D I P L O M A R B E I T

Autonomes bildschirmorientiertes PAL - Programmiergerät

M. Boretius & P. Zecherle SS 83

FACHHOCHSCHULE DARMSTADT
FACHBEREICH ELEKTROTECHNIK

D I P L O M A R B E I T

THEMA :

Autonomes bildschirmorientiertes
PAL - Programmiergerät

Fachrichtung : Automatisierungstechnik

Semester : SS 1983

Vorgelegt von : Boretius, Michael
Zecherle, Peter

Betreuender Dozent: Professor Dipl.Ing. Blecken, Klaus

Darmstadt, den 19.06.83

Die Unterzeichner erklären hiermit, daß sie die vorliegende Arbeit selbständig und nur unter Zuhilfenahme der im Anhang Q angegebenen Literatur angefertigt haben

Boretius Zecherle

I N H A L T S V E R Z E I C H N I S

	Seite
1. Einleitung	1
1.1 Beschreibung eines PAL	1
1.2 Innerer Aufbau, Funktion	7
1.3 Programmierinformationen	9
2. Analyse der Aufgabenstellung	11
2.1 Pflichtenheft	11
3. Realisierung der Hardware	12
3.1 Realisierung der logischen Zustände	12
3.1.1 Konzeptfindung	12
3.1.1.1 Anwahl einer Diodensicherung	12
3.1.1.2 Lösungsansatz aufschalten logischer Zustände	15
3.1.1.3 Konzeption der Spannungsquelle	20
3.1.1.4 Realisierung mit Schaltregler	21
3.1.1.5 Realisierung mit Linearregler	22
3.1.2 Berechnung der Beschaltung des L 200	23
3.2 Gesamtkonzeption	24
3.2.1 Konzeption der CPU-karte	24
3.2.1.1 Zeitkritische Abschnitte	24
3.2.1.2 Steuerung des Programmierpulses	25
3.2.2 Beschreibung des Einplatinenkomputers	26
3.2.3 Änderungen und Erweiterungen	26
3.3 Aufbau der Programmiereinrichtung	28
3.3.1 Beschreibung der Interfacekarte	28
3.3.1.1 Decodierung	28
3.3.1.2 Ansteuerung des L 200	29
3.3.1.3 Portexpanersteuerung	32
3.3.2 Beschreibung der Treiberkarte	35
3.3.2.1 Beschaltung der Pins 1 - 11 des PAL	35
3.3.2.2 Beschaltung der Pins 14 - 23 des PAL	38

Seite

4. Realisierung der Software 42
4.1 Konzeptfindung 42

4.1.1 Ein-Ausgabe Paket 43
4.1.1.1 Erweiterte Aufgabenstellung 43
4.1.1.2 Entwicklungsgang 43

4.1.2 Editor 44
4.1.2.1 Erweiterung des Pflichtenhefts 44
4.1.2.2 Vorgehensweise bei der Entwicklung des Editors 44

4.1.3 Monitor 45
4.1.3.1 Erweitertes Pflichtenheft 45
4.1.3.2 Abfolge der Entwicklung 45

4.1.4 PAL Betriebssystem 46
4.1.4.1 Erweiterung des Pflichtenhefts 46

4.2 Erläuterung zu den Softwarepaketen 49

4.2.1 Erläuterung zum Ein- Ausgabepaket 49
4.2.1.1 Ablaufdiagramm "MOD" 56
4.2.1.2 Ablaufdiagramm "XFUNCT" 57
4.2.1.3 Ablaufdiagramm "MSKTS" 58

4.2.2 Erläuterungen zum Editor 59
4.2.2.1 Ablaufdiagramm "Editor" (Hauptschleife) 61
4.2.2.2 Ablaufdiagramm "Editor" (Insert&Delete) 62
4.2.2.3 Ablaufdiagramm "Editor" (Cursor links) 63
4.2.2.4 Ablaufdiagramm "Editor" (Cursor rechts) 63
4.2.2.5 Ablaufdiagramm "Editor" (löschen Zeile) 64
4.2.2.6 Ablaufdiagramm "Editor" (Backspace) 64

4.2.3 Erläuterungen zum Monitor 65
4.2.3.1 Ablaufdiagramm "Monitorhauptschleife" 66
4.2.3.2 Ablaufdiagramm "MODIFY" (Hauptschleife) 67
4.2.3.3 Ablaufdiagramm "MODIFY" (Texteingabe) 68
4.2.3.4 Ablaufdiagramm "INPO" 69
4.2.3.5 Ablaufdiagramm "OUTPO" 69

4.3 Erläuterung zum PAL Betriebssystem 70

4.3.1 Bedienerebene 70
4.3.1.1 Ablaufdiagramm "PAL-Betriebssystem" 73
4.3.1.2 Ablaufdiagramm "FUSE" 74

		Seite
4.3.2	Hardwareebene	75
4.3.2.1	Der Programmierzyklus	75
4.3.2.2	Der Verifizierungszyklus	82
4.3.2.3	Taktdiagramm, Ablaufdiagramm Programmierzyklus	91
4.3.2.4	Alternatives Steuerungskonzept	94
4.3.2.5	Das Programm MATRIX	96
4.4	Speicherbelegungsplan	98
4.4.1	EPROM	99
4.4.2	1. RAM	100
4.4.3	2. RAM	101

		Seite
ANHANG	B	102
B1	Bedienungsanleitung	103
B1.1	Befehle Gesamtübersicht	103
B1.2	Allgemeines	103
B1.3	Bedienung der einzelnen Befehle	103
B1.3.1	Memory modify	103
B1.3.2	Tabulate	107
B1.3.3	Go and Execute Program	108
B1.3.4	Out to Port	109
B1.3.5	In from Port	110
B1.3.6	Userinstruction	110
B2	Bedienungsanleitung des Editors	111
B2.1	Aktionen im Editiermodus	111
B3	Bedienungsanleitung PAL-Programmer	113
B3.1	Liste der Befehle im PAL-Betriebssystem	113
B3.2	Beschreibung zu den Befehlen	113
B3.2.1	EDIT - Befehl	114
B3.2.2	PROGRAM - Befehl	114
B3.2.3	MONITOR - Befehl	114
B3.2.4	FUSE - Befehl	114
B3.2.5	LOAD - Befehl	114
B3.2.6	SAVE - Befehl	115
B3.2.7	NEW - Befehl	115
B3.2.8	HELP - Befehl	116
ANHANG	D	117
	Datenblätter PAL	118
	Datenblätter D/A - Wandler DAC 1006	122
ANHANG	H	131
	Schaltpläne	132
	Lagepläne	139
	Stiftbelegung Expanderstecker P3	142
	Detailbeschreibung der CPU-karte	143
ANHANG	P	148
	Programmlisting Ein- Ausgabe- Paket	149
	Programmlisting Monitor	161
	Programmlisting PAL - Betriebssystem	173
	Programmlisting PALPGM	188
	Programmlisting MATRIX	200
ANHANG	Q	206
	Quellennachweis	207

1. Einleitung

1.1 Beschreibung eines PAL.

Das Kürzel PAL steht für "Programable Array Logic". Der deutsche Begriff den man für array-logic, bzw. logic-array gefunden hat, ist "Gatterfeld". Grundsätzlich ist ein PAL ein IC mit i.a. 20 oder 24 Anschlüssen in Slimlinebauweise. Die PAL's sind schon einige Jahre auf dem Markt erhältlich und sind von ihrer Architektur her vergleichbar mit den wesentlich häufiger vorkommenden PROM. Die Unterschiede verdeutlicht BILD 1.

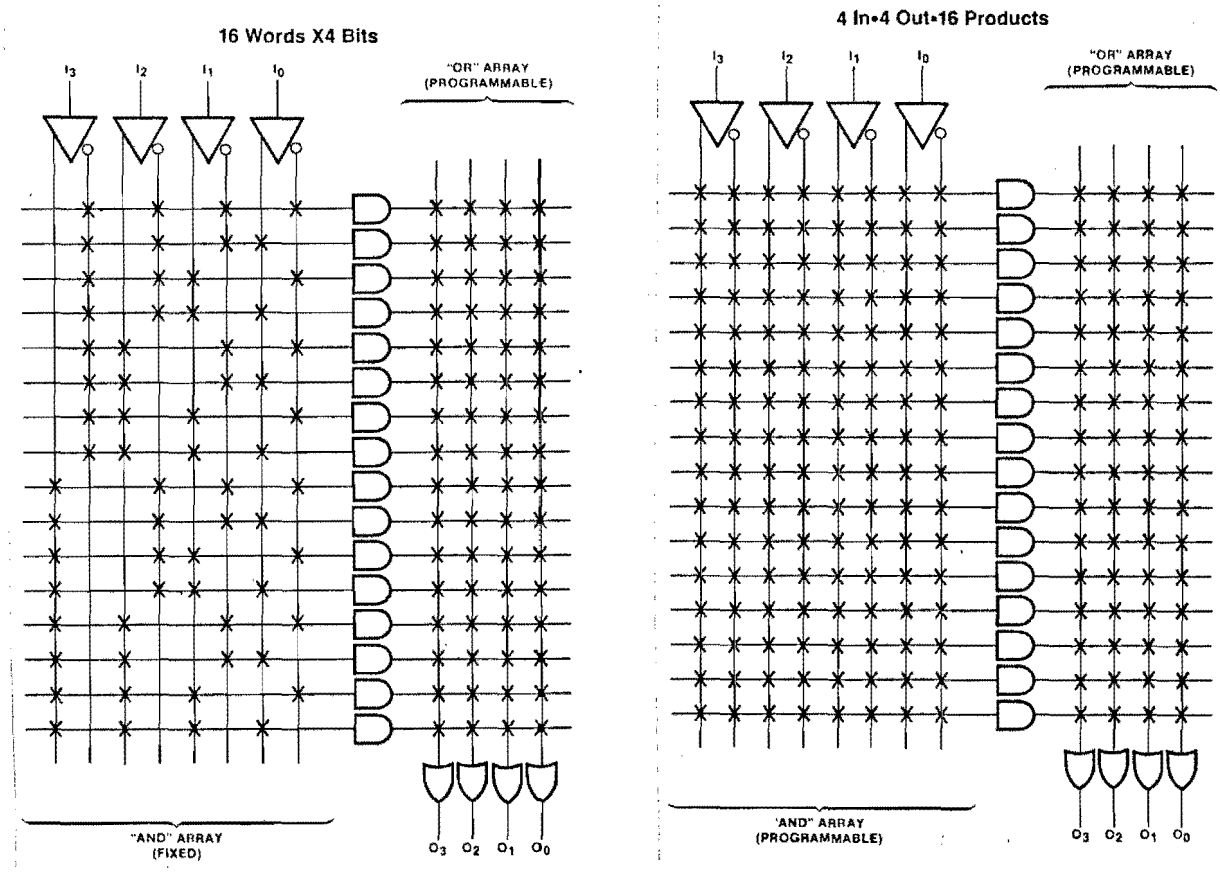


BILD 1a : Architektur eines PROM 1b : und eines FPLA

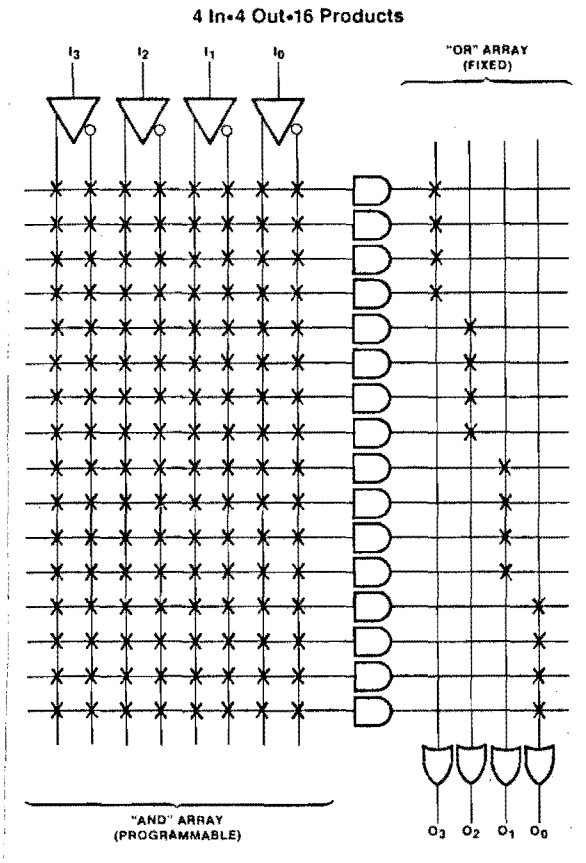


BILD 1c : PAL-Architektur

Bei einem PROM erhält man, je nach Speicherorganisation, an den Ausgängen ein wohldefiniertes Bitmuster in Abhängigkeit des an den Eingängen anliegenden Adresswortes. Die PAL-Architektur dagegen ist bitorientiert, d.h. man definiert die Eingangsvariablen immer nur für einen Ausgang. Die Bezeichnung eines PAL weist daher genau darauf hin, wieviel Ein- und Ausgänge der betreffende Baustein besitzt. (BILD 2a). Der weitestgehende Unterschied des PAL zu einem PROM besteht darin, daß die ODER-Ausgänge des programmierbaren UND-Feldes intern oft noch mit D-Registern beschaltet sind, die dann noch Lastentreiber steuern. (Bild 2b)

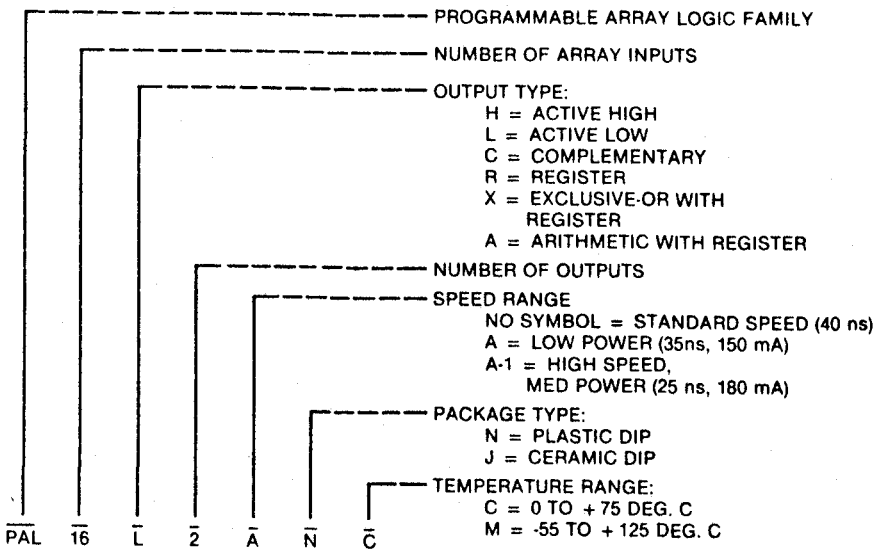


BILD 2a : aufgedruckte Typenbezeichnung für PAL 16L2

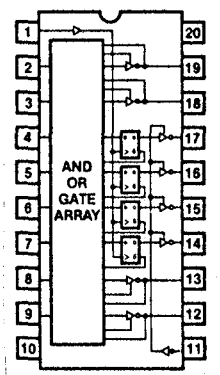


BILD 2b : Logiksymbol für PAL 16R4

Der Verwendungszweck ist nun schnell erklärt: mit den bekannten PAL-Bausteinen lassen sich alle bool'schen Grundfunktionen realisieren.(BILD 3)

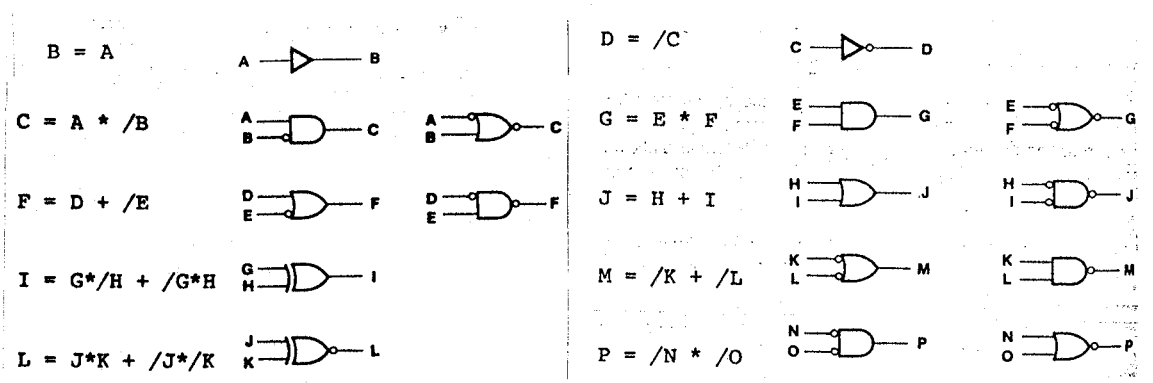


BILD 3 : Diese bool'schen Funktionen lassen sich mit allen PAL's realisieren.

Die Bilder 4a-4c zeigen Anwendungsbeispiele.

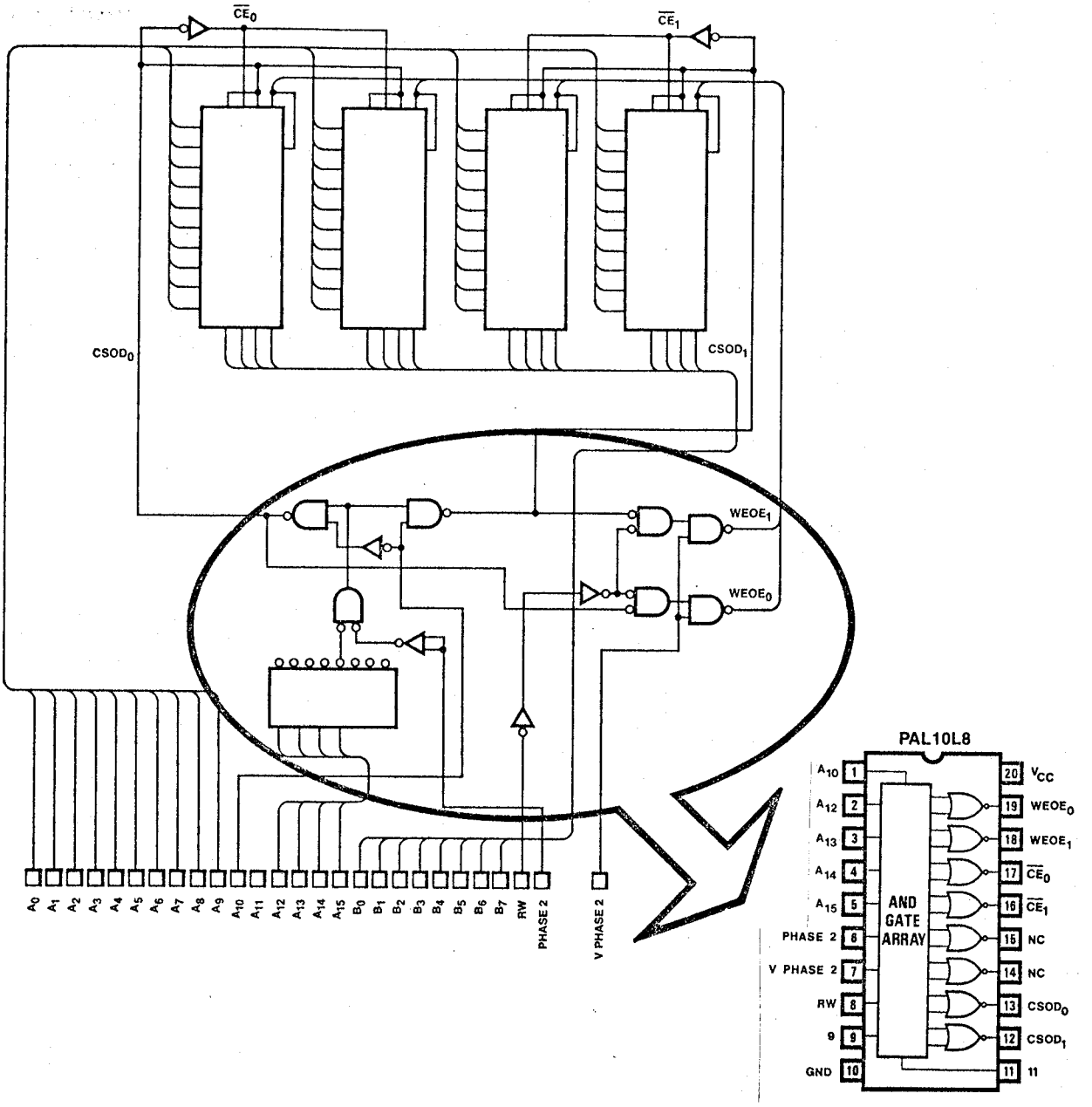


BILD 4a : Speicher interfacelogik für 6800 Mikroprozessorbus

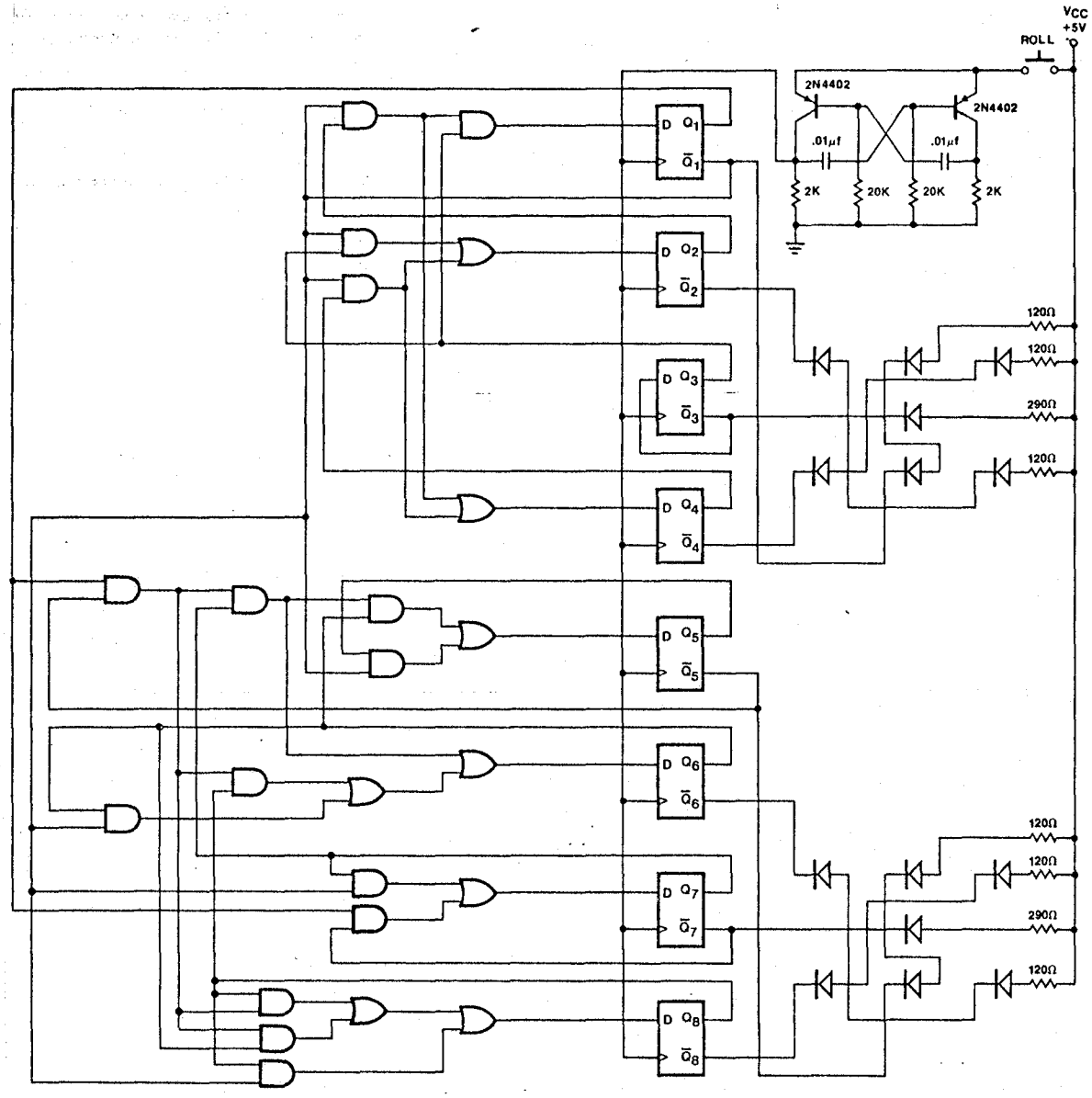


BILD 4b : Elektronisches Würfelspiel, realisiert mit PAL 16R8

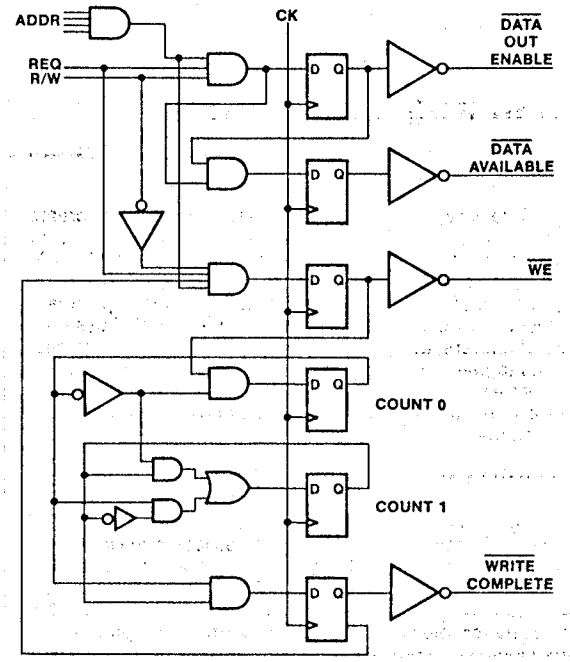
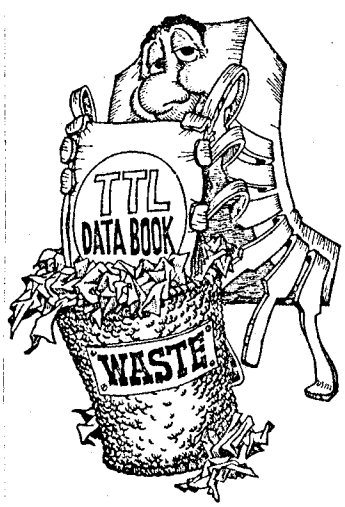
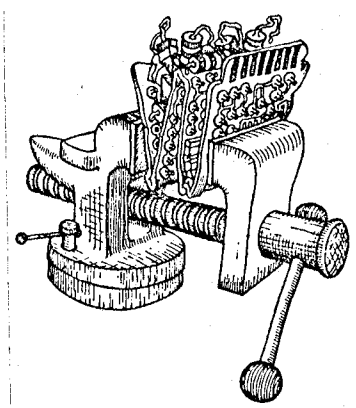


BILD 4c : Speicherquittungslogik

Die entscheidenden Vorzüge der PAL's gegenüber herkömmlicher TTL-logik sind :

- voll TTL-Kompatibel
- kann ca.90% der üblichen TTL Schaltungen ersetzen
- sehr große Platzersparnis auf der Leiterplatte und dadurch höhre Zuverlässigkeit der Schaltung
- typische Signallaufzeiten tp 35 ns
- weitgehend frei wählbare Pinbelegung (einfachere Leiterplattenentflechtung)
- die Schaltung ist fast absolut nachbausicher





Programmiert werden kann ein PAL nur mit PROM-programmiergeräten mit speziellem Zusatzaufbau, oder mit PAL- Programmierern, von denen es bis heute nur einige wenige Hersteller gibt. Deren Geräte kosten um DM 10 000,-

Manufacturer Model	Standard PALs		High-Speed PALs	
	Pers. Card	Adapter	Pers. Card	Adapter
National Semi Starplex	DM9068		TBD	TBD
Cybernetic Prog. Systems	CYMPC-1			
DATA I/O	909-1427	715 1428-1 715 1428-2 715 1428-3	TBD	TBD
Pro-Log	DM9068		TBD	TBD
Stag Systems	PM202	AM10H8 AM12H6 AM14H4 AM16H2 AM16C1 AM10L8 AM14L4 AM16L2		
Structured Design	SD-20/24		TBD	TBD

Einige Programmiergerätehersteller

1.2 Innerer Aufbau, Funktion

Das programmierbare UND-Feld ist eine Diodenmatrix. Die Verbindungen von Spalten- und Zeilenleitungen werden durch abbrennbare Sicherungen hergestellt. Jeder Ausgang eines PAL ist grundsätzlich über jeden Eingang und seiner Negation schaltbar. Dabei bilden die Eingänge die Spalten-leitungen (Inputlines) und die Ausgänge die Zeilenleitungen. Die Dioden der Eingangsleitungen bilden die programmierbare UND-Ver- knüpfung, deren Ergebnis (das Produkt) auf einer einzigen Zeilenleitung (der Produktline) dem Eingang einer fest verdrahteten ODER-Verknüpfung zugeführt wird.

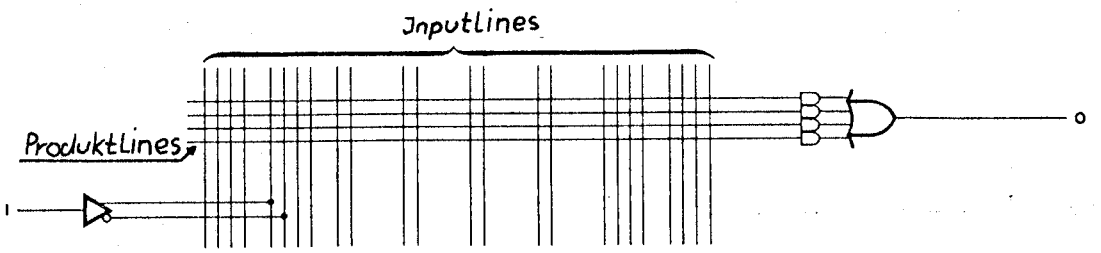


BILD 5a : Schema einer UND-ODER-Zelle

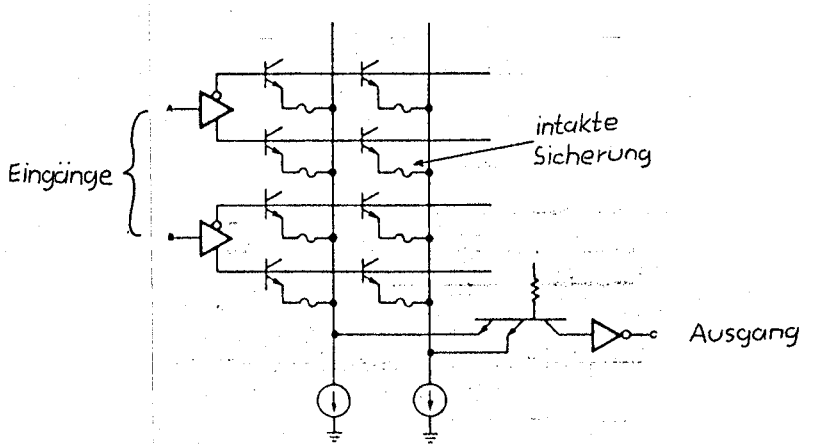


BILD 5b : Teilschaltung der Diodenmatrix, mit Eingängen, Ausgang und intakten Sicherungen

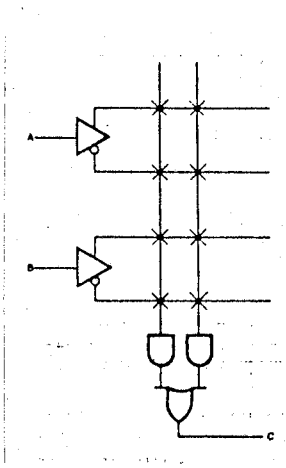


BILD 5c : Logikdiagramm zu Bild 5b

1.3 Programmierungsinformationen

Die PAL-Kennzeichnung beeinhaltet unter Anderem die Angabe des Ausgangspegels im Aktivzustand. Beispielsweise bezeichnet "10H8" ein aktiv-H PAL mit 10 Eingängen und 8 Ausgängen. Jeder PAL Ausgang wird dabei von einem ODER-Gatter gesteuert (oder auch von einem D- Register, das aber seinerseits an seinem D- Eingang von jenem ODER- Gatter gesteuert wird.). Im unprogrammierten Zustand sind die Eingänge aller ODER- Gatter LOG 0, wegen (BILD 6a)

$$I_1 \wedge \overline{I_1} = \emptyset$$

denn es bestehen ja noch alle Sicherungsverbindungen der UND- Matrix. Um nun einen Ausgang aktiv zu schalten, muß mindestens ein Eingang seiner ODER- Gatter LOG 1 gemacht werden. Dazu werden dann alle Sicherungen auf der Eingangsleitung des ODER- Gatters getrennt, bis auf eine, die die Verbindung zu dem Eingang beibehält, an dem der betreffende Ausgang aktiv geschaltet werden soll. Dabei kann gewählt werden, ob dieser Eingang mit LOG 0 oder LOG 1-Zustand den Ausgang aktivieren soll. (BILD 6b)

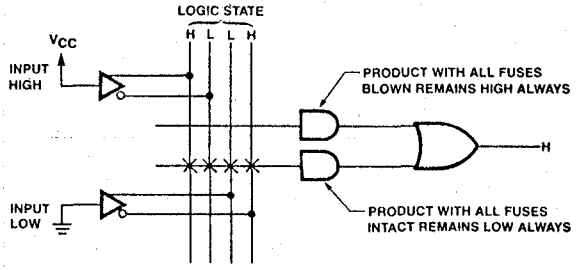


BILD 6a : Konstante Zustände an den ODER- eingängen

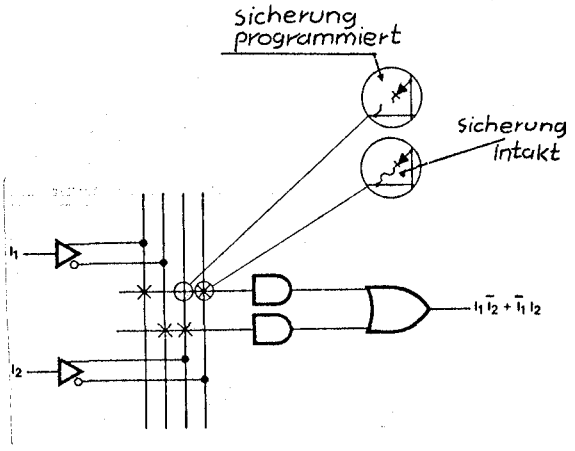


BILD 6b : variabele Zustände an den ODER- eingängen

Die Programmierungsinformation besteht nun darin, daß derjenige Kreuzungspunkt gekennzeichnet wird, der die

Verbindung des ODER- Gatters zum gewählten Eingang beibehält.

Zusammenfassung :

Die UND- Verknüpfungen der Eingangsvariablen werden realisiert durch nicht weggebrannte Dioden, die eine Wired-AND- Verbindung auf ODER- Eingänge herstellen. Die angekreuzten quasi-UND- Gatter weisen darauf hin, daß für diese Produktleitung keine Verbindung gebrannt wird. (BILD 7a)

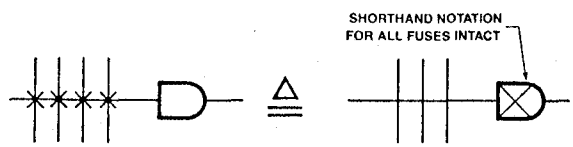


BILD 7a : Kurzkennzeichnung für gesperrte PRL.

ACTIVE HIGH OUTPUT	ACTIVE LOW OUTPUT
$O = \overline{I1}$	$\overline{O} = I1$
$O = \overline{I1} \cdot \overline{I2}$	$\overline{O} = I1 + I2$
$O = I1 + \overline{I2}$	$\overline{O} = \overline{I1} \cdot I2$
$O = I1 + \overline{I2} \cdot I3$	$\overline{O} = \overline{I1} \cdot I2 + \overline{I1} \cdot \overline{I3}$
$O = I1 \ddot{+} I2$	$\overline{O} = I1 \ddot{\cdot} I2$

BILD 7b : Grundfunktionen in disjunktiver- und konjunktiver Normalform

2. Analyse der Aufgabenstellung

Autonom bildschirmorientiert heißt, daß das PAL-Programmiergerät unabhängig von einer elektronischen Rechenanlage PAL- Bausteine programmieren kann. Zur Eingabe der notwendigen Parameter in die Programmiereinrichtung dient dann ein Bildschirmterminal mit Schreibmaschinentastatur. Die grundlegenden Programmierinformationen bestehen in der Angabe, welcher Kreuzungspunkt durchgebrannt werden soll und in der Angabe, um welchen PAL- Typ es sich handelt. In dieser Kennnzeichnung steckt die Infomation über die Größe der Matrix und über den aktiven Ausgangspegel.

Für diese komplexe Aufgabenstellung versteht sich die Verwendung einer Mikroprozessorsteuerung von selbst.

Entwicklungsziel ist es, am Bildschirmterminal den PAL- Typ einzugeben, Angaben zu machen über die nicht zu trennenden Verbindungen, die ja durch Kreuzchen in der Liste gekennzeichnet sind, und die verbleibenden zu brennen.

Die Lösung dieser Aufgabe macht die Trennung in zwei Teile notwendig. Der eine Teil stellt die Entwicklung eines Softwarepakets zur Terminalkommunikation dar, der andere Teil erfordert die Konstruktion einer Hardware zum umsetzen der berechneten Programmierdaten in die notwendigen Spannungsimpulse.

2.1 Pflichtenheft

Im Einzenen müßen folgende Teilaufgaben gelöst werden:
Bereich Hardware:

a) Erarbeiten eines Konzepts zur Selektions- und Programmierspannungserzeugung und aufschalten der logischen Zustände.
b) Gesamtkonzeption und Realisierung des Programmiergerätes.

Bereich Software:

a) Entwicklung und Implementierung eines Ein- Ausgabe-Pakets mit Zusatzfunktionen
b) Entwicklung und Implementierung eines Monitors
c) Entwicklung und Implementierung des PAL Betriebssystems.

3 Realisierung der Hardware

3.1 Realisierung der logischen Zustände

3.1.1 Konzeptfindung

3.1.1.1 Anwahl einer Diodensicherung

Zu jedem von max. 8 PAL- Ausgängen (bei 20pol. PAL) gehört eine Produktlinegruppe, die ihrerseits aus max. 8 Produktlines gebildet wird. (BILD 8) Die Selektion einer aus 40 (für 24pol. PAL) bzw. 32 (für 20pol. PAL) Inputlines übernimmt das Auftreten eines von drei logischen Zuständen an den Pins mit Namen I0 - I9 bzw I7.(BILD 9)

Die Beschaltung der Pins mit Namen A0-A2 mit zwei logischen Zuständen wählt eine aus acht Produktlines einer Produktlinegruppe. Zur Auswahl einer der 10 bzw 8 Produktlinegruppen bleiben nun nicht mehr genug Pins am Baustein übrig. Deswegen wird die Matrix in zwei Hauptgruppen geteilt:

- 1. HG für Produktline 0 - 39 (0 - 31)
- 2. HG für Produktline 40- 79 (32- 63)

Wird nun die erste Hauptgruppe programmiert, werden nur aus fünf bzw vier Produktlinegruppen gewählt. Wobei das Erscheinen des Programmierimpulses an einem der Pins mit Namen O0 - O4 (O0 - O3) diese Auswahl trifft.(BILD 10)

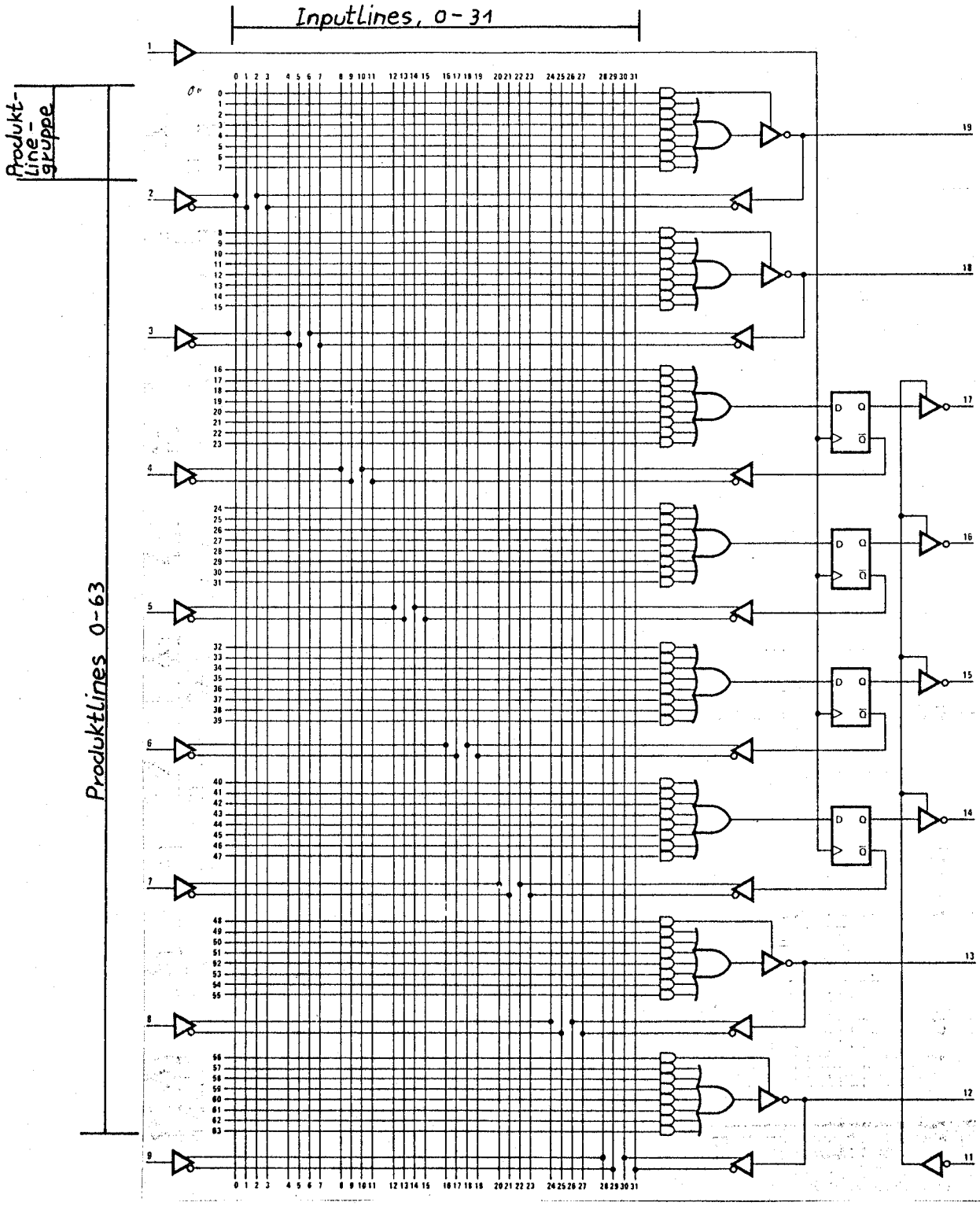


BILD 8 : Aufteilung einer Diodenmatrix in Inputlines, Produktgruppen und Produktlines. (PAL - 16R4)

INPUT LINE NUMBER	PIN IDENTIFICATION									
	I_9	I_8	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0
0	HH	HH	HH	HH	HH	HH	HH	HH	HH	L
1	HH	HH	HH	HH	HH	HH	HH	HH	HH	H
2	HH	HH	HH	HH	HH	HH	HH	HH	HH	L
3	HH	HH	HH	HH	HH	HH	HH	HH	HH	H
4	HH	HH	HH	HH	HH	HH	HH	HH	L	HH
5	HH	HH	HH	HH	HH	HH	HH	HH	H	HH
6	HH	HH	HH	HH	HH	HH	HH	HH	L	HH
7	HH	HH	HH	HH	HH	HH	HH	HH	H	HH
8	HH	HH	HH	HH	HH	HH	HH	L	HH	HH
9	HH	HH	HH	HH	HH	HH	HH	H	HH	HH
10	HH	HH	HH	HH	HH	HH	HH	L	HH	HH
11	HH	HH	HH	HH	HH	HH	HH	H	HH	HH
12	HH	HH	HH	HH	HH	HH	L	HH	HH	HH
13	HH	HH	HH	HH	HH	HH	H	HH	HH	HH
14	HH	HH	HH	HH	HH	HH	L	HH	HH	HH
15	HH	HH	HH	HH	HH	HH	H	HH	HH	HH
16	HH	HH	HH	HH	HH	L	HH	HH	HH	HH
17	HH	HH	HH	HH	HH	H	HH	HH	HH	HH
18	HH	HH	HH	HH	HH	L	HH	HH	HH	HH
19	HH	HH	HH	HH	HH	H	HH	HH	HH	HH
20	HH	HH	HH	HH	L	HH	HH	HH	HH	HH
21	HH	HH	HH	HH	H	HH	HH	HH	HH	HH
22	HH	HH	HH	HH	L	HH	HH	HH	HH	HH
23	HH	HH	HH	HH	H	HH	HH	HH	HH	HH
24	HH	HH	HH	L	HH	HH	HH	HH	HH	HH
25	HH	HH	HH	H	HH	HH	HH	HH	HH	HH
26	HH	HH	HH	L	HH	HH	HH	HH	HH	HH
27	HH	HH	HH	H	HH	HH	HH	HH	HH	HH
28	HH	HH	L	HH	HH	HH	HH	HH	HH	HH
29	HH	HH	H	HH	HH	HH	HH	HH	HH	HH
30	HH	HH	L	HH	HH	HH	HH	HH	HH	HH
31	HH	HH	H	HH	HH	HH	HH	HH	HH	HH
32	HH	L	HH	HH	HH	HH	HH	HH	HH	HH
33	HH	H	HH	HH	HH	HH	HH	HH	HH	HH
34	HH	L	HH	HH	HH	HH	HH	HH	HH	HH
35	HH	H	HH	HH	HH	HH	HH	HH	HH	HH
36	L	HH	HH	HH	HH	HH	HH	HH	HH	HH
37	H	HH	HH	HH	HH	HH	HH	HH	HH	HH
38	L	HH	HH	HH	HH	HH	HH	HH	HH	HH
39	H	HH	HH	HH	HH	HH	HH	HH	HH	HH

BILD 9a : Drei logische Zustände zur IPL-Anwahl

PRODUCT LINE NUMBER	PIN IDENTIFICATION							
	O_4	O_3	O_2	O_1	O_0	A_2	A_1	A_0
0, 40	Z	Z	Z	Z	HH	Z	Z	Z
1, 41	Z	Z	Z	Z	HH	Z	Z	HH
2, 42	Z	Z	Z	Z	HH	Z	HH	Z
3, 43	Z	Z	Z	Z	HH	Z	HH	HH
4, 44	Z	Z	Z	Z	HH	HH	Z	Z
5, 45	Z	Z	Z	Z	HH	HH	Z	HH
6, 46	Z	Z	Z	Z	HH	HH	HH	Z
7, 47	Z	Z	Z	Z	HH	HH	HH	HH
8, 48	Z	Z	Z	HH	Z	Z	Z	Z
9, 49	Z	Z	Z	HH	Z	Z	Z	HH
10, 50	Z	Z	Z	HH	Z	Z	HH	Z
11, 51	Z	Z	Z	HH	Z	Z	HH	HH
12, 52	Z	Z	Z	HH	Z	HH	Z	Z
13, 53	Z	Z	Z	HH	Z	HH	Z	HH
14, 54	Z	Z	Z	HH	Z	HH	HH	Z
15, 55	Z	Z	Z	HH	Z	HH	HH	HH
16, 56	Z	Z	HH	Z	Z	Z	Z	Z
17, 57	Z	Z	HH	Z	Z	Z	Z	HH
18, 58	Z	Z	HH	Z	Z	Z	HH	Z
19, 59	Z	Z	HH	Z	Z	Z	HH	HH
20, 60	Z	Z	HH	Z	Z	HH	Z	Z
21, 61	Z	Z	HH	Z	Z	HH	Z	HH
22, 62	Z	Z	HH	Z	Z	HH	HH	Z
23, 63	Z	Z	HH	Z	Z	HH	HH	HH
24, 64	Z	HH	Z	Z	Z	Z	Z	Z
25, 65	Z	HH	Z	Z	Z	Z	Z	HH
26, 66	Z	HH	Z	Z	Z	Z	HH	Z
27, 67	Z	HH	Z	Z	Z	Z	HH	HH
28, 68	Z	HH	Z	Z	Z	HH	Z	Z
29, 69	Z	HH	Z	Z	Z	HH	Z	HH
30, 70	Z	HH	Z	Z	Z	HH	HH	Z
31, 71	Z	HH	Z	Z	Z	HH	HH	HH
32, 72	HH	Z	Z	Z	Z	Z	Z	Z
33, 73	HH	Z	Z	Z	Z	Z	Z	HH
34, 74	HH	Z	Z	Z	Z	Z	HH	Z
35, 75	HH	Z	Z	Z	Z	Z	HH	HH
36, 76	HH	Z	Z	Z	Z	HH	Z	Z
37, 77	HH	Z	Z	Z	Z	HH	Z	HH
38, 78	HH	Z	Z	Z	Z	HH	HH	Z
39, 79	HH	Z	Z	Z	Z	HH	HH	HH

BILD 9b : Zwei logische Zustände zur PRL-Anwahl

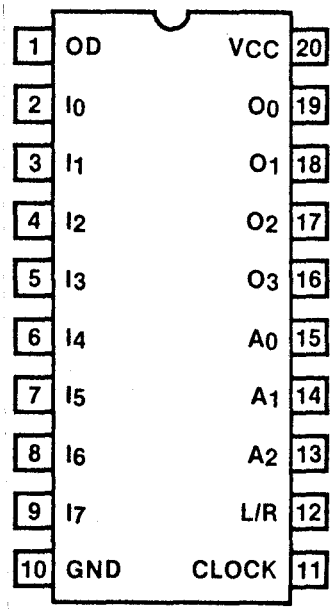


BILD 10a : Pinbelegung für erste HG

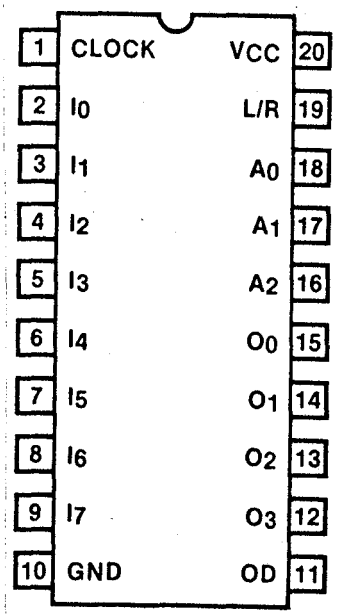


BILD 10b : Pinbelegung für zweite HG

Zusammenfassung:

Anhand der Tabellen (BILD 9) kann man sehen, daß die Anwahl der zu schmelzenden Sicherungen in der Matrix durch unterschiedliche Spannungspegel vorgenommen wird. Es reicht also nicht aus, wie zB. bei einem EPROM, daß man mit 5V-0V - Pegeln selektiert, und nur an einem einzigen Pin den Programmierimpuls anlegt.

Die Selektionsschritte lauten:

a) Anfangszustand (deselect): I0-I9 = HH
0,A0-A2, L/R = Z

b) Anwahl IPL: zugehöriger Pin I0-I9 nach L bzw H

c) Hauptgruppenentscheidung nach PRL-Nr.

d) demnach L/R -> HH an Pin 14 oder 23 (BILD 10)

e) A0-A2 -> HH (Pinbelegung BILD 10)

f) Programmierpuls -> HH (Pinbelegung BILD 10)

3.1.1.2 Lösungsansatz zur Aufschaltung der logischen Zustände

Ausgangspunkte zum Lösungsansatz bilden das Taktdiagramm (BILD 11) , die Wahrheitstabellen zur Selektion (BILD 12), und die Datenblattangaben zur Programmierung (BILD 13).

SYMBOL	PARAMETER		LIMITS MIN	TYP	MAX	UNIT
V_{IHH}	Program-level input voltage		11	11.5	12	V
I_{IHH}	Program-level input current	Output Program Pulse			50	mA
		OD. L/R			25	
		All Other Inputs			5	
I_{CCH}	Program Supply Current				400	mA
T_P	Program Pulse Width		10		50	µs
t_D	Delay time		100			ns
t_{DV}	Delay Time to Verify		100			µs
	Program Pulse duty cycle				25	%
V_P	Verify-Protect-input voltage		20	21	22	V
I_P	Verify-Protect-input current				400	mA
T_{PP}	Verify-Protect Pulse Width		20		50	msec

BILD 13 : Parameter für den Programmierzyklus

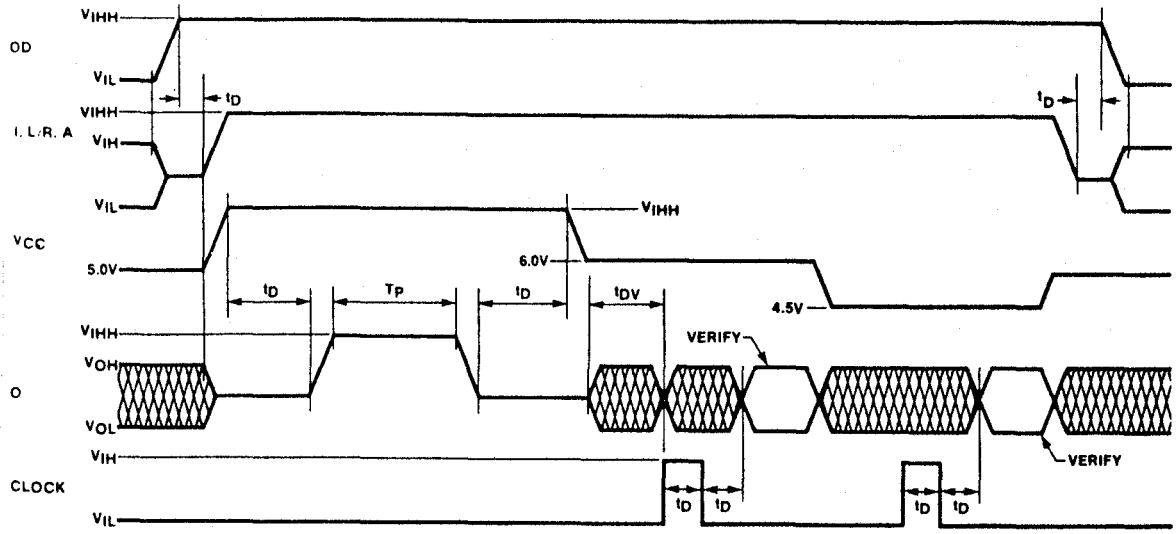


BILD 11 : Programmierungstaktdiagramm

Input Line Number	Pin Identification								
	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	L/R
0	HH	HH	HH	HH	HH	HH	HH	L	R
1	HH	HH	HH	HH	HH	HH	HH	H	R
2	HH	HH	HH	HH	HH	HH	HH	L	HH
3	HH	HH	HH	HH	HH	HH	HH	H	HH
4	HH	HH	HH	HH	HH	HH	L	HH	R
5	HH	HH	HH	HH	HH	HH	H	HH	R
6	HH	HH	HH	HH	HH	HH	L	HH	HH
7	HH	HH	HH	HH	HH	HH	H	HH	HH
8	HH	HH	HH	HH	HH	L	HH	HH	R
9	HH	HH	HH	HH	HH	H	HH	HH	R
10	HH	HH	HH	HH	HH	L	HH	HH	HH
11	HH	HH	HH	HH	HH	H	HH	HH	HH
12	HH	HH	HH	HH	L	HH	HH	HH	R
13	HH	HH	HH	HH	H	HH	HH	HH	R
14	HH	HH	HH	HH	L	HH	HH	HH	HH
15	HH	HH	HH	HH	H	HH	HH	HH	HH
16	HH	HH	HH	L	HH	HH	HH	HH	R
17	HH	HH	HH	H	HH	HH	HH	HH	R
18	HH	HH	HH	L	HH	HH	HH	HH	HH
19	HH	HH	HH	H	HH	HH	HH	HH	HH
20	HH	HH	L	HH	HH	HH	HH	HH	R
21	HH	HH	H	HH	HH	HH	HH	HH	R
22	HH	HH	L	HH	HH	HH	HH	HH	HH
23	HH	HH	H	HH	HH	HH	HH	HH	HH
24	HH	L	HH	HH	HH	HH	HH	HH	R
25	HH	H	HH	HH	HH	HH	HH	HH	R
26	HH	L	HH	HH	HH	HH	HH	HH	HH
27	HH	H	HH	HH	HH	HH	HH	HH	HH
28	L	HH	HH	HH	HH	HH	HH	HH	R
29	H	HH	HH	HH	HH	HH	HH	HH	R
30	L	HH	HH	HH	HH	HH	HH	HH	HH
31	H	HH	HH	HH	HH	HH	HH	HH	HH

Product Line Number	Pin Identification						
	O_3	O_2	O_1	O_0	A_2	A_1	A_0
0.32	R	R	R	V_{PH}	R	R	R
1,33	R	R	R	V_{PH}	R	R	HH
2,34	R	R	R	V_{PH}	R	HH	R
3,35	R	R	R	V_{PH}	R	HH	HH
4,36	R	R	R	V_{PH}	HH	R	R
5,37	R	R	R	V_{PH}	HH	R	HH
6,38	R	R	R	V_{PH}	HH	HH	R
7,39	R	R	R	V_{PH}	HH	HH	HH
8,40	R	R	V_{PH}	R	R	R	R
9,41	R	R	V_{PH}	R	R	R	HH
10,42	R	R	V_{PH}	R	R	HH	R
11,43	R	R	V_{PH}	R	R	HH	HH
12,44	R	R	V_{PH}	R	HH	R	R
13,45	R	R	V_{PH}	R	HH	R	HH
14,46	R	R	V_{PH}	R	HH	HH	R
15,47	R	R	V_{PH}	R	HH	HH	HH
16,48	R	V_{PH}	R	R	R	R	R
17,49	R	V_{PH}	R	R	R	R	HH
18,50	R	V_{PH}	R	R	R	HH	R
19,51	R	V_{PH}	R	R	R	HH	HH
20,52	R	V_{PH}	R	R	HH	R	R
21,53	R	V_{PH}	R	R	HH	R	HH
22,54	R	V_{PH}	R	R	HH	HH	R
23,55	R	V_{PH}	R	R	HH	HH	HH
24,56	V_{PH}	R	R	R	R	R	R
25,57	V_{PH}	R	R	R	R	R	HH
26,58	V_{PH}	R	R	R	R	HH	R
27,59	V_{PH}	R	R	R	R	HH	HH
28,60	V_{PH}	R	R	R	HH	R	R
29,61	V_{PH}	R	R	R	HH	R	HH
30,62	V_{PH}	R	R	R	HH	HH	R
31,63	V_{PH}	R	R	R	HH	HH	HH

BILD 12 : Wahrheitstabelle zur Selektion einer Dioden-sicherung

Zusammenfassung:

Bereitgestellt werden müssen folgende Spannungen:

Für Pins OD,I,CLOCK (siehe BILD 10)

a) L-zustand, U_{ILmax} = 0,8V

b) H-zustand, U_{IHmin} = 2,0V

c) HH-zustand, U_{IHHtyp} =11,5V

Für die Pins O,A (siehe BILD 10)

a) Z-zustand, definiert mit 10kΩ nach +5V

b) HH-zustand, U_{IHHtyp} =11,5V

Für die Versorgungsspannung

a) Ucc_1 = 5V Betriebsphase

b) Ucc_2 = 6V Verifizierungsphase

c) Ucc_3 = 4,5V Verifizierungsphase

d) Ucc_4 =11,5V Programmierungsphase, mit
Icc_H = 400 mA

Die Eingangsströme an den anderen Pins überschreiten nicht den Wert I_{IHH} = 50 mA.

