
Zeichenvorrat:	siehe Abb. 3
Zeilenabstand:	4,25 mm
Zeilenschaltung:	1-; 1,5; 2-zeilig
Farbband:	14,3 mm breit, schwarz
Anschlagregler:	3 markierte Einstellmöglichkeiten
Textschreiben	manuell, nur Großbuchstaben

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	Q	A		Y	W	S	3	X	E	4	D	C	R	F	5	V	T	G	6	B	Z	H	KUT
MW				★	◇	‰	%	–	0	1	2	3	4	5	6	7	8	9	(	&	"	+	$\overline{\text{KUT}}$

Leerschritt

24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	
N	7	U	J	8	M	I	K	9	,	0	L	0	.	P	Ö	_	–	Ü	Ä		2		KUT
=	)	/	9	8	7	6	5	4	3	2	1	0	–	%	‰	◇	★				MW		$\overline{\text{KUT}}$

Leerschritt

Abb. 3a Schreibbarer Zeichenvorrat des Modells TA 10/1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	Q	A		Y	W	S	3	X	E	4	D	C	R	F	5	V	T	G	6	B	Z	H	KUT
MW				★	◇	‰	%	–	0	1	2	3	4	5	6	7	8	9	(	&	"	+	$\overline{\text{KUT}}$

Leerschritt

24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	
N	7	U	J	8	M	I	K	9	,	0	L	0	.	P	Ö	_	–	Ü	Ä		2		KUT
=	)	/	9	8	7	6	5	4	3	2	1	0	–	%	‰	◇	★				MW		$\overline{\text{KUT}}$

Leerschritt

Abb. 3b Schreibbarer Zeichenvorrat der Modelle TA 10/2 und TA 10/3

KUT = Korb umgeschaltet

$\overline{\text{KUT}}$ = Korb nicht umgeschaltet

◤ = mit Magneten angesteuerte Typenhebel

2.1.2 Verarbeitungseinheit

Die in Kofferbauweise ausgeführte Verarbeitungseinheit beinhaltet außer dem Netzgerät 5 Schaltplatten:

- Platte 1: Wagensteuerung
- Platte 2: Rechner und Speicher
- Platte 3: Leitwerk
- Platte 4 + 5: Programme (PROM auf Platz 5)

Die verwendeten Schaltkreise sind TTL – MOS – und DTL – Schaltkreise, die sich durch große Zuverlässigkeit auszeichnen.

Durch die Steckbarkeit der einzelnen Baugruppen (Schaltplatten) wurde ein service-freundlicher Aufbau des Modells erreicht.

Außerdem sind an der Verarbeitungseinheit, von links ausgehend, die 39-polige Steckerleiste für den Anschluß der kombinierten Ein- /Ausgabeeinheit, die 39-polige Steckerleiste für den Peripherieanschluß und die 3-polige Europasteckdose für den Stromanschluß der Ein- /Ausgabeeinheit zu finden.

Der zwischen der 39-poligen Steckerleiste und der Europa-Steckdose angeordnete Schiebeschalter ist der Hauptschalter (EIN/AUS-Schalter) der Maschine. Mit ihm werden beide Einheiten gemeinsam geschaltet.

An dem am Netzgerät angeordneten Spannungswähler können abhängig von dem eingebauten Schreibwerksmotor, dessen Anschlußspannung mit der Netzspannung identisch sein muß, folgende Netzspannungen eingestellt werden:

220 / 130 / 240 oder 110 Volt

Werkseitig sind 220 Volt eingestellt und ein Schreibwerksmotor für 220 Volt Anschlußspannung eingebaut.

Da sämtliche Gleichspannungen elektronisch stabilisiert sind, können Netzschwankungen innerhalb der Toleranzen von + 10% und - 15% ausgeglichen werden.

Das Modell kann wahlweise für 50 oder 60 Hz ausgerüstet werden.

Die Leistungsaufnahme beträgt etwa 150 Watt.

Als peripheres Gerät kann an das Modell TA 10/1 ein:

- Lochstreifen-(Karten) Stanzer (8-Kanal) angeschlossen werden.

2.1.3 Zusatzeinrichtungen

- **Einfache mechanische Endlosformular-Einrichtung**

  Diese Einrichtung ermöglicht es mit der Maschine alternativ entweder Endlosformulare oder herkömmliche Formulare zu verarbeiten. Der Antrieb erfolgt mechanisch von der Schreibwerkswalze.

- **Endlosformularständer**

  Dieser Ständer ist für die Aufnahme sowie für die Abnahme der mit der Maschine zu verarbeitenden Endlosformulare notwendig. Er dient jedoch auch zu einer besseren Führung der zusammenhängenden Formulare.

- **Lochstreifen- (Karten-) Stanzer (8-Kanal)**

  Der Lochstreifen- (Karten-) Stanzer ist eine in einem Beistelltisch zusammengefaßte separate Einheit mit eigenem Netzanschluß, die mit einem Kabel steckbar mit der Verarbeitungseinheit verbunden ist. Damit können numerische Daten auf 8-Kanal-Streifen oder Karten gestanzt werden.
  Durch die fahrbare Ausführung des Beistelltisches kann er in beliebiger Anordnung dem Modell TA 10/1 beigestellt werden.

2.2 Modell TA 10/2 (Abrechnungsautomat)

Grundausstattung dieses Modelles sind folgende in einem Maschinentisch integrierte Einheiten (Abb. 4):

- **kombinierte Ein-/Ausgabeeinheit** mit separater Sondertastatur
- **Verarbeitungseinheit** mit auswechselbarer Programmkassette

2.2.1 **Ein-/Ausgabeeinheit** ist ein Typenhebelschreibwerk, das im Maschinentisch auf einem Klapprahmen montiert ist.

Die Bedienertastatur des Schreibwerkes Abb. 5 ist nur für den Arbeitsmodus „Text" ausgelegt. Textschreiben ist jedoch nur an vom Programm her festgelegten Positionen möglich.

Im Arbeitsmodus „Rechnen" ist die Bedienertastatur des Schreibwerkes gesperrt.

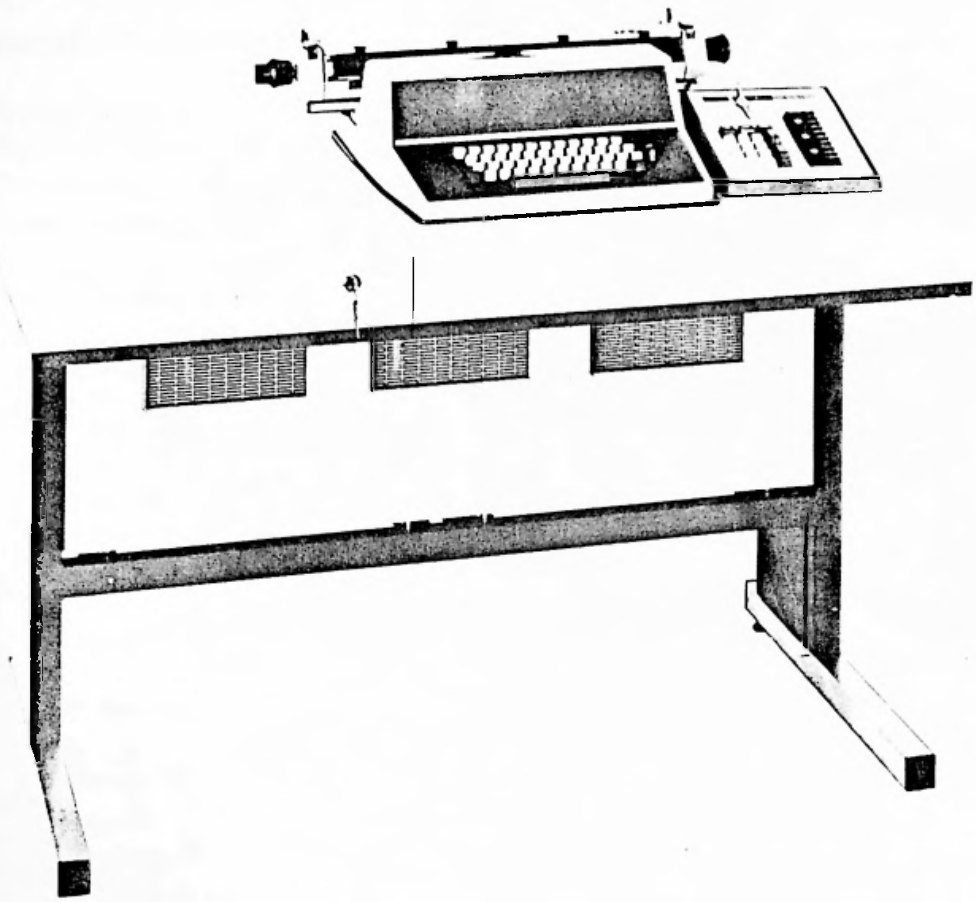


Abb. 4 TA 10/2 – Programmkassette hinter verschlossener Tischblende

Numerische Daten werden im Arbeitsmodus „Rechnen" über den Zehnertastenblock in der separat angeordneten Sondertastatur (Abb. 5) in die Maschine eingegeben. Der Zehnertastenblock ist im Arbeitsmodus „Text" gesperrt.

Außer dem Zehnertastenblock sind in der Sondertastatur Abb. 5 die:

- Steuertasten
- C-Taste (Korrekturtaste)
- Minustaste
- Fehlerbehebungstaste
- Alphasymboltasten
- und ein Schlüsselschalter

zu finden.

Die Steuertaste (4) in der Sondertastatur ist mit der in der Schreibwerkstastatur angeordneten Taste (4) parallel geschaltet.

Die Steuertasten werden vom Programm u.a. für die Freigabe folgender Operationen eingesetzt:

- Eingaben
- Ausgaben
- Verrechnungen
- Registerwahl
- Festwertwahl
- Fehlbedienungsabsicherungen
- Fehlerkorrekturen
- Programmwahl
- Programmverzweigungen
- Datenträgerausgabe

Mit der C-Taste (Korrekturtaste) Abb. 5 kann vor dem Betätigen einer Steuertaste das Tast/Rechenregister (eingetasteter Wert) sowie eine fälschlicherweise ausgeführte Minustastenbetätigung gelöscht werden.

Die Minustaste hat keine auslösende Funktion. Sie wirkt lediglich vorbereitend und muß vor der jeweiligen Steuertaste betätigt werden.

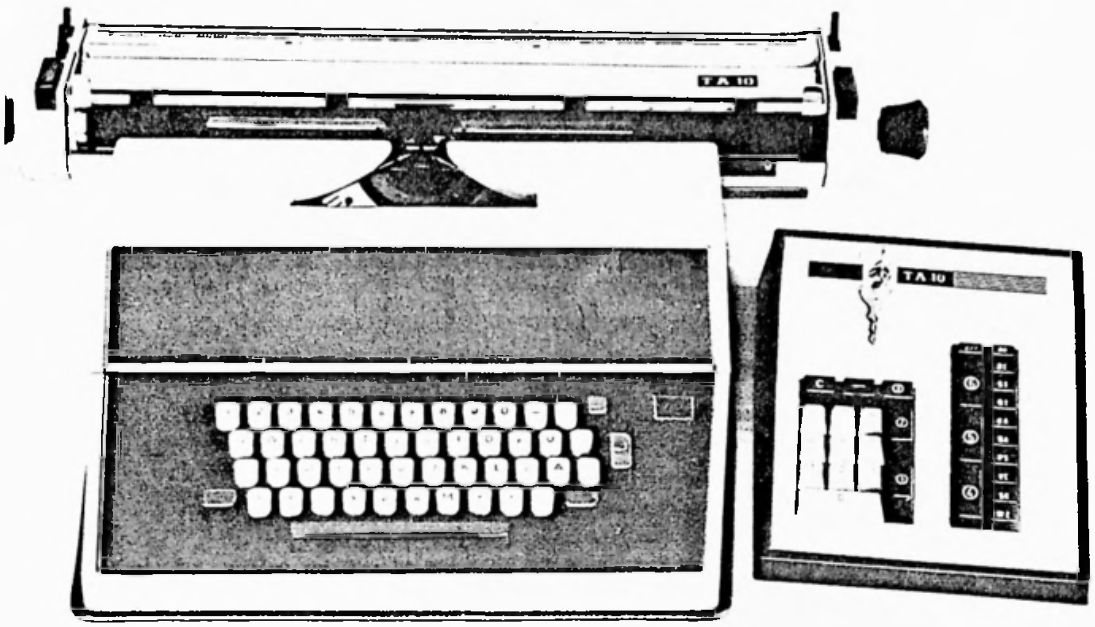


Abb. 5 TA 10/2 – Bedienungstastatur Schreibwerk und Sondertastatur

Mit der Fehlerbehebungstaste (⁄) Abb. 5 wird der im Peripheriefehlerfalle gestoppte Programmlauf nach dem Beheben des Fehlers wieder gestartet.

Mit den 9 Alphasymboltasten können in Verbindung mit dem Programm Ziffernniederschriften (z.B. Buchungsposten) mit anhängend niedergeschriebenen zweistelligen Buchstabenkombinationen gekennzeichnet werden.

Die Alphasymboltasten sind mechanisch gegenseitig entrastende, selbst jedoch rastende Tasten.

Für die Entsperrung einer gerasteten Alphasymboltaste ist die Entsperrtaste (⊗) zu betätigen.

Mit dem Schlüsselschalter sind an den Modellen TA 10/2, TA 10/3 zwei verschiedene Arbeitsmodi einstellbar.

- Abarbeiten des Programmes **mit wirksamen Peripheriebefehlen**
- Abarbeiten des gleichen Programmes (mit oder ohne gestecktem peripheren Gerät) **jedoch mit wirkungslosen Peripheriebefehlen.** Die Peripheriebefehle werden in dieser Schalterstellung überlaufen.

Die grüne Signallampe im rechten Tastenfeld der Schreibwerkstastatur Abb. 5 zeigt die Betriebsbereitschaft der Maschine an.

Die Modelle TA 10/2, TA 10/3 können wahlweise mit den Wagengrößen II, IV oder VI ausgerüstet werden.

Im Arbeitsmodus „Rechnen" werden abhängig von der verwendeten Wagengröße folgende maximalen Schreibgeschwindigkeiten erreicht.

Wagengröße:	II	IV	VI
max. Schreibgeschwindigkeit (Zeichen/s)	18	14	10
sonstige technische Daten:			
Walzenlänge (mm)	339	477	638
max. zulässige Papierbreite (mm)	352	490	651
Zeichenabstand (Zoll)	1/10	1/10	1/10
Anschläge pro Zeile	127	171	235
Schriftart	Ro 587	Ro 587	Ro 587
Anzahl Typenhebel	45	45	45
Zeichenvorrat, siehe	Abb. 3	Abb. 3	Abb. 3
Zeilenabstand (mm)	4,25	4,25	4,25
Zeilenschaltung (-zeilig)	1, 1,5, 2	1, 1,5, 2	1, 1,5, 2
Farbband *)	schwarz	schwarz	schwarz
Anschlagregler (markierte Einstellmöglichkeit)	3	3	3
Textschreiben	manuell, nur Großbuchstaben		

*) Werkseitig werden alle Modelle mit schwarzem Farbband ausgeliefert. Durch Umschalten, vom Kundendienst vorgenommen, kann auch mit Farbband rot/schwarz geschrieben werden.

2.2.2 Verarbeitungseinheit

Die Verarbeitungseinheit ist bei diesem Modell im Maschinentisch eingebaut. Sie ist von der Rückseite des Tisches zugängig. Um unbefugte Eingriffe an der Verarbeitungseinheit zu verhindern, ist der Zugang (Klappe in der Rückwand) mit einer Schlüsselverriegelung gesichert.

In Standardausführung beinhaltet sie, in einem Rahmengestell zusammengefaßt, 4 Schaltplatten und das Netzgerät.

- Platte 1: Wagensteuerung
- Platte 2: Rechner und Speicher
- Platte 3: Leitwerk
- Platte 4: – – –
- Platte 5: Adaption – für Programmkassette / TA 10/2 oder
  Adaption – für Programmspeicher / TA 10/3

Die Platte 4 – Endlosformularsteuerung – wird nur dann benötigt, wenn eine der Zusatzeinrichtungen wie EF, EFW oder EFV an der Maschine verwendet wird.

Das Programm ist bei diesem Modell in einer Kassette, die maximal 2 Programmplatten mit je 128 Befehlswortplätzen beinhaltet, gespeichert. Die Programmkassette befindet sich in dem verschließbaren Einschub an der linken Tischvorderseite (Abb. 6). Sie ist von der Bedienungskraft leicht auswechselbar.

Die verwendeten Schaltkreise sind TTL-, MOS und DTL-Schaltkreise, die sich durch große Zuverlässigkeit auszeichnen.

Durch die Steckbarkeit der einzelnen Baugruppen (Schaltplatten) wurde ein servicefreundlicher Aufbau des Modells erreicht. **Verstärkt wird die Servicefreundlichkeit durch die Tatsache, daß die Baugruppen des Modells TA 10/1 mit jenen des Modells TA 10/2 sowie denen der TA 10/3-Verarbeitungseinheit identisch sind.**

Rechts neben der Programmkassette Abb. 6 dem MB-Kassettengerät Abb. 8 ist der Hauptschalter (EIN-/AUS-Schalter) der Maschine eingebaut.

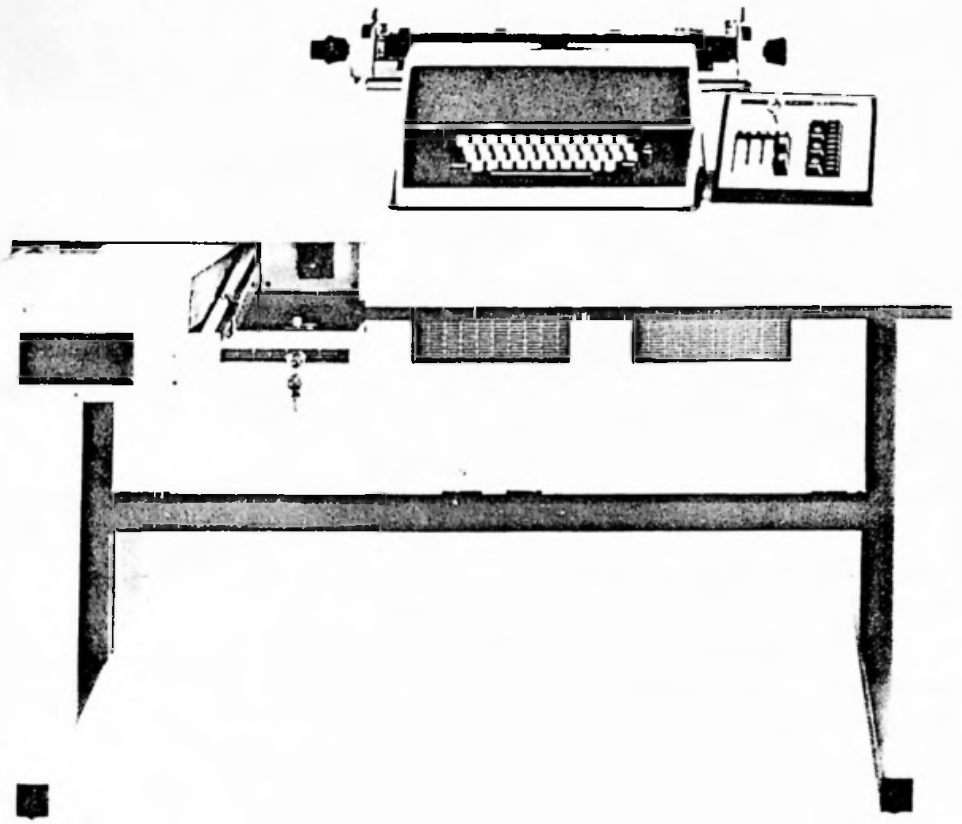


Abb. 6 TA 10/2 – mit herausgezogener Programmkassette

An dem am Netzgerät angeordneten Spannungswähler können abhängig von dem eingebauten Schreibwerksmotor und Vorsteckeinrichtungs-Antriebsmotor, dessen Spannungen immer mit der Netzspannung identisch sein müssen folgende Netzspannungen eingestellt werden:

220 / 130 / 240 oder 110 Volt

Werkseitig sind 220 Volt eingestellt und Motoren für 220 Volt Anschlußspannungen eingebaut.

Da sämtliche Gleichspannungen elektronisch stabilisiert sind, können Netzschwankungen innerhalb der Toleranzen von + 10% und - 15% ausgeglichen werden.

Beide Modelle TA 10/2 und TA 10/3 können wahlweise für 50 oder 60 Hz ausgerüstet werden.

Die Leistungsaufnahme beträgt etwa 150 Watt.

Als peripheres Gerät kann an den Modellen TA 10/2, TA 10/3 ein:

– **Lochstreifen (Karten-) Stanzer (8-Kanal)**

angeschlossen werden.

Die an der Rückseite des Maschinentisches herausgeführten Steckerleisten Abb.7 stellen die max. Anschluß-Ausführung dar und dienen dem Anschluß folgender Zusatzeinrichtungen.

- Die 39-polige Steckerleiste (1) gehört zu dem Anschluß der Peripherie.
- Die 30-polige Steckerleiste (2) gehört zu dem Anschluß der Formular-Einrichtung.
- Der 3-polige Europastecker (3) ist der Stromanschluß für motorisch angetriebene Vorsteckeinrichtungen. (Formulareinrichtungen).

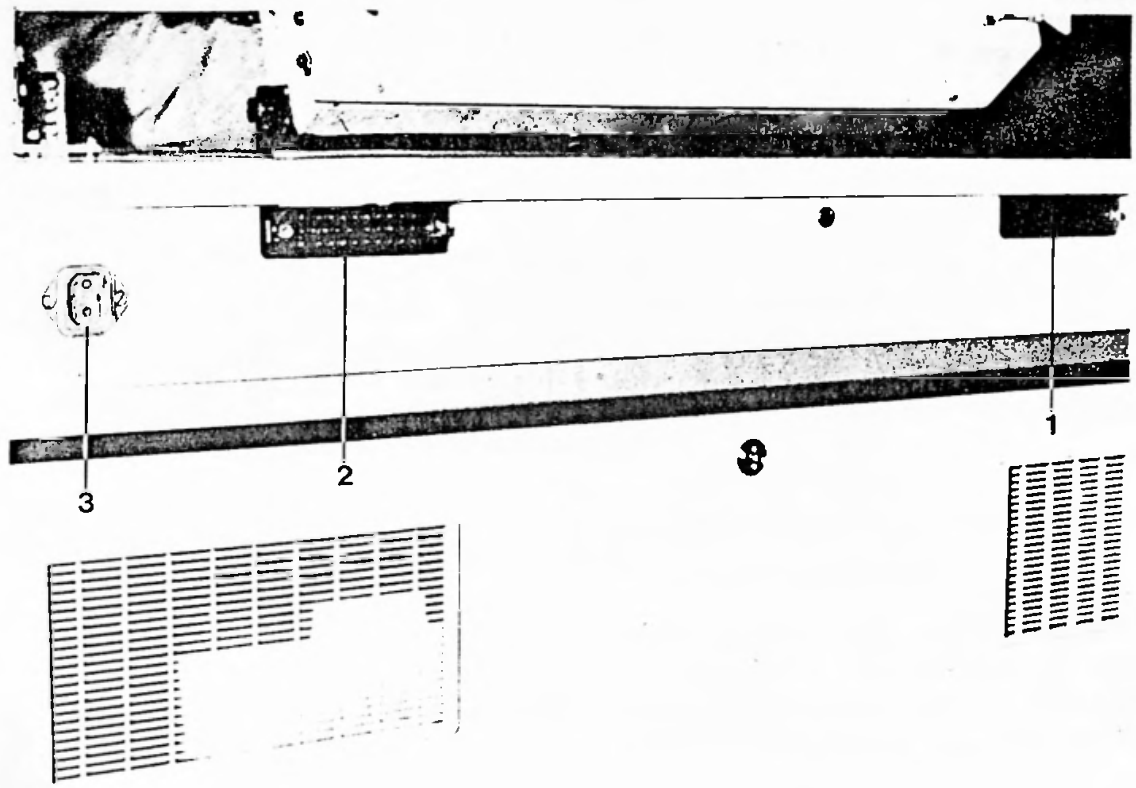


Abb. 7 TA 10/2 (TA 10/3) – Maschinentischrückseite

2.2.3 Zusatzeinrichtungen

- **Einfache Endlosformular-Einrichtung (EFS)**
  für Wagengröße II oder IV
- **Endlosformular-Einrichtung (EF)**
  für Wagengröße II, IV oder VI
- **Endlosformular-Einrichtung (EFW)**
  für Spezialwagen, Wagengröße IV oder VI
- **Endlosformular-Einrichtung mit Vorsteckeinrichtung (EFV)**
  für Wagengröße IV oder VI (Spezialwagen)
- **Doppelte Konten-Vorsteckeinrichtung (DKE)**
  für Wagengröße IV oder VI (Spezialwagen)
- **Endlosformularständer**
- **Lochstreifen- (Karten-) Stanzer** (8-Kanal)

Die **einfache mechanische Endlosformular-Einrichtung** ermöglicht es mit der Maschine alternativ entweder Endlosformulare oder herkömmliche Formulare zu verarbeiten.
Ihr Antrieb erfolgt mechanisch von der Schreibwerkswalze.

Die **Endlosformular-Einrichtung** gestattet über einen programmierten Lochstreifen (Steuerstreifen) einen programmbezogenen Transport des Endlosformulares. Der Antrieb ist elektromechanisch. Sie kann jedoch nur auf einen Normalwagen mit nur einer – der üblichen – Papierführungswanne aufgesetzt werden. Somit können 2 Formulararten nur alternativ beschriftet werden; und zwar entweder ein Endlosformular oder ein herkömmliches Formular.

Die **Endlosformular-Einrichtung EFW** unterscheidet sich von der EF-Einrichtung hauptsächlich dadurch, daß sie nur auf einen Spezialwagen aufgesetzt werden kann, der neben der normalen Papierführungswanne für das Journal mit einer weiteren Papierführungswanne für das Endlosformular ausgestattet ist. Es können also gleichzeitig 2 Formulararten, auch überlappend, verarbeitet werden:

- ein herkömmliches Formular
- und ein Endlosformular

Die **Endlosformular-Einrichtung mit Vorsteckeinrichtung** (EFV) kann, wie die EFW nur auf einen Spezialwagen aufgesetzt werden, der neben der normalen Papierführungswanne für das Journal, mit einer weiteren Papierführungswanne für ein Endlosformular und/oder einer Kontokarte ausgerüstet ist. Somit können mit dieser Einrichtung gleichzeitig bis zu 3 Formulararten, auch überlappend, beschriftet werden:

- ein Endlosformular
- ein herkömmliches Journal
- und eine Kontokarte

Die **doppelte Vorsteckeinrichtung (DKE)** kann ebenfalls nur auf einen Spezialwagen aufgesetzt werden, der neben der normalen Papierführungswanne für das Journal, mit einer weiteren Papierführungswanne für die Kontokarten ausgestattet ist. Mit dieser Einrichtung können ebenfalls gleichzeitig bis zu 3 Formulararten, auch überlappend, verarbeitet werden:

- ein herkömmliches Journal
- und zwei Kontokarten

Der Endlosformularständer ist für die Aufnahme sowie für die Abnahme der in der Maschine zu verarbeitenden Endlosformulare notwendig. Er dient jedoch auch einer besseren Führung der zusammenhängenden Formulare.

Der Lochstreifen- (Karten-) Stanzer ist eine in einem Beistelltisch zusammengefaßte separate Einheit mit eigenem Netzanschluß, die mit einem Kabel steckbar mit der Grundmaschine verbunden ist.

Mit ihm können numerische Daten auf 8-Kanal-Streifen oder Karten gestanzt werden.

Durch die fahrbare Ausführung des Beistelltisches kann er in beliebiger Anordnung der Grundmaschine beigestellt werden.

2.3 Modell TA 10/3 (Abrechnungscomputer mit Programm (-Tastatur)-Kassettenspeicherung)

Das Modell TA 10/3 unterscheidet sich u.a. vom Modell TA 10/2 durch eine zusätzliche Programmschnittstelle.

Eine weitere Besonderheit dieser Maschine sind die 7 freien Speicher von je 12 Ziffern und Vorzeichen (I) und als neue Modifikation 15 freie Speicher mit einer Kapazität von je 15 Ziffern und Vorzeichen (II). Dementsprechend sind bei den folgenden Befehlen, soweit erforderlich, die Unterschiede hervorgehoben. **Die Modifikation mit 7 freien Speichern steht jeweils im ersten Teil der Befehlsbeschreibung und ist mit (I) gekennzeichnet, die Ausführung mit 15 freien Speichern steht im zweiten Teil der Befehlsbeschreibung und ist mit (II) gekennzeichnet.**

Die Grundausstattung der TA 10/3 besteht aus folgenden Maschineneinheiten:

- **Kombinierte Ein-/Ausgabeeinheit**
  mit separater Sondertastatur
- **Programmeingabe-Tastatur (PET 1) ***)
- **Verarbeitungseinheit**
  mit Programmspeicher-Elektronik
- **Magnetbandeinheit**
  mit auswechselbarer Bandendlos-Programmkassette

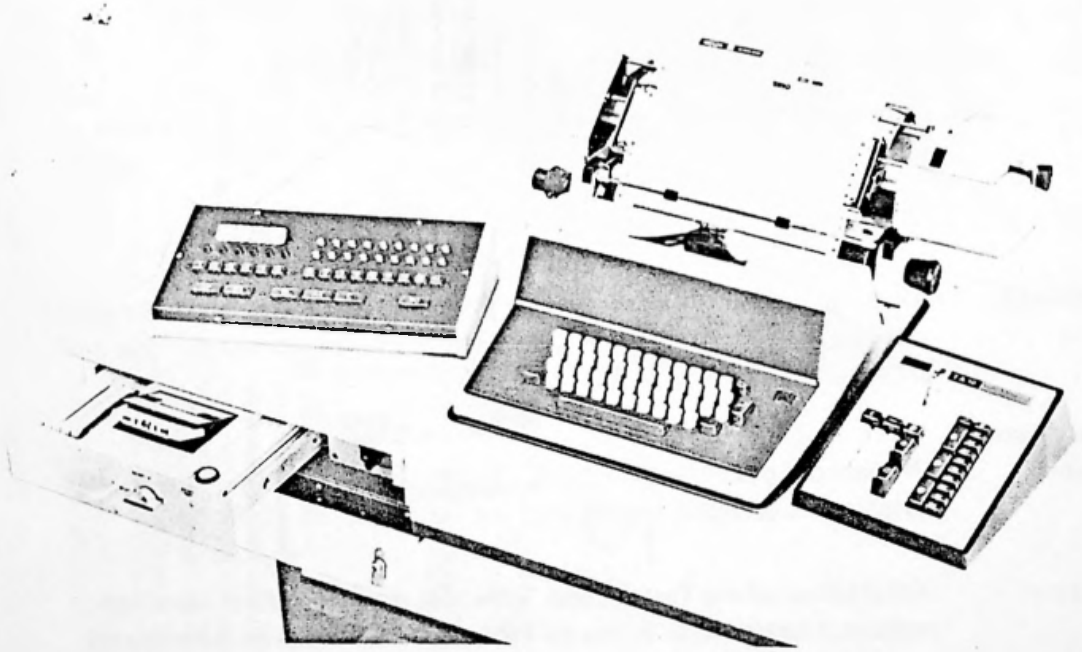


Abb. 8 TA 10/3 – mit angeschlossener Programm-Eingabetastatur, herausgezogener Magnetbandeinheit, eingesetzter Bandendlos-Programmkassette und Sondertastatur

*) wird nur bei Bedarf mitgeliefert (Generalvertretungen usw.)

2.3.1 Kombinierte Ein-/Ausgabeeinheit
mit separater Sondertastatur (s. dazu Pkt. 2.2.1)

2.3.2 Programmeingabe-Tastatur (Abb. 10)

An dem Modell TA 10/3 kann zusätzlich für Programmierungsarbeiten eine Programmeingabe-Tastatur (PET 1) angeschlossen werden.

Über die Programmeingabe-Tastatur erfolgt die Eintastung der binär verschlüsselten Befehlsworte. Es können bis zu max. 512 Befehlsworte eingegeben werden. Mit nachfolgender Aufzeichnung des Programmes auf Magnetband (MB) ist eine Form der Datenspeicherung geschaffen, die es jederzeit ermöglicht, die Programmeingabe auch vom MB-Gerät aus vorzunehmen.

Durch die Konzeption der Programmeingabe-Tastatur sind binnen kürzester Zeit Fehlerquellen in der Programmeingabe zu erkennen, Programme zu kontrollieren und Befehlsworte zu korrigieren.

Die Programmeingabe-Tastatur (num.) läßt nachstehenden Aufbau erkennen (s. Abb. 9)

- Lampenfeld oben (gelb) – Programmschrittzähler-Anzeige
- Lampenfeld darunter (rot/gelb) – Befehlswortinhalt-Anzeige
- Tastatur Betriebsart – ET; MBA; MBW; Prog.
- Befehlstasten – 1-14 (1-6 Operationsteil/7a-14 Adreßteil)
- Auslösetasten – LP; RS; BL; AÜ; C; BÜ.

Die Betriebsarttasten werden für die Freigabe folgender Betriebsarten eingesetzt:
(s. auch B.H.B. TA 10/3)

Betriebsarten

ET	:	(Eingabe-Tastatur)-**Programmeingabe nach MOS-Speicher**
MBA	:	(Magnetband-Aufzeichnung)-**Übernahme** MOS-Speicher **nach MB**
MBW	:	(Magnetband-Wiedergabe)-**Übernahme** MB **nach MOS-Speicher**
Prog.	:	(Programm-Betrieb)-**Übernahme** MOS-Speicher **nach TA 10/3-Elektronik**

Befehlstasten:
1 bis 14 — Die Befehlstasten 1 - 14 dienen der Programmeingabe, d.h. der Eingabe einzelner Befehlsworte. (Bitmustereingabe im Operationsteil Taste 1-6/ im Adreßteil Taste 7a - 14).

Auslösetasten:

C - Taste: (Rücksetzen des Pufferspeichers). Mit der C-Taste wird ein fehlerhaftes Befehlswort gelöscht, jedoch stets vor Betätigen der BÜ-Taste.

BÜ - Taste: (Befehlsübernahme-Taste) Diese Taste löst die Übernahme des eingetasteten Bitmusters, d. h. das im Pufferspeicher stehende Befehlswort in den MOS-Speicher aus.

Der Programmschrittzähler wird anschließend um eins erhöht und das Pufferregister gelöscht. (Lampenreihe rot/gelb verlöscht).

Das Bitmuster des nächsten Befehlswortes kann eingegeben werden.

BL - Taste: (Befehl-Lesen) Mit dieser Taste wird eine Überprüfung zuvor eingetasteter Befehlsworte (Programm) durch wortweises Lesen vorgenommen. Der Programmschrittzähler wird dabei um 1 erhöht und das einzelne Befehlswort angezeigt.

AÜ - Taste: (Adreßübernahme-Taste) Die gewünschte Adresse ist über die Adreßteil-Tasten 7a - 14 anzuwählen. Damit ist der Adresseninhalt auf einen beliebig angewählten Wert gesetzt und wird bei Betätigen der AÜ-Taste in den Programmschrittzähler übernommen.

LP - Taste: (Lampen-Prüfen) Die Kontrolle über ein funktionsfähiges Gesamt-Lampenfeld erfolgt mit der LP-Taste. Das Lampenfeld wird auf Ausfälle geprüft.

RS - Taste: (Rücksetzen des Programmschrittzählers auf 0)
Mit dieser Taste wird der Programmschrittzähler auf „0" gesetzt. Das im MOS-Speicher vorhandene Programm bleibt erhalten.

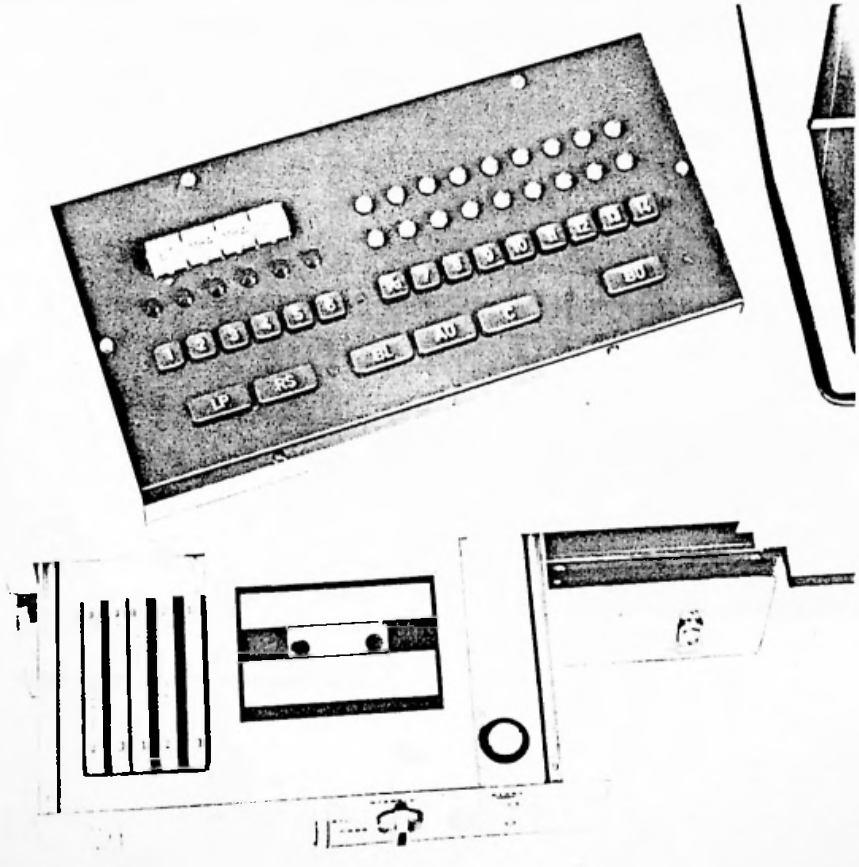


Abb. 9 TA 10/3 – Programmeingabe-Tastatur mit herausgezogener Magnetband-Einheit, Bandendlos-Programmkassette und Programm-Datei

2.3.3. Verarbeitungseinheit

(s. dazu Pkt. 2.2.2 außer letztem Abschnitt S. 16)
mit **Programmspeicher-Elektronik** — wie folgt:

Zusätzlich ist neben der Verarbeitungseinheit des Modells die Programmspeicher-Elektronik installiert.
Die Programmspeicher-Elektronik beinhaltet 2 Elektronik- und Adapterplatten sowie ein Netzgerät.

- Platte 1 : Steuerung
- Platte 2: MOS-Speicher
- Platte 3: Adaption zur Verarbeitungseinheit
- Platte 4: Adaption zur Magnetbandeinheit

Die Programmspeicher-Kapazität ist in ihrer max. Ausbaustufe mit 512 Programmwortplätzen konzipiert.

2.3.4 **Magnetbandeinheit** mit auswechselbarer Bandendlos-Programmkassette

Die Programmeingabe wird in dem Modell TA 10/3

a) über die Programmeingabe-Tastatur

b) über die Magnetbandeinheit

durchgeführt.

Anstelle der bisherigen Programmkassette beim Modell TA 10/2 ist beim Modell TA 10/3 die Magnetbandeinheit im Maschinentisch-Einschub untergebracht (Abb. 8).

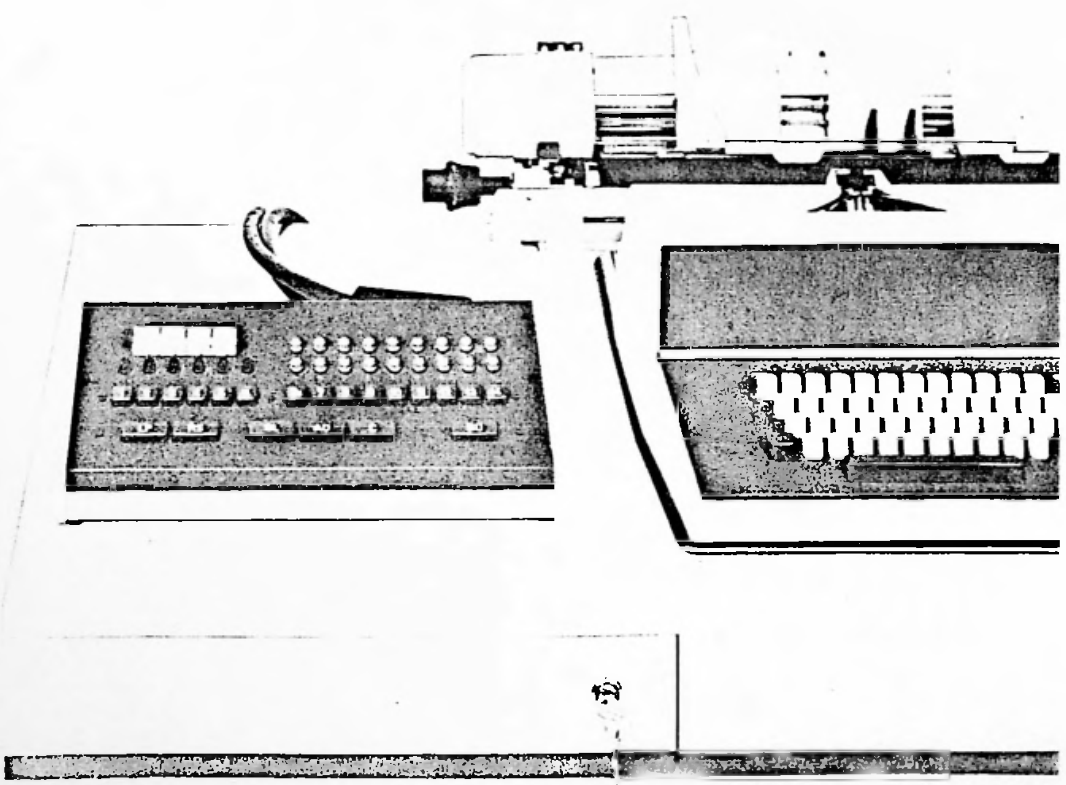


Abb. 10 TA 10/3 — mit Programmeingabe - Tastatur PET 1 und verschlossener Tischblende

Über eine Elektronik-Platte im MB-Gerät, welche durch Adaption mit der Programmspeicher-Elektronik verbunden ist, erfolgt die Magnetbandsteuerung. Sämtliche Bandinformationen müssen zur Verarbeitung in den MOS-Speicher der Grundmaschine eingelesen werden. Erst nach Erkennen des Bandendbefehls und eines fehlerfrei eingelesenen Programms erfolgt eine automatische Abschaltung des Lesevorganges. Die Lese- und Schreibgeschwindigkeit (Bandgeschwindigkeit) beträgt max. 500 bit/sek bei 4,75 cm/sek.

Als Programm-Speichermedium dient ein Endlos-Magnetband-(Kassette) mit der Länge $9 \pm 0{,}4$ m und der Breite 3,81 mm. Die MB-Speicherkapazität ist dabei $\sim 4 \times 10^4$ Bit. Die MB-Programmkassette ist von der Bedienungskraft leicht auswechselbar.

Schaltbereich des MB-Kassettengerätes

Knebelschalter (nur mech.) – 3 Schaltmarkierungen:

„START-▶"	:	Magnetbandgerät – Startbereit für Schreiben oder Lesen
„STOP"	:	Mech. Stoppen des Magnetbandes
„CASS"	:	(Kassettenwechsel)-Öffnen des Kassettenschachtes – Entnehmen oder Einlegen der Bandendlos-Programmkassette

Taste - „START" (Leuchtdrucktaste / gelb): Start des Magnetbandgerätes

ACHTUNG: Generell ist bei Kassettenwechsel die Grundmaschine nicht abzuschalten!
(z.B. falls versehentlich eine leere bzw. falsche Bandendlos-Programmkassette eingelegt wurde).
Hierzu ist nach den Schaltmarkierungen des Knebelschalters vorzugehen.

2.3.5 Zusatzeinrichtungen

(s. dazu Pkt. 2.2.3)

3. FUNKTIONSEINHEITEN

3.1 Programm

3.1.1 Diodenplatten

Das Programm ist eine Folge strukturierter Befehle mit festen Wortlängen, die in der Reihenfolge der Ausführung hintereinander programmiert und auf Diodenplatten gespeichert sind.

Die Befehlswortlänge beträgt 15 Binärstellen, die wie folgt gegliedert sind:

Die Binärstelle 7a ist bei Diodenplatten ohne Bedeutung.

1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
Operationsteil						Adreßteil								

Maximal stehen für die Programmierung der Diodenplatten 256 derartiger Befehlswortplätze (Programmschritte) auf 2 Programmplatten a 128 Befehlswortplätzen zur Verfügung.

3.1.2 Magnetbandkassetten und PROM-Bausteine

Bei der Programmierung von Magnetbandkassetten und PROM-Bausteinen wird die Stelle 7a in die Adressierung mit einbezogen. Es sind dabei 6 Binärstellen dem Operationsteil und 9 Binärstellen dem Adreßteil zugeordnet.

Für die Programmierung sind auf einer PROM-Schaltplatte 256 Befehle (2 PROM Bausteine) oder 512 Befehle (4 PROM Bausteine) möglich.

Im Operationsteil wird der Befehlsname, der aussagt „was von der Maschine zu tun ist" – also die Benennung der auszuführenden Operation –, und im Adreßteil dazu die Anweisung „mit wem oder wo" (was zu tun ist) verschlüsselt.

Das Programm steht hauptsächlich mit dem Leitwerk (3.2) in Verbindung. Von ihm wird der jeweils auszuführende Befehl auf den Programmplatten oder Magnetband ausgewählt und dessen Codierung dem Leitwerk zurückgemeldet.

3.2 Leitwerk

Im Leitwerk befindet sich der Programmzähler (bei PROM-Bausteinen auf der PROM-Schallplatte) (= Programmschrittschalter), mit dessen Hilfe der jeweils auszuführende Befehl auf den Programmplatten ausgewählt und angesteuert wird.

Normalerweise werden von ihm die Befehle in der auf den Programmplatten gespeicherten Reihenfolge angesteuert. Dieser sequentielle Programmzwangsablauf wird damit erreicht, da mit jedem an das Leitwerk rückgemeldeten Ende einer Befehlsausführung ein Weiterschaltimpuls an den Programmschrittschalter abgeleitet wird.

Wird vom Leitwerk jedoch ein Sprungbefehl interpretiert, so wird der Programmschrittschalter nicht einfach weitergeschaltet, sondern auf die im Adreßteil dieses Befehles programmierte Befehlswortplatzadresse (Sprungadresse) gesetzt.

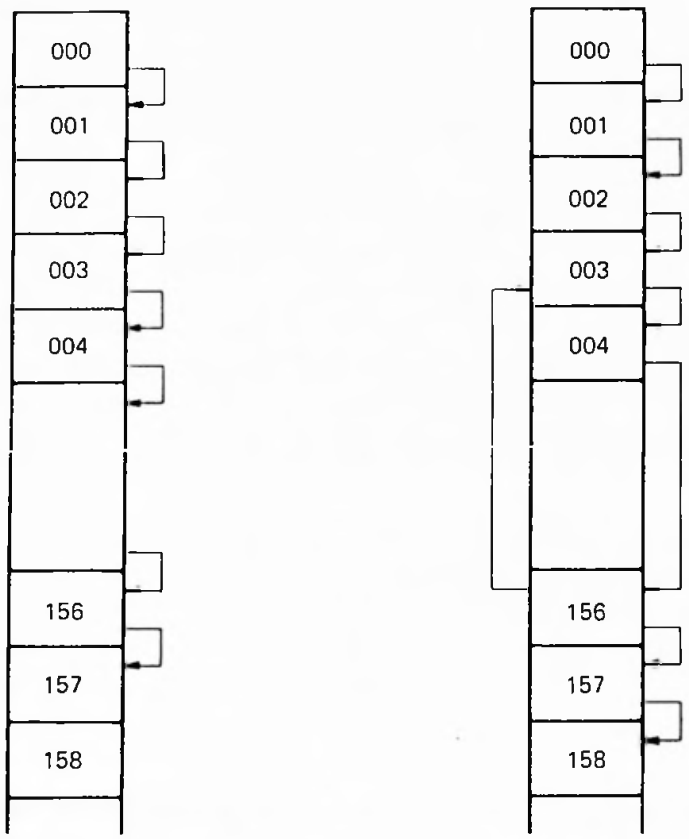
Sprungbefehle sind Befehle, die im Adreßteil an Stelle der Operationsadresse eine Befehlsadresse beinhalten, die den Befehlswortplatz angibt, an dem der nächstauszuführende Befehl gespeichert ist. Mit ihnen sind innerhalb des sequentiellen Programmablaufes Vorwärts- und Rückwärtssprünge möglich. Die Programmsprünge können von unbedingten und bedingten Sprungbefehlen ausgeführt werden.

Unbedingte Sprungbefehle bewirken, daß der Programmschrittschalter **unbedingt** auf die im Adreßteil dieser Befehle programmierte Sprung- (Befehls-) Adresse gesetzt wird. Bedingte Sprungbefehle bewirken das Setzen des Programmschrittschalters nur dann, wenn die diesen Befehlen zugeordneten Bedingungen erfüllt sind. Sprungbedingungen können

- Merker und
- Ergebnisse von Vergleichen sein.

Die Kriterien der Sprungbedingungen werden in separaten Befehlen, die jeweils vor den ihnen zugeordneten bedingten Sprungbefehlen programmiert werden müssen, angegeben.

4 Merker, die jeweils von 6 Steuertasten mit oder ohne Zifferntastenbestätigung gesetzt werden können, sowie die Vergleiche < 0 oder $\neq 0$ mit dem Register, dessen Inhalt verglichen werden soll, stehen als programmierbare Sprungkriterien zur Verfügung. Merker und Steuertasten können beliebig kombiniert programmiert werden. Vergleiche können in 8 Registern ausgeführt werden.



Sequentieller Programmablauf Zyklischer Programmablauf

Abb. 11 Veranschaulichung des sequentiellen und zyklischen Programmablaufs

Die Sprungbefehle ermöglichen auch das Programmieren von sogenannten „Programmschleifen"

Unter einer Programmschleife ist eine Befehlsfolge mit bedingten Sprungbefehlen zu verstehen, an deren Ende ein unbedingter Sprungbefehl an den Anfang zurückführt. Sie wird so lange durchlaufen, bis eine von den in der Schleife programmierten Bedingungen erfüllt ist und somit ein Sprung aus der Schleife erfolgt.

Beim Einschalten der Maschine werden grundsätzlich alle Schaltelemente in ihre definierten Anfangspositionen gesetzt.

Dazu gehört auch der Programmschrittschalter, der dabei auf 0 gesetzt wird und somit den ersten programmierten Befehl im Programm anspricht.

3.3 Rechner und Register

3.3.1 Der Rechner ist ein Schaltwerk, das u.a. aus einem Addierwerk mit Komplementierungs- und Schiebemöglichkeiten besteht. Die arithmetischen Operationen:

- Addition
- Subtraktion
- Multiplikationen

werden auf Additionen und Schiebevorgänge zurückgeführt.

Jeweils 4 Bit zweier Zeichen (bitparallel) werden dabei seriell (zeichenseriell) addiert.

Bei Multiplikationen kann das Produkt abgestrichen und gerundet werden.

Außer den rein arithmetischen Operationen werden mit dem Rechner auch Vergleiche ausgeführt. Hierbei wird ermittelt, ob der Inhalt eines bestimmten Registers

< 0 oder

$\neq 0$

ist. Das Ergebnis dieser Prüfungen kann für Programmverzweigungen ausgewertet werden.

Die Rechenzeiten betragen:

für Addition und Subtraktion	max. 0,7 ms
für Multiplikationen	~ 20 – 150 ms

3.3.2 (I) Die Modelle TA 10/1 und 2 verfügen über 10 Register, wovon 8 Register vom Programm angesprochen werden können. Es handelt sich hierbei um die 7 variablen Register (VR 1 - 7) die maximal 12 Ziffern + Vorzeichen speichern können und um den Akkumulator (BR), der wie die zwei übrigen Register maximal 16 Ziffern speichern kann.

Die übrigen zwei Register sind in der Maschine eingesetzt als:

- Tast-/Rechenregister (AR)
- Ausgaberegister (CR)

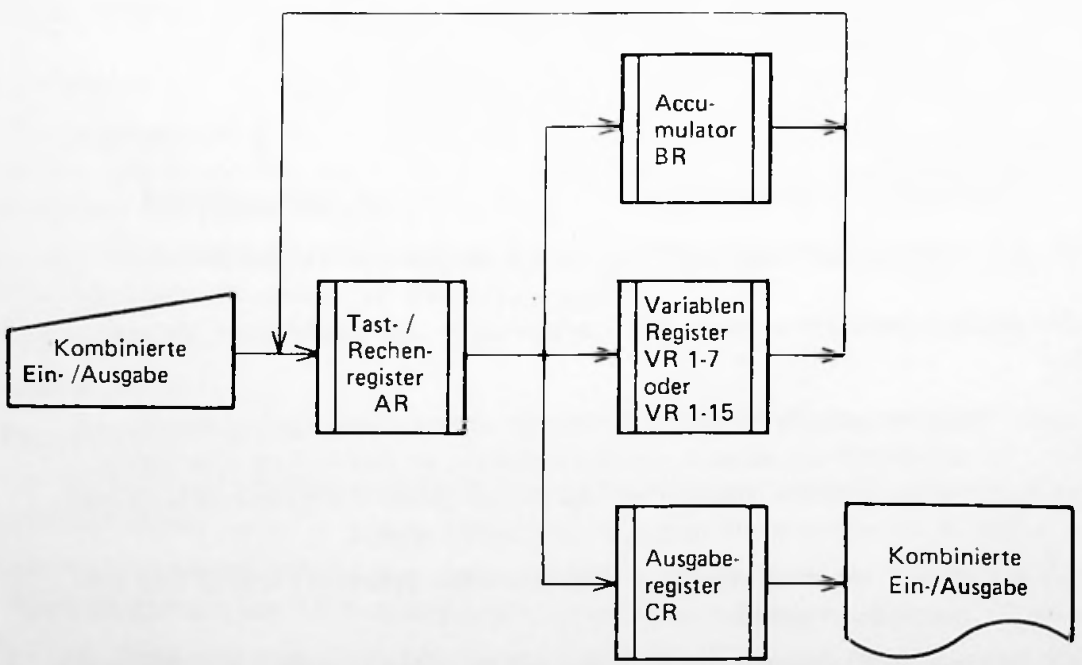


Abb. 12 Programmierbare Datenflußmöglichkeiten innerhalb der Register

Alle Register sind dynamisch eingesetzte MOS – (Metal-Oxide-Semiconductor) Schieberegister, in die die Informationen zeichenseriell eingeschrieben und ausgegeben werden.

(II) Das Modell TA 10/3 verfügt über insgesamt 18 Speicher, davon sind 15 frei verfügbare Speicher, 3 sind als Arbeitsregister vorgesehen. Die Speicherkapazität beträgt maximal 15 Ziffern + Vorzeichen.

Die programmierbaren Datenflußmöglichkeiten innerhalb der Register sind in Abb. 12 veranschaulicht.

Nähere Einzelheiten sind den Befehlsbeschreibungen zu entnehmen.

3.4 Wagensteuerung

Die Wagensteuerung führt alle Befehle aus, die die Ausgabe betreffen.

Von hier erfolgt die Steuerung der aus dem Ausgaberegister niederzuschreibenden (und/oder abzulochenden) Daten, Tabulationen und Wagenrückläufe.

Die jeweilige Position des Wagens wird durch einen Zähler gespeichert. Dieser Zähler verliert beim Ausschalten der Maschine seinen Wert und wird beim Wiedereinschalten stets auf die Position „0" gestellt. Beim Wiedereinschalten der Maschine fährt der Schreibwerkswagen generell in seine Grundstellung.

Grundstellung oder Wagenanfangsposition ist bei Wagengröße II die jeweils eingestellte, geradzahlige Randstellerposition und bei Wagengröße IV oder VI jeweils die Teilung 10. Teilung 10 bezieht sich hierbei nur auf die in dieser Wagenposition auf dem Teilungslineal abzulesende Positionsangabe.

Der Teilungszähler steht in dieser Position bei den Wagengrößen IV und VI auf Null und beginnt auch ab dieser Position von Null ab zu zählen.

Bei Wagengröße II steht der Zähler an der jeweils eingestellten geradzahligen Randstellerposition auf Null.

Wird bei eingeschalteter Maschine der Schreibwerkswagen von Hand bewegt, kann es z.B. vorkommen, daß Niederschriften nicht an den programmierten Schreibpositionen beginnen, sondern an Positionen, an die der Wagen von Hand geschoben wurde. Folgt jedoch nach dem Verschieben des Wagens ein Tabulationsbefehl, so wird die Synchronisation zwischen Zähler und Adresse wieder hergestellt.

3.4.1 Schreibwerk

Als Ausgabeeinheit wird ein Typenhebelschreibwerk verwendet, mit dem Niederschriften aus dem Ausgaberegister (im Arbeitsmodus Rechnen, Ziffern und Sonderzeichen) mit hohen Schreibgeschwindigkeiten erfolgen.

Je nach Wagengröße und Farbbandposition werden folgende maximalen Schreibgeschwindigkeiten (Anschläge pro Sekunde) erreicht:

Wagengröße II	14 - 18
Wagengröße IV	14
Wagengröße VI	10

Auf Wunsch kann auch Schwarz/Rot-Farbband geliefert werden, bei Rot-Ausdruck der negativen Werte.

Tabulationen werden mit 4-facher und Wagenrückläufe mit 11-facher Schreibgeschwindigkeit (ausgehend von 18 Anschlägen pro Sekunde) ausgeführt.

Die hohe Niederschriftgeschwindigkeit wird durch die doppelt belegten Typenhebel (in Abb. 3 die stärker umrandeten Typen) die wechselseitig abgerufen werden, erreicht.

Auf Befehl kann jede geradzahlige Druckposition erreicht werden. Wird eine Niederschrift in einer ungeradzahligen Teilung beendet, so kann anschließend mit einem Tabulationsbefehl in die nächstfolgende geradzahlige Teilung tabuliert werden.

Die Positionierung der Wagenbewegungen erfolgt dergestalt, daß ein metallisches Lochband fotoelektrisch abgetastet wird. (Abb. 13)

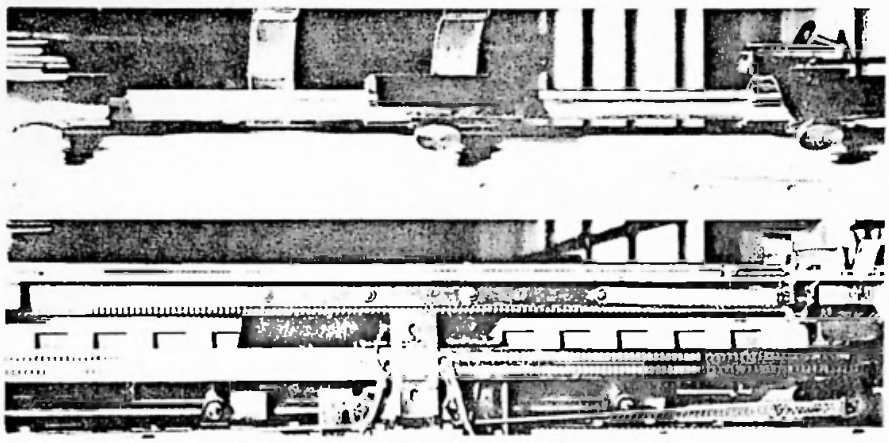


Abb. 13 Fotoelektrische Wagensteuerung

Je nachdem, ob sich der Wagen vorwärts oder rückwärts bewegt, löst jedes abgetastete Loch beim Teilungszähler in der Baugruppe „Wagensteuerung" eine Addition oder Subtraktion um „1" aus. Die so ermittelten geradzahligen Positionen des Wagens werden dabei ständig mit der programmierten Wagenadresse verglichen. Bei Übereinstimmung der beiden Positionen wird der Wagen in der adressierten Teilung gestoppt.

Ohne Leerschritt können maximal 15 Ziffern, das Minuszeichen und ein Symbol im Arbeitsmodus „Rechnen" geschrieben werden. Gelangt kein Minuszeichen und kein Symbol zur Ausgabe, so bleibt das Schreibwerk an der Schreibstelle stehen, an der sonst das Minuszeichen geschrieben wird.

Im Boden des Schreibwerkes sind ferner 4 Platten angeordnet, die folgende schreibwerksbezogene Elektronik beinhalten:

Platte 1: Tastaturcodierer, Magnetdecoder und Magnetverstärker (Schreibwerksplatte)

Platte 2: Entkopplungen für Druckmagnete

Platte 3: Haltestrombegrenzungen für die Druckmagnete

Platte 4: Fotoverstärker für Wagenstand

3.5 Alphasymbol — Codierer

Am Modell TA 10/2 und TA 10/3 ist in der Sondertastatur der Codierer für die Alphasymbolisierung eingebaut.

Werkseitig sind auf den Tastenknöpfen folgende Alphasymbole graviert

RE = Rechnung	KA = Kasse	VE = Verrechnung
ST = Storno	BA = Bank	SK = Scheck (Sparkasse, Skonto)
RT = Retouren	PS = Postscheck	WE = Wechsel

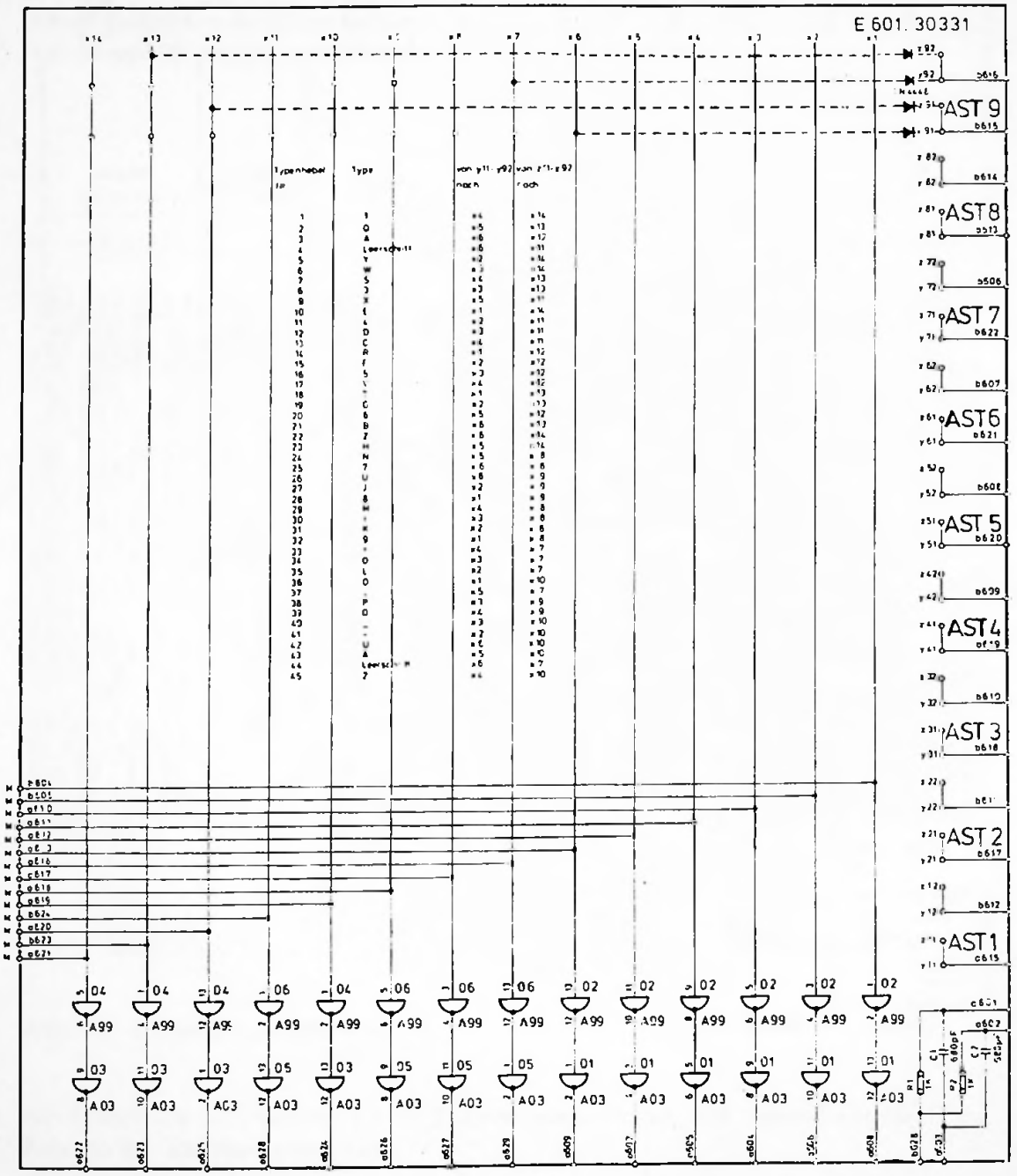
Andere zweistellige Schriftzeichenkombinationen können wie folgt eingestellt werden:

Soll z.B. die Buchstabenkombination AB eingestellt werden, so sind wie in der Tabelle Abb. 14 ersichtlich, die Diodenverbindungen x 6 und x 12 für den Buchstaben A und die Diodenverbindungen x 6 und x 13 für den Buchstaben B mit der Alphasymboltaste (AST 1-9) herzustellen, die diese Schriftzeichenkombination in Verbindung mit dem Programm auslösen soll. (In unserem Beispiel (Abb. 14) ist die Buchstabenkombination AB auf die Alphasymboltaste AST 9 rangiert).