Nach dem Studium der Bilder 10,11,12 und 13 ergibt sich folgende, für die konzeption der Zustandsschalter wichtige Erkenntnis: Die Spannung U_{HH} bleibt, mit Ausnahme der Pins 10 (gnd) und 20 (Vcc) für alle Pins konstant in einem Programmierzyklus. Das heißt, sie kann ihnen aus einer Spannungsquelle zugeführt werden. Getrennt geschaltet werden müssen Pins OD,I und CLOCK nach Masse und +5V.

Eine Schaltungsidee für einen solchen Zustandsschalter zeigt BILD 14

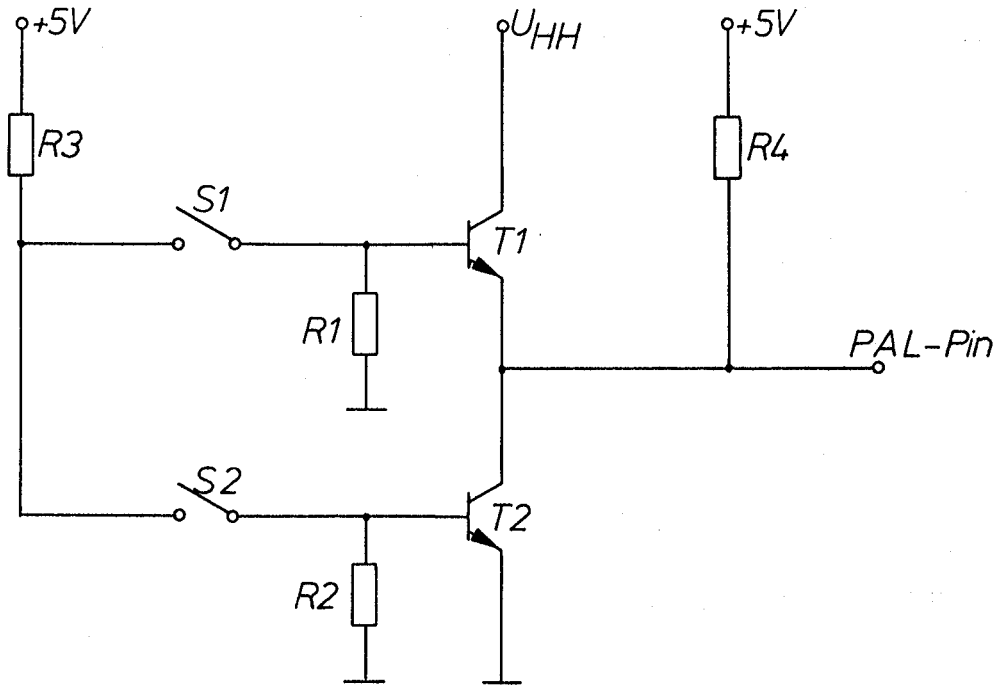


BILD 14 : Konzept für einen Zustandsschalter

Die Schaltung nach BILD 14 realisiert an ihrem Ausgang folgende Zustände in Abhängigkeit der Schalterstellungen von S1 und S2:

S1 offen, S2 offen : +5V

S1 offen, S2 gesch : 0V

S1 gesch, S2 offen : U_{HH}

S1 gesch, S2 gesch : *** unzulässig ***

Für die PAL-Pins O und A gilt die gleiche Konfiguration jedoch ohne T2,R2,S2 dafür mit R4 = 10k = Rz lt. Z-spezifikation. Getrennt von allen anderen Pins muß die Spannung Ucc eingestellt werden können, wobei gilt:

$$4{,}5V < Ucc < 12V$$

Nachteil dieses Schaltungskonzeps:

Es werden 34 diskrete Transistoren benötigt um ein 24pol. PAL zu programmieren. Ein anderes Schaltungskonzept zeigt BILD 15.

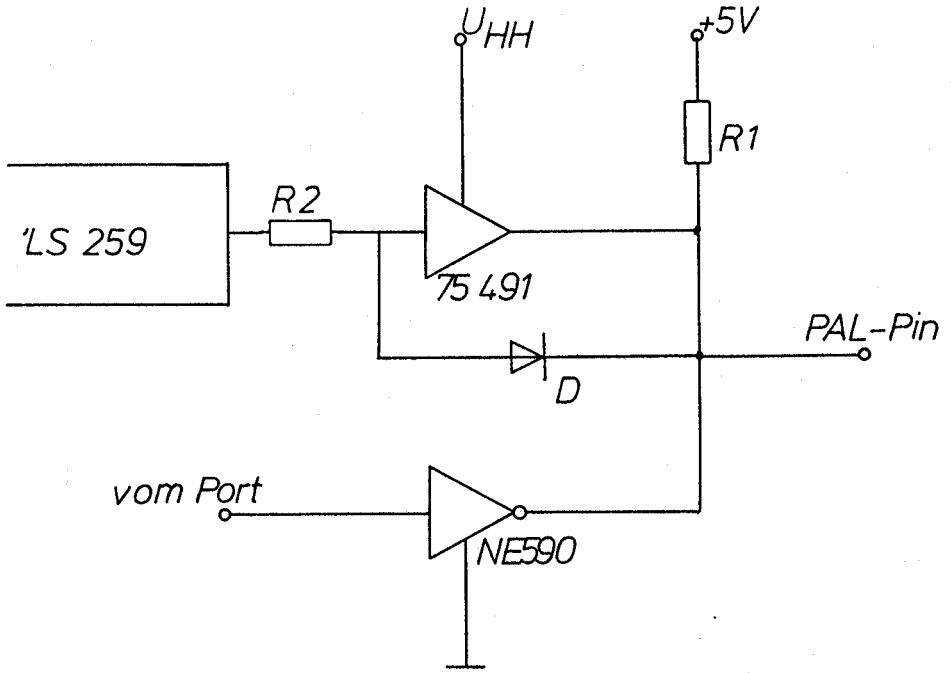


BILD 15 : Zustandsschalter, konzipiert mit Transistorarray

In die nähere Überlegung wird dieses Konzept einbezogen. Der NE 590 ist ein in invertierender Kathodentreiber, pinkompatibel zu 'LS 259, der das vierfach-Transistorarray 75 491 ansteuert. Die zulässigen Spannungen an den Eingängen von NE 590 U_{CEmax}= 15 V bei Ice = 250mA , und von 75 491 U_{CEmax}= 20 V bei Ice = 50mA . Diese Schaltung ermöglicht eine Absicherung des unerlaubten Zustandes mit einer Diode D .

Zusammenfassung:

Die Parameter sind den MMI-Spezifikationen entnommen. Alle Anderen sind weitgehend Second-Source Hersteller und lehnen sich an die MMI-Daten an. Das gilt auch für die Selektionstabelle. Ganz abweichend davon werden PAL's von AMD programmiert. Dort gibt es Spezifikationen zur maximalen Verzögerungszeit eines Zustandswechsels im Programmierungstaktdiagramm die im Nanosekundenbereich liegt. Diese Spezifikation kann Softwaremäßig kaum realisiert werden.

3.1.1.3 Konzept für die Spannungsquelle.

Benötigt werden also neben den üblichen +5V noch eine Festspannungsquelle für U_{IHH} und eine einstellbare Spannungsquelle für Ucc.

Die Berücksichtigung der Spezifikationen verschiedener PAL- Hersteller lassen es günstig erscheinen, insgesamt drei, voneinander getrennt einstellbare Spannungsquellen zu realisieren. Und zwar jeweils eine zur Versorgung der linken Seite PAL, der rechten Seite PAL und für Ucc.

Strombedarfsermittlung (aus MMI-Datenblatt)

Linke Seite PAL :

I_{IHH} = 5mA pro Pin I = 5mA x 10 Pins = 50mA

$+I_{IHH}$ = 50mA an Pin 1 = = 50mA

zusätzlich soll die Verify-Protect-Sicherung gebrannt werden :

Ip = =400mA

Σ 500mA

Rechte Seite PAL : (nur jeweils ein "0" - Pin)

I_{IHH} = 50mA pro Pin14-23 = 50mA x 6 Pins = 300mA

Ip = = 400mA

Σ 700mA

Versorgung an Pin 24 (20) :

$Icc_{H_{max}}$ = Σ = 400mA

3.1.1.4 Realisierung mit Schaltregler TL 497

Die Verwendung des TL 497 erschien günstig, da seine Beschaltung im "Step Up" Betrieb und mit externem Transistor (aus Texas Instruments Schaltvorschlag) die Erfüllung der Spezifikationen versprach. Außerdem kann die komplette Programmiereinrichtung auf diese Weise mit nur einer einzigen Versorgungsspannung von +5V betrieben werden.

Die Idee zur Steuerung des TL 497 war, in den Gegenkopplungszweig das R2R- Netzwerk eines D/A- Wandlers zu schalten. (BILD 16)

Die Beschaltung zeigte sehr gutes Lastregelverhalten, jedoch nur bis zu $I_{HH} = 300mA$, trotz externem Transistor. Der Grund für die geringe Strombelastbarkeit liegt in den Anforderungen an den Schalttransistor :

- hohe Transitfrequenz $f_t \approx 6MHz$
- hohe Stromverstärkung $B > 50$
- hoher Kollektorstrom $I_C \approx 3A$

Das leistet z.B. der Typ BU 426, oder spezielle Sendertransistoren, die schwer zu Haben sind. Nachteil dieses Konzepts ist die hohe Strombelastung der 5V-Quelle.

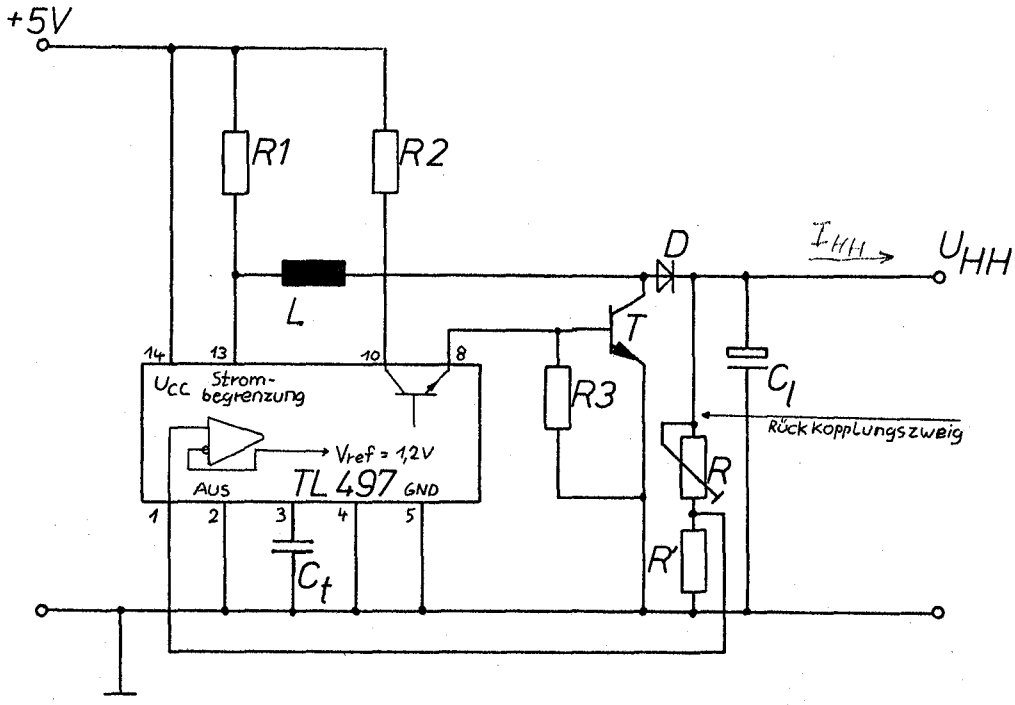


BILD 16 : Versuchsaufbau mit TL 497 ohne D/A-Wandler.

3.1.1.5 Realisierung mit Linearregler L 200

Die Idee der Steuerung des L 200 mit dem D/A- Wandler kann übernommen werden. Der L 200 garantiert einen Ausgangsstrom bis zu $I_{a\,max}$= 2A ohne externe, aktive Beschaltung bei $U_{a\,max}$= 37 V. Der Nachteil dieses Konzepts gegenüber dem o.g. ist, die Notwendigkeit einer zweiten Versorgungsspannung für das Programmiergerät, und die hohe Verlustleistung des Linearreglers.

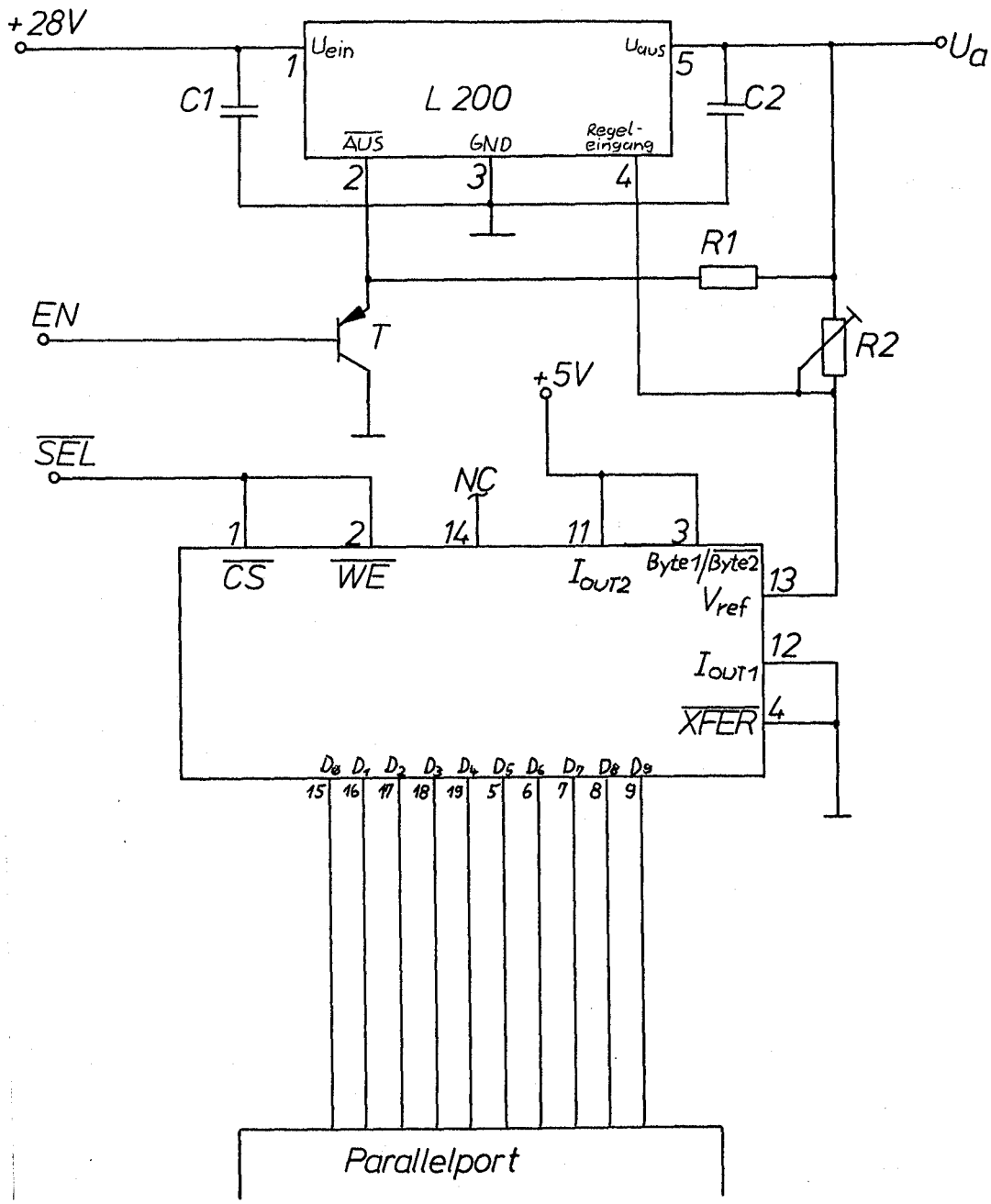


BILD 17 : Versuchsaufbau mit L 200 und D/A-Wandler DAC 1006

3.1.2 Berechnung

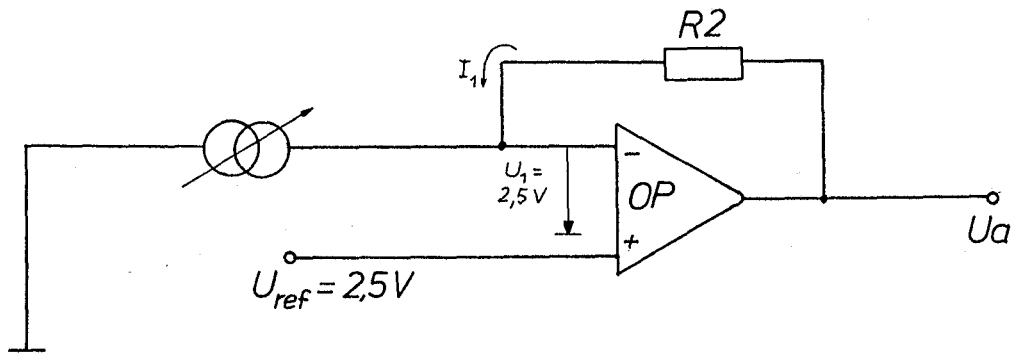


BILD 18 : L 200 als nichtinvertierender Operationsverstärker.

R1 = R2R-netzwerk

$$\frac{Ua}{Ue} = \left(1 + \frac{R2}{\left(\frac{U1}{I1}\right)}\right) \longrightarrow Ua = Ue\left(1 + \frac{R2}{\left(\frac{U1}{I1}\right)}\right)$$

mit : R2 = 200 k, U1 = 2,5V

folgt :

$$Ua = 2{,}5\left(1 + \frac{200\ k}{15\ k}\right) = \underline{35{,}8V} \quad \text{und} \quad I = \frac{U1}{R1} = \frac{2{,}5\ V}{15\ k} = 160\ uA$$

$$Ua = 2{,}5\left(1 + \frac{200\ k}{200\ k}\right) = \underline{4{,}5\ V} \quad \text{und} \quad I = \frac{U1}{R1} = \frac{2{,}5\ V}{200\ k} = 12{,}5\ uA$$

$$Ua = 2{,}5\left(1 + \frac{200\ k}{1\ M}\right) = \underline{3\ V} \quad \text{und} \quad I = \frac{U1}{R1} = \frac{2{,}5\ V}{1M} = 2{,}5\ uA$$

3.2 Gesamtkonzeption

Die Programmiereinrichtung muß folgende Elemente enthalten:

- Mikroprozessorsteuerung
- min. 34 Steuerleitungen zum Schalten der Zustände
- drei getrennt steuerbare Spannungsquellen
- 6 Stück Transistorarray 75 491
- 2 Stück 8 Bit-Rerister NE 590
- 3 Stück 8 Bit-Register 74 LS 259
- 1 Parallelportbaustein mit 3 x 8 Leitungen
- 10 Spannungskomparatoren für den Verifizierzyklus, gewählt wurde LM 339

Durch die 5 Stück 8 Bit-Register werden 40 Steuerleitungen bereitgestellt. Die Bausteine sind untereinander pinkompatibel. Die alternative Verwendung eines Serien-Parallel-Umsetzers z.B. MC 144100 von Motorola (ähnlicher Baustein ist auch von Valvo angekündigt) in Dreileiterbus-Technik scheidet wegen mangelnder Verfügbarkeit aus.

3.2.1 Konzeption der CPU- Karte

3.2.1.1 Zeitkritische Abschnitte der Selektions und Programmierphase

Die Programmierspezifikation schreibt die Einhaltung der Programmierimpulsdauer Tp vor, für die gilt :

$$10\ \mu s < Tp < 50\ \mu s$$

Verzögerungszeiten Td für Zustandswechsel sind angegeben mit :

$$T_{dmin} = 100\ ns$$

welche unkritisch sind.

3.2.1.2 Steuerung des Programmierpulses

Einen Teil der Fuseselektion übernimmt auch der Programmierpuls, bzw sein Erscheinen an einem der Pins 14 - 23 (siehe Kap.3.1.1.1). Wegen des Spannungspeges U_{IHH} kann der Puls nicht unmittelbar vom Integrierten Portbaustein gesteuert werden. Der Programmteil zur Steuerung des Programmierpulses und seine Zeitbetrach- tung lautet :

(Register sind entsprechend vorbereitet)

	LD A,B	
	OUT (#10),A	; sel.byte für HH
	LD A,D	;----------
	OUT (#02),A	; aktiver Strobe I
$4T_Z$	LD A,E	
$11T_Z$	OUT (#02),A	
T_P $4T_Z$	LD A,C	
$11T_Z$	OUT (#10),A	; deselect => L
$4T_Z$	LD A,D	
$11T_Z$	OUT (#02),A	; aktiver Strobe I
$\Sigma\ 45T_Z$		

d.h, 45 Taktzustände müssen in max. 50 usec ausgeführt sein :

$$t_{T_Z} = \frac{50 \cdot 10^{-6}\ sec}{45\ T_Z} = 1{,}1 \cdot 10^{-6}\ sec$$

für einen Taktzyklus. Damit ist die Mindestsystemtaktfrequenz :

$$f_{s\,min} = \frac{1}{1{,}1 \cdot 10^{-6}\ sec} = 9 \cdot 10^{5}\ Hz = 900\ KHz$$

und bei 2,5 MHz Systemtaktfrequenz :

$$\underline{\underline{T_D}} = \frac{45\,T_Z}{2{,}5\ MHz} = 1{,}8 \cdot 10^{-5} sec = \underline{\underline{18\,\mu sec}}$$

Zusammenfassung :

Zur Verfügung steht ein Einplatinenkomputer in folgender Konfiguration :

- Z 80 Mikroprozessor
- 2 kByte Schreib-Lesespeicher
- 4 kByte Festwertspeicher
- 6 x 8 Bit Parallelports (2 St. 8255)
- 2 serielle Kommunikationsbausteine 8251
- Systemtaktfrequenz f_S = 2,5 MHz

Zur Steuerung der gesamten Hardware genügt ein Parallelportbaustein 8255. Er bedient mit Port A 5 St. 8 Bit-Register. Port B und Bit A0, A1 steuern in der Ausgaberichtung die D/A - Wandler, und in der Eingaberichtung lesen sie die Zustände an den PAL- Pins 14 - 23 zur Verifizierung.

3.2.2 Beschreibung des Einplatinenkomputers

siehe Anhang

3.2.3 Änderungen und Erweiterungen am Einplatinenkomputer

Die Terminalkommunikation in RS 232- Norm erfordert die Beschaltung der TxD leitungen des 8251 mit 12 V Treibern und der RxD leitungen mit 12 V Empfängern. Zur Erzeugung der 12 V Senderspannung dient ein Integrierter Schaltregler ICL 7660 von Intersil, der für solche Zwecke konzipiert wurde. Die Steuerleitungen der RS 232 Norm werden für den Drucker und den Terminalbetrieb nicht verwendet. Diese Erweiterung wurde auf eine Lochrasterplatte (50 x 100 mm) aufgebaut, mit einer 64pol. VG Stiftleiste und den RS 232 - kabeln versehen. Ferner erhält diese Erweiterung noch den Anschlußstecker für die +5V Systemspannung.

Änderungen :

Sehr unschön ist die getrennte Power-On Resetsteuerung der CPU und der Ports. Um sichere Zustände der Ports zu garantieren hat man die Zeitkonstante ihres R - C Glieds viel größer gewählt, als die des für die CPU. Die anfängliche Zeitschleife um auf die Ports zu warten ist sehr lästig. Für die gemeinsame Resetsteuerung wurde eine Negierung auf der RS 232-karte benutzt.

Die Konstruktion der CPU-karte erlaubt nicht die Arbeitsweise des einen UART mit 9200 Baud und des zweiten UART mit 300 Baud (Druckerbetrieb). Folgende Änderung ist nötig :

- Auftrennen der Leitung zwischen 3 & 4 am Baudratengenerator
- Auftrennen der CLK-leitungen die beide UART verbinden
- Verbinden einer freien CLK-leitung mit Stecker 5 am Baudratengenerator

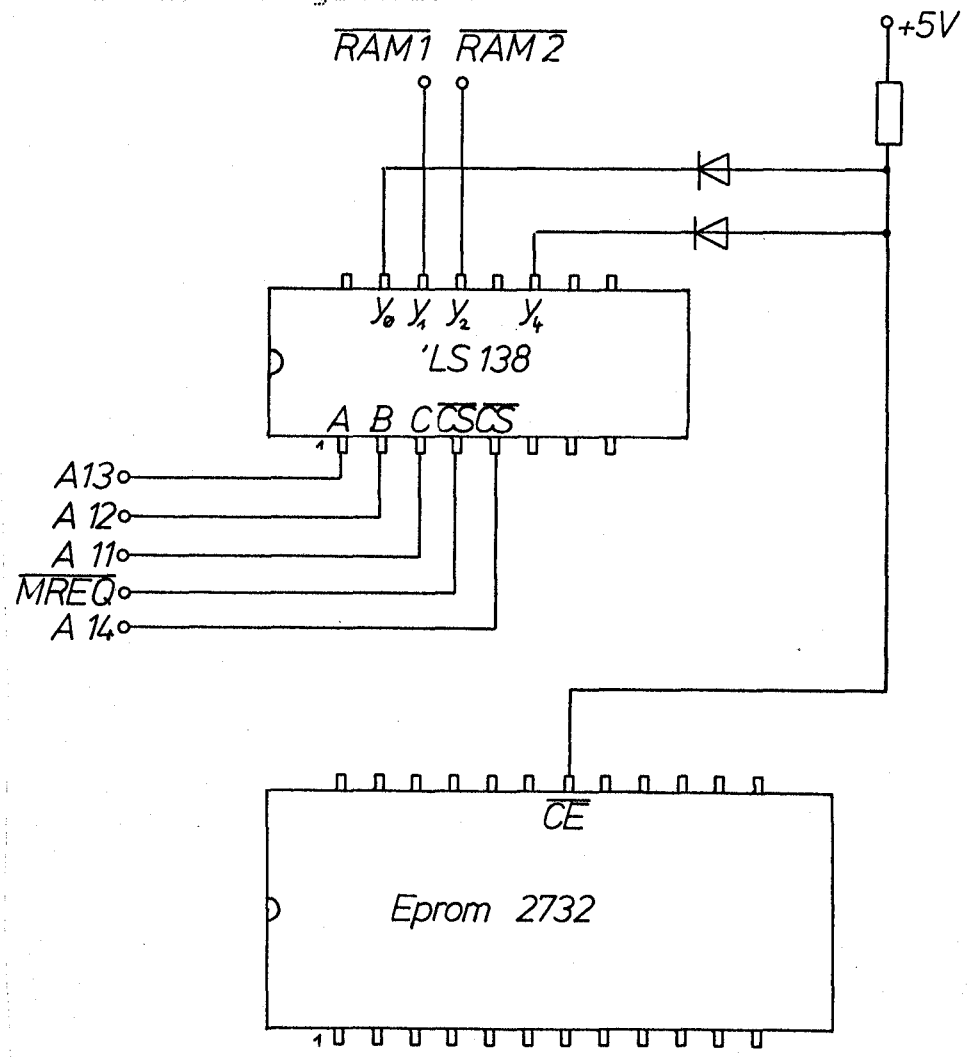


BILD 19 : So wird der ROM Speicherbereich vergrößert

3.3 Aufbau der Programmiereinrichtung

3.3.1 Beschreibung der Interfacekarte

Zur Herstellung des Prototyps wurden die notwendigen Bausteine auf Lochrasterplatte handverdrahtet. Insgesamt gibt es drei Europakarten, ausschließlich der CPU - karte. Die Interfacekarte beinhaltet folgende Funktionen (siehe Zeichnung Nr 8, Anhang) :

- Drei steuerbare Spannungsquellen mit den zugehörigen D/A- wandlern (Zeichnung Nr. 3, Anhang)
- Fünf Porterweiterungsbausteine mit Ansteuerlogik (Zeichnung Nr. 2, Anhang)
- Einen 50pol. Pfostensteckverbinder P3
- Einen 14pol. Pfostensteckverbinder P2 (siehe Zeichnung Nr. 2, Anhang)

Zusätzlich wurde mit auf die Karte ein Integrierter Portbaustein 8255 (INTEL) mit Decodierung vorgesehen, um in der Anfangsphase der Realisierung die korrekte Funktion der Spannungseinstellung am Systembus des Entwicklungsrechners zu testen. Dazu wurde die Interfacekarte noch mit einer 64pol. VG Stiftleiste versehen. (P4) Im entgültigen Betrieb wird die Interfacekarte mit der CPU- karte über Portstecker P1 verbunden (siehe Zeichnung 2, Anhang).

3.3.1.1 Decodierung des Interfaceport

A7	A6	A5	A4	A3	A2	A1	A0	
0	0!	0	0	0	0	!x	x	Registeradressen
0	0!	0	0	0	1	!x	x	
0	0!	0	0	1	0	!x	x	
0	0!	0	0	1	1	!x	x	
0	0!	0	1	0	0	!		
0	0!	0	1	0	1	!		
0	0!	0	1	1	0	!		
0	0!	0	1	1	1	!		
0	0!	1	0	0	0	!		
0	0!	1	0	0	1	!		
0	0!	1	0	1	0	!		
0	0!	1	0	1	1	!		
0	0!	1	1	0	0	!		
0	0!	1	1	0	1	!		
0	0!	1	1	1	0	!		
0	0!	1	1	1	1	!		

Adressierungsbereich der Interfaceportregister :

00 - 03	20 - 23
04 - 07	24 - 27
08 - 0B	28 - 2B
0C - 0F	2C - 2F
10 - 13	30 - 33
14 - 17	34 - 37
18 - 1B	38 - 3B
1C - 1F	3C - 3F

TABELLE 1 : Diese Adressen sind durch Jumper "J" wählbar.

3.3.1.2 Beschreibung der Ansteuerung des Spannungsreglers L 200

Zum Setzen der Spannungspegel muß folgendes Wortformat an die Ports A und B ausgegeben werden :

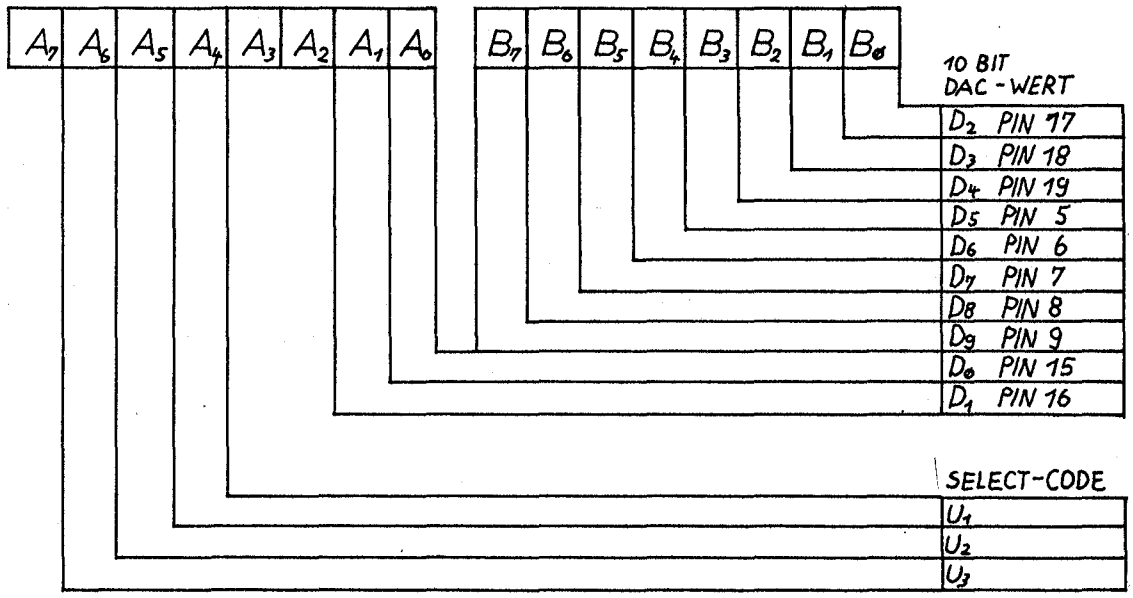


Bild 20 : Steuerwortformat zum Programmieren der Spannungsquellen

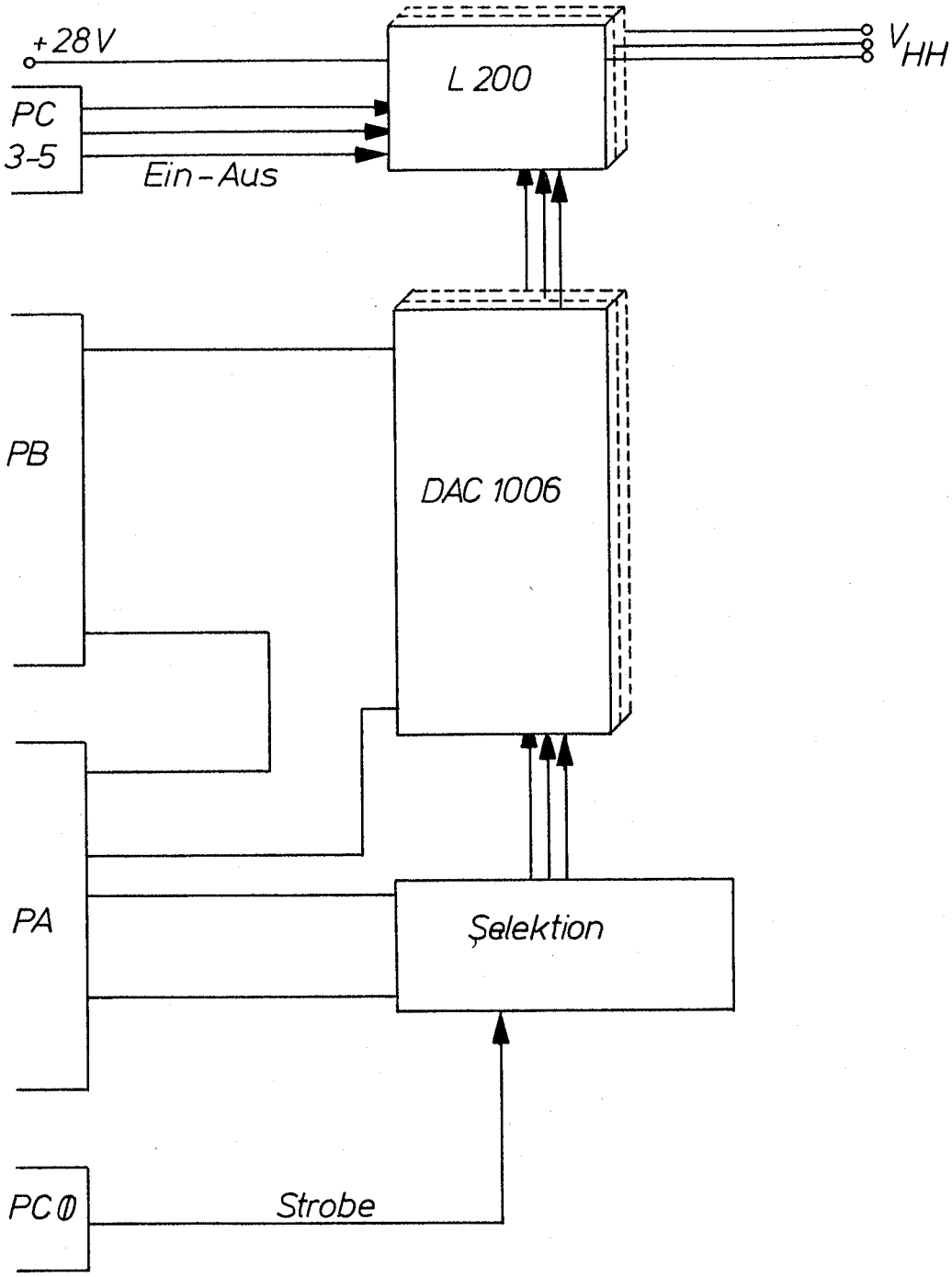


BILD 21 : Blockschaltbild der Programmierspannungssteuerung.

Zum Einschalten der Spannungsquellen werden im Steuerport C die Bits 4, 5 oder 6 = 0 gesetzt. Die Zuordnung der Spannungsquellen zu den Schalterbits geht aus BILD 22 hervor.

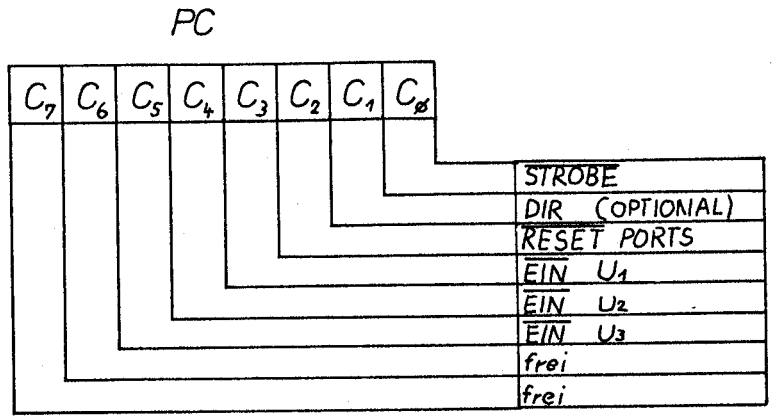


BILD 22 : Funktion des Steuerport C

Der Programmteil zur Spannungsschaltung lautet :

```
                1930 ;
                1940 ; Unterprogramm VOLT : RA, RB, RC, RHL
0E15 E3         1950 VOLT   EX   (SP),HL      ;-----------------------------
0E16 7E         1960        LD   A,(HL)       ;      hier werden
0E17 D310       1970        OUT  (#10),A      ;      alle drei
0E19 23         1980        INC  HL           ;      Parameter
0E1A 7E         1990        LD   A,(HL)       ;      zur steuerung
0E1B D311       2000        OUT  (#11),A      ;      der Spannungs-
0E1D 23         2010        INC  HL           ;      quellen ausge-
0E1E 7E         2020        LD   A,(HL)       ;      geben.
0E1F 23         2030        INC  HL           ;
0E20 E3         2040        EX   (SP),HL      ;
0E21 CD350E     2050        CALL STROBE
0E24 07         2060        RLCA ; Bitposition zum
0E25 07         2070        RLCA ; umschalten setzen
0E26 07         2080        RLCA
0E27 C5         2090        PUSH BC
0E28 47         2100        LD   B,A
0E29 3EC7       2110        LD   A,#C7        ; vorbereitung der
0E2B B0         2120        OR   B            ; Masken
0E2C 4F         2130        LD   C,A
0E2D DB02       2140        IN   A,(#02)      ; alten Steuerwert
0E2F B0         2150        OR   B            ; lesen und maskieren
0E30 A1         2160        AND  C
0E31 C1         2170        POP  BC
0E32 D302       2180        OUT  (#02),A      ; nur die Bits 3,4,5
0E34 C9         2190        RET  ; werden verändert
                2200 ;-----------------------------------------------------
```

Das Unterprogramm VOLT schaltet die Teilstücke die im Taktdiagramm BILD 33 Seite 91 mit U1 EIN, U2 EIN, U3 EIN überschrieben sind. Dazu holt es die drei Variablen, die nach dem Unterprogrammaufruf angegeben sind :

```
CALL VOLT
DEFB #50,#00,#01
```

Dabei ist #50 der Wert für Port A, #00 der Wert für Port B, und #01,der Wert für Port C.(siehe BILD 22 und 20). #50 und #00 können so direkt ausgegeben werden; Das Byte für Port C muß vorher dreimal links geshiftet, und mit dem alten Steuerbyte von Port C maskiert werden, um nur die Bits 3, 4,u.5 zu ändern. Das o.g. Beispiel bewirkt nun folgendes :

- Ausgabe des 10- Bit Digitalwertes Null,-> $U_{HH} \approx 3V$
- Diesen Wert auf Spannungsquelle U (#50)
- Die Spannungsquellen U_2 und U_3 werden ein, und U_1 ausgeschaltet, unabhängig von ihrem eingespeicherten Digitalwert. (#01, siehe auch Programmlisting im Anhang)

3.3.1.3 Beschreibung der Expandersteuerung

Die Portexpanderbausteine wurden bereits in Kapitel 3.1.1.2 Seite 19 , und Kapitel 3.2 Seite 24 beschrieben. Das Setzen einer der 40 Ausgabeleitungen erfolgt durch Einschreiben eines Steuerbytes in Port A nach dem, im BILD 23 angegebenen Format, und durch Schalten des Bit 0 Port C. (Strobe, siehe BILD 22).

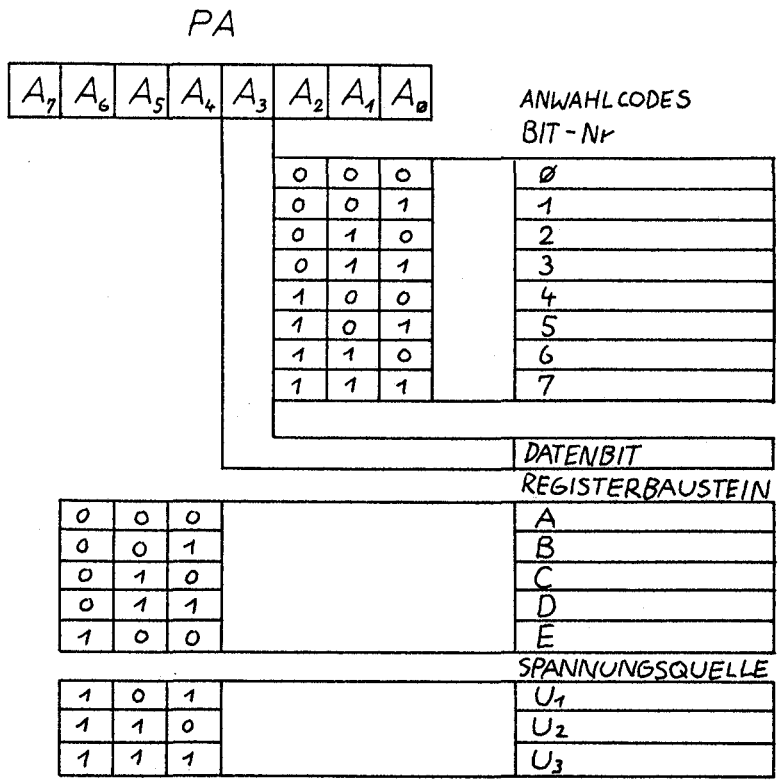


BILD 23 : Byteformat zur Anwahl einer aus 40 Leitungen mit Datenausgabe.

Das Unterprogramm zum Schalten der Leitung PC 0 hat folgenden Wortlaut :

```
                2200 ;
                2210 ; Unterprogramm STROBE
0E35 F5         2220 STROBE PUSH AF
0E36 DB02       2230        IN   A,(#02)      ; alten Steuerportwert
0E38 E6FE       2240        AND  #FE          ; lesen und maskieren
0E3A D302       2250        OUT  (#02),A      ; Bit 0 ist aktive Flanke
0E3C CBC7       2260        SET  0,A          ; LS 155 sperren
0E3E D302       2270        OUT  (#02),A
0E40 F1         2280        POP  AF
0E41 C9         2290        RET
                2300 ;
```

Zusammenfassung :

Der Strobe schaltet das Datenbit was an Port A, Bit 3 ausgegeben wurde, an die entsprechend angewählte Ausgabeleitung (BILD 23) weiter. Wenn der Inhalt des Halbbytes A4 - A7 größer ist als #4, dann schaltet das Unterprogramm STROBE den Wert auf Port B und A0 - A1 in eines der in A4 - A6 selektierten D/A- wandler.

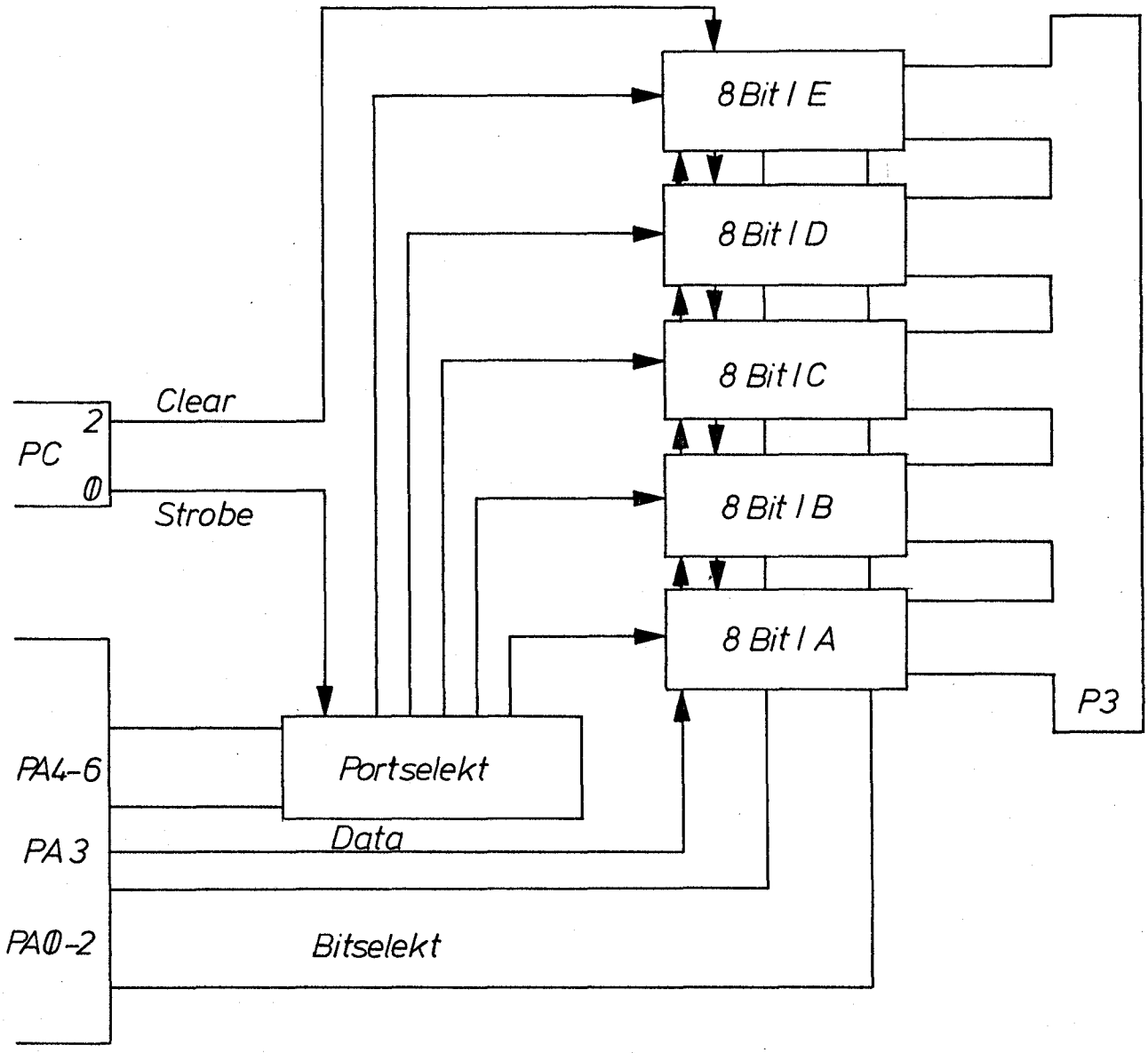


BILD 24 : Blockschaltbild der Portexpandersteuerung

3.3.2. Beschreibung der Treiberkarte

Die Treiberkarte stellt den zweiten Teil der Programmierhardware dar (siehe Zeichnung Nr. 7 , Anhang). Das in Kapitel 3.1.1.2 aufgestellte Lösungskonzept für die Zustandsaufschaltung mußte leider verworfen werden. Nach Fertigstellung der Interfacekarte waren die vorgesehenen Treiberbausteine 75 491 noch nicht verfügbar. Ausgehend vom Vorhandensein der Portexpanderbausteine sieht das sofort geänderte Konzept wie folgt aus :

3.3.2.1 Schaltung zur Realisierung der logischen Zustände an den PAL - Pins 1-11,13

Es gibt insgesamt 2 St. ULN 2003 , von denen 12 Transistoren benutzt sind, deren Basis von den 'LS 259 gesteuert werden. Drei St. NE 590 steuern zusammen 22 St. diskrete Darlingtontransistoren. Bei den Transistoren T1 und T12 handelt es sich um Leistungsdarlington, die an den Pins 1 und 13 den Strom für das Brennen der Verify-Protect-Sicherung (I_P = 400 mA) schalten müssen. Sonst wurden die Typen BC 516 verwendet.

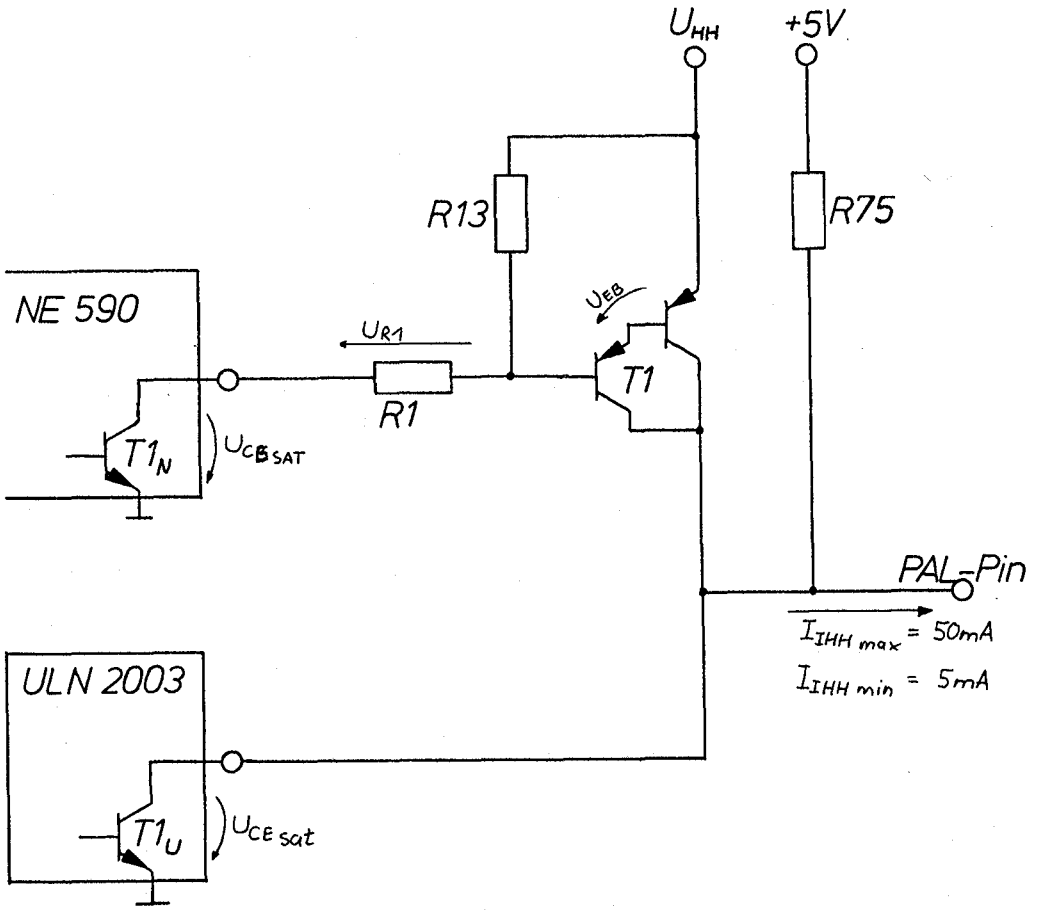


BILD 25 : Schaltung zur Dimensionierung der Widerstände.

Transistorkenndaten :

NE 590 : Ic_{max}= 250 mA, Uce_{0max}= 30 V

ULN 2003 : Ic_{max}= 500 mA, Uce_{0max}= 30 V

T1 BC 516 : Ic_{max}= 400 mA, Uce_{0max}= 30 V, B = 30 000

T1 BD 646 : Ic_{max}= 8 A , Uce_{0max}= 60 V, B < 750

Dimensionierung der Widerstände R1, R13, R , durchgeführt für Transistor BC 516 :

H - Zustand :

T1 gesperrt -> T1 gesperrt, T1 gesperrt.

Die hohe Stromverstärkung des BC 516 ermöglicht sein Durchschalten mit dem Reststrom des $T1_N$:

$$I_{B_{min}} = \frac{Ic_{T_1}}{B}$$

$$I_{B_{min}} = \frac{5mA}{30\ 000} = \underline{0{,}16\ \mu A}$$

Zum Sperren des BC 516 ist R13 mit 10 k gewählt.

HH - Zustand :

T1 leitend, -> T1 leitend , T1 gesperrt

NE 590 : Ic_{T1_N} = 250mA

$$R1 = \frac{U_{HH} - U_{EBT_1} - U_{CB_{SAT\,T1N}}}{Ic}$$

$$R1 = \frac{12V - 1{,}2V - 0{,}6V}{250mA}$$

$$\underline{R1 = 40{,}8\,\Omega}$$

R1 gewählt : 10k ->

$$I_{BT_1} = \frac{12V - 1{,}2V - 0{,}6V}{10k}$$

$$\underline{I_{BT_1} = 1{,}02\ mA}$$

Damit wird T1 sicher durchgesteuert :

$$Ic_{T1} = 50 \text{ mA}$$

$$I_{BT1min} = \frac{50\text{mA}}{30\ 000} = 1{,}6 \text{ uA}$$

Die gleiche Dimensionierung gilt auch <u>für BD 646 :</u>

Ic_{T1} = 400mA (Brennen der Veryfy-Potect-Sicherung)

$$I_{BT1_{min}} = \frac{400 \text{ mA}}{750} = 0{,}53\text{mA}$$

mit R1 = 10 k ist I_{BT1} = 1,02 mA, also noch etwa zweifache Sicherheit.

L - Zustand :

T1 gesperrt -> T1 gesperrt und T1 leitend :

ULN 2003 : Ic_{T1U} = 500mA

$$R75_{min} = \frac{5V}{500 \text{ mA}} = 10\,\Omega$$

R gewählt 2,7k .

Die Spezifikation des ULN 2003 gibt eine Spannung Uce_{sat} = 1,1 V an, für einen Kollektorstrom Ic_{T1U} = 100mA. Die L - Zustandsspezifikation für PAL , ist angegeben mit U_{ILmax} = 0,8 V (siehe PAL- Datenblätter im Anhang). Der L-strom für R75 muß deutlich kleiner als 100mA sein.

Damit ist $Ic_{T1U} = \frac{5V - Uce_{T1U_{sat}}}{R}$

$$Ic_{T1U} = \frac{5V - 1,2V}{2,7\ k} = 1,4\ mA$$

Eine Messung der Spannung Uce_{T1U} ergab einen Wert von :

$$Uce_{T1U} = 0,6\ V$$

Damit ist die Bedingung für den L - Zustand am PAL erfüllt.

Diese Berechnungen beziehen sich auf Dimensionierungsschaltplan BILD 25.

3.3.2.2 Schaltung zur Realisierung der Zustände an PAL-Pin 14 - 23

Bild 26 zeigt den, den folgenden Berechnungen zugrundeliegenden Teilschaltplan, der (wie auch BILD 25 für alle PAL-Pins 1 - 11, und 13) für alle PAL-Pins 14 - 23 realisiert werden muß. Da zur Aufschaltung des Zustandes HH das gleiche Konzept wie BILD 25 benutzt wird, und die gleichen Lastströme gelten (vergl. Kapitel 3.1.1.3) kann auch die Dimensionierung der Widerstände R45 und R25 für den Transistor T13 (ebenfalls BC 516) übernommen werden.

Der Spannungskomparator erhält an seinem negativ wirkenden Eingang zwei Pegel, U_{ref_L} und U_{ref_H} , die mit einer Leitung NE 590, R88, R89 und einer Zenerdiode ZPD 2,6 realisiert werden. Die Treiberkarte enthält drei Bausteine LM 339. Damit stehen insgesamt 12 Komparatoren zur Verfügung. 10 St. Komparatoren bekommen nun an ihrem negativ wirkenen Eingang gemeinsam den Pegel U_{ref} aufgeschaltet, und können so die Spannungen an allen PAL-ausgängen während eines Verifikationszykluses messen. (vergl. Taktdiagramme BILD 11 und BILD 34 und Beschreibung des Programmes PALPGM , Abschnitt 4.3.2.1).

Dimensionierung der Widerstände R87 ,R88 :

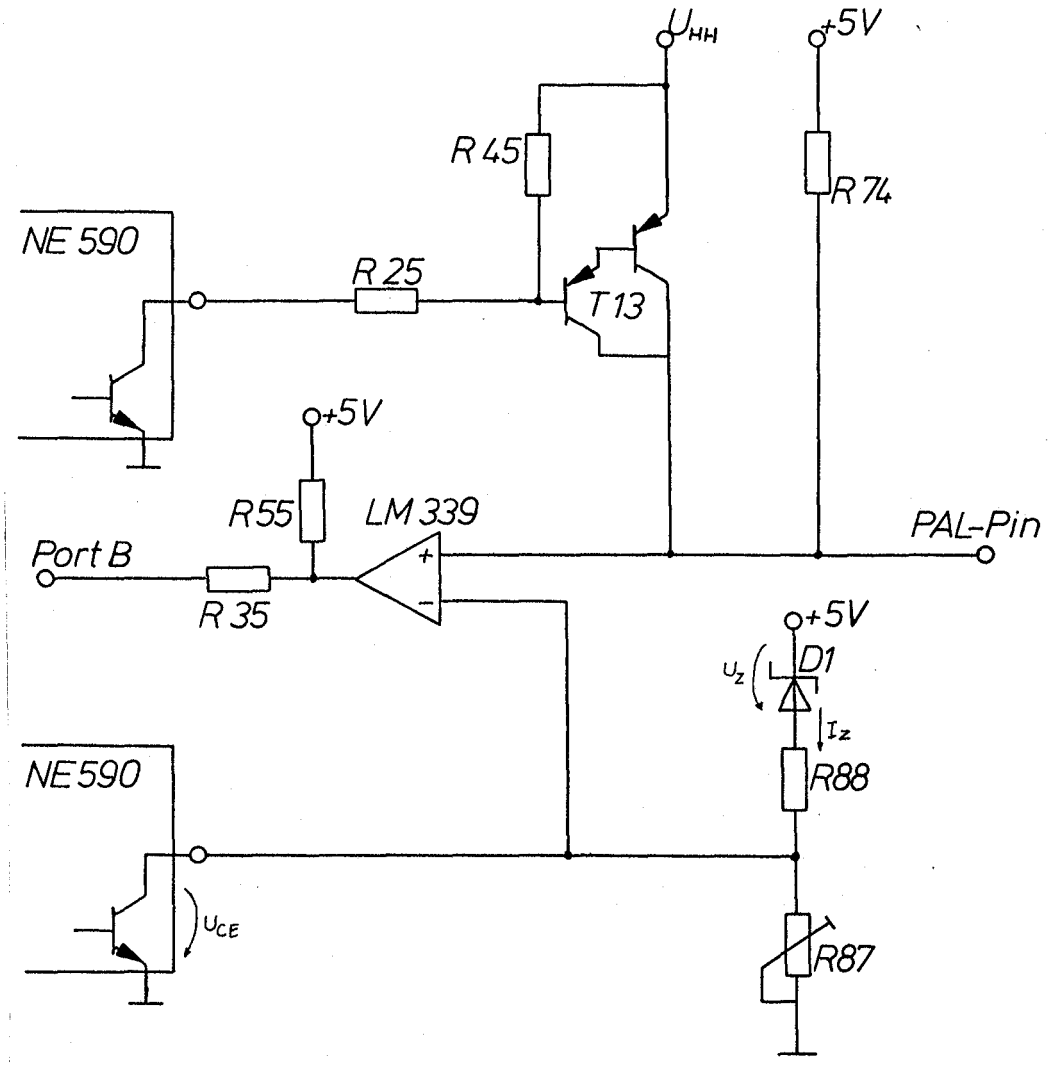


BILD 26 : So sind die PAL-Pins 14 - 23 beschaltet. (siehe auch Zeichnung Nr. 5 & 6)

Der Widerstand R74 ist in der Spezifikation des Z-zustandes (Three-State) mit 10 k vorgeschrieben. (siehe PAL-Datenblatt im Anhang).