Zu beachten ist, daß die jeweils auf die unteren Tastenkontakte (Stützpunkte b 615, b 613, b 622, b 621, b 620, b 619, b 618, b 617, und a 615) rangierten Schriftzeichen jeweils die 1. Zeichen der 2-stelligen Schriftzeichenkombination sind.

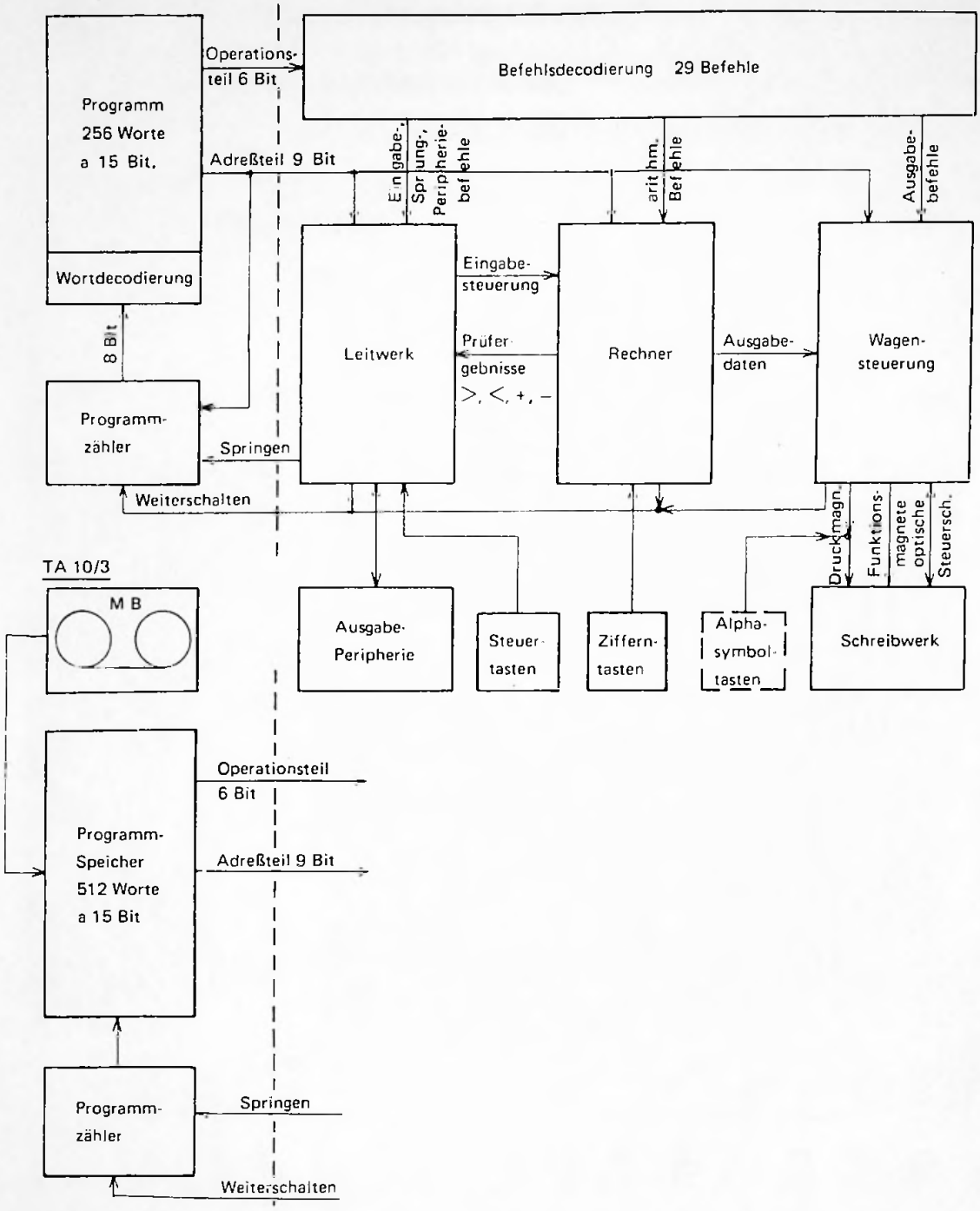
Abb. 14 Tab. Codierungsmöglichkeiten auf der Codierplatte (Sondertastatur)

Es sind auch andere Typen und Typenhebel-Codierungen möglich



Außer dem Codierer beinhaltet die Sondertastatur 1 Schaltplatte, auf der sich der (Sonder-) Tastaturcodierer befindet.

3.6 Blockschaltbild TA 10



4. PROGRAMMIERTECHNIKEN

4.1 Mit Dioden gelötete Programmplatten (TA 10/1 und TA 10/2)

Die Standardprogrammplatte (Abb. 15) beinhaltet 128 Befehlswortplätze, die in zwei Feldern mit je 64 Befehlswortplätzen auf der Platte aufgeteilt sind. Die Befehlswortplätze sind jeweils 14 Diodenplätze in die die programmierten Befehlsbitmuster mit Dioden eingelötet sind.

Im Mittelstreifen der Programmplatte befindet sich in IC-Bestückung die Decodierung der 1 aus 128 Befehlswortplätzen.

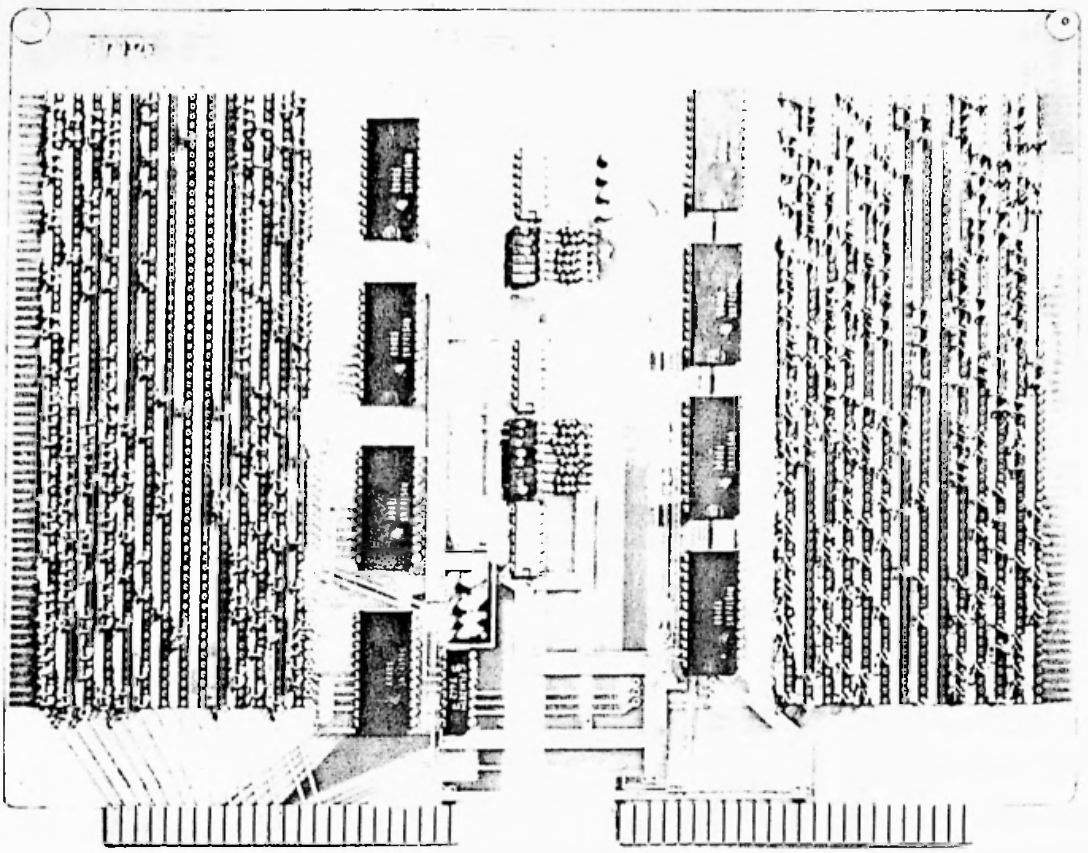


Abb. 15 Diodenprogrammplatte

Am Modell TA 10/1 stecken 1 oder 2 Diodenplatten (max. 256 Befehlswortplätze) im Rahmen der Verarbeitungseinheit.

Sie sind der Bedienungskraft nicht zugängig.

Am Modell TA 10/2 stecken diese Platten in einer handlichen Kassette, die mit einem an der linken Tischvorderseite angeordneten verschließbaren Einschub (Abb. 6) steckbar angeschlossen wird. Die Kassette kann von der **Bedienungskraft leicht ausgewechselt** werden. Statt dessen ist auch 1 PROM-Platte mit 256 oder 512 Befehlen einsetzbar.

Am Modell TA 10/3 werden die Programme in Endlos-Magnetbandkassetten gespeichert (Abb. 8) und je nach Bedarf in das MB-Kassettengerät eingelegt und in die Grundmaschine eingelesen. Auch hier ist der **MB-Kassettenwechsel leicht vorzunehmen.**

4.2 Stecktafeln (Abb. 16)

An die Modelle TA 10/1 und TA 10/2 können als Programmierungshilfe zwei Diodenstecktafeln, die je 128 Befehlswortplätze beinhalten, angeschlossen werden. In diesen Tafeln werden die Befehlsbitmuster ohne Lötarbeiten mit Steckdioden gesetzt. Sie können leicht verändert werden und sind somit ein ideales Hilfsmittel für das Testen der Programme.

Angeschlossen werden die Tafeln an die Modelle TA 10/1 und TA 10/2 über eine Adapterplatte, die bei beiden Modellen im Plattenrahmen an dem Platz 5 zu stecken ist.

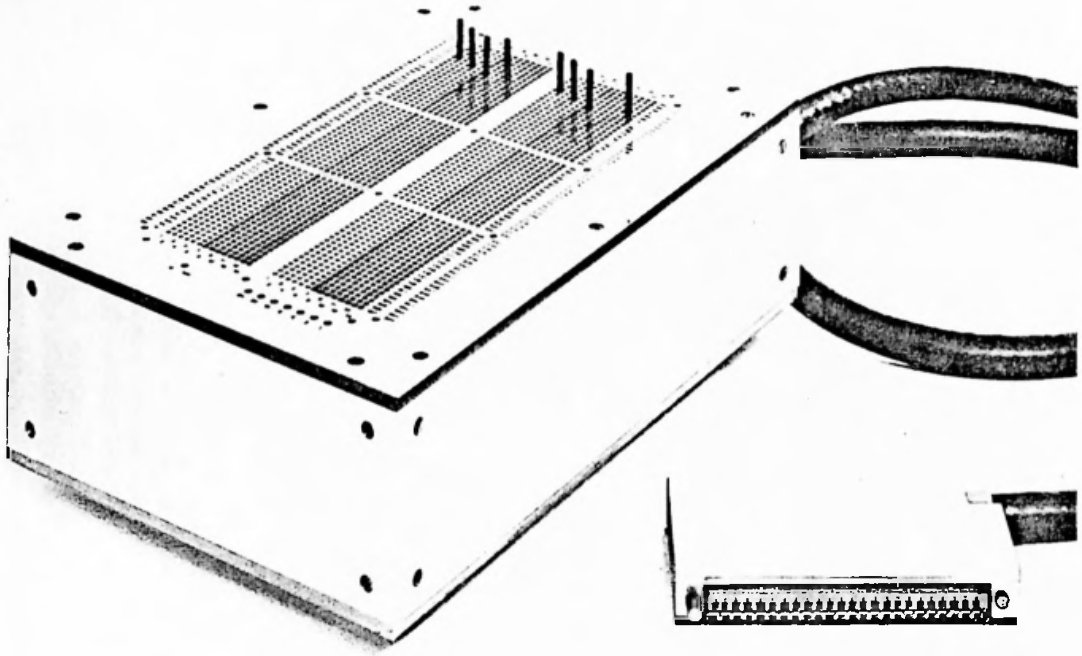


Abb. 16 Diodenstecktafel

4.3 Programmplattenprüfgerät

Als weiteres Hilfsmittel für eine rationelle Programmherstellung steht ein Programmplattenprüfgerät Abb. 17 zur Verfügung.

Mit diesem Gerät werden mit Dioden gelötete Programmplatten auf Fehler geprüft, die bei der Bestückung der Platten auftreten können. Derartige Fehler können falsch oder unsachgemäß eingelötete Dioden, defekte Dioden oder Verbindungen in den Plattenkaschierungen sein.

Geprüft wird die Platte durch einen Vergleich mit einer bereits geprüften mit Dioden gelöteten Programmplatte gleichen Programmes, oder mit der Stecktafel auf der das getestete, gleiche Programm noch gesteckt ist.

Dazu werden sowohl die zu testende als auch die bereits getestete Programmplatte oder Stecktafel an das Programmplattenprüfgerät gesteckt.

Stehen für den Vergleich weder eine bereits geprüfte mit Dioden gelötete Programmplatte noch eine Stecktafel zur Verfügung, so muß er visuell durchgeführt werden.

Für diese beiden Prüfmöglichkeiten sind an dem Gerät 2 unterschiedliche Arbeitsmodi einstellbar.

– manuelle Programmschrittweiterschaltung und

– automatische Programmschrittweiterschaltung.

Manuelle Programmschrittweiterschaltung wird betrieben, wenn die zu prüfende Programmplatte visuell mit den Befehlsblättern verglichen werden muß. Dabei werden die einzelnen Programmschritte sowie die in den einzelnen Programmschritten programmierten Befehlsbitmuster am Prüfgerät mit Lampen angezeigt, die mit den Befehlsblättern verglichen werden müssen.

Automatische Programmschrittschaltung wird eingestellt, wenn die zu prüfende Programmplatte mit einer bereits geprüften Programmplatte oder mit den Stecktafeln verglichen werden soll.

Dieser Vergleichsvorgang bzw. dessen Programmschrittweiterschaltung erfolgt automatisch. Er wird nur unterbrochen, wenn das Gerät auf einen der bereits genannten Fehler stößt.

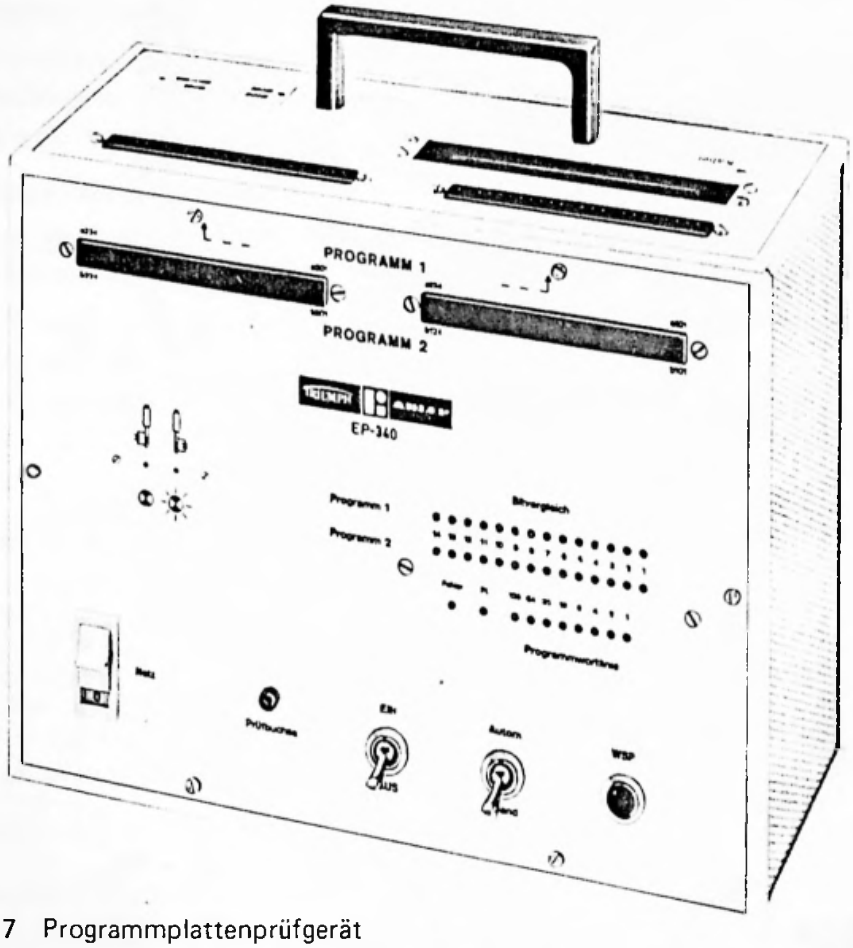


Abb. 17 Programmplattenprüfgerät

Wird beim Vergleichsvorgang ein Fehler erkannt, so können am Prüfgerät der Programmschritt, in dem der Fehler aufgetreten ist und die in diesen Programmschritten programmierten Befehlsbitmuster an Lampen abgelesen werden.

Bei der Programmierung von PROM-Bausteinen (Programmable-Read-Only-Memories = PROM) werden diese über ein Programmiergerät PET 3 eingeschrieben. (Vergl. dazu auch „Bedienungshandbuch für PET 3")

Es sind dabei jeweils 2 Bausteine angeschlossen, wobei die erste Hälfte des Programmwortes in dem einen, die andere Hälfte im anderen Baustein untergebracht wird.
2 PROM-Bausteine nehmen max. 256, 4 PROM-Bausteine max. 512 Befehle auf. Die Platte mit den PROM-Bausteinen ist auf Platz 5 oder Verarbeitungseinheit untergebracht. (siehe 2.1.2).

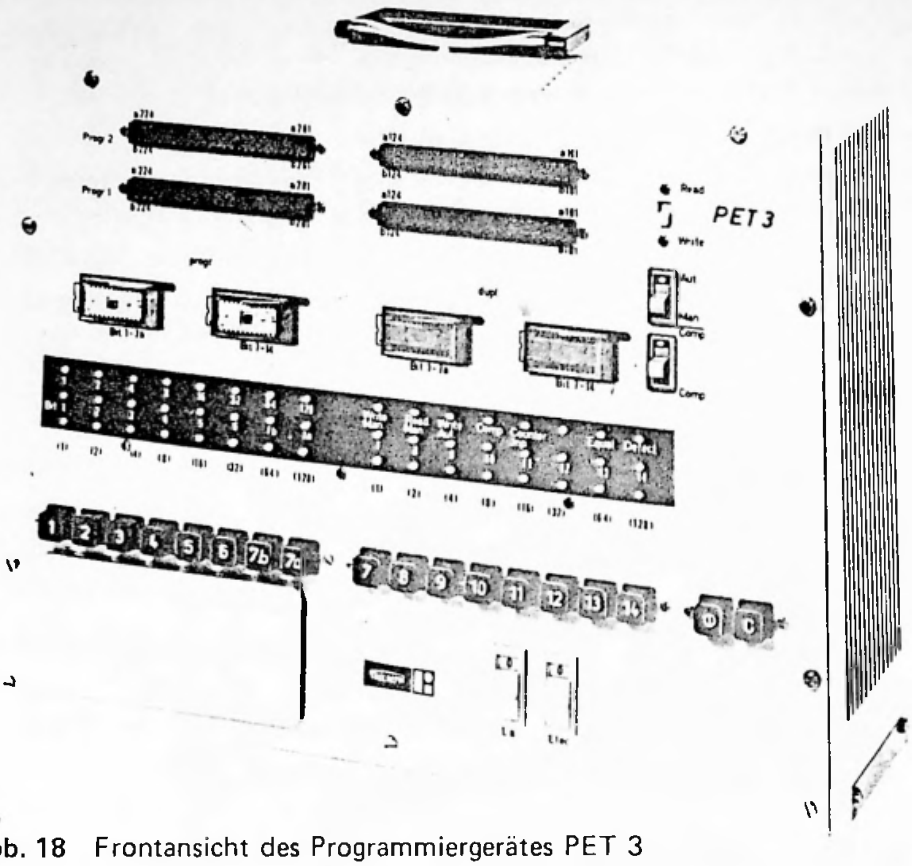


Abb. 18 Frontansicht des Programmiergerätes PET 3

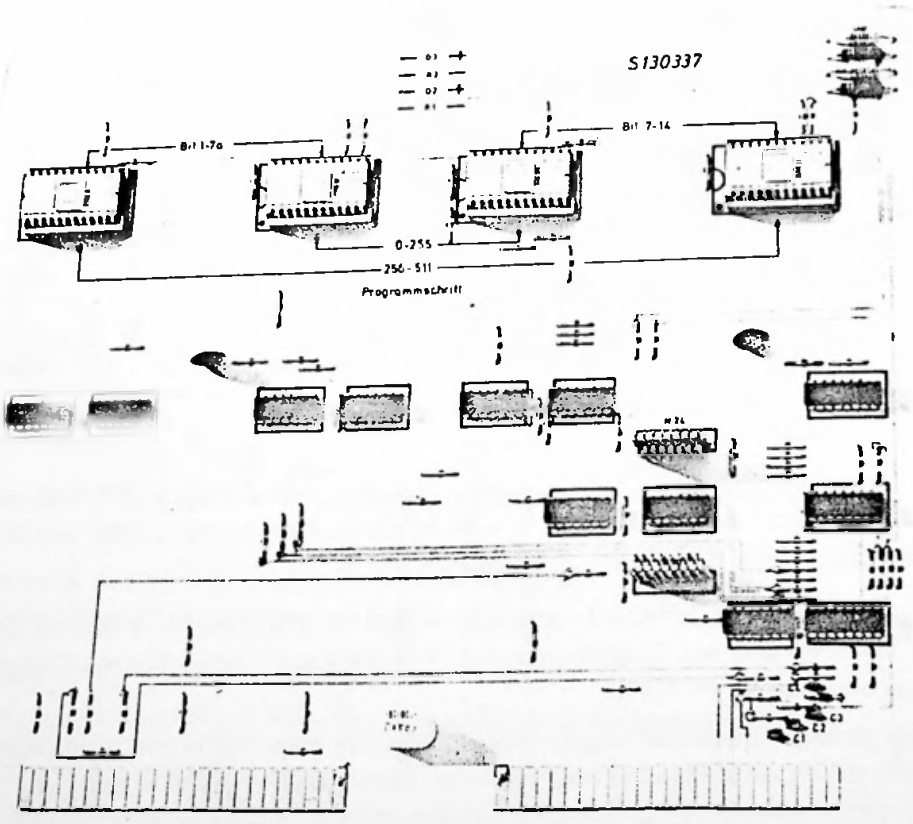


Abb. 19 Steckplatte mit 4 PROM-Bausteinen

4.4 Arbeitsroutine für die Erstellung eines TA 10/3-Programms (s. Abb. 9)

a) Programmeingabe über die Programmeingabe-Tastatur

– Grundmaschine und Programmspeicher einschalten.

(Einschalten gemeinsam – EIN/AUS-Schalter der Grundmaschine)
Mit dem Einschalten der beiden Aggregate wird dem Programmspeicher ein Rückstellimpuls übertragen. Der Programmschrittzähler steht auf „0".

– Betätigen der Betriebsarttaste „EI"

Betriebsart: – Programmeingabe (Tastatur) nach MOS-Speicher ist eingeschaltet.

– Betätigen der Auslösetaste „LP"

Gesamtlampenfeld wird auf Ausfälle geprüft.

– Bitmuster-Eingabe

An Hand des Programmblattes wird über die Befehlstasten 1 bis 14 das Bit-Muster des Befehlswortes „0" eingetastet. Das eingetastete Bit-Muster wird über die Lampenreihe (rot/gelb) 1 bis 14 angezeigt.

– Betätigen der Auslösetaste „BÜ"

Durch die Funktion der BÜ-Taste (Befehls-Übernahme) wird das anstehende Befehlswort in den MOS-Speicher übernommen/eingespeichert. (Hier Programmschritt „0").

Der Programmschrittzähler (obere Lampenreihe) schaltet auf den ersten Programmschritt und zeigt diesen optisch an.

Die Lampenreihe darunter (rot/gelb) erlischt. Das Bit-Muster des Befehlswortes „1" kann darauf eingetastet und der Arbeitsablauf „Eingeben – Befehl übernehmen" fortgesetzt werden.

– Programmende-Befehl (s. Befehl „END" S. 112)

▶ falls erforderlich

– RS-Taste (Programmschrittzähler auf „0" zurücksetzen) betätigen und evtl. eine Überprüfung des zuvor eingetasteten Programms auf Richtigkeit durchführen.

– Hierzu ist die BL-Taste nacheinander zu betätigen, so daß die einzelnen Befehlsworte, von „0" beginnend, wortweise gelesen werden können.

– Anschließend RS-Taste betätigen – Programm ist für die Übernahme auf MB bereit.

b) Übernahme des Programms auf Magnetband

– Drehschalter am MB-Kassettengerät in Schaltstellung „Kassettenwechsel" (CASS") schalten. Kassettenschacht springt auf.

– MB-Kassette in den geöffneten Schacht einschieben und Kassettenschacht durch vertikalen Druck schließen.

– Drehschalter in Vertikalstellung „START ▶" schalten.
MB-Kassettengerät ist vorbereitet (s. Abb. 9).

– Betätigen der Betriebsart - Taste „MBA"

Betriebsart: – Übernahme MOS-Speicher nach MB ist eingeschaltet.

– Taste – START betätigen.

– Auslösen der Taste START (gelbe Leuchtdrucktaste) am MB-Gerät – Programmaufzeichnung auf MB läuft, angezeigt durch das signalisierende Lampenfeld der Programmeingabe-Tastatur.

Ist der Befehl „Programmende" erkannt und magnet. aufgezeichnet, erfolgt automatische Stoppung des Magnetbandes. Eine sichere MB-Programmspeicherung ist erst dann gewährleistet, wenn der Aufzeichnungsvorgang eines Programmes mehrmals wiederholt wird (dazu Taste START – eine Zeitdauer siehe Tab. – betätigt lassen).
Die MB-Endloskassette besitzt eine Standardbandlänge von $9 \pm 0{,}4$ m, die dazugehörige Aufzeichnungs-Durchlaufzeit sowie die mögliche Anzahl der Aufzeichnungen ist wie folgt aus untenstehender Tabelle zu entnehmen.

Tabelle

Befehls-Anzahl	Bandlänge	mögliche Anzahl der Aufzeichnungen	MB-Aufzeichgs-Durchlaufzeit
256	9,4 m	12 x	3,3 Min
512		6 x	
256	8,6 m	11 x	3 Min
512		5,5 x	

c) Korrekturen bei fehlerhafter Programmeingabe

– **Korrekturen** können **grundsätzlich** nur **vor der Befehlsübernahme** durchgeführt werden.

–1– Wurde, während der Programmeingabe vor dem Betätigen der BÜ-Taste (Befehls-Übernahme-Taste) ein falsch eingetastetes Bitmuster bemerkt, so ist die C-Taste zu drücken, die das Bitmuster löscht, und das neue Befehlswort einzugeben.
Anschließend mit der BÜ-Taste das Befehlswort in den MOS-Speicher übernehmen und Arbeitsablauf fortsetzen.

–2– Wurde der Fehler erst nach dem Betätigen der BÜ-Taste oder mitten im Programm erkannt, so wird mit der RS-Taste der Programmschrittzähler auf „0" zurückgesetzt und das fehlerhafte Bitmuster erneut angewählt. Mit der AÜ-Taste (Adreß-Übernahme) ist das fehlerhafte Bitmuster im Programmschrittzähler zu kontrollieren und mit der C-Taste zu löschen. Anschließend ist das richtige Befehlswort einzugeben und mit der BÜ-Taste zu übernehmen.

Mit der BL-Taste (Befehl Lesen) können jetzt fortlaufend alle folgenden Befehlsworte optisch gelesen oder der Programmschrittzähler mit der RS-Taste auf „0" zurückgesetzt werden.

Das fertige Programm kann nun auf Magnetband (s. Pkt. 4.5b) aufgezeichnet bzw. der Arbeitsablauf fortgesetzt werden.

–3– Zum Löschen einzelner oder hintereinanderfolgender Befehlsworte ist ebenfalls nacheinander die C-Taste, danach jeweilig die BÜ-Taste zu betätigen. Die Programmschritte in der Programmschrittzähler-Anzeige (Lampenreihe oben) bleiben dabei erhalten.

5. PROGRAMMFORMALISMUS

5.1 Programmablaufplan

Bei der Erstellung eines Programmes ist es von Vorteil, ein gegebenes Problem zunächst in einer Form zu notieren, die den Ablauf des Programmes klar erkenntlich wiedergibt.

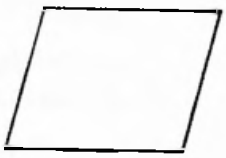
Dazu eignen sich besonders gut Programmablaufpläne.

Der Programmablaufplan nimmt eine Mittelstellung zwischen Datenflußplan und Befehlsdiagramm ein. Er zeigt den genauen Verlauf des Programmes, insbesondere hinsichtlich der im Programm enthaltenen Verzweigungen.

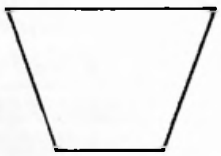
Sinnbilder für Datenfluß- und Programmablaufpläne sind in der DIN 66001, Ausgabe Oktober 1969 festgelegt und können schnell mit der dort empfohlenen Schablone gezeichnet werden.

Für das Erstellen eines Programmablaufplanes genügt im allgemeinen die Kenntnis einiger weniger Sinnbilder:

Grenzstelle, z.B. Start, Stop

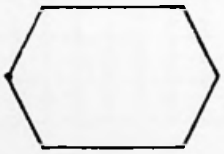


Eingabe, Ausgabe,
ob es sich um maschinelle oder manuelle Eingabe oder Ausgabe handelt, soll aus der Beschriftung des Sinnbildes hervorgehen, z.B. Eintastungen, Niederschriften usw.

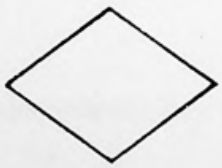


Operationen von Hand
z.B. Formularwechsel

Operationen allgemein
z.B. Rechnen, Transfer, Speichern



Programmodifikation
z.B. Setzen von Merkern



Verzweigung

Übergangsstelle
Der Übergang kann von mehreren Stellen aus, aber nur zu einer Stelle hin erfolgen.

Unterprogramm
Es können mehrere Ein- und Ausgänge vorhanden sein.

Start und Stop sind Beginn und Ende eines Programmablaufplanes. Dazwischen liegen in der Reihenfolge der Bearbeitung alle durchzuführenden Operationen. Pfeile geben die Verkettung der einzelnen Operationen an, deren Typus an den festgelegten Sinnbildern erkenntlich ist. Formeln, Befehle oder Text in den Sinnbildern spezifizieren die Operationen. Abfrage und Entscheidungsoperationen sind Weichen für die Verzweigung des Programmablaufes. Damit werden auch die sogenannten Programmschleifen dargestellt.

Ist für ein gegebenes Problem in der genannten Weise der Programmablaufplan erstellt, so können danach leicht die noch benötigten Befehlsblätter ausgearbeitet werden.

5.2 Befehlsblätter

Die Befehlsblätter (Abb. 19) sind so zu erstellen, daß die Reihenfolge der in ihnen eingetragenen Befehle der Reihenfolge der Befehlsausführung bzw. des Programmablaufes entsprechen.

Das Bit-Muster der einzelnen Befehle ist entweder aus der Codeliste (s. 8.) oder aus den einzelnen Befehlsbeschreibungen in die Befehlsblätter zu übertragen.

Die in den Unterlagen dargestellte Binärstelle 7a ist für die Programmierung der Diodenplatten bedeutungslos. Sie ist nur zu programmieren, wenn über 256 Programmschritte benötigt werden. (MB, PROM)

Abb. 20 Befehlsblatt

Programm:					Blatt - Nr.													
Erläuterung	Programm Schritt	Befehl	Adresse	Bitbelegung / Programmspeicher 1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
	0																	
	1																	
	2																	
	3																	
	4																	
	5																	
	6																	
	7																	
	8																	
	9																	
	0																	
	1																	
	2																	

Auf den Programmplatten werden die in den Befehlsblättern eingetragenen Befehlsbitmuster mit Dioden gesetzt.

6. TA 10 – BEFEHLSLISTE (DEUTSCH UND ENGLISCH)

Befehlsname (deutsch)	COMMAND CODE	Mnemonics
Transfer	Transfer to variable register	TRV
Abgriff	Bring from variable register	BRV
Addition	ADD	ADD
Multiplikation	Multiply	MUL
Warte auf Taste	Wait for key	WAK
Setze Merker M bei Taste I	Mark if key	MAK
Setze Merker Mv bei Wertprüfung	Mark by value-test	MAT
Unbedingter Sprung	Jump unconditional	JUN
Sprung bei Merker 1	Jump if mark 1	JM1
Sprung bei Merker 2	Jump if mark 2	JM2
Sprung bei Merker 3	Jump if mark 3	JM3
Sprung bei Merker 4	Jump if mark 4	JM4
Sprung bei WP < 0	Jump if negative	JNE
Sprung bei WP $\neq 0$	Jump if not zero	JNZ
Sprung bei Selektor	Jump if selector	JSE
Leerbefehl	No operation	NOP
Tabuliere	Tabulate (following duplication or not)	)t)TAB
Tabuliere, Tastatur frei	Tabulate, free keyboard	TAK
Ausgabe vorbereiten	Output preparation	OUP
Ausgabe	Output	OUT
Ausgabe, Tastatur frei	Output, free keyboard	OUK
Numerischer Festwert	Constant	CON
Formularvorschub	Skip for form	SKP
Programmende	END	END
Lochen ohne Kennzeichen	Punch with 0 sign	PUO
Lochen mit 1-stell. Kennzeichen	Punch with 1 sign	PU1
Lochen mit 2-stell. Kennzeichen	Punch with 2 signs	PU2
Springen (Ausgabeperipherie	Skip on record medium	SK1
Blocklücke	Record gap	GAP
Lochen variables Kennzeichen	Punch constant	PUC
Speicherdefinitionen		
Tast-/Rechnen-Register	Key in/Calculating register	AR
Accumulator-Register	Accumulator register	BR
Ausgabe-Register	Output register	CR
Variablen-Register	Variable register	VR

6.1 Transfer to variable register

(I)

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
X			X				–	vr 7	vr 6	vr 5	vr 4	vr 3	vr 2	vr 1

Symbolischer Befehlscode: TRV/–/BR–VR 7

Wirkung

Aus dem AR positiver oder negativer Transfer nach VR 1-7 oder BR. Im VR 1-7 oder BR gespeicherte Werte werden gleichzeitig gelöscht. Der im AR stehende Wert bleibt erhalten.

Die VR 1-7 werden uncodiert im Adreßteil des Befehles programmiert. Ist kein VR im Adreßteil programmiert, so wird automatisch der BR angeschlossen. Soll der Wert im AR mit umgekehrten Vorzeichen nach VR 1-7 oder BR übertragen werden, so muß im Adreßteil Bit 7 programmiert werden.

Mit einem Befehl kann ein Transfer maximal in alle 7 VR erfolgen.

(II)

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
X			X				–				vr 8	vr 4	vr 2	vr 1

Symbolischer Befehlscode: TRV–/BR–VR 15

Wirkung

Aus dem AR positiver oder negativer Transfer nach VR 1-15 oder BR: Im VR 1-15 oder BR gespeicherte Werte werden gleichzeitig gelöscht. Der im AR stehende Wert bleibt erhalten.

Die VR 1-15 werden codiert im Adreßteil des Befehls programmiert. Ist kein VR im Adreßteil programmiert, so wird automatisch der BR angeschlossen. Soll der Wert im AR mit umgekehrten Vorzeichen nach VR 1-15 oder BR übertragen werden, so muß im Adreßteil Bit 7 programmiert werden.

Mit einem Befehl kann jeweils ein Transfer in ein VR erfolgen.

TRV

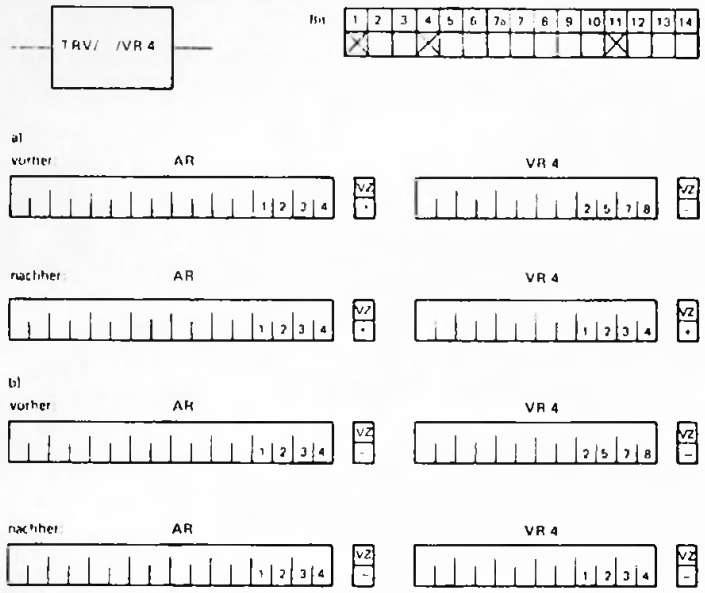
Beispiele

(I)

1. Transfer nach VR 4

a) AR beinhaltet einen positiven Wert

b) AR beinhaltet einen negativen Wert

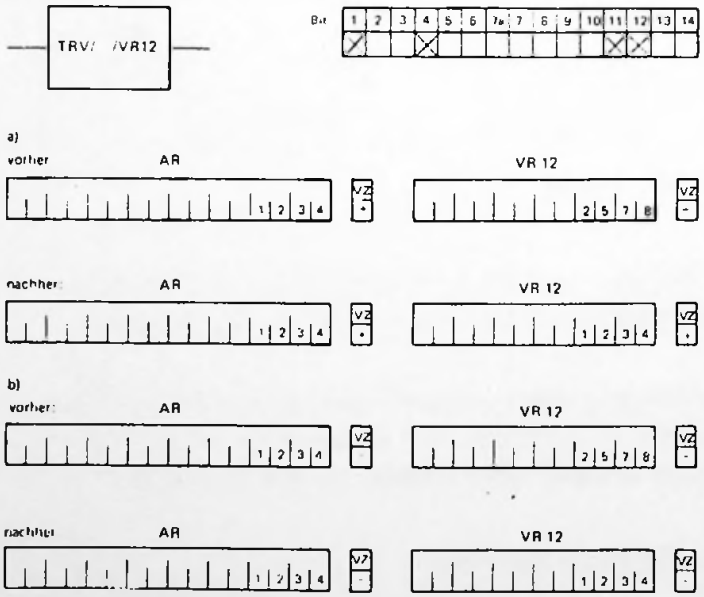


(II)

2. Transfer nach VR 12

a) AR beinhaltet einen positiven Wert

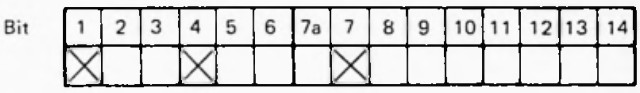
b) AR beinhaltet einen negativen Wert



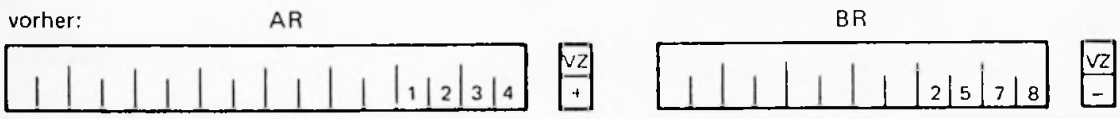
2. Transfer nach BR mit umgekehrten Vorzeichen

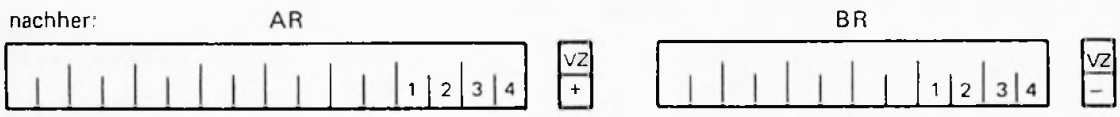
a) AR beinhaltet einen positiven Wert

b) AR beinhaltet einen negativen Wert

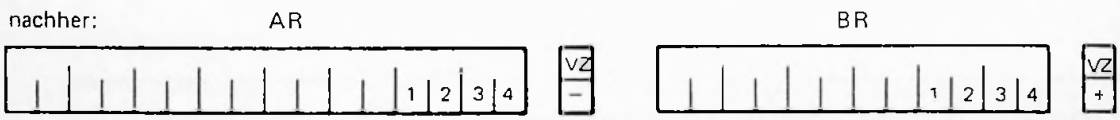


a)





b)



BRV

6.2 Bring from variable register

(I)

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
X							*	vr 7	vr 6	vr 5	vr 4	vr 3	vr 2	vr 1

Symbolischer Befehlscode: BRV/*/BR – VR 7

Wirkung:

Aus VR 1-7 oder BR wird der Wert nach dem AR übertragen.

Die entsprechenden VR werden im Adreßteil des Befehles uncodiert programmiert. Ist kein VR programmiert, so wird automatisch der Inhalt des BR abgegriffen.

Ein im AR stehender Wert wird bei der Übertragung gelöscht.

Soll ein oder mehrere VR mit Endsumme (*) abgegriffen werden, d.h. das oder die Register sollen nach dem Abgriff gelöscht sein, so ist Bit 7 im Adreßteil zu programmieren.

Mit einem Befehl können maximal alle 7 VR gelöscht werden.

(II)

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
X							*				vr 8	vr 4	vr 2	vr 1

Symbolischer Befehlscode: BRV/*/BR – VR 15

Wirkung

Aus VR 1-15 oder BR wird der Wert nach dem AR übertragen.

Die entsprechenden VR werden im Adreßteil des Befehls codiert programmiert. Ist kein VR programmiert, so wird automatisch der Inhalt des BR abgegriffen.

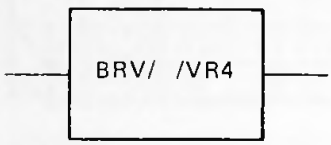
Ein im AR stehender Wert wird bei der Übertragung gelöscht.

Soll ein VR mit Endsumme (*) abgegriffen werden, d.h., das oder die Register sollen nach dem Abgriff gelöscht sein, so ist Bit 7 im Adreßteil zu programmieren.

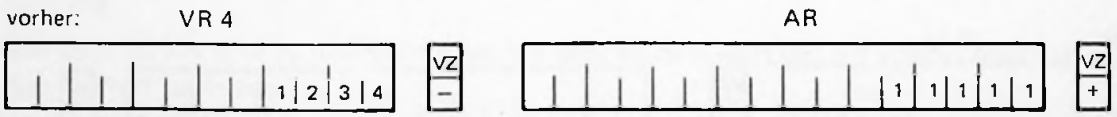
Mit einem Befehl kann jeweils nur ein VR gelöscht werden.

Beispiele

1. Abgriff des VR 4 mit Zwischensumme



2. Abgriff des VR 4 mit Endsumme

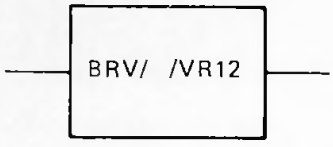


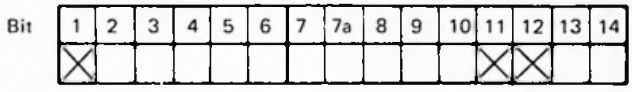
BRV

Beispiele

(II)

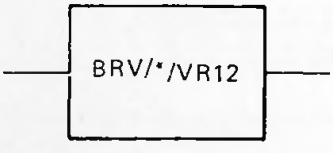
1. Abgriff des VR 12 mit Zwischensumme

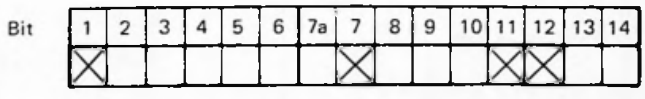


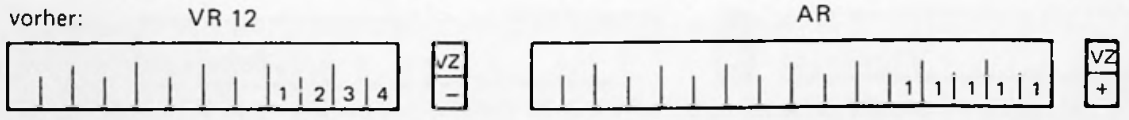


(II)

2. Abgriff des VR 12 mit Endsumme







6.3 ADD

(I)

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
X	X						–	vr 7	vr 6	vr 5	vr 4	vr 3	vr 2	vr 1

Symbolischer Befehlscode: ADD/–/BR –VR 7

Wirkung

Der Wert im AR wird zu dem Wert eines VR (oder mehrere VR) oder zu dem Wert des BR addiert oder von diesem Wert subtrahiert.

In einem Befehl können maximal 7 VR im Adreßteil uncodiert programmiert werden. Ist kein VR im Adreßteil programmiert, so wird automatisch in das BR addiert, bzw. subtrahiert.

Der Wert im AR bleibt erhalten.

Für Subtraktionen ist im Adreßteil zusätzlich Bit 7 zu programmieren.

Die Vorzeichen des im AR stehenden Wertes und des ADD-Befehles werden arithmetisch verknüpft.

(II)

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
X	X						–				vr 8	vr 4	vr 2	vr 1

Symbolischer Befehlscode: ADD/–/BR –VR 15

Wirkung

Der Wert im AR wird zu dem Wert eines VR oder zu dem Wert des BR addiert oder von diesem Wert subtrahiert.

In einem Befehl kann jeweils nur ein VR im Adreßteil codiert programmiert werden. Ist kein VR im Adreßteil programmiert, so wird automatisch in das BR addiert bzw. subtrahiert. Der Wert im AR bleibt erhalten.

Für Subtraktionen ist im Adreßteil zusätzlich Bit 7 zu programmieren. Die Vorzeichen des im AR stehenden Wertes und des ADD-Befehls werden arithmetisch verknüpft.

ADD

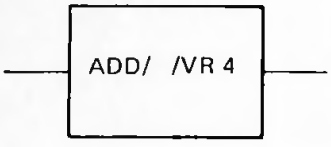
Beispiele:

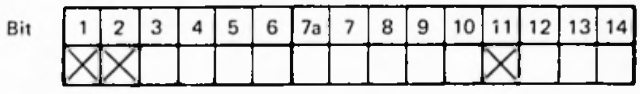
(I)

1. Addition nach VR 4

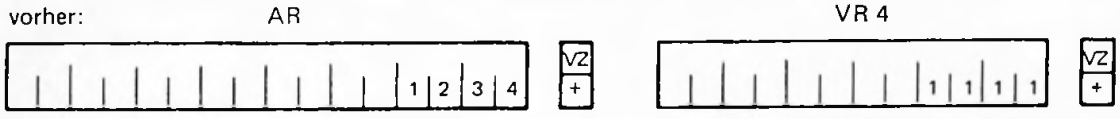
a) AR und VR 4 beinhalten positive Werte

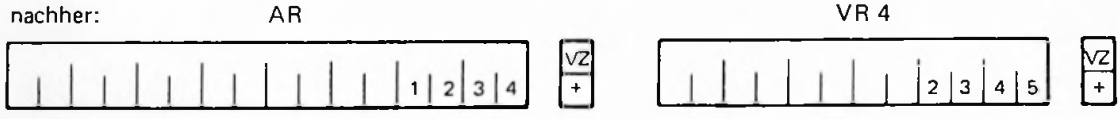
b + c) AR beinhaltet einen negativen Wert und VR 4 beinhaltet einen positiven Wert.



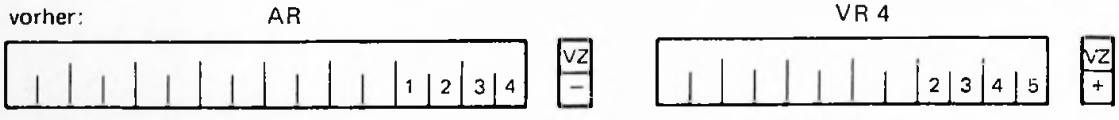


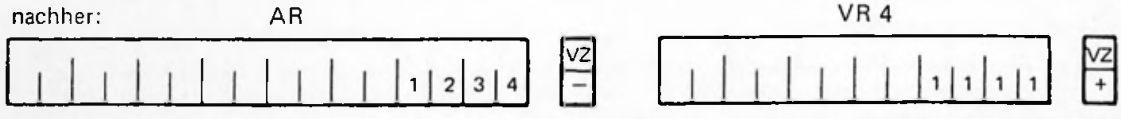
a)



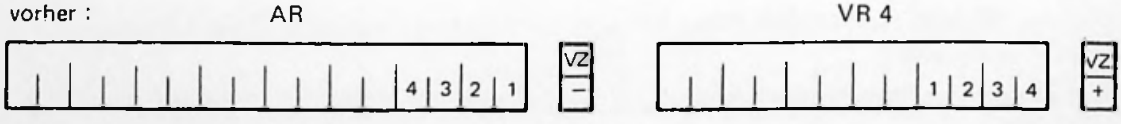


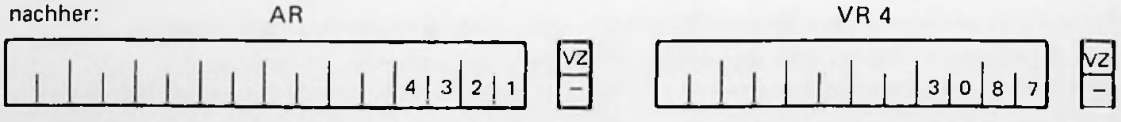
b)





c)





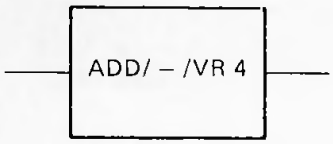
2. Subtraktion nach VR 4

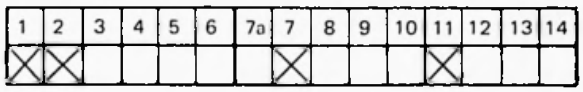
(I)

a) AR und VR 4 beinhalten positive Werte.

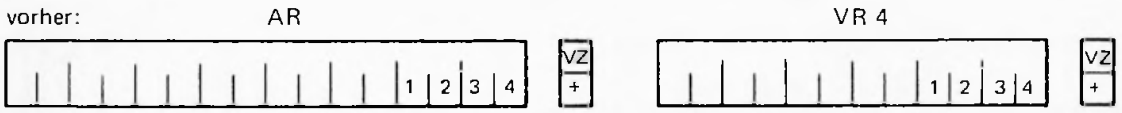
b) AR beinhaltet einen positiven Wert und VR 4 beinhaltet einen negativen Wert.

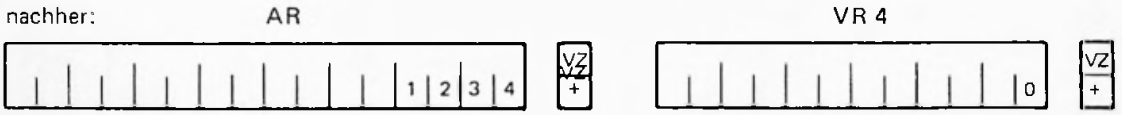
c) AR und VR 4 beinhalten negative Werte.



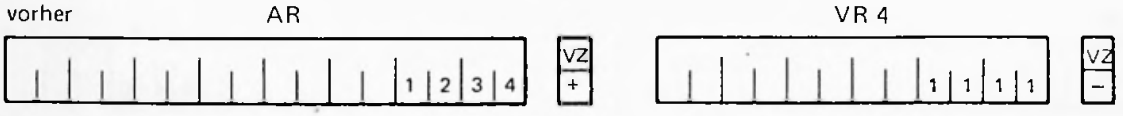


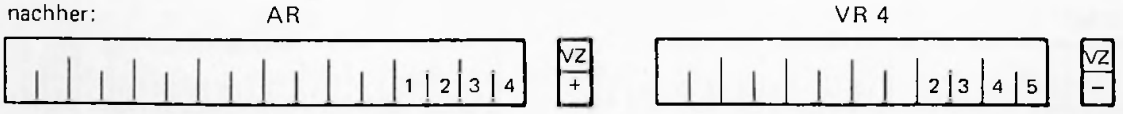
a)



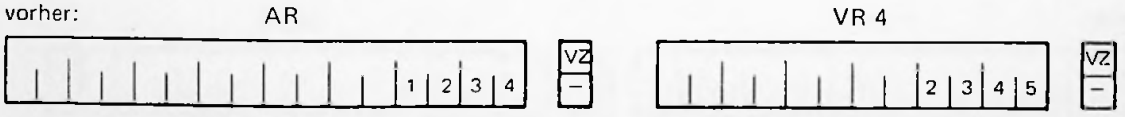


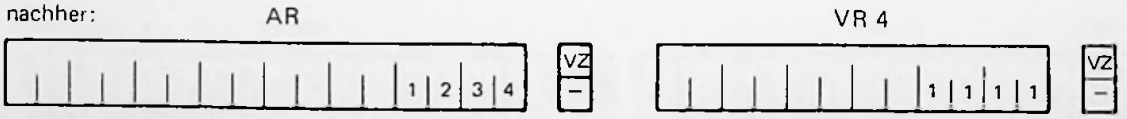
b)





c)





ADD

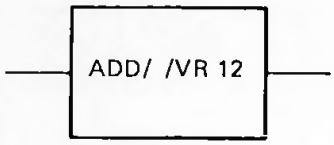
Beispiele:

(II)

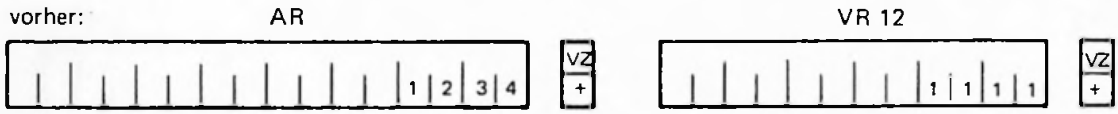
1. Addition nach VR 12

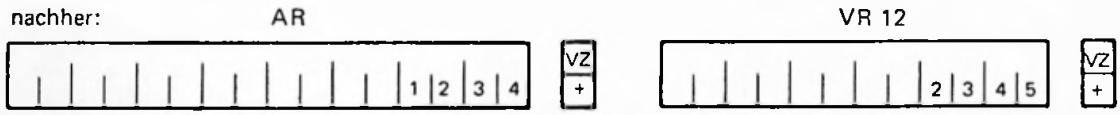
a) AR und VR 12 beinhalten positive Werte

b + c) AR beinhaltet einen negativen Wert und VR 12 beinhaltet einen positiven Wert.

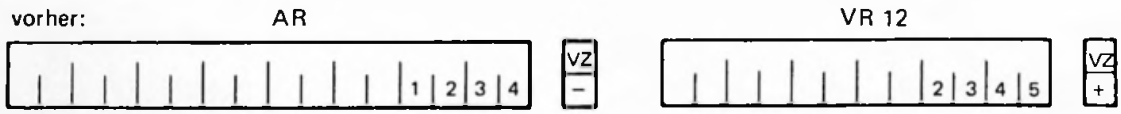


a)

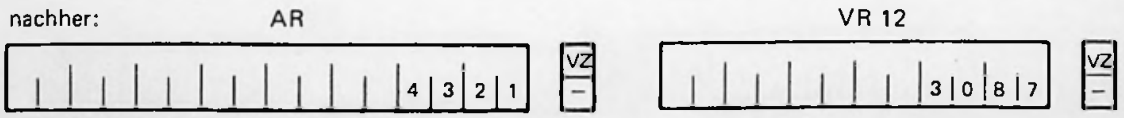




b)



c)



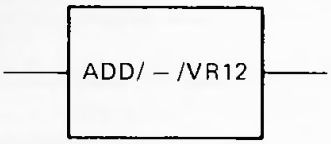
2. Subtraktion nach VR 12

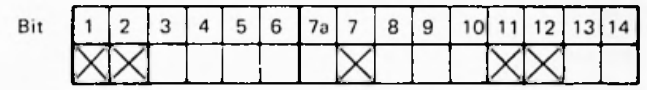
(II)

a) AR und VR 12 beinhalten positive Werte

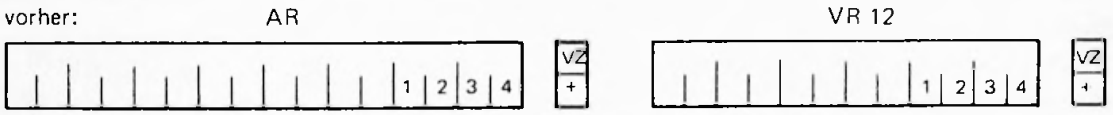
b) AR beinhaltet einen positiven Wert und VR 12 beinhaltet einen negativen Wert.

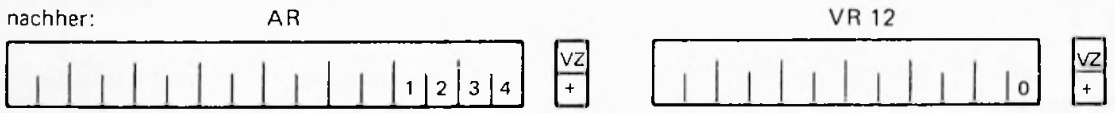
c) AR und VR 12 beinhalten negative Werte.



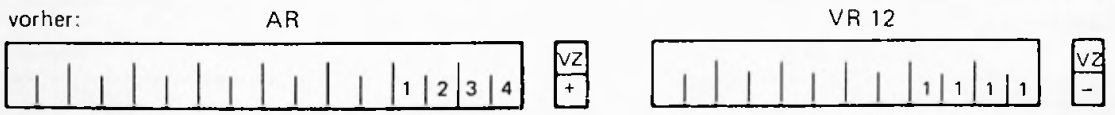


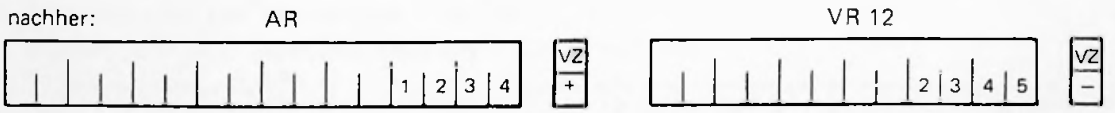
a)



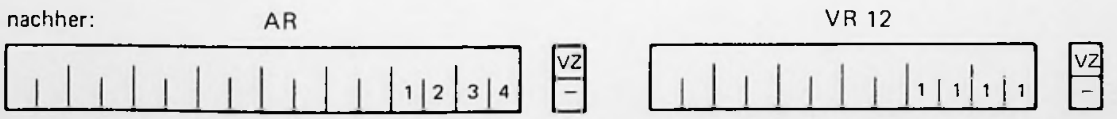


b)





c)



MUL

6.4 Multiply

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
X		X					1. f.	$\overline{ro}$			sr 8	sr 4	sr 2	sr 1

Symbolischer Befehlscode: MUL/1.f./$\overline{ro}$ /15–0

Wirkung

Der Wert im BR wird mit dem Wert im AR multipliziert. Nach der Operation steht das Produkt im AR sowie im BR. Wird jedoch im Adreßteil dieses Befehles Bit 7 programmiert, so wird der im BR stehende 1. Faktor erhalten. Das Produkt steht dann nur im AR zur Verfügung.

Die Vorzeichen für das Produkt werden nach den arithmetischen Regeln errechnet.

Nichtrunden ist im Adreßteil durch Bit 8 zu programmieren. Ist Bit 8 nicht programmiert, so wird das Produkt automatisch gerundet. Gerundet wird immer die nach der Abstreichung verbleibende wertniedrigste Stelle des nunmehr verkürzten Produktes.

Das Runden geschieht in der Form, daß eine 5 an die Stelle des Produktes addiert wird, die gewährleistet, daß ein dabei evtl. entstehender Übertrag sicher an die wertniedrigste Stelle des Produktes übertragen wird.

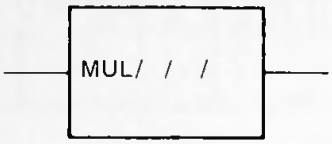
Die Abstreichung ist im Adreßteil mit Bit 11-14 binär verschlüsselt programmierbar (max. 15 Stellen). Unter Abstreichen ist das Unterdrücken der programmierten Anzahl Stellen zu verstehen, um die das Produkt zu verkürzen ist.

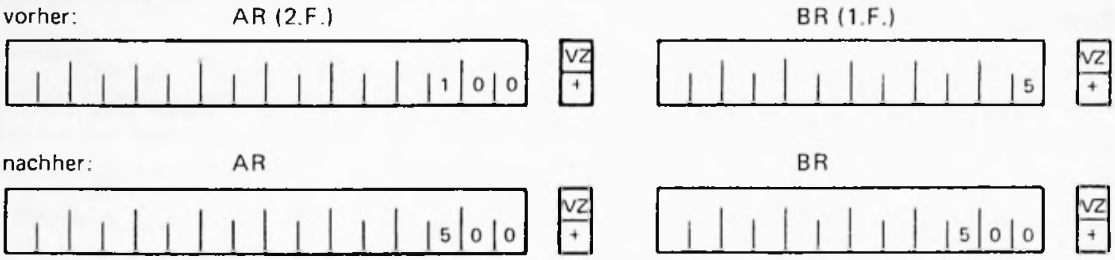
Bemerkung

Das unabgestrichene Produkt darf niemals größer als 15 Stellen sein.

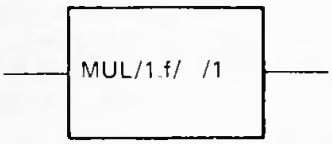
Beispiele:

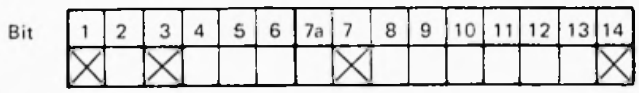
1. Multiplikation ohne Abstreichen

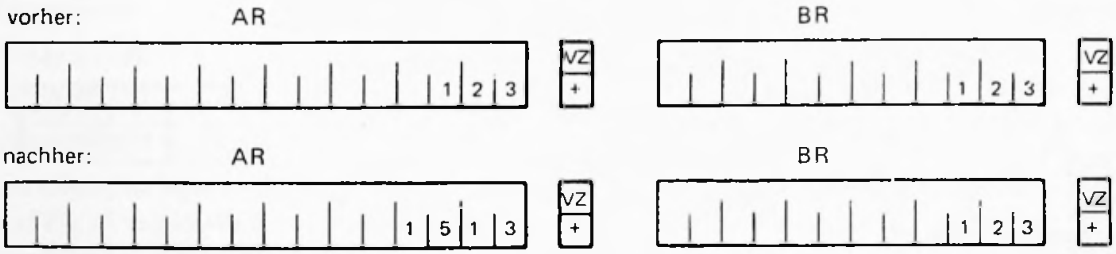




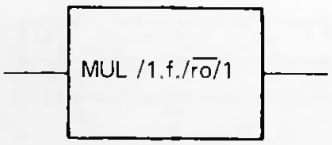
2. Multiplikation mit Abstreichen (1 Stelle)
und Konstanthalten des 1. Faktors im BR

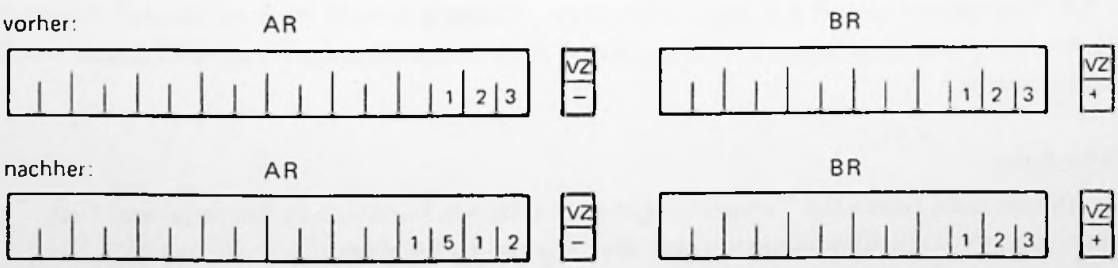






3. Multiplikation mit Abstreichen (1 Stelle)
Nichtrunden und Konstanthalten des 1. Faktors im BR





6.5 WAIT FOR KEY

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
X				X			ni					ky 4	ky 2	ky 1

Symbolischer Befehlscode: WAK/ni/6–1

Wirkung

Dieser Befehl stoppt solange den Programmablauf, bis die im Adreßteil dieses Befehles binär verschlüsselt programmierte Steuertaste (Bit 12-14) betätigt wird. 6 Steuertasten stehen dafür zur Auswahl.

Weiterhin kann im Adreßteil dieses Befehles durch Programmieren von Bit 7 die Zifferntastatur (für Tastwerteingaben) zugelassen werden. Wird Bit 7 nicht programmiert, können keine Tastwerte eingegeben werden.