Für $Uref_L$ gilt :

Iz = 2mA, Uce(NE 590) = 0,6 V

$$R88 = \frac{Uz - Uce(NE\ 590)}{Iz}$$

$$R = \frac{2{,}6V - 0{,}6V}{2\ mA} = 1\ k$$

Für Uref = 2,3 V gilt :

Spannungsteilergleichung :

$$Uref = (5V - Uz)\ \frac{R87}{R88 + R87} \quad \rightarrow$$

$$R87 = \frac{-\ Uref \times R88}{Uref - (5\ V - Uz)}$$

$$R87 = \frac{2{,}3\ V \times 1\ k}{0{,}1\ V} = 23\ k$$

gewählt für R : Trimmer 25 k

BILD 27a : PAL-sicherung im unprogrammierten-

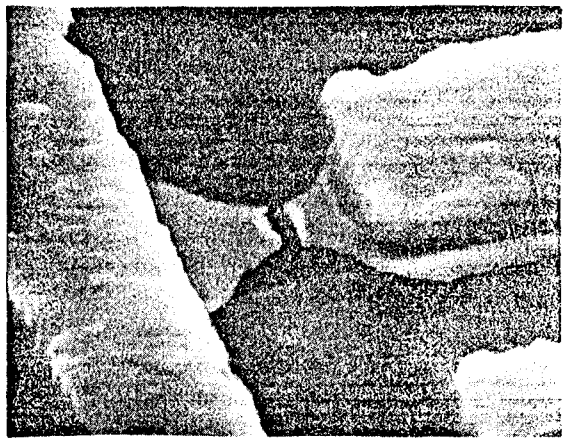


BILD 27b : PAL-sicherung und programmierten Zustand.

Zusammenfassung :

Den Sinn der Verwendung von Spannungskomparatoren beim Bestimmen des Pegels am PAL-ausgang während eines Verify-zykluses kann man sich anhand der BILDer 27 leicht klar-machen. Die in den Programmierspezifikationen angegebenen Toleranzen für die Programmierpulsdauer mit:

$$10\mu sec < Tp < 50\mu sec$$

und für die Programmierspannungspegel :

$$11{,}5V < U < 12V$$

sind schon sehr eng. Trotzdem ist es möglich, daß die Titan-Wolfram Sicherungsverbindung nicht vollständig ge-trennt wird. Der Ausgangspegel am Pin ist dann logisch nicht bestimmbar, (BILD 28) und der Programmierzyklus muß dann wiederholt werden (bis zu 5 mal ; siehe auch Programmiervorschrift im Anhang)

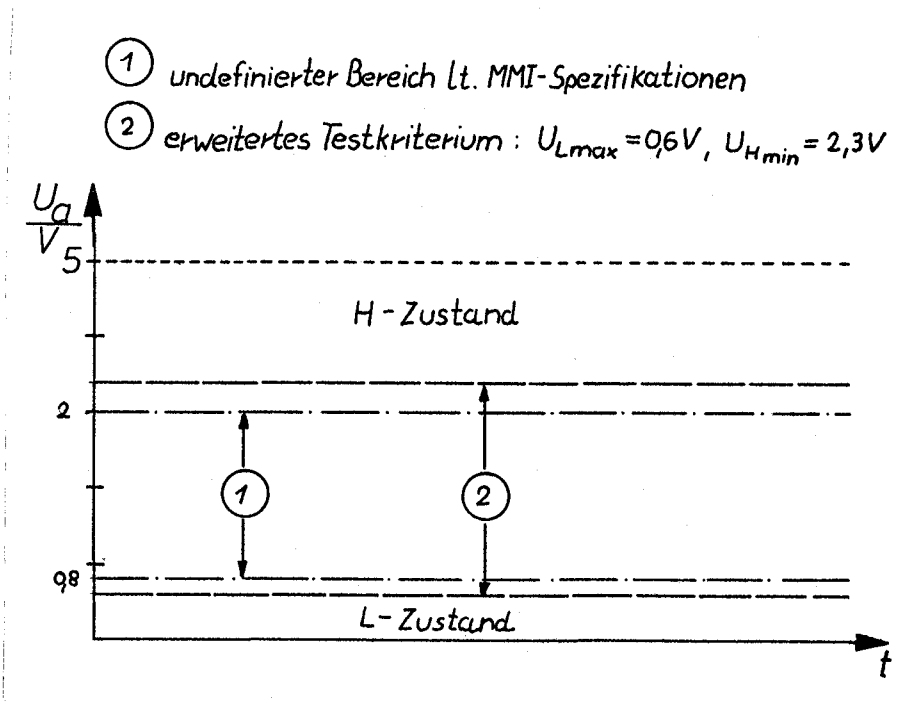


BILD 28 : Spannungs-Zeit-Diagramm der TTL-pegel. Die Span-nung am PAL-ausgang kann mit den Komparatoren auf Zulässigkeit getestet werden.

S O F T W A R E

4. Realisierung der Software

Kurze übersicht Kapitel 4

PUNKT	TITEL	SEITENNUMMER
4.1	Konzeptfindung und Realisierung	42
4.2	Erläuterungen der Software Pakete	49
4.3	Das PAL- Betriebssystem	70
4.4	Speicherbelegungspläne	98

4.1 Konzeptfindung und Realisierung

Gemäß unserer Aufgabenstellung und Forderung nach einem eigenständigen Bildschirm orientierten System ergaben sich für die Software folgende Aufgabenstellungen.

4.1.1 Ein-Ausgabe-Paket

4.1.1.1 Erweiterte Aufgabenstellung

- Entwicklung eines funktionalen und flexiblen Ein / Ausgabe Paketes , das neben den Treibern für die Periphergeräte auch die dazugehörigen Möglichkeiten zur Zeichenkettenausgaben und Zahlenwandlung bieten soll. Weiterhin soll dieses Programmpaket noch weitere Routinen enthalten, welche dem Softwareumfeld eine möglichst einfache und klare Strukturierung der Programme erlaubt. Da diese Routinen von überall erreichbar sein sollen muß dieses Programmpaket einen festen Einsprungverteiler haben.

4.1.1.2 Entwicklungsgang

Standardmäßig muß dieses Programmpaket natürlich die Treiber für die "Portbausteine der Kommunikation" beinhalten, auch die dazugehörigen Initialisierungs-Routinen (INITD, OTCHR, INCHR, SCAN).

Da auch ganze Zeichenketten verarbeitet werden sollen, werden Programme nötig welche diese Aufgabe Lösen (XOTSTR, OTSTR). Eine leichte Abwandlung von "XOTSTR" ist "XOTCHR", wobei nur ein Zeichen ausgegeben wird.

Die Routine zur Zeitverzögerung (TDEL) wurde widererwarten nicht benötigt, und so ist für dieses Programm nur der Einsprung vorgesehen.

Die Möglichkeit der leichten Anpassung an andere Terminal, (verschiedene Steuerkodes) wurde durch "MOD" verwirklicht.

Eine standardisierte Art eines Sprungverteiler- alogrithmus wurde durch das Programm "XFUNCT" geschaffen.

Das Ausgeben von Zahlenwerten machte die Routinen "OUTA" und "OUTHL" erforderlich, später stellte sich heraus, daß in einigen Fällen eine Darstellung in Dezimalzahlenwerten günstiger wäre (Einsprungpunkte "DEZA" & "DEZHL").

Die "MSKTS" Routine war ursprünglich für den Gültigkeitstest der Eingabedaten (z.B. im Monitor) gedacht, später wurde dieses Programm auch sehr oft dazu benutzt um den Tabellenplatzbedarf gering zu halten (siehe Benutzung im zusammenhang mit der Erzeugung der gültigen PRL und IPL).

4.1.2 Editor

4.1.2.1 Erweiterung des Pflichtenheftes

Wie sich schnell herausstellte war eine Korrektueinrichtung für die diversen Zeichenketten nötig.

Notwendige Funktionen

(a). Bewegungsfunktionen durch Cursor Tasten

(b). Löschen von Einzelzeichen, mit Zusammenschieben des verbleibenden Textes.

(c). Einfügen von Einzelzeichen, mit autom. Verlängern der Textzeile

(d). Textspeicher optimieren (begrenzter Speicherplatz)

(e). Zeilenende durch <ENTER> (Kode #0D)

4.1.2.2 Vorgehenweise bei der Entwicklung des Editors

Dieser Programmteil ist durch seine klare Aufgabenstellung gekennzeichnet. Er konnte daher fast ausnahmslos am Schreibtisch (mit der Hilfe einiger Ablaufdiagramms- Modelle) verwirklicht werden.

Hinweise zur Implementierung siehe 4.2.2

4.1.3 Monitor

4.1.3.1 Erweitertes Pflichtenheft

- Schaffen eines Monitors , der speziell für kleinere Systeme , d.h. mit möglichst wenig Speicherplatz auskommt und dabei sicher (Fehlbedienungen verhindern) und benutzerfreundlich bleiben soll. Die Kommunikation soll benutzerführent aufgebaut sein .

Welche Funktionen sind notwendig ?

(a). Inhalte von Speicherzellen anzeigen und verändern

(b). Darstellung von Speicherblöcken (überblick über Speicherbelegungen).

(c). Starten von Programmen (Testen der Software)

(d). Einlesen von Portbausteinen
(Testen der Hardware)
(e). Ausgeben an Portbausteine

(f). Einsprung in des eigentliche Hauptprogramm

4.1.3.2 Abfolge der Entwicklung

Die Programme zu den Anforderungen nach c - f waren recht schnell inplementiert (G-,I-,O- und U- Befehl).

Der Entwicklungsgang der Grundstrukturen zu den Punkten a+b war zwar bald abgeschlossen, doch durch eine intensive Erprobung ergaben sich noch weitere Aspekte, die auch noch verwirklicht werden sollten. So bekammen diese zwei Routinen noch einige Erweiterungen (multifunktionszeile bei (a), Kopfzeile für (b), vor-/ und rück-blättern bei (b) u.s.w. Bedienungsfunktionen der entgültigen Version siehe bei B.1)

4.1.4 PAL- Betriebssystem

4.1.4.1 Erweiterung des Pflichtenheftes

- Das Hauptbetriebssystem.

Dieses soll (in Anlehnung an das Betriebssystem CP/M) den CCP (Consol Comand Prozessor) und den BPOS (vergleiche BDOS bei CP/M) verwirklichen.

BPOS=Basic Pal Operating System .

In diesen System soll speziell der CCP sehr flexibel gehalten werden , um auch nachträglich von Benutzer kommende Wünsche erfüllen zu können.

Speziell zum BPOS

(1). Verarbeitungmöglichkeit verschiedener Quelldaten

- Der Befehlssatz ist um die entsprechenden Anweisungen zu erweitern, um :

  (a) direkt über Tastatur
  (b) von der externen Einheit

  die Programmierdaten zu empfangen.

(2). Abfrage des PAL- Types

- Eingabe des PAL - Types in einer editierbaren Zeile, inklusive melden von Fehleingaben.

  Fehlermeldung : "PAL - unbekannt"

(3). Möglichkeit zum nachträglichen Editieren der Programmierdaten

- Dieser Programmteil ist optional und soll je nach Anwendungsfall das Editieren der der Eingabezeile ermöglichen, oder die Matrix des PAL's direkt einlesen, diese in einer Stern- Strich-Matix darstellen und dort das ändern erlauben.

(4). Eingabe der Programmierdaten nach Kapitel 2 (Abschnitt 3)

- Wie in Kapitel 2 (Abs.3) angeführt, werden nur die nicht zu trennenden Verbindungen angegeben, alles weitere wird errechnet.

- Für die zwei Grenzfälle (alle Sicherungen trennen-/ nicht- trennen) sollen folgend Bedienungsvereinfachungen eingeführt werden :

(a) Sollen alle Sicherungen einer PRL getrennt werden soll die Eingabe von <ENTER> ausreichend sein.

(b) Soll keine Sicherung einer PRL getrennt werden, soll dieses durch das Stern-Zeichen ("*" geschehen.

(5). Aufbereiten der Programmierdaten

Die in der oben genannten Routine eingegebenen Werte stellen noch keine vollständige Information dar, die an das Programmiergerät ausgegeben werden kann. Eine Matrixroutine berechnet die Programmiertabelle aus den PAL- Typ spezifischen Konstanten und schließt die eingegebenen Variablen aus dieser Tabelle aus. Zweckmäßigerweise soll es dann zwei zusammenhängende Tabellen geben, und zwar eine für alle zu selektierende PRL's und eine für alle zu selektierende IPL's. Beide Tabellen beinhalten dabei die hexadezimalen Werte der PRL- bzw. IPL- nummern.

(6). übersichtliche Darstellung der Programmiermatrix auf Bildschirm oder Drucker

Um eine echte Kontrollfunktion in diesem Programm mitzuverwirklichen, soll die erzeugte Programmiertabelle direkt als Datenquelle benutzt werden (nicht der Eingabestring).

Einige einfache PAL- Typen (z.B. 12H6,10L8,...) haben eine unvollständige Matrix, d.h. es fehlen RPL's und auch IPL's. Diese Tatsache soll bei der Darstellung der Matrix in folgender Weise berücksichtigt werden :

(a) Nicht vorhandene PRL's brauchen weder Programmiert noch verifiziert werden (,da sie keine Information darstellen).

(b) Nicht vorhandene IPL's sollen auch nicht dargestellt werden. Da aber ein einfaches Weglassen der IPL's die Gesamtdarstellung unübersichtlich macht, solle an Stelle der nicht vorhandenen IPL's ein Leerzeichen eingesetzt werden.

Diese Darstellung entspricht dem, von der Firma MMI (Monolithic Memories) vorgeschlagenen Format.

(7). Programmierzyklus & verify

Die eigentliche Programmierungshauptschleife kann nun die in Pkt. 5 errechneten Werte mit geringen Modifikationen an die Hardware ausgeben, und so das Programmierungstaktdiagramm realisieren. Dabei muß die Reihenfolge der Selektionsschritte strikt eingehalten werden (siehe Programmierungsvorschrift im Anhang). Die Programmierungshauptschleife muß auch den Verifikationszyklus enthalten.

4.2.1 Erläuterungen zum Ein/Ausgabe Paket

Allgemeines :Die in den Beispielen benutzten Assembler Programme sind in Z80-ASSEMBLER menonic von Zilog geschrieben.

```
**********************
**  Routine OTCHR   **
**********************
```

(Siehe Seite 152 bei Zeilennummer 1720)

Dieses Programm gibt den Inhalt des Akku an die Ausgabeeinheit aus. Der Akku wird dabei nicht verändert , und ohne weitere überprüfungen ausgesendet. Achtung Register C wird mit den Inhalt des Akku geladen, dieses geschieht, damit das Programm CP/M kompertipel , d.h. die Funktion und Bedienung der Register mit der entsprechenden Routine aus den BIOS übereinstimmt.

Beispiel 1 :

```
                    LD   A,#31   ;ASCII-Kode für "1"
                    CALL OTCHR   ;"1" ausgeben
```

Das Zeichen "1" wird an die Ausgabeeinheit ausgegeben.

```
**********************
**  Routine XOTCHR  **
**********************
```

(Siehe Seite 150 bei Zeilennummer 1050)

Mehr als viele Worte verdeutlicht folgendes Beispiel die Benutzung der Routine XOTCHR

Beispiel 2 :

```
                    CALL XOTCHR
                    DEFB "B
          MARK      ...
                    ..
```

In diesem Fall wird das ASCII Zeichen für "B" ausgegeben . Das Programm wird bei dem Label MARK fortgesetzt.

```
**********************
**  Routine XOTSTR  **
**********************
```

(Siehe Seite 150 bei Zeilennummer 0770)

Auch hier soll ein kleines Beispiel die Wirkungsweise dieses Programmes verdeutlichen.

Beispiel 3 :

```
                    CALL XOTSTR
                    DEFM "DAS IST EIN BEISPIEL"
                    DEFB CR,LF,END
        MARK        ....
                    ..
                    .
```

Was wird getan ? . Alle Bytes die nach dem Aufruf der Routine XOTSTR folgen, werden als ASCII Zeichen an die Ausgabeeinheit ausgegeben. Dieses geschieht so lange bis das Endekennzeichen gelesen ist. (END hat den Kode 00)
Das Programm wird nach Ausgabe der Zeichenkette bei dem Label MARK weitergeführt.

Steuerkodes innerhalb der Zeichenkette

```
        CR      => Wagenrücklauf,Zeilenstart
        LF      => Neue Zeile
        IND     => Indizierte Zeichenkette , Adresse
        REP     => Wiederhole Zeichen , Anzahl , Zeichen
```

Benutzung des Steuerkodes REP

Beispiel 4 :

```
                    CALL XOTSTR
                    DEFB REP,10,"*,END
        MARK        ....
                    ..
                    .
```

In Beispiel 4 wird das Zeichen "*" zehn mal ausgegeben.
Das Programm wird bei dem Label MARK weitergeführt.

Benutzung des Steuerkodes IND

Beispiel 5 :

```
                    CALL XOTSTR
                    DEFB IND
                    DEFW TEXT
                    DEFB END
        MARK        ....
                    ..
                    .

        TEXT        DEFM "ADRESSIERT MIT IND"
                    DEFB END
```

Der IND Kode bewirkt, daß die nächsten zwei Bytes als Word (Adresse) für einen weiteren Textbereich benuzt werden. Dieses kleine Programm wird also "ADRESSIERT MIT IND" ausgeben.

Anmerkung :Steuerkodes und Texte können durchaus auch gemischt in einer Zeichenkette auftreten. Der Programmteil zu REP-Funktion befindet sich auf Seite 155 bei Zeilennummer 3330 , der zu IND-Funktion auf Seite 155 bei Zeilennummer 3220 .

```
**********************
**  Routine OTSTR   **
**********************
```

(Siehe Seite 150 bei Zeilennummer 0890)

Dieses Programm verarbeitet die gleichen Zeichenketten und Steuerkodes wie das Programm XOTSTR , jedoch folgt der String nicht nach dem Aufruf des Programmes , sondern ein Register dient hier als Zeiger auf die Zeichenkette. Die Startadresse des Strings wird im Registerpaar HL übergeben.

Beispiel 6 :

```
                    LD   HL,TEXT
                    CALL OTSTR
        MARK        ....
                    ..
                    .

        TEXT        DEFM "TESTTEXT"
                    DEFB END
```

Nach Ausgeben des Textes wird das Programm bei dem Label MARK weitergeführt. Achtung das Registerpaar HL wird verändert. HL zeigt auf die Adresse des Textendes+1 .

```
**********************
**  Routine INCHR   **
**********************
```

(Siehe Seite 151 bei Zeilennummer 1430)

Diese Routine wartet auf ein Zeichen von der Tastatur.
Der Kode des eingegbenen Zeichens wird dem Akku übergeben.
Dieser Softwaretreiber ist für die SIO 8251 geschrieben.

```
**********************
**  Routine  SCAN   **
**********************
```

(Siehe Seite 151 bei Zeilennummer 1570)

Die Tastatur wird nur einmal abgefragt und nach setzen der übergabeparameter zurückgesprungen.

- Taste wurde gedrückt : Der Akku wird mit den empfangenen Zeichen geladen , das Carry Flag wird gelöscht .
- Taste nicht gedrückt : Das Carry Flag wird gesetzt , der Akku wird mit den Wert Null geladen.

```
**********************
**  Routine XFUNCT  **
**********************
```

(Siehe Seite 153 bei Zeilennummer 2210)

Der Name besagt bereits einiges XFUNCT =EXECUTE FUNCTION .
Der Sprungverteiler der je nach Funktionskode verzweigt benötigt folgende Verteilerliste. Bevor XFUNCT aufgerufen wird muß der Akku mit den zu testenden Kode geladen sein.

Beispiel 7 :

```
                    CALL XFUNCT
                    DEFB Kode1
                    DEFW EX1
                    DEFB Kode2
                    DEFW EX2
                    DEFB Kode3
                    DEFW EX3
                    .    .
                    .    .
                    .    .
                    DEFB END
                    DEFW ERROR
        MARK        ....
                    ..
                    .
```

Nach dem Aufruf der XFUNCT Routine folgt eine Byte / Word Tabelle . Anders als bei den anderen Tabellen endet diese nicht mit dem END Kode sondern mit einer weitern Adresse. Diese Adresse hat eine besondere Aufgabe, sie wird benutzt wenn der zu übergebende Kode nicht in der Tabelle gefunden werden kann. Findet sich der übergebene Kode in der Tabelle, so wird die darauf folgende Adresse als Sprungvektor benutzt und das Programm dort fortgesetzt. Dieses Programm muß mit einen 'RET' abschließen, daraufhin wird das Programm bei der Adresse MARK weitergeführt.

```
**********************
**  Routine   MOD   **
**********************
```

(Siehe Seite 159 bei Zeilennummer 2730)

Dieses Programm sucht in einer Tabelle nach einem Kode der ihr übergeben wurde. Wird dieser Kode nicht gefunden dann kehrt der Akku ohne Änderung zurück (Die Routine ist dann auch beendet). Wird dagegen eine übereinstimmung mit einen der entsprechenden Bytes aus der Tabelle festgestellt, so wird der Akku mit dem darauffolgenden Byte geladen.

Eine praktiche Anwendung soll die Nützlichkeit und Anwendung dieses Programmes verdeutlichen. (Beispiel 8)

Beispiel 8 :

```
                CALL INCHR     ;hole Zeichen von Tastatur
                LD   HL,LIST   ;setze Listenzeiger
                CALL MOD       ;tausche falls nötig !
MARK            ....
                ..
                .


LIST            DEFB "Y,"Z
                DEFB "Z,"Y
                DEFB "y,"z
                DEFB "z,"y
                DEFB END
```

Einsatz als Tastenbelegungsvertauscher, hier werden die Buchstaben y + z vertauscht ,d.h. aus einer amerikanichen Tastaturbelegung eine deutsche gemacht.

```
**********************
**  Routine  MSKTS  **
**********************
```

(Siehe Seite 159 bei Zeilennummer 5440)

Das übergebene Zeichen wird darauf getestet ob es sich innerhalb, oder außerhalb der Maske befindet. Das ist an sich nichts besonderes, jedoch das interessante ist der Aufbau der Maske. Die Maske hat folgenden grundsätzlichen Aufbau.

Beispiel 8 :

```
MASK            DEFB von,bis,von,bis,...,von,bis,END
```

In der praktichen Anwendung kann man so sehr schön Masken für verschiedene Zahlensysteme aufstellen. (siehe Beispiel 9)

Beispiel 9 :

```
OKTAL           DEFB "0,"8,END    ;oktalles  Zahlensystem
DEZIMAL         DEFB "0,"9,END    ;dezimales Zahlensystem
HEX             DEFB "0,"9,"A,"F,END ;sed.  Zahlensystem
```

Wie wird getestet ? :Aus der Tabelle sieht man,daß immer ein Bytepaar zusammengehört (von , bis). Das erste Byte ist der Startwert einschließlich sich selbst ,dort wird der Akku mit "von" auf größergleich getestet.

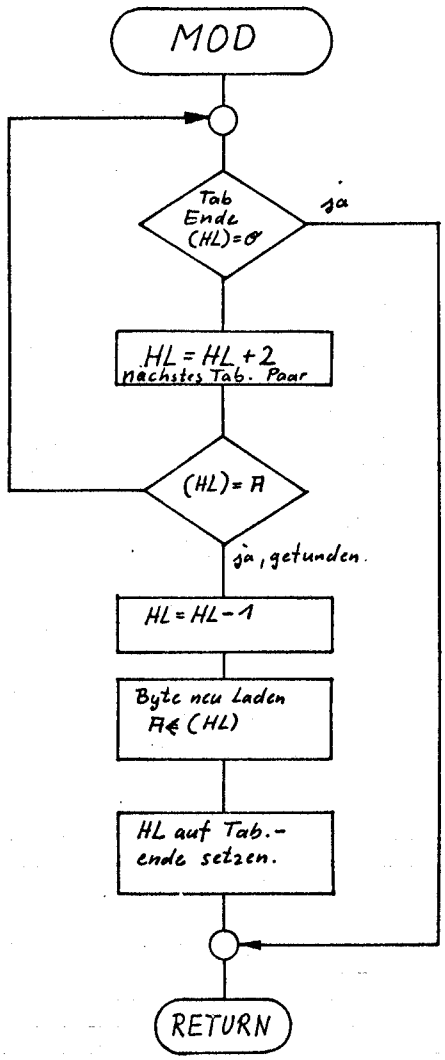
War der Akku kleiner , so wird dieser Test mit den nächsten Bytepaar durchgeführt usw. bis das END Kennzeichen gelesen wird.

War der Akku gleich , wird das Carry Flag gelöscht und zurückgesprungen.

War der Akku größer , wird er mit den Wert des Bytes "bis" getestet. Wenn der Akku nun kleiner ist dann befindet sich der Wert innerhalb der Maske, das Carry Flag wird gelöscht und die Routine beendet, war er größer so wird mit der Testung des nächsten Bytepaares begonnen.

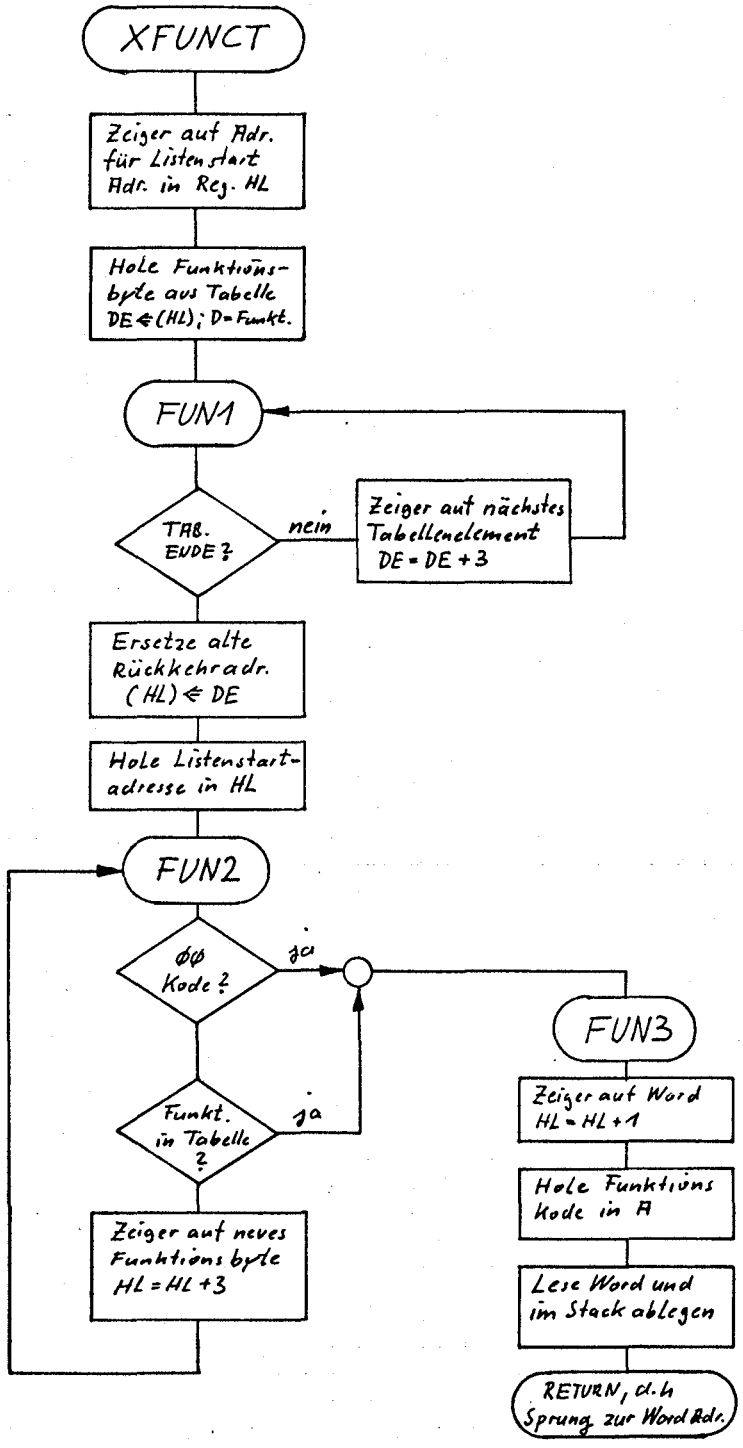
Weitere Auskünfte über die Wirkungsweise der Routinen geben das Listing "Ein- Ausgabe- Paket" und die Ablaufdiagramme 4.2.1.1 bis 4.2.1.3 .

4.2.1.1 Ablaufdiagramm "MOD"

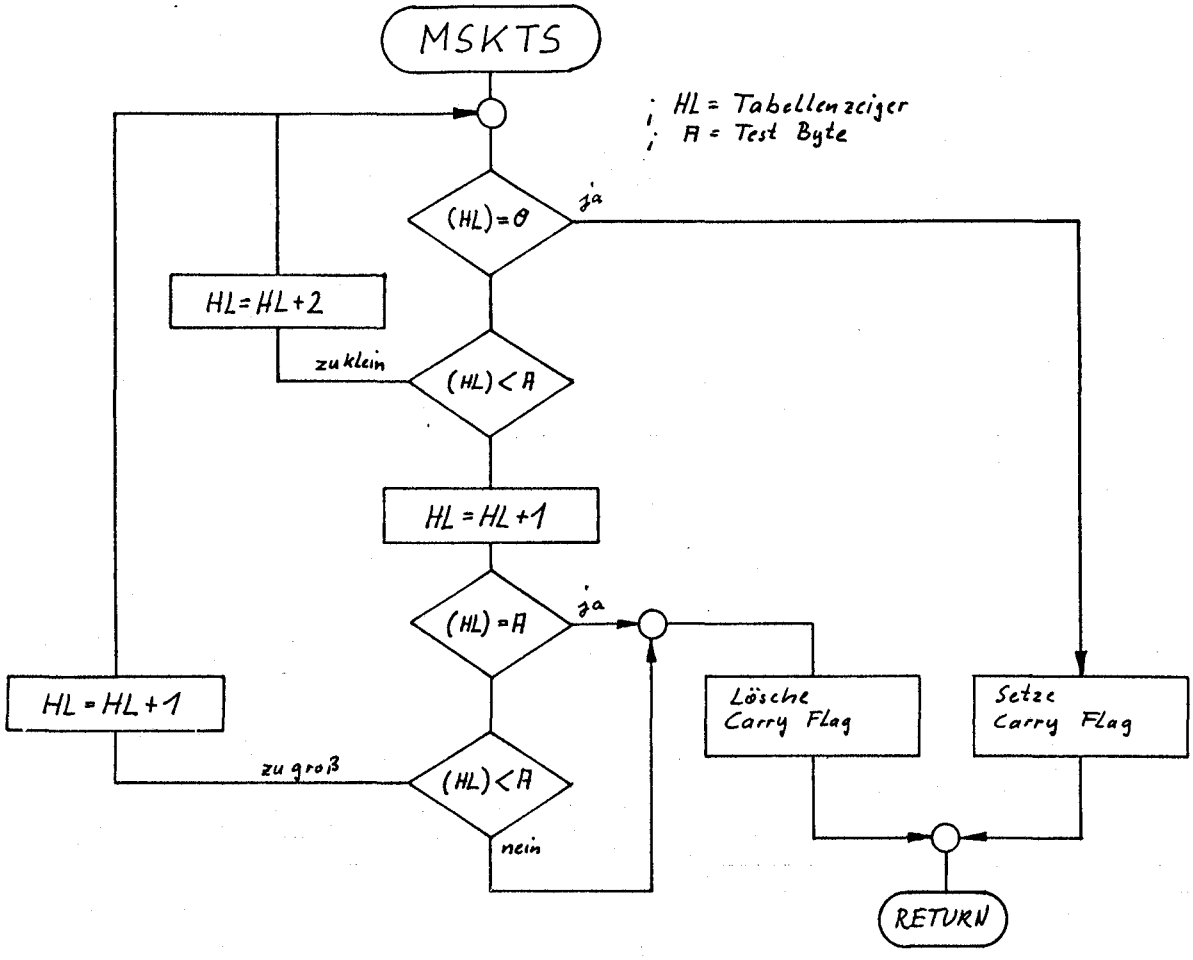


; HL = Tabellenzeiger
; A = Test Byte

4.2.1.2 Ablaufdiagramm "XFUNCT"



4.2.1.3 Ablaufdiagramm "MSKTS"



4.2.2 Erläuterungen zum Editor

Der Editor hat durch die vielen Eingaben innerhalb des PAL-Betriebssystem eine zentrale Stellung, aus diesem Grund erfolgt eine gesonderte Behandlung in diesem Kapitel. (Sein Listing ist im Ein-Ausgabe-Paket enthalten).

Seine Bedienung ist durch eine einfache und klare Struktur gekennzeichnet, d.h. möglichst wenig aber effektive Befehle, in Form von Cursor Steuertasten u.s.w.

Von dem Programm sind folgende Steuerfunktionen unterstützt :

CL	:	Lösche Textzeile	(clear line)
CUR	:	Cursor rechts	
CUL	:	Cursor links	
INS	:	Zeichen Einfügen	(insert character)
DEL	:	Zeichen Löschen	(delete chr.)
RUB	:	Zurück & Leerzeich.	(rub out)
CR	:	Textzeile beenden	(carriage return)

Der Editor ist so aufgebaut, daß er durch einige Änderungen auch in einem "full screen editor" umgewandelt werden kann. In der jetzigen Version ist der Editor Zeilenorientiert. Er benutzt für seine gesamte Steuerung nur drei "Word" Speicherplätze, welche folgende Aufgaben haben.

(1) Speicherzellen "TST"

"TST" = Textstartadr.

Um nicht an einen festen Speicherplatz für den Textbuffer gebunden zu sein, wurde "TST" eingeführt. Durch die Variable ist es außerdem möglich, nach Zurückkehren des Editors, diesem sofort erneut mit einer erhöhten Textstartadr. aufzurufen. Auf diese Weise können größere Texte in den Speicher übernommen werden, ohne die einzelnen Textzeilen sofort verarbeiten zu müssen.

(2) Speicherzellen "TZ"

"TZ" = Textzeiger-Cursor

Da unser Editor mit einem Terminal zusammenarbeitet und diese den Nachteil haben, daß man von ihnen weder den Standort des Cursors noch eine Zeile auslesen kann. Aus diesem Grund muß im Speicher jede Bewegung des Cursors mitvollzogen werden, d.h. der Zeiger "TZ" reagiert auf jede Cursor Veränderung.

(3) Speicherzellen "TEND"

"TEND" = Textendeadr.

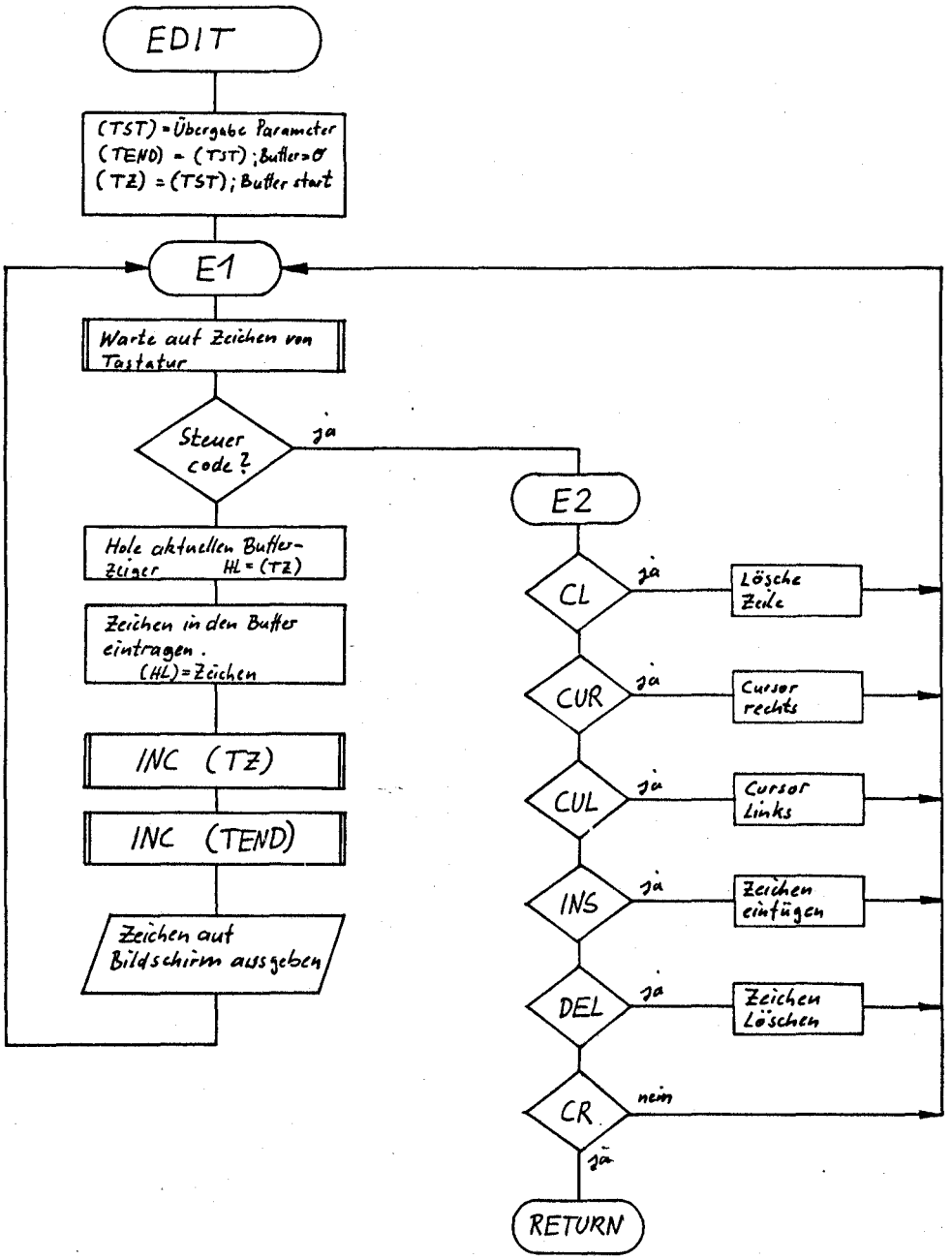
Dieser Zeiger markiert das Ende des Textbuffers. Durch diese Speicherzellen wird es ermöglicht, die verschiedenen Operationen auf Plausibilität zu überprüfen.

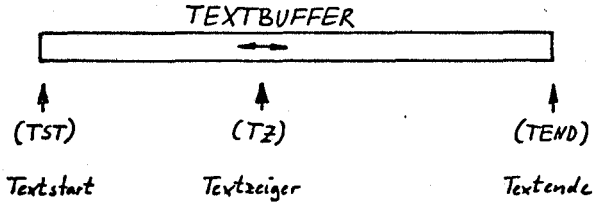
Weiterer Aufschluß über die Benutzung dieser Speicherzellen geben folgende Quellen:

- Ablaufdiagramme (nächste Seite) 4.2.2.1 bis 4.2.2.6
- Listing des "Ein-Ausgabe-Paket" ab Zeilennummer 3680

4.2.2.1 Ablaufdiagramm "EDITOR"

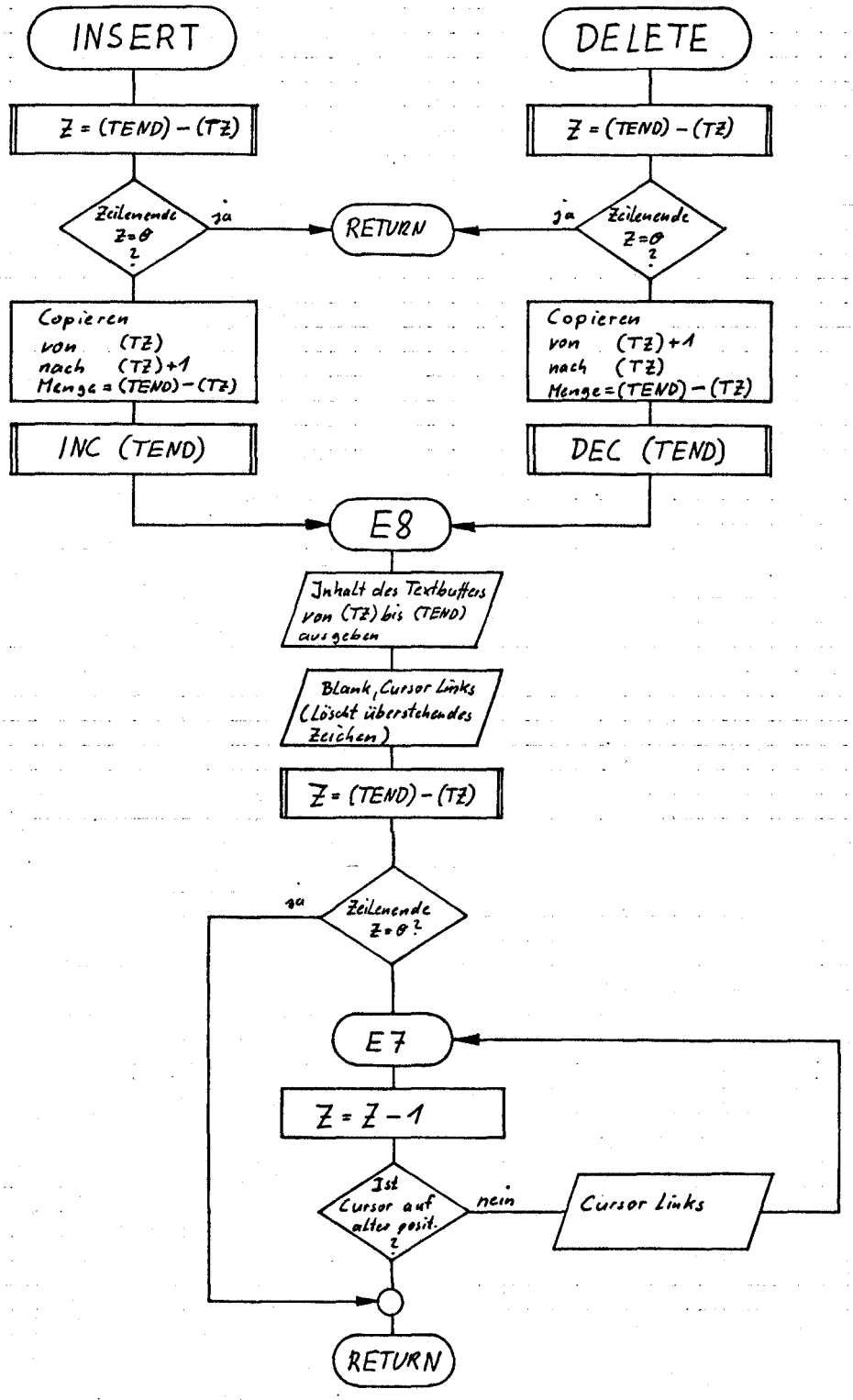
Hauptschleife





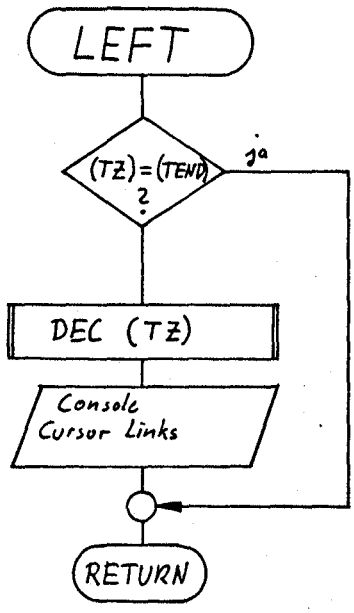
4.2.2.2 Ablaufdiagramm "EDITOR"

Insert & Delete



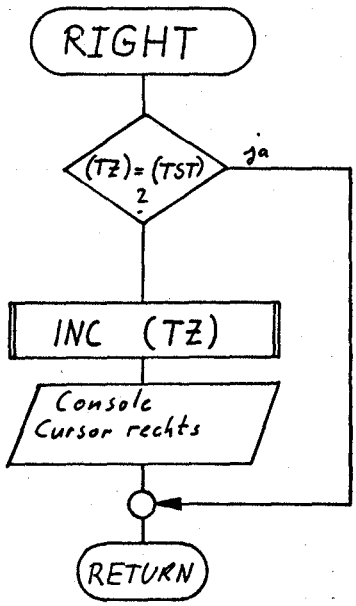
4.2.2.3 Ablaufdiagramm "EDITOR"

Cursor Links



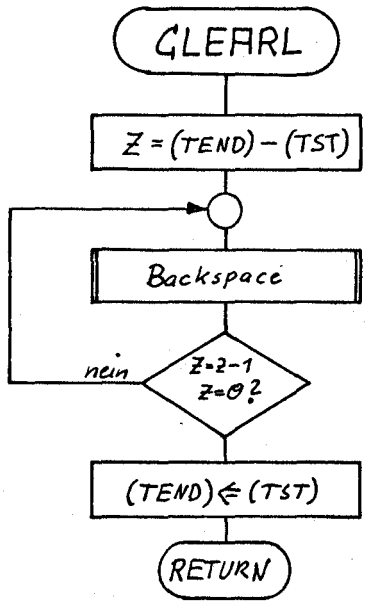
4.2.2.4 Ablaufdiagramm "EDITOR"

Cursor rechts



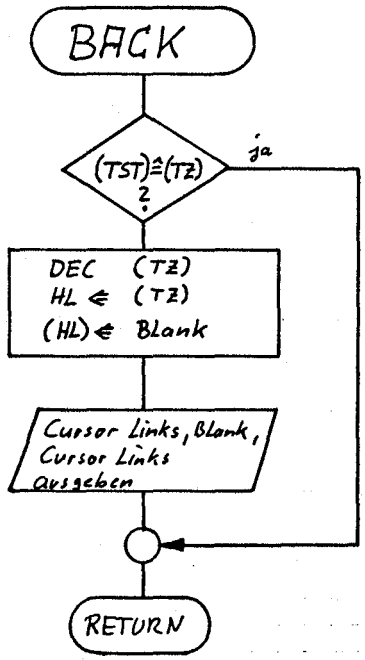
4.2.2.5 Ablaufdiagramm "EDITOR"

Lösche Zeile



4.2.2.6 Ablaufdiagramm "EDITOR"

Backspace



4.2.3 Erläuterungen zum Monitor

Konzeption : Der gesamte Monitor tätigt seine Aus- und Eingaben über das 'I/O Pack' ,d.h. um den Monitor auf einer anderen Rechnerkarte laufen zu lassen , muß lediglich die Routine INCHR,SCAN und OTCHR geändert werden. Diese Routinen befinden sich im 'I/O Pack'. Durch diese änderung können gleichzeitig die anderen Programme auch benutzt werden.

Bedienung : Durch eine fast 100% Benutzerführung ist der Monitor äußerst einfach zu bedienen. Auch die weiteren Eingaben werden auf Gültigkeit überprüft , hier werden falsche oder sinnlose Eingaben gar nicht erst angenommen. Das sind sehr nützliche Eigenschaften die leider die wenigsten Monitore besitzen.

Befehle : Es werden möglichst wenig Grundbefehle gebraucht (sind dann auch einfacher zu behalten).
Liste der Befehle :

Befehle Gesammtübersicht

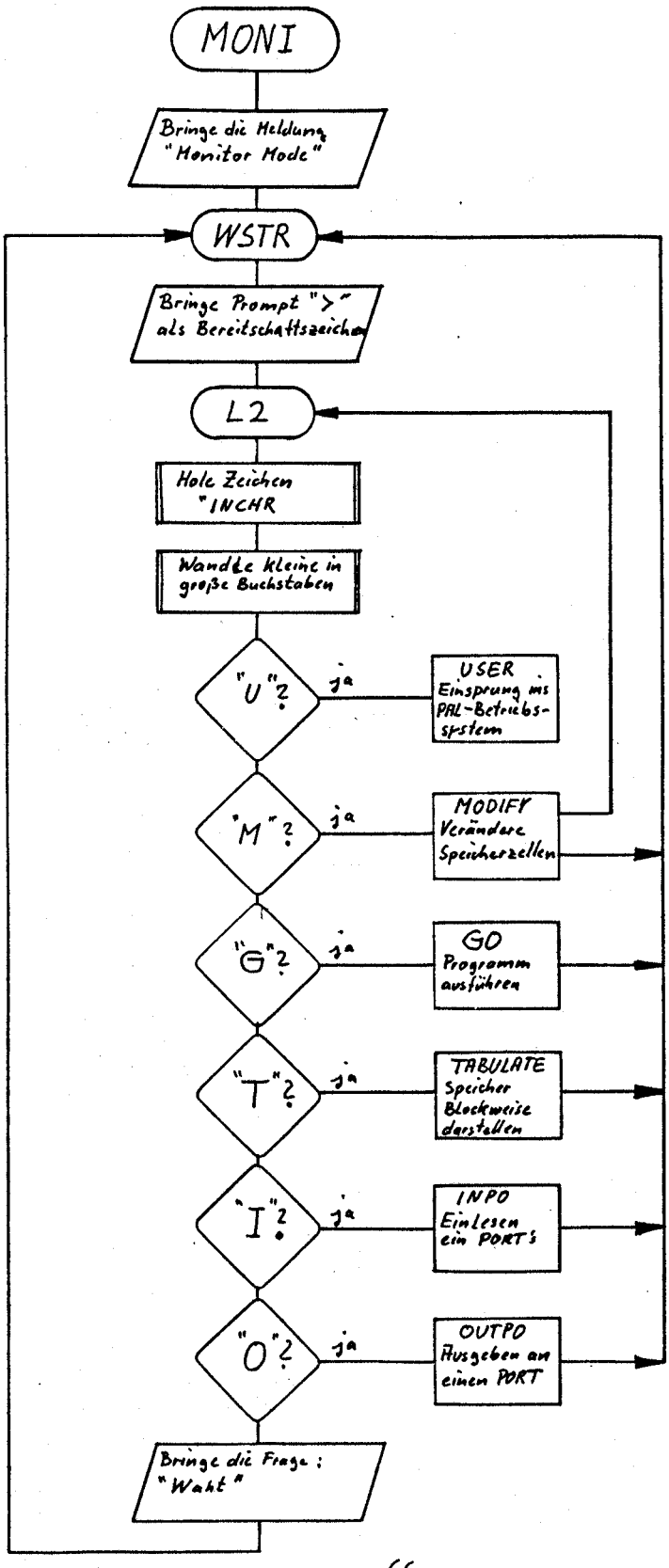
M	Memory modify
T	Tabulate memory
G	Go and execute program
O	Out to port
I	In from port
U	Userstart

Der "Memory modify" und der "Tabulate" haben eine weitere Befehlsempfangsschleife die dann weitere Möglichkeiten der Bedienung erschließt. (Man siehe auch in Kapitel 7.1)

Der Monitor ist so das er keine einzige Speicherzelle im RAM benötigt, auch die FIFO Routine (Monitorlisting bei Zeilennummer 790) arbeitet nur mit Registern (HL= FIFO für vier Zeichen).

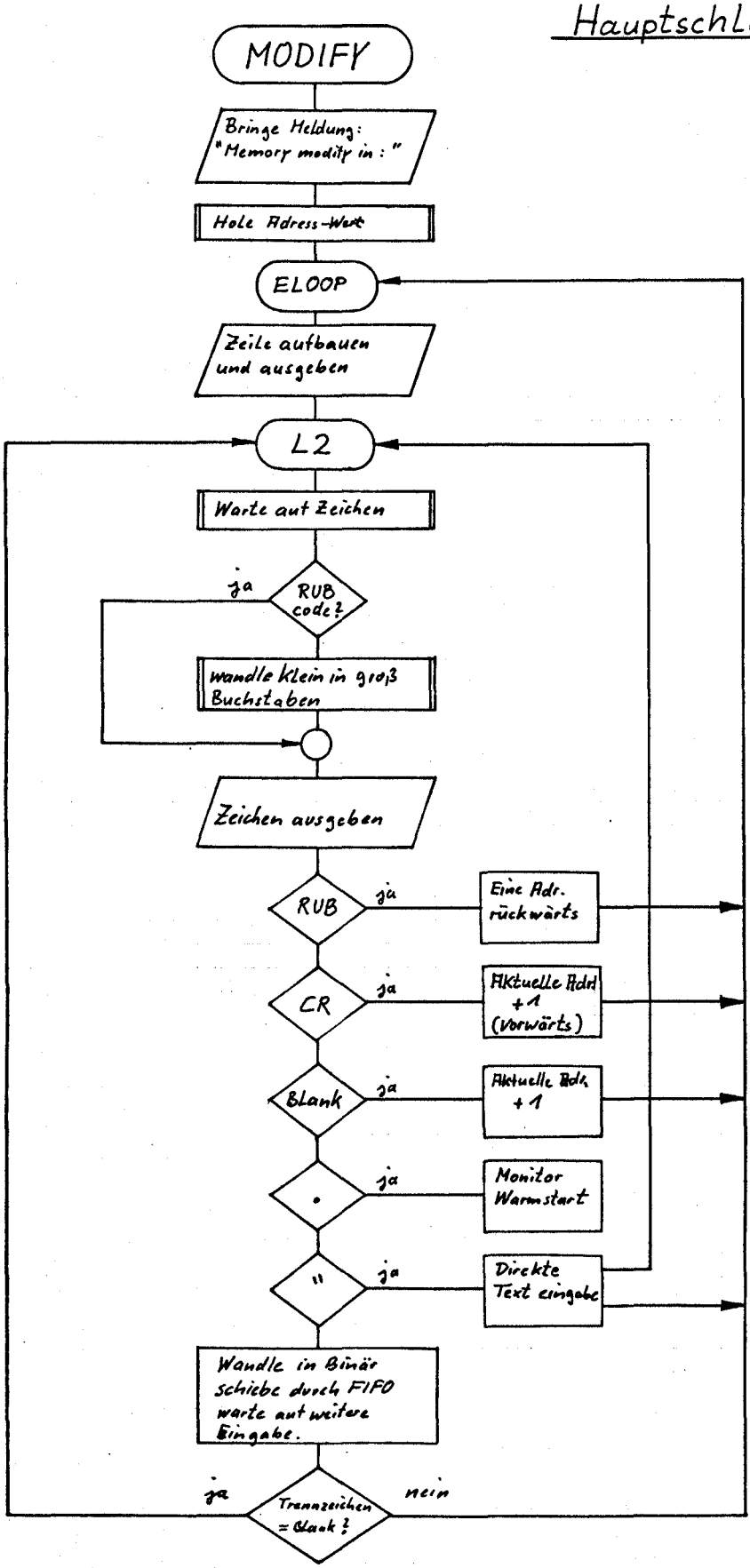
Die folgenden Ablaufdiagramme legen die Arbeitsweise des Monitors dar. Ein detail Einblick in die Funktion des Monitor bietet das kommentierte Listing auf Seite 161 .

4.2.3.1 Ablaufdiagramm "Monitor Hauptschleife"



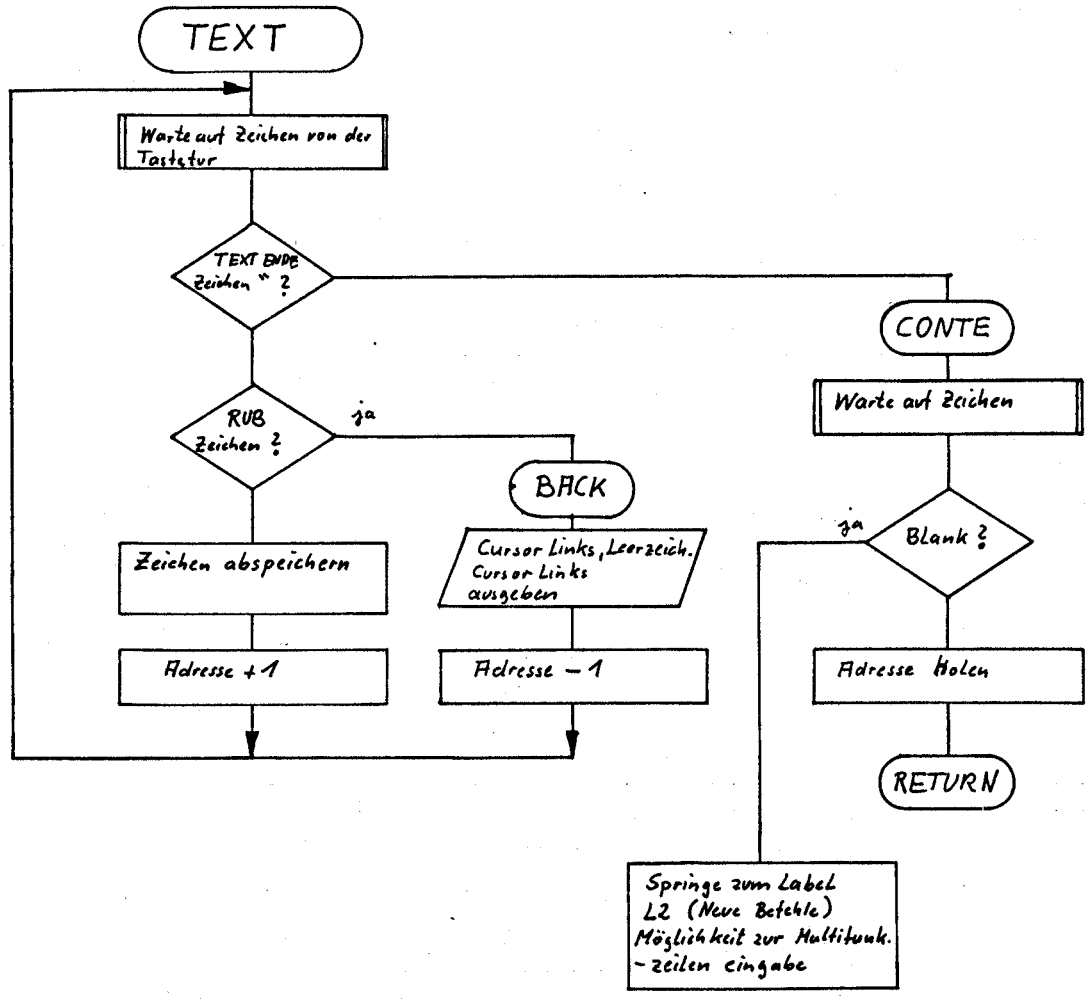
4.2.3.2 Ablaufdiagramm "MODIFY"

Hauptschleife

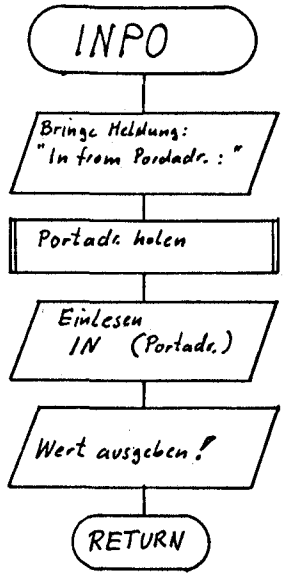


4.2.3.3 Ablaufdiagramm "MODIFY"

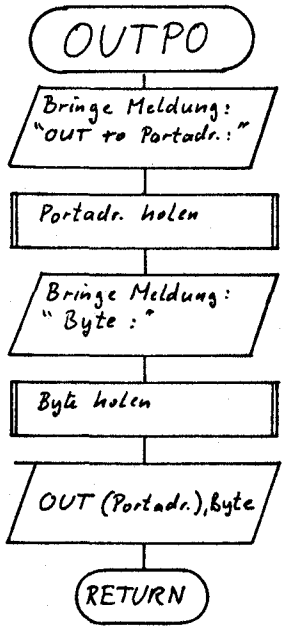
Texteingabe



4.2.3.4 Ablaufdiagramm "INPO"



4.2.3.5 Ablaufdiagramm "OUTPO"



4.3 Erläuterungen zum PAL- Betriebssystem

4.3.1 Bedienerebene

Der Teil Bedienerebene benutzt das "2.RAM" als Arbeitsspeicher, genauergesagt der Editor wird so angesteuert das seine Bufferbereiche in das "2.RAM" fallen. Dieser RAM Teil wird bei jeden Kaltstart mit Nullen vollgeschrieben. Die Steuerung des Systemeditors ist eine der zentralen Aufgaben des CCP (Consol Comand Prozessors), d.h. für den Anwender sind die Editierfunktionen immer aktiv. Es sind zwei Bufferbereiche für den Systemeditor vorgesehen, ein Buffer für die Befehlseingaben und einer für die Programmierinformationen. Das kleine Programm "CHBUFZ" steuert das wechseln zwischen den Befehls- und Textbuffer. Durch die zahlreichen Bearbeitungen von Tabellen werden die INDEX Register (IX &IY) innerhalb der Programme oft benutzt.