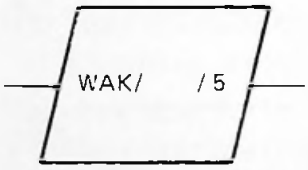
Unzulässig ist, Bit 7 ohne Steuertaste zu programmieren.

Bemerkung

Unzulässig ist es, Bit 12, Bit 13 und Bit 14 (binär verschlüsselte 7) zusammen zu programmieren.

Beispiele

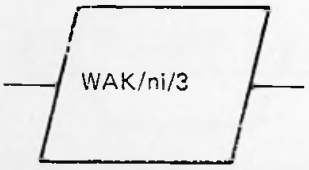
1.



1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
X				X								X		X

Die Maschine wartet so lange, bis die Steuertaste 5 betätigt wird.

2.



1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
X				X			X						X	X

Die Maschine wartet so lange, bis mindestens 1 Ziffer eingetastet und die Steuertaste 3 betätigt werden.

Bemerkung

Die Steuertasten haben bei Tastwerteingaben auslösende Funktionen. Bei negativen Tastwerteingaben ist die Minustaste vor den Steuertasten zu betätigen.

6.6 MARK IF KEY

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
				X			ni		m 4	m 2	m 1	ky 4	ky 2	ky 1

Symbolischer Befehlscode: MAK/ni/4–1/6–1 ← Merker → ← Steuertaste →

Wirkung

Dieser Befehl bewirkt beim Betätigen der im Adreßteil programmierten Steuertaste das Setzen des ebenfalls im Adreßteil programmierten Merkers. Steuertaste sowie Merker werden im Adreßteil binär verschlüsselt programmiert.

6 Steuertasten (Steuertaste 1-6) sowie 4 unterschiedliche Merker (Merker 1-4) stehen dafür beliebig kombinierbar zur Verfügung.

Durch Programmieren von Bit 7 können zusätzlich die Zifferntasten für Tastwerteingaben zugelassen werden.

Wird Bit 7 nicht programmiert, können keine Tastwerte eingegeben werden.

Zu beachten ist, daß immer nur ein Merker gesetzt sein kann. Er wird mit dem nächstfolgenden „Mark if key"- Befehl vor dem Setzen des neuen Merkers gelöscht.

Bemerkung

Dieser Befehl ist ein Vorbefehl der „Jump if mark"- Befehle. Mit ihnen werden für Programmverzweigungen die sogenannten „Programmschleifen" gebildet.

Unzulässig ist es, den Befehl ohne Merker zu programmieren.

Unzulässig ist es, Bit 12, Bit 13 **und** Bit 14 (binär verschlüsselte 7) sowie Bit 9, Bit 10 und Bit 11 (ebenfalls binär verschlüsselte 7) sowie Bit 9 und 11 (binär verschlüsselte 5) zusammen zu programmieren.

Beispiel:

1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
				X			X			X		X		

In diesem Beispiel wird der Merker 2 gesetzt, wenn mindestens 1 Zifferntaste und die Steuertaste 4 betätigt werden.

MAT

6.7 MARK BY VALUE-TEST

(I)

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
X	X	X						vr 7	vr 6	vr 5	vr 4	vr 3	vr 2	vr 1

Symbolischer Befehlscode: MAT/BR–VR 7

Wirkung

Dieser Befehl prüft den Inhalt des im Adreßteil uncodiert programmierten Registers (VR 1-7) oder falls kein VR programmiert ist, den des BR.

Abhängig von dem Inhalt des zu prüfenden Registers werden in der Maschine Merker gesetzt, die von den nachfolgend programmierten Sprungbefehlen „Jump if negativ", oder „Jump if not zero" abgefragt werden.

Geprüft wird der Inhalt des Registers auf $\neq 0$ (ungleich Null) und < 0 (kleiner Null).

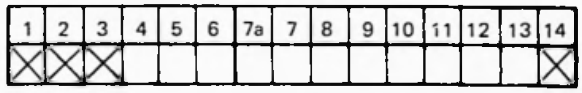
Die dafür zur Verfügung stehenden 2 Merker werden dabei in folgender Weise gesetzt:

Wert = 0 : Es wird kein Merker gesetzt

Wert > 0 : Es wird Merker $\neq 0$ gesetzt

Wert < 0 : Es werden Merker $\neq 0$ und Merker < 0 gesetzt.

Beispiel:



6.7 MARK BY VALUE-TEST

(II)

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
X	X	X									vr 8	vr 4	vr 2	vr 1

Symbolischer Befehlscode: MAT/BR–VR 15

Wirkung

Dieser Befehl prüft den Inhalt des im Adreßteil codiert programmierten Registers (VR 1-15) oder, falls kein VR programmiert ist, den des BR.

Abhängig von dem Inhalt des zu prüfenden Registers werden in der Maschine Merker gesetzt, die von den nachfolgend programmierten Sprungbefehlen „Jump if negativ", oder „Jump if not zero" abgefragt werden.

Geprüft wird der Inhalt des Registers auf $\neq$ 0 (ungleich Null) und $<$ 0 (kleiner Null).

Die dafür zur Verfügung stehenden 2 Merker werden dabei in folgender Weise gesetzt:

Wert = 0 : Es wird kein Merker gesetzt

Wert $>$ 0 : Es wird Merker $\neq$ 0 gesetzt

Wert $<$ 0 : Es werden Merker $\neq$ 0 und Merker $<$ 0 gesetzt.

Beispiel:

1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
X	X	X									X	X	X	X

JUN

6.8 JUMP UNCONDITIONAL

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
	X					ps256	ps128	ps64	ps 32	ps 16	ps 8	ps 4	ps 2	ps 1

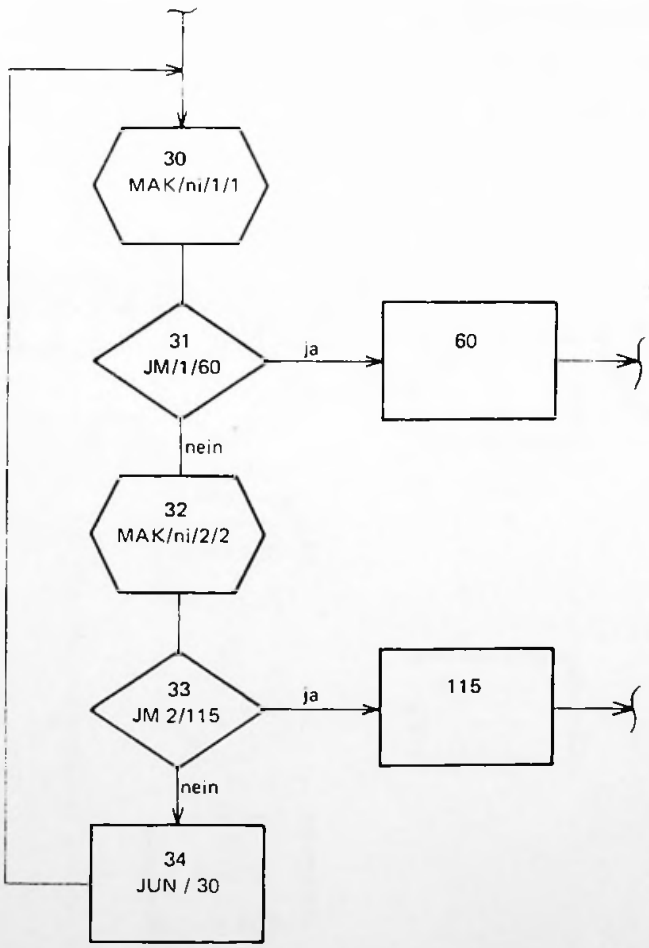
Symbolischer Befehlscode: JUN/511 – 0

Wirkung

Dieser Befehl ist ein unbedingter Sprungbefehl. Er ermöglicht innerhalb des linearen Programmablaufes Vorwärts- und Rückwärtssprünge nach den im Adreßteil binär verschlüsselt adressierten Befehlswortplätzen.

Beispiel:

1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
	X									X	X	X	X	



6.9 JUMP IF MARK 1

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
	X			X		ps256	ps128	ps 64	ps 32	ps 16	ps 8	ps 4	ps 2	ps 1

Symbolischer Befehlscode: JM 1/0–511

Wirkung

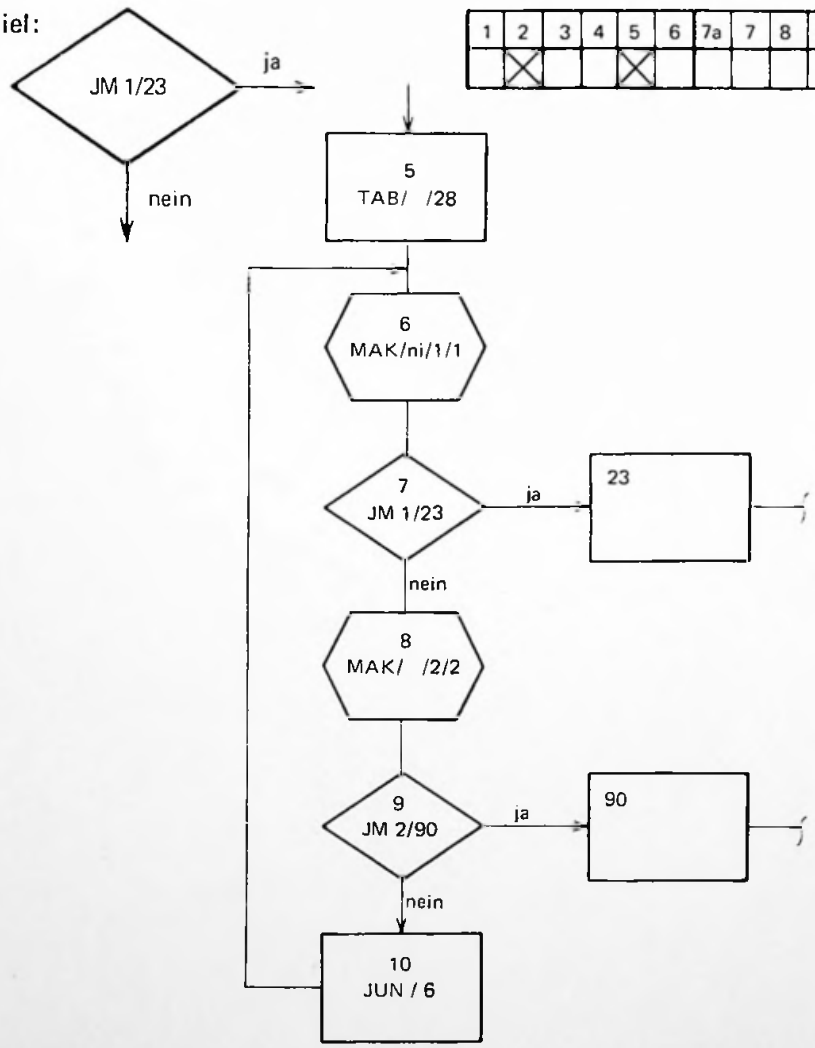
Dieser Befehl ist ein bedingter Sprungbefehl. Er wird nur wirksam, wenn mit dem ihm zugeordneten Vorbefehl der Merker 1 gesetzt wurde.

Wurde der Merker 1 gesetzt, so erfolgt nun ein Sprung nach dem Adreßteil dieses Befehles verschlüsselt adressierten Befehlswortplatz (0 – 511)

Wurde der Merker jedoch nicht gesetzt, so wird der Befehl wirkungslos überlaufen und der nächstfolgend programmierte Befehl ausgeführt.

Beispiel:

1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
	X			X						X		X	X	X



JM 1

Das Beispiel veranschaulicht eine Programmschleife mit bedingten Sprungbefehlen und mit den diesen Befehlen zugeordneten Vorbefehlen. Durch den am Ende der Schleife programmierten unbedingten Sprungbefehl wird die Schleife so lange durchlaufen, bis ein in den Vorbefehlen programmierte Sprungbedingung erfüllt wird. Werden z.B. mindestens eine Zifferntaste und die Steuertaste 1 betätigt (programmierte Sprungkriterien des am Befehlswortplatz 6 programmierten Befehles „MAK"), so wird der Merker 1 gesetzt. Der gesetzte Merker 1 bedeutet für den nachfolgend programmierten bedingten Sprungbefehl (JM 1/23) eine Ausführung des Befehles d.h. einen Sprung nach dem im Adreßteil dieses Befehles programmierten Befehlswortplatz (23).

6.10 JUMP IF MARK 2

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
	X		X			ps256	ps128	ps 64	ps 32	ps 16	ps 8	ps 4	ps 2	ps 1

Symbolischer Befehlscode: JM 2/511 – 0

Wirkung

Wie Jump if Mark 1, jedoch abhängig davon, ob der **Merker 2** gesetzt ist.

JM 3

6.11 JUMP IF MARK 3

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
	X		X	X		ps256	ps128	ps 64	ps 32	ps 16	ps 8	ps 4	ps 2	ps 1

Symbolischer Befehlscode: JM 3/511 – 0

Wirkung

Wie „Jump if mark 1", jedoch abhängig davon, ob der MERKER 3 gesetzt ist.

6.12 JUMP IF MARK 4

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
	X	X				ps256	ps128	ps 64	ps 32	ps 16	ps 8	ps 4	ps 2	ps 1

Symbolischer Befehlscode: JM 4/511 — 0

Wirkung

Wie „Jump if mark 1", jedoch abhängig davon, ob der **Merker 4** gesetzt ist.

JNE

6.13 JUMP IF NEGATIV

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
	X	X		X		ps256	ps128	ps 64	ps 32	ps 16	ps 8	ps 4	ps 2	ps 1

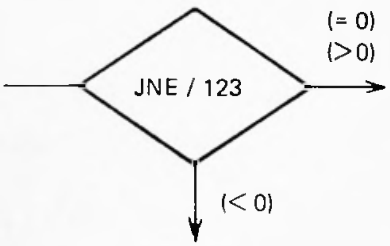
Symbolischer Befehlscode: JNE/511 – 0

Wirkung

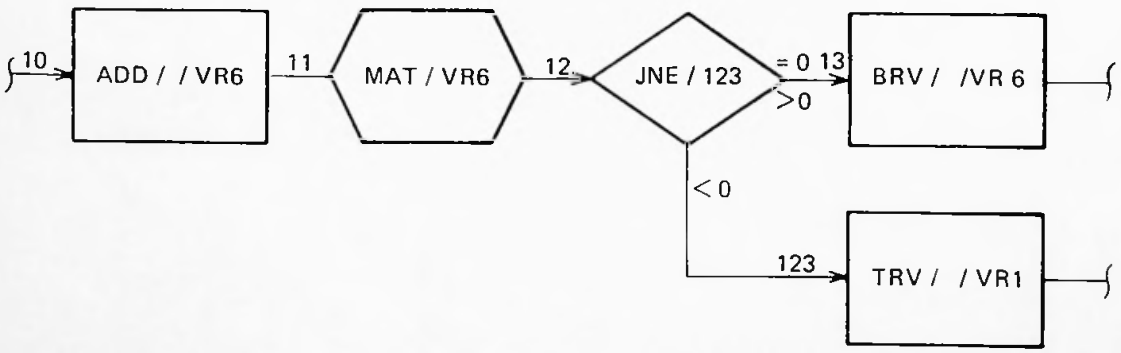
Dieser bedingte Sprungbefehl fragt die mit dem Befehl „Mark by value-test" gesetzten Merker (≠ 0 und < 0) ab. Sind beide Merker gesetzt, (der Wert im Register war negativ) so erfolgt mit diesem Befehl ein Sprung nach dem im Adreßteil binär verschlüsselt adressierten Befehlswortplatz.

Sind die beiden Merker jedoch nicht gesetzt, so wird der Befehl wirkungslos überlaufen und der nächstfolgend programmierte Befehl ausgeführt.

Beispiel:



1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
	X	X		X				X	X	X	X		X	X



Wenn die Maschine den auf dem Befehlswortplatz 12 programmierten Befehl JNE/123 interpretiert und die Merker ≠ 0 und < 0 bis dahin nicht gesetzt sind, wird der auf dem Befehlswortplatz 13 programmierte Befehl ausgeführt. Sind die Merker ≠ 0 und <0 jedoch gesetzt, so erfolgt ein Sprung nach dem im Adreßteil des Befehles 12 programmierten Befehlswortplatz (123).

6.14 JUMP IF NOT ZERO

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
	X	X	X	q		ps256	ps128	ps 64	ps 32	ps 16	ps 8	ps 4	ps 2	ps 1

Symbolischer Befehlscode: JNZ/511 – 0

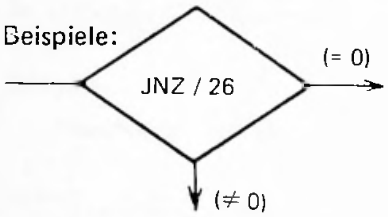
Wirkung

Dieser Befehl fragt den mit dem Befehl „Mark by value-test" gesetzten Merker ≠ 0 ab.

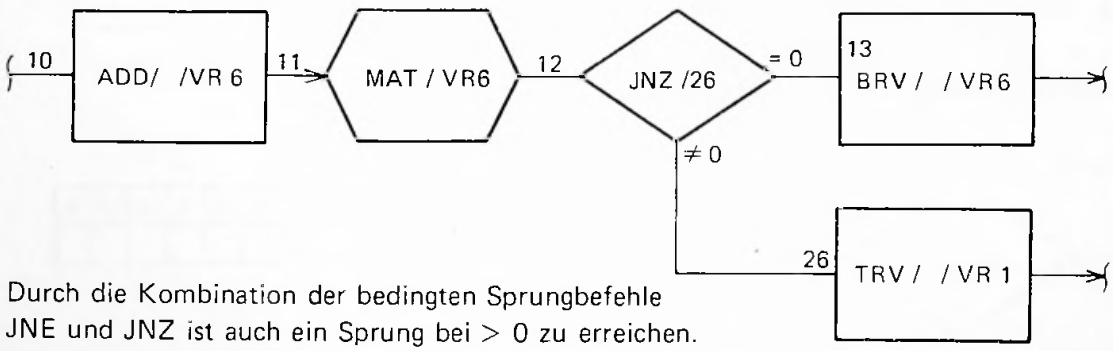
Wurde der Merker ≠ 0 gesetzt, so erfolgt mit diesem Befehl ein Sprung nach dem im Adreßteil binär verschlüsselt programmierten Befehlswortplatz.

Wurde der Merker ≠ 0 jedoch nicht gesetzt, so wird der Befehl wirkungslos überlaufen und der nächstfolgend programmierte Befehl ausgeführt.

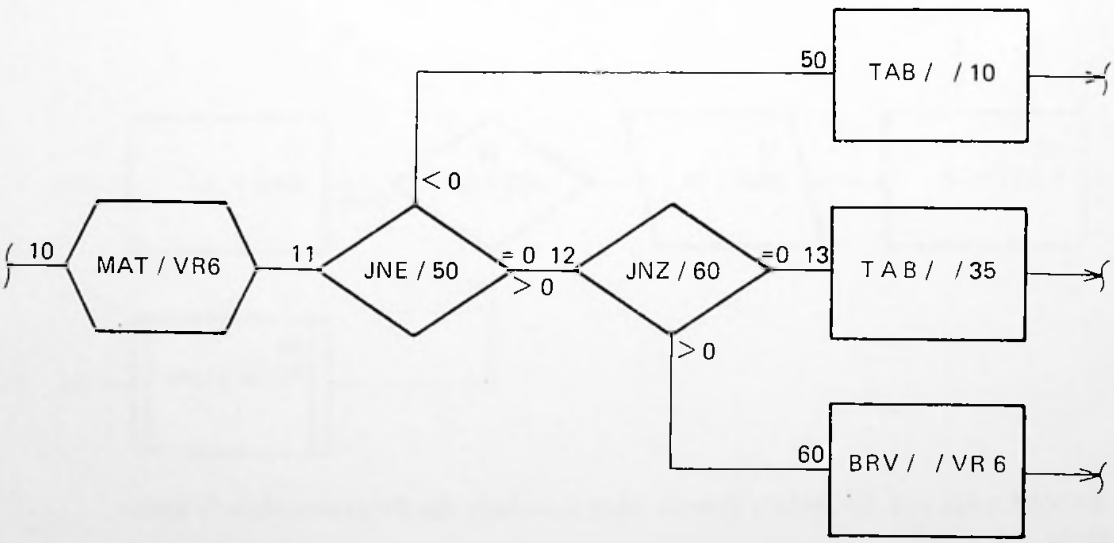
Beispiele:



1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
	X	X	X							X	X		X	



Durch die Kombination der bedingten Sprungbefehle JNE und JNZ ist auch ein Sprung bei > 0 zu erreichen.



JSE

6.15 JUMP IF SELECTOR

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
		X	X	X		ps256	ps128	ps 64	ps 32	ps 16	ps 8	ps 4	ps 2	ps 1

Symbolischer Befehlscode: JSE/511 – 0

Wirkung

Dieser Befehl ist ebenfalls ein bedingter Sprungbefehl. Er wird jedoch nur mit angeschlossener Peripherie wirksam.

Abhängig von bestimmten Peripheriefehlern (F2-Fehlern) wie:

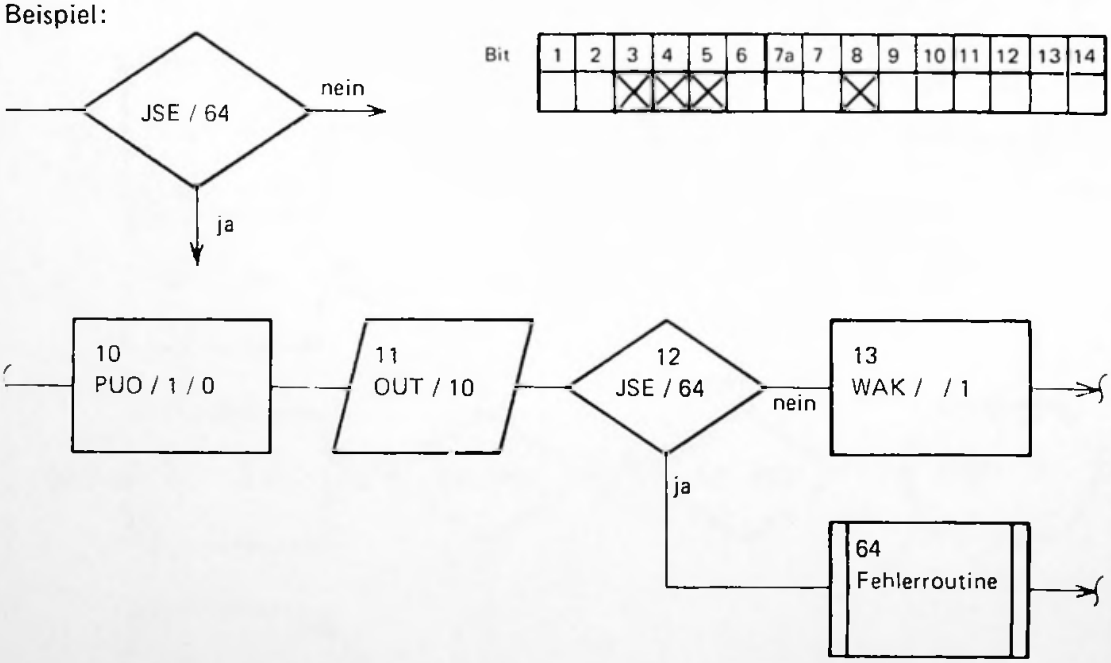
- Paritätsfehler und
- Transporthemmung

wird in der Maschine ein Selector gesetzt, der von diesem Befehl abgefragt und gelöscht wird.

Ist ein Selector gesetzt, so erfolgt ein Sprung nach dem im Adreßteil dieses Befehles binär verschlüsselt programmierten Befehlswortplatz.

Ist kein Selector gesetzt, so wird der lineare Programmablauf fortgesetzt, d.h. es wird der nächstfolgend programmierte Befehl ausgeführt.

Beispiel:



Das Beispiel zeigt wie der Befehl zweckmäßig innerhalb des Programmablaufs einzusetzen ist.

3.16 NO OPERATION

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14

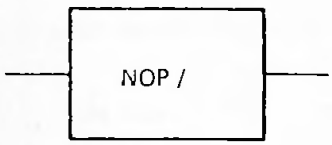
Symbolischer Befehlscode: NOP /

Wirkung

Dieser Befehl schaltet den Programmwortzähler einen Schritt weiter. Der nicht belegte Befehlswortplatz wird dabei übersprungen. An dieser Stelle kann im Programm später ein Befehl nachträglich eingefügt werden, ohne die vorhergehenden oder nachfolgenden Befehlswortplätze verändern zu müssen.

Es können beliebig viele Leerbefehle hintereinander programmiert werden.

Beispiel:



1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14

TAB

6.17 TABULATE

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
		X					te	pc128	pc 64	pc 32	pc 16	pc 8	pc 4	pc 2

Symbolischer Befehlscode: TAB/ te/ 254 – 0

Wirkung

Im Adreßteil dieses Befehles ist die Zieladresse, d.h. die Position, in die der Wagen geführt werden soll, binär verschlüsselt programmierbar. Sie kann kleiner oder größer als die Standadresse des Wagens sein. Ansteuerbar sind jedoch nur **geradzahlige** Positionen.

Achtung: Bei Wagengröße II ist entweder die Teilung 0 oder die jeweilig eingestellte geradzahlige Randstellerposition Wagenanfangs – bzw. Wagengrundstellung.

Bei den Wagengrößen IV und VI ist die Wagenanfang- bzw. Wagengrundstellung grundsätzlich die Teilung 10.

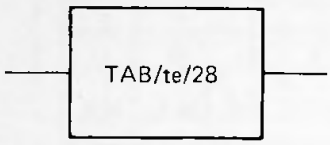
Auf den genannten Wagengrundstellungen steht der Teilungszähler immer auf 0 und beginnt auch ab diesen Positionen von 0 ab zu zählen.

Wird im Adreßteil dieses Befehles Bit 7 (Text) programmiert, so wird im Anschluß an die Tabulation die Alphatastatur für Textschreiben freigegeben.

Ein nicht mit Bit 7 programmierter TAB-Befehl schaltet einen evtl. vorher programmierten Text ab (sperrt die Alphatasten).

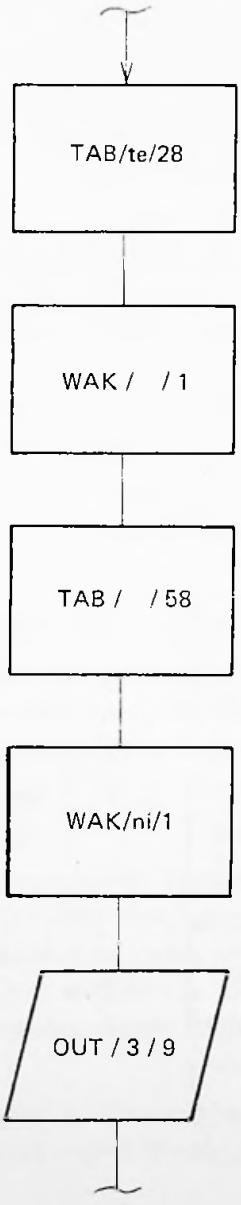
Wird in einem TAB-Befehl Bit 7 (Text) programmiert, so darf in einem nachfolgend programmierten WAK-, MAK-Befehl nicht ebenfalls das Bit 7 (Zifferneingabe zulässig) belegt werden, weil sonst die evtl. per Text geschriebenen Ziffern in das Tastregister (AR) gelangen (siehe Befehlsfolge im Beispiel)

Beispiele:



1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
		X					X				X	X	X	

Befehlsfolge mit richtig eingesetzten TAB/WAK - Befehlen



TAK

6.18 TABULATE, FREE KEYBOARD

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
		X		X				pc128	pc 64	pc 32	pc 16	pc 8	pc 4	pc 2

Symbolischer Befehlscode: TAK/254 – 0

Wirkung

Bis auf Bit 7 (Text), das in diesem Befehl nicht programmiert werden darf, ist dieser Befehl in der Wirkung und Programmierung mit dem Befehl „TAB" identisch.

Ein evtl. vorher programmierter Text wird mit diesem Befehl ebenfalls abgeschaltet.

Nach TAK Befehl muß WAK oder MAK mit Bit 7 (ni) programmiert werden.

Dann können während der Befehlsausführung (Tabulation) Ziffern überlappend in das Tastregister (AR) eingetastet werden (siehe Befehlsfolge im Beispiel)

Beispiele:

1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
		X		X					X					

Befehlsfolge mit richtig eingesetzten TAB/TAK und WAK-Befehlen.

6.19 OUTPUT PREPARATION

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
		X	X				ac	np			sy 8	sy 4	sy 2	sy 1

Symbolischer Befehlscode: OUP/ac/np/15-1

Wirkung

Dieser Befehl ist ein Vorbefehl der Niederschriftsbefehle OUT und OUK.

In seinem Adreßteil können binär verschlüsselt folgende Symbole für eine Niederschriftkennzeichnung programmiert werden:

0	=	Bit 11, 12	5	=	Bit 12, 14	%	=	Bit 11, 13
1	=	Bit 14	6	=	Bit 12, 13	‰	=	Bit 11, 13, 14
2	=	Bit 13	7	=	Bit 12, 13, 14	*	=	Bit 11, 12, 14
3	=	Bit 13, 14	8	=	Bit 11	◊	=	Bit 11, 12, 13
4	=	Bit 12	9	=	Bit 11, 14	MW	=	Bit 11, 12, 13, 14

Das jeweils programmierte Symbol wird in der Maschine gespeichert und im Anschluß an die nachfolgend programmierte Niederschrift (OUT oder OUK) am Journal niedergeschrieben. Pro Niederschrift kann nur ein Symbol als Kennzeichen programmiert werden. Bit 8 ist zu programmieren, wenn eine nichtschreibende Ausgabe auf den Datenträger Lochstreifen (-Karte) erfolgen soll.

6.19a Erweiterung zu OUT PUT PREPARATION

Symbolischer Befehlscode: OUP/ac/np/Text zu- / Text ab- / 15 – 1
schalten schalten

Wirkung:

Klartextzuschaltung ohne Leerschritt nach automatischer Niederschrift

Bit 9 bzw. Bit 10 ist im Adreßteil zu programmieren, wenn Text ohne Leerschritt an eine automatische Niederschrift geschrieben wird. Die bestehenden Möglichkeiten für Kennzeichen (vgl. Befehl CON) werden außerdem erweitert.

Bit 9 bewirkt die Textzuschaltung.
Bit 10 bewirkt die Textabschaltung.

Kennzeichnungen mit den programmierbaren Symbolen der unteren Typenhebelreihe (vgl. Seite 8, $\overline{\text{KUT}}$), die unmittelbar (also ohne Leerschritt) auf eine automatische Niederschrift folgen sollen, lassen sich durch eine zweite programmierte Niederschrift realisieren. Die erste, automatische Niederschrift wird für die Ziffern verwendet (Vorzeichen beachten!), die zweite in Verbindung mit dem Befehl CON für die unmittelbar folgenden Kennzeichen (z.B. % ∫ Ψ ⊣ I).

Es ist also zu beachten, daß mehrstellige Kennzeichen, die unmittelbar auf eine automatische Niederschrift folgen sollen, als eigene Niederschrift mit dem Befehl CON programmiert werden müssen.

Wird nur eine Kennzeichnung nach der automatischen Niederschrift benötigt, so kann die Kennzeichnung im OUP Befehl (Bit 11-14) mitprogrammiert werden (CON Befehl kann entfallen.).

Erweiterung der Kennzeichnung

Die durch Bit 9 veranlaßte Korbumschaltung ermöglicht ferner eine Erweiterung der bestehenden Kennzeichnungsmöglichkeiten. Es können die in Abb. 3/Seite 5 stark umrandeten Schriftzeichen der automatischen Niederschrift programmiert, jedoch die darüber stehenden Ziffern und Buchstaben damit niedergeschrieben werden (je 15 Zeichen der rechten und linken Korbhälfte).

Mehrere Zeichen können nur niedergeschrieben werden, wenn die Bedingung eingehalten wird, daß das erste niederzuschreibende Zeichen aus der rechten Korbhälfte, das zweite aus der linken usw. stammt. Beginnt eine Niederschrift mit einem Zeichen aus der linken Korbhälfte, muß vorher ein Leerschritt (aus der rechten Hälfte) ausgeführt werden. Die Kapazität der Niederschrift muß dann, wegen des Leerschrittes, um 1 Stelle größer programmiert werden.