Nun einiges zu dem Aufbau der PAL- Listen. Für jeden PAL-Type werden drei Listen benötigt. Die Hauptliste (TABST siehe bei Zeilennummer 4710) ist in folgender Weise aufgebaut:

Kennzeichung des PAL- Types	z.B.	'16R4'
Anzahl der PIN's des PAL		Byte
Anzahl der Eingänge des PAL		Byte
Anzahl der Ausgänge des PAL		Byte
Adresszeiger zur PRL- Liste		Word
Adresszeiger zur IPL- Liste		Word

Die erste Eintragung eines Datenblockes wird mit dem jeweiligen Eingabestring verglichen. Ist der Eingabestring mit der Eintragung in der Liste gleich, so ist der PAL- Type gefunden. Die Startadresse dieses Datenblockes wird in der Speicherzelle "PALT" eingetragen. Die folgenden drei Tabelleneinträge (Anzahl der PIN's, der Eingänge, der Ausgänge) sind zu Zeit noch nicht benutzt. Sie sind dazu vorgesehen bei späteren Versionen mit direkter Eingabe der Gleichung eine Plausbilitätskontrolle durchzuführen.

Die weiteren zwei "Word's" sind Zeiger auf die PRL und IPL Liste dieses PAL - Types. Da die verschiedenen PAL's auch verschieden große Programmierfelder haben (vergleiche Matrix von 20 und 24 PIN PAL), werden hierfür entsprechende Listen benötigt, welche zum Ersten eine klare Kennzeichung der Größe der Programmiermatrix ermöglichen. Zum Zweiten ist zu berücksichtigen das selbst bei PAL's mit gleicher PIN- Anzahl teilweise unvollständige Programmier- felder vorliegen. Alle diese Punkte sind in diesen Tabellen verwirklicht. Auch der Aufbau der Listen ist interessant, man sehe sich dazu die Erläuterung zu der "MSKTS" Routine an.

```
4790  DEFM '16R4'
4800  DEFB 20,16,8
4810  DEFW P16R4,I16R4
4820  DEFM '16R6'
4830  DEFB 20,16,8
4840  DEFW P16R6,I16R6
4850  DEFM '16R8'
4860  DEFB 20,16,8
4870  DEFW P16R8,I16R8
4880  DEFM '20R4'
4890  DEFB 24,20,8
4900  DEFW 0,0
4910  DEFM '20R6'
4920  DEFB 24,20,8
4930  DEFW 0,0
4940  DEFM '20R8'
4950  DEFB 24,20,8
4960  DEFW 0,0
4970  DEFM '10H8'
4980  DEFB 20,10,8
4990  DEFW P10H8,I10H8
5000  DEFM '10L8'
5010  DEFB 20,10,8
5020  DEFW P10L8,I10L8
```

Bild 29
Prakticher Aufbau der PAL-Haupttabelle

```
5600 ;
5610 I16R4 EQU $
5620 I16R6 EQU $
5630 I16R8 EQU $
5640  DEFB #80,#9F
5650  DEFB END
5660 ;
5670 ;
5680 P10H8 EQU $
5690 P10L8 EQU $
5700  DEFB #80,#81,#88,#89,#90,#91,#98,#99
5710  DEFB #A0,#A1,#A8,#A9,#B0,#B1,#B8,#B9
5720  DEFB END
5730 ;
5740 I10H8 EQU $
5750 I10L8 EQU $
5760  DEFB #80,#85,#88,#89,#8C,#8D,#90,#91
5770  DEFB #94,#95,#98,#99,#9C,#9F
5780  DEFB END
```

Bild 30
Prakticher Aufbau der PRL & IPL Listen

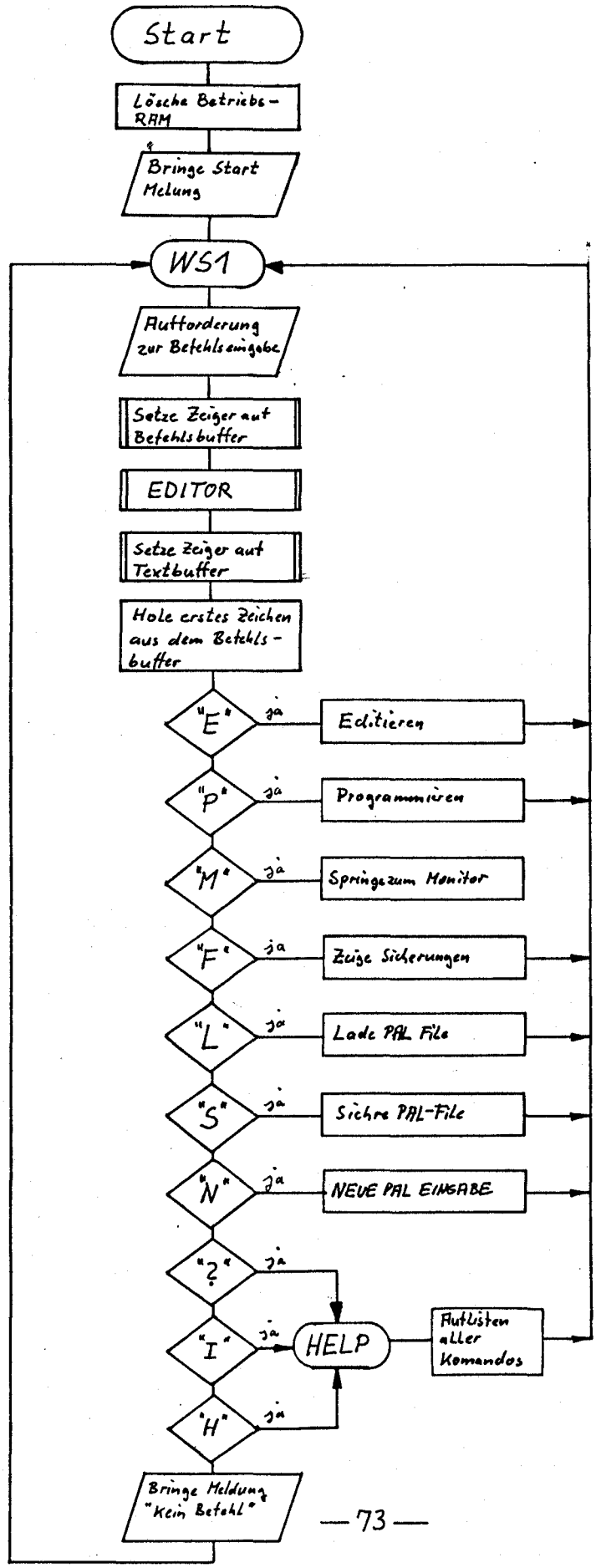
Was geschieht beim Aufruf von "NEW"? Nach der Eingabe des PAL- Types wird der Eingabestring in den Tabellenblöcken gesucht. Das Programm "SEARCH" ist speziell für diese Aufgabe geschrieben, kehrt dieses Programm mit gesetztem Carry zurück, so wurde die Eintragung nicht gefunden. Ist der Tabellendatensatz bestimmt, so wird der Programmteil "PALASM" ausgeführt. Es wird zunächst eine gültige PRL ermittelt. Ist diese gefunden werden über dem Editor die dazugehörigen Programmierinformationen abgefragt. Dieser Vorgang wiederholt sich so lange bis alle gültigen PRL's abgearbeitet sind.

Der Programmteil "FUSE" setzt zunächst seine Zeiger auf die von der Routine "MATRIX" (siehe bei 4.3.2.5) erzeugten Programmierdaten. Diese werden von dem Programm "FUSE" ausgewertet und in eine graphische Darstellung umgearbeitet. Für diese Umwandlung werden im wesentlichen drei Zeiger benötigt (einer auf Prog.-tabellenteil PRL, einer auf IPL, und einem voreilenden Testzeiger). Gesamtablauf dieses Programmes siehe in Ablaufdiagramm 4.3.1.2 .

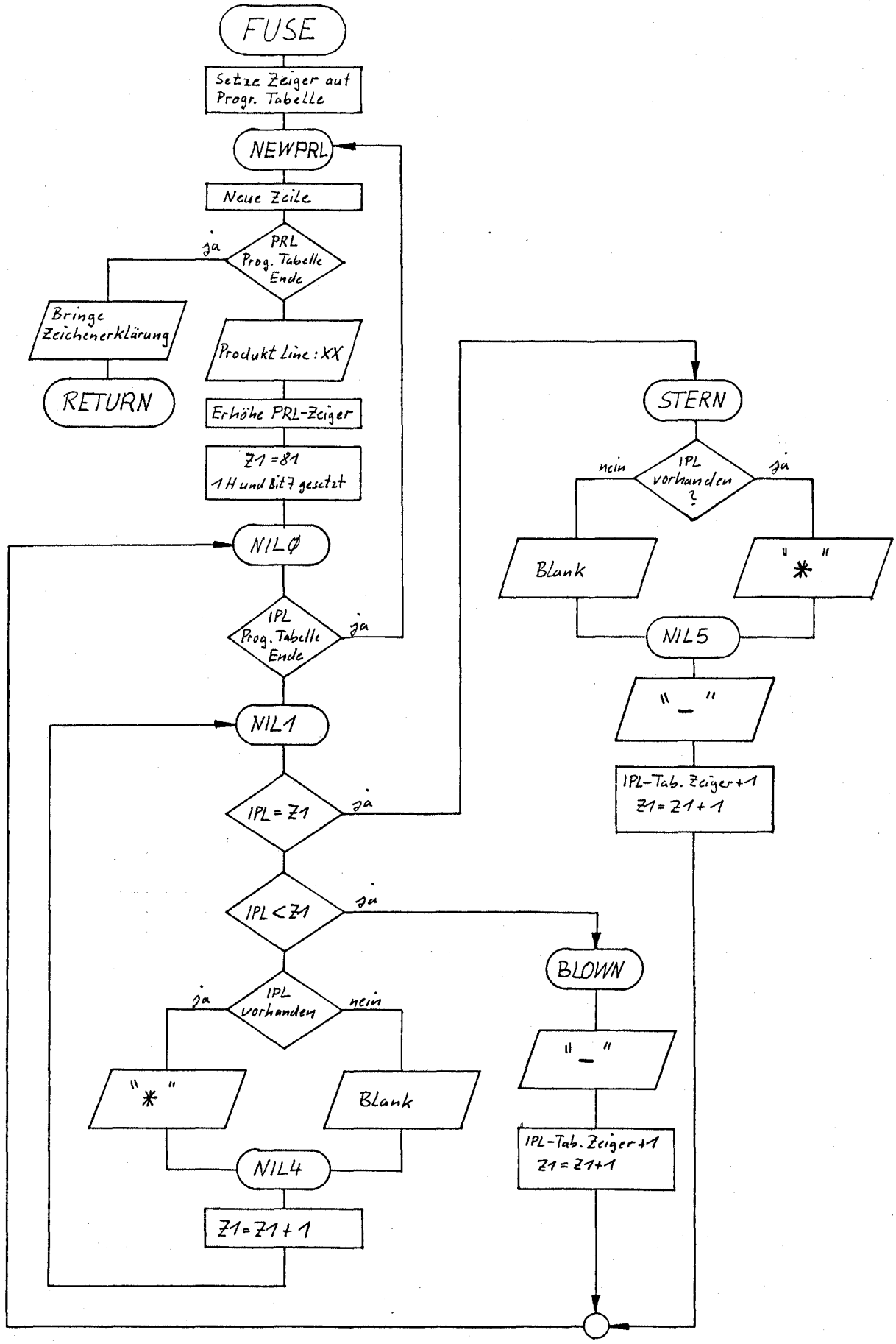
```
Produkt line :00  --¥---  --  --  --  --  --  ----
Produkt line :01  -----¥  --  ¥¥  ¥¥  --  --  ----
Produkt line :04  ------  -¥  --  --  --  --  ----
Produkt line :15  ------  --  --  --  --  --  ----
Produkt line :20  -¥-¥--  --  --  --  --  --  ----
Produkt line :21  ------  --  --  --  --  --  ----
Produkt line :30  ------  -¥  --  --  --  --  ----
Produkt line :38  -¥----  --  --  --  --  ¥-  ----

   - = Fuse blown / ¥ = Fuse not blown
```

Bild 31
Darstellung eines von "FUSE" erzeugten Bildes

4.3.1.1 Ablaufdiagramm "PAL-Betriebssystem"

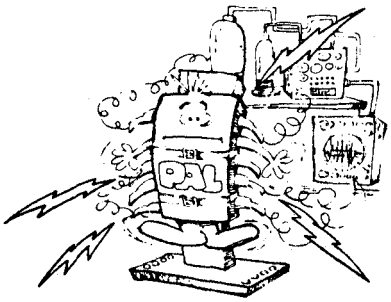


4.3.1.2 Ablaufdiagramm "FUSE"



4.3.2 Hardwareebene

4.3.2.1 Der Programmierzyklus



Folgende Grundlagen bilden die Ausgangsbasis zur Entwicklung eines Steuerungsprogrammes zur Realisierung des Programmierungstaktdiagrammes (BILD 11 Seite 16):

a) Zur Ansteuerung der linken Seite PAL müssen zwei Bits pro Pin ausgegeben werden

b) Zur Ansteuerung der rechten Seite PAL muß ein Bit pro Pin ausgegeben werden

c) Es kann jeweils nur eines dieser Bits pro Ausgabe an den Port gesetzt werden

d) Der Anfangszustand der Selektonsphase ist Log. HH, für die Pins 1 - 11 & 13 und Log Z,für die Pins 14 -23 (bei 20pol. PAL gilt die Pinbelegung entsprechend)

e) Strikte Einhaltung der Selektionsvorschriften

f) Im Übrigen wird auf die Ausführungen im Kapitel 3.1.1.1 hingewiesen

Die naheliegenste Lösung zur Herstellung der geforderten logischen Zustände ist die Ausgabe von Elementen einer Tabelle. Jedes Byte der Tabelle verändert den Zustand an einer der 40 vorhandenen Steuerleitungen, die ihrerseits die Transistortreiber schalten. Die Zusammenfassung der erlaubten Bytes geben die Tabellen 12 und 13 wieder.

Die Anwahl einer, durch Spalten- und Zeilennummer(IPL-Nr. und PRL-Nr.) definierten Matrixsicherung erfordert zunächst die Ausgabe von 6 Bytes über Port A (8255).

Im Einzelnen :

- Selektion einer IPL : 2 Bytes an linke Seite PAL
  +1 Byte zum L/R-select

- Selekton einer PRL : 3 Bytes an rechte Seite PAL
  (A0 - A2)

- Der Programmierimpuls übernimmt die letzte Phase der Selektion und bildet das siebente auszugebende Byte.

Aus den hexadezimalen Nummerierungen der anzuwählenden PRL's und IPL's kann man sofort die Informationen über das Ausgabebyte herleiten, das einen zulässigen Selektionspegel schaltet. Zur Anwahl einer IPL genügt es, einen der Pins 2 - 11 aus dem Anfangszustand heraus entweder L oder H zu schalten. (vergl. BILD 12) Dazu wird dann der hex-Wert der IPL vollständig auf den Zeiger aufaddiert, der auf den Anfang der IOI9- Tabelle zeigt.

```
                4950 ; Selecttabelle IOI9
OFD9 4129       4960 IOI9   DEFB #41,#29          ; 1.OG => L
OFDB 4121       4970        DEFB #41,#21        ;       => H
OFDD 422A       4980        DEFB #42,#2A        ; 2.OG => L
OFDF 4222       4990        DEFB #42,#22        ;       => H
OFE1 432B       5000        DEFB #43,#2B        ; 3.OG => L
OFE3 4323       5010        DEFB #43,#23        ;       => H
OFE5 442C       5020        DEFB #44,#2C        ; 4.OG => L
OFE7 4424       5030        DEFB #44,#24        ;       => H
OFE9 452D       5040        DEFB #45,#2D        ; 5.OG => L
OFEB 4525       5050        DEFB #45,#25        ;       => H
OFED 462E       5060        DEFB #46,#2E        ; 6.OG => L
OFEF 4626       5070        DEFB #46,#26        ;       => H
OFF1 472F       5080        DEFB #47,#2F        ; 7.OG => L
OFF3 4727       5090        DEFB #47,#27        ;       => H
OFF5 3018       5100        DEFB #30,#18        ; 8.OG => L
OFF7 3010       5110        DEFB #30,#10        ;       => H
```

TABELLE 2 : 2 Bytes selektieren eine Inputline.

Die Addition läßt den Zeiger auf eine Obergruppe zeigen, der dann durch ein Testen der Bits 0 und 1 im IPL-nummerbyte korrekt positioniert wird.

Der zugehörige Programmteil lautet :

```
                2640 ; Unterprogramm IXSEL: R A,BC,DE,HL
0E6A 21D90F     2650 IXSEL  LD   HL,IOI9
0E6D 0E10       2660        LD   C,#10    ; Portadresse
0E6F CD7C0E     2670        CALL HLPOS ; Zeiger HL positionieren
0E72 0602       2680        LD   B,#2     ; 2 Bytes ausgeben
0E74 EDA3       2690 LIXSEL OUTI ; einzelbyteausgabe
0E76 CD350E     2700        CALL STROBE
0E79 C8         2710        RET  Z
0E7A 18F8       2720        JR   LIXSEL
                2730 ;
0E7C 1600       2740 HLPOS  LD   D,00 ; Initialisieren Summand*
0E7E 3E03       2750        LD   A,#3       ; Ausblendmaske
0E80 FD5E00     2760        LD   E,(IY)     ;*register DE
0E83 CBBB       2770        RES  7,E
0E85 A3         2780        AND  E
0E86 19         2790        ADD  HL,DE     ; Zeiger in Obergruppe
0E87 FE03       2800        CP   #3         ; Testen bits 0 und 1
0E89 280C       2810        JR   Z,DECHL
0E8B FE02       2820        CP   #2
0E8D 2807       2830        JR   Z,DECHL2
0E8F FE01       2840        CP   #1
0E91 2801       2850        JR   Z,INCHL
0E93 C9         2860        RET  ; bei RET ist der
0E94 23         2870 INCHL  INC  HL
0E95 C9         2880        RET  ; Zeiger HL korrekt
0E96 2B         2890 DECHL2 DEC  HL
0E97 2B         2900 DECHL  DEC  HL
0E98 C9         2910        RET  ; positioniert
```

TABELLE 3 : Unterprogramm zur Selektion einer Inputline.

Es muß nun aber auch getestet werden, in welcher Hauptgruppe der PAL- Matrix programmiert werden soll. Dazu nimmt man das PRL-byte her, und testet auf größer, gleich oder kleiner als #1F, bei 20pol. PAL bzw #27, bei 24pol. PAL, die ja eine größere Matrix haben. Der Programmteil hierzu :

```
                1030 ; Hauptgruppenentscheidung
0D86 08         1040 TAKT   EX   AF,AF'          ; registertausch
0D87 DD7E00     1050        LD   A,(IX)          ; teste PRL-Nr.
0D8A FE9F       1060        CP   #9F             ; letzte PRL in 1.HG
0D8C 08         1070        EX   AF,AF'          ; C & Z - flag gerettet
```

TABELLE 4 : Dieser Programmteil setzt im Z 80- Zweitregister AF' das CARRY, das dann je nach Bedarf abgefragt werden kann.

Die Vorbereitung zur Hauptgruppenselektion wird von dem Zustand an Pin 1 und Pin 11 festgelegt. Erhält zB Pin 1 den Zustand HH, dann programmiert man in der ersten HG, und Pin 11 ist dann der CLOCK, der beim Verifizieren einen Ausgang aktiv schaltet. Analog dazu ist die Bedeutung der Pins 1 und 11 (20pol. PAL) vertauscht, wird in der zweiten HG programmiert. Das Schalten dieser Zustände ist der erste Schritt bei der Realisierung des Programmierungstaktdiagrammes.

Die zu auszugebenden Bytes für diesen Zyklus lauten :

1. Hauptgruppe : #48 , #20

2. Hauptgruppe : #3B , #13

Der Anfangzustand beim DESELECT wird für diese PAL-Pins = L gesetzt. Das Unterprogramm OD wie folgt :

```
                1740 ; Unterprogramm OD : RA,RB
0DF6 08         1750 OD      EX   AF,AF'
0DF7 3814       1760         JR   C,OD1HG ; Hauptgruppenentscheidung
0DF9 2812       1770         JR   Z,OD1HG
0DFB 08         1780 OD2HG   EX   AF,AF'
0DFC 3E3B       1790         LD   A,#3B ;
0DFE 47         1800         LD   B,A   ; 2 Steuercodes für VHH an
0DFF 3E13       1810         LD   A,#13 ; Pin 11 des 20-pol.PAL
0E01 D310       1820 ODHG    OUT  (#10),A
0E03 CD350E     1830         CALL STROBE
0E06 78         1840         LD   A,B
0E07 D310       1850         OUT  (#10),A ; senden
0E09 CD350E     1860         CALL STROBE
0E0C C9         1870         RET
0E0D 08         1880 OD1HG   EX   AF,AF'
0E0E 3E48       1890         LD   A,#48 ; 2 Steuercodes für Pin 1
0E10 47         1900         LD   B,A   ; laden
0E11 3E20       1910         LD   A,#20
0E13 18EC       1920         JR   ODHG
                1930 ;-----------------------------------------------
```

TABELLE 5 : Dieses Unterprogramm leitet den Selektionsprozess ein.

Der nächste Selektionsschritt ist die Entscheidung, ob L/R an Pin 12, oder Pin 19 nach HH geschaltet wird. An dieser Stelle muß darauf hingewiesen werden, daß das Programm PALPGM für das Programmieren von 20pol. PAL geschrieben wurde. Zum Unterprogramm LR : Ein Test von Bit 1 der IPL-Nr entscheidet über den zu schaltenen Pegel :

```
                2930 ; Unterprogramm LR:  RA, RB
0E99 FD7E00     2940 LR      LD   A,(IY)   ; IPL
0E9C 47         2950         LD   B,A
0E9D 08         2960         EX   AF,AF'     ; Hauptgruppentest
0E9E 380F       2970         JR   C,LRHG1    ; es werden die ent-
0EA0 280D       2980 LRHG2   JR   Z,LRHG1    ; sprechenden Steuer-
0EA2 08         2990         EX   AF,AF'     ;codes geladen und
0EA3 CB48       3000         BIT  1,B       ; mit SWLR ausgegeben
0EA5 2804       3010         JR   Z,LRZ2
0EA7 3E0B       3020 LRHH2   LD   A,#0B ; PIN 19 => HH
0EA9 180F       3030         JR   SWLR
0EAB 3E03       3040 LRZ2    LD   A,#03   ; PIN 19 => Z
0EAD 180B       3050         JR   SWLR
                3060 ;
0EAF 08         3070 LRHG1   EX   AF,AF'
0EB0 CB48       3080         BIT  1,B
0EB2 2804       3090         JR   Z,LRZ1
0EB4 3E3C       3100 LRHH1   LD   A,#3C ; PIN 12 => HH
0EB6 1802       3110         JR   SWLR
0EB8 3E34       3120 LRZ1    LD   A,#34   ; PIN 12 => Z
0EBA D310       3130 SWLR    OUT  (#10),A
0EBC CD350E     3140         CALL STROBE
0EBF C9         3150         RET
```

TABELLE 6 : Es wird der Zustand an Pin L/R geschaltet.

Die nächste Phase ist die Wahl einer aus acht Produktlines. Dazu wird eine 3 - Byte-tabelle berechnet auf folgender Grundlage :

Wahl einer PRL
eins aus acht

PRL - Byte	PRL- Nr.	Bytetabelle		
		A2	A1	A0
xxxxx000	0	35	36	37
xxxxx001	1	35	36	3F
xxxxx010	2	35	3E	37
xxxxx011	3	35	3E	3F
xxxxx100	4	3D	36	37
xxxxx101	5	3D	36	3F
xxxxx110	6	3D	3E	37
xxxxx111	7	3D	3E	3F

TABELLE 7 : Drei Steuerbytes wählen eine PRL. (vergl. PINTABELLE 13)

Die Aufgabe besteht nun darin, die Bits 0,1 und 2 des PRL-Bytes in das Datenbit des Steuerbytes zu übertragen, wobei der Anfangswert für die 1. HG = #35, und für die 2. HG = #00 ist.

```
                3170 ; Unterprogramm AXSEL: RA,RB,RC,RHL
0EC0 08         3180 AXSEL  EX   AF,AF'
0EC1 380F       3190        JR   C,AXHG1      ; Hauptgruppentest
0EC3 280D       3200 AXHG2  JR   Z,AXHG1
0EC5 08         3210        EX   AF,AF'       ;
0EC6 AF         3220        XOR  A            ; anfangswert für 2. HG
0EC7 CDD70E     3230 AXLOP  CALL ATAB ; 3 Steuerbytes errechnen
0ECA CD420E     3240        CALL OUT10      ; 3 errechnete Bytes
0ECD 0510       3250        DEFW TABA       ; ausgeben
0ECF 0310       3260        DEFB #03,#10
0ED1 C9         3270        RET
0ED2 08         3280 AXHG1  EX   AF,AF'
0ED3 3E35       3290        LD   A,#35        ; anfangswert für 1.HG
0ED5 18F0       3300        JR   AXLOP
                3310 ;
0ED7 210510     3320 ATAB   LD   HL,TABA ; Arbeitsbereich
0EDA 0603       3330        LD   B,3    ; drei bits testen und setzen
0EDC DD4E00     3340        LD   C,(IX)  ;testbyte PRL
0EDF CB51       3350 AXLOP1 BIT  2,C
0EE1 2804       3360        JR   Z,AXNUL
0EE3 CBDF       3370        SET  3,A       ; AX => HH
0EE5 1802       3380        JR   AXNXT      ; nächstes Byte errechnen
0EE7 CB9F       3390 AXNUL  RES  3,A ; AX => L
                3400 ;
0EE9 77         3410 AXNXT  LD   (HL),A ; erst einmal abspeichern
0EEA 23         3420        INC  HL           ; nächster freier Platz
0EEB CB01       3430        RLC  C            ; Testbyte bitposition 2
0EED 3C         3440        INC  A
0EEE 10EF       3450        DJNZ AXLOP1
0EF0 C9         3460        RET
```

TABELLE 8 : AXSEL errechnet drei Steuerbytes und gibt sie aus

Den Abschluß der Selektionsphase bildet das Auftreten des Programmierimpulses an den Pins 16 - 19 für die erste HG und an den Pins 12 - 15 für die zweite HG, in Abhängigkeit der Bits 3 und 4 des PRL - Bytes.

Die Steuerbytes zum Schalten des Programmierimpulses lauten für die erste Hauptgruppe :

```
Pin 19, 00 : #0B      HH
Pin 18, 01 : #0A
Pin 17, 02 : #09
Pin 16, 03 : #08
```

Und für die zweite Hauptgruppe :

Pin 15, 00 : #3F HH
Pin 14, 01 : #3E
Pin 13, 02 : #3D
Pin 12, 03 : #3C

Zur Berechnung des Programmimpulssteuerbytes wird das PRL -Byte dreimal rechtsgeshiftet (Testbit 3 und 4) und Bit 7 - 2 = 1 gesetzt. Nach dessen Komplementbildung wird es zum entsprechenden Anfangswert aufaddiert.

Das Unterprogramm PGM wie folgt :

```
                3480 ; Unterprogramm PGM: RA,RBC,RDE,RHL
0EF1 08         3490 PGM     EX   AF,AF'
0EF2 3807       3500         JR   C,PGM1HG   ; Hauptgruppentest
0EF4 2805       3510 PGM2HG  JR   Z,PGM1HG
0EF6 08         3520         EX   AF,AF'
0EF7 063C       3530         LD   B,#3C      ; Startwert, Bit 0 & 1 = 0
0EF9 1803       3540         JR   PGM0
0EFB 08         3550 PGM1HG  EX   AF,AF'
0EFC 0608       3560         LD   B,#08      ; Startwert, Bit 0 & 1 = 0
0EFE DD7E00     3570 PGM0    LD   A,(IX); Testbits 2 & 3 von PRL-Nr
0F01 2F         3580         CPL
0F02 1F         3590         RRA
0F03 1F         3600         RRA
0F04 1F         3610         RRA
0F05 E603       3620         AND  #03
0F07 80         3630         ADD  A,B   ; 0-selectbyte für HH
0F08 47         3640         LD   B,A    ; vorbereitendes Registerladen
0F09 CB9F       3650         RES  3,A
0F0B 4F         3660         LD   C,A
0F0C DB12       3670         IN   A,(#12)
0F0E E6FE       3680         AND  #FE
0F10 57         3690         LD   D,A
0F11 CBC7       3700         SET  0,A
0F13 5F         3710         LD   E,A
                3720 ;  ***  Start programzykus ***
0F14 78         3730         LD   A,B
0F15 D310       3740         OUT  (#10),A  ; sel.byte für HH
0F17 7A         3750         LD   A,D       ;-------------------
0F18 D302       3760         OUT  (#02),A  ; aktiver Strobe  I
0F1A 7B         3770         LD   A,E
0F1B D302       3780         OUT  (#02),A
0F1D 79         3790         LD   A,C
0F1E D310       3800         OUT  (#10),A  ; deselect => L
0F20 7A         3810         LD   A,D
0F21 D302       3820         OUT  (#02),A  ; aktiver Strobe  I
0F23 7B         3830         LD   A,E       ;-------------------
0F24 D302       3840         OUT  (#02),A
0F26 48         3850         LD   C,B  ; selectbyte retten
0F27 C9         3860         RET
```

TABELLE 9 : Das Steuerbyte wird ermittelt und in einem Programmierzyklus = 18 µs ausgegeben.

Zusammenfassung :

Die oben beschriebenen Teilprogramme realisieren nur den Selektions und Programmierprozeß, der für 20pol. PAL gilt. Die Hardware ist jedoch konzipiert um auch 24pol. oder PAL-bausteine anderer Hersteller zu programmieren. Das Programm PALPGM muß dazu erweitert werden. Ein wichtiger Teil des Programmierzyklusses ist der Verifizierungsprozeß, der anschließend beschrieben werden soll.

4.3.2.2 Der Verifizierungszyklus

Die Notwendigkeit einer gewissenhaften Verifizierung der gebrannten Diodensicherung wurde schon im Kapitel 3.3.2.3 Seite 39 behandelt.

Entscheident ist, daß an dem PAL-Pin verifiziert wird, der, durch einen CLOCK-impuls aktiviert, gerade einen Programmierpuls erhalten hat. Durch einen Einlesebefehl steht das Zustandsbyte im Akkumulator. (siehe auch Zeichnung Nr 5 und 2) Die Position des zu testenden Bits wird aus dem vorausgegangenen Programmierimpuls-steuerbyte hergeleitet. Dabei ergeben sich folgende Abbildungen :

Register :	D7	D6	D5	D4	D3	D2	D1	D0
	!	!	!	!	!	!	!	!
Steuerbyte:	0B	0A	09	08	3F	3E	3D	3C

Der Grundgedanke ist, das jeweils zu testende Bit in die Position 7 zu schieben. Es wird aber in einem Verifyzyklus mit zwei Spannungspegel gemessen, und man erhält dann auch zwei Werte. So wird das Meßergebnis aus 2,4V in Bit 7 und das aus 0,4V in Bit 6 abgelegt. Da der Ausgangspegel für aktiv L-PAL (0,8V und für aktiv H-PAL) 2 V ist, ist auch der Inhalt von Bit 6 und 7 entsprechend.

Die Zusammenhänge zeigen BILD 32 :

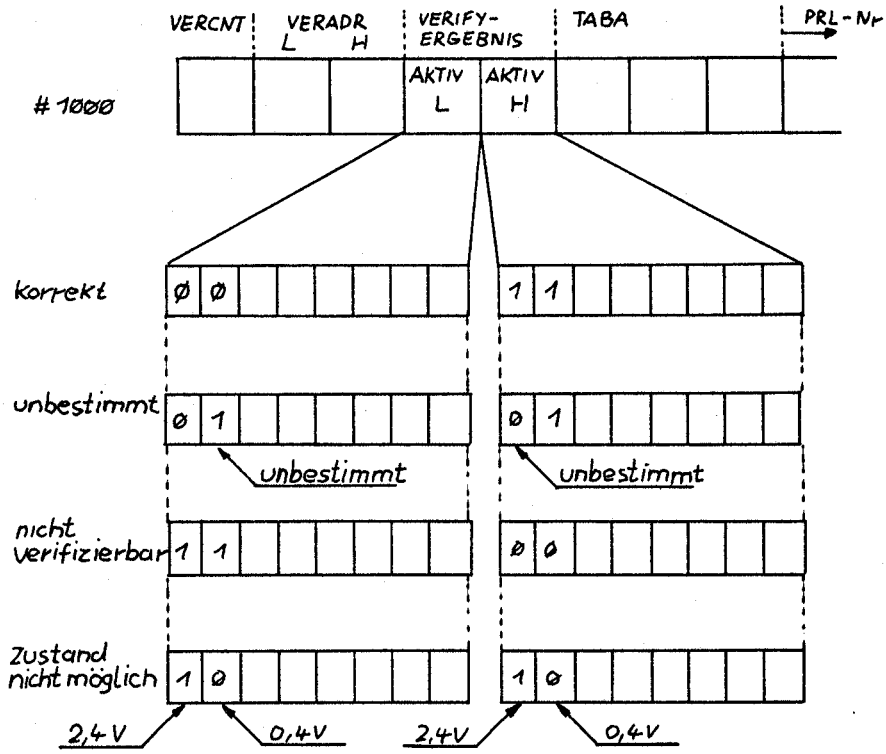


BILD 32 : Die Bedeutung des Ergebnisses aus dem Verify - zyklus im Arbeitsbereich von PALPGM (siehe auch Spannungs- Zeit -Diagramm BILD 28 seite 41)

Das Verifizierungsunterprogramm VERYFY setzt, abhängig vom PAL -typ die Bytes #1003 bzw #1004. Im Hauptprogramm wird dann e ntsprechend verzweigt. (siehe auch Ablaufdiagramm S 93)

Das Unterprogramm VERIFY hat folgenden Wortlaut :

```
                3880 ; Unterprogramm VERYFY
0F28 CD300F     3890 VERIFY CALL INPEG
0F2B 3E80       3900        LD   A,#80
0F2D D313       3910        OUT  (#13),A    ; Port B => ausgabe
0F2F C9         3920        RET
                3930 ;
0F30 1605       3940 INPEG  LD   D,#5; shiftzähler Startwert
0F32 CD760F     3950        CALL TEST   ;setztRE,benötigt RC(PGM-Byte)
0F35 3E06       3960        LD   A,#06     ;Vref = 2,4 V
0F37 D310       3970        OUT  (#10),A
0F39 CD350E     3980        CALL STROBE
0F3C CD8E0F     3990        CALL CLCK;schaltet den aktivierungsimpuls
0F3F 3E82       4000        LD   A,#82 ;   "eingabe"-code
0F41 D313       4010        OUT  (#13),A    ; Port B => eingang
0F43 DB11       4020        IN   A,(#11)
0F45 1002       4030 INPGL1 DJNZ INPGL2 ;TEST setzte den Zähler
0F47 1803       4040        JR   INPGL3
0F49 07         4050 INPGL2 RLCA ; positionierung auf Bit 7
0F4A 18F9       4060        JR   INPGL1
0F4C 5F         4070 INPGL3 LD   E,A  ; Bit 7 retten
                4080 ;
0F4D 1606       4090        LD   D,#6      ; Shiftzähler startwert
0F4F CD760F     4100        CALL TEST   ;RC, RE
0F52 3E0E       4110        LD   A,#0E     ; Vref = 0,4V
0F54 D310       4120        OUT  (#10),A
0F56 CD350E     4130        CALL STROBE
0F59 CD8E0F     4140        CALL CLCK ; aktivierungstakt
0F5C DB11       4150        IN   A,(#11)
0F5E 1002       4160 INPGL4 DJNZ INPGL5;TEST setzte den Zähler
0F60 1803       4170        JR   INPGL6
0F62 07         4180 INPGL5 RLCA ; Positionierung in CARRY
0F63 18F9       4190        JR   INPGL4
0F65 3805       4200 INPGL6 JR   C,SETB6
0F67 7B         4210 RESB6  LD   A,E   ; Bit 7 holen
0F68 CBB7       4220        RES  6,A       ; Verifizierungsergebnis
0F6A 1803       4230        JR   LDTBT      ; abhängig von akt L oder H
                4240 ;
0F6C 7B         4250 SETB6  LD   A,E  ; Bit 7 holen
0F6D CBF7       4260        SET  6,A       ; Verifizierungsergebnis
0F6F E6C0       4270 LDTBT  AND  #C0; abhängig von akt L oder H
0F71 210110     4280        LD   HL,VERADR
0F74 77         4290        LD   (HL),A
0F75 C9         4300        RET  ; vom Pegellesezyklus
                4310 ;                  mit Ergebnis in (VERADR)
```

```
                4320 ;
0F76 63         4330 TEST   LD   H,E     ;save
0F77 1E0F       4340        LD   E,#0F    ; ausblendmaske
0F79 3E3F       4350        LD   A,#3F    ; vergleichsstartwert
0F7B 0604       4360        LD   B,#4     ; schleifenzähler
0F7D B9         4370 TESTL  CP   C
0F7E 280B       4380        JR   Z,TESTNX; gefunden
0F80 3D         4390        DEC  A       ; nicht gefunden
0F81 14         4400        INC  D       ; nachher eins weiter shiften
0F82 10F9       4410        DJNZ TESTL
0F84 A3         4420        AND  E       ; weiter mit testwert #0B
0F85 0604       4430        LD   B,#4     ; wieder 4mal versuchen
0F87 CB9A       4440        RES  3,D     ; shifter = 1,-> nicht shiften
0F89 18F2       4450        JR   TESTL
0F8B 42         4460 TESTNX LD   B,D;Shiftzählerendwert=Schlei-
0F8C 5C         4470        LD   E,H     ; reload///fenzähler für DJNZ
0F8D C9         4480        RET  ; ***DJNZ steht vor shift****
                4490 ;
0F8E 08         4500 CLCK   EX   AF,AF'; Ausgabe der Tabelle zum
0F8F 3816       4510        JR   C,CLKHG1  ; schalten des aktivieruns-
0F91 2814       4520        JR   Z,CLKHG1  ; taktes in Abhängigkeit der
0F93 08         4530 CLKHG2 EX   AF,AF'; selektierten Hauptgruppe
0F94 1805       4540        JR   CKHG2
0F96 40282040   4550 CK2TAB DEFB #40,#28,#20,#40,#28
     28
0F9B C5         4560 CKHG2  PUSH BC
0F9C D5         4570        PUSH DE
0F9D CD420E     4580        CALL OUT10
0FA0 960F       4590        DEFW CK2TAB
0FA2 0510       4600        DEFB #05,#10
0FA4 D1         4610        POP  DE
0FA5 C1         4620        POP  BC
0FA6 C9         4630        RET
0FA7 08         4640 CLKHG1 EX   AF,AF'
0FA8 1805       4650        JR   CKHG1
0FAA 331B1333   4660 CK1TAB DEFB #33,#1B,#13,#33,#1B
     1B
0FAF C5         4670 CKHG1  PUSH BC
0FB0 D5         4680        PUSH DE
0FB1 CD420E     4690        CALL OUT10
0FB4 AA0F       4700        DEFW CK1TAB
0FB6 0510       4710        DEFB #05,#10
0FB8 D1         4720        POP  DE
0FB9 C1         4730        POP  BC
0FBA C9         4740        RET
```

TABELLE 10 : Dieses Unterprogramm testet den aktiven Ausgangspegel am programmierten PAL - PIN

Der Ablauf des Hauptprogrammes gestaltet sich sehr einfach und beinhaltet den Enscheidungszyklus im Anschluß an jeweils einen Verify mit Ucc = 6V und einen Verify mit Ucc = 4,5V. Er ist wiedergegeben durch das Ablaufdiagramm S 93 und die Taktdiagramme Nr.33 und Nr.34 seite 91 & 92

```
                0480 ;-----------------------------------------------------
                0490 ;       VORBEREITUNG
                0500 ;-----------------------------------------------------
0D20 DD2A0020   0510         LD   IX,(PALT); Wird akt.H oder akt.L prog?
0D24 DD7E02     0520         LD   A,(IX+2) ; H lesen aus (PALT)
0D27 FE48       0530         CP   #48   ; nur aktiv H-PAL entscheid
0D29 2805       0540         JR   Z,HTYP
0D2B 210310     0550         LD   HL,AKLTBT;Merkzellenadresse laden
0D2E 1803       0560         JR   LV
0D30 210410     0570 HTYP    LD   HL,AKHTBT;Merkzellenadresse laden
0D33 220110     0580 LV      LD   (VERADR),HL
                0590 ;
0D36 DD210810   0600         LD   IX,#1008  ;Start der PRL-Tabelle
0D3A FD214810   0610         LD   IY,#1048  ;Start der IPL-Tabelle
0D3E AF         0620         XOR  A
0D3F DD77F8     0630         LD   (IX-8),A  ; Zähler = 0
0D42 CD630E     0640         CALL INITA       ; Portinitialisierung
0D45 CD5A0E     0650         CALL REEX        ; Reset expander
0D48 CD150E     0660         CALL VOLT
0D4B 600007     0670         DEFB #60,#00,#07 ; alle spannungen aus.
0D4E C9         0680         RET  ; Systemmeldung : ***Vpp Ein***
                0690 ;-----------------------------------------------------
                0700 ;          START MIT Vpp EIN
                0710 ;-----------------------------------------------------
0D4F CD150E     0720         CALL VOLT        ; Versorgungsspannung EIN
0D52 500B06     0730         DEFB #50,#0B,#06 ; V1 = 5V
                0740 ;
                0750 ; Hauptprogrammschleife
0D55 CD420E     0760 HLOOP   CALL OUT10       ; Deselct all
0D58 BB0F       0770         DEFW DESTAB      ; OD/CLK=>L,I=>HH,
0D5A 1E10       0780         DEFB #1E,#10     ; O,L/R,A=>Z
0D5C CD150E     0790         CALL VOLT
0D5F 708002     0800         DEFB #70,#80,#02 ; V3 = 11,5V
0D62 CD150E     0810         CALL VOLT
0D65 60BB00     0820         DEFB #60,#BB,#00 ; V2 = 11,5V
                0830 ;
                0840 ; Programmendeabfrage
0D68 DD7E00     0850         LD   A,(IX)
0D6B FE00       0860         CP   #00;alle Prod.lines abgearbeitet?
0D6D 2810       0870         JR   Z,END; programmierung abgeschlossen
0D6F FD7E00     0880         LD   A,(IY)
0D72 FE00       0890         CP   #00;erste Prod.line abgearbeitet?
0D74 CA790D     0900         JP   Z,ZEIGER ; beide Zeiger neu setzen
0D77 180D       0910         JR   TAKT ; Schmelzen einer Sicherung
0D79 DD23       0920 ZEIGER  INC  IX ; Neuer PRL-Wert
0D7B FD23       0930         INC  IY ; Neuer IPL-Wert
0D7D 18D6       0940         JR   HLOOP
                0950 ;
0D7F CD150E     0960 END     CALL VOLT
0D82 600007     0970         DEFB #60,#00,#07 ; Alle Spannungen AUS
0D85 C9         0980         RET  ; Systemmeldung:*** Progr.beendet***
                0990 ;       Systemmeldung:***    FEHLER    ***
```

```
                1020 ;
                1030 ; Hauptgruppenentscheidung
0D86 08         1040 TAKT    EX    AF,AF'          ; registertausch
0D87 DD7E00     1050         LD    A,(IX)          ; teste PRL-Nr.
0D8A FE9F       1060         CP    #9F             ; letzte PRL in 1.HG
0D8C 08         1070         EX    AF,AF'          ; C & Z - flag gerettet
                1080 ;-----------------------------------------------
                1090 ; Start Taktdiagramm
                1100 ;-----------------------------------------------
0D8D CDF60D     1110         CALL OD             ; select Hauptgruppe
0D90 CD6A0E     1120         CALL IXSEL          ; select Inputline
0D93 CD990E     1130         CALL LR             ;   II        II
0D96 CDC00E     1140         CALL AXSEL          ; select Productline
0D99 CD150E     1150         CALL VOLT
0D9C 508000     1160         DEFB #50,#80,#00 ; V1 = 11,5V
0D9F CDF10E     1170         CALL PGM            ; Programmierpuls
0DA2 CD150E     1180         CALL VOLT
0DA5 500400     1190         DEFB #50,#04,#00 ; V1 = 6,5V
0DA8 CD280F     1200         CALL VERIFY         ;Kehrt mit 1003/1004
                1210 ;                    zurück
0DAB 210110     1220         LD    HL,VERADR
0DAE 7E         1230         LD    A,(HL)    ; verifyergebnis
0DAF CB45       1240         BIT   0,L      ; (L)steht der progr.Baustein
                1250 ;
0DB1 2806       1260         JR    Z,HPAL
0DB3 FE00       1270 LPAL    CP    #00 ;LPAL korrekt programmiert?
0DB5 2811       1280         JR    Z,VERFY2; Ja
0DB7 1804       1290         JR    VERSL1
0DB9 FEC0       1300 HPAL    CP    #C0 ;HPAL korrekt programmiert?
0DBB 280B       1310         JR    Z,VERFY2; Ja
0DBD 2B         1320 VERSL1  DEC   HL      ; nein, versuchszähler
0DBE 34         1330         INC   (HL)    ; erhöhen
0DBF 3E05       1340         LD    A,#5
0DC1 BE         1350         CP    (HL)      ; schon 5mal versucht?
0DC2 2091       1360         JR    NZ,HLOOP; neuer versuch an glei.Stelle
0DC4 CBFE       1370         SET   7,(HL) ; Fehterbit in 1000
0DC6 18B7       1380         JR    END
                1390 ;
```

```
0DC8 CD150E   1400 VERFY2 CALL VOLT
0DCB 500400   1410        DEFB #50,#04,#00 ; V1 = 4,5V
0DCE CD280F   1420        CALL VERIFY
              1430 ;
0DD1 210110   1440        LD   HL,VERADR
0DD4 7E       1450        LD   A,(HL)    ; verifyergebnis
0DD5 CB45     1460        BIT  0,L       ; (L)steht der progr.Baustein
              1470 ;
0DD7 2806     1480        JR   Z,HPAL2
0DD9 FE00     1490 LPAL2  CP   #00 ;LPAL korrekt programmiert?
0DDB 2812     1500        JR   Z,NEWIPL; Ja
0DDD 1804     1510        JR   VERSL2
0DDF FEC0     1520 HPAL2  CP   #C0 ;HPAL korrekt programmiert?
0DE1 280C     1530        JR   Z,NEWIPL; Ja
0DE3 2B       1540 VERSL2 DEC  HL      ; nein, versuchszähler
0DE4 34       1550        INC  (HL)    ; erhöhen
0DE5 3E05     1560        LD   A,#5
0DE7 BE       1570        CP   (HL)      ; schon 5mal versucht?
0DE8 C2550D   1580        JP   NZ,HLOOP; neuer versuch an glei.Stelle
0DEB CBFE     1590        SET  7,(HL) ; Fehlerbit in 1000
0DED 1890     1600        JR   END
              1610 ;
0DEF AF       1620 NEWIPL XOR  A
0DF0 77       1630        LD   (HL),A
0DF1 FD23     1640        INC  IY
0DF3 C3550D   1650        JP   HLOOP
              1660 ;-----------------------------------------------------
              1670 ;
              1680 ;
              1690 ;
              1700 ;
              1710 ;-----------------------------------------------------
```

TABELLE 11 : Das Hauptprogramm PALPGM

Tabelle 12 Pin schalten für PAL-Pin Nummern 1-13 (1-11)

Pin Nr.	Name	Zustand	Pegel/Leitung		Pegel/Leitung		Byte(PA)	Byte(PA)	Hex.	Hex.
1	CLK/OD	L	H		H		x0101000	x1000000	28	40
		H	L	C0	H	A0	x0100000	x1000000	20	40
		HH	L		L		x0100000	x1001000	20	48
		*CANCEL	H		L		x0101000	x1001000	28	48
2	I0	L	H		H		x0101001	x1000001	29	41
		H	L	C1	H	A1	x0100001	x1000001	21	41
		HH	L		L		x0100001	x1001001	21	49
		*CANCEL	H		L		x0101001	x1001001	29	49
3	I1	L	H		H		x0101010	x1000010	2A	42
		H	L	C2	H	A2	x0100010	x1000010	22	42
		HH	L		L		x0100010	x1001010	22	4A
		*CANCEL	H		L		x0101010	x1001010	2A	4A
4	I2	L	H		H		x0101011	x1000011	2B	43
		H	L	C3	H	A3	x0100011	x1000011	23	43
		HH	L		L		x0100011	x1001011	23	4B
		*CANCEL	H		L		x0101011	x1001011	2B	4B
5	I3	L	H		H		x0101100	x1000100	2C	44
		H	L	C4	H	A4	x0100100	x1000100	24	44
		HH	L		L		x0100100	x1001100	24	4C
		*CANCEL	H		L		x0101100	x1001100	2C	4C
6	I4	L	H		H		x0101101	x1000101	2D	45
		H	L	C5	H	A5	x0100101	x1000101	25	45
		HH	L		L		x0100101	x1001101	25	4D
		*CANCEL	H		L		x0101101	x1001101	2D	4D
7	I5	L	H		H		x0101110	x1000110	2E	46
		H	L	C6	H	A6	x0100110	x1000110	26	46
		HH	L		L		x0100110	x1001110	26	4E
		*CANCEL	H		L		x0101110	x1001110	2E	4E
8	I6	L	H		H		x0101111	x1000111	2F	47
		H	L	C7	H	A7	x0100111	x1000111	27	47
		HH	L		L		x0100111	x1001111	27	4F
		*CANCEL	H		L		x0101111	x1001111	2F	4F
9	I7	L	H		H		x0011000	x0110000	18	30
		H	L	D0	H	B0	x0010000	x0110000	10	30
		HH	L		L		x0010000	x0111000	10	38
		*CANCEL	H		L		x0011000	x0111000	18	38
10	I8	L	H		H		x0011001	x0110001	19	31
		H	L	D1	H	B1	x0010001	x0110001	11	31
		HH	L		L		x0010001	x0111001	11	39
		*CANCEL	H		L		x0011001	x0111001	19	39
11	I9	L	H		H		x0011010	x0110010	1A	32
		H	L	D2	H	B2	x0010010	x0110010	12	32
		HH	L		L		x0010010	x0111010	12	3A
		*CANCEL	H		L		x0011010	x0111010	1A	3A
13	OD/CLK	L	H		H		x0011011	x0110011	1B	33
		H	L	D3	H	B3	x0010011	x0110011	13	33
		HH	L		L		x0010011	x0111011	13	3B
		*CANCEL	H		L		x0011011	x0111011	1B	3B

Tabelle 13 Pin schalten für PAL-Pin Nummern 14-23 (12-19)

Pin Nr.	Name	Zustand	Peg/Leit		Peg/Leit		Peg/Leit		Byte(PA)	Byte(PC)	Byte(PA)*	Hex.	Hex.	Hex.
14	L/R	L	L		H		H	PB0	x0110100	xxxxx111	00000001	34	07	01
		HH	H	B4	L	PC1	/		x0111100	xxxxx101	/	36	05	/
	O4	Z	L		L		/		x0110100	xxxxx101	/	34	05	/
15	A0	L	L		H		H	PB1	x0110101	xxxxx111	00000010	35	07	02
		HH	H	B5	L	PC1	/		x0111101	xxxxx101	/	3D	05	/
	O3	Z	L		L		/		x0110101	xxxxx101	/	35	05	/
16	A1	L	L		H		H	PB2	x0110110	xxxxx111	00000100	36	07	04
		HH	H	B6	L	PC1	/		x0111110	xxxxx101	/	3E	05	/
	O2	Z	L		L		/		x0110110	xxxxx101	/	36	05	/
17	A2	L	L		H		H	PB3	x0110111	xxxxx111	00001000	37	07	08
		HH	H	B7	L	PC1	/		x0111111	xxxxx101	/	3F	05	/
	O1	Z	L		L		/		x0110111	xxxxx101	/	37	05	/
18	NC	L	L		H		H	PB4	x0000000	xxxxx111	00010000	00	07	10
		HH	H	E0	L	PC1	/		x0001000	xxxxx101	/	08	05	/
	O0	Z	L		L		/		x0000000	xxxxx101	/	00	05	/
19	O4	L	L		H		H	PB5	x0000001	xxxxx111	00100000	01	07	20
		HH	H	E1	L	PC1	/		x0001001	xxxxx101	/	09	05	/
	NC	Z	L		L		/		x0000001	xxxxx101	/	01	05	/
20	O3	L	L		H		H	PB6	x0000010	xxxxx111	01000000	02	07	40
		HH	H	E2	L	PC1	/		x0001010	xxxxx101	/	0A	05	/
	A2	Z	L		L		/		x0000010	xxxxx101	/	02	05	/
21	O2	L	L		H		H	PB7	x0000011	xxxxx111	10000000	03	07	80
		HH	H	E3	L	PC1	/		x0001011	xxxxx101	/	0B	05	/
	A1	Z	L		L		/		x0000011	xxxxx101	/	03	05	/
22	O1	L	L		H		H	PA0	x0000100	xxxxx111	xxxxxx01	04	07	01
		HH	H	E4	L	PC1	/		x0001100	xxxxx101	/	0C	05	/
	A0	Z	L		L		/		x0000100	xxxxx101	/	04	05	/
23	O0	L	L		H		H	PA1	x0000101	xxxxx111	xxxxxx10	05	07	02
		HH	H	E5	L	PC1	/		x0001101	xxxxx101	/	0D	05	/
	L/R	Z	L		L		/		x0000101	xxxxx101	/	05	05	/

Die Leitung E6 schaltet Uref. Uref = 2,4V : Byte PA = ‡06. Uref = 0,4V : Byte PA = ‡0E

* Das "L"-Schalten dieser Pinreihe ist option.Die Pin's "L/R (O4)" und "A0 (O3)" werden von Port A bedient.

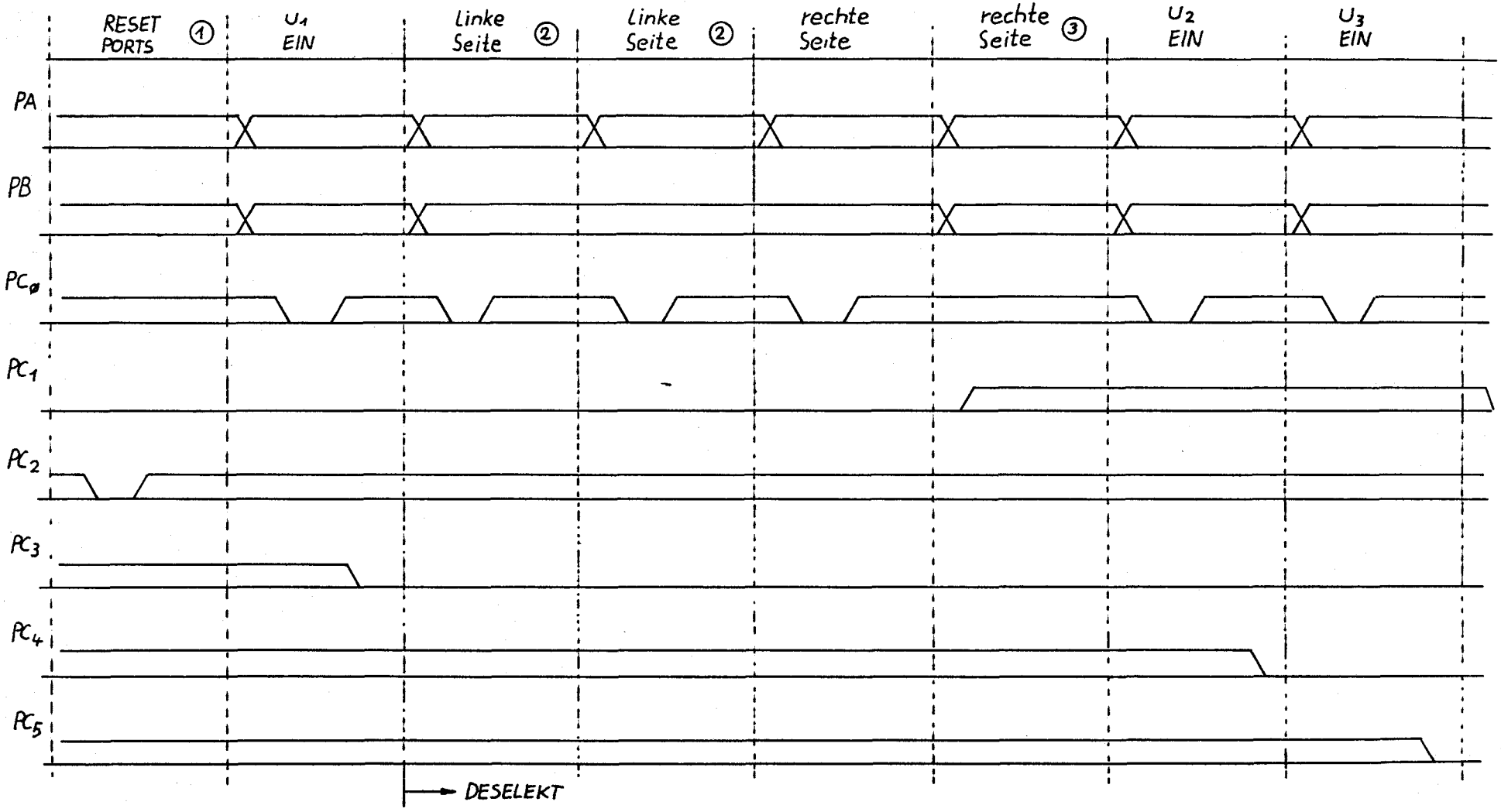


BILD 33: TAKTDIAGRAMM ZUM SETZEN DER ANFANGSZUSTÄNDE AN DEN PAL-PINS

① DIE 74 LS 259 WERDEN LOG Ø GESETZT, DIE NE 590 LOG 1
② SCHALTEN DER ZUSTÄNDE: L, H, HH
③ SCHALTEN DES ZUSTANDES: L ALS OPTION

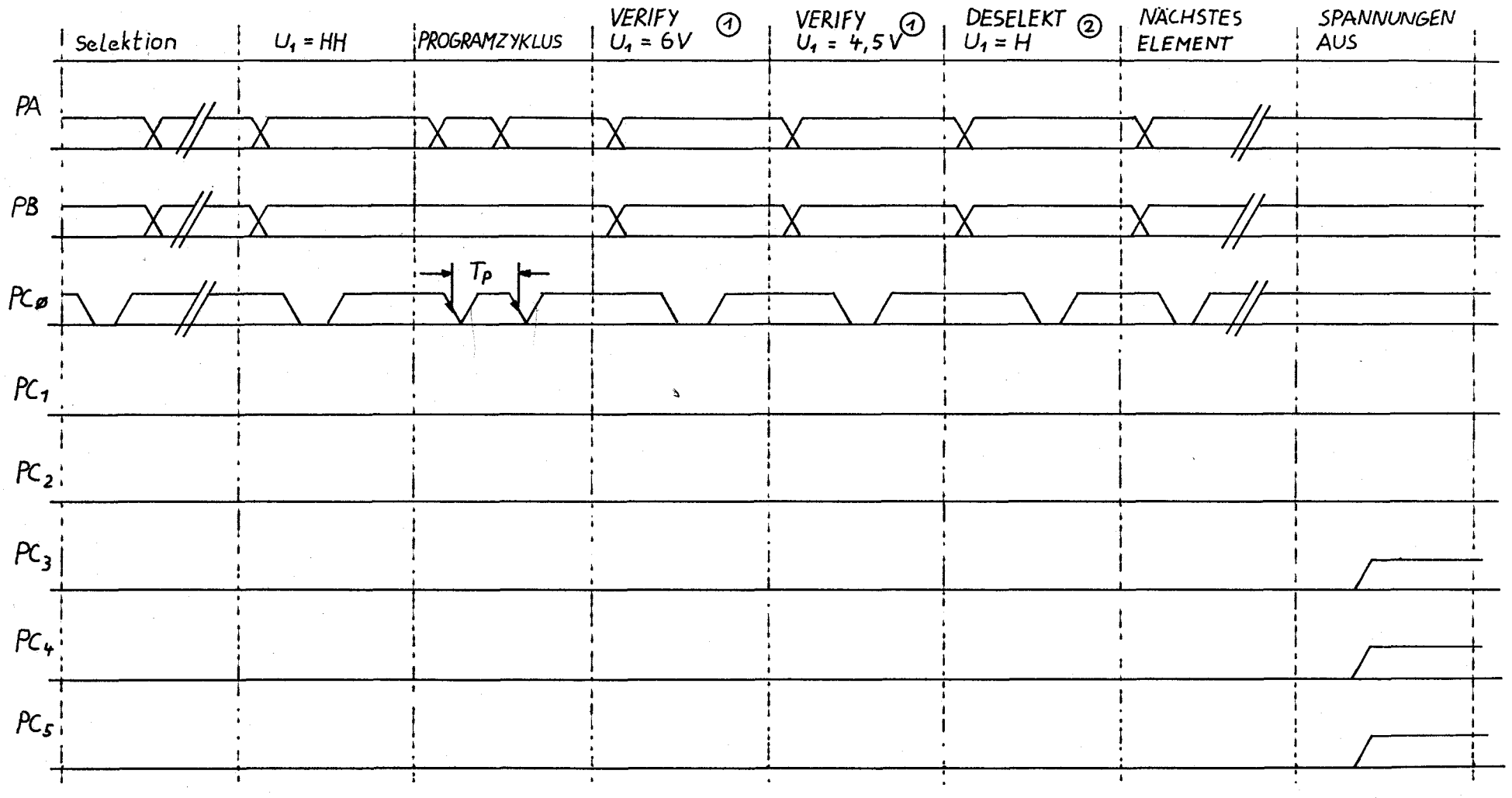
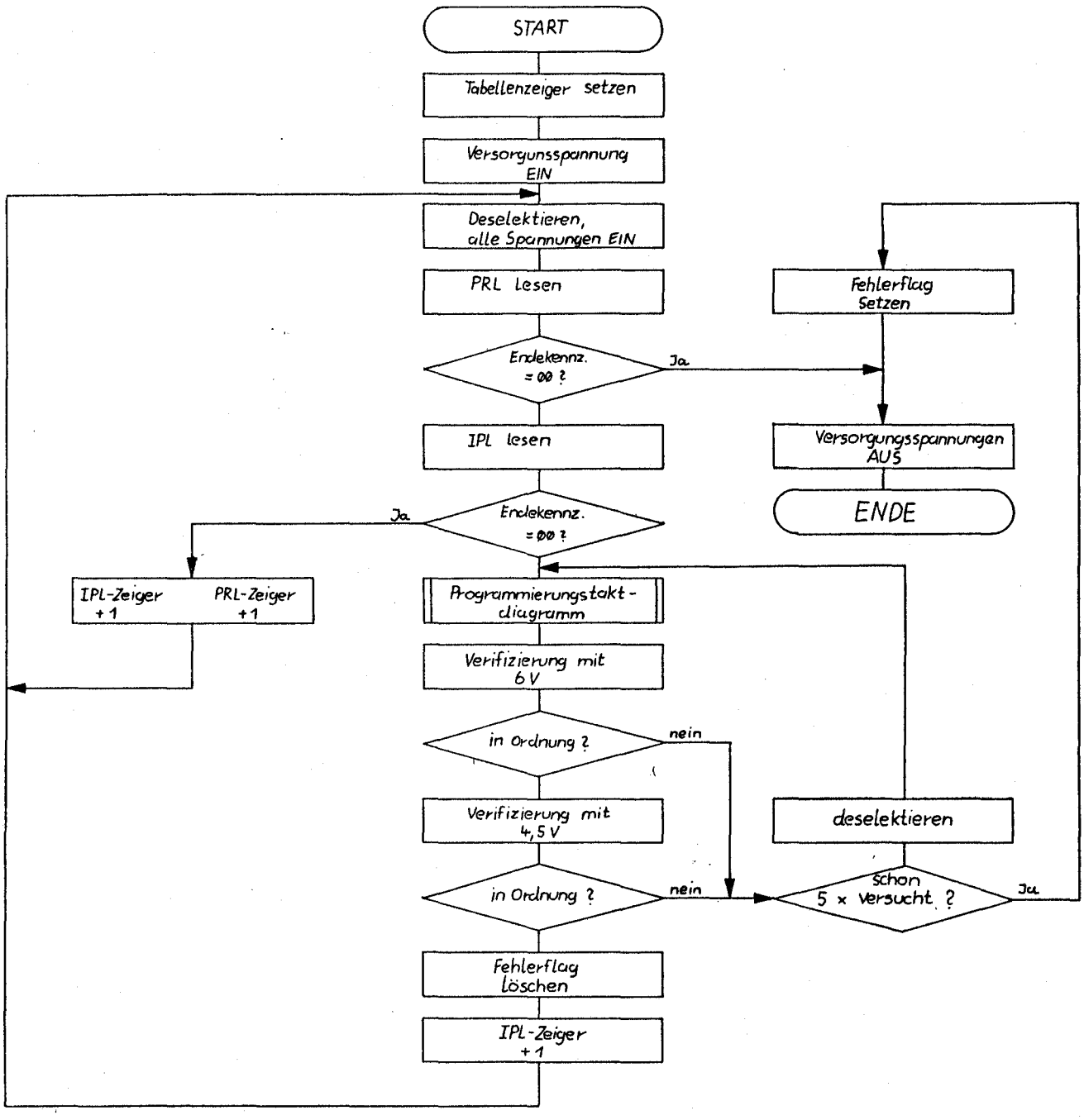


BILD 34: TAKTDIAGRAMM FÜR DIE HAUPTPROGRAMMSCHLEIFE

① WENN PCø = H, DANN PORT B UND Aø A1 AUF LESEN SCHALTEN
② SIEHE TAKTDIAGRAMM BILD

ABLAUFDIAGRAMM — PALPGM —



4.3.2.4. Alternatives Steuerungskonzept

Es existiert eine Tabelle konstanter Größe mit folgendem Aufbau :

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
PRL 0 :	xxxxxxxx	xxxxxxxx	xxxxxxxx	xxxxxxxx	xxxxxxxx

usw. , für alle Produktlines. Der Grundgedanke ist, daß ein Bit in dieser Tabelle über Zweierlei informiert :

- sein Zustand, über programmieren oder nicht
- seine Position, über die anzuwählende Diodensicherung.

Der Nachteil dieses Konzepts, das übrigens von allen Anderen angewendet wird ist, daß man immer vollständige Angaben machen muß, wenn man zB. nur eine Sicherung trennen will. Die Größe dieser Tabelle ist dann auch identisch mit der "Speicherkapazität" des betreffenden PAL. Also 256 Byte (64x4) beim 20pol. PAL und 400 Byte (80x5) beim 24pol. PAL. BILD 35 stellt ein Eingabeformblatt für eine solche Tabelle dar.

Nachdem aus dieser Tabelle noch kein PAL-programmiergerät gesteuert werden kann, muß ein Algorithmus diese Informationen erst einmal umrechnen um die Steuerbytes ausgeben zu können.

Dieses Konzept ist nur dann kein Umweg, wenn die Programmierinformation die bool'schen Gleichungen sind, und man eine Vereinheitlichung erreichen möchte.

Programming Format **PAL16R4**

PATTERN: _____ NAME: _____

INPUTS (0-31)

PRODUCT TERMS (0-63)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
WORD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
0_4 24																																
0_3 16																																
0_2 8																																
0_1 0																																
WORD	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
0_4 25																																
0_3 17																																
0_2 9																																
0_1 1																																
WORD	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
0_4 26																																
0_3 18																																
0_2 10																																
0_1 2																																
WORD	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
0_4 27																																
0_3 19																																
0_2 11																																
0_1 3																																
WORD	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
0_4 28																																
0_3 20																																
0_2 12																																
0_1 4																																
WORD	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AC	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
0_4 29																																
0_3 21																																
0_2 13																																
0_1 5																																
WORD	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
0_4 30																																
0_3 22																																
0_2 14																																
0_1 6																																
WORD	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF
0_4 31																																
0_3 23																																
0_2 15																																
0_1 7																																
WORD	100	101	102	103	104	105	106	107	108	109	10A	10B	10C	10D	10E	10F	110	111	112	113	114	115	116	117	118	119	11A	11B	11C	11D	11E	11F
0_4 56																																
0_3 48																																
0_2 40																																
0_1 32																																
WORD	120	121	122	123	124	125	126	127	128	129	12A	12B	12C	12D	12E	12F	130	131	132	133	134	135	136	137	138	139	13A	13B	13C	13D	13E	13F
0_4 57																																
0_3 49																																
0_2 41																																
0_1 33																																
WORD	140	141	142	143	144	145	146	147	148	149	14A	14B	14C	14D	14E	14F	150	151	152	153	154	155	156	157	158	159	15A	15B	15C	15D	15E	15F
0_4 58																																
0_3 50																																
0_2 42																																
0_1 34																																
WORD	160	161	162	163	164	165	166	167	168	169	16A	16B	16C	16D	16E	16F	170	171	172	173	174	175	176	177	178	179	17A	17B	17C	17D	17E	17F
0_4 59																																
0_3 51																																
0_2 43																																
0_1 35																																
WORD	180	181	182	183	184	185	186	187	188	189	18A	18B	18C	18D	18E	18F	190	191	192	193	194	195	196	197	198	199	19A	19B	19C	19D	19E	19F
0_4 60																																
0_3 52																																
0_2 44																																
0_1 36																																
WORD	1A0	1A1	1A2	1A3	1A4	1A5	1A6	1A7	1A8	1A9	1AA	1AB	1AC	1AD	1AE	1AF	1B0	1B1	1B2	1B3	1B4	1B5	1B6	1B7	1B8	1B9	1BA	1BB	1BC	1BD	1BE	1BF
0_4 61																																
0_3 53																																
0_2 45																																
0_1 37																																
WORD	1C0	1C1	1C2	1C3	1C4	1C5	1C6	1C7	1C8	1C9	1CA	1CB	1CC	1CD	1CE	1CF	1D0	1D1	1D2	1D3	1D4	1D5	1D6	1D7	1D8	1D9	1DA	1DB	1DC	1DD	1DE	1DF
0_4 62																																
0_3 54																																
0_2 46																																
0_1 38																																
WORD	1E0	1E1	1E2	1E3	1E4	1E5	1E6	1E7	1E8	1E9	1EA	1EB	1EC	1ED	1EE	1EF	1F0	1F1	1F2	1F3	1F4	1F5	1F6	1F7	1F8	1F9	1FA	1FB	1FC	1FD	1FE	1FF
0_4 63																																
0_3 55																																
0_2 47																																
0_1 39																																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

DATE: _____

BILD 35: Die Programmierinformationen bei anderen PAL-programmiergeräteherstellern

4.3.2.5 Das Programm MATRIX

Das Programm PALPGM erwartet an der Position seiner Zeiger IX und IY aktuelle Hex-werte der Input- und Produktlinenummern. Das Programm MATRIX berechnet zwei Tabellen, und zwar eine PRL-Nr.-tabelle die bei Speicherplatz $1008 beginnt, und eine IPL-Nr.-tabelle, die bei Speicherplatz $1048 beginnt. Beide Tabellen werden mit dem Endekennzeichen 00 abgeschlossen.

Arbeitsweise von MATRIX :

a) Erzeugen einer gültigen PRL-Nr. durch Maskieren mit einer vorhandenen Konstantentabelle (benutzung der Routine MASKS)

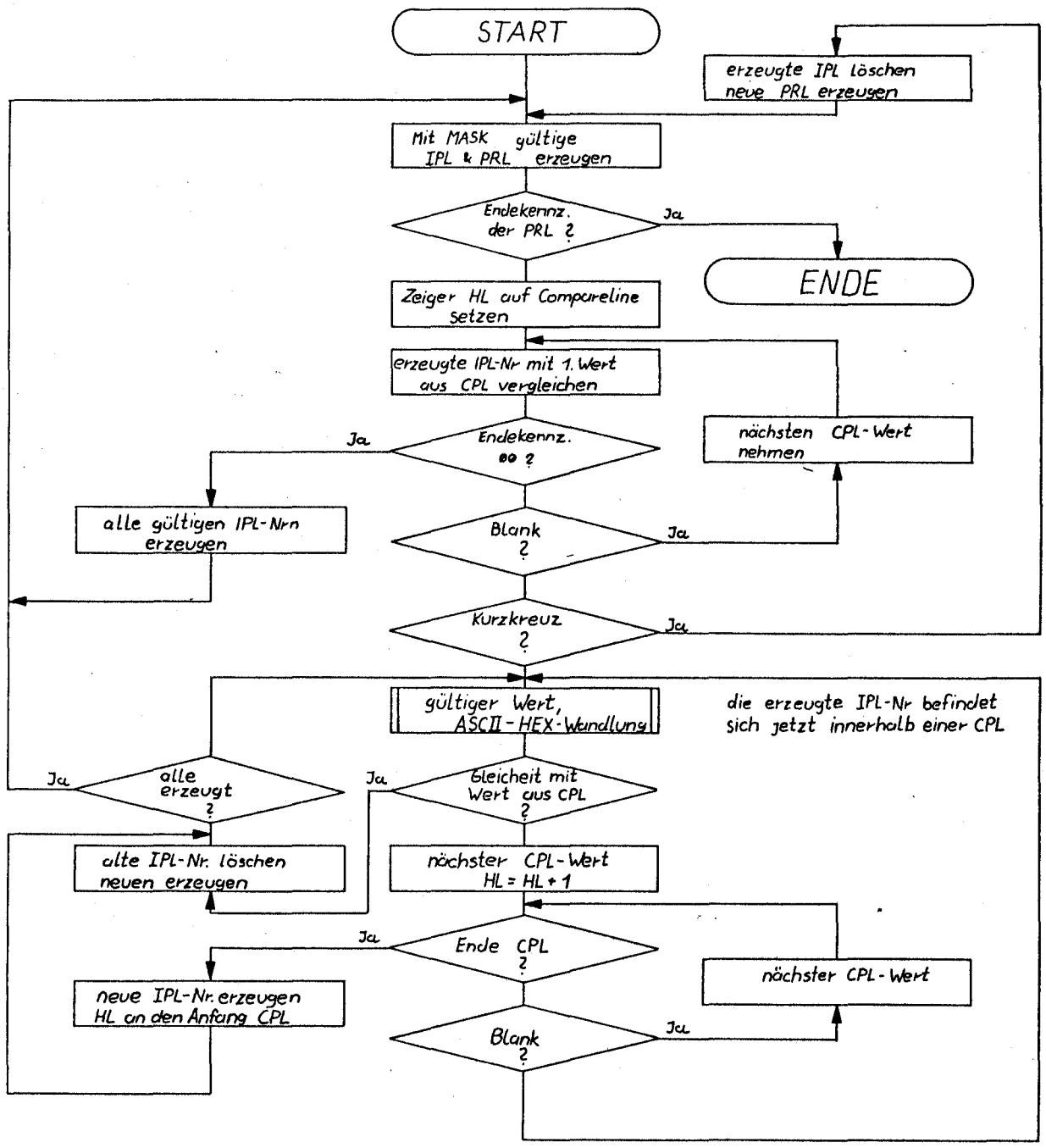
b) Erzeugen einer gültigen IPL-Nr. wie a)

c) Abfragen der Variablentabelle (Programmierinformation) ob die erzeugten Werte vorhanden sind

d) Wenn ja, dann überschreiben der alten Nr. mit einem, um 1 erhöhten (und durch MASKS auf Gültigkeit getesteten) Wert. Wenn nein, dann gilt der gleiche Prozess nur mit incrementiertem Tabellenzeiger.

Zusätzlich muß der, aus ASCII-werten bestehende Variablenstring (bei $2020) noch auf Endekennzeichen = 00 , gültigen Nullen (die 00 unmittelbar nach dem Endekennzeichen bedeutet "Alle Sicherungen brennen" , gegeben durch CR am Terminal) , Blanks und "Kurzkreuz" = $2A untersucht werden. Die detaillierte Arbeitsweise entnehme man dem Ablaufdiagramm S 97 und dem Programmlisting MATRIX im Anhang.

ABLAUFDIAGRAMM —MATRIX—



4.4 Speicherbelegungsplan

Durch die Gegebenheiten unserer CPU - Karte sind die Adressbereiche zerstückelt. Inklusive unseren Änderungen ergibt sich folgende Speicheraufteilung.

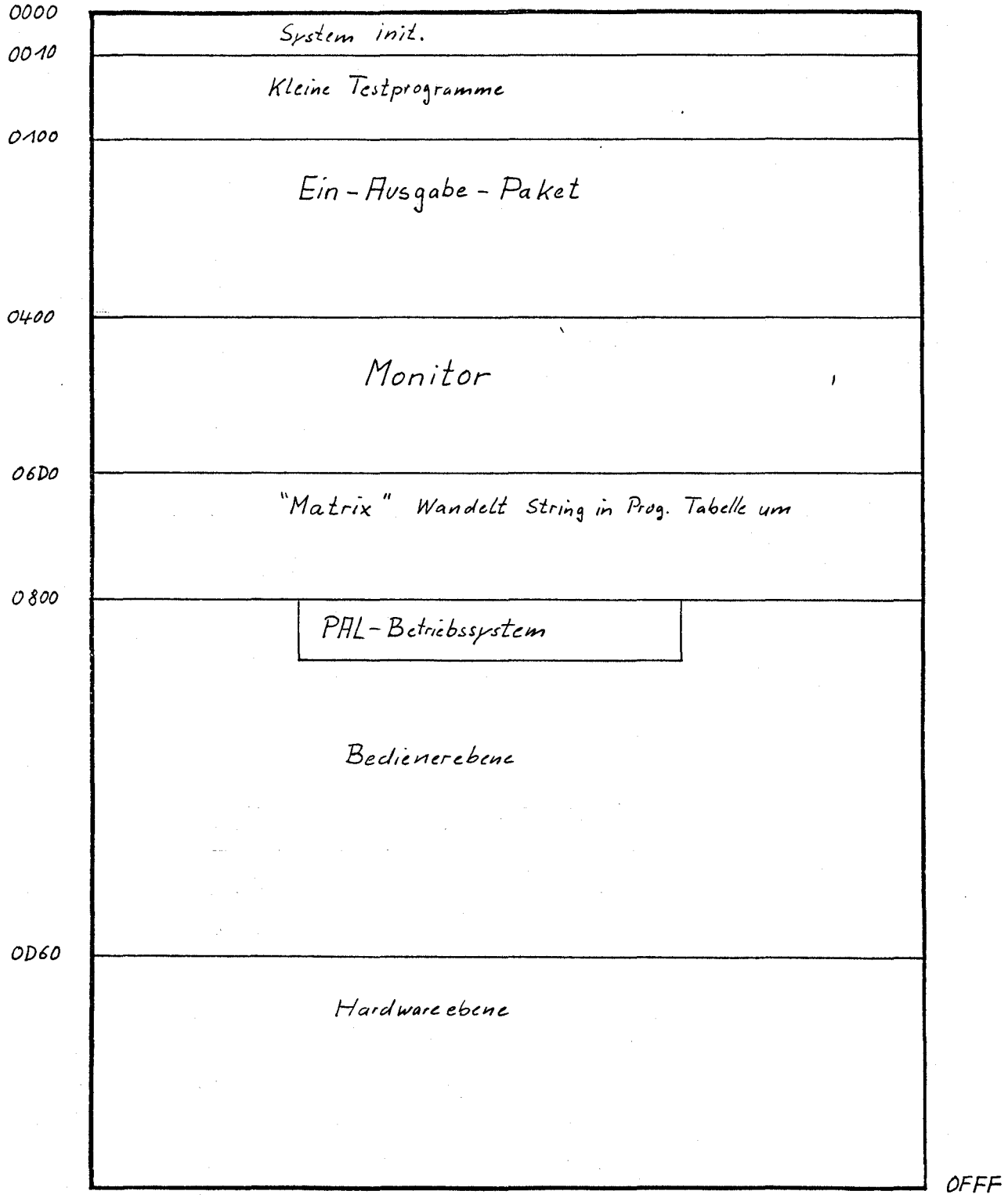
Speicherdekodierung der CPU - Karte

Name	von	bis
EPROM	0000	0FFF
1.RAM	1000	13FF
2.RAM	2000	23FF

Die folgenden Speicherbelegungspläne zeigen die Lage der Programmteile, Stack, Tabellenbereiche u.s.w. an.

Anmerkung : Die Adressen der Portbausteine sind in der Beschreibung der CPU - Karte angeführt.

4.4.1 Speicherbelegungsplan "EPROM"



4.4.2 Speicherbelegungsplan "1. RAM"

1000	Verify counter VERCNT	Zeiger auf H- oder L- PAL VERADR	PAL L	PAL H	Steuerbyte für Axselect TABA1	Steuerbyte für Axselect TABA2	Steuerbyte für Axselect TABA3
1008	PRL-Prog.-Tabelle						
1048	IPL-Prog.-Tabelle						

13FF

4.4.3 Speicherbelegungsplan "2. RAM"

2000	Zeiger auf aktuellen PAL-Type	Textbuffer-Start-Adresse	Textbuffer-Zeiger Adresse	Textbuffer-Ende Adresse
2008	Befehlsbuffer Start-Adresse	Befehlsbuffer Zeiger-Adresse	Befehlsbuffer Ende-Adresse	unbenutzt
2010	Befehls-Buffer			
2020	Text-Buffer			
2300	Stack			
23FF				

A N H A N G *** B ***

B.1 Bedienungsanleitung zum Monitor

B.1.1 Befehle Gesammtübersicht

M	Memory modify
T	Tabulate memory
G	Go and execute program
O	Out to port
I	In from port
U	Userstart

B.1.2 Allgmeines

Beim Einsprung meldet sich das Monitorprogramm mit :

```
MONITOR MODE
>
```

Innerhalb des Monitor erscheint jeweils nach Beendigung einer Routine immer das "Prompt" Sympol.

```
>
```

Mit diesen Zeichen symbolisiert der Monitor seine Empfangsbereitschaft zur Eingabe neuer Komandos, wie sie bei Kapitel B.1.1 aufgelistet sind.

B.1.3 Bedienung der einzelnen Befehle

B.1.3.1 Memory modify

Der Modifymodus ermöglicht das Verändern und Darstellen einzelner Speicherinhalte. Dieser Modus wird durch die Eingabe des groß- oder kleinbuchstaben "M" gestartet. Unmittelbar danach erscheint folgende Meldung :

```
Memory modify in :
```

Nun muß der Anwender die Speicheradresse (in sedezimal) die er verändern oder betrachten möchte, eingeben. Die Eingabe wird mit der <ENTER> oder auch <RETURN> Taste abgeschloßen.

Nach dieser Eingaben meldet sich diese Adresse auf den Bildschirm. Sie ist gefolgt von den Wert der angewählten Speicherstelle und kurz danach ihrer interpretation als ASCII Zeichen (falls möglich).

```
Memory modify in :1000 <ENTER>
```

darauf reagiert das Programm mit:

```
1000 30 :0:
```

Nun gibt es verschiedene Möglichkeiten fortzufahren

(A) Eingabe von <ENTER>

```
1000 30 :0:<ENTER>
1001 31 :1:
```

Die Adresse wird hochgezählt ohne dabei den Inhalt der Speicherzelle zu verändern.

(B) Eingabe von <RUB> oder bei einigen Terminals <DEL>

```
1000 30 :0:<RUB>
0FFF 39 :9:
```

Die Adresse wird um eins erniedrigt, ohne dabei den Inhalt der Speicherzelle zu verändern.

(C) Eingabe von "." (Punkt)

```
1000 30 :0:.
>
```

Mit den "Punkt" Zeichen wird der modify Modus verlassen und in die Monitor Hauptschleife zurückgesprungen.

(D) Eingabe von "

```
1000 30 :0:"
```

In diesem Fall folgt keine direkte Reaktion , der Monitor erwartet nun die Eingabe von Text. Der Text wird wiederum mit einem " Zeichen abgeschloßen.

```
1000 30 :0:"123456789"<ENTER>
100A 6F :r:.
>
```

Die Zahlen innerhalb der "....." sind als Text interpretiert und in den Speicher geschrieben worden. Im Speicher steht nun :

```
1000   31 32 33 34 ...... 39
```

Auch das korrigieren des Textes ist möglich. Die <RUB> oder <DEL> Taste realisiert die Backspace-funktion, d.h. der Cursor rückt nach links das, zuletzt eingegbene Zeichen wird gelöscht.

(E) Eingabe von Sedezimal Zahlenwerten

```
1000 30 :0:01<ENTER>
1001 31 :1:02<ENTER>
1002 32 :2:03<ENTER>
1003 33 :3:.
>
```

Speicherinhalt bei Adresse 1000H

```
1000   01 02 03 33 ....
```

Die eingegebenen Zahlenwerte sind in den Speicher geschrieben worden.

(F) Eingabe von Zahlenfolgen

```
1000 01 :.:20 21 22 23 24 25 26<ENTER>
1007 37 :7:.
>
```

Auch hier werden die Zahlenwerte in den Speicher geschrieben.

(G) Eingabe von Multifunktionszeilen

```
1000 20 : :"Peter" 00 00 "Michael" 00 .
>
```

Speicherinhalt bei Adresse 1000H

```
1000   50 65 74 65 72 00 00 4D 69 63 68 61 65 6C 00
```

(H) Eingabe von ungültigen Zeichen an erster Stelle (d.h. keine sedezimal Ziffer)

```
1000 50 :P:n
1001 65 :e:n
1002 74 :t:n
1003 65 :e:n
1004 72 :r:.
>
```

Die Adresse wird hochgezählt und in diese Speicherzelle eine Null eingeschrieben. Besitzt die Tastatur eine Repeatfunktion so kann auf diese Wiese in kurzer Zeit ein mittelgroßer Speicherbereich mit Nullen vollgeschrieben werden.

Speicherbelegung bei 1000H

```
1000   00 00 00 00 72 .......
```

B.1.3.2 Tabulate memory

Es wird ein Speicherbereich auf dem Bildschirm dargestellt. Am Anfang jeder Zeile steht die jeweilige Speicheradresse, danach folgen 16 Datenbytes und nach diesen noch deren interpretation als ASCII Zeichen.

Der Tabulate Mode wird durch die Eingabe des großen oder kleinen "T" gestartet. Das Programm meldet sich dann sofort mit :

TABULATE FROM :

Nun wird die Eingabe einer Adresse erwartet, die mit <ENTER> abgeschloßen werden muß. Ist das Bild aufgebaut können wir durch drücken der <ENTER> Taste diesen Modus verlassen. (Der Monitor meldet sich dann wieder mit dem "Prompt").

Durch drücken der <RUB> Taste blättern wir rückwärts, durch jede andere Taste vorwärts.

Besonderheiten: Im Anschluß an die Speicherblockausgabe folgt noch die errechnete Prüfsumme für diesen Block. Die Speicherblockdarstellung überschreibt sich beim Blättern selbst, d.h. das lästige Scrollen des Bildschirms entfällt.

Darstellungformat auf den Bilschirm :

```
ADRESS  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
 1000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
 1010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
 1020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
 1030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
 1040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
 1050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
 1060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
 1070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Checksum : 0000

B.1.3.3 Go and execute program

Der Go Befehl ermöglicht das Starten eines Programmes, das z.B. mit Modify in den Speicher geschrieben wurde.
Beginn der Go Routine durch tippen der kleinen oder großen "G" Taste. Das Programm meldet sich sofort mit :

EXECUTE PRG. BY :

Nun muß die Startadresse des Programmes folgen.

Bedienungsbeispiel 1:

Memory modify in :1000<ENTER>

```
1000 30 :0:21 00 20<ENTER>    ;LD HL,#2000
1003 33 :3:C9 .               ;RET
>

EXECUTE PRG. BY :1000<ENTER> ;starte Programm
>                            ;Programm ist beendet,(mit RET)
```

Wichtige Besonderheit !!!!

In der Regel sollte das Anwenderprogramm mit der "RET" Anweisung abgeschloßen werden , nicht mit einer Sprungadresse !!!!

Vorteile :

- Man muß sich keine Einsprungadresse für den Monitor merken.
- Ein weiterer ganz großer Vorteil ergibt sich dadurch, daß auf diese Weise auch Unterprogramme der Software getestet werden können. (Auch wenn sie im EPROM ist).

B.1.3.4 Out to Port

Ermöglicht die Ausgabe eines Datenbytes an eine Portadresse nach Wahl. Anwahl des Programmes durch tippen des großen oder kleinen "O". Das Programm meldet sich darauf sofort mit :

Out to Portadr. :

Hier möchte das Programm die Portadresse des Bausteines haben. Nach abschließen der Eingabe mit <ENTER> erscheint eine weitere Meldung .

Byte :

Jetzt muß das Datenbyte welches ausgegeben werden soll angegeben werden.

Bedienungsbeispiel 2 :

```
Out to Portadr. :20<ENTER>
        Byte    :0F<ENTER>
>
```

Das Byte 0F wird über der Port 20 ausgegeben.

B.1.3.5 In from Port

Dieses Programm gibt dem Anwender die Möglichkeit über Portadressen ansprechbare Einheiten auszulesen. (Auch DIP Schalter wenn diese mit /IOREQ dekodiert sind, siehe auch auf CPU-Karte).

Ein großes oder kleines "I" wählt dieses Programm aus. Wird seletiert, so meldet es sich sofort mit :

In from Portadr. :

Nach Anweissung geben wir die Portadresse ein und schließen die Eingabe mit <ENTER> ab. Daraufhin erscheint in der nächsten Zeile der Wert des gelesenen Bytes.

Bedienungsbeispiel 3 :

```
In from Portadr. :20<ENTER>
    FF
>
```

7.1.3.6 User instruction

Der Befehl "U" (Groß- oder Kleinbuchstabe) beendet den Monitormode und springt in das PAL - Betriebssystem.

B.2 Bedienung des Editors

Bedienertasten bei den Terminal "Hazeltine 80 Model 30"

Funktion	Taste
Cursor rechts	< >
Cursor links	< >
Insert	<CTRL> & <I>
Delete	<CTRL> & <D>
Return	<ENTER>

Der Editor akzeptiert nur die Kontrollzeichen die festgelegt sind, aber auch nur wenn ihre Benutzung sinnvoll ist.

B.2.1 Aktionen im Editiermodus

(A) Das überschreiben von Text

Mit den Cursor Funktionen positionieren und den neuen Text einfach eintippen (der alte Text wird dadurch überschrieben).

(B) Positioniern mit den Cursor Tasten

Die Cursor Tasten reagieren nur innerhalb des bereits eingebenen Textes, d.h. mit ihnen kann der Textbuffer nicht vergrößert werden.

(C) Zeicheneingabe

Die Textzeichen können an jeder Stelle eingegeben werden, der Zeichenbuffer wird bei Bedarf automatich vergrößert.

(D) Zeilenende durch <ENTER>

Die <ENTER> Taste kann an beliebiger Stelle des Textes gedrückt werden, es wird immer die gesammte Textzeile in den Buffer übernommen.

(E) Die INSERT Funktion

Diese Funktion arbeitet zeichenweise, d.h. für jedes Zeichen das eingefügt werden soll muß die INSERT Funktion einmal betätigt werden. Die Zeile und der Textbuffer werden dabei automatich vergrößert.

(F) Die DELETE Funktion

Auch diese Funktion arbeitet zeichenweise, d.h. für jedes Zeichen das gelöscht werden soll muß die DELETE Funktion je einmal gewählt werden. Die Textzeile und der Textbuffer werden automatich verkleinert.

(G) Die BACKSPACE Funktion

Das zuletzt eingegebene Zeichen wird durch ein Leerzeichen ersetzt, der Cursor wartet dann an dieser Stelle auf weiter Eingaben. Die Textbuffergröße wird nicht beeinflußt.

B.3 Bedienungsanleitung PAL-Programmer

Nach dem Einschalten des Systems meldet sich das PAL-Betriessystem wie folgt :
(leicht verkleinert)

```
                    PAL - Assembler V1.0
  ¥¥¥ Copyright (C) 1983 by M. Boretius & P. Zecherle ¥¥¥