Erweiterte Kennzeichnung (KUT) (nach Abb. 3 Seite 5) Typenhebel der		Bitbelegung (im CON-Befehl)	
linken Korbhälfte 1; 5 – 19;	rechten Korbhälfte 27 – 41; 45	(im Adreßteil binär verschlüsselt)	
E	0	Bit	11, 12
4	L	Bit	14,
D	0	Bit	13,
C	,	Bit	13, 14
R	9	Bit	12,
F	K	Bit	12, 14
5	I	Bit	12, 13
V	M	Bit	12, 13, 14
T	8	Bit	11,
G	J	Bit	11, 14
3	P	Bit	11, 13
S	Ö	Bit	11, 13, 14
Y	- (minus)	Bit	11, 12, 14
W	– (Unterstrg.)	Bit	11, 12, 13
1	2	Bit	11, 12, 13, 14

Nur für Modell TA 10/2 und TA 10/3:

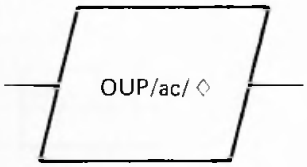
Bit 7 ist im Adreßteil zu programmieren, wenn an die Ziffernniederschrift, entweder zusätzlich zu der bereits genannten Symbolisierung oder ohne diese, ein zweistelliges Alphasymbol niedergeschrieben werden soll. Die Niederschrift sowie die Auswahl der Alphasymbolkombination sind jedoch von dem Betätigen einer (von neun zur Auswahl stehenden) Alphasymboltaste abhängig.

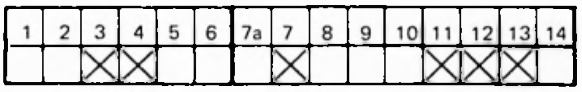
Ist keine Alphasymboltaste bis zum Zeitpunkt der Befehlsbearbeitung betätigt, so wird die Alphasymbolisierung von der Maschine ignoriert.

Am Journal geschrieben werden die Alphasymbole im Anschluß an eine nachfolgend programmierte Ziffernniederschrift, wobei zwischen der zuletzt niedergeschriebenen Ziffer oder dem zuletzt niedergeschriebenen Symbol und dem ersten Alphasymbol grundsätzlich ein Leerschritt eingeblendet wird.

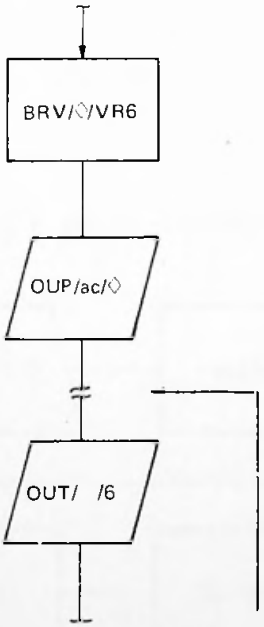
(Als Beispiel hierzu s. Seite 71 Pkt. 2.)

Beispiele:





Befehlsfolge mit OUP/OUT - Befehlen:

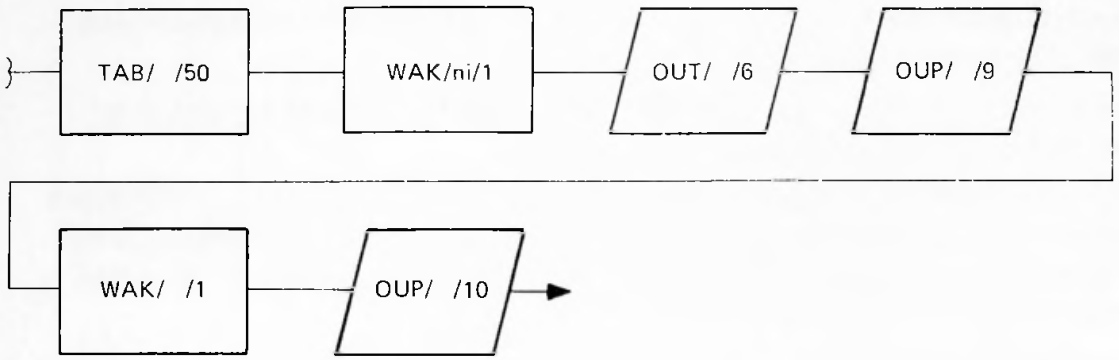


Zwischen dem Befehl OUP und den Befehlen OUT oder OUK können beliebig viele andere Befehle programmiert werden. Erst der Niederschriftsbefehl (OUT oder OUK) löscht die programmierten Kriterien.

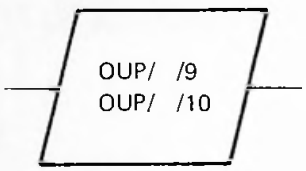
OUP

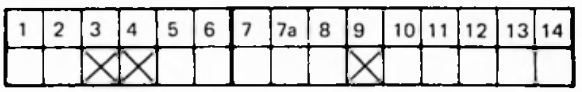
Beispiele:

Ziffernniederschrift (6 Stellen) mit Klartexteingabe, ohne Teilungszwischenraum (Leerschritt)



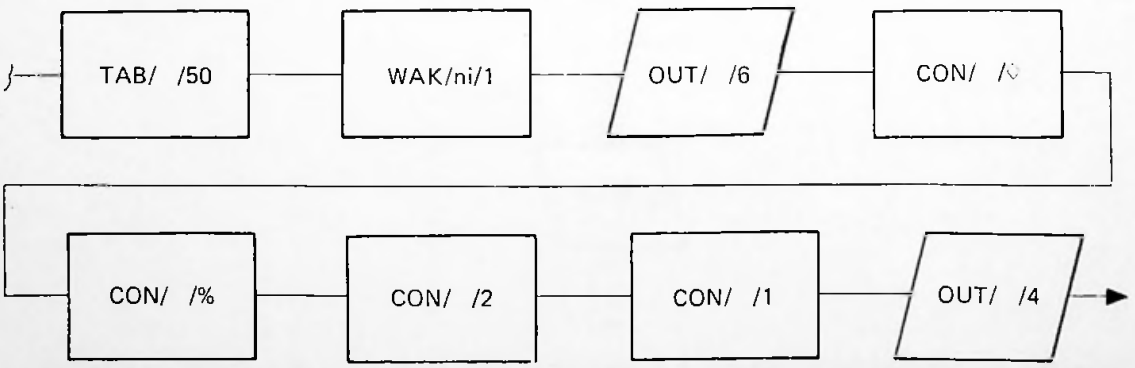
Notwendige Vorbefehle:





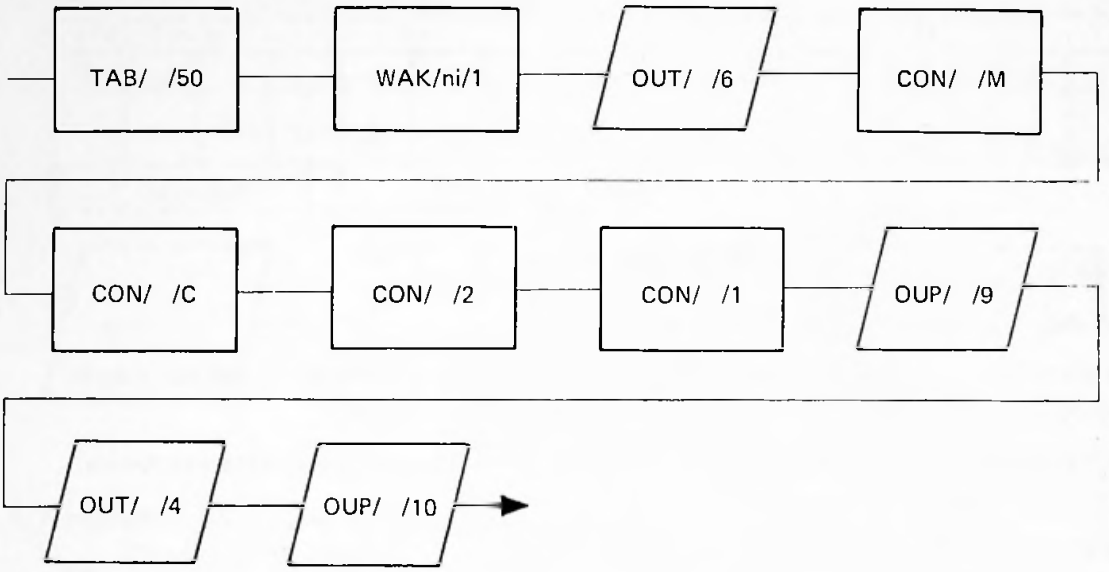
positive Ausschrift : 1 2 3 4 5 6 A B C D . . .
negative Ausschrift : 1 2 3 4 5 6 – A B C D

Ziffernniederschrift (6 Stellen) mit Symbolisierung (4 St.) der unteren Typenreihe (Grundstellung, $\overline{\text{KUT}}$)



positive Ausschrift : 1 2 3 4 5 6 % 2 1
negative Ausschrift : 1 2 3 4 5 6 – % 2 1

Ziffernniederschrift (6 Stellen) mit Symb. (4 Stellen), der oberen Typenreihe (KUT)



positive Ausschrift : 1 2 3 4 5 6 M C 2 1
negative Ausschrift : 1 2 3 4 5 6 – M C 2 1

OUT

6.20 OUTPUT

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
			X				sp 6	sp 4	sp 3	sp 2	cp 8	cp 4	cp 2	cp 1
							← Leerschritte →				← Kapazität →			

Symbolischer Befehlscode: OUT/6, 4, 3, 2/15 – 1

Wirkung

Mit diesem Befehl wird der im AR stehende Wert in das CR übernommen und aus diesem niedergeschrieben. Der im AR stehende Wert bleibt dabei erhalten.

Die für das Schriftbild notwendigen Angaben wie Kapazität und Leerschritte werden im Adreßteil programmiert.

Mit Bit 11 - 14 wird die Kapazität des Schriftbildes binär verschlüsselt programmiert.

Mit Bit 7 - 10 werden die für das Schriftbild benötigten Leerschritte uncodiert angegeben.

Mögliche Leerschritte sind an der 2. 3. 4. 6. (und 8.) Schreibstelle programmierbar.

Die zu programmierende Kapazität setzt sich aus der Anzahl der zu schreibenden Ziffern plus den dabei vorgesehenen Leerschritten zusammen. Ohne Leerschritte können maximal 15 Ziffern aus dem CR niedergeschrieben werden.

Symbole und/oder Alpha-Kurztextzeichen haben keinen Einfluß auf die zu programmierende Kapazität.

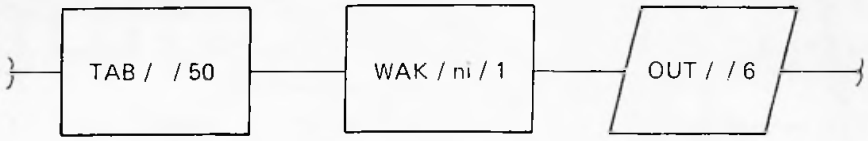
Eine überlappende Eingabe ist mit diesem Befehl **nicht** möglich.

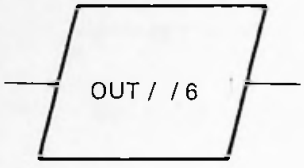
Für das Einblenden der Leerschritte bestehen folgende festen Regeln:

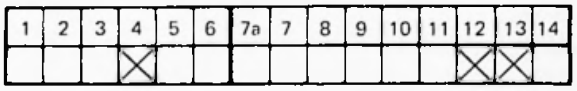
- Die programmierte Kapazität muß grundsätzlich eine Schreibstelle größer sein als die Schreibstelle an der der werthöchste Leerschritt programmiert ist. Bei einem Leerschritt an der 3. Schreibstelle muß die Kapazität also mindestens 4 betragen.
- Ein Leerschritt an der 8. Schreibstelle wird automatisch eingeblendet, wenn eine Kapazität >8 und der Leerschritt an der 4. Schreibstelle programmiert ist.
- Es können beliebig viele Leerschritte in einem Befehl programmiert werden.
- Getastete und nicht verrechnete führende Nullen werden geschrieben.
- Abgegriffene, verrechnete, führende Nullen gehen verloren. Abgegriffene Werte werden daher eine Stelle vor dem werthöchsten Leerschritt mit Nullen ergänzt. Ist kein Leerschritt programmiert und der aus einem Register abgegriffene und niederzuschreibende Wert ist Null, so wird an der wertniedrigsten Schreibstelle eine Null geschrieben.

Beispiele:

1. Niederschrift (6 Stellen) ohne zusätzliche Symbolisierung

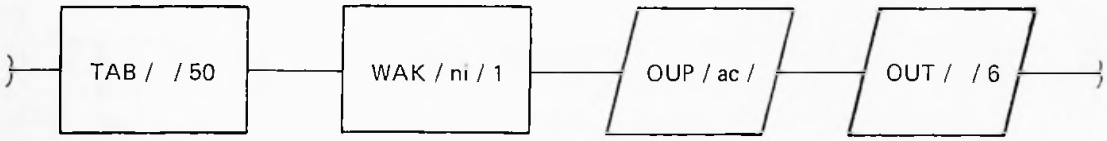




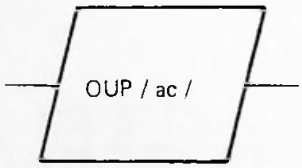


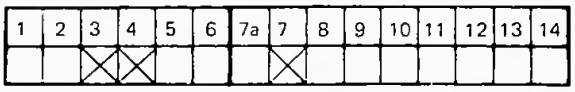
Ausschrift positiv : 1 2 3 4 5 6
Ausschrift negativ : 1 2 3 4 5 6 -

2. Niederschrift (6 Stellen) mit Alphasymbolisierung



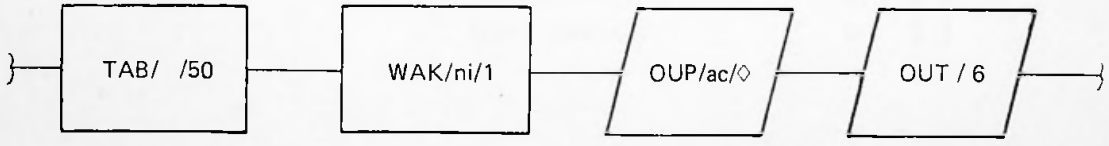
Notwendiger Vorbefehl



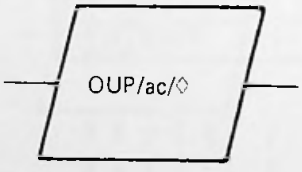


Ausschrift positiv : 1 2 3 4 5 6 B A
Ausschrift negativ : 1 2 3 4 5 6 - B A

3. Niederschrift (6 Stellen) mit Kennzeichnung und Alphasymbolisierung



Notwendiger Vorbefehl



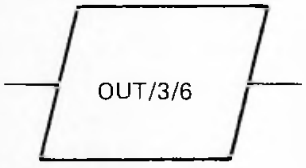
Ausschrift positiv : 1 2 3 4 5 6 ◊ B A
Ausschrift negativ : 1 2 3 4 5 6 - ◊ B A

OUT

4. Niederschrift mit Leerschritten

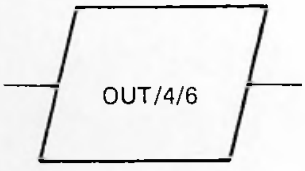
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
			X							X		X	X	

Ausschrift positiv : 1 2 3 4 5
Ausschrift negativ : 1 2 3 4 5-
Nullergänzung : 0 1



1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
			X						X			X	X	

Ausschrift positiv : 1 2 3 4 5
Ausschrift negativ : 1 2 3 4 5-
Nullergänzung : 0 0 1



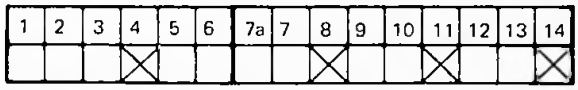
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
			X					X				X	X	

Ausschrift positiv : 1 2 3 4 5
Ausschrift negativ : 1 2 3 4 5-
Nullergänzung : 0 0 0 1

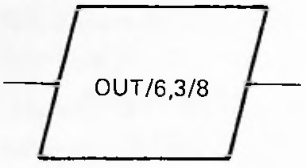
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
			X				X					X	X	X

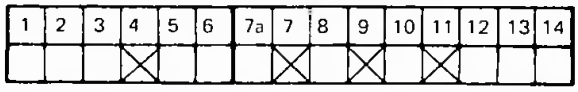
Ausschrift positiv : 1 2 3 4 5 6
Ausschrift negativ : 1 2 3 4 5 6-
Nullergänzung : 0 0 0 0 0 1-

OUT

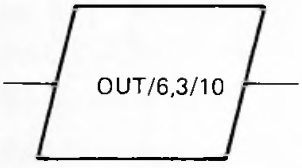


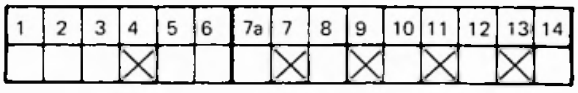
Ausschrift positiv : 1 2 3 4 5 6 7
Ausschrift negativ : 1 2 3 4 5 6 7-
Nullergänzung : 0 0 0 0 0 0 1



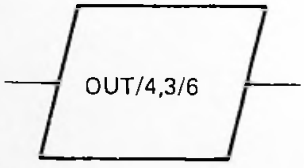


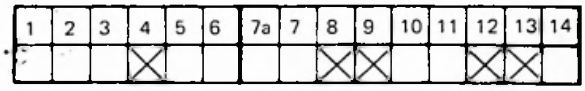
Ausschrift (Datum) 2 4 1 0 7 1



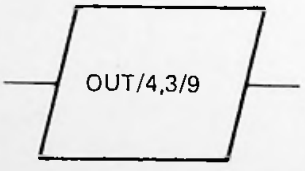


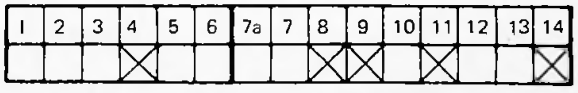
Ausschrift (Datum) 1 9 7 1 1 0 2 4





Ausschrift positiv : 1 2 3 4
Ausschrift negativ : 1 2 3 4-
Nullergänzung : 0 0 1

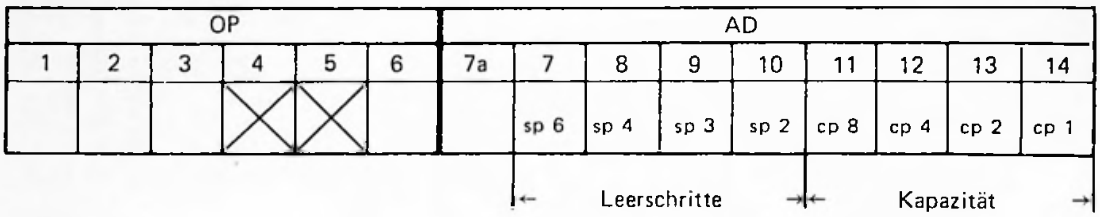




Ausschrift positiv : 1 2 3 4 5 6
Ausschrift negativ : 1 2 3 4 5 6-
Nullergänzung : 0 0 0 0 0 1

OUK

6.21 OUTPUT, FREE KEYBOARD

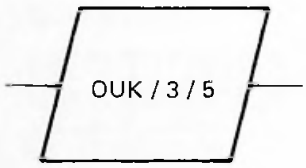


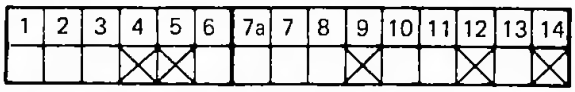
Symbolischer Befehlscode: OUK/6,4,3,2/15 – 1

Wirkung

Wie Befehl „OUT", nur ist mit diesem Befehl eine überlappende Zifferneingabe möglich. Er darf jedoch nur dann programmiert werden, wenn im nächstfolgenden Befehlswort „Zifferneingabe zulässig" programmiert wird (WAK/ni/ oder MAK/ni/)

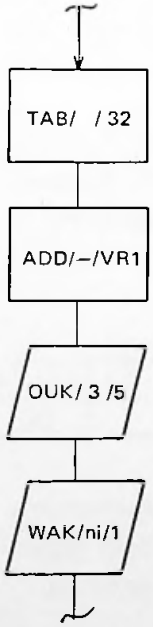
Beispiele





Ausschrift positiv	:	1 2 3 4
Ausschrift negativ	:	1 2 3 4 -
Nullergänzung	:	0 0 1

Befehlsfolge mit richtig eingesetzten OUK/WAK-Befehlen:



6.22 CONSTANT

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
X	X		X				–				co 8	co 4	co 2	co 1

Symbolischer Befehlscode: CON/–/15–0

Wirkung

Mit diesem Befehl werden numerische Festwerte (Ziffern 0-9) positiv oder negativ und Symbole (%, ‰, *, ◇, MW) in das Tastregister (AR) eingeschrieben.

Das jeweils in das Register einzuschreibende Zeichen ist im Adreßteil dieses Befehles binär verschlüsselt zu programmieren. Die Bitmuster der Zeichen sind in der Beschreibung des OUP-Befehles zu finden.

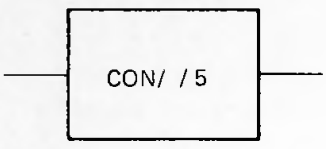
Sollen mehrere Zeichen in das AR eingeschrieben werden, so ist der Befehl ohne Unterbrechung mehrmals hintereinander in der Reihenfolge der Tastung zu programmieren. Soll eine negative Ziffer oder ein negativer Wert in das AR eingeschrieben werden, so ist im Adreßteil dieses Befehles Bit 7 zu programmieren. Bei einer negativen Ziffernfolge genügt es jedoch, nur eine der einzuschreibenden Ziffern mit Bit 7 zu kennzeichnen.

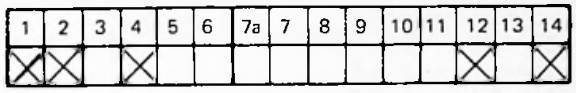
Ein im AR stehender Wert wird mit diesem Befehl gelöscht.

Eine Verrechnung der in das AR eingeschriebenen Zeichen ist nur dann möglich, wenn diese Zeichen rein numerisch sind. Zur Niederschrift können jedoch alle programmierten Zeichen gelangen.

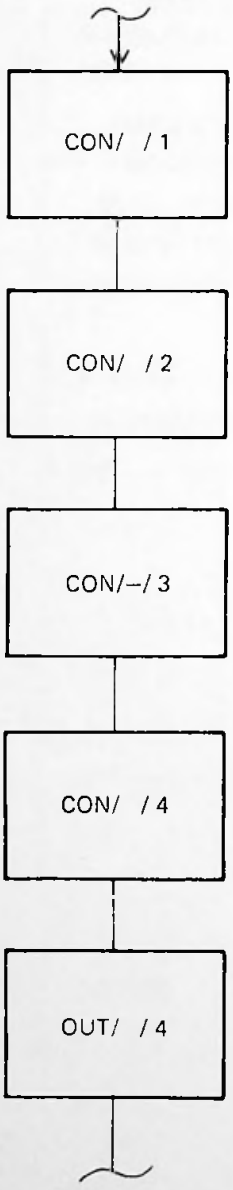
CON

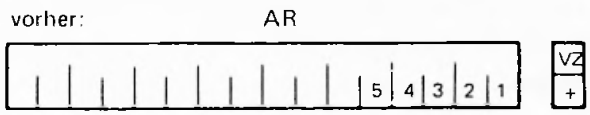
Beispiele:

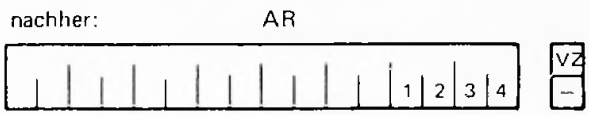




Befehlsfolge einer in das AR einzuschreibenden negativen Ziffernfolge mit anschließender Niederschrift.

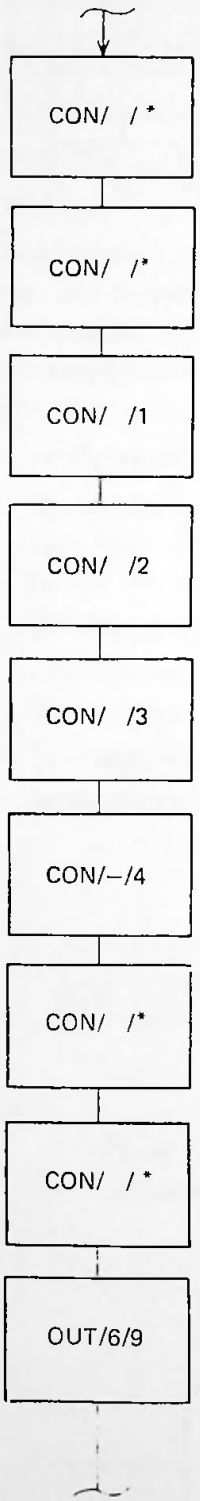


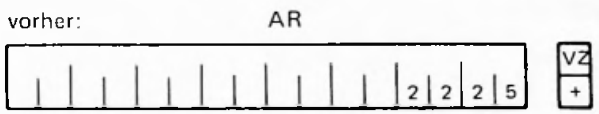


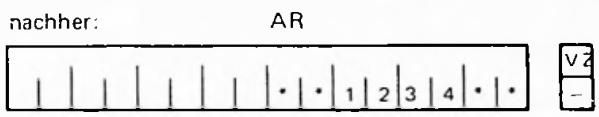


Ausschrift: 1 2 3 4 –

Befehlsfolge einer in das AR einzuschreibenden Zeichenfolge mit anschließender Niederschrift.



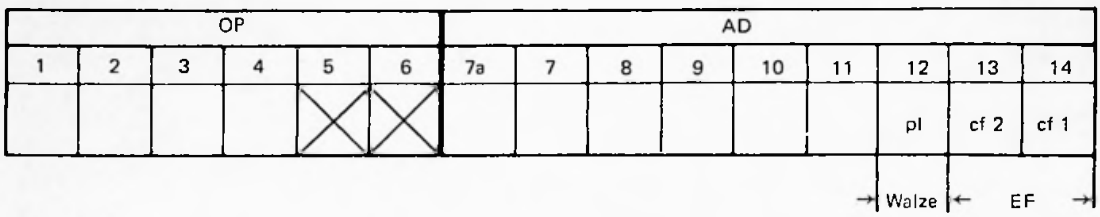




Ausschrift: * * 1 234 * * –

SKP

6.23 SKIP FOR FORM



Symbolischer Befehlscode: SKP/pl/3–1

Wirkung

Dieser Befehl ist nur beim Modell TA 10/2 und TA 10/3 mit angeschlossener EF wirksam. In seinem Adreßteil sind folgende für die vertikale Formatsteuerung von der EF benötigten Startkriterien binär verschlüsselt programmierbar:

1 = Zeilenschaltung am Endlosformular

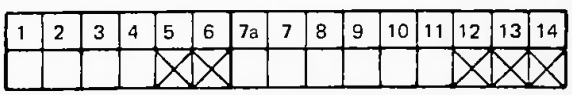
2 = Vertikaltabulation am Endlosformular

3 = Vertikalgrundstellung am Endlosformular

Die dazu gehörenden Stopcodierungen sind in dem Loch- (Steuer)streifen der EF zu programmieren.
Bit 12 bewirkt gleichzeitig eine Zeilenschaltung an der Schreibwerkswalze.

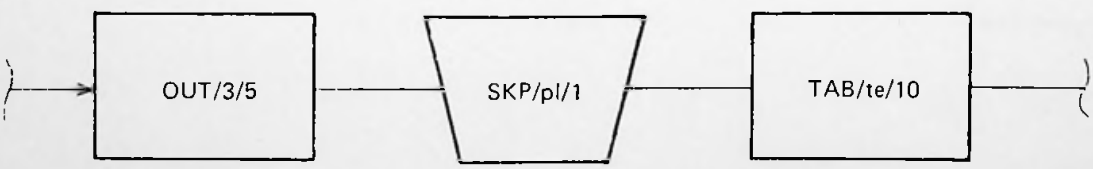
Während der Ausführung dieses Befehles werden von der Maschine außer Niederschriftsbefehle (OUT, OUK) und außer einem nochmaligen SKP-Befehl alle weiteren programmierten Befehle gleichzeitig ausgeführt.

Beispiele:



In diesem Beispiel wird an der Schreibwerkwalze eine Zeilenschaltung und an der EF ein Transport des Endlosformulares in die definierte Grundstellung des Formulares bewirkt.

Befehlsfolge mit richtig eingesetztem SKP-Befehl:



6.24 END

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
X	X	X	X	X	X	ps256	ps128	ps 64	ps 32	ps 16	ps 8	ps 4	ps 2	ps 1

Symbolischer Befehlscode: END/511 – 0

Wirkung

Dieser Befehl bildet den Abschluß der Programmeingabe beim **Modell TA 10/3** und ist **nur** mit **angeschlossener Programmeingabe-Tastatur wirksam.** Ist die Programmeingabe beendet, so wird im Anschluß an das zuletzt in den MOS-Speicher übernommene Befehlswort der **Programmende-Befehl** eingegeben. Die Eingabe des Programmende-Befehls ist wie folgt:

- **im Operationsteil** sind sämtliche **Tasten** (Bit 1 bis 6) **einzutasten**
- **im Adreßteil** wird der letzte Programmschrittzähler-Stand (obere Lampenreihe) imitiert. D.h., es wird die im Programmschrittzähler **aufleuchtende Lampenkombination** des letzten Befehlsschrittes durch **Eingeben der darunterliegenden Adreßtasten nachgebildet.**

  Anschließend wird der eingegebene **Programmende-Befehl mit der BÜ-Taste** in den MOS-Speicher **übernommen.**

  Der Programmende-Befehl „END" richtet sich stets nach dem zuletzt eingegebenen Befehlsschritt und variiert deshalb mit jeweiligem Programm.

7. LOCHSTREIFEN (–KARTEN) –AUSGABE

7.1 Organisation der Daten auf dem Datenträger (allgemein)

Von der Grundmaschine können aus dem Ausgaberegister (CR) maximal 15 Ziffern zusammenhängend auf den Datenträger Lochstreifen oder Lochstreifenkarten ausgegeben bzw. gelocht werden.

Mehrere solcher zusammenhängender Informationen, auch „Wörter" genannt, werden nun je nach Organisation zu einer logisch zusammengehörenden Informationseinheit zusammengefaßt **und bilden einen Satz.** Ein Satz kann bei der Fakturierung z. B. eine Zeile oder auch mehrere gleichartige Zeilen beinhalten.

Unterschiedliche Sätze, auch unterschiedliche Satzarten, genannt, werden mit unterschiedlichen Kennzeichnungen markiert.

Unter einer Kennzeichnung ist zu verstehen, daß sowohl Wörter als auch Sätze mit bestimmten, vorher festgelegten Kennzeichen versehen werden, die eine Aussage über den Inhalt der Informationen geben.

Kennzeichen können am Anfang oder am Ende der Wörter und Sätze verlangt werden. Die Anzahl, Art und Position der Kennzeichen werden in der Regel von den Auswertern des Datenträgers bestimmt.

Ebenso wird bestimmt, ob die Daten mit konstanter oder variabler Wort- und Satzlänge auf den Datenträger aufgezeichnet werden. Unter **konstanter Wortlänge** ist zu verstehen, daß die innerhalb einer Satzart enthaltenen Wörter eine feste Anzahl sind, die jeweils eine feste Informationslänge haben. Eine feste Wortlänge wird erreicht, wenn sowohl führende als auch evtl. schleppende Ergänzungszeichen am Datenträger gelocht werden.

Nehmen wir an, daß als feste Informationslänge (Kapazität) eines Wortes 10 Zeichen programmiert sind. In dem dazu gehörenden Lochbefehl: Loche 10 numerische Zeichen – ohne Kennzeichen – wird, wenn z.B. die Ziffernfolge 123400 eingetastet wird:

123400 am Journal geschrieben

und XXXX 123400 am Datenträger gelocht.

Die an den 4 werthöchsten Dekaden gelochten Zeichen (X) sind **führende Ergänzungszeichen,** die von der Maschine bei eingestellter konstanter Wortlänge (am Wahlschalter im Beistelltisch Schalterstellung 2, 3 oder 4) die in dem Lochbefehl programmierte Adresse ergänzen.

Wäre in dem Beispiel ein Lochbefehl programmiert der aussagt: Loche 13 numerische Zeichen – ohne Kennzeichen – so würde am Datenträger die Zeichenfolge:

XXXX 123400YYY

gelocht.

Die an den wertniedrigsten Dekaden gelochten Zeichen (Y) sind **schleppende Ergänzungszeichen,** die von der Maschine ebenfalls automatisch ergänzt werden.

Die Bitmuster der führenden und schleppenden Ergänzungszeichen sind in der im Beistelltisch eingebauten Kennzeichenmatrix beliebig einstellbar.

Variable Wortlänge heißt, daß grundsätzlich nur die getasteten oder aus einem Register abgegriffenen Zeichen, **ohne führende,** jedoch evtl. mit schleppenden Ergänzungszeichen auf den Datenträger ausgegeben werden.