Ihr Befehl >
```

Das System befindet sich nun Befehls Modus (hier sind die bedienungsmöglichkeiten des Editors voll aktiv).

B.3.1 Liste der Befehle im PAL- Betriebssystem

EDIT	siehe bei B.3.2.1
PROGRAMMIERE	siehe bei B.3.2.2
MONITOR	siehe bei B.3.2.3
FUSE	siehe bei B.3.2.4
LOAD	siehe bei B.3.2.5
SAVE	siehe bei B.3.2.6
NEW	siehe bei B.3.2.7
INFO	siehe bei B.3.2.8
HELP	siehe bei B.3.2.8
?	siehe bei B.3.2.8

B.3.2 Beschreibung zu den Befehlen

Es ist immer nur das erste Zeichen signifikant. Dadurch wird eine schnelle und einfache Bedienung möglich. Weitere Sicherheitsabfragen in den Programmteilen schützen vor ungewollter Fehlbedienung.

Alle anderen Befehle (außer die unter Punkt B.3.1) führen zu der Meldung :

Kein Befehl

Ihr Befehl >

B.3.2.1 EDIT Befehl

Dieser Programmteil ist noch nicht inplementiert

B.3.2.2 PROGRAMMIERE

Das Programm verzweigt zu dem eigentlichen Programmierprogramm.

B.3.2.3 MONITOR Befehl

Das PAL- Betriebssystem wird verlassen. Der Monitor meldet sich. Dieser Befehl wird nur zum Testen von Softwareteilen benötigt (dieser Befehl ist auch nicht bei der HELP Funktion aufgeführt).

B.3.2.4 FUSE Befehl

Die Programmiermatrix des PAL- Bausteines wird auf dem Bildschirm oder Drucker als Strich/Stern Matrix dargestellt. Es werden nur die relevanten Produktlines gelistet, nicht vorhandenen Inputlines werden durch Leerzeichen ersetzt.

B.3.2.5 LOAD Befehl

Von dem externen Datenträger wird ein bereits erzeugtes PAL- File abgerufen.

B.3.2.6 SAVE Befehl

Das erzeugte PAL- File wird auf den externen Datenträger ausgelagert.

B.3.2.7 NEW Befehl

Der "NEW" Befehl eröffnet ein neues PAL - File, da diese Entscheidung wesentlich ist, wird vor der eigentlichen Ausführung der Anweisung erst noch eine Sicherheitsabfrage gebracht.

Ganz neue Eingabe? (J/N):

Alle Zeichen außer "J" führen zu einen Rücksprung in dem Befehlsmodus. Ist das Zeichen "J" erkannt, beginnt der Aufbau eines neuen PAL - Files. Es erscheint die Meldung :

PAL Type bitte :

Hier wird der PAL- Type ohne dem Vorsatz "PAL" eingeben. Die Eingabezeile ist voll editierfähig und wird duch <ENTER> abgeschlossen. Sollte ein in den Tabellen nicht vorhandener PAL - Type eingegeben werden so kommt die Meldung:

PAL Type unbekannt PAL Type bitte :

Ist der PAL Type erkannt worden so folgen nun die Eingaben der Programmierinformationen. Es wird folgendes Bild aufgebaut:

----------/-------Input line bitte----------
Produkt line : 00 >

An dieser Stelle gibt es drei verschiedene Eingabemöglichkeiten (die Zeile ist voll editierfähig).

(1). Eingabe von "*" und "<ENTER>"
Alle Input line's dieser Produkt line werden nicht programmiert.

(2). Eingabe von "<ENTER>"
Alle Input line's dieser Produkt line werden programmiert.

(3) Eingabe von sedezimal Zahlenwerten für die nicht zu trennenden Sicherungen. Die Zahlen sind duch ein Leerzeichen zu trennen. Die Eingabe der Zahlenwerte wird mit <ENTER> beendet.

B.3.2.8 INFO / HELP / ? - Befehl

Jeder dieser Befehle listet die möglichen Befehle auf.

A N H A N G *** D ***

Anhang - D -

**** MMI PAL - Datenblätter ****

PAL Series 20

Absolute Maximum Ratings

	Operating
Supply Voltage, V_{CC}	7
Input Voltage	5.5V
Off-state output Voltage	5.5V
Storage temperature	-65° to +150°C

Operating Conditions

SYMBOL	PARAMETER		MILITARY MIN	MILITARY TYP	MILITARY MAX	COMMERCIAL MIN	COMMERCIAL TYP	COMMERCIAL MAX	UNIT
V_{CC}	Supply voltage		4.5	5	5.5	4.75	5	5.25	V
t_w	Width of clock	Low	25	10		25	10		ns
		High	25	10		25	10		
t_{su}	Set up time from input or feedback	16R8 16R6 16R4	45	25		35	25		ns
		16X4 16A4	55	30		45	30		
t_h	Hold time		0	-15		0	-15		ns
T_A	Operating free-air temperature		-55			0	5	75	°C
T_C	Operating case temperature				125				°C

Electrical Characteristics Over Operating Conditions

SYMBOL	PARAMETER	TEST CONDITIONS			MIN	TYP	MAX	UNIT
V_{IL}*	Low-level input voltage						0.8	V
V_{IH}*	High-level input voltage				2			V
V_{IC}	Input clamp voltage	V_{CC} = MIN	I_I = -18mA			-0.8	-1.5	V
I_{IL}	Low-level input current †	V_{CC} = MAX	V_I = 0.4V			-0.02	-0.25	mA
I_{IH}	High-level input current †	V_{CC} = MAX	V_I = 2.4V				25	μA
I_I	Maximum input current	V_{CC} = MAX	V_I = 5.5V				1	mA
V_{OL}	Low-level output voltage	V_{CC} = MIN, V_{IL} = 0.8V, V_{IH} = 2V	10H8, 12H6, 14H4, 16H2, 16C1, 10L8, 12L6, 14L4, 16L2	MIL, COM I_{OL} = 8mA		0.3	0.5	V
			16L8 16R8 16R6 16R4 16X4 16A4	MIL I_{OL} = 12mA				
				COM I_{OL} = 24mA				
V_{OH}	High-level output voltage	V_{CC} = MIN, V_{IL} = 0.8V, V_{IH} = 2V	MIL	I_{OH} = -2mA	2.4	2.8		V
			COM	I_{OH} = -3.2mA				
I_{OZL}	Off-state output current †	V_{CC} = MAX, V_{IL} = 0.8V, V_{IH} = 2V	16L8 16R8 16R6 16R4 16X4 16A4	V_O = 0.4V			-100	μA
I_{OZH}				V_O = 2.4V			100	μA
I_{OS}	Output short-circuit current**	V_{CC} = 5V		V_O = 0V	-30	-70	-130	mA
I_{CC}	Supply current	V_{CC} = MAX	10H8, 12H6, 14H4, 16H2, 16C1, 10L8, 12L6, 14L4, 16L2			55	90	mA
			16R4, 16R6, 16R8, 16L8			120	180	
			16X4			160	225	
			16A4			170	240	

† I/O pin leakage is the worst case of I_{OZX} or I_{IX} e.g., I_{IL} and I_{OZH}.

†† All typical values are at V_{CC} = 5V, T_A = 25°C.

* These are absolute voltages with respect to pin 10 on the device and includes all overshoots due to system and/or tester noise. Do not attempt to test these values without suitable equipment.

** Only one output shorted at a time.

Anhang - D -

**** MMI PAL - Datenblätter ****

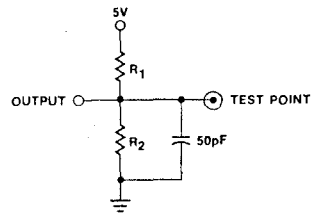
PAL Series 20

Switching Characteristics

Over Operating Conditions

SYMBOL	PARAMETER		TEST CONDITIONS	MILITARY MIN	MILITARY TYP	MILITARY MAX	COMMERCIAL MIN	COMMERCIAL TYP	COMMERCIAL MAX	UNIT
t_{PD}	Input to output	10H8 12H6 14H4 16H2 10L8 12L6 14L4 16L2	R_1 = 560Ω R_2 = 1.1kΩ		25	45		25	35	ns
		16C1			25	45		25	40	
t_{PD}	Input or feedback to output	16R6 16R4 16L8	R_1 = 200Ω R_2 = 390Ω		25	45		25	35	ns
		16X4 16A4			30	45		30	40	
t_{CLK}	Clock to output or feedback				15	25		15	25	ns
t_{PZY}	Pin 11 to output enable				15	25		15	25	ns
t_{PXZ}	Pin 11 to output disable				15	25		15	25	ns
t_{PZX}	Input to output enable	16R6 16R4 16L8			25	45		25	35	ns
		16X4 16A4			30	45		30	40	
t_{PXZ}	Input to output disable	16R6 16R4 16L8			25	45		25	35	ns
		16X4 16A4			30	45		30	40	
f_{MAX}	Maximum frequency	16R8 16R6 16R4		14	25		16	25		MHz
		16X4 16A4		12	22		14	22		

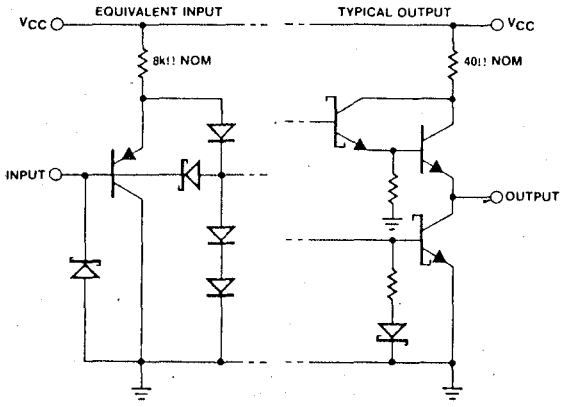
Test Load



Available Programmers

MANUFACTURER	PERSONALITY CARD SET	SOCKET ADAPTER CONFIGURATION
Data I/O Corporation	909-1427	715 1428-1 715 1428-2 715 1428-3
Pro-Log Corporation	PM9068	
Stag Systems	PM202	AM10H8 AM10L8 AM12H6 AM12L6 AM14H4 AM14L4 AM16H2 AM16L2 AM16C1
Structured Design	SD20/24	

Schematic of Inputs and Outputs



Anhang - D -

**** MMI PAL - Datenblätter ****

PAL Series 20

Programming

PAL fuses are programmed using a low-voltage linear-select procedure which is common to all 15 PAL types. The array is divided into two groups, products 0 thru 31 and products 32 thru 63, for which pin identifications are shown in Pin Configurations below. To program a particular fuse, both an input line and a product line are selected according to the following procedure:

Step 1 Raise Output Disable, OD, to V_{IHH}

Step 2 Select an input line by specifying I_0, I_1, I_2, I_3, I_4, I_5, I_6, I_7 and L/R as shown in Table 1.

Step 3 Select a product line by specifying A_0, A_1 and A_2 one-of-eight select as shown in Table 2.

Step 4 Raise V_{CC} (pin 20) to V_{IHH}

Step 5 Program the fuse by pulsing the output pins, O, of the selected product group to V_{IHH} as shown in Programming Waveform.

Step 6 Lower V_{CC} (pin 20) to 6.0 V

Step 7 Pulse the CLOCK pin and verify the output pin, O, to be Low for active Low PAL types or High for active High PAL types.

Step 8 Lower V_{CC} (pin 20) to 4.5 V and repeat step 7.

Step 9 Should the output not verify, repeat steps 1 thru 8 up to five (5) times.

This procedure is repeated for all fuses to be blown (see Programming Waveforms).

To prevent further verification, two last fuses may be blown by raising pin 1 and pin 11 to V_P. V_{CC} is not required during this operation.

Voltage Legend

L = Low-level input voltage, V_{IL}
H = High-level input voltage, V_{IH}
HH = High-level program voltage, V_{IHH}
Z = High impedance (e.g., 10kΩ to 5.0V)

INPUT LINE NUMBER	PIN IDENTIFICATION								
	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	L/R
0	HH	HH	HH	HH	HH	HH	HH	L	Z
1	HH	HH	HH	HH	HH	HH	HH	H	Z
2	HH	HH	HH	HH	HH	HH	HH	L	HH
3	HH	HH	HH	HH	HH	HH	HH	H	HH
4	HH	HH	HH	HH	HH	HH	L	HH	Z
5	HH	HH	HH	HH	HH	HH	H	HH	Z
6	HH	HH	HH	HH	HH	HH	L	HH	HH
7	HH	HH	HH	HH	HH	HH	H	HH	HH
8	HH	HH	HH	HH	HH	L	HH	HH	Z
9	HH	HH	HH	HH	HH	H	HH	HH	Z
10	HH	HH	HH	HH	HH	L	HH	HH	HH
11	HH	HH	HH	HH	HH	H	HH	HH	HH
12	HH	HH	HH	HH	L	HH	HH	HH	Z
13	HH	HH	HH	HH	H	HH	HH	HH	Z
14	HH	HH	HH	HH	L	HH	HH	HH	HH
15	HH	HH	HH	HH	H	HH	HH	HH	HH
16	HH	HH	HH	L	HH	HH	HH	HH	Z
17	HH	HH	HH	H	HH	HH	HH	HH	Z
18	HH	HH	HH	L	HH	HH	HH	HH	HH
19	HH	HH	HH	H	HH	HH	HH	HH	HH
20	HH	HH	L	HH	HH	HH	HH	HH	Z
21	HH	HH	H	HH	HH	HH	HH	HH	Z
22	HH	HH	L	HH	HH	HH	HH	HH	HH
23	HH	HH	H	HH	HH	HH	HH	HH	HH
24	HH	L	HH	HH	HH	HH	HH	HH	Z
25	HH	H	HH	HH	HH	HH	HH	HH	Z
26	HH	L	HH	HH	HH	HH	HH	HH	HH
27	HH	H	HH	HH	HH	HH	HH	HH	HH
28	L	HH	HH	HH	HH	HH	HH	HH	Z
29	H	HH	HH	HH	HH	HH	HH	HH	Z
30	L	HH	HH	HH	HH	HH	HH	HH	HH
31	H	HH	HH	HH	HH	HH	HH	HH	HH

Table 1 Input Line Select

PRODUCT LINE NUMBER	PIN IDENTIFICATION						
	O_3	O_2	O_1	O_0	A_2	A_1	A_0
0, 32	Z	Z	Z	HH	Z	Z	Z
1, 33	Z	Z	Z	HH	Z	Z	HH
2, 34	Z	Z	Z	HH	Z	HH	Z
3, 35	Z	Z	Z	HH	Z	HH	HH
4, 36	Z	Z	Z	HH	HH	Z	Z
5, 37	Z	Z	Z	HH	HH	Z	HH
6, 38	Z	Z	Z	HH	HH	HH	Z
7, 39	Z	Z	Z	HH	HH	HH	HH
8, 40	Z	Z	HH	Z	Z	Z	Z
9, 41	Z	Z	HH	Z	Z	Z	HH
10, 42	Z	Z	HH	Z	Z	HH	Z
11, 43	Z	Z	HH	Z	Z	HH	HH
12, 44	Z	Z	HH	Z	HH	Z	Z
13, 45	Z	Z	HH	Z	HH	Z	HH
14, 46	Z	Z	HH	Z	HH	HH	Z
15, 47	Z	Z	HH	Z	HH	HH	HH
16, 48	Z	HH	Z	Z	Z	Z	Z
17, 49	Z	HH	Z	Z	Z	Z	HH
18, 50	Z	HH	Z	Z	Z	HH	Z
19, 51	Z	HH	Z	Z	Z	HH	HH
20, 52	Z	HH	Z	Z	HH	Z	Z
21, 53	Z	HH	Z	Z	HH	Z	HH
22, 54	Z	HH	Z	Z	HH	HH	Z
23, 55	Z	HH	Z	Z	HH	HH	HH
24, 56	HH	Z	Z	Z	Z	Z	Z
25, 57	HH	Z	Z	Z	Z	Z	HH
26, 58	HH	Z	Z	Z	Z	HH	Z
27, 59	HH	Z	Z	Z	Z	HH	HH
28, 60	HH	Z	Z	Z	HH	Z	Z
29, 61	HH	Z	Z	Z	HH	Z	HH
30, 62	HH	Z	Z	Z	HH	HH	Z
31, 63	HH	Z	Z	Z	HH	HH	HH

Table 2 Product Line Select

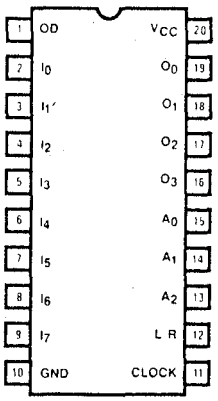
Anhang - D -

**** MMI PAL - Datenblätter ****

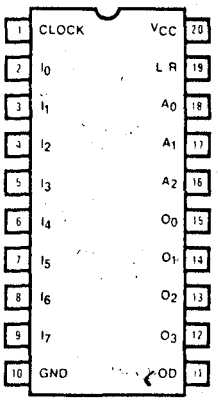
PAL Series 20

Pin Configurations

PRODUCTS 0 THRU 31



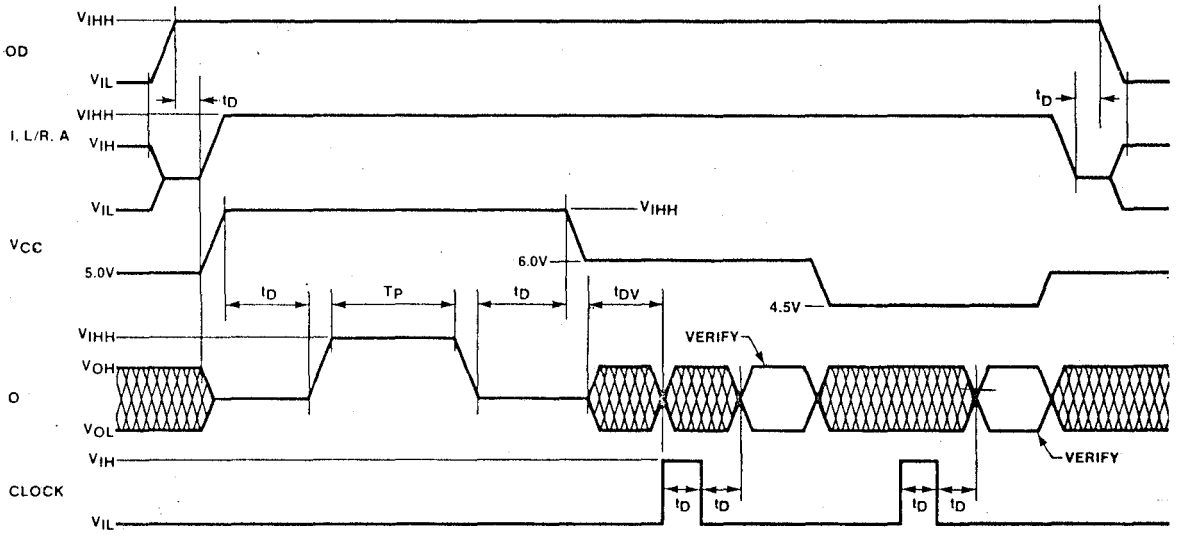
PRODUCTS 32 THRU 63



Programming Parameters $T_A = 25°C$

SYMBOL	PARAMETER		LIMITS MIN	TYP	MAX	UNIT
V_{IHH}	Program-level input voltage		11	11.5	12	V
I_{IHH}	Program-level input current	Output Program Pulse			50	mA
		OD, L/R			25	
		All Other Inputs			5	
I_{CCH}	Program Supply Current				400	mA
T_P	Program Pulse Width		10		50	µs
t_D	Delay time		100			ns
t_{DV}	Delay Time to Verify		100			µs
	Program Pulse duty cycle				25	%
V_P	Verify-Protect-input voltage		20	21	22	V
I_P	Verify-Protect-input current				400	mA
T_{PP}	Verify-Protect Pulse Width		20		50	msec

Programming Waveforms



Anhang - D -

**** Datenblätter DAC 1006 ****

National Semiconductor

A to D, D to A

MICRO-DAC:™ DAC1000/1/2 and DAC1006/7/8, μP Compatible, Double-Buffered D to A Converters

General Description

The DAC1000/1/2 and DAC1006/7/8 are advanced CMOS/Si-Cr 10-, 9- and 8-bit accurate multiplying DACs which are designed to interface directly with the 8080, 8048, 8085, Z-80 and other popular microprocessors. These DACs appear as a memory location or an I/O port to the μP and no interfacing logic is needed.

These devices, combined with an external amplifier and voltage reference, can be used as standard D/A converters; and they are very attractive for multiplying applications (such as digitally controlled gain blocks) since their linearity error is essentially independant of the voltage reference. They become equally attractive in audio signal processing equipment as audio gain controls or as programmable attenuators which marry high quality audio signal processing to digitally based systems under microprocessor control.

All of these DACs are double buffered. They can load all 10 bits or two 8-bit bytes and the data format can be either right justified or left justified. The analog section of these DACs is essentially the same as that of the DAC1020.

The DAC1000 series are the 10-bit members of a family of microprocessor-compatible DAC's (MICRO-DAC's™). For applications requiring other resolutions, the DAC0830 series (8 bits) and the DAC1208 and DAC1230 (12 bits) are available alternatives.

Part #	Accuracy (bits)	Pin	Description
DAC1000	10	24	Has all logic features
DAC1001	9		
DAC1002	8		
DAC1006	10	20	For left-justified data
DAC1007	9		
DAC1008	8		

MICRO-DAC™ is a trademark of National Semiconductor Corp.

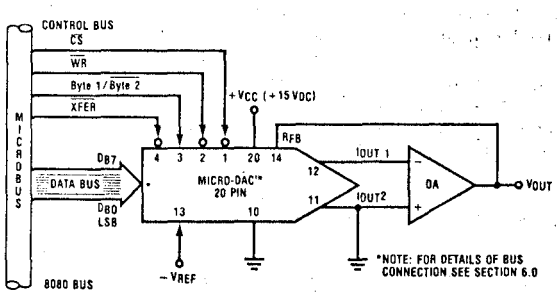
Features

- Uses easy to adjust END POINT specs, NOT BEST STRAIGHT LINE FIT
- Low power consumption
- Direct interface to all popular microprocessors.
- Integrated thin film on CMOS structure
- Double-buffered, single-buffered or flow through digital data inputs.
- Loads two 8-bit bytes or a single 10-bit word.
- Logic inputs which meet T^2L voltage level specs (1.4V logic threshold).
- Works with ±10V reference — full 4-quadrant multiplication.
- Operates STAND ALONE (without μP) if desired.
- Available in 0.3" standard 20-pin and 0.6" 24-pin package.
- Differential non-linearity selection available as special order.

Key Specifications

- Output Current Settling Time — 500ns
- Resolution — 10 bits
- Linearity — 10, 9, and 8 bits (guaranteed over temp.)
- Gain Tempco — −0.0003% of FS/°C
- Low Power Dissipation (including ladder) — 20mW
- Single Power Supply — 5 to 15V_{DC}

Typical Application



DAC1006/1007/1008

Anhang - D -

**** Datenblätter DAC 1006 ****

Absolute Maximum Ratings (Notes 1 and 2)

Supply Voltage (V_{CC})	17 V_{DC}
Voltage at any digital input	V_{CC} to GND
Voltage at V_{REF} input	±25V
Storage temperature range	−65°C to +150°C
Package dissipation at T_A = 25°C (Note 3)	500 mW
DC voltage applied to I_{OUT1} or I_{OUT2} (Note 4)	−100 mV to V_{CC}
Lead temperature (soldering, 10 seconds)	300°C

Operating Ratings

Temperature Range	
Part numbers with 'LCN' suffix	0°C to 70°C
Part numbers with 'LCD' suffix	−40°C to +85°C
Part numbers with 'LD' Suffix	−55°C to +125°C
Voltage at any digital input	V_{CC} to GND

General Electrical Characteristics T_A = 25°C, V_{REF} = 10.000 V_{DC} unless otherwise noted

Parameter	Conditions	See Note	V_{CC} = 12V_{DC} ± 5% to 15V_{DC} ± 5%			V_{CC} = 5V_{DC} ± 5%			Units
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Resolution					10			10	bits
Linearity Error	Endpoint adjust only	4,7							
	$T_{MIN} < T_A < T_{MAX}$	6							
	−10V ≤ V_{REF} ≤ +10V	5							
	DAC1000 and 1006				0.05			0.05	% of FSR
	DAC1001 and 1007				0.1			0.1	% of FSR
	DAC1002 and 1008				0.2			0.2	% of FSR
Differential Nonlinearity	Endpoint adjust only	4,7							
	$T_{MIN} < T_A < T_{MAX}$	6							
	−10V ≤ V_{REF} ≤ +10V	5							
	DAC1000 and 1006				0.1			0.1	% of FSR
	DAC1001 and 1007				0.2			0.2	% of FSR
	DAC1002 and 1008				0.4			0.4	% of FSR
Monotonicity	$T_{MIN} < T_A < T_{MAX}$	4,6							
	−10V ≤ V_{REF} ≤ +10V	5							
	DAC1000 and 1006		10			10			bits
	DAC1001 and 1007		9			9			bits
	DAC1002 and 1008		8			8			bits
Gain Error	Using internal R_{fb}								
	−10V ≤ V_{REF} ≤ +10V	5	−1.0	±0.3	1.0	−1.0	±0.3	1.0	% of FS
Gain Error Tempco	$T_{MIN} < T_A < T_{MAX}$	6							
	Using internal R_{fb}	9		−0.0003	−0.001		−0.0006	−0.002	% of FS/°C
Power Supply Rejection	All digital inputs latched high								
	V_{CC} = 14.5V to 15.5V			0.003	0.008				% FSR/V
	11.5V to 12.5V			0.004	0.010				% FSR/V
	4.75V to 5.25V						0.033	0.10	%FSR/V
Reference Input Resistance			10	15	20	10	15	20	kΩ
Output Feedthrough Error	V_{REF} = 20V_{P-P}, f = 100 kHz All data inputs latched low								
	D Package			130			130		mV_{P-P}
	N Package			90			90		mV_{P-P}
Output Capacitance I_{OUT1}	All data inputs			60			60		pF
I_{OUT2}	latched low			250			250		pF
I_{OUT1}	All data inputs			250			250		pF
I_{OUT2}	latched high			60			60		pF
Supply Current Drain	$T_{MIN} \leq T_A \leq T_{MAX}$	6		0.5	2.0		0.5	2.0	mA
Output Leakage Current	$T_{MIN} \leq T_A \leq T_{MAX}$	6							
I_{OUT1}	All data inputs latched low	10			200			200	nA
I_{OUT2}	All data inputs latched high				200			200	nA
Digital Input Voltages	$T_{MIN} \leq T_A \leq T_{MAX}$	6							
	Low level								
	LD suffix				0.8			0.6	V_{DC}
	LCD or LCN suffix				0.8			0.8	V_{DC}
	High level (all parts)		2.0			2.0			V_{DC}

Anhang - D -

**** Datenblätter DAC 1006 ****

General Electrical Characteristics $T_A = 25°C$, $V_{REF} = 10.000\,V_{DC}$ unless otherwise noted

Parameter		Conditions	See Note	$V_{CC} = 12V_{DC} \pm 5\%$ to $15V_{DC} \pm 5\%$ Min.	Typ.	Max.	$V_{CC} = 5V_{DC} \pm 5\%$ Min.	Typ.	Max.	Units
Digital Input Currents		$T_{MIN} \leq T_A \leq T_{MAX}$	6							
		Digital inputs < 0.8V			−40	−150		−40	−150	μA_{DC}
		Digital inputs > 2.0V			1.0	+10		1.0	+10	μA_{DC}
Current Settling Time	t_S	$V_{IL} = 0V$, $V_{IH} = 5V$			500			500		ns
Write and $\overline{XFER}$ Pulse Width	t_W	$V_{IL} = 0V$, $V_{IH} = 5V$, $T_A = 25°C$	8	150	60		320	200		ns
		$T_{MIN} \leq T_A \leq T_{MAX}$	9	320	100		500	250		ns
Data Set Up Time	t_{DS}	$V_{IL} = 0V$, $V_{IH} = 5V$, $T_A = 25°C$	9	150	80		320	170		ns
		$T_{MIN} \leq T_A \leq T_{MAX}$		320	120		500	250		ns
Data Hold Time	t_{DH}	$V_{IL} = 0V$, $V_{IH} = 5V$ $T_A = 25°C$	9	200	100		320	220		ns
		$T_{MIN} \leq T_A \leq T_{MAX}$		250	120		500	320		ns
Control Set Up Time	t_{CS}	$V_{IL} = 0V$, $V_{IL} = 5V$, $T_A = 25°C$	9	150	60		320	180		ns
		$T_{MIN} \leq T_A \leq T_{MAX}$		320	100		500	260		ns
Control Hold Time	t_{CH}	$V_{IL} = 0V$, $V_{IH} = 5V$, $T_A = 25°C$	9	10	0		10	0		ns
		$T_{MIN} \leq T_A \leq T_{MAX}$		10	0		10	0		ns

Note 1: *"Absolute Maximum Ratings"* are those values beyond which the safety of the device cannot be guaranteed. These specifications are not meant to imply that the devices should be operated at these *"Absolute Maximum"* limits.

Note 2: All voltages are measured with respect to GND, unless otherwise specified.

Note 3: This 500 mW specification applies for all packages. The low intrinsic power dissipation of this part (and the fact that there is no way to significantly modify the power dissipation) removes concern for heat sinking.

Note 4: For current switching applications, both I_{OUT1} and I_{OUT2} must go to ground or the "Virtual Ground" of an operational amplifier. The linearity error is degraded by approximately $V_{OS} \div V_{REF}$. For example, if $V_{REF} = 10V$ then a 1 mV offset, V_{OS}, on I_{OUT1} or I_{OUT2} will introduce an additional 0.01% linearity error.

Note 5: Guaranteed at $V_{REF} = \pm 10\,V_{DC}$ and $V_{REF} = \pm 1\,V_{DC}$.

Note 6: $T_{MIN} = 0°C$ and $T_{MAX} = 70°C$ for "LCN" suffix parts.
$T_{MIN} = -40°C$ and $T_{MAX} = 85°C$ for "LCD" suffix parts.
$T_{MIN} = -55°C$ and $T_{MAX} = 125°C$ for "LD" suffix parts.

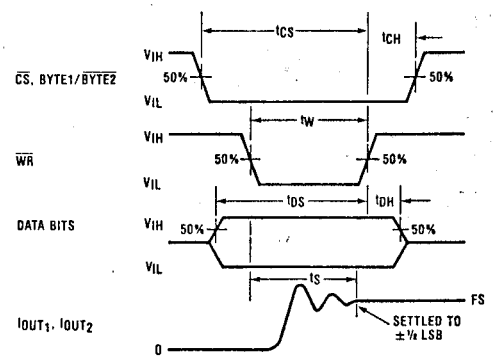
Note 7: The unit "FSR" stands for "Full Scale Range." "Linearity Error" and "Power Supply Rejection" specs are based on this unit to eliminate dependence on a particular V_{REF} value and to indicate the true performance of the part. The "Linearity Error" specification of the DAC1000 is "0.05% of FSR (MAX)." This guarantees that after performing a zero and full scale adjustment (See Sections 2.5 and 2.6), the plot of the 1024 analog voltage outputs will each be within $0.05\% \times V_{REF}$ of a straight line which passes through zero and full scale.

Note 8: This specification implies that all parts are guaranteed to operate with a write pulse or transfer pulse width (t_W) of 320 ns. A typical part will operate with t_W of only 100 ns. The entire write pulse must occur within the valid data interval for the specified t_W, t_{DS}, t_{DH}, and t_S to apply.

Note 9: Guaranteed by design but not tested.

Note 10: A 200 nA leakage current with $R_{fb} = 20k$ and $V_{REF} = 10V$ corresponds to a zero error of $(200 \times 10^{-9} \times 20 \times 10^3) \times 100 \div 10$ which is 0.04% of FS.

Switching Waveforms

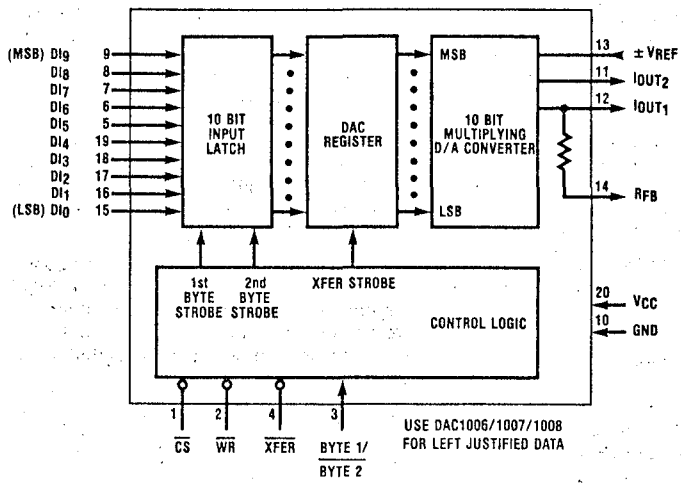


Anhang - D -

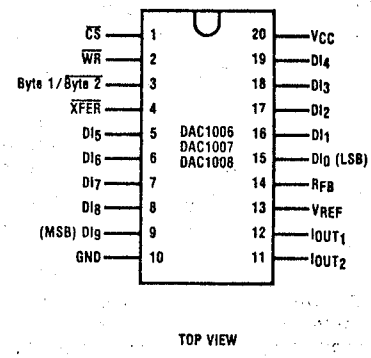
**** Datenblätter DAC 1006 ****

Block and Connection Diagrams (cont'd)

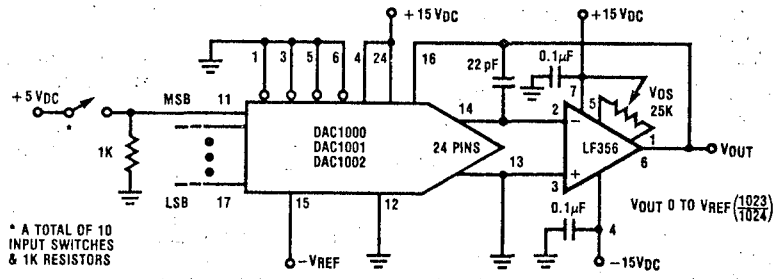
DAC1006/1007/1008 (20-Pin Parts)



DAC1006/1007/1008 (20-Pin Parts)



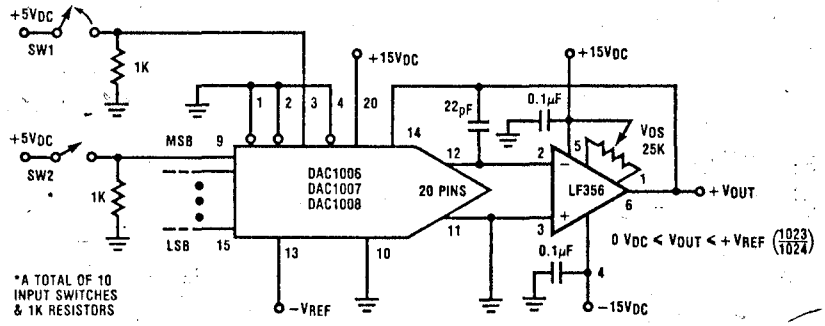
DAC1000/1001/1002 — Simple Hookup for a "Quick Look"



Notes:

1. For $V_{REF} = -10.240 V_{DC}$ the output voltage steps are approximately 10 mV each.
2. Operation is set up for flow through — no latching of digital input data.
3. Single point ground is strongly recommended.

DAC1006/1007/1008 — Simple Hookup for a "Quick Look"



Notes:

1. For $V_{REF} = -10.240 V_{DC}$ the output voltage steps are approximately 10 mV each.
2. SW1 is a normally closed switch. While SW1 is closed, the DAC register is latched and new data can be loaded into the input latch via the 10 SW2 switches. When SW1 is momentarily opened the new data is transferred from the input latch to the DAC register and is latched when SW1 again closes.

Anhang - D -

**** Datenblätter DAC 1006 ****

fore the error introduced is negligible. BI-FET™ op amps are strongly recommended for these DACs.

The distance from the I_{OUT1} pin of the DAC to the inverting input of the op amp should be kept as short as possible to prevent inadvertent noise pickup.

5.0 Analog Applications

The analog section of these DACs uses an R-2R ladder which can be operated both in the current switching mode and in the voltage switching mode.

The major product changes (compared with the DAC1020) have been made in the digital functioning of the DAC. The analog functioning is reviewed here for completeness. For additional analog applications, such as multipliers, attenuators, digitally controlled amplifiers and low frequency sine wave oscillators, refer to the DAC1020 data sheet. Some basic circuit ideas are presented in this section in addition to complete applications circuits.

5.1 Operation in Current Switching Mode

The analog circuitry, Figure 2, consists of a silicon-chromium (Si-Cr) thin film R-2R ladder which is deposited on the surface oxide of the monolithic chip. As a result, there is no parasitic diode connected to the V_{REF} pin as would exist if diffused resistors were used. The reference voltage input (V_{REF}) can therefore range from −10V to +10V.

The digital input code to the DAC simply controls the position of the SPDT current switches, SW0 to SW9. A logical 1 digital input causes the current switch to steer the available ladder current to the I_{OUT1} output pin. These MOS switches operate in the current mode with a small voltage drop across them and can therefore switch currents of either polarity. This is the basis for the 4-quadrant multiplying feature of this DAC.

5.1.1 Providing a Unipolar Output Voltage with the DAC in the Current Switching Mode

A voltage output is provided by making use of an external op amp as a current-to-voltage converter. The idea is to use the internal feedback resistor, R_{FB}, from the output of the op amp to the inverting (−) input. Now, when current is entered at this inverting input, the feedback action of the op amp keeps that input at ground potential. This causes the applied input current to be diverted to the feedback resistor. The output voltage of the op amp is forced to a voltage given by:

$$V_{OUT} = -(I_{OUT1} \times R_{FB})$$

Notice that the sign of the output voltage depends on the direction of current flow through the feedback resistor.

In current switching mode applications, both current output pins (I_{OUT1} and I_{OUT2}) should be operated at $0\,V_{DC}$. This is accomplished as shown in Figure 3. The capacitor, C_C, is used to compensate for the output capacitance of the DAC and the input capacitance of the op amp. The required feedback resistor, R_{FB}, is available on the chip (one end is internally tied to I_{OUT1}) and must be used since an external resistor will not provide the needed matching and temperature tracking. This circuit can therefore be simplified as shown in

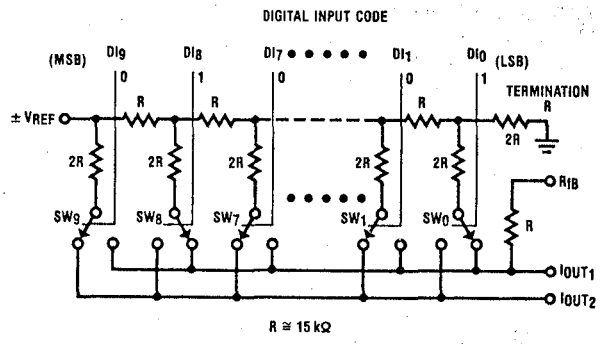
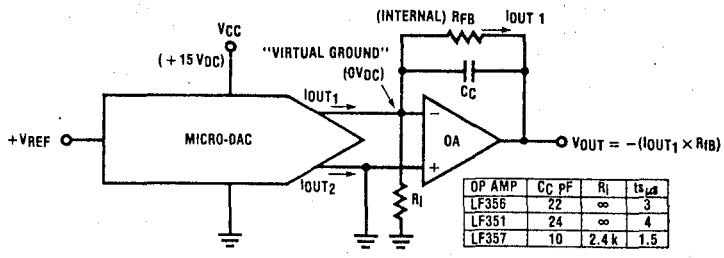


Figure 2. Current Mode Switching



OP AMP	C_C pF	R_f	t_s µs
LF356	22	∞	3
LF351	24	∞	4
LF357	10	2.4 k	1.5

Figure 3. Converting I_{OUT} to V_{OUT}

Anhang - D -

**** Datenblätter DAC 1006 ****

5.2 Analog Operation in the Voltage Switching Mode

Some useful application circuits result if the R-2R ladder is operated in the voltage switching mode. There are two very important things to remember when using the DAC in the voltage mode. The reference voltage (+V) must always be positive since there are parasitic diodes to ground on the I_{OUT1} pin which would turn on if the reference voltage went negative. To maintain a degradation of linearity less than ±0.005%, keep $+V \leq 3V_{DC}$ and V_{CC} at least 10V more positive than +V. Figures 6 and 7 show these errors for the voltage switching mode. This operation appears unusual, since a reference voltage (+V) is applied to the I_{OUT1} pin and the voltage output is the V_{REF} pin. This basic idea is shown in Figure 8.

This V_{OUT} range can be scaled by use of a non-inverting gain stage as shown in Figure 9.

Notice that this is unipolar operation since all voltages are positive. A bipolar output voltage can be obtained by using a single op amp as shown in Figure 10. For a digital input code of all zeros, the output voltage from the V_{REF} pin is zero volts. The external op amp now has a single input of +V and is operating with a gain of −1 to this input. The output of the op amp therefore will be at −V for a digital input of all zeros. As the digital code increases, the output voltage at the V_{REF} pin increases.

Notice that the gain of the op amp to voltages which are applied to the (+) input is +2 and the gain to voltages which are applied to the input resistor, R, is −1. The output voltage of the op amp depends on both of these inputs and is given by:

$$V_{OUT} = (+V)(-1) + V_{REF}(+2)$$

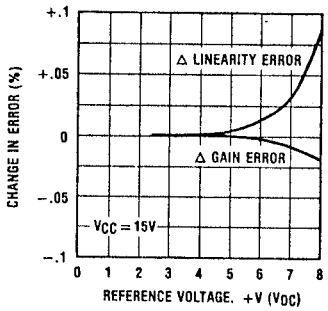


Figure 6.

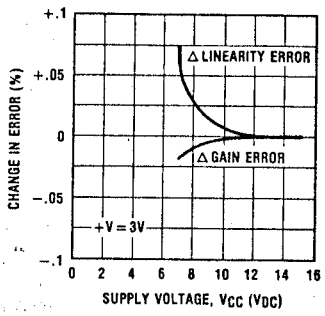


Figure 7.

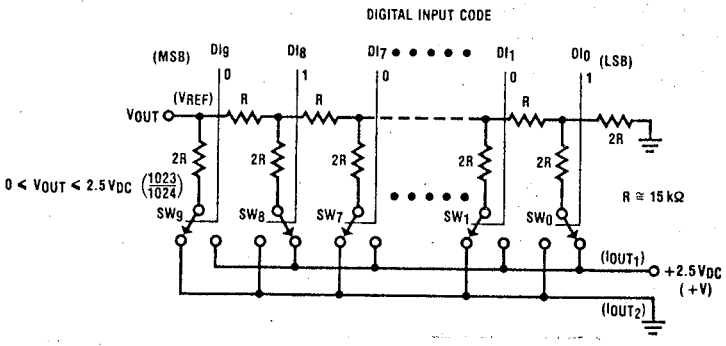


Figure 8. Voltage Mode Switching

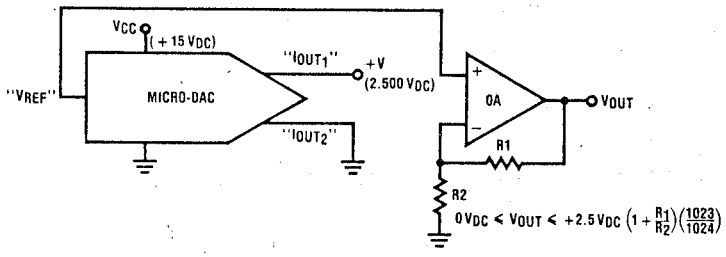


Figure 9. Amplifying the Voltage Mode Output (Single Supply Operation)

Anhang - II -

**** Datenblätter DAC 1006 ****

DAC1000/1001/1002 (24-Pin Parts)

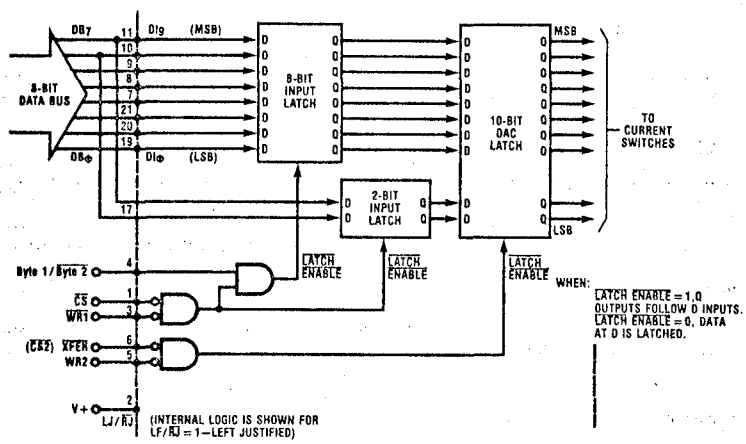


Figure 17. Input Connections and Controls for DAC1000-1002 Left Justified Data Option

DAC1006/1007/1008 (20-Pin Parts for Left Justified Data)

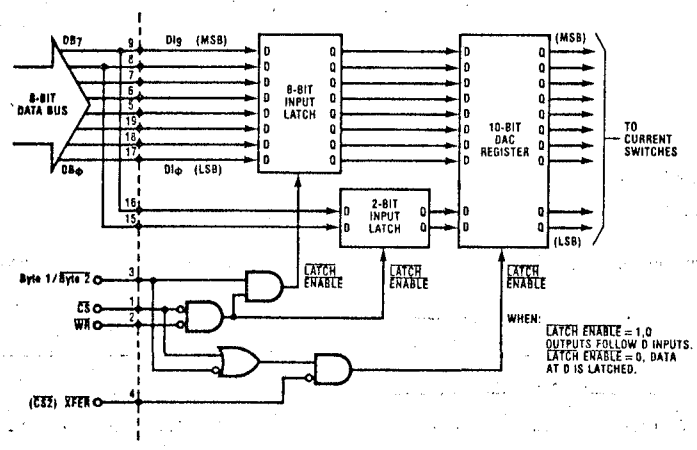


Figure 18. Input Connections and Controls for DAC1006/1007/1008 Left Justified Data

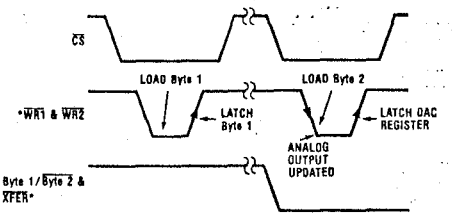
Anhang - D -

**** Datenblätter DAC 1006 ****

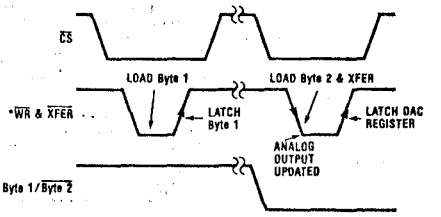
6.2.1 Automatic Transfer

This makes use of a double byte (double precision) write. The first byte (8 bits) is strobed into the input latch and the second byte causes a simultaneous strobe of the two remaining bits into the input latch and also the transfer of the complete 10-bit word from the input latch to the DAC register. This is shown in the following timing diagrams; the point in time where the analog output is updated is also indicated on these diagrams.

DAC1000/1001/1002 (24-Pin Parts)



DAC1006/1007/1008 (20-Pin Parts)

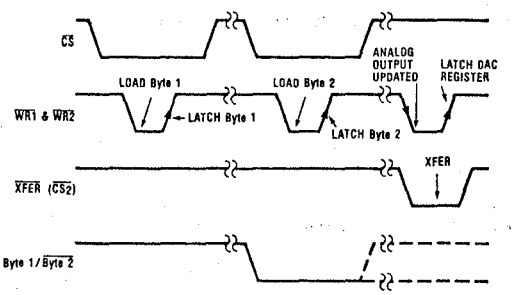


*SIGNIFIES CONTROL INPUTS WHICH ARE DRIVEN IN PARALLEL

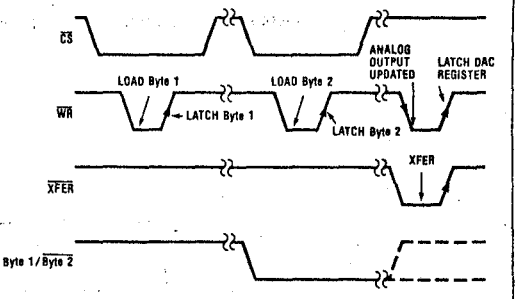
6.2.2 Transfer Using μP Write Strobe

The input latch is loaded with the first two write strobes. The XFER signal is provided by external logic, as shown below, to cause the transfer to be accomplished on a third write strobe. This is shown in the following diagrams:

DAC1000/1001/1002 (24-Pin Parts)



DAC1006/1007/1008 (20-Pin Parts)



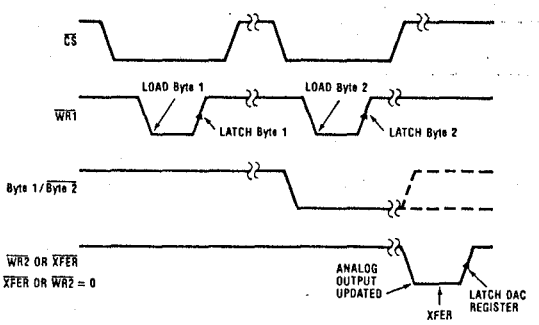
WHERE THE XFER CONTROL CAN BE GENERATED BY USING A SECOND CHIP SELECT AS:

AND THE BYTE CONTROL CAN BE DERIVED FROM THE ADDRESS BUS SIGNALS.

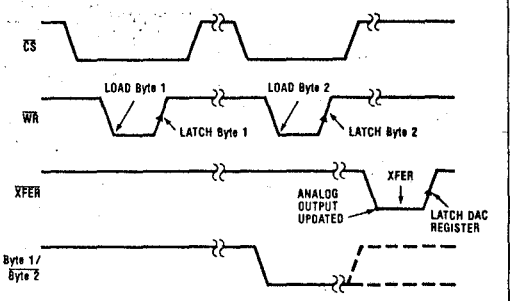
6.2.3 Transfer Using an External Strobe

This is similar to the previous operation except the XFER signal is not provided by the μP. The timing diagram for this is:

DAC1000/1001/1002 (24-Pin Parts)



DAC1006/1007/1008 (20-Pin Parts)



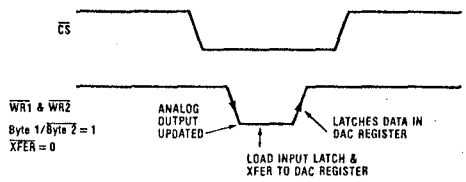
Anhang - D -

**** Datenblätter DAC 1006 ****

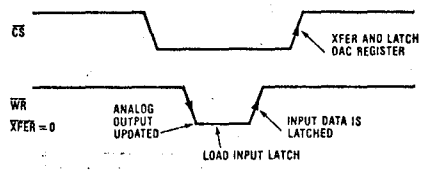
Three operating modes are possible: flow through, single buffered, or double buffered. The timing diagrams for these are shown below:

6.3.1 Single Buffered

DAC1000/1001/1002 (24-Pin Parts)

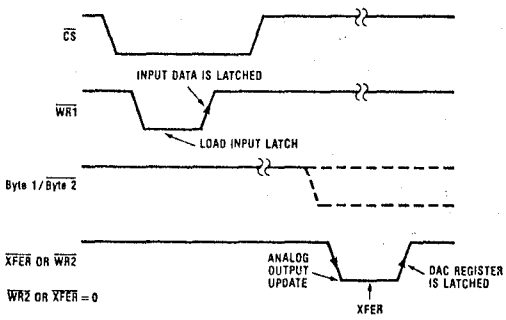


DAC1006/1007/1008 (20-Pin Parts)

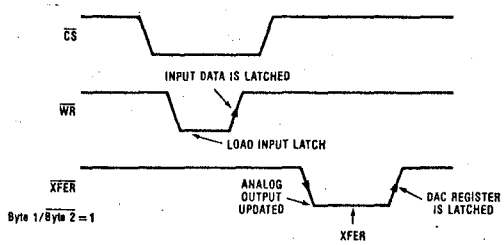


6.3.2 Double Buffered

DAC1000/1001/1002 (24-Pin Parts)



DAC1006/1007/1008 (20-Pin Parts)

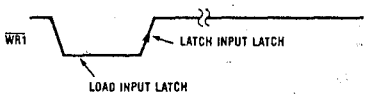


6.4 Stand Alone Operation

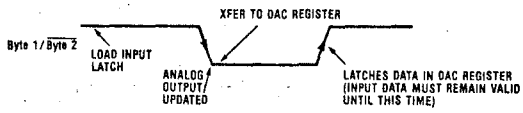
For applications for a DAC which are not under μP control (stand alone) there are two basic operating modes, single buffered and double buffered. The timing diagrams for these are shown below:

6.4.1 Single Buffered

DAC1000/1001/1002 (24-Pin Parts)

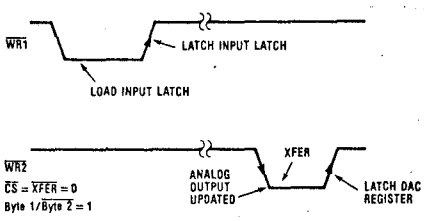


DAC1006/1007/1008 (20-Pin Parts)*

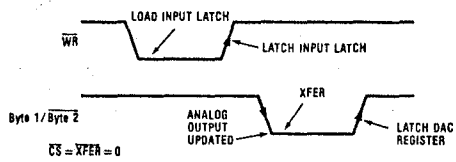


6.4.2 Double Buffered

DAC1000/1001/1002 (24-Pin Parts)

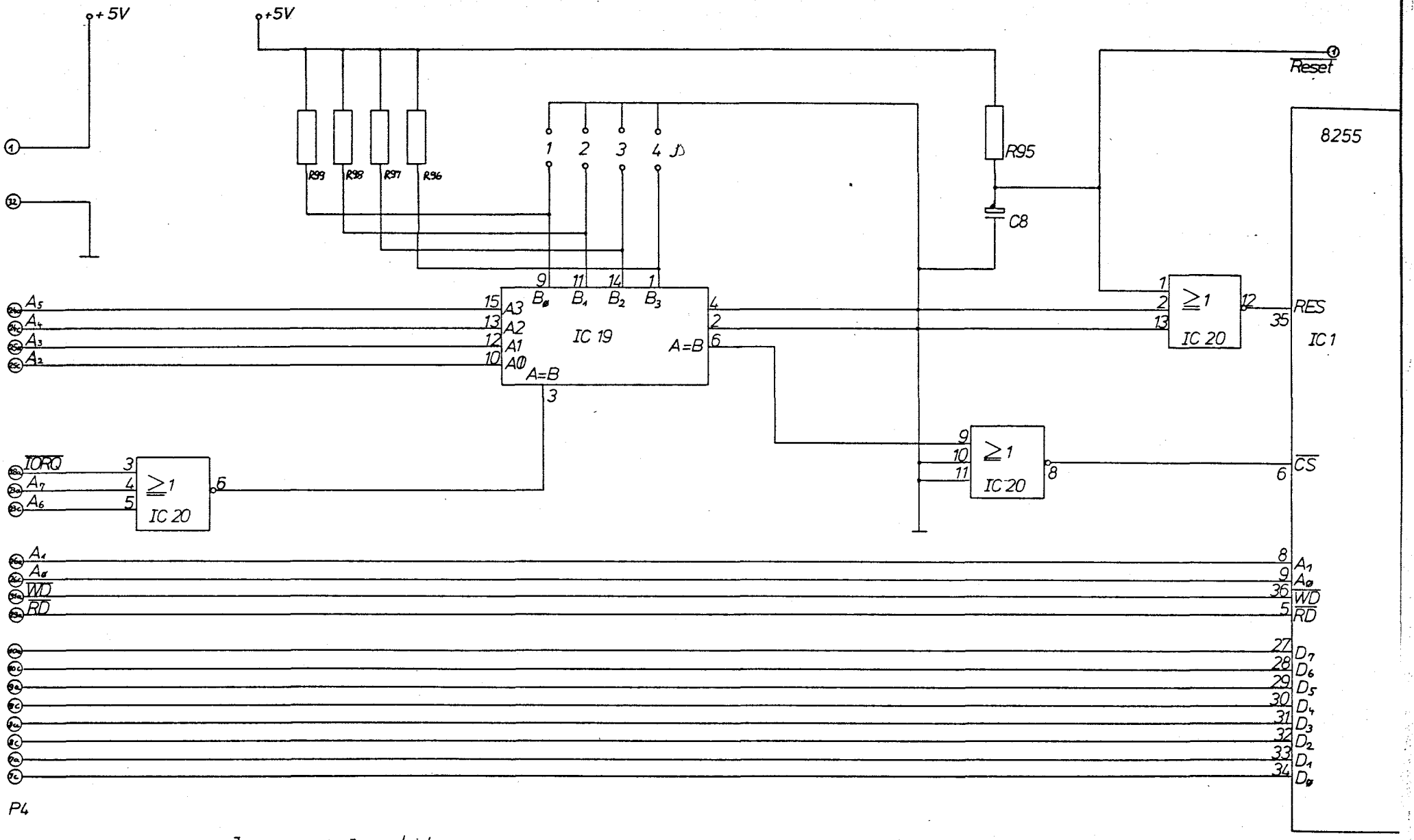


DAC1006/1007/1008 (20-Pin Parts)*



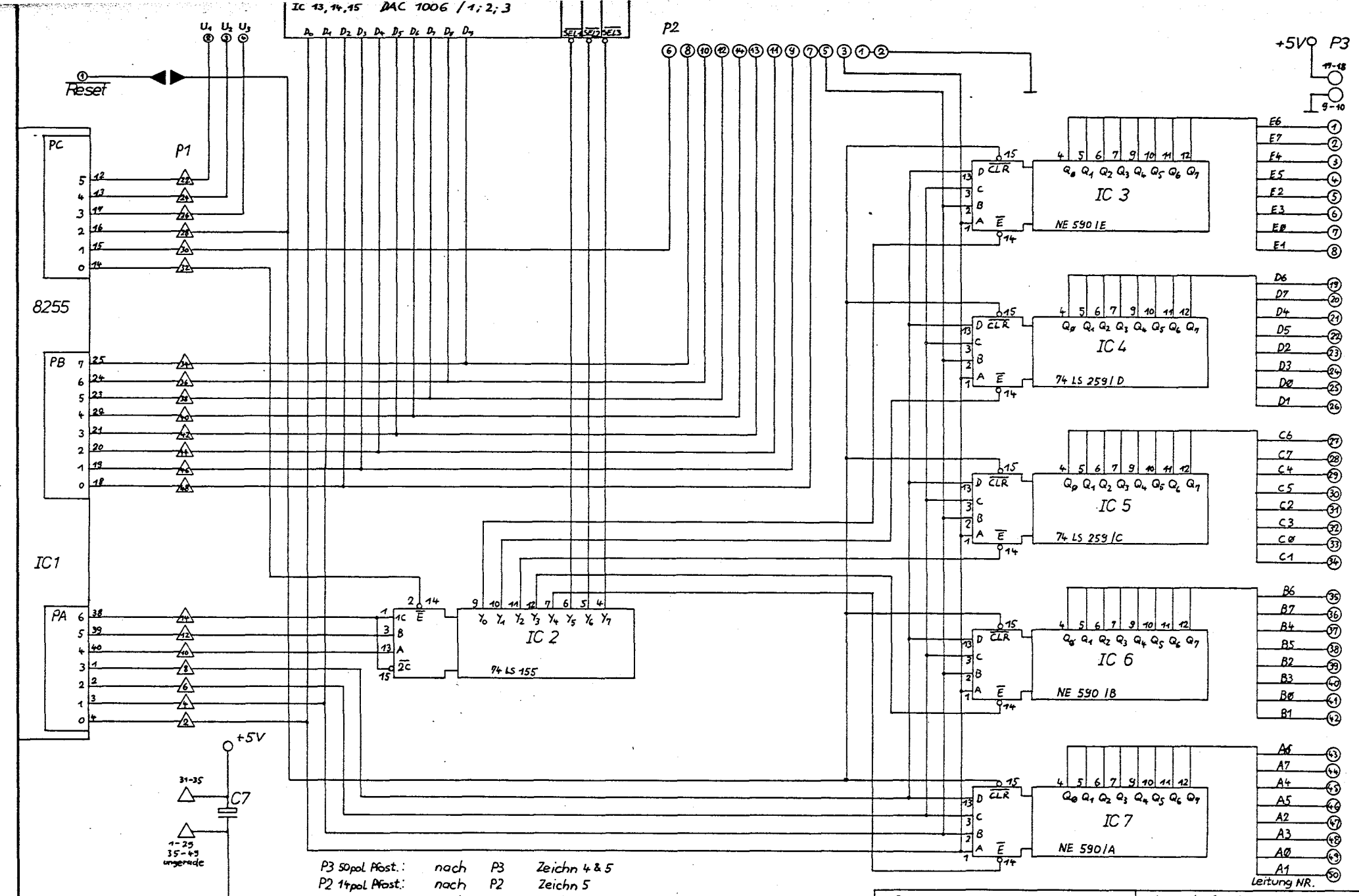
*For a connection diagram of this operating mode use Figure 18 for the Logic and Figure 20 for the Data Input connections.

A N H A N G *** H ***



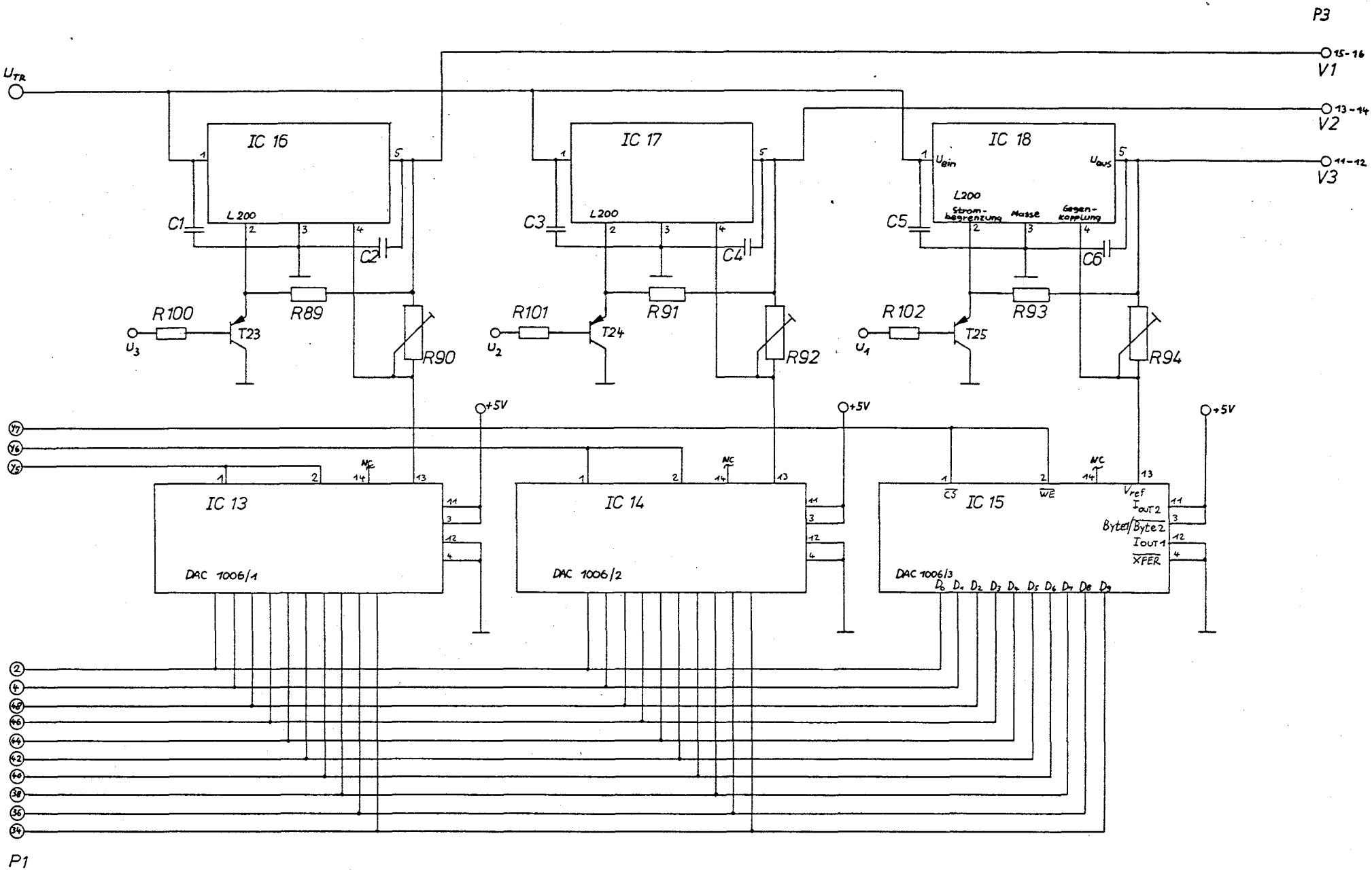
J	:	Jumperleiste
P4	:	Busstecker 64pol VG
C8	:	Tantal 4,7µF; 35 V
R95	:	22 KΩ PU 4,7 KΩ
R96 - R99	:	PU 4,7 KΩ
IC 1	:	8255
IC 19	:	74 LS 85
IC 20	:	74 LS 27

Benennung: Interfaceportdecodierung	Teil: Interfacekarte
	Zeichn. Nr.: 1 von: 10
	Gez: Boretius 10.06.83
Abschlußarbeit PAL-Programmer	Gepr:



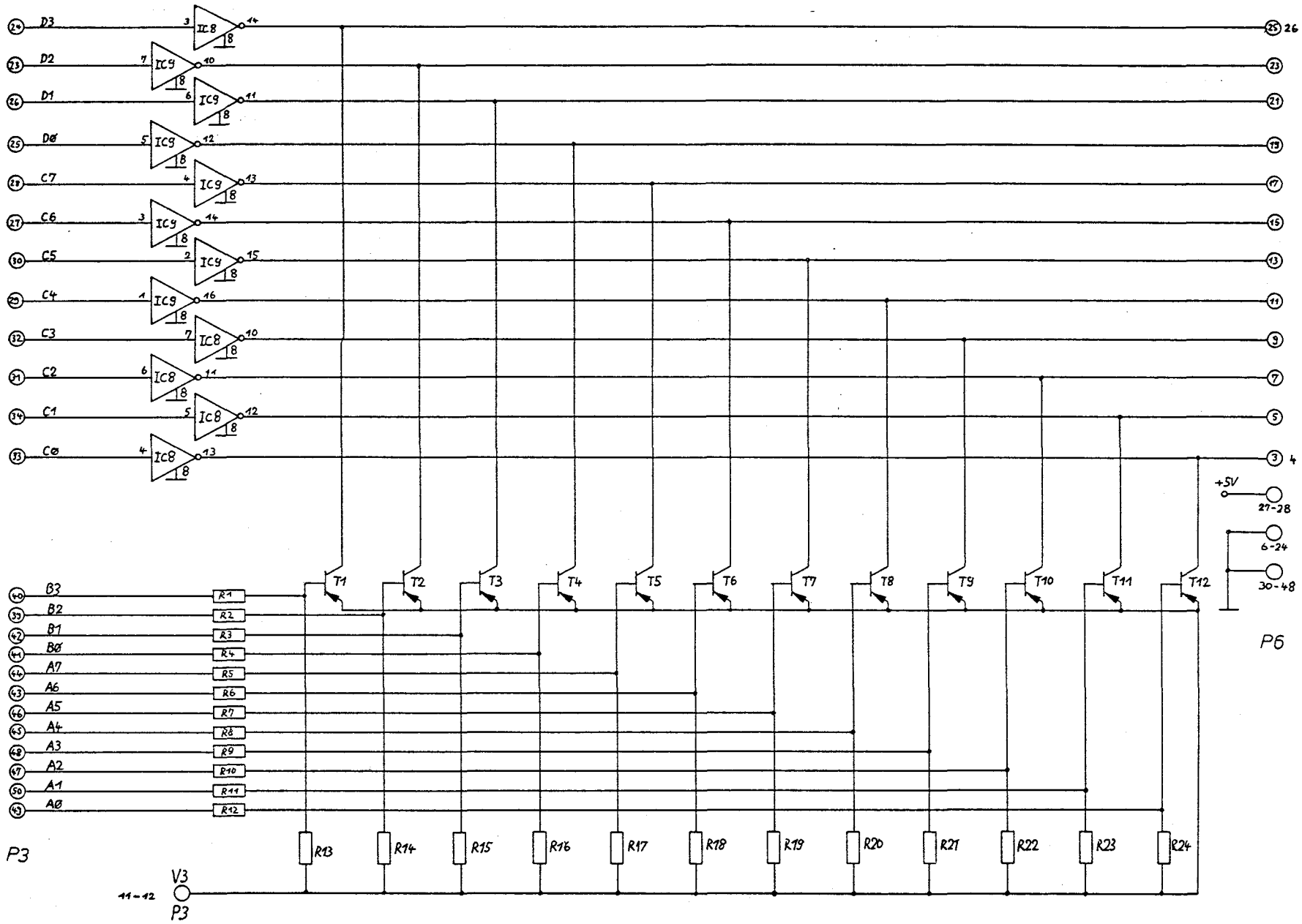
P3 50pol. Pfost.:	nach	P3	Zeichn 4 & 5				
P2 14pol. Pfost.:	nach	P2	Zeichn 5				
U_1 :	nach	R 102	Zeichn 3				
U_2 :	nach	R 101	Zeichn. 3				
U_3 :	nach	R100	Zeichn. 3	$\overline{RESET}$	:	nach $\overline{RESET}$ Zeich 1	
C7 :	10µF	Tantal	35 V	P1	:	nach P1 Zeichn. 10	

Benennung: Portexpandersteuerung	Teil: Interfacekarte
	Zeichn Nr.: 2 von: 10
	Gez: Boretius 09. 06. 83 SS 83
Abschlußarbeit PAL-Programmer	Gepr:



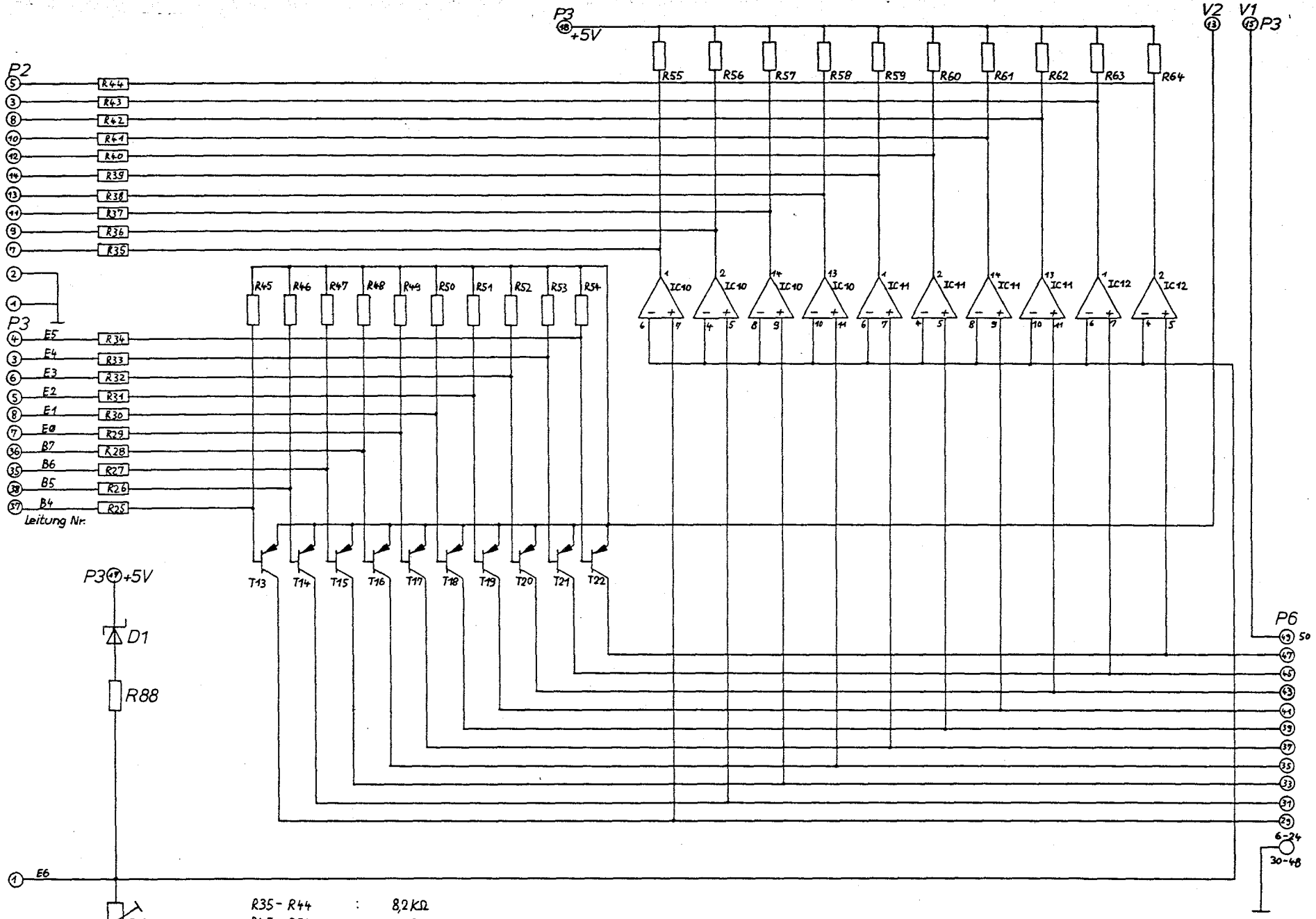
P1	:	nach	P1	Zeichn. 2	R90, R92, R94	: 500 kΩ
Y_5 - Y_7	:	nach	IC2	Zeichn. 2	R89, R91, R93	: 1 kΩ
V_1 V_2	:	nach		Zeichn. 5	R100 - 102	: 1 kΩ
V_3	:	nach		Zeichn. 4	C2, C4, C6	: 0,1 µF
U_1-U_3	:	nach		Zeichn. 2	C1, C3, C5	: 0,22 µF
U_{TR}	:		≈ 28 V unstab.		T23 - T25	: BC 516

Benennung: Spannungssteuerung	Teil: Interfacekarte
	Zeichn.Nr. 3 Von 10
	Gez: Boretius 03.06.83 5583
Abschlußarbeit PAL-Programmer	Gepr:



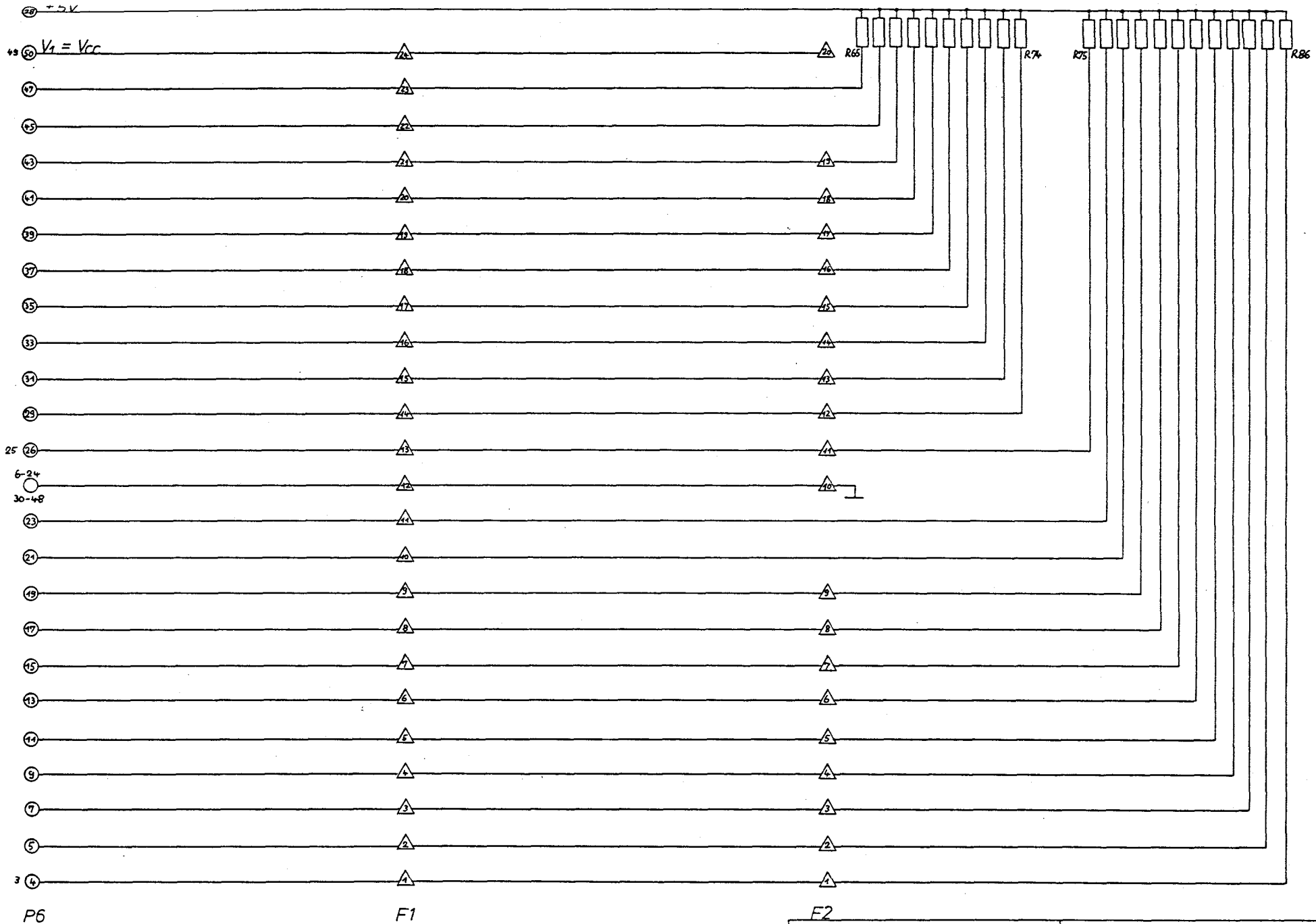
P6	:	nach P6 Zeichn. Nr. 6
P3	:	nach P3 Zeichn. Nr 2
IC8, IC9	:	ULN 2003
R1 – R24	:	10 KΩ
T2 – T11	:	BC 516
T1 & T12	:	BD 646

Benennung : Ansteuerung Linke Seite PAL	Teil : Treiberkarte
	Zeichnung Nr. 4 Von 10
	Gez : Boretius 10.06.83 SS8
Abschlußarbeit PAL-Programmer	Gepr:



R35 - R44	:	8,2 kΩ			
R45 - R54	:	10 kΩ			
R25 - R34	:	10 kΩ	P6	:	nach P6 Zeichn. Nr. 6
R87	:	25 kΩ	P2	:	nach P2 Zeichn. Nr 2
R88	:	1 kΩ	P3	:	nach P3 Zeichn. Nr. 2
D1	:	ZPD 2,6	R55 - R64 PU	:	4,7 kΩ
T13 - T22	:	BC 516	IC 10-12	:	LM 339

Benennung : Ansteuerung rechte Seite PAL	Teil : Treiberkarte
	Zeichn. Nr : 5 Von 10
Abschlußarbeit PAL-Programmer	Gez : Boretius 08. 06. 83 SS 83
	Gepr :

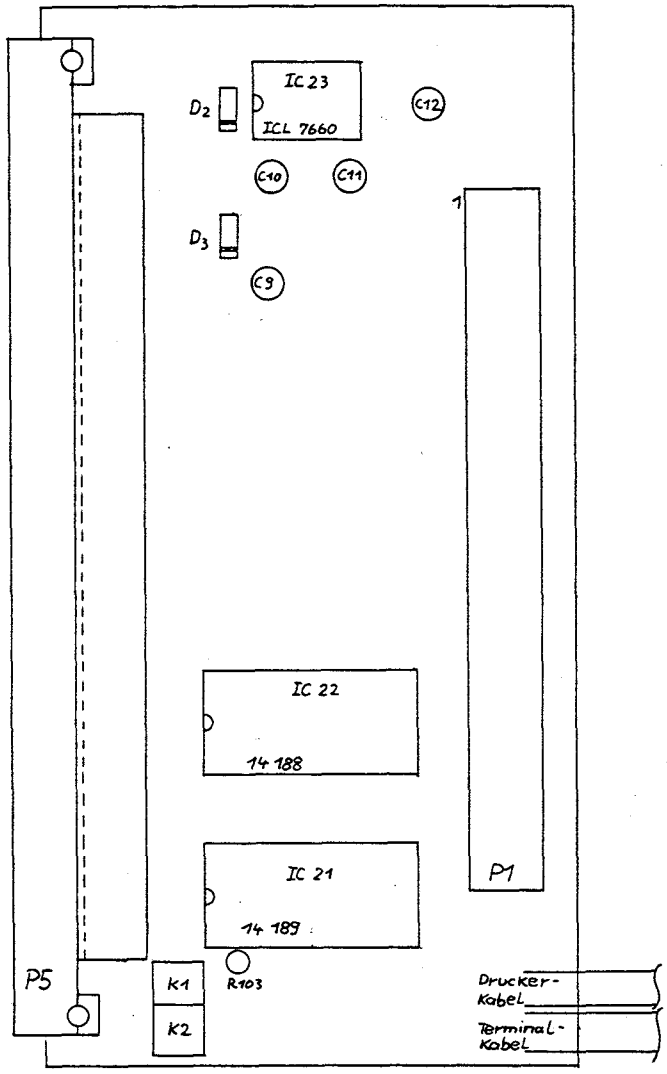


P6 → P6 Treiberkarte/ZNr: 4 & 5

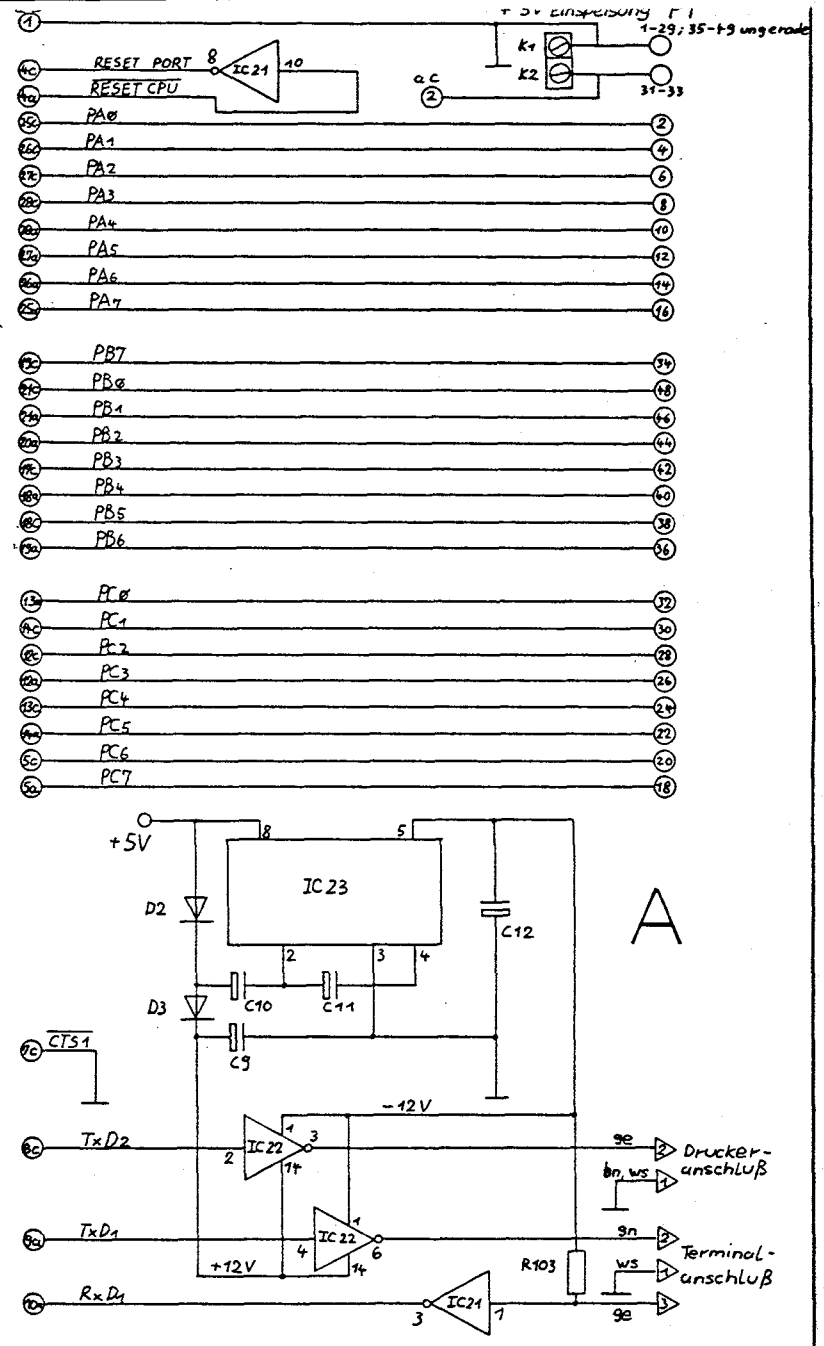
R65 – R74 = 10 kΩ
R75 – R86 = 2,7 kΩ
F1 : 24 pol. Textoolfassung
F2 : 20 pol. Textoolfassung

Benennung: Verdrahtungsplan der Fassungen	Teil: Fassungskarte
	Zeichn. Nr. 6 Von 10
	Gez.: Boretius 13.06.83 SS83
Abschlußarbeit PAL-Programmer	Gepr.:

362



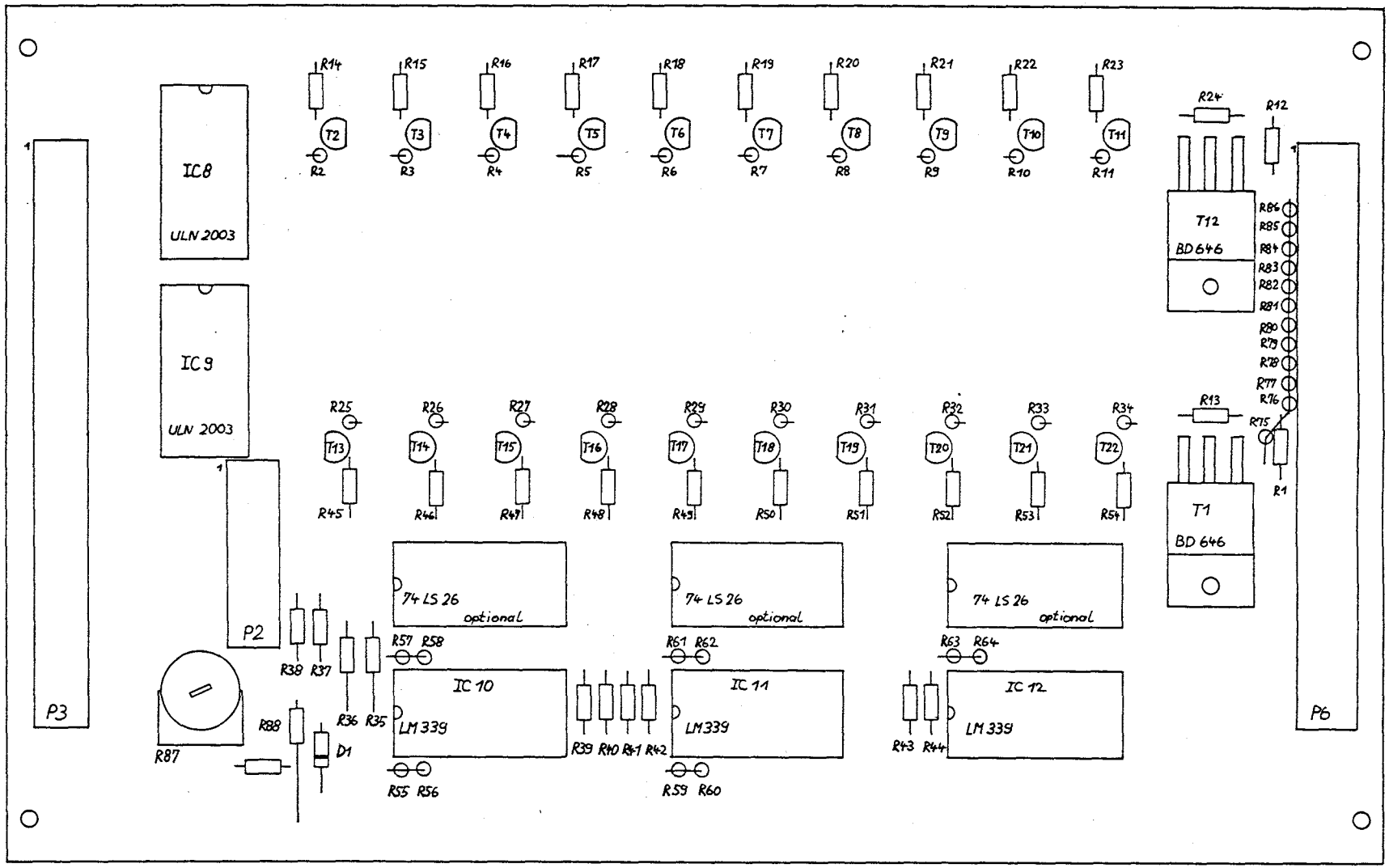
B



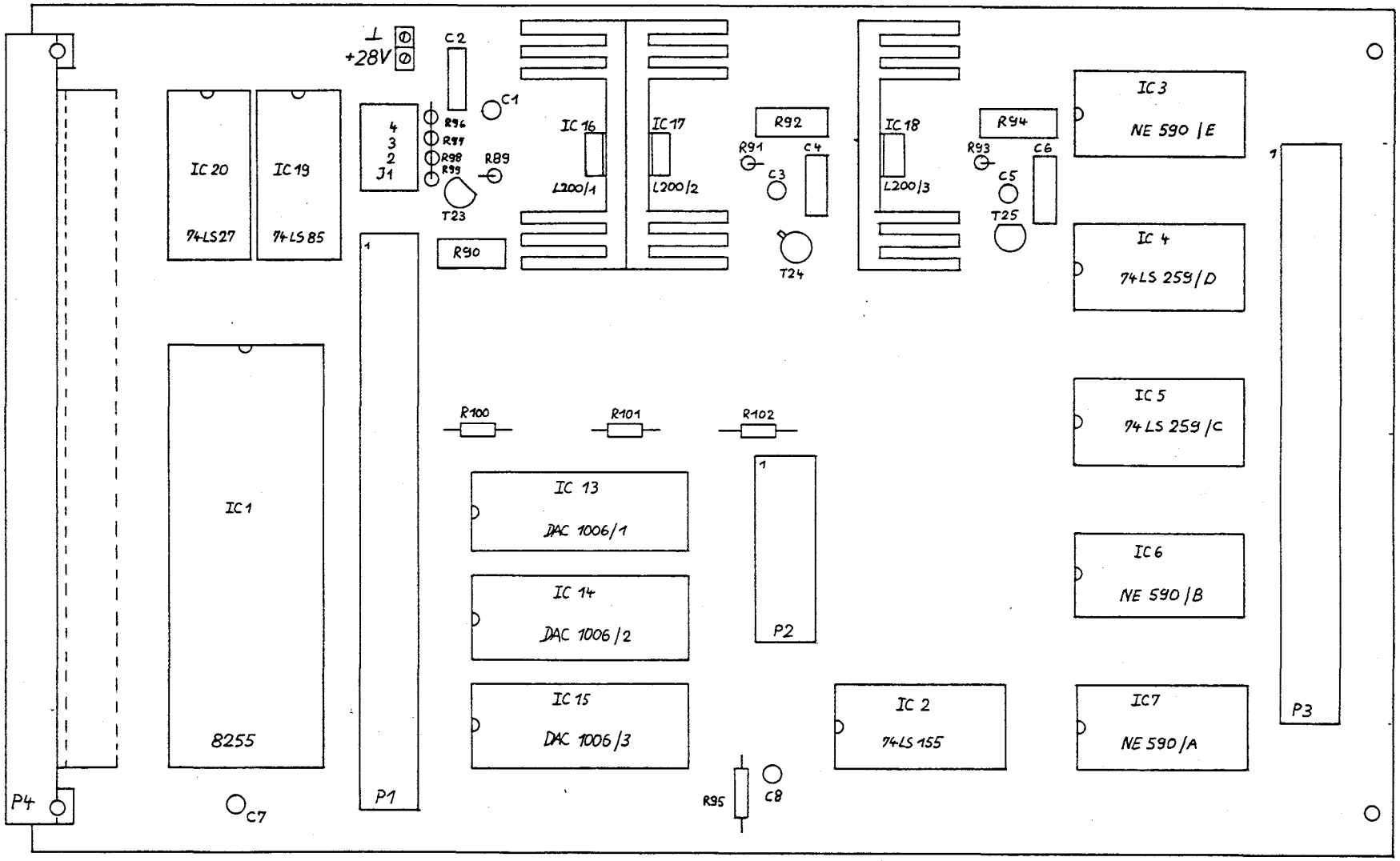
A

T-Kabel, D-Kabel :	verdrilltes Adernpaar, nach RS 232-Norm
P5 64pol VG :	nach CPU-Karte (Anhang)
P1 50pol Pfost :	nach P1, Zeichnung Nr 2
K1 - K2 :	Klemmanschluß
R103 :	4,7 KΩ
D2 D3 :	Universaldiode
C9 - C12 :	10 µF Tantal 16V

Benennung: RS 232 - Karte	Teil: A: Schaltplan B: Lageplan
	Zeichn. Nr 10 Von 10
	Gez: Boretius 18.06.83 SS83
Abschlußarbeit PAL-Programmer	Gepr:

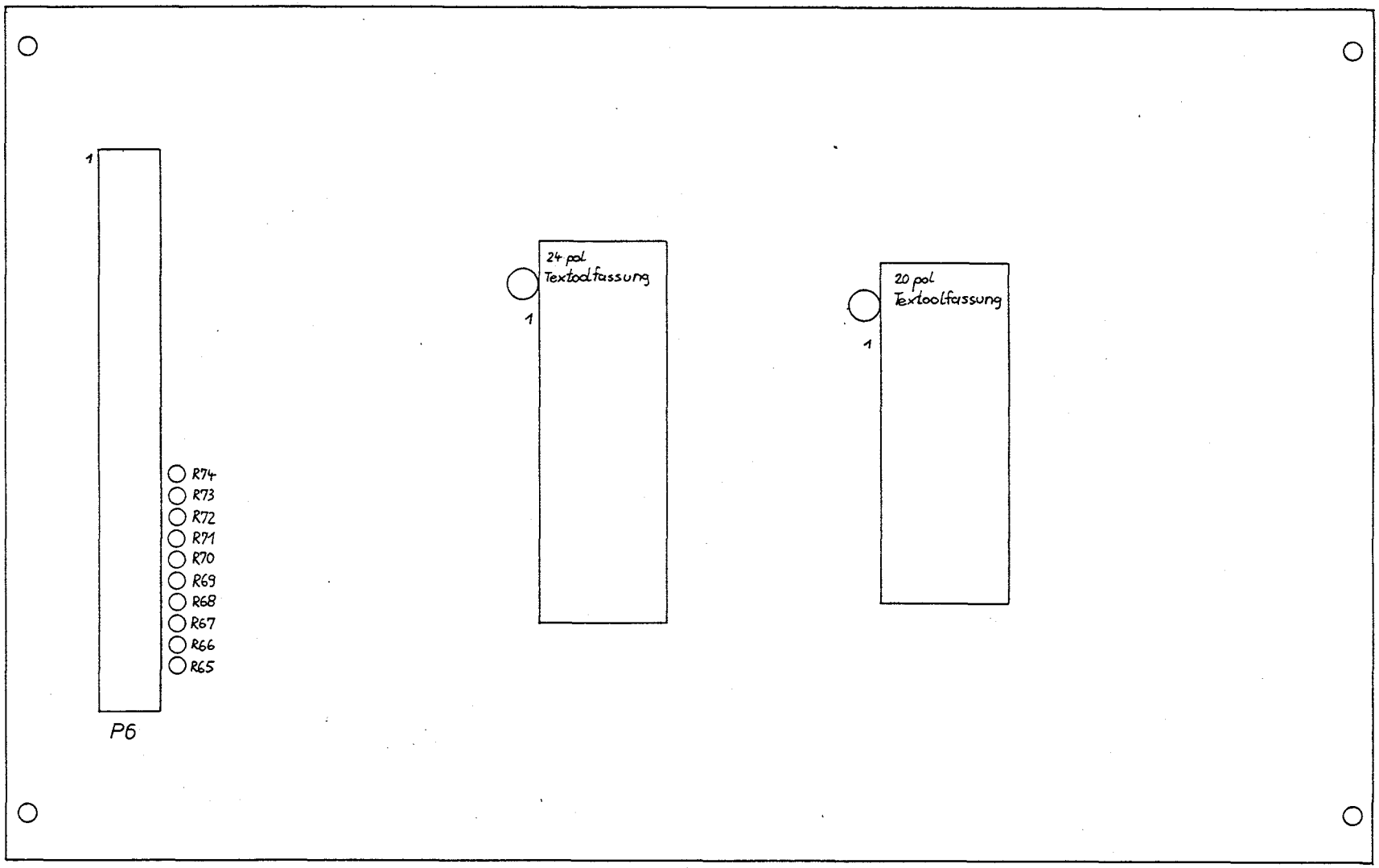


BENENNUNG : Treiberkarte	TEIL : Bauteilelageplan
	ZEICHN Nr: 7 VON: 10
	GEZ: Boretius 14.06.83 SS83
Abschlußarbeit PAL-Programmer	GEPR:



+28V = U_{TR}

Benennung : Interfacekarte	TEIL : Bauteilelageplan
	ZEICHN.Nr : 8 VON : 10
	GEZ : Boretius 13.06.83 SS83
Abschlußarbeit PAL-Programmer	GEPR:



Benennung: Fassungskarte	Teil: Lageplan
	Zeichn. Nr: 9 von 10
	Gez: Boretius 15.06.83 SS83
Abschlußarbeit PAL-Programmer	Gepr:

Stiftbelegung Expanderstecker P3

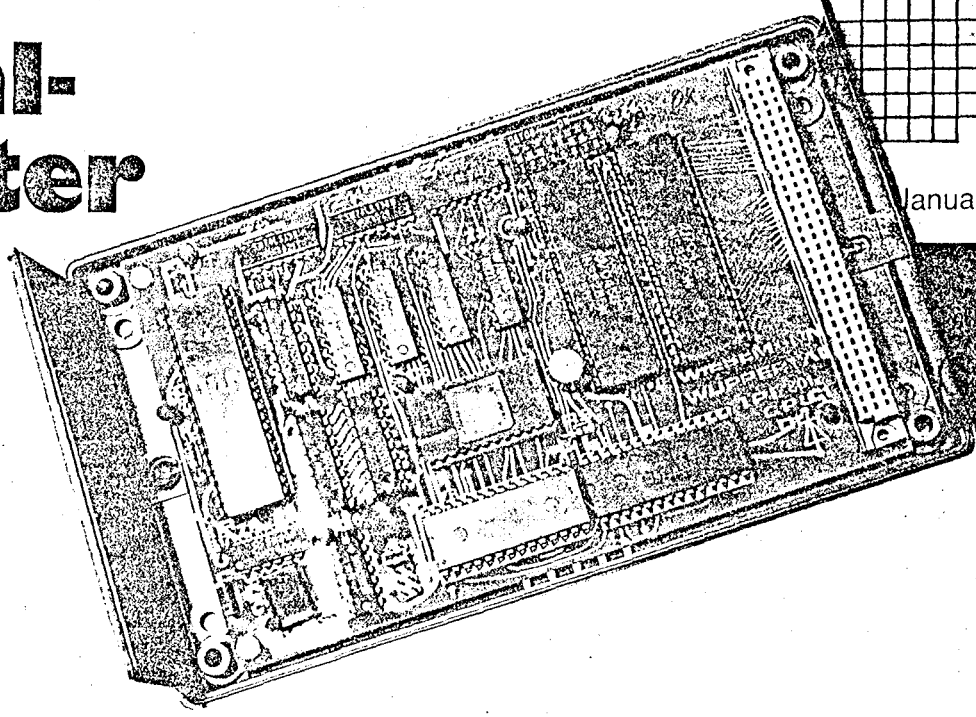
Interfacekarte

Treiberkarte

		Nr	P3		Nr		
IC3/11	Verf	E6	1	2	E7		IC3/1
IC3/9	R33	E4	3	4	E5	R34	IC3/10
IC3/6	R31	E2	5	6	E3	R33	IC3/7
IC3/4	R29	E0	7	8	E1	R30	IC3/5
gnd	gnd		9	10		gnd	gnd
V3	V3		11	12		V3	V3
V2	V2		13	14		V2	V2
V1	V1		15	16		V1	V1
+5V	+5V		17	18		+5V	+5V
IC4/11	nc	D6	19	20	D7	nc	IC4/12
IC4/9	nc	D4	21	22	D5	nc	IC4/10
IC4/6	IC9/7	D2	23	24	D3	IC8/3	IC4/7
IC4/4	IC9/5	D0	25	26	D1	IC9/6	IC4/5
IC5/11	IC9/3	C6	27	28	C7	IC9/4	IC5/12
IC5/9	IC8/1	C4	29	30	C5	IC9/2	IC5/10
IC5/6	IC8/6	C2	31	32	C3	IC8/7	IC5/7
IC5/4	IC8/4	C0	33	34	C1	IC8/5	IC5/5
IC6/11	nc	B6	35	36	B7	nc	IC6/12
IC6/9	nc	B4	37	38	B5	nc	IC6/10
IC6/6	R2	B2	39	40	B3	R1	IC6/7
IC6/4	R4	B0	41	42	B1	R3	IC6/5
IC7/11	R6	A6	43	44	A7	R5	IC7/12
IC7/9	R8	A4	45	46	A5	R7	IC7/10
IC7/6	R10	A2	47	48	A3	R9	IC7/7
IC7/4	R12	A0	49	50	A1	R11	IC7/5

Reinhard Wiesemann

Minimalcomputer mit Z80-CPU



Nachdem wir in Heft 2/81 einen Minimalcomputer mit der CPU 6504 vorgestellt haben, folgt jetzt ein ähnliches System mit dem Z80. Gegenüber seinem Vorgänger bietet es mehr Ein-/Ausgabe-Leitungen und einen größeren RAM-Bereich. Auch die Programmspeicherkapazität ist höher. Geeignete Systeme für die Programmentwicklung sind Tischcomputer mit der CPU Z80 – z. B. Nascom, TRS-80 und Video-Genie.

In kleinen Steuerungen, Schnittstellen und ähnlichen Anwendungen lassen sich Mikrocomputer aufgrund ihrer hohen Leistungsfähigkeit und Flexibilität mit Vorteil einsetzen. Für diese Bereiche werden bereits zahlreiche Systeme angeboten, die im Baukastenprinzip an das spezielle Problem angepaßt werden können: Aus einem meist umfangreichen Angebot an CPU-, Speicher- und Ein-/Ausgabe-Platinen stellt man sich das gewünschte System zusammen. Der Vorteil dieses Prinzips liegt in der leichten und praktisch unbegrenzten Erweiterbarkeit eines so aufgebauten Mikrocomputers.

Nachteilig ist, daß kleine Systeme nahezu nicht realisierbar sind: Ein minimales System benötigt meist bereits drei Computerplatinen (CPU, Speicher, Ein-/Ausgabe), eine Buskarte sowie ein Netzteil mit meist mehreren Versorgungsspannungen. Berücksichtigt man dazu das dann häufig notwendige 19-Zoll-Gehäuse, so kann das so aufgebaute System in vielen Anwendungen schon allein aus preislichen Gründen nicht eingesetzt werden.

Der hier beschriebene Minimal-Mikrocomputer vermeidet diese Nachteile, da er speziell für den Einsatz in kleinen Systemen entwickelt wurde: Der gesamte Computerteil (inklusive CPU, RAM, EPROM, Ein-/Ausgabe) befindet sich auf einer einzigen Platine, und es ist nur eine 5-V-Versorgungsspannung nötig. Auf diese Weise können kleine Mikrocomputer sehr kompakt und preiswert in Standardgehäuse eingebaut werden. Ein vollständiges System benötigt neben der Computerplatine nur eine zusätzliche Karte, die die anwendungsspezifische Schaltung (Relais, Treiber usw.) und das Netzteil beinhaltet. *Bild 1* zeigt die Schaltung des Einplatinencomputers MMC-1. In den *Bildern 2* und *3* sind Platinenlayout und Bestückungsplan dargestellt. Die Platine ist unbestückt oder fertig bestückt vom Autor beziehbar (Postfach 20 16 05, 5600 Wuppertal 2).

Schaltungsdetails

Beim Einschalten der Versorgungsspannung wird die CPU durch das aus R8 und C7 bestehende RC-Glied zurückgesetzt. Der $\overline{\text{INT}}$-Eingang kann von einer speziellen Anwenderschaltung frei benutzt oder über C10 mit dem Baud-Rate-Generator verbunden werden. In diesem Fall erhält die CPU alle 256 Taktzyklen ein INT-Signal. Da auf der Platine keine Interrupt-Vektoren erzeugt werden, muß Interrupt-Mode 1 (Restart bei 038H) programmiert werden.

Die Karte bietet die Möglichkeit, bis zu 2 KByte EPROM (1 × Typ 2716 – Z10) sowie 2 KByte RAM (4 × Typ 2114 – Z2...Z5) zu bestücken. Die acht niedrigwertigen Bits des Adreßbusses sowie der Datenbus sind durch die Bausteine Z16...Z18 gepuffert.

Z13 arbeitet als Speicher-Adreßdecoder und kann bis zu acht Speicherblöcke zu jeweils 2 KByte adressieren. Das EPROM sowie jeder RAM-Block belegen jeweils einen Bereich, so daß sich folgende Speicheraufteilung ergibt:

2400H	FFFFH	frei		
2000H	23FFH	RAM 2	Z2, Z3	1 KByte
1400H	1FFFH	frei		
1000H	13FFH	RAM 1	Z4, Z5	1 KByte
800H	FFFH	frei		
0H	7FFH	EPROM	Z10	2 KByte

Obwohl die Blockadressierung in Schritten von 2 KByte vorgenommen wird, liegt zwischen EPROM und dem ersten RAM ein Bereich von 2 KByte. Auf diese Weise ist es sehr einfach möglich, statt des EPROMs 2716 den Typ 2732 (mit doppelter Kapazität) zu verwenden (+5-V-Leitung zu Anschluß 21 auf der Platine auftrennen und Anschluß 21 an Adresse 11 legen).

Die Platine arbeitet mit einer Taktfrequenz von 2,4576 MHz, die durch den Baustein Z11 erzeugt wird. Diese Frequenz erlaubt einen besonders einfachen Aufbau des Baud-Rate-Generators: Ein Binärteiler (Z12) stellt die benötigten Frequenzen an Steckbrücken zur Verfügung:

Z12-Anschluß	Steckbrücke	Frequenz
13	1	153,6 kHz
11	2	76,8 kHz
10	3	38,4 kHz
9	4	19,2 kHz
8	5	9,6 kHz

Durch Wahl einer der fünf Brücken wird die entsprechende Frequenz an die Sende- und Empfangsregister der UARTs (Z8 und Z9) gelegt. Die Baud-Rate jedes Kanals ergibt sich aus dieser Frequenz sowie einem Teilerfaktor, der per Software programmiert werden kann. Dieser Faktor kann 1, 16 oder 64 sein, so daß die Baud-Rate im Bereich von 150 Bd (= 9,6/64) bis 153 600 Bd (= 153 600/1) liegen könnte. Erlaubt sind jedoch nur Geschwindigkeiten bis 9600 Bd (*Tabelle 1*). In den meisten Fällen bietet es sich an, den Faktor 16 zu programmieren, da damit die üblichen Geschwindigkeiten einstellbar sind. Durch Wahl unterschiedlicher Faktoren oder durch Einbau eines zweiten Satzes von Steckbrükken sind unterschiedliche Baud-Raten für beide UARTs möglich.

Ein-/Ausgabe-Möglichkeiten

Z6 und Z7 arbeiten als PIO-Bausteine und stellen dem Anwender insgesamt bis zu 48 TTL-kompatible Leitungen zur Verfügung, die als Ein- oder Ausgang zur Steuerung von Relais usw. bidirektional oder mit Handshake-Signalen benutzt werden können. Beim Einschalten der Versorgungsspannung werden diese Bausteine durch das aus R9 und C8 bestehende RC-Glied zurückgesetzt.

Der Baustein Z15 erlaubt die Abfrage von sechs DIL-Schaltern per Programm. Ein geschlossener Schalter legt das an diesem Eingabekanal liegende Bit auf „0", bei geöffnetem Schalter liegt über Pull-up-Widerstände „1" an.
Z14 arbeitet als Adreßdecoder für sämtliche Ein-/Ausgabe-Bausteine. Da die niedrigstwertigen zwei Adreßbits von den PIO-Bausteinen für interne Adressierungen verwendet werden, decodiert dieser Baustein die Adressen 2...4. Von den damit möglichen acht I/O-Adressen sind sechs belegt. Sie ergeben zusammen mit der internen Decodierung der PIOs und UARTs den in *Tabelle 2* dargestellten Ein-/Ausgabe-Bereich.
Alle Ein-/Ausgabe-Signale sowie die Versorgungsspannung sind an eine 64polige VG-Leiste geführt. An dieser Leiste können die anwenderspezifische Schaltung sowie das Netzteil aufgesteckt werden (*Tabelle 3*). Die Bezeichnungsweise der einzelnen Signale entspricht weitestgehend der in den Datenbüchern üblichen Art. Da die Bausteine 8255 und 8251 jeweils zweimal vorhanden sind, gibt eine nachgesetzte Ziffer an, um welchen es sich handelt (/1 = Z6 bei PIO-Signalen; Z8 bei UART-Signalen; /2 = Z7 bei PIO-Signalen; Z9 bei UART-Signalen).
Die Funktion der Stifte a4 und c4 ist nicht festgelegt. Je nach Anwendung

Tabelle 1: Die Baud-Rate der UARTs ergibt sich aus Taktfrequenz und programmiertem Faktor

Brücke	Faktor	Baud-Rate
1	1	nicht benutzen
1	16	9600 Baud
1	64	2400 Baud
2	1	nicht benutzen
2	16	4800 Baud
2	64	1200 Baud
3	1	nicht benutzen
3	16	2400 Baud
3	64	600 Baud
4	1	nicht benutzen
4	16	1200 Baud
4	64	300 Baud
5	1	9600 Baud
5	16	600 Baud
5	64	150 Baud

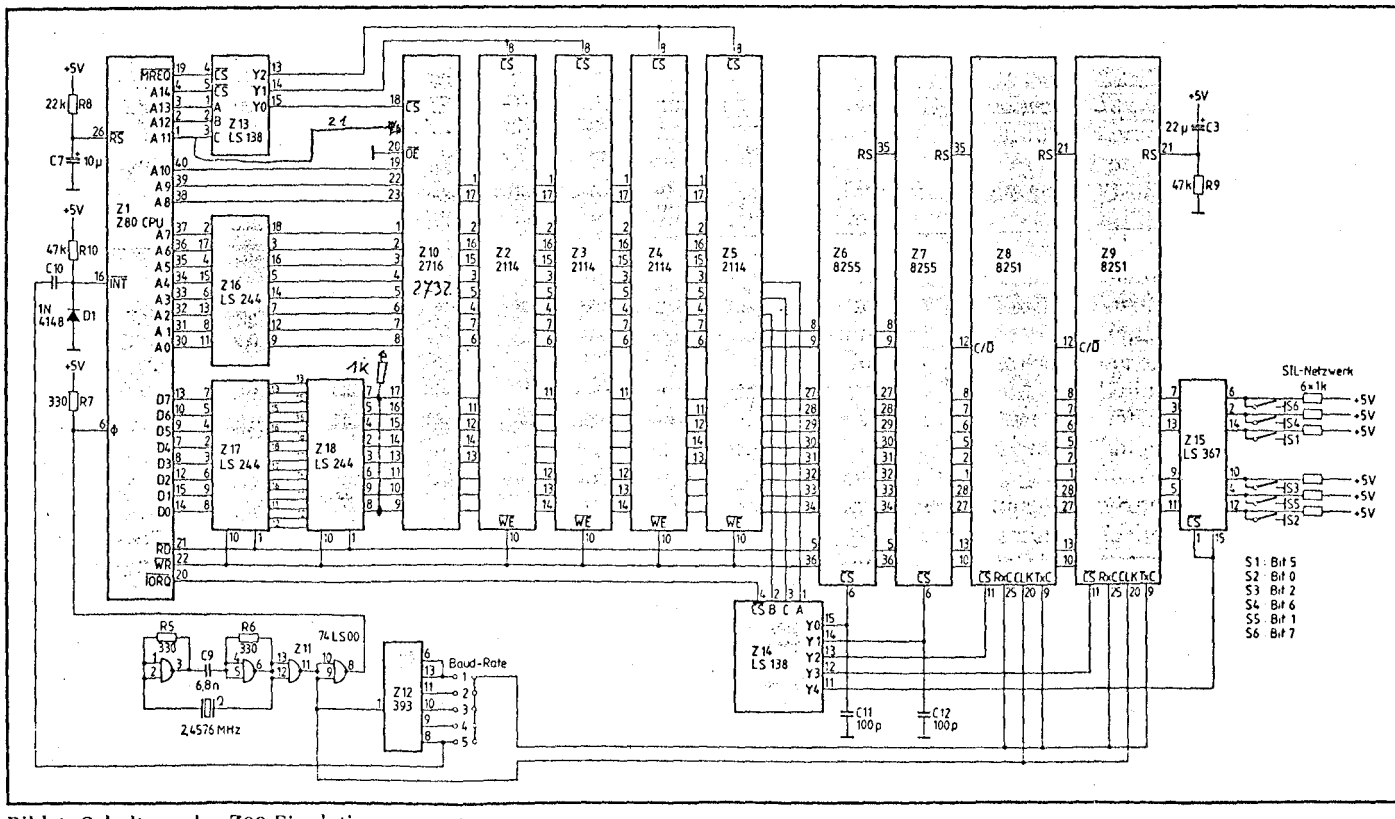


Bild 1. Schaltung des Z80-Einplatinencomputers

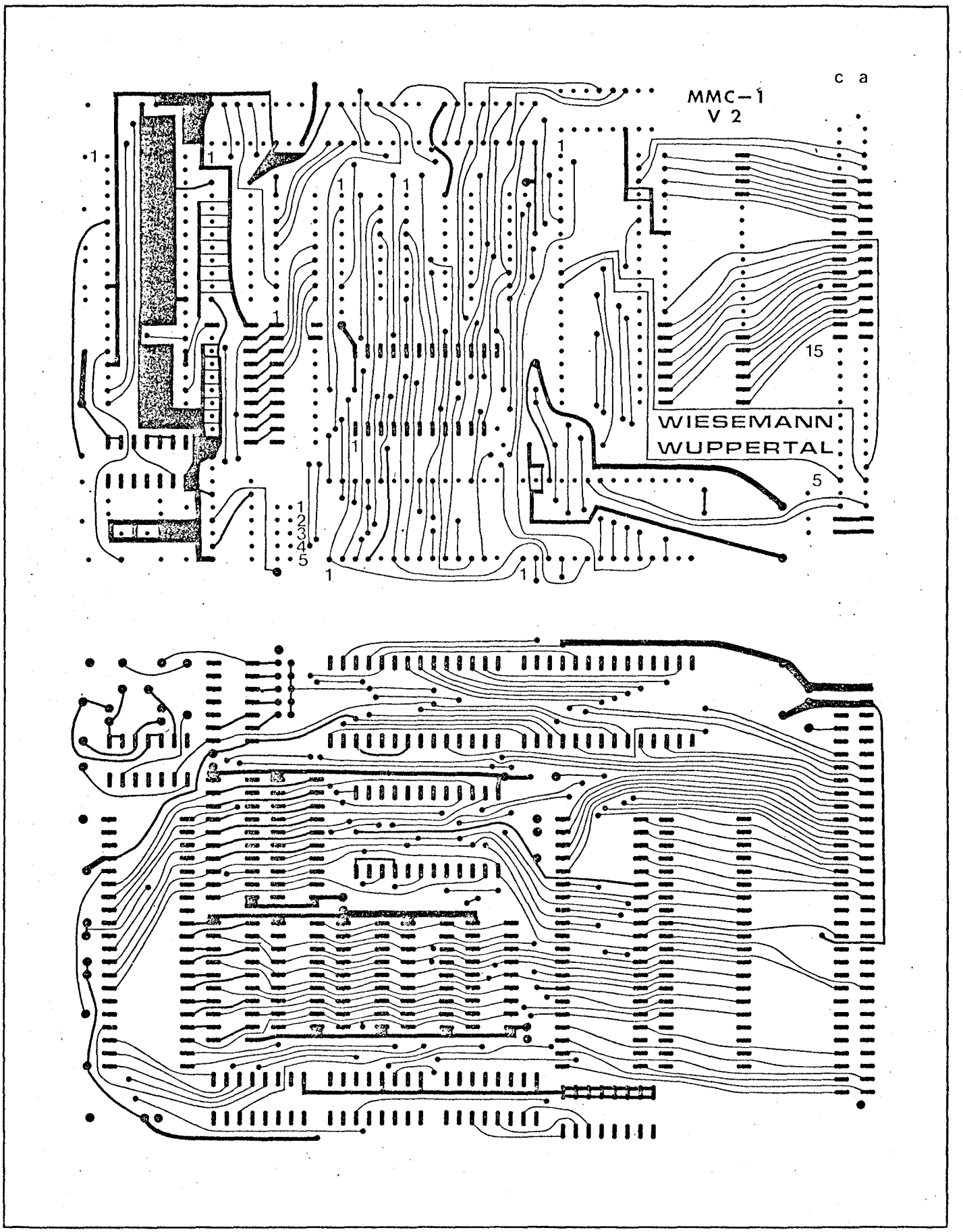


Bild 2. Platinenlayout

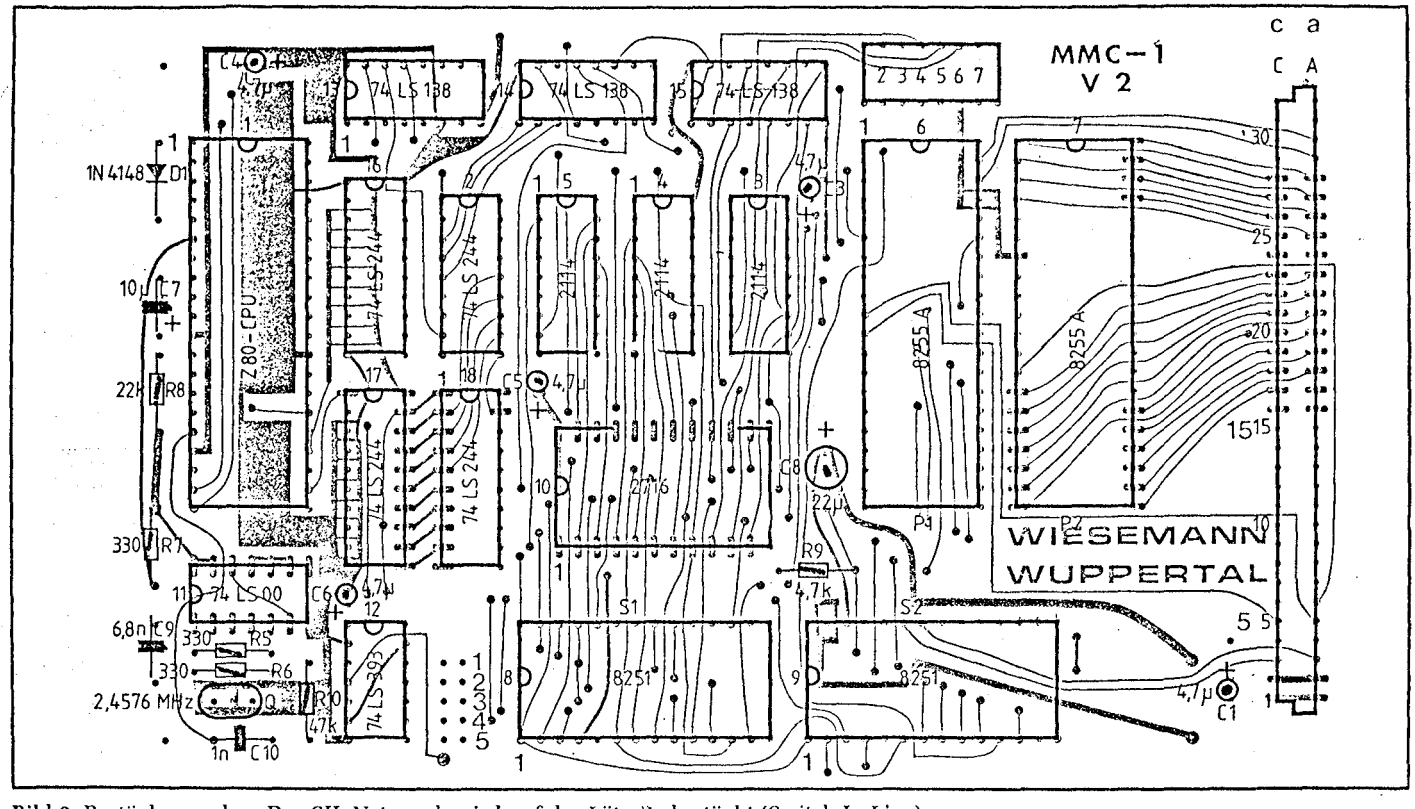


Bild 3. Bestückungsplan. Das SIL-Netzwerk wird auf der Lötseite bestückt (Switch-In-Line)

können hier z. B. die Signale $\overline{DSR1}$ und $\overline{RTS1}$ oder die $\overline{INT}$ und Takt-Leitung angeschlossen werden, indem eine Brücke von den entsprechenden Punkten an diesen Anschluß gelegt wird:

$\overline{DSR1}$ = Stift 22 von Z8. Dieser Anschluß ist auf der Lötseite der Platine mit Stift 26 (+5 V) verbunden. Soll dieser Stift als Eingang benutzt werden, so ist diese Verbindung aufzutrennen.

$\overline{RTS1}$ = Stift 23 von Z8. Dieser Anschluß liegt auf der Platine frei, so daß er direkt benutzt werden kann.

$\overline{INT}$ = Stift 16 der CPU – liegt über R10 als Pull-up-Widerstand an +5 V. Dieser Eingang kann direkt benutzt werden. Als Schutz vor negativen Eingangsspannungen dient D1.

Takt = Stift 1 von Z12 (2,4576 MHz).

Hinweise zur Programmierung

Zur Programmierung des MMC-1 sind die Datenblätter der Bausteine 8251 und 8255 (z. B. im Intel-Datenbuch) sowie das Z80-Manual (Zilog) notwendig. Nach dem Anlegen der Versorgungsspannung werden µP und I/O-Bausteine durch getrennte RC-Glieder zurückgesetzt. Der µP legt Adresse 0 an den Adreßbus und spricht damit die erste EPROM-Adresse an. Die Parallelports der I/O-Bausteine 8255 werden in Mode 0 (Input) geschaltet. Um zu vermeiden, daß der µP (wenn seine Reset-Zeit kürzer als die der I/O-Bausteine ist) zu früh die I/O-Bausteine anspricht, sollte am Anfang des Programms eine Verzögerungsschleife vorgesehen werden. Zum Beispiel:

```
0000          LD    A, 08H
0002   V1     LD    C, 0FFH
0004   V2     LD    B, 0FFH
0006   V3     DJNZ  V3
0008          DEC   C
0009          JR    NZ, V2
000B          DEC   A
000C          JR    NZ, V1
```

Bei Verwendung von Interrupts sollte zuvor der Stackpointer initialisiert sein. Die Lage des Stacks im RAM ist beliebig; beachten Sie, daß er in Richtung auf niedrigere Adressen wächst. Die Platine erlaubt nur den Betrieb im Interrupt-Mode 1 des Z80. In dieser Betriebsart führt die CPU nach Erhalt eines Interrupt einen RST-Befehl nach Adresse 038H aus. An dieser Stelle kann das Interrupt-Unterprogramm programmiert werden. Zur Rückkehr aus dem Interrupt-Programm reicht ein RET-Befehl aus (RETI ist nicht notwendig).
Die Verwendung der $\overline{INT}$-Leitung des Z80 ist beliebig (ein Pull-up-Widerstand befindet sich auf der Platine, so daß eine externe Schaltung z. B. mit Open-Collec-

Tabelle 2: Ein-/Ausgabe-Bereich der PIOs und UARTs

15H	8251/2	Control	(8251/2 = Z9 auf MMC-Platine)
14H	8251/2	Data	
13H	8255/2	Control	(8255/2 = Z7 auf MMC-Platine)
12H	8255/2	Port C	
11H	8255/2	Port B	
10H	8255/2	Port A	
8H	74LS367	(Z15)	(DIL-Schalter 1...6) – diese Adresse nur lesen
5H	8251/1	Control	(8251/1 = Z8 auf MMC-Platine)
4H	8251/1	Data	
3H	8255/1	Control	(8255/1 = Z6 auf MMC-Platine)
2H	8255/1	Port C	
1H	8255/1	Port B	
0H	8255/1	Port A	

tor-Ausgang angeschlossen werden kann) – diese Leitung kann über eine kurze Brücke an einen der freien Stifte auf der VG64-Leiste gelegt werden. Wird im Programm eine Zeitbasis benötigt, so kann durch Einbau von C10 ein Interrupt-Signal alle 256 Taktzyklen erzeugt werden (alle 0,1042 ms).

Programmbeispiele

Einige Unterprogramme können in vielen Anwendungen des MMC-1 unverändert übernommen werden. Alle nachfolgenden Unterprogramme (*Bilder 4* und 5) kehren mit gesetztem Z-Flag zurück, wenn die entsprechende Ein- oder Ausgabe nicht möglich war (Eingabe: Es ist kein Zeichen empfangen worden; Ausgabe: Das zuvor gesendete Zeichen ist von der Peripherieschaltung noch nicht abgerufen worden). Auszugebende Zeichen müssen in Register C liegen, empfangene Zeichen gelangen in Register C.

Tabelle 3: Signale an der 64poligen VG-Leiste des MMC-1

	Reihe a	Reihe c
1	GND	GND
2	+5 V	+5 V
3	DSR2	RTS2
4	sh. Text	sh. Text
5	PC7/1	PC6/1
6	PC4/2	DTR2
7	DTR1	CTS1
8	CTS2	T × D 2
9	T × D 1	R × D 2
10	R × D 1	PB2/1
11	PB1/1	PB0/1
12	PC3/1	PC2/1
13	PC0/1	PC4/1
14	PC5/1	PC1/1
15	PB3/1	PB4/1
16	PB5/1	PB6/1
17	PB7/1	PB3/2
18	PB4/2	PB5/2
19	PB6/2	PB7/2
20	PB2/2	PC5/2
21	PB1/2	PB0/2
22	PC3/2	PC2/2
23	PC1/2	PC0/2
24	PC6/2	PC7/2
25	PA7/2	PA0/2
26	PA6/2	PA1/2
27	PA5/2	PA2/2
28	PA4/2	PA3/2
29	PA6/1	PA7/1
30	PA5/1	PA0/1
31	PA1/1	PA4/1
32	PA2/1	PA3/1

Die Programme verändern den Inhalt der Register A und F. Vor Aufruf muß der entsprechende Baustein für diese Betriebsart vorbereitet – d. h. ein PIO-Baustein muß zuvor den entsprechenden Betriebsartbefehl, ein UART-Baustein den Betriebsartbefehl und ein Kommando erhalten haben.

```
;Eingabe mit Strobe über Port A in Z6
PA1IN     IN      A,(PC1)
          BIT     5,A          ;Bit 5 = 1, wenn Daten
                               ;empfangen wurden
          RET     Z            ;Z-Flag, wenn keine Daten
          LD      C,A          ;Sonst Rückkehr mit Daten
          RET                  ;in Register C

;Ausgabe mit Strobe über Port B in Z6
PB1OUT    IN      A,(PC1)
          BIT     1,A          ;Bit 1 = 1, wenn Daten
                               ;gesendet werden dürfen
          RET     Z            ;Wenn nicht, zurück mit
                               ;Z-Flag
          LD      A,C          ;Sonst Daten aus Register
                               ;C ausgeben
          OUT     (PB1), A
          RET
```

Bild 4. Unterprogramme für die parallele Ein-/Ausgabe

```
;Serielle Eingabe über Z8
S1IN      IN      A,(SC1)
          BIT     1,A          ;Bit 1 = 1, wenn Daten
                               ;empfangen wurden
          RET     Z            ;Wenn nicht, zurück mit
                               ;Z-Flag
          IN      A,(SD1)      ;Sonst Daten einlesen
          RET

;Serielle Ausgabe über Z8
S1OUT     IN      A,(SC1)      ;Bit 0 = 1, wenn Daten
          BIT     0,A          ;ausgegeben werden
                               ;können
          RET     Z            ;Wenn nicht, zurück mit
                               ;Z-Flag
          LD      A,C          ;Sonst Daten aus Register
          OUT     (SC1), A     ;C ausgeben
          RET
```

Bild 5. Unterprogramme für die serielle Ein-/Ausgabe

Ist Ihre EMUF-Anwendung „top secret"?

Der in mc 1981, Heft 2, beschriebene „Einplatinen-Mikrocomputer für universelle Festprogramm-Anwendung" (EMUF) eignet sich für tausend Dinge – oder sogar noch mehr. Vielleicht realisieren Sie mit der EMUF-Platine gerade eine allgemein interessierende Anwendung? Rufen Sie uns an oder schreiben Sie uns! Wir freuen uns über jede neue EMUF-Idee, und vielleicht sind andere Leser ganz gierig darauf, Ihre Anwendung kennenzulernen. Oder ist Ihre EMUF-Applikation etwa „top secret"? *Ihre mc-Redaktion*

A N H A N G *** P ***

**** Ein-Ausgabe-Paket ***

ZEAF Z80 Assembler - Source Listing

```
3EF5 0100      0010 IOPACK EQU  #0100
               0020 ;*****************************
               0030 ;      Ein/Ausgabe Modul
               0040 ;       Abschlussarbeit
               0050 ;  M. Boretius P. Zecherle
               0060 ;
               0070 ; Funktion : Verschiedene Aus-
               0080 ;    gangstreiber und leistungs-
               0090 ;    starken Zeileneditor
               0100 ;
               0110 ; Autor: Peter Zecherle
               0120 ;*****************************
0100           0130        ORG  IOPACK
               0140 ;
               0150 ; SPRUNGVERTEILER
               0160 ;
0100 C39401    0170        JP   INITD    ;Init ports
0103 C33C01    0180        JP   XOTSTR   ;Print String
0106 C38901    0190        JP   OTCHR    ;Print A
0109 C34F01    0200        JP   XOTCHR   ;Print chr.
010C C34201    0210        JP   OTSTR    ;Print String
010F C30402    0220        JP   TDEL     ;Time delay
0112 C37301    0230        JP   INCHR    ;Get chr.
0115 C34101    0240        JP   NOP      ; INDAT
0118 C37C01    0250        JP   SCAN     ;Scan Console
011B C3D001    0260        JP   MOD      ;Modify Bytes
011E C3A501    0270        JP   XFUNCT   ;Distributor
0121 C30A02    0280        JP   OUTA     ;Hex -ASCII
0124 C30502    0290        JP   OUTHL    ;Hex -ASCII
0127 C32002    0300        JP   EDIT     ;Line editor
012A C33E03    0310        JP   MSKTS    ;Mask test
012D C34101    0320        JP   NOP      ; REDIT
0130 C35303    0330        JP   DEZHL    ;Dec -ASCII
0133 C35303    0340        JP   DEZA     ;Dec -ASCII
0136 C35303    0350        JP   BIN      ;Binary -ASCII
0139 C34101    0360        JP   NOP
               0370 ;*********************
               0380 ;* KONSTANTEN
               0390 ;*********************
013C 0000      0400 END    EQU  00
013C 0001      0410 IND    EQU  01
013C 0002      0420 REP    EQU  02
013C 000A      0430 LF     EQU  #0A
013C 001B      0440 ESC    EQU  #1B
013C 000D      0450 CR     EQU  #0D
               0460 ; Cursor keys
013C 0010      0470 CUR    EQU  #10
013C 0008      0480 CUL    EQU  #08
013C 000C      0490 CUU    EQU  #0C ;ESC CUU
013C 000B      0500 CUD    EQU  #0B ;ESC CUD
               0510 ; Keys
013C 0012      0520 HOME   EQU  #12 ;ESC HOME
013C 0013      0530 CL     EQU  #13 ;ESC CL
013C 000E      0540 INS    EQU  #0E ;Controll I
013C 0004      0550 DEL    EQU  #04 ;Controll D
013C 007F      0560 BACKS  EQU  #7F ;Backspace
               0570 ;
```

```
                        **** Ein-Ausgabe-Paket ***
                  0580 ; Portadr.
013C 0005         0590 CONST  EQU  #05 ;8251 STATUSREGISTER
013C 0015         0600 CONST2 EQU  #15 ;8251 STATUSREGISTER
013C 0004         0610 CONIN  EQU  #04 ;8251 DATENREGISTER
013C 0014         0620 CONIN2 EQU  #14 ;8251 DATENREGISTER
013C 0004         0630 CONOUT EQU  #04
                  0640 ;
                  0650 ; Memory def.
013C 2000         0660 RAM2S  EQU  #2000
013C 23FF         0670 RAM2E  EQU  #23FF
013C 0080         0680 STACK  EQU  #80 ;from 237F to 23FF
013C 237F         0690 RAMTOP EQU  RAM2E-STACK
013C 2002         0700 TST    EQU  RAM2S+#02
013C 2004         0710 TZ     EQU  RAM2S+#04
013C 2006         0720 TEND   EQU  RAM2S+#06
013C 0400         0730 MON    EQU  IOPACK+#300
                  0740 ;
                  0750 ;P R O R A M M   S T A R T
                  0760 ;
013C 013C         0770 XOTSTR EQU  $
                  0780 ; PRINT STRING AFTER CALL
                  0790 ;
                  0800 ; ENT: NON
                  0810 ; RET: NON
                  0820 ; CALLED: OTSTR
                  0830 ;
013C E3           0840        EX   (SP),HL
013D CD4201       0850        CALL OTSTR
0140 E3           0860        EX   (SP),HL
0141 C9           0870 NOP    RET
                  0880 ;
0142 0142         0890 OTSTR  EQU  $
                  0900 ; PRINT STRING
                  0910 ; ENT: HL=START OF STRING
                  0920 ; RET: HL=END+1
                  0930 ; CALLED: OTCAT
                  0940 ;
0142 F5           0950        PUSH AF
0143 7E           0960 OTS1   LD   A,(HL)
0144 B7           0970        OR   A          ;STRING END
0145 2805         0980        JR   Z,OTSR
                  0990 ;
0147 CD5501       1000        CALL OTCAT
014A 18F7         1010        JR   OTS1       ;NEXT CHR.
014C 23           1020 OTSR   INC  HL
014D F1           1030        POP  AF
014E C9           1040        RET
014F 014F         1050 XOTCHR EQU  $
                  1060 ; PRINT ONE CHR. AFTER CALL
                  1070 ; ENT: NON
                  1080 ; RET: NON
                  1090 ; CALLED: OTCAT
                  1100 ;
014F E3           1110        EX   (SP),HL
0150 CD5501       1120        CALL OTCAT
0153 E3           1130        EX   (SP),HL
0154 C9           1140        RET
```

```
                    **** Ein-Ausgabe-Paket ***
                1150 ;
0155 0155       1160 OTCAT  EQU  $
                1170 ; ENT: (HL)=PRINTED CONTROLL OR CHR
                1180 ; CALLED: OTCHR,XFUNCT
                1190 ;
0155 D5         1200        PUSH DE
0156 C5         1210        PUSH BC
0157 F5         1220        PUSH AF
0158 7E         1230        LD   A,(HL)
0159 23         1240        INC  HL
015A FE20       1250        CP   #20    ;STEUERZEICHEN?
015C 300E       1260        JR   NC,OTCA1 ;YES
015E CDA501     1270        CALL XFUNCT
0161 01         1280        DEFB IND    ;Index Adr. folgt
0162 ED01       1290        DEFW INDSTR
0164 02         1300        DEFB REP    ;Repetiere Zeichen
0165 F801       1310        DEFW REPEAT
0167 00         1320        DEFB END
0168 8901       1330        DEFW OTCHR
016A 1803       1340        JR   OTCAE
016C CD8901     1350 OTCA1  CALL OTCHR      CD B4 01
016F F1         1360 OTCAE  POP  AF       ;FETCH REG
0170 C1         1370        POP  BC
0171 D1         1380        POP  DE
0172 C9         1390        RET
                1400 ;
                1410 ; ROUTINES BETWEEN HARD AND SOFTWARE
                1420 ;
                1430 ; ***INCHR***
0173 0173       1440 INCHR  EQU  $
                1450 ; Function:  Console input
                1460 ;   waiting by Key is pressed
                1470 ; ENT: NON
                1480 ; RET: A=pressed Key code
                1490 ; CALLED: NOTHING
                1500 ;
0173 DB05       1510        IN   A,(CONST)  ;GET STATUS
0175 E602       1520        AND  02
0177 28FA       1530        JR   Z,INCHR
0179 DB04       1540        IN   A,(CONIN)
017B C9         1550        RET
                1560 ;
                1570 ;***SCAN***
                1580 ; Function: Scan console by Key pressed
                1590 ; RET: Carry set if new Key pressed
                1600 ;      A=Key code
                1610 ; CALLED: NOTHING
                1620 ;
017C DB05       1630 SCAN   IN   A,(CONST)
017E E602       1640        AND  02
0180 2803       1650        JR   Z,NOKEY    ;No Key pressed
0182 DB04       1660        IN   A,(CONIN)
0184 C9         1670        RET
0185 37         1680 NOKEY  SCF
0186 3E00       1690        LD   A,00       ;for CP/M accu=00
0188 C9         1700        RET
```

**** Ein-Ausgabe-Paket ***

```
                  1710 ;
0189 0189         1720 OTCHR  EQU  $
                  1730 ; OUTPUT ON  CONSOLE
                  1740 ; ENT: A=CHR.
                  1750 ; CALLED: NOTHING
                  1760 ;
                  1770 ; Label CO is entery for CP/M systems
                  1780 ;
0189 4F           1790        LD   C,A
018A DB05         1800 CO     IN   A,(CONST) ;Get Status
018C E601         1810        AND  01
018E 28FA         1820        JR   Z,CO      ;waiting of ready
0190 79           1830        LD   A,C
0191 D304         1840        OUT  (CONOUT),A
0193 C9           1850        RET
                  1860 ;
0194 0194         1870 INITD  EQU  $
                  1880 ;INIT. 8251 BAUSTEIN
                  1890 ;
                  1900 ; MODE INSTRUCTION
                  1910 ;   2 Stop bits
                  1920 ;   parity disabled
                  1930 ;   8 bit characters
                  1940 ;   Baud rate factor 16
                  1950 ;
                  1960 ; COMMAND INSTRUCTION
                  1970 ;
                  1980 ;   No hunt mode
                  1990 ;   Not (RTS) forced to 0
                  2000 ;   Receive enabeld
                  2010 ;   Data terminal ready
                  2020 ;   Transmit enabeld
                  2030 ;
                  2040 ; First SIO   LDA 740 OUT (CONST),A & Reset
0194 3ECE         2050        LD   A,#CE    ;LOAD MODE
0196 D305         2060        OUT  (CONST),A
0198 3E27         2070        LD   A,#27    ;LOAD COMMAND
019A D305         2080        OUT  (CONST),A
                  2090 ; Second SIO
                  2100 ;   Baudrate factor 64
                  2110 ;
019C 3ECF         2120        LD   A,#CF    ;LOAD MODE
019E D315         2130        OUT  (CONST2),A
01A0 3E27         2140        LD   A,#27    ;LOAD COMMAND
01A2 D315         2150        OUT  (CONST2),A
                  2160 ;
01A4 C9           2170        RET
```

```
                    **** Ein-Ausgabe-Paket ***
                2180 ;
                2190 ; GENERAL FUNCTION  ROUTINES
                2200 ;
01A5 01A5       2210 XFUNCT EQU  $
                2220 ; EXECUTE FUNCTION
                2230 ; ENT: A=FUNCT. CODE
                2240 ;      (SP)=FUNCT. CODE
                2250 ;
                2260 ;LIST:X=FIRST FUNC. CODE
                2270 ;      XX=FIRST FUNC. ADR.
                2280 ;      X=NEXT ....
                2290 ;      .
                2300 ;      00 END BYTE
                2310 ;      XX DEFAULT ADR.
                2320 ;
                2330 ;      RET:(SP)=LIST END+1
                2340 ;
01A5 E5         2350         PUSH HL      ;Save Reg.
01A6 D5         2360         PUSH DE
01A7 F5         2370         PUSH AF
01A8 210600     2380         LD   HL,06
01AB 39         2390         ADD  HL,SP   ;(HL)=LIST START
01AC 5E         2400         LD   E,(HL)
01AD 23         2410         INC  HL
01AE 56         2420         LD   D,(HL)
01AF D5         2430         PUSH DE
                2440 ;                     FINDE LIST END
01B0 1A         2450 FUN1    LD   A,(DE)   ;TEST OF
01B1 B7         2460         OR   A        ;END?
01B2 13         2470         INC  DE
01B3 13         2480         INC  DE
01B4 13         2490         INC  DE
01B5 20F9       2500         JR   NZ,FUN1
                2510 ;                     PUT END+1 INTO STACK
01B7 72         2520         LD   (HL),D
01B8 2B         2530         DEC  HL
01B9 73         2540         LD   (HL),E
01BA E1         2550         POP  HL      ;GET LIST START
01BB D1         2560         POP  DE      ;D=FUNCTION
01BC 7E         2570 FUN2    LD   A,(HL)
01BD 23         2580         INC  HL
01BE B7         2590         OR   A       ;DEFAULT?
01BF 2807       2600         JR   Z,FUN3
01C1 BA         2610         CP   D       ;FUNCTION FOUND
01C2 2804       2620         JR   Z,FUN3
01C4 23         2630         INC  HL
01C5 23         2640         INC  HL
01C6 18F4       2650         JR   FUN2
01C8 D5         2660 FUN3    PUSH DE
01C9 F1         2670         POP  AF      ;A=FUNCTION
01CA D1         2680         POP  DE
01CB CDE601     2690         CALL CADHL   ;GET FUNCTIONS START
01CE E3         2700         EX   (SP),HL
01CF C9         2710         RET  ;JUMP TO FUNCTION
```

**** Ein-Ausgabe-Paket ***

```
                  2720 ;
01D0 01D0         2730 MOD     EQU  $
                  2740 ; REPLACE BYTE
                  2750 ;
                  2760 ; ENT: A=BYTE
                  2770 ;      HL=LISTE POINTER
                  2780 ;
                  2790 ; LISTE: FIRST OLD BYTE
                  2800 ;        FIRST NEW BYTE
                  2810 ;        NEXT
                  2820 ;        .......
                  2830 ;        00 END
                  2840 ;
                  2850 ; RET:A=NEW BYTE IF REPLACED ELSE OLD BYTE
                  2860 ;     HL=END+1
                  2870 ;
01D0 C5           2880          PUSH BC      ;SAVE IT
01D1 4F           2890          LD   C,A     ;BYTE INPUT
01D2 7E           2900 MOD1     LD   A,(HL)  ;GET LIST BYTE
01D3 23           2910          INC  HL
01D4 B7           2920          OR   A       ;LIST END
01D5 280C         2930          JR   Z,MODR  ;YES
01D7 23           2940          INC  HL
01D8 B9           2950          CP   C       ;FOUND?
01D9 20F7         2960          JR   NZ,MOD1 ;NOT FOUND
01DB AF           2970          XOR  A       ;A=0
01DC 2B           2980          DEC  HL
01DD 4E           2990          LD   C,(HL)  ;NEW BYTE
                  3000 ;                       MOVE HL TO END
01DE 23           3010 MOD2     INC  HL
01DF BE           3020          CP   (HL)    ;END?
01E0 23           3030          INC  HL
01E1 20FB         3040          JR   NZ,MOD2
01E3 79           3050 MODR     LD   A,C
01E4 C1           3060          POP  BC
01E5 C9           3070          RET
                  3080 ;
                  3090 ;
                  3100 ;
01E6 01E6         3110 CADHL    EQU  $ ;FUNCTION LD HL,(HL)
01E6 F5           3120          PUSH AF
01E7 7E           3130          LD   A,(HL)     ;GET LSB
01E8 23           3140          INC  HL
01E9 66           3150          LD   H,(HL)     ;GET MSB
01EA 6F           3160          LD   L,A
01EB F1           3170          POP  AF
01EC C9           3180          RET
                  3190 ;
```

**** Ein-Ausgabe-Paket ***