Bei genannten Beispielen würden bei variabler Wortlänge also einmal nur die eingetastete und niedergeschriebene Ziffernfolge 123400, und einmal die Zeichenfolge 123400YYY auf dem Datenträger gelocht.

Variable Satzlänge liegt vor, wenn die in einem Satz zusammengefaßten Wörter keine festen Informationslängen haben.

Während bei **konstanter Satzlänge** gleicher Satzart unterschiedliche Wortarten durch die Stellung im Satz definiert sind, werden bei variabler Satzlänge unterschiedliche Wortarten durch Kennzeichen definiert.

Kennzeichen sind in der Kennzeichenmatrix beliebig einstellbare Bitmuster, die von den Maschinenkriterien +, –, % und ‰ in der Matrix angesteuert werden und mit den entsprechenden Lochbefehlen am Datenträger im Anschluß an die Ziffern abgelocht werden.

7.2 OBJEKTBESCHREIBUNG

Der in einem fahrbaren Beistelltisch untergebrachte Lochstreifen- (Karten-) Stanzer beinhaltet außer dem mechanischen Stanzaggregat:

- die Abspuleinrichtung und die elektromotorisch angetriebene Aufspuleinrichtung für Lochstreifen
- die Adaptionselektronik für das Zusammenspiel von Grundmaschine und Stanzer.
- den Codierer
- die Kennzeichenmatrix
- und das Netzgerät

Außerdem liegen ihm ein Zufuhr- und Ablagemagazin für Lochstreifenkarten bei.

Mit der Grundmaschine wird der Lochstreifen (Karten-) Stanzer mit dem aus dem Beistelltisch herausgeführten 39-poligen Kabel verbunden.

Innerhalb dieser Kabellänge kann er der Grundmaschine beliebig beigestellt werden (s. Abb. 21)

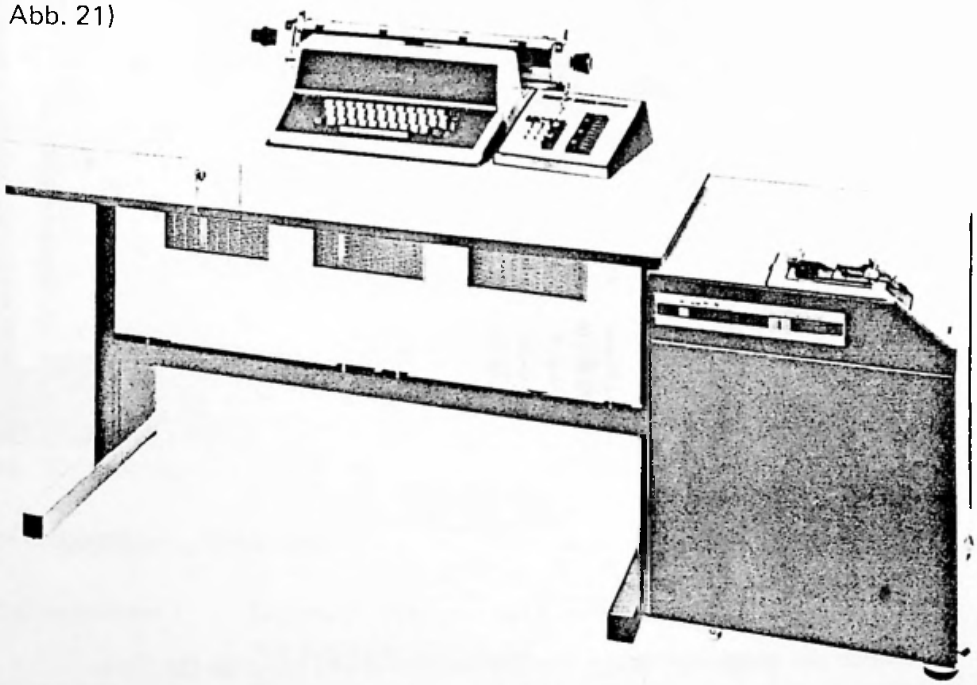


Abb. 21 TA 10/2 (TA 10/3) mit angeschlossenem Lochstreifen (-Karten) -Stanzer

Die Stanzgeschwindigkeit beträgt ca. 18 Zeichen pro Sekunde.

Folgende Datenträger können am Lochstreifen- (Karten-) Stanzer verarbeitet werden.

a) **Lochstreifen**

Je nach Lochstreifenführungseinstellung können drei unterschiedliche Lochstreifenbreiten 17,4 mm, 22,2 mm oder 25,4 mm verwendet werden. Die Lochstreifen dürfen jedoch nicht länger als 250 m sein.

Abmessungen und Eigenschaften von Lochstreifenpapier müssen DIN 6720, Blatt 2, Ausgabe Mai 1967: **Papiere für Fernschreibgeräte**, Lochstreifenpapiere für Fernschreib- und Datengeräte, entsprechen.

Die Anordnung der Löcher auf dem Lochstreifen als Informationsträger ist in DIN 66016, Blatt 1, Ausgabe Juni 1966, und Blatt 2, Vornorm, Juni 1966, festgelegt.

b) **Lochstreifenkarten**

Lochstreifenkarten können je nach Einstellung der Kartenführung bis **3 1/4 Zoll (82.55)** breit sein

Am Beistelltisch sind frontseitig in einem geschlossenen Bedien- und Anzeigefeld folgende Bedien- und Signalisierungselemente angeordnet:

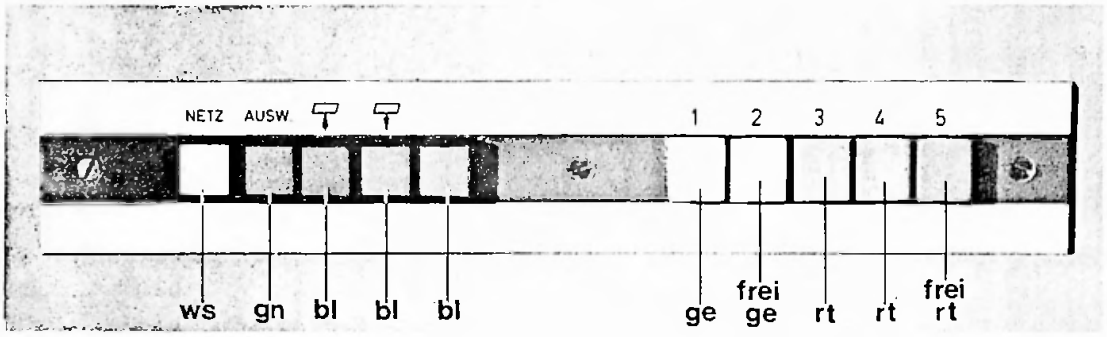


Abb. 22 Lochstreifen- (Karten-) Stanzer-Bedienungs- und Anzeigefeld

Leuchttasten (Wirkung)

NETZ: EIN/AUS; Das Leuchten der Taste zeigt die Betriebsbereitschaft des Lochstreifen- (Karten-) stanzers an. Grundmaschine und Stanzer können in beliebiger Reihenfolge ein/ausgeschaltet werden.

AUSW: keine Wirkung: Das Leuchten der Taste zeigt an, daß ein Peripheriebefehl vom Lochstreifen- (Karten-) Stanzer abgearbeitet wird

: a) **Lochstreifen**
Transportspurlochen, solange die Taste betätigt wird.

b) **Lochstreifenkarte**
Kartenwechsel (beim Einschalten der LSKA oder aus der Grundstellung einer Karte). Die Taste ist nichtleuchtend.

: a) **Lochstreifen**
Lochen der programmierten Streifenlaufcodierung (solange die Taste betätigt wird)

b) **Lochstreifenkarte**
keine Wirkung
Die Taste ist nichtleuchtend

nichtbezeichnete
Taste: a) **Lochstreifen**
Fehlerbehebungstaste bei Dateiende

b) **Lochstreifenkarte**
keine Wirkung
Die Taste ist nichtleuchtend

Signallampen (Bedeutung)

1 (ge) keine Streifenrolle eingelegt (Dateiende)
2 (ge) nicht angeschlossen
3 (rt) Transporthemmung
4 (rt) Stanzfehler (Paritätsfehler)
5 (rt) nicht angeschlossen

Innerhalb des Beistelltisches Abb. 23 sind der Kippschalter (A), mit dem der Aufspulmotor ein- ausgeschaltet wird, und der Wahlschalter (B) angeordnet.

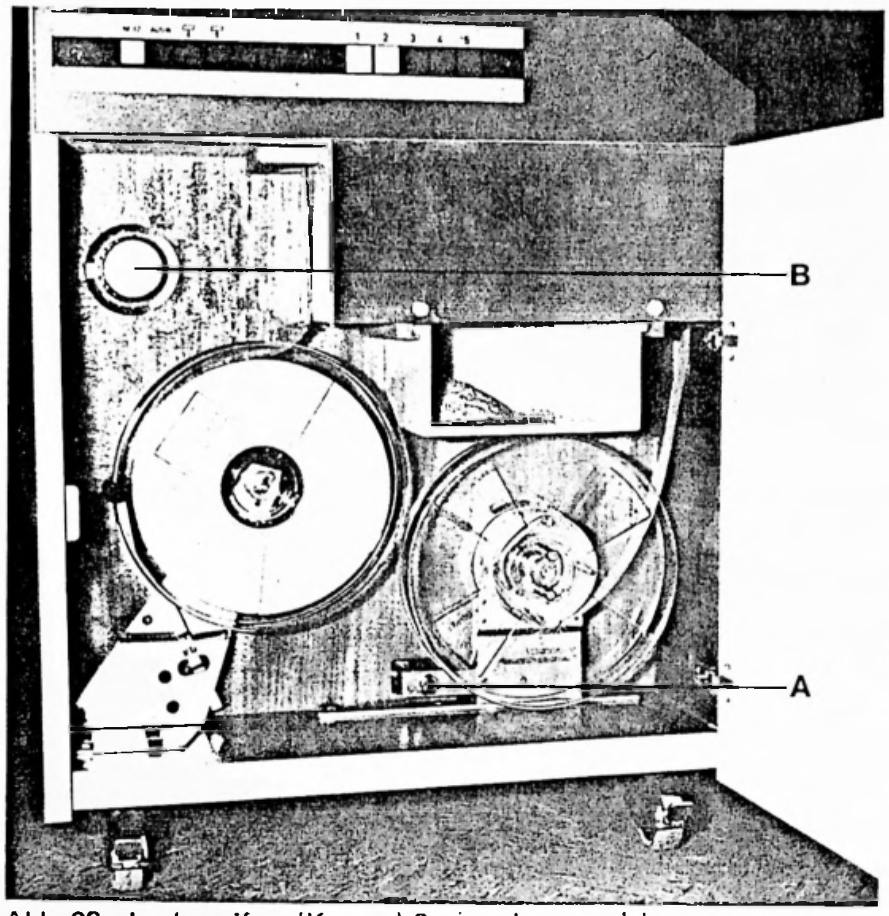


Abb. 23 Lochstreifen- (Karten-) Stanzer-Innenansicht

Am Wahlschalter (B) sind folgende 4 Arbeitsmodi einstellbar:

Schalterstellung 1: Datenaufzeichnung auf Lochstreifen, mit variabler Wortlänge.

Schalterstellung 2: Datenaufzeichnung auf Lochstreifen, mit konstanter Wortlänge.

Schalterstellung 3: Datenaufzeichnung auf Lochstreifenkarten mit konstanter Wortlänge

Schalterstellung 4: Datenaufzeichnung auf Lochstreifenkarten,
mit konstanter Wortlänge für TA 100-Leser (SKE)

Der jeweils eingestellte Arbeitsmodus ist am Sichtfenster des Drehschalters angezeigt (1, 2, 3, 4). Ein falsch eingestellter Arbeitsmodus führt grundsätzlich zu einem falsch erstellten Datenträger. Wahlschalter nur bei ausgeschaltetem Lochstreifen-/ Lochstreifenkartenlocher und ausgeschalteter Grundmaschine betätigen.

An dem im Netzgerät angeordneten Spannungswähler (C) Abb. 24 können folgende Betriebsspannungen eingestellt werden:

220/130/240 oder 110 Volt

Werkseitig sind 220 Volt eingestellt.

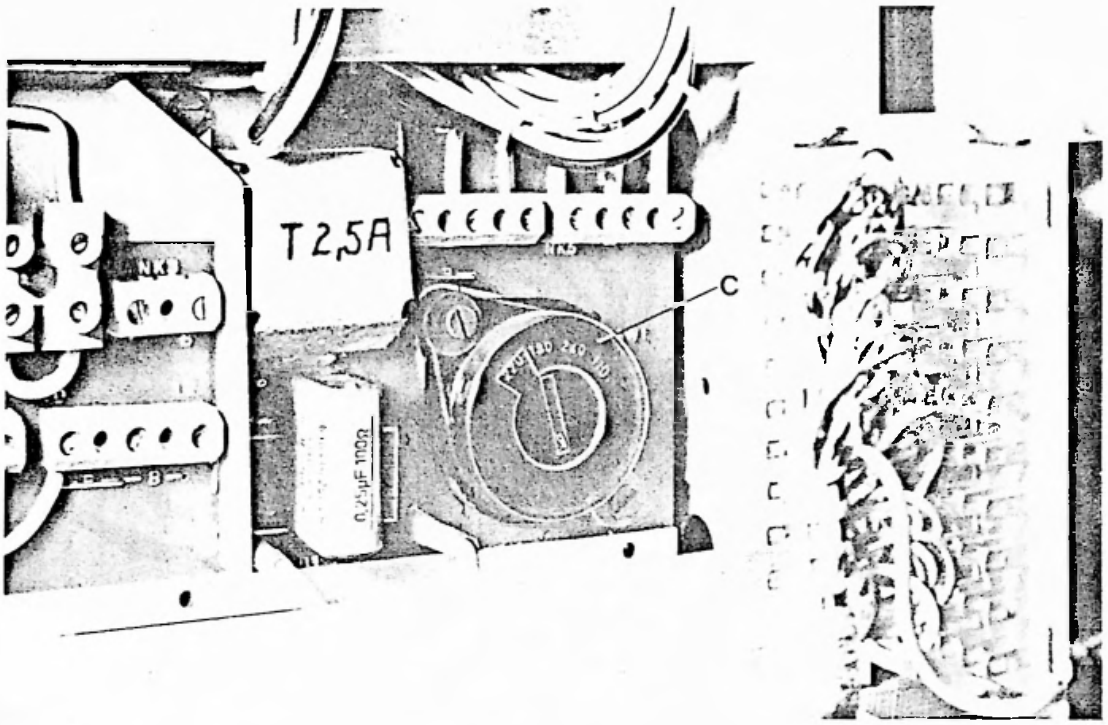


Abb. 24 LSKA-Netzgerät mit Spannungswähler (C)

Da sämtliche Gleichspannungen elektronisch stabilisiert sind, können Netzschwankungen innerhalb der Toleranzen + 10% und - 15% ausgeglichen werden.

Der Lochstreifen- (Karten-)stanzer kann wahlweise für 50 oder 60 Hertz ausgerüstet werden.

Die Leistungsaufnahme beträgt etwa 220 Watt.

7.3 FUNKTIONSEINHEITEN

7.3.1 Codierer

Im Elektronikkasten des Beistelltisches befindet sich die Schaltplatte (AW1), die den Codierer für den Informationsaustausch zwischen Grundmaschine und Peripheriegerät darstellt.

Auf dieser Platte ist der Ausgabecode für die Ziffern 0 - 9 mit Dioden eingestellt.

Werkseitig ist dafür der 7-Bit-Code (nach DIN 66003, 7-Bit-Code; Informationsverarbeitung; Ausgabe Juli 1968) mit Prüfbit (b8) (nach DIN 66 004, Blatt 1, Darstellung des 7-Bit Code auf Datenträgern, Lochstreifen, Informationsverarbeitung; Ausgabe März 1967) im Codierer eingestellt (siehe Abb. 24)

Beliebig andere 7/8-Bit-Code können nachträglich eingestellt werden.

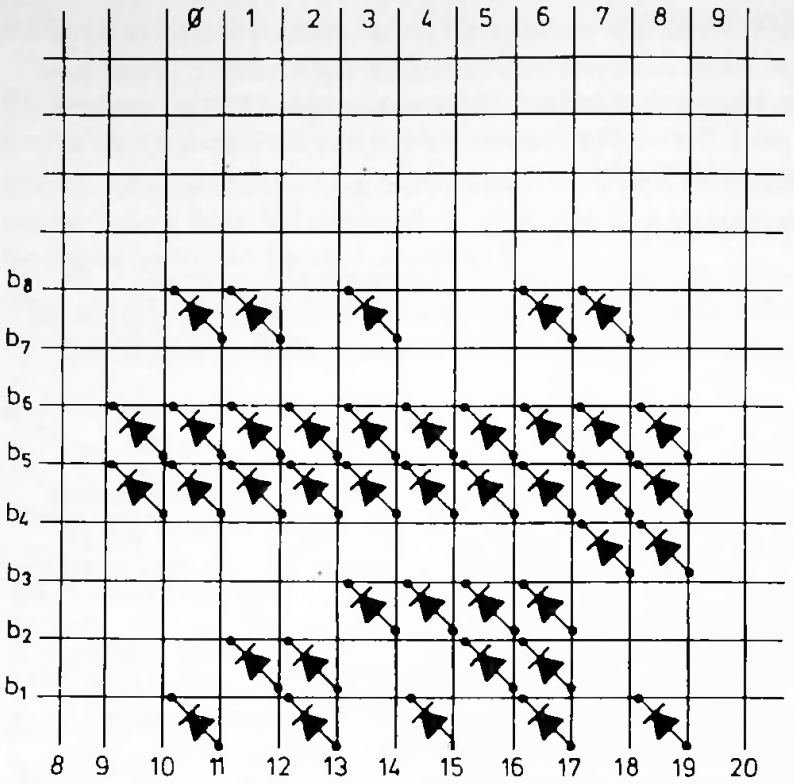


Abb. 25 Darstellung 7-Bit Code im Codierer

7.3.2 KENNZEICHENMATRIX

Ferner ist im Elektronikkasten die Schaltplatte AW 2 zu finden, auf der die Kenn- und Ergänzungszeichen für die Ausgabe programmiert sind.

Maximal kann der Code für 12 verschiedene Zeichen eingestellt werden:

1. Streifenlauf-Codierung
2. \+ 1-stellig
3. — 1-stellig
4. % 2-stellig
5. ‰ 2-stellig
6. \+ 2-stellig
7. — 2-stellig
8. frei
9. Blocklückenzeichen
10. führende Ergänzungszeichen
11. führende Ergänzungszeichen (Blanks)
12. schleppende Ergänzungszeichen
13. % oder ‰ 2-stellig

Die Ansteuerung dieser Zeichen ist teilweise abhängig von dem programmierten Hauptsteuerbefehl und der eingestellten Wahlschalterstellung.

Zu Schalterstellung 1 (variable Wortlänge):

Führende Ergänzungszeichen werden in dieser Schalterstellung nicht gelocht, jedoch gegebenenfalls schleppende. Der Lochvorgang beginnt mit dem ersten von „Nichts" verschiedenen Zeichen, das im CR steht. Die Abarbeitung der Adresse beginnt jedoch mit dem CR-Abgriff gemäß dem programmierten KB-Fall.

Zu Schalterstellung 2 und 3 (konstante Wortlänge):

Bei diesen Schalterstellungen werden führende und gegebenenfalls auch schleppende Ergänzungszeichen gemäß den auf der Kennzeichenmatrix eingestellten Codierungen gelocht. Die Abarbeitung der Adresse beginnt mit dem CR-Abgriff gemäß dem programmierten KB-Fall.

Zu Schalterstellung 4 (konstante Wortlänge)

Bei dieser Schalterstellung werden führende Blanks und gegebenenfalls auch schleppende Ergänzungszeichen gemäß den auf der Kennzeichenmatrix eingestellten Codierungen gelocht. Die Abarbeitung der Adresse beginnt mit dem CR-Abgriff gemäß dem programmierten KB-Fall.

Während bei den Schalterstellungen 1 und 2 (Lochstreifen) die Betätigung der Taste (⏏) einen Lochstreifentransport für die Dauer der Betätigung bewirkt, verursacht bei den Schalterstellungen 3 und 4 die Betätigung der Taste (⏏) (nur nach dem Einschalten der LSKA oder in der Kartengrundstellung) einen Lochstreifenkartenwechsel (Transport bis zum nächsten Ansprechen des Kartenanfangskontaktes). Grundsätzlich ist ein Karten- oder Streifentransport nur möglich, wenn kein HSB ansteht.

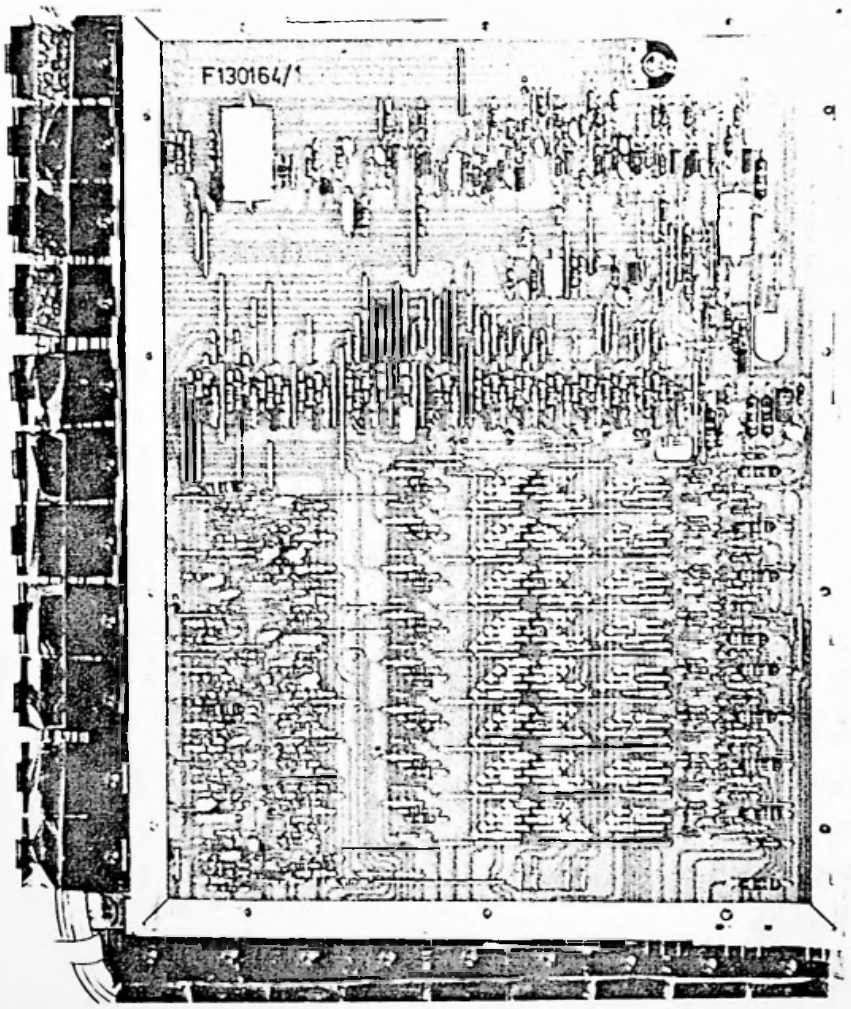


Abb. 26 Schaltplatte AW 2 (Kennzeichenmatrix)

Vorgang der Codeeinstellung für die Kennzeichen und weiteren Sonderzeichen:

Die für die einzelnen Zeichen gewünschten Codierungen sind darauf in Form von Dioden eingelötet. Hierzu ist jedoch folgendes zu beachten:

Die den jeweiligen Spuren (des Lochstreifens) zugeordneten Dioden pro Kennzeichen sind aus der Tabelle Seite 126 zu entnehmen. Für jedes Kennzeichen sind die Dioden zu setzen, die den zu lochenden Spuren zugeordnet sind.

Dabei bewirkt jede gesetzte Diode beim Abruf des Kennzeichens eine Lochung in der der Diode zugeordneten Spur.

Spur / Kennzeichen	1	2	3	4	5	6	7	8
Streifenlauf-Codierung	199	202	205	208	211	214	217	220
+ 1-stellig	200	203	206	209	212	215	218	221
– 1-stellig	128	138	148	158	168	178	188	198
% 2-stellig	126	136	146	156	166	176	186	196
‰ 2-stellig	120	130	140	150	160	170	180	190
+ 2-stellig	122	132	142	152	162	172	182	192
– 2-stellig	124	134	144	154	164	174	184	194
frei	119	129	139	149	159	169	179	189
Blocklückenzeichen	121	131	141	151	161	171	181	191
führende Ergänzungs-zeichen	123	133	143	153	163	173	183	193
führende Ergänzungs-zeichen (Blanks)	125	135	145	155	165	175	185	195
schleppende Ergänzungs-zeichen	127	137	147	157	167	177	187	197
% oder ‰	227	228	229	230	231	232	233	234

Tabelle 2: Kennzeichendioden

(Am Kreuzungspunkt von Spur und Kennzeichen steht jeweils die Diodennummer. Diese Diodennummern sind auch auf der Schaltplatte AW 2 aufgedruckt und nach Bedarf dort zu setzen).

Werkseitig sind in der Kennzeichenmatrix nachstehende Codierungen eingestellt (7-Bit-Code nach DIN 66 003).

Kennzeichen	Spuren des (r) Lochstreifen (—karte) **) 1	2	3	4	5	6	7	8
Streifenlauf-Codierung	X	X	X	X	X	X	X	X
+ 1-stellig *)	X	X		X		X		
— 1-stellig *)	X		X	X		X		
% 2-stellig	X		X			X		X
‰ 2-stellig							X	X
frei								
Blocklückenzeichen	X	X	X		X			
führ. Ergänzungszeichen					X	X		
führ. Ergänzungszeichen (Blanks)						X		X
schlepp. Ergänzungszeichen					X	X		
% oder ‰	X	X	X	X	X	X	X	X

*) Für 1-stellige und 2-stellige Kennzeichnung

**) Die angekreuzten Spuren sollen gelocht werden

Die gesetzen Dioden sind in der folgenden Tabelle eingetragen; nicht eingetragene Dioden brauchen nicht gesetzt zu werden.

Spur / Kennzeichen	1	2	3	4	5	6	7	8
Streifenlauf-Codierung	199	202	205	208	211	214	217	220
+ 1-stellig	200	203		209		215		
— 1-stellig	128		148	158		178		
% 2-stellig	126		146			176		196
‰ 2-stellig							180	190
+ 2-stellig	122	132		152		172		
— 2-stellig	124		144	154		174		
frei								
Blocklückenzeichen	121	131	141		161			
führ. Ergänzungszeichen					163	173		
führ. Ergänzungszeichen (Blanks)						175		195
schlepp. Ergänzungszeichen					167	177		
% oder ‰	227	228	229	230	231	232	233	234

Ansteuerung der Kenn- und Ergänzungszeichen

Schalterstellung / Kenn-Ergänzungszeichen	1	2	3	4
Streifenlauf-Codierung				
Kennzeichen + 1-stellig	PU 1	PU 1	PU 1	PU 1
Kennzeichen – 1-stellig	PU 1	PU 1	PU 1	PU 1
Kennzeichen % 2-stellig	PU 2	PU 2	PU 2	PU 2
Kennzeichen ‰ 2-stellig	PU 2	PU 2	PU 2	PU 2
Kennzeichen + 2-stellig	PU 2	PU 2	PU 2	PU 2
Kennzeichen – 2-stellig	PU 2	PU 2	PU 2	PU 2
Blocklückenzeichen	GAP	GAP	GAP	GAP
führende Ergänzungszeichen		PU 0 PU 1 PU 2	PU 0 PU 1 PU 2	
führende Ergänzungszeichen (Blanks)				PU 0 PU 1 PU 2
schleppende Ergänzungszeichen	PU 0 PU 1 PU 2	PU 0 PU 1 PU 2	PU 0 PU 1 PU 2	PU 0 PU 1 PU 2
$\overline{\%}$ oder $\overline{‰}$ 2-stellig	PU 2	PU 2	PU 2	PU 2

7.3.3 SCHLÜSSELSCHALTER

An den Modellen TA 10/2 und TA 10/3 sind für ein Peripherieprogramm mit dem in der Sondertastatur angeordneten Schlüsselschalter zwei unterschiedliche Arbeitsmodi einstellbar.

- Abarbeiten des Programmes **mit Peripherieausgabe**
- Abarbeiten des Programmes mit oder ohne gestecktem Peripheriegerät **ohne Peripherieausgabe.**

In der Schalterstellung „EIN" muß ein peripheres Gerät gesteckt und eingeschaltet sein, ist dies nicht der Fall, kommt das Programm beim ersten Niederschrifts- oder Peripheriebefehl zum Stillstand.

In der Schalterstellung „AUS" kann das periphere Gerät gesteckt und eingeschaltet sein; Peripheriebefehle werden im Programmablauf wirkungslos.

Am Modell TA 10/1 sind diese beiden Arbeitsmodi im Schreibwerk an der Schreibwerkplatte (Platte 1) einstellbar.

7.3.4 DATENSICHERUNG

Beim Erstellen des Lochstreifens wird bei der Ausgabe jedes einzelnen Zeichens automatisch geprüft, ob die Anzahl der belegten Binärstellen gerade oder ungerade ist.

Gerade Code werden auf ungerade und ungerade Code werden auf gerade Parität geprüft.

Dafür sind im Codierer und in der Kennzeichenmatrix die einzelnen Codeelemente in der Prüfspur (b8) auf eine gerad- oder ungeradzahlige Codeelementeanzahl zu ergänzen.

Die Einstellung, ob auf gerade oder ungerade Parität geprüft werden soll, ist auf der Kennzeichenmatrix (AW 2) vorzunehmen (Abb. 27)

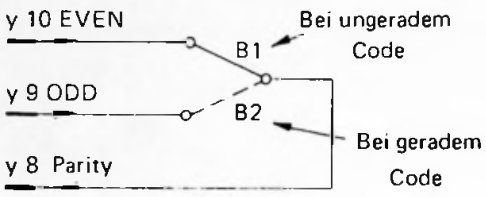


Abb. 27 Kennzeichenmatrix (AW 2)

Für einen ungeraden Code, der auf gerade Parität geprüft werden soll, ist die Brücke B 1 und für einen geraden Code, der auf ungerade Parität geprüft werden soll, ist die Brücke B2 einzulöten.

Werkseitig ist die Brücke B 1 eingelötet und im Codierer und in der Kennzeichenmatrix sind die Codeelemente auf eine geradzahlige Anzahl ergänzt.

Wird bei diesen Einstellungen während des Lochens ein ungeradzahliges Zeichen erkannt, so wird vom Stanzer **Paritätsfehler** (s. 7.3.5) gemeldet.

7.3.5 MÖGLICHE FEHLERFÄLLE

Beim Lochstreifen-(Karten-)Stanzer können folgende Fehler auftreten:

- **Dateneiende:**
  Dateiende wird durch den mechanischen Auflagekontakt an der Abwicklungsspule gemeldet und bedeutet: keine Lochstreifenrolle eingelegt, oder der Lochstreifen ist zu Ende.

  Er ist durch Einlegen einer neuen Rolle und anschließendem Betätigen der Fehlerbehebungstasten (1. Fehlerbehebungstaste am peripheren Gerät und 2. Fehlerbehebungstaste in der Sondertastatur oder in der Schreibwerkstastatur betätigen) zu beheben. Dateiende wird immer erst nach dem Abarbeiten des jeweils anstehenden Peripheriebefehls gemeldet.

- **Peripheriefehler:**
  Peripheriefehler bedeutet, daß ein Zeichen unvollständig gelocht wurde. Er wird sofort nach dem Erkennen des fehlerhaften Zeichens gemeldet.

  Dieser Fehler bewirkt das Setzen eines Selectors, der von einem bedingten Sprungbefehl programmabhängig abgefragt werden kann (Sprung in eine Fehlerroutine). Behoben wird der Fehler durch Betätigen der in der Sondertastatur oder in der Schreibwerktastatur angeordneten Fehlerbehebungstaste die einen Wiederstart des Programmes bewirkt.

– Transporthemmung:
Der Fehler Transporthemmung bedeutet, daß der Lochvorgang gestört ist (z.B. die Stanzstempel sind verklemmt oder verkantet). Zu beheben ist der Fehler, indem versucht werden muß, evtl. Papierreste in der Stanzstempelbahn zu entfernen.

Da dieser Fehler ebenfalls den Selector setzt, ist die Fehlerbehebung wie im Paritätsfehlerfalle vorzunehmen.

Bei allen Fehlerfällen werden an den Grundmaschinen die Eingabetastaturen gesperrt. Sie werden erst nach den jeweiligen Fehlerbehebungen wieder freigegeben.

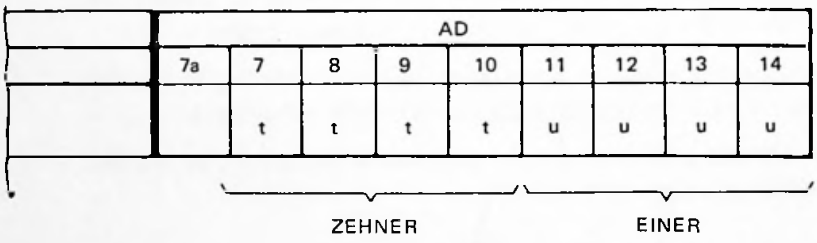
7.4 PERIPHERIEBEFEHLE

In den Adreßteilen der Lochbefehle sind die Anzahl der jeweils abzulochenden Zeichen im 3-Exzeß-Code zu programmieren.

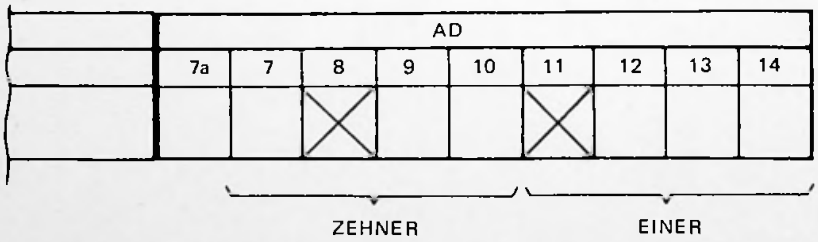
Die Verschlüsselungen der Dezimalziffern im 3-Exzeß-Code sind nachstehender Tabelle zu entnehmen.

Dezimalziffer	3-Exzeß-Code
0	0011
1	0100
2	0101
3	0110
4	0111
5	1000
6	1001
7	1010
8	1011
9	1100

Für die Darstellung 2-stelliger Dezimalziffern ist der Adreßteil in zwei Gruppen unterteilt:



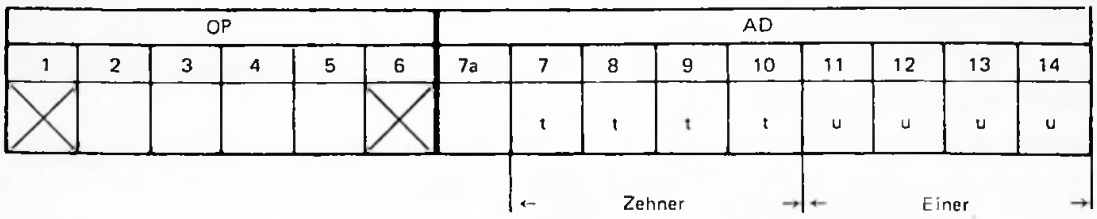
Soll z.B. die Dezimalzahl 15 im 3-Exzeß-Code dargestellt werden, so ist wie folgt zu programmieren:



Für die 0-Aussagen sind keine Binärstellen zu belegen, sie werden von der Maschine automatisch generiert.

PUO

7.4.1 PUNCH WITH O SIGN



Symbolischer Befehlscode: PUO/t/u

Wirkung

Dieser Befehl bewirkt „Lochen numerisch ohne Kennzeichen".

Die Anzahl der aus dem CR abzulochenden, numerischen Zeichen ist im Adreßteil im 3-Exzeß-Code in zwei Gruppen (ZEHNER (t) und EINER (u)) zu programmieren.

Maximal können 15 numerische Zeichen aus dem CR ausgegeben werden.

Auftretende Kennzeichen (wie +, −, %, ‰) werden hierbei nicht gelocht. Ebenso werden keine Leerschritte auf den Datenträger übernommen.

Im Programmablauf ist dieser Befehl vor den jeweiligen Niederschriftsbefehl (OUT oder OUK) zu programmieren.

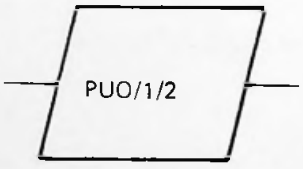
Wird ein Lochbefehl nach einem Niederschriftsbefehl programmiert, so geht die abzulochende Information dem Locher verloren.

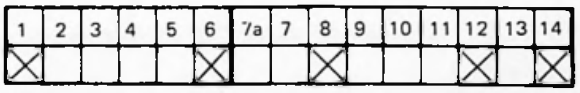
Die im Niederschriftsbefehl programmierte Kapazität bestimmt, ab welcher Dekade im CR die Abarbeitung der in den PU-Befehlen programmierten Adressen beginnt.

Ist die im PU 0 angegebene Adresse kleiner als die programmierte Kapazität, so werden die wertniedrigsten Stellen der im CR stehenden Information nicht mitgelocht.

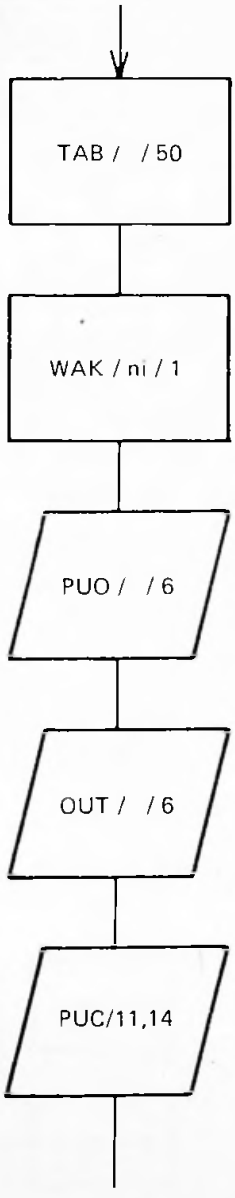
Ist die im PU 0 angegebene Adresse größer als die programmierte Kapazität, so werden schleppende Ergänzungszeichen gelocht.

Beispiele:





Nachstehendes Beispiel zeigt den PU 0-Befehl in richtig programmierter Befehlsreihenfolge:



PUO

Weitere Beispiele:

Wert im Speicher:

Schriftbild bei KB 8 1 2 3 4 5 6

Variable Wortlänge (Schalterstellung 1):

Lochung bei PU0/ /4 1 2

Lochung bei PU0/ /6 1 2 3 4

Lochung bei PU0/ /8 1 2 3 4 5 6

Lochung bei PU0/1/0 1 2 3 4 5 6 Ø Ø

Konstante Wortlänge (Schalterstellung 2 oder 3):

Lochung bei PU0/ /6 Ø Ø 1 2 3 4

Lochung bei PU0/ /8 Ø Ø 1 2 3 4 5 6

Lochung bei PU0/1/0 Ø Ø 1 2 3 4 5 6 Ø Ø

Konstante Wortlänge (Schalterstellung 4):

Lochung bei PU0/ /6 ƀ ƀ 1 2 3 4

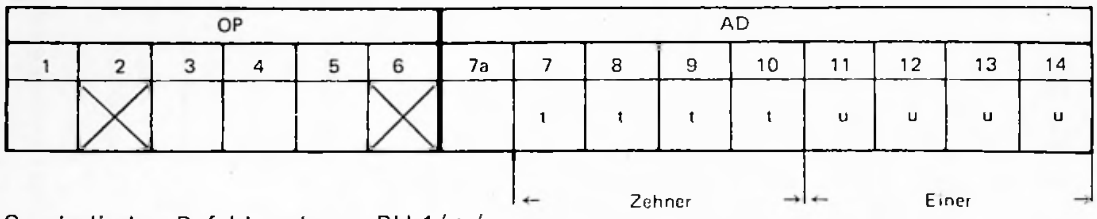
Lochung bei PU0/ /8 ƀ ƀ 1 2 3 4 5 6

Lochung bei PU0/1/0 ƀ ƀ 1 2 3 4 5 6 Ø Ø

Ø = führende bzw. schleppende Ergänzungszeichen

ƀ = führende Blanks

7.4.2 PUNCH WITH 1 SIGN



Symbolischer Befehlscode: PU 1/ t / u

Wirkung:

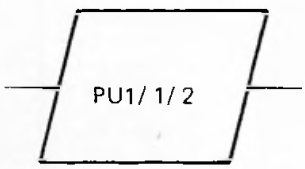
Dieser Befehl bewirkt „Lochen numerisch mit 1 Kennzeichen".

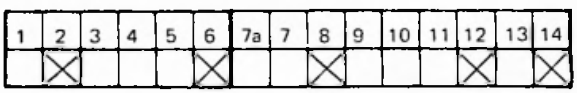
Im Adreßteil dieses Befehles ist dazu die Anzahl der aus dem CR zu lochenden numerischen Zeichen plus 1 (das Kennzeichen muß in die Adresse mit einbezogen werden) im 3-Exzeß-Code, in 2 Gruppen ZEHNER (t) und EINER (u) zu programmieren.

Das Kennzeichen (je nach Speicherwert das Vorzeichen + oder – (1-stellig)) wird im Anschluß an die Ziffern gelocht. Die Codemuster der Kennzeichen sind in der Kennzeichenmatrix eingestellt.

Ansonsten gilt für diesen Befehl das Gleiche wie das bei PU0 Gesagte.

Beispiele:

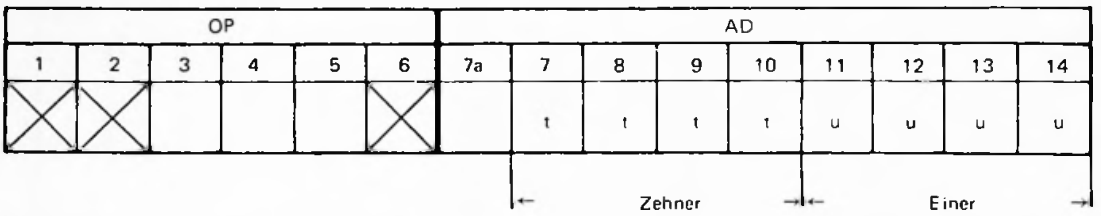




Nachstehendes Beispiel zeigt den PU 1-Befehl in richtig programmierter Befehlsreihenfolge.

PU 2

7.4.3 PUNCH WITH 2 SIGN



Symbolischer Befehlscode: PU 2/t/u

Wirkung

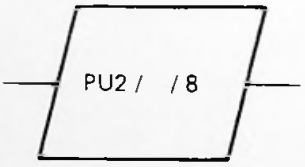
Dieser Befehl bewirkt „Lochen numerisch mit 2 Kennzeichen.

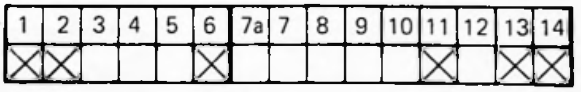
Im Adreßteil dieses Befehles ist dazu die Anzahl der aus dem CR zu lochenden numerischen Zeichen plus 2 (die Kennzeichen müssen in die Adresse mit einbezogen werden) im 3-Exzeß-Code, in 2 Gruppen ZEHNER (t) und EINER (u) zu programmieren. Hierbei werden im Anschluß an die Ziffern die Kennzeichen entsprechend den in der Kennzeichenmatrix eingestellten Codierungen wie folgt gelocht:

Kennzeichen des Speicherwertes (gemäß der gesetzten Kennzeichen –FF's)	Abgelochte Kennzeichen (gem. der in der Matrix eingestellten Codierung)	
	1. Kennzeichen	2. Kennzeichen
keine	+	$\overline{\%}$ oder $\overline{‰}$
–	–	$\overline{\%}$ oder $\overline{‰}$
%	+	%
–, ‰	–	‰

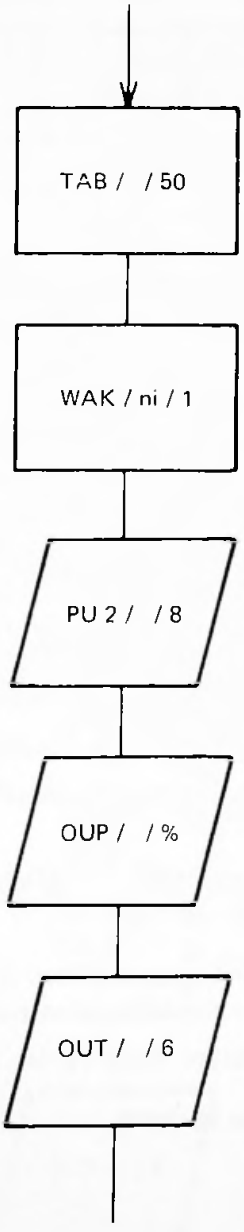
Ansonsten gilt auch für diesen Befehl das Gleiche wie das bei PU0 Gesagte.

Beispiele:



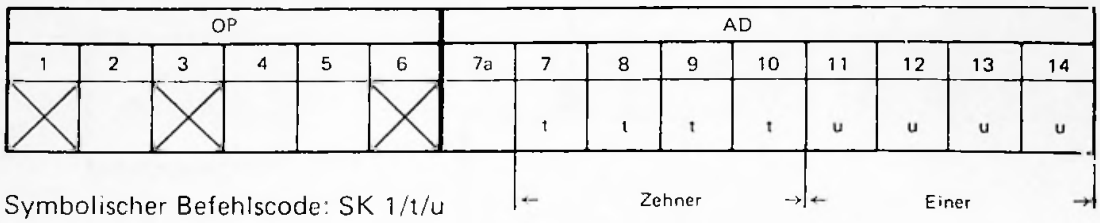


Nachstehendes Beispiel zeigt den PU 2-Befehl in richtig programmierter Befehlsreihenfolge:



SK 1

7.4.4 SKIP ON RECORD MEDIUM



Symbolischer Befehlscode: SK 1/t/u

Wirkung:

Dieser Befehl darf nur dann programmiert werden, wenn im Arbeitsmodus „Datenaufzeichnung auf Lochstreifenkarten mit konstanter Wortlänge" (Schalterstellung 3 oder 4) gearbeitet wird.

Er bewirkt, daß die Lochstreifenkarte bis zu der Sprosse transportiert wird, bei der die im Adreßteil im 3-Exzeß-Code, in zwei Gruppen (ZEHNER (t) und EINER (u)) programmierte Adresse mit dem Zähler übereinstimmt. Ein adreßlos programmierter Befehl bewirkt einen Kartentransport bis zum nächstfolgenden Kartenanfang. Gleichzeitig wird der Absolutzähler (zählt die Sprossen) auf „0" gesetzt.

Das Kriterium für den „Kartenanfang" ist der Kartenanfangskontakt. Da dieser jedoch mechanisch versetzt ist, befindet sich beim Ansprechen des Kontaktes anstelle der ersten Sprosse bereits die dritte Sprosse (bezogen auf den tatsächlichen Kartenanfang) unter der Stanzstation, sodaß sich bezüglich des Zählerstandes eine Differenz von 3 ergibt.

Für die Berechnung der Sprungadresse können zwei Wege beschritten werden:

a) man beginnt das Abzählen der Sprossen an der 3. Sprosse mit „0" und erhält sofort die zu programmierende Sprungadresse.

b) man beginnt das Abzählen der Sprosse mit „1" und erhält die zu programmierende Sprungadresse durch die Subtraktion der Zahl „3".

Da der Locher beim Lochvorgang zuerst transportiert und dann erst ablocht, ist das erste Zeichen, das nach einem Sprungbefehl abgelocht wird, das Zeichen Sprungadresse +1.

Die Lochstreifenkarte kann also frühestens in der vierten Sprosse eine Lochung erhalten.

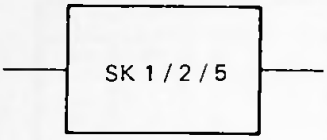
Zu beachten ist, daß der Zähler nur bis 99 zählt.

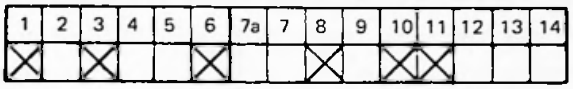
Beispiele:

gegenwärtiger Stand		Sprungbefehl	Stand nach Sprung		nächste
Sprosse	Zähler		Sprosse	Zähler	lochbare Sprosse
24	21	SK1/2/3	26	23	27
24	21	SK1/4/5	48	45	49
24	21	SK1/ /0	3*	0*	4

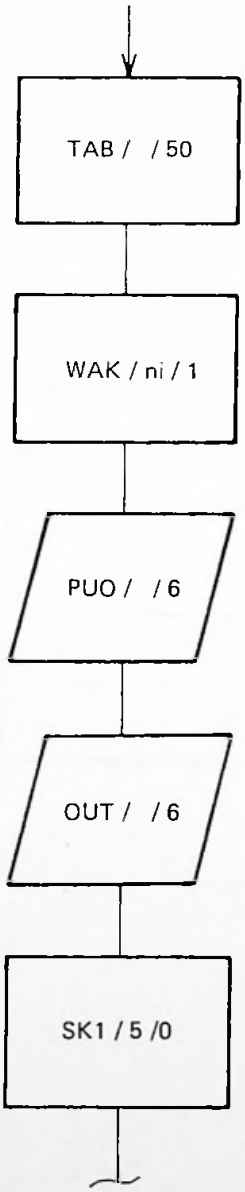
* = nächste Karte

Beispiele:



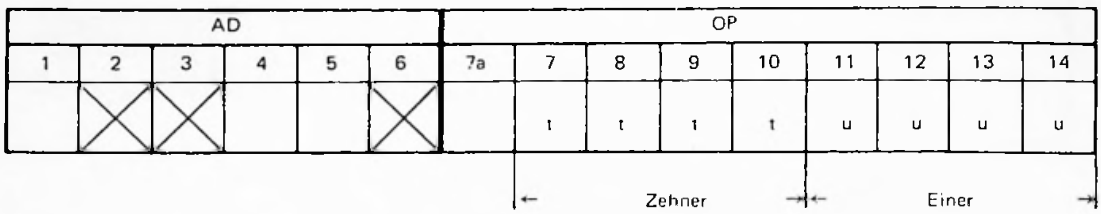


Nachstehendes Beispiel zeigt den SK 1-Befehl in richtig programmierter Befehlsreihenfolge:



GAP

7.4.5 RECORD GAP



Symbolischer Befehlscode: GAP / t / u

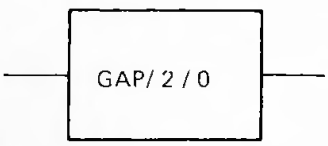
Wirkung

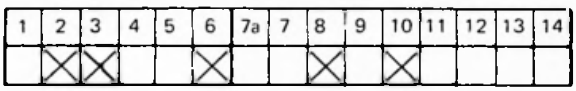
Dieser Befehl bewirkt, daß die im Adreßteil im 3-Exceß-Code in zwei Gruppen (ZEHNER (t) und EINER (u)) programmierbare Anzahl Zeichen am Datenträger gelocht werden.

Das abzulochende Bitmuster des Zeichens ist in der Kennzeichenmatrix auf dem Codeplatz „Blocklückenzeichen" eingestellt.

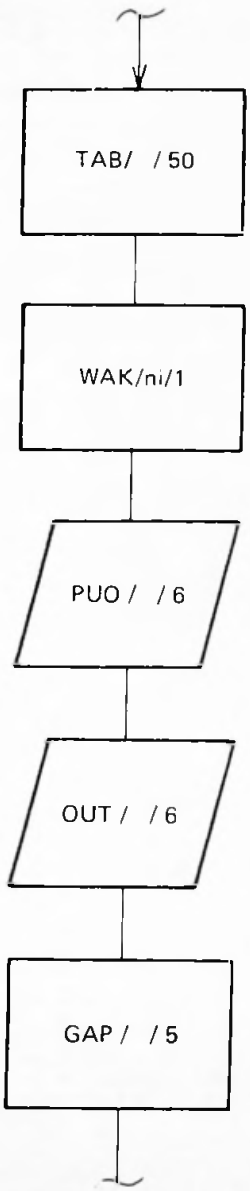
Innerhalb des Programmablaufes ist dieser Befehl zweckmäßigerweise ebenfalls nach den jeweiligen Niederschriftsbefehlen zu programmieren.

Beispiele





Nachstehendes Beispiel zeigt den GAP-Befehl in richtig programmierter Befehlsreihenfolge:



PUC

7.4.6 PUNCH CONSTANT

OP						AD								
1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
X	X	X			X		tr 4	tr 3	tr 2	tr 1	tr 8	tr 7	tr 6	tr 5

Symbolischer Befehlscode: PUC /7-14

Wirkung

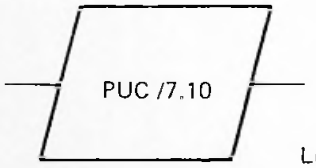
Dieser Befehl bewirkt das Ablochen des im Adreßteil programmierten Bit-Musters.

Die Binärstellen sind den Datenspuren wie folgt zugeordnet:

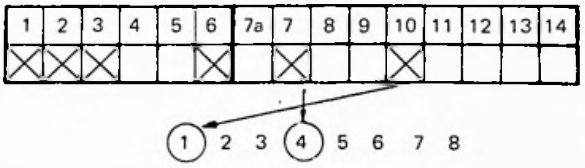
Beispiele:

	AD								
	7a	7	8	9	10	11	12	13	14
		tr4	tr 3	tr 2	tr 1	tr 8	tr 7	tr 6	tr 5

Lochstreifenspur:



Lochstreifenspur



No.	Mnemonics	1	2	3	4	5	6	7a	7	8	9	10	11	12	13	14
		Operating part						Adreß part								
1Ø	TRV	x			x				–	vr7	vr6	vr5	vr4	vr3	vr2	vr1
1Ø	TRV	x			x				–				vr8	vr4	vr2	vr1
2Ø	BRV	x							•	vr7	vr6	vr5	vr4	vr3	vr2	vr1
2Ø	BRV	x							•				vr8	vr4	vr2	vr1
3Ø	ADD	x	x						–	vr7	vr6	vr5	vr4	vr3	vr2	vr1
3Ø	ADD	x	x						–				vr8	vr4	vr2	vr1
4	MUL	x		x					1.f.	$\overline{ro}$			sr8	sr4	sr2	sr1
5	WAK	x				x			ni					ky4	ky2	ky1
6	MAK					x			ni		m4	m2	m1	ky4	ky2	ky1
7Ø	MAT	x	x	x						vr7	vr6	vr5	vr4	vr3	vr2	vr1
7Ø	MAT	x	x	x									vr8	vr4	vr2	vr1
8	JUN		x					ps256	ps128	ps64	ps32	ps16	ps8	ps4	ps2	ps1
9	JM1		x			x		ps256	ps128	ps64	ps32	ps16	ps8	ps4	ps2	ps1
10	JM2		x		x			ps256	ps128	ps64	ps32	ps16	ps8	ps4	ps2	ps1
11	JM3		x		x	x		ps256	ps128	ps64	ps32	ps16	ps8	ps4	ps2	ps1
12	JM4		x	x				ps256	ps128	ps64	ps32	ps16	ps8	ps4	ps2	ps1
13	JNE		x	x		x		ps256	ps128	ps64	ps32	ps16	ps8	ps4	ps2	ps1
14	JNZ		x	x	x			ps256	ps128	ps64	ps32	ps16	ps8	ps4	ps2	ps1
15	JSE			x	x	x		ps256	ps128	ps64	ps32	ps16	ps8	ps4	ps2	ps1
16	NOP															
17	TAB			x					te	pc128	pc64	pc32	pc16	pc8	pc4	pc2
18	TAK			x		x				pc128	pc64	pc32	pc16	pc8	pc4	pc2
19	OUP			x	x				ac	np			sy8	sy4	sy2	sy1
20	OUT				x				sp6	sp4	sp3	sp2	cp8	cp4	cp2	cp1
21	OUK				x	x			sp6	sp4	sp3	sp2	cp8	cp4	cp2	cp1
22	CON	x	x		x				–				co8	co4	co2	co1
23	SKP					x	x							pl	cf2	cf1
24	PUO	x					x		t	t	t	t	u	u	u	u
25	PU1		x				x		t	t	t	t	u	u	u	u
26	PU2	x	x				x		t	t	t	t	u	u	u	u
27	SK1	x		x			x		t	t	t	t	u	u	u	u
28	GAP		x	x			x		t	t	t	t	u	u	u	u
29	PUC	x	x	x			x		tr4	tr3	tr2	tr1	tr8	tr7	tr6	tr5
30	END	x	x	x	x	x	x	ps256	ps128	ps64	ps32	ps16	ps8	ps4	ps2	ps1

Definition:

ac = alphanumeric characters
co = constant
cp = capacity
ky = key
m = mark
ni = numeric input
np = non printing
pc = positions counter
ps = program step
$\overline{ro}$ = no rounding
sp = space
sr = shift right
sy = symbol
t = ten

te = text
tr = track
u = unit
vr = variable register
1.f = constant 1 factor

Command 19: sy 1-9 ≙ 1-9
sy% ≙ 10
sy ‰ ≙ 11
sy0 ≙ 12
sy• ≙ 13
sy^ ≙ 14
syMW ≙ 15

The command 23 is applicable only for model TA 10/2

Command 23: cf 1 ≙ skip line
cf 2 ≙ skip tabulat.
cf 1+2 ≙ skip page
pl = skip platen

Beispiel für
Programmflußdiagramme

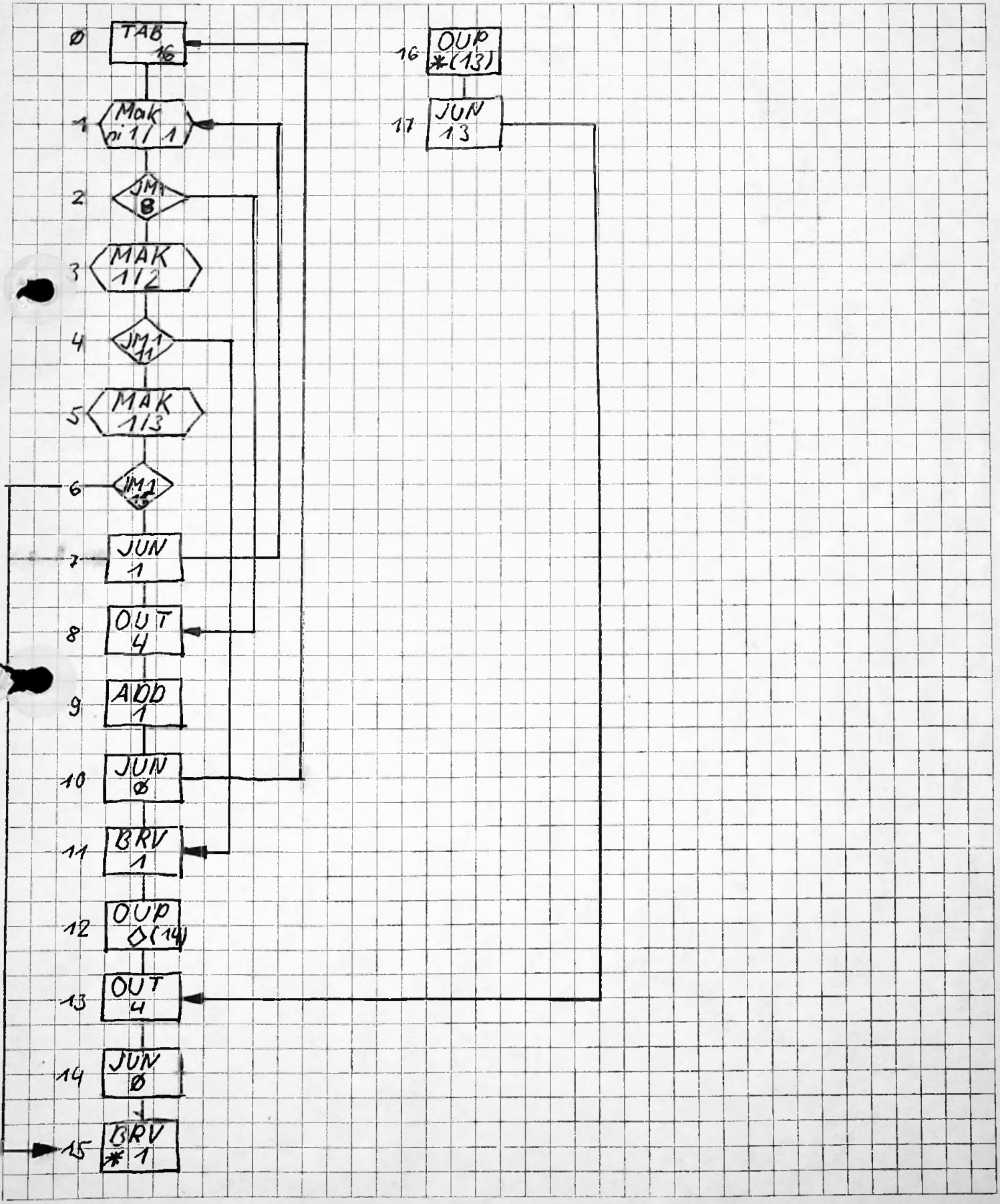
Fakturierprogramm (Schulung) TR 10

Z Nr.

Start
Tab Tlg 10 te
MAK 4/4
4
ja
MAK 1/1
1
JUN
Tab. 40 R.Nr
MAK 1/1 ni
1
OUT Re Nr.
Tab. Tlg. 50
WAK 1 ni Dat
TRV VR1
OUT Dat
Tab. Tlg 10 te
ja
JUN
MAK 2/2 ni
2
JN1
nein
BRV ◇ VR1
JUN
MAK 1/1
1
TAB Tlg. 40
MAK 2/2
2
Tab. Tlg. 62
BRV ◇ VR2
MAK 3/3
3
Tab. Tlg. 52
WAK 2 ni
OUP %
OUT Zuschl.
Mult %
MAK 4/4
4
MAK 5/1
1
Tab Tlg. 38
BRV VR2
JUN
OUP ◇
OUT Σ Betrg.
TRV BR
JUN
MAK 1/1 ni
1
MAK 2/2 ni
2
JUN
OUT Menge
TRV BR
Tab Tlg 52
OUT Menge
TRV BR
Tab Tlg 52
MAK 1/1 ni
1
OUT Preis
Mult *0
Tab Tlg 62
OUT Betrag
ADD VR2
JUN
MAK 2/2 ni
2
OUP %
OUT Preis
Mult *2
JUN
MAK 3/3 ni
3
OUP %o
OUT Preis.
Mult *3
JUN
JUN
OUT Betrag
Tab Tlg 52
CON 1
CON 1
CON 0
CON 0
OUT MW St.
Mult *4
ADD VR2
Tab Tlg 62
BRV VR2 ◇
OUT Betrag
JUN
MAK 1/1 ni
1
OUT Preis
Mult *2
JUN
MAK 2/2 ni
2
OUP %
OUT Preis
Mult *4
JUN
MAK 3/3 ni
3
OUP %o
OUT Preis
Mult *5
JUN
JUN

MH

HORN & GÖRWITZ
Büro-Organisation · Büro-Technik · Büro-Einrichtungen
An der Urania 16-18 · 1000 Berlin 30 · Tel. 21 0 91

KALENDER 1977

	Januar	Februar	März	April	Mai	Juni	Juli	August	September	Oktober	November	Dezember
M	3 10 17 24 31	7 14 21 28	7 14 21 28	4 11 18 25	2 9 16 23 30	6 13 20 27	4 11 18 25	1 8 15 22 29	5 12 19 26	3 10 17 24 31	7 14 21 28	5 12 19 26
D	4 11 18 25	1 8 15 22	1 8 15 22 29	5 12 19 26	3 10 17 24 31	7 14 21 28	5 12 19 26	2 9 16 23 30	6 13 20 27	4 11 18 25	1 8 15 22 29	6 13 20 27
M	5 12 19 26	2 9 16 23	2 9 16 23 30	6 13 20 27	4 11 18 25	1 8 15 22 29	6 13 20 27	3 10 17 24 31	7 14 21 28	5 12 19 26	2 9 16 23 30	7 14 21 28
D	6 13 20 27	3 10 17 24	3 10 17 24 31	7 14 21 28	5 12 19 26	2 9 16 23 30	7 14 21 28	4 11 18 25	1 8 15 22 29	6 13 20 27	3 10 17 24	1 8 15 22 29
F	7 14 21 28	4 11 18 25	4 11 18 25	1 8 15 22 29	6 13 20 27	3 10 17 24	1 8 15 22 29	5 12 19 26	2 9 16 23 30	7 14 21 28	4 11 18 25	2 9 16 23 30
S	1 8 15 22 29	5 12 19 26	5 12 19 26	2 9 16 23 30	7 14 21 26	4 11 18 25	2 9 16 23 30	6 13 20 27	3 10 17 24	1 8 15 22 29	5 12 19 26	3 10 17 24 31
S	2 9 16 23 30	6 13 20 27	6 13 20 27	3 10 17 24	1 8 15 22 29	5 12 19 26	3 10 17 24 31	7 14 21 28	4 11 18 25	2 9 16 23 30	6 13 20 27	4 11 18 25