```
                3200 ;
                3210 ;
01ED E5         3220 INDSTR PUSH HL      ;Load Pointer with ind
01EE CDE601     3230        CALL CADHL   ;Adresse and transmit
01F1 CD4201     3240        CALL OTSTR   ;String
01F4 E1         3250        POP  HL
01F5 23         3260        INC  HL
01F6 23         3270        INC  HL      ;set to next chr. byte
01F7 C9         3280        RET
                3290 ;
                3300 ;
                3310 ;
                3320 ;
01F8 46         3330 REPEAT LD   B,(HL) ;repeat byte
01F9 23         3340        INC  HL    ;ENT: (SP) counter byte
01FA 54         3350        LD   D,H   ;     next chr. byte
01FB 5D         3360        LD   E,L
01FC 62         3370 REP1   LD   H,D
01FD 6B         3380        LD   L,E
01FE CD5501     3390        CALL OTCAT
0201 10F9       3400        DJNZ REP1
0203 C9         3410        RET
                3420 ;
0204 0204       3430 TDEL   EQU  $
                3440 ; ENT : (SP) Delay time (ms )
                3450 ; RET :
                3460 ; CALLED : NOTHING
                3470 ;
0204 C9         3480        RET
                3490 ;
                3500 ; Hex ASCII converting
                3510 ;
0205 7C         3520 OUTHL  LD   A,H     ;Print HL in ASCII
0206 CD0A02     3530        CALL OUTA   ;4 chr.
0209 7D         3540        LD   A,L
020A F5         3550 OUTA   PUSH AF     ;Print A in ASCII
020B 07         3560        RLCA
020C 07         3570        RLCA
020D 07         3580        RLCA
020E 07         3590        RLCA
020F CD1302     3600        CALL CHAR   ;Nibbel  by nibbel
0212 F1         3610        POP  AF
0213 E60F       3620 CHAR   AND  #0F      ;cut higher nibbel
0215 FE0A       3630        CP   #0A      ;>9?
0217 3802       3640        JR   C,CONV   ;
0219 C607       3650        ADD  A,07     ;
021B C630       3660 CONV   ADD  A,#30    ;in ASCII wandeln
021D C38901     3670        JP   OTCHR
```

```
                    **** Ein-Ausgabe-Paket ***
                3680 ;
0220 0220       3690 EDIT   EQU  $
                3700 ; Extented Line Editor
                3710 ; ENT: HL=Textbuffer start adr.
                3720 ; RET: HL=Textbuffer end adr.
                3730 ;
                3740 ; CALLED:OTCHR,XOTCHR,XFUNCT
                3750 ;
0220 220220     3760        LD   (TST),HL ;Init. Pointer
0223 220620     3770        LD   (TEND),HL
0226 220420     3780        LD   (TZ),HL
0229 3600       3790        LD   (HL),0
                3800 ;
022B CD7301     3810 E1     CALL INCHR    ;Get chr.
022E FE20       3815        CP   #20
0230 3845       3820        JR   C,E2
                3830 ;
0232 2A0420     3840        LD   HL,(TZ)  ;Normal key
0235 77         3850        LD   (HL),A
0236 F5         3860        PUSH AF
0237 CD4802     3870        CALL E6       ;expand text line?
023A 2003       3880        JR   NZ,E5     ;no, because not line end
023C CD6D02     3890        CALL INCTE
023F CD5302     3900 E5     CALL INCTZ
0242 F1         3910        POP  AF
0243 CD8901     3920        CALL OTCHR    ;Print chr.
0246 18E3       3930        JR   E1
                3940 ;
0248 2A0620     3950 E6     LD   HL,(TEND)  ;(TEND)=(TZ) ??
024B ED5B0420   3960        LD   DE,(TZ)    ;Cursor on line
024F B7         3970        OR   A          ;end ?
0250 ED52       3980        SBC  HL,DE      ;test it and set flages
0252 C9         3990        RET
                4000 ;
0253 2A0420     4010 INCTZ  LD   HL,(TZ) ; INC (TZ)
0256 23         4020        INC  HL
0257 220420     4030        LD   (TZ),HL
025A C9         4040        RET
                4050 ;
025B 2A0420     4060 DECTZ  LD   HL,(TZ)   ; DEC (TZ)
025E 2B         4070        DEC  HL
025F 220420     4080        LD   (TZ),HL
0262 C9         4090        RET
                4100 ;
0263 2A0620     4110 DECTE  LD   HL,(TEND) ;INC (TEND)
0266 2B         4120        DEC  HL
0267 220620     4130        LD   (TEND),HL
026A 3600       4140        LD   (HL),00
026C C9         4150        RET
                4160 ;
026D 2A0620     4170 INCTE  LD   HL,(TEND) ;DEC (TEND)
0270 23         4180        INC  HL
0271 220620     4190        LD   (TEND),HL
0274 3600       4200        LD   (HL),00
0276 C9         4210        RET
                4220 ;
```

**** Ein-Ausgabe-Paket ***

```
0277 CDA501    4230 E2      CALL XFUNCT  ;Control code distributor
027A 1B        4240         DEFB ESC
027B 9702      4250         DEFW EXESC
027D 13        4260         DEFB CL
027E B702      4270         DEFW CLEARL
0280 10        4280         DEFB CUR
0281 9802      4290         DEFW RIGHT
0283 08        4300         DEFB CUL
0284 A402      4310         DEFW LEFT
0286 0E        4320         DEFB INS
0287 F302      4330         DEFW INSERT
0289 04        4340         DEFB DEL
028A 2503      4350         DEFW DELETE
028C 7F        4360         DEFB BACKS
028D D302      4370         DEFW BACK
028F 0D        4380         DEFB #0D
0290 EE02      4390         DEFW QUIT
0292 00        4400         DEFB 00
0293 A302      4410         DEFW E3
0295 1894      4420         JR   E1
               4430 ;
0297 C9        4440 EXESC   RET  ;Free for optiones
               4450 ;
0298 CD4802    4460 RIGHT   CALL E6     ;Cursor right instruction
029B C8        4470         RET  Z
029C CD5302    4480         CALL INCTZ
029F CD4F01    4490         CALL XOTCHR
02A2 10        4500         DEFB CUR
02A3 C9        4510 E3      RET
               4520 ;
02A4 2A0420    4530 LEFT    LD   HL,(TZ)  ;Cursor left instruction
02A7 ED5B0220  4540         LD   DE,(TST)
02AB AF        4550         XOR  A        ;Clear Carry
02AC ED52      4560         SBC  HL,DE    ;on left end?
02AE C8        4570         RET  Z        ;yes
02AF CD5B02    4580         CALL DECTZ
02B2 CD4F01    4590         CALL XOTCHR
02B5 08        4600         DEFB CUL
02B6 C9        4610         RET
               4620 ;
               4630 ;
02B7 02B7      4640 CLEARL  EQU  $    ;Clear line instruction
02B7 2A0620    4650         LD   HL,(TEND)
02BA ED5B0220  4660         LD   DE,(TST)
02BE AF        4670         XOR  A
02BF ED52      4680         SBC  HL,DE
02C1 C8        4690         RET  Z      ;Buffer is clear
02C2 23        4700 E4      INC  HL
02C3 CDD302    4710         CALL BACK ;Clear one chr. on screen
02C6 7C        4720         LD   A,H
02C7 B5        4730         OR   L       ;all clear ?
02C8 20F8      4740         JR   NZ,E4  ;no then clear more chr.
02CA 2A0220    4750         LD   HL,(TST)
02CD 220620    4760         LD   (TEND),HL
02D0 3600      4770         LD   (HL),#00 ;Clear buffer
02D2 C9        4780         RET
```

**** Ein-Ausgabe-Paket ***

```
                 4790 ;
02D3 02D3        4800 BACK   EQU  $        ;Backspace instruction
02D3 2A0420      4810        LD   HL,(TZ)
02D6 ED5B0220    4820        LD   DE,(TST)
02DA AF          4830        XOR  A         ;Clear Carry
02DB ED52        4840        SBC  HL,DE     ;Cursor on left Side
02DD C8          4850        RET  Z         ;Yes then ret
02DE CD5B02      4860        CALL DECTZ
02E1 2A0420      4870        LD   HL,(TZ)
02E4 3620        4880        LD   (HL),#20
02E6 CD3C01      4890        CALL XOTSTR    ;Clear old chr. and move
02E9 08200800    4900        DEFB CUL,#20,CUL,END  ;cursor back
02ED C9          4910        RET
                 4920 ;
02EE E1          4930 QUIT   POP  HL          ;CR instruction
02EF 2A0620      4940        LD   HL,(TEND) ;Editor end
02F2 C9          4950        RET  ;HL=Textend adr.
                 4960 ;
                 4970 ;
02F3 02F3        4980 INSERT EQU  $     ;Insert instruction
02F3 CD4802      4990        CALL E6       ;is it allow?
02F6 C8          5000        RET  Z        ;no,line end
                 5010 ;                  adjust memorybuffer
02F7 E5          5020        PUSH HL
02F8 C1          5030        POP  BC ;Transfer Bytes
02F9 ED5B0620    5040        LD   DE,(TEND)
02FD D5          5050        PUSH DE
02FE E1          5060        POP  HL
02FF 2B          5070        DEC  HL
0300 EDB8        5080        LDDR
0302 2A0420      5090        LD   HL,(TZ)
0305 3620        5100        LD   (HL),#20  ;Clear last old chr.
0307 CD6D02      5110        CALL INCTE
                 5120 ;                    adjust on screen
030A 2A0420      5130 E8     LD   HL,(TZ)
030D CD4201      5140        CALL OTSTR
0310 CD3C01      5150        CALL XOTSTR    ;Clear last old chr.
0313 200800      5160        DEFB #20,CUL,END
0316 CD4802      5170        CALL E6
0319 7C          5180 E7     LD   A,H  ;Cursor on right end?
031A B5          5190        OR   L
031B C8          5200        RET  Z    ;yes then ret
031C 2B          5210        DEC  HL
031D E5          5220        PUSH HL
031E CD4F01      5230        CALL XOTCHR
0321 08          5240        DEFB CUL
0322 E1          5250        POP  HL
0323 18F4        5260        JR   E7
```

**** Ein-Ausgabe-Paket ***

```
                5270 ;
0325 0325       5280 DELETE EQU  $   ;Delete instruction
0325 CD4802     5290        CALL E6     ;Cursor on line end?
0328 C8         5300        RET  Z      ;yes then ret
                5310 ;
0329 E5         5320        PUSH HL     ;Delete and transfer bytes
032A C1         5330        POP  BC     ;in Textbuffer
032B ED5B0420   5340        LD   DE,(TZ)
032F D5         5350        PUSH DE
0330 E1         5360        POP  HL
0331 23         5370        INC  HL
0332 EDB0       5380        LDIR
0334 CD6302     5390        CALL DECTE
0337 CD0A03     5400        CALL E8   ;Print Text without del. chr.
033A C9         5410        RET
                5420 ;
                5430 ;
033B CDE601     5440 MSKTX  CALL CADHL
033E 033E       5450 MSKTS  EQU  $
                5460 ; Function: Test with mask
                5470 ; ENT: A=BYTE
                5480 ;      HL=MASK LIST
                5490 ; RET: Carry set if not in Mask
                5500 ;
033E C5         5510        PUSH BC
033F 4F         5520        LD   C,A
0340 AF         5530        XOR  A
0341 BE         5540        CP   (HL) ;List end ?
0342 79         5550        LD   A,C
0343 37         5560        SCF
0344 C1         5570        POP  BC
0345 C8         5580        RET  Z
                5590 ;
0346 BE         5600        CP   (HL) ;compare 1. or 3. .... byte
0347 23         5610        INC  HL
0348 23         5620        INC  HL
0349 38F3       5630        JR   C,MSKTS ;if underflow
                5640 ;
034B 2B         5650        DEC  HL
034C BE         5660        CP   (HL) ;compare 2. or 4. .... byte
034D 23         5670        INC  HL
034E C8         5680        RET  Z
                5690 ;
034F 30ED       5700        JR   NC,MSKTS ;if overflow
0351 3F         5710        CCF  ;reset Carry
0352 C9         5720        RET
                5730 ;
                5740 ;
0353 0353       5750 DEZHL  EQU  $
0353 0353       5760 DEZA   EQU  $
0353 0353       5770 BIN    EQU  $
```

**** Ein-Ausgabe-Paket ***

ZEAP Z80 Assembler - Symbol Table

02D3H	4800	BACK	007FH	0560	BACKS
0353H	5770	BIN	01E6H	3110	CADHL
0213H	3620	CHAR	0013H	0530	CL
02B7H	4640	CLEARL	018AH	1800	CO
0004H	0610	CONIN	0014H	0620	CONIN2
0004H	0630	CONOUT	0005H	0590	CONST
0015H	0600	CONST2	021BH	3660	CONV
000DH	0450	CR	000BH	0500	CUD
0008H	0480	CUL	0010H	0470	CUR
000CH	0490	CUU	0263H	4110	DECTE
025BH	4060	DECTZ	0004H	0550	DEL
0325H	5280	DELETE	0353H	5760	DEZA
0353H	5750	DEZHL	022BH	3810	E1
0277H	4230	E2	02A3H	4510	E3
02C2H	4700	E4	023FH	3900	E5
0248H	3950	E6	0319H	5180	E7
030AH	5130	E8	0220H	3690	EDIT
0000H	0400	END	001BH	0440	ESC
0297H	4440	EXESC	01B0H	2450	FUN1
01BCH	2570	FUN2	01C8H	2660	FUN3
0012H	0520	HOME	0173H	1440	INCHR
026DH	4170	INCTE	0253H	4010	INCTZ
0001H	0410	IND	01EDH	3220	INDSTR
0194H	1870	INITD	000EH	0540	INS
02F3H	4980	INSERT	0100H	0010	IOPACK
02A4H	4530	LEFT	000AH	0430	LF
01D0H	2730	MOD	01D2H	2900	MOD1
01DEH	3010	MOD2	01E3H	3050	MODR
0400H	0730	MON	033EH	5450	MSKTS
033BH	5440	MSKTX	0185H	1680	NOKEY
0141H	0870	NOP	016CH	1350	OTCA1
016FH	1360	OTCAE	0155H	1160	OTCAT
0189H	1720	OTCHR	0143H	0960	OTS1
014CH	1020	OTSR	0142H	0890	OTSTR
020AH	3550	OUTA	0205H	3520	OUTHL
02EEH	4930	QUIT	23FFH	0670	RAM2E
2000H	0660	RAM2S	237FH	0690	RAMTOP
0002H	0420	REP	01FCH	3370	REP1
01F8H	3330	REPEAT	0298H	4460	RIGHT
017CH	1630	SCAN	0080H	0680	STACK
0204H	3430	TDEL	2006H	0720	TEND
2002H	0700	TST	2004H	0710	TZ
01A5H	2210	XFUNCT	014FH	1050	XOTCHR
013CH	0770	XOTSTR			

**** Monitorlisting ****

ZEAP Z80 Assembler - Source Listing

```
0400            0010 START  ORG  #0400
                0020 ;*********************************
                0030 ;*      Abschlussarbeit
                0040 ;* M. Boretius P. Zecherle
                0050 ;*
                0060 ;*   MONITOR PRORAMM V1.2
                0070 ;*
                0080 ;* Befehle :
                0090 ;* MEMORY MODIFY   M
                0100 ;* TABULATE        T
                0110 ;* EXECUTE PROG.   G
                0120 ;* OUT TO PORT     O
                0130 ;* IN FROM PORT    I
                0140 ;* USER PROGRAM    U
                0150 ;*
                0160 ;* Autor:  Peter  Zecherle
                0170 ;*********************************
0400 0100       0180 IOPACK EQU  #0100
0400 0112       0190 CI     EQU  IOPACK+#12
0400 0109       0200 XOTCHR EQU  IOPACK+#09
0400 0118       0210 CSTS   EQU  IOPACK+#18
0400 0121       0220 PRAC   EQU  IOPACK+#21
0400 0124       0230 PRHL   EQU  IOPACK+#24    ;OUT REG. HL
0400 0121       0240 OUTA   EQU  PRAC
0400 0124       0250 OUTHL  EQU  PRHL
0400 0100       0260 INIT   EQU  IOPACK+#00
0400 0103       0270 XOTSTR EQU  IOPACK+03
0400 011E       0280 XFUNCT EQU  IOPACK+#1E
0400 0106       0290 OTCHR  EQU  IOPACK+#06
0400 012A       0300 MSKTS  EQU  IOPACK+#2A
                0310 ;
0400 0800       0320 PALASM EQU  #800
                0330 ;******KONSTANTEN******
0400 0000       0340 END    EQU  00
0400 0001       0350 IND    EQU  01
0400 0007       0360 BEL    EQU  07
0400 001B       0370 ESC    EQU  #1B   ;ESCAPE CODE
0400 007F       0380 DEL    EQU  #7F   ;DELETE CODE
                0390 ;
                0400 ; CURSOR TASTEN
                0410 ;
0400 0010       0420 CUR    EQU  #10 ;RECHTS
0400 0008       0430 CUL    EQU  #08 ;LINKS
0400 000C       0440 CUU    EQU  #0C ;ESC CUU /80MODEL30
0400 000B       0450 CUD    EQU  #0B ;ESC CUD /80MODEL30
                0460 ;
```

**** Monitorlisting ****

```
                  0470 ;
0400 185A         0480        JR    MONI   ;MONITOR ANSPRUNG
                  0490 ;
0402 0402         0500 SPACE  EQU   $
                  0510 ; Funktion: Ausgabe von Leerzeichen
                  0520 ;
                  0530 ; ENT: C=Anzahl der Leerzeichen
                  0540 ; RET: C=00
                  0550 ; CALLED: XOTCHR
                  0560 ;
0402 F5           0570        PUSH  AF
0403 CD0901       0580 LOOP2  CALL  XOTCHR   ;Print ein Blank
0406 20           0590        DEFB  "
0407 0D           0600        DEC   C        ;C=Zahler
0408 20F9         0610        JR    NZ,LOOP2
040A F1           0620        POP   AF
040B C9           0630        RET
                  0640 ;
040C 040C         0650 LITBIG EQU   $
                  0660 ; Funktion: Wandle klein in gross Buch-
                  0670 ;           staben um
                  0680 ; ENT: A=Zeichen (klein oder gross)
                  0690 ; RET: A=Gross Buchstabe
                  0700 ; CALLED: NOTHING
                  0710 ;
040C E67F         0720        AND   #7F      ;7Bit ASCII
040E FE60         0730        CP    #60      ;Klein
0410 D8           0740        RET   C        ;Nein
0411 D620         0750        SUB   #20      ;wandeln
0413 C9           0760        RET
                  0770 ;
```

**** Monitorlisting ****

```
                  0780 ;
                  0790 ;EXPR HOLT ZEICHEN VON DER CONSOLE
                  0800 ;UND SPEICHERT DIESE GEPACKT ALS SED. ZAHL
                  0810 ;IM REG. HL. DIE EINGABE IST FORMATFREI.
                  0820 ;DIE LETZTEN 4 ZEICHEN SIND SIGNIFKANT.
                  0830 ;HL ARBEITET ALS 4 ZEICHEN FIFO .
                  0840 ;
0414 210000       0850 EXPR   LD   HL,0000   ;Setze FIFO Null
0417 CD1201       0860 EX0    CALL CI        ;HL= FIFO
041A CD0C04       0870        CALL LITBIG    ;klein =>gross
041D E5           0880        PUSH HL
                  0890 ;
041E 212606       0900        LD   HL,HEXM   ;Teste auf sedezimal
0421 CD2A01       0910        CALL MSKTS     ;Zahl !
0424 E1           0920        POP  HL
0425 38F0         0930        JR   C,EX0     ;keine sed. Zahl
0427 CD0601       0940        CALL OTCHR     ;gultig
                  0950 ;
042A CD3704       0960 EX1 *  CALL NIBBEL    ;4 Bit
042D 3818         0970        JR   C,EX2
042F 29           0980        ADD  HL,HL     ;Schiebe FIFO
0430 29           0990        ADD  HL,HL
0431 29           1000        ADD  HL,HL
0432 29           1010        ADD  HL,HL
0433 B5           1020        OR   L
0434 6F           1030        LD   L,A       ;neuer FIFO code
0435 18E0         1040        JR   EX0
                  1050 ;
0437 D630         1060 NIBBEL SUB  "0
0439 D8           1070        RET  C
043A FE17         1080        CP   #17
043C 3F           1090        CCF
043D D8           1100        RET  C
043E FE0A         1110        CP   #0A
0440 3F           1120        CCF
0441 D0           1130        RET  NC
0442 D607         1140        SUB  #07
0444 FE0A         1150        CP   #0A
0446 C9           1160        RET
                  1170 ;
0447 79           1180 EX2    LD   A,C
0448 FE0D         1190        CP   #0D
044A 2801         1200        JR   Z,ECHU
044C C9           1210        RET
                  1220 ;
044D F5           1230 ECHU   PUSH AF
044E CD0901       1240        CALL XOTCHR
0451 0A           1250        DEFB #0A
0452 F1           1260        POP  AF
0453 4F           1270        LD   C,A
0454 C9           1280        RET
                  1290 ;
0455 CD0301       1300 CRLF   CALL XOTSTR
0458 0A0D00       1310        DEFB #0A,#0D,END
045B C9           1320        RET
```

* Siehe Q11

**** Monitorlisting ****

```
                  1330 ;*************************;
                  1340 ;*************************;
                  1350 ; MONITOR HAUPTPROGRAMM
                  1360 ;*************************;
                  1370 ;*************************;
                  1380 ;
045C 045C         1390 MONI    EQU  $
045C CD0301       1400 MLOOF   CALL XOTSTR    ;Kaltstart Meldung
045F 0A0D         1410         DEFB #0A,#0D
0461 204D4F4E     1420         DEFM ' MONITOR MODE '
     49544F52
     204D4F44
     4520
046F 00           1430         DEFB 00
0470 CD0301       1440 WSTR    CALL XOTSTR    ;Warmstart Meldung
0473 0A0D         1450         DEFB #0A,#0D   ;durch neue Zeile
0475 3E           1460         DEFB ">        ;und Prompt Zeichen
0476 00           1470         DEFB 00
0477 CD1201       1480         CALL CI
                  1490 ; Wandle alles in Grossbuchstaben
047A CD0C04       1500         CALL LITBIG
                  1510 ; General Hauptsprungverteiler
047D CD1E01       1520         CALL XFUNCT
0480 55           1530         DEFM 'U'       ;USER PAL PRORAMMER
0481 A304         1540         DEFW USER
0483 4D           1550         DEFM 'M'       ;MODIFY ?
0484 A305         1560         DEFW MODIFY
0486 47           1570         DEFM 'G'       ;GO ?
0487 8905         1580         DEFW ENTER
0489 54           1590         DEFM 'T'
048A A704         1600         DEFW TABU      ;TABULATE ?
048C 49           1610         DEFM 'I'
048D 9006         1620         DEFW INPO      ;INPUT PORT
048F 4F           1630         DEFM 'O'
0490 6206         1640         DEFW OUTPO     ;OUT TO PORT
0492 00           1650         DEFB 00
0493 9704         1660         DEFW ERROR     ;KEIN BEFEHL
0495 18D9         1670         JR   WSTR      ;WARMSTART
                  1680 ;
0497 CD0301       1690 ERROR   CALL XOTSTR    ;Bringe Fehlermeldung
049A 57484154     1700         DEFM 'WHAT? '
     3F20
04A0 0700         1710         DEFB BEL,END
04A2 C9           1720         RET
                  1730 ;
```

**** Monitorlisting ****

```
                  1740 ;
                  1750 ; MONITOR AUSFUHRUNGS ROUTINEN
                  1760 ;
                  1770 ;
                  1780 ; User Befehl
                  1790 ;
04A3 E1           1800 USER   POP  HL        ;erniedrige Stack
04A4 C30008       1810        JP   PALASM    ;zum PAL-Betriebssystem
                  1820 ;
                  1830 ; Tabulate Befehl
                  1840 ;
04A7 CD0301       1850 TABU   CALL XOTSTR    ;Zeige Speicherinhalt im
04AA 0D           1860        DEFB #0D       ;ubersichtlichen Block-
04AB 54414255     1870        DEFM 'TABULATE FROM:'  ;format .
     4C415445
     2046524F
     4D3A
04B9 00           1880        DEFB 00
04BA CD1404       1890        CALL EXPR   ;Hole Parameter
04BD E5           1900        PUSH HL     ;siche Parameter
04BE CD5504       1910        CALL CRLF   ;neue Zeile
04C1 CD0301       1920        CALL XOTSTR
04C4 20204164     1930        DEFM '  Adress   '
     72657373
     202020
04CF 00           1940        DEFB END
                  1950 ; Erstellen der Uberschrieft
04D0 3E30         1960        LD   A,"0      ;Zahlen von '0....9'
04D2 212A06       1970 TI1    LD   HL,HEX    ;und von 'A...F'
04D5 CD2A01       1980        CALL MSKTS
04D8 D4E004       1990        CALL NC,TI2
04DB 3C           2000        INC  A
04DC 280B         2010        JR   Z,MBO     ;fertig ?
04DE 18F2         2020        JR   TI1
                  2030 ;
04E0 CD0601       2040 TI2    CALL OTCHR
04E3 0E02         2050        LD   C,2       ;dazwischen 2 Leerzeich.
04E5 CD0204       2060        CALL SPACE
04E8 C9           2070        RET
                  2080 ;
04E9 E1           2090 MBO    POP  HL        ;HL=Eingangsparameter
04EA CD5504       2100        CALL CRLF      ;d.h. Startadr.
04ED E5           2110 MB     PUSH HL
04EE E5           2120        PUSH HL
04EF CD4605       2130        CALL FIELD     ;Bild aufbauen
04F2 D1           2140        POP  DE
04F3 CD1B05       2150        CALL PSUM      ;errechne Prufsumme
04F6 CD5504       2160        CALL CRLF      ;neue Zeile
04F9 CD1201       2170        CALL CI        ;Warte auf Anweissung
04FC FE0D         2180        CP   #0D       ;Wenn CR
04FE E1           2190        POP  HL
04FF C8           2200        RET  Z         ;dann beenden .
                  2210 ;
```

**** Monitorlisting ****

```
0500 118000    2220          LD   DE,#0080 ;Blockgr. 80H
0503 FE7F      2230          CP   DEL      ;Ruckwartz blattern ?
0505 280F      2240          JR   Z,MOC    ;Ja
0507 19        2250          ADD  HL,DE    ;Nein,dann Vorwartz
               2260 ;
0508 E5        2270 MOC1     PUSH HL       ;Siche neuen Parameter
0509 0609      2280          LD   B,9
050B CD0301    2290 MOC2     CALL XOTSTR   ;Bildaufbau am gleichen
050E 1B0C00    2300          DEFB ESC,CUU,END  ;Platz !
0511 10F8      2310          DJNZ MOC2
0513 E1        2320          POP  HL
0514 18D7      2330          JR   MB
               2340 ;
0516 B7        2350 MOC      OR   A        ;Losche Carry
0517 ED52      2360          SBC  HL,DE    ;eine Blockgr. weniger
0519 18ED      2370          JR   MOC1
               2380 ;
051B 051B      2390 PSUM     EQU  $
               2400 ; Funktion: Errechnet die Pruefsumme fuer
               2410 ;           den Adressbereich (DE) bis
               2420 ;           (DE) + 80H
               2430 ; ENT :DE=Start Adr.
               2440 ; RET :
               2450 ; CALLED :OUTHL XOTSTR
               2460 ;
051B CD0301    2470          CALL XOTSTR
051E 0D        2480          DEFB #0D
051F 43686563  2490          DEFM 'Checksum ='
     6B73756D
     203D
0529 00        2500          DEFB END
052A 218000    2510          LD   HL,#0080   ;Blockgr. 80H
052D 19        2520          ADD  HL,DE
052E EB        2530          EX   DE,HL   ;HL=Start DE=End
052F DD210000  2540          LD   IX,0    ;IX=Summationsreg.
0533 0600      2550          LD   B,0
0535 4E        2560 LOOP     LD   C,(HL)
0536 DD09      2570          ADD  IX,BC    ;aufaddieren
0538 A7        2580          AND  A        ;Loesche Carry
0539 ED52      2590          SBC  HL,DE
053B 19        2600          ADD  HL,DE
053C 23        2610          INC  HL       ;alles abgearbeitet ?
053D 20F6      2620          JR   NZ,LOOP ;Nein
053F DDE5      2630          PUSH IX      ;IX=Ergebnis
0541 E1        2640          POP  HL      ;HL=IX
0542 CD2401    2650          CALL OUTHL   ;Ergebnis ausgeben
0545 C9        2660          RET
```

**** Monitorlisting ****

```
                2670 ;
0546 0546       2680 FIELD  EQU  $
                2690 ;Funktion: Aufbau und ausgabe eines
                2700 ;          Speicherbereiches mit ASCII
                2710 ;          interpretation .
                2720 ; ENT: HL=Startadresse
                2730 ; RET: HL+80H
                2740 ; CALLED: SPACE,OUTA,OUTHL,CRLF
                2750 ;
                2760 ;
0546 0607       2770        LD   B,07     ;8 Zeilen aufbauen
0548 C5         2780 LLOOP  PUSH BC      ;und Ausgeben
0549 CD4F05     2790        CALL LINE    ;eine Zeile aufbauen
054C C1         2800        POP  BC
054D 10F9       2810        DJNZ LLOOP   ;noch eine Zeile ?
054F E5         2820 LINE   PUSH HL      ;neue Zeile
0550 0E03       2830        LD   C,3
0552 CD0204     2840        CALL SPACE   ;drei Leerzeichen
0555 CD2401     2850        CALL OUTHL   ;Adresse ausgeben
0558 0E03       2860        LD   C,3     ;noch drei
055A CD0204     2870        CALL SPACE   ;Leerzeichen
055D 0610       2880        LD   B,#10   ;16 Bytes pro Zeile
055F 7E         2890 LOOP1  LD   A,(HL)
0560 CD2101     2900        CALL OUTA    ;Byte ausgeben ,
0563 0E01       2910        LD   C,1     ;und danach ein
0565 CD0204     2920        CALL SPACE   ;Leerzeichen
0568 23         2930        INC  HL      ;Adressenzaehler+1
0569 10F4       2940        DJNZ LOOP1   ;Weiter alle 16 Bytes !
                2950 ; Als ASCII auswerten
056B E1         2960        POP  HL
056C 010210     2970        LD   BC,#1002 ;16 Bytes,zwei Blanks
056F CD0204     2980        CALL SPACE
0572 7E         2990 ASCII  LD   A,(HL)
0573 23         3000        INC  HL
0574 E67F       3010        AND  #7F     ;Maskiere auf 7 Bit ASCII
0576 FE7F       3020        CP   #7F     ;Delete code ?
0578 2804       3030        JR   Z,NONC  ;wechsle in "."
057A FE20       3040        CP   #20     ;Steuerzeichen ?
057C 3002       3050        JR   NC,OUT  ;Nein
057E 3E2E       3060 NONC   LD   A,"."
0580 CD0601     3070 OUT    CALL OTCHR   ;ASCII Zeichen ausgeben
0583 10ED       3080        DJNZ ASCII   ;alle 16 ?
0585 CD5504     3090        CALL CRLF    ;neue Zeile
0588 C9         3100        RET
                3110 ;
```

**** Monitorlisting ****

```
                3120 ;
0589 CD0301     3130 ENTER   CALL XOTSTR
058C 0D         3140         DEFB #0D
058D 45584543   3150         DEFM 'EXECUTE PRG. BY :'
     55544520
     5052472E
     20425920
     3A
059E 00         3160         DEFB 00
059F CD1404     3170         CALL EXPR     ;Hole Start Adr.
05A2 E9         3180         JP   (HL)      ;Sprung zum Anwender
                3190 ;  Das Anwenderprogramm soll mit einem
                3200 ;  RET Befehl abschliessen !!!!
                3210 ;  ( Automaticher Rueckspruns zum Monitor)
                3220 ;
                3230 ;
05A3 05A3       3240 MODIFY EQU   $       ;Verandere Speicherplatz
05A3 CD0301     3250         CALL XOTSTR
05A6 0D         3260         DEFB #0D
05A7 4D656D6F   3270         DEFM 'Memory modify in :'
     7279206D
     6F646966
     7920696E
     203A
05B9 00         3280         DEFB 00
05BA CD1404     3290         CALL EXPR    ;Hole Argument
05BD CD5504     3300 ELOOP   CALL CRLF
05C0 CD2401     3310 LOOPA   CALL OUTHL   ;Adr. in ASCII ausgeben
05C3 E5         3320         PUSH HL      ;Zeiger retten
05C4 CD0901     3330 WHOLEB  CALL XOTCHR
05C7 20         3340         DEFB #20     ;Leerzeichen
05C8 7E         3350         LD   A,(HL)
05C9 CD2101     3360         CALL OUTA    ;Byte in HEX ausgeben
05CC 23         3370         INC  HL      ;Zeiger auf neues Byte
05CD D1         3380         POP  DE      ;Zeiger alt
05CE CD0901     3390         CALL XOTCHR
05D1 20         3400         DEFB #20     ;Leerzeichen
                3410 ;
05D2 CD0901     3420         CALL XOTCHR
05D5 3A         3430         DEFB ":
05D6 1A         3440         LD   A,(DE) ;Als ASCII interpretiere
05D7 E67F       3450         AND  #7F     ;7 Bit ASCII
05D9 FE20       3460         CP   #20     ;Steuerzeichen ?
05DB 3806       3470         JR   C,CONT
05DD FE7F       3480         CP   #7F     ;Delete code ?
05DF 2802       3490         JR   Z,CONT ;Ja dann Blank
05E1 1802       3500         JR   CONT1
                3510 ;
```

**** Monitorlisting ****

```
05E3 3E20       3520 CONT    LD   A,#20   ;Leerzeichen
05E5 CD0601     3530 CONT1   CALL OTCHR
05E8 CD0901     3540         CALL XOTCHR
05EB 3A         3550         DEFB ":
05EC CD1201     3560 L2      CALL CI
05EF FE7F       3570         CP   DEL     ;Delete code ?
05F1 2803       3580         JR   Z,L11
05F3 CD0C04     3590         CALL LITBIG ;klein => gross wandeln
                3600 ;
05F6 CD0601     3610 L11     CALL OTCHR
                3620 ; Steuercode Sprungverteiler
05F9 CD1E01     3630 L3      CALL XFUNCT
05FC 7F         3640         DEFB #7F     ;Zurueck ?
05FD 2F06       3650         DEFW BACKGO
05FF 0D         3660         DEFB #0D     ;CR ?
0600 3406       3670         DEFW LPPA
0602 20         3680         DEFB #20     ;Blank ?
0603 3106       3690         DEFW L5      ;erhoehe
0605 2E         3700         DEFB ".      ;Punkt ?
0606 3206       3710         DEFW L6      ;Warmstart Monitor
0608 22         3720         DEFB #22     ;"=Text-start -ende
0609 3706       3730         DEFW TEXT    ;Text Eingabe
060B 00         3740         DEFB 00
060C 1006       3750         DEFW L4      ;Kein Befehl
060E 18AD       3760         JR   ELOOP
0610 E1         3770 L4      POP  HL
0611 CD0C04     3780         CALL LITBIG  ;wandle in gross Buchst
0614 210000     3790         LD   HL,0000 ;Loesche FIFO
0617 CD2A04     3800         CALL EX1
061A 47         3810         LD   B,A     ;B=Trennzeichen
061B 7D         3820         LD   A,L
061C 12         3830         LD   (DE),A
061D 13         3840         INC  DE
061E 78         3850         LD   A,B
061F FE20       3860         CP   "       ;Trennzeichen=Blank?
0621 28C9       3870         JR   Z,L2    ;Ja,weitere Eingaben
0623 EB         3880         EX   DE,HL
0624 1897       3890         JR   ELOOP
                3900 ;
                3910 ; Masken fuer MSKTS Routine
                3920 ;
0626 0D0D2020   3930 HEXM    DEFB #0D,#0D," ,"
062A 30394146   3940 HEX     DEFB "0,"9,"A,"F,END  ;sed. Zeichen
     00
```

**** Monitorlisting ****

```
                 3950 ;
                 3960 ; UNTERROUTINEN ZU MODIFY
                 3970 ;
                 3980 ;
                 3990 ; DELETE TASTE
                 4000 ;
062F 1B          4010 BACKGO DEC  DE     ;Adr.-1
0630 EB          4020        EX   DE,HL
0631 C9          4030 L5     RET
                 4040 ;
                 4050 ; PUNKT BEFEHL
                 4060 ;
0632 E1          4070 L6     POP  HL     ;Erniedige Stack
0633 C9          4080        RET  ;Monitor Warmstart
                 4090 ;
                 4100 ; CR BEFEHL
                 4110 ;
0634 EB          4120 LPPA   EX   DE,HL
0635 23          4130        INC  HL     ;Adr.+1
0636 C9          4140        RET
                 4150 ;
                 4160 ; Text Modus
                 4170 ;
0637 CD1201      4180 TEXT   CALL CI       ;nun empfange Text
063A CD0601      4190        CALL OTCHR    ;sofort ausgeben
063D FE22        4200        CP   #22      ;Textende Zeichen ?
063F 2808        4210        JR   Z,CONTE  ;Ja
0641 FE7F        4220        CP   #7F      ;BACKSPACE ?
0643 2813        4230        JR   Z,BACK   ;Ja
0645 12          4240        LD   (DE),A   ;einspeichern
0646 13          4250        INC  DE       ;Zeiger+1
0647 18EE        4260        JR   TEXT
                 4270 ;
0649 CD1201      4280 CONTE  CALL CI
064C CD0601      4290        CALL OTCHR
064F FE20        4300        CP   "        ;weitere eingaben in der
0651 2802        4310        JR   Z,C01    ;gleiche Zeile ?
0653 EB          4320        EX   DE,HL
0654 C9          4330        RET
                 4340 ;
0655 E1          4350 C01    POP  HL   ;anderer Sprung , da weitere
0656 1894        4360        JR   L2   ;eingaben
                 4370 ;
                 4380 ;  Backspace im Textmodus
                 4390 ;
0658 CD0301      4400 BACK   CALL XOTSTR      ;rueck,blank,rueck
065B 08200800    4410        DEFB CUL,#20,CUL,END
065F 1B          4420        DEC  DE          ;Zeiger-1
0660 18D5        4430        JR   TEXT
```

**** Monitorlisting ****

```
                4440 ;
                4450 ;
                4460 ;Ein / Ausgabe an Ports
                4470 ;
                4480 ;
0662 0662       4490 OUTPO   EQU  $
0662 CD0301     4500         CALL XOTSTR
0665 0D         4510         DEFB #0D
0666 4F757420   4520         DEFM 'Out to'
     746F
066C 20506F72   4530 PORT    DEFM ' Portadr. :'
     74616472
     2E203A
0677 00         4540         DEFB 00
0678 CD1404     4550         CALL EXPR
067B 4D         4560         LD   C,L      ;C=Portadr.
067C C5         4570         PUSH BC       ;Sichre Portadr.
067D CD0301     4580         CALL XOTSTR
0680 20425954   4590         DEFM ' BYTE :'
     45203A
0687 00         4600         DEFB END
0688 CD1404     4610         CALL EXPR
068B 7D         4620         LD   A,L      ;A=Byte
068C C1         4630         POP  BC       ;Hole Portadr. in C
068D ED79       4640         OUT  (C),A    ;ausgabe !
068F C9         4650         RET
                4660 ;
                4670 ;
0690 0690       4680 INPO    EQU  $
0690 CD0301     4690         CALL XOTSTR
0693 0D         4700         DEFB #0D
0694 496E2066   4710         DEFM 'In from'
     726F6D
069B 01         4720         DEFB IND
069C 6C06       4730         DEFW PORT     ;' Portadr. :'
069E 00         4740         DEFB 00
069F CD1404     4750         CALL EXPR
06A2 4D         4760         LD   C,L      ;C=Portadr.
06A3 ED78       4770         IN   A,(C)    ;Hole von Port
06A5 CD0901     4780         CALL XOTCHR
06A8 20         4790         DEFB "
06A9 CD2101     4800         CALL OUTA     ;gelesenes ausgeben !
06AC C9         4810         RET
                4820 ;
```

**** Monitorlisting ****

ZEAP Z80 Assembler - Symbol Table

0572H	2990	ASCII	0658H	4400	BACK
062FH	4010	BACKGO	0007H	0360	BEL
0655H	4350	CO1	0112H	0190	CI
05E3H	3520	CONT	05E5H	3530	CONT1
0649H	4280	CONTE	0455H	1300	CRLF
0118H	0210	CSTS	000BH	0450	CUD
0008H	0430	CUL	0010H	0420	CUR
000CH	0440	CUU	007FH	0380	DEL
044DH	1230	ECHU	05BDH	3300	ELOOP
0000H	0340	END	0589H	3130	ENTER
0497H	1690	ERROR	001BH	0370	ESC
0417H	0860	EXO	042AH	0960	EX1
0447H	1180	EX2	0414H	0850	EXPR
0546H	2680	FIELD	062AH	3940	HEX
0626H	3930	HEXM	0001H	0350	IND
0100H	0260	INIT	0690H	4680	INPO
0100H	0180	IOPACK	05F6H	3610	L11
05ECH	3560	L2	05F9H	3630	L3
0610H	3770	L4	0631H	4030	L5
0632H	4070	L6	054FH	2820	LINE
040CH	0650	LITBIG	0548H	2780	LLOOP
0535H	2560	LOOP	055FH	2890	LOOP1
0403H	0580	LOOP2	05C0H	3310	LOOPA
0634H	4120	LPPA	04EDH	2110	MB
04E9H	2090	MBO	045CH	1400	MLOOP
0516H	2350	MOC	0508H	2270	MOC1
050BH	2290	MOC2	05A3H	3240	MODIFY
045CH	1390	MONI	012AH	0300	MSKTS
0437H	1060	NIBBEL	057EH	3060	NONC
0106H	0290	OTCHR	0580H	3070	OUT
0121H	0240	OUTA	0124H	0250	OUTHL
0662H	4490	OUTPO	0800H	0320	PALASM
066CH	4530	PORT	0121H	0220	PRAC
0124H	0230	PRHL	051BH	2390	PSUM
0402H	0500	SPACE	0400H	0010	START
04A7H	1850	TABU	0637H	4180	TEXT
04D2H	1970	TI1	04E0H	2040	TI2
04A3H	1800	USER	05C4H	3330	WHOLEB
0470H	1440	WSTR	011EH	0280	XFUNCT
0109H	0200	XOTCHR	0103H	0270	XOTSTR

**** PAL-Betriebssystem ***

ZEAP Z80 Assembler - Source Listing

```
0000            0010          ORG  0
                0020 ;*************************
                0030 ;*      Abschlussarbeit        *
                0040 ;* M. Boretius F. Zecherle     *
                0050 ;*                             *
                0060 ;* Funktion: Pal Betriebs-     *
                0070 ;*            system           *
                0080 ;*                             *
                0090 ;* Autor:  Peter  Zecherle     *
                0100 ;*************************
                0110 ;
0000 0D60       0115 HPACK  EQU  #0D60
0000 0D60       0120 PALPRG EQU  HPACK+#00
                0130 ;
0000 0100       0140 IOPACK EQU  #0100
0000 0100       0150 INIT   EQU  IOPACK
0000 0103       0160 XOTSTR EQU  IOPACK+#03
0000 0106       0170 OTCHR  EQU  IOPACK+#06
0000 0109       0180 XOTCHR EQU  IOPACK+#09
0000 010C       0190 OTSTR  EQU  IOPACK+#0C
0000 010F       0200 TDEL   EQU  IOPACK+#0F
0000 0112       0210 INCHR  EQU  IOPACK+#12
0000 0115       0220 INDAT  EQU  IOPACK+#15
0000 0118       0230 SCAN   EQU  IOPACK+#18
0000 011B       0240 MOD    EQU  IOPACK+#1B
0000 011E       0250 XFUNCT EQU  IOPACK+#1E
0000 0121       0260 OUTA   EQU  IOPACK+#21
0000 0124       0270 OUTHL  EQU  IOPACK+#24
0000 0127       0280 EDIT   EQU  IOPACK+#27
0000 012A       0290 MSKTS  EQU  IOPACK+#2A
                0300 ;
                0310 ; Weitere Externe Programme
                0320 ;
0000 0400       0330 MON    EQU  IOPACK+#300
```

**** PAL-Betriebssystem ***

```
                0340 ;
                0350 ; Konstanten Def.
                0360 ;
0000 2000       0370 RAM2S  EQU  #2000       ;2. RAM Bereich Start
0000 23FF       0380 RAM2E  EQU  RAM2S+#3FF  ;2. RAM Bereich Ende
0000 2010       0390 BBUFS  EQU  RAM2S+#10   ;Befehlsbuffer
0000 2020       0400 TBUFS  EQU  RAM2S+#20   ;Start Textbuffer
                0410 ;
                0420 ;
0000 0080       0430 STACK  EQU  #80 ;Platz fuer Stack
0000 237F       0440 RAMTOP EQU  RAM2E-STACK ;Freies Ram bis ...
                0450 ;
0000 0000       0460 END    EQU  00
0000 0001       0470 IND    EQU  01
0000 0002       0480 REP    EQU  02
0000 0007       0490 BEL    EQU  07
0000 001B       0500 ESC    EQU  #1B ;Escape code
                0510 ;
0000 007F       0520 DEL    EQU  #7F
0000 000B       0530 INS    EQU  #0B
0000 000E       0540 CINS   EQU  #0E  ;Controll I
0000 0004       0550 CDEL   EQU  #04  ;Controll D
                0560 ;  CURSOR TASTEN
0000 0010       0570 CUR    EQU  #10
0000 0008       0580 CUL    EQU  #08
0000 000C       0590 CUP    EQU  #0C ;ESC CUP
0000 000B       0600 CUD    EQU  #0B ;ESC CUD
0000 0012       0610 HOME   EQU  #12 ;ESC HOME
0000 0013       0620 CL     EQU  #13 ;ESC CL Loesche Zeile
                0630 ;
                0640 ; PAL Typen Liste Def.
0000 0018       0650 TLEN   EQU  24 ;Anzahl der PAL Typen
0000 000B       0660 DLEN   EQU  11 ;Datenbyte pro Satz
                0670 ;
                0680 ;
                0690 ; Speicher Def.
2000            0700        ORG  RAM2S
2000 0000       0710 PALT   DEFW 0 ;Zeiger PAL Type
2002 0000       0720 TST    DEFW 0 ;Textbuffer Start
2004 0000       0730 TZ     DEFW 0 ;Textbuffer Zeiger
2006 0000       0740 TEND   DEFW 0 ;Textbuffer Ende
2008 0000       0750 BST    DEFW 0 ;Befehlsbuffer Start
200A 0000       0760 BZ     DEFW 0 ;Befehlsbuffer Zeiger
200C 0000       0770 BEND   DEFW 0 ;Befehlsbuffer Ende
                0780 ;
0000            0790        ORG  0000
0000 31FF23     0800        LD   SP,RAM2E  ;Setze Stack
0003 CD0001     0810        CALL INIT     ;Init. Ports
0006 C30008     0820        JP   #800
                0830 ;
```

**** PAL-Betriebssystem ***

```
                0840 ;
                0850 ;
                0860 ;
                0870 ; P R O G R A M M  S T A R T
                0880 ;
0800            0890 START  ORG  #0800
0800 210020     0900        LD   HL,RAM2S      ;Loesche RAM
0803 110120     0910        LD   DE,RAM2S+1
0806 017F03     0920        LD   BC,RAM2E-RAM2S-STACK
0809 3600       0930        LD   (HL),#00
080B EDB0       0940        LDIR
                0950 ;
080D CD7E0B     0960        CALL CSC      ;Loesche Bildschirm
0810 CD0301     0970        CALL XOTSTR        ;Startmeldung
0813 1B12       0980 WSM    DEFB ESC,HOME
0815 026B20     0990        DEFB REP,107,"
0818 50414C2D   1000        DEFM 'PAL-Assembler V 0.1'
     41737365
     6D626C65
     72205620
     302E31
082B 022D20     1010        DEFB REP,45,#20
082E 2A2A2A20   1020        DEFM '*** Copyright by M. Boretius &'
     436F7079
     72696768
     74206279
     204D2E20
     426F7265
     74697573
     2026
084C 20502E20   1030        DEFM ' P. Zecherle ***'
     5A656368
     65726C65
     202A2A2A
085C 0A0D0A0D   1040        DEFB #0A,#0D,#0A,#0D,END
     00
                1050 ;
0861 CD8A0B     1060 WS1    CALL CRLF
0864 CD0301     1070        CALL XOTSTR
0867 01         1080        DEFB IND
0868 710B       1090        DEFW INMELD
086A 00         1100        DEFB END
                1110 ; Warte auf Befehlsanweisung
                1120 ;
086B CDB708     1130        CALL CHBUFZ    ;Rette alte Zeiger
086E 211020     1140        LD   HL,BBUFS ;Setze Befehls Buffer Zeig
0871 CD2701     1150        CALL EDIT
0874 CDB708     1160        CALL CHBUFZ    ;Hole alte Zeiger
0877 2A0820     1170        LD   HL,(BST)
087A 7E         1180        LD   A,(HL)    ;A=Befehls Code
                1190 ;
```

**** PAL-Betriebssystem ***

```
087B CD1E01    1200          CALL XFUNCT   ;Befehls  Verteiler
087E 45        1210          DEFB "E
087F C908      1220          DEFW PLE  ;Editiere Produktline
0881 50        1230          DEFB "P
0882 4C0B      1240          DEFW LEND ;Programmiere PAL
0884 4D        1250          DEFB "M
0885 C908      1260          DEFW JMON ;Springe zum Monitor
0887 46        1270          DEFB "F
0888 CD08      1280          DEFW FUSE ;Zeige Sicherungen
088A 4C        1290          DEFB "L
088B C908      1300          DEFW LOAD ;Lade PAL File
088D 53        1310          DEFB "S
088E C908      1320          DEFW SAVE ;Sichre PAL File
0890 4E        1330          DEFB "N
0891 320A      1340          DEFW NEW  ;Neue Eingaben
0893 3F        1350          DEFB "?
0894 BD09      1360          DEFW HELP ;Hilfe Stellung
0896 49        1370          DEFB "I
0897 BD09      1380          DEFW HELP ;Hilfe Stellung
0899 48        1390          DEFB "H
089A BD09      1400          DEFW HELP ;Hilfe Stellung
089C 00        1410          DEFB END
089D A108      1420          DEFW BERROR
089F 18C0      1430          JR   WS1
               1440 ;
08A1 CD8A0B    1450 BERROR CALL CRLF  ;Meldung bei falscher
08A4 CD0301    1460          CALL XOTSTR     ;Eingabe
08A7 0707      1470          DEFB BEL,BEL
08A9 4B65696E  1480          DEFM 'Kein Befehl '
     20426566
     65686C20
08B5 00        1490          DEFB END
08B6 C9        1500          RET
               1510 ;
08B7 08B7      1520 CHBUFZ EQU  $
               1530 ; Funktion: Tausche Buffer Zeiger aus.
               1540 ;    (Befehls-  <=> Text- Buffer)
               1550 ;
08B7 0606      1560          LD   B,6       ;Drei Adressen tauschen
08B9 210220    1570          LD   HL,TST    ;Textbuffer Start Zeiger
08BC 110820    1580          LD   DE,BST    ;Befehlsbuffer Start Zeiger
               1590 ;
08BF 4E        1600 BZ1      LD   C,(HL)
08C0 1A        1610          LD   A,(DE)
08C1 77        1620          LD   (HL),A
08C2 79        1630          LD   A,C
08C3 12        1640          LD   (DE),A
08C4 23        1650          INC  HL
08C5 13        1660          INC  DE
08C6 10F7      1670          DJNZ BZ1
08C8 C9        1680          RET
```

**** PAL-Betriebssystem ***

```
                1690 ;*********************************
                1700 ;  Unterprogramme zu den Befehlen
                1710 ;*********************************
                1720 ;
08C9 08C9       1730 PLE     EQU  $  ;nicht vorhanden
08C9 08C9       1740 SAVE    EQU  $  ;nicht vorhanden
08C9 08C9       1750 LOAD    EQU  $  ;nicht vorhanden
                1760 ;
                1770 ; Springe zum System Monitor
                1780 ;
08C9 E1         1790 JMON    POP  HL
08CA C30004     1800         JP   MON
                1810 ;
                1820 ;
08CD 08CD       1830 FUSE    EQU  $
                1840 ; Funktion: Darstellen der Programmier-
                1850 ;           matrix.Augabe auf Bildschirm
                1860 ; ENT: NON
                1870 ; RET: NON
                1880 ; CALLED: XOTSTR,XOTCHR,INCHR,CSC,CRLF
                1890 ;
08CD DD210810   1900         LD   IX,#1008   ;Zeiger auf PRL
08D1 FD214710   1910         LD   IY,#1048-1 ;Zeiger auf IPL
08D5 CD7E0B     1920         CALL CSC       ;Losche Bildschirm
                1930 ;
08D8 CD0301     1940 F1      CALL XOTSTR
08DB 01         1950         DEFB IND
08DC 1308       1960         DEFW WSM
08DE 41756620   1970         DEFM 'Auf Printer (P) oder Display (D):'
     5072696E
     74657220
     28502920
     6F646572
     20446973
     706C6179
     20284429
     3A
08FF 0A0D00     1980         DEFB #0A,#0D,END
0902 CD1201     1990         CALL INCHR
0905 FE50       2000         CP   "P
0907 28CF       2010         JR   Z,F1 ;Wenn Printer dann Jump
                2020 ;  (Printer Treiber noch nicht vorhanden )
                2030 ;
0909 AF         2040 NEWPRL  XOR  A   ;Teste auf PRL Ende
090A DDBE00     2050         CP   (IX+0)
090D F5         2060         PUSH AF
090E CD8A0B     2070         CALL CRLF
0911 F1         2080         POP  AF
0912 202A       2090         JR   NZ,NIL ;Weiter kein Ende
```

**** PAL-Betriebssystem ***

```
                2100 ;
0914 CD0301     2110          CALL XOTSTR
0917 202D203D   2120          DEFM ' - =Fuse blown  / * =Fuse not '
     46757365
     20626C6F
     776E2020
     2F202A20
     3D467573
     65206E6F
     7420
0935 626C6F77   2130          DEFM 'blown'
     6E
093A 0A0D00     2140          DEFB #0A,#0D,END
093D C9         2150          RET
                2160 ;
093E CD0301     2170 NIL      CALL XOTSTR ;PRL Wert mit Text
0941 01         2180          DEFB IND    ;Ausgeben
0942 2A0B       2190          DEFW PRLM
0944 00         2200          DEFB END
0945 DD7E00     2210          LD   A,(IX+0)
0948 CBBF       2220          RES  7,A
094A CD2101     2230          CALL OUTA
094D CD0901     2240          CALL XOTCHR ;Leerzeichen
0950 20         2250          DEFB "
0951 DD23       2260          INC  IX     ;PRL+1
0953 FD23       2270          INC  IY     ;IPL+1
0955 0681       2280          LD   B,#81
                2290 ;
0957 AF         2300 NIL0     XOR  A      ;IPL END ?
0958 FDBE00     2310          CP   (IY+0)
095B 283C       2320          JR   Z,NPRLC ;Teste auf Input Line
                2330 ;
095D 095D       2340 NIL1     EQU  $      ;A=IPL
095D FD7E00     2350          LD   A,(IY+0) ;B=Testzeiger
0960 B8         2360          CP   B
0961 2814       2370          JR   Z,STERN
0963 382B       2380          JR   C,BLOWN
0965 CDA909     2390          CALL IPLT   ;IPL vorhanden?
0968 3807       2400          JR   C,NIL3
                2410 ;
096A CD0901     2420          CALL XOTCHR ;Stern Ausgeben
096D 2A         2430          DEFB "*
096E 04         2440 NIL4     INC  B
096F 18EC       2450          JR   NIL1
                2460 ;
0971 CD0901     2470 NIL3     CALL XOTCHR ;Leerzeichen
0974 20         2480          DEFB "
0975 18F7       2490          JR   NIL4
```

**** PAL-Betriebssystem ***

```
                2500 ;
0977 0977       2510 STERN   EQU  $
0977 CDA909     2520         CALL IPLT    ;IPL vorhanden?
097A 380E       2530         JR   C,NIL6
097C CD0901     2540         CALL XOTCHR
097F 2A         2550         DEFB "*
0980 CD0901     2560 NIL5    CALL XOTCHR
0983 2D         2570         DEFB "-
0984 04         2580         INC  B        ;Testzeiger+2
0985 04         2590         INC  B
0986 FD23       2600         INC  IY       ;Nachste IPL
0988 18CD       2610         JR   NILO
                2620 ;
098A CD0901     2630 NIL6    CALL XOTCHR   ;Leerzeichen
098D 20         2640         DEFB "
098E 18F0       2650         JR   NIL5
                2660 ;
0990 0990       2670 BLOWN   EQU  $     ;Sicherung geschossen
0990 CD0901     2680         CALL XOTCHR   ; - Ausgeben
0993 2D         2690         DEFB "-
0994 04         2700         INC  B
0995 FD23       2710         INC  IY
0997 18BE       2720         JR   NILO
                2730 ;
0999 FD7EFF     2740 NPRLC   LD   A,(IY-1) ;Teste op PRL innerhalb
099C 3C         2750 NIL7    INC  A         ; der PLR Tabelle ist
099D CDAB09     2760         CALL TMIL
09A0 DA0909     2770         JP   C,NEWPRL
09A3 CD0901     2780         CALL XOTCHR
09A6 2A         2790         DEFB "*
09A7 18F3       2800         JR   NIL7
                2810 ;
09A9 09A9       2820 IPLT    EQU  $
                2830 ; Funktion: Testet op IPL in IPL Tabelle
                2840 ;           vorhanden ist
                2850 ; CALLED: MSKTS
                2860 ;
09A9 78         2870         LD   A,B
09AA 3D         2880         DEC  A
                2890 ;
09AB DDE5       2900 TMIL    PUSH IX
09AD DD2A0020   2910         LD   IX,(PALT)
09B1 DD6E09     2920         LD   L,(IX+9)
09B4 DD660A     2930         LD   H,(IX+10)  ;HL=Tabellenadr.
09B7 CD2A01     2940         CALL MSKTS
09BA DDE1       2950         POP  IX
09BC C9         2960         RET
                2970 ;
```

**** PAL-Betriebssystem ***

```
                  2980 ;
                  2990 ;
                  3000 ;
                  3010 ;
09BD 09BD         3020 HELP   EQU  $
                  3030 ; Funktion: Listet die vorhanden Befehle
                  3040 ;           auf
                  3050 ;
                  3060 ; ENT: NON
                  3070 ; RET: NON
                  3080 ; CALLED: XOTSTR
                  3090 ;
09BD CD0301       3100        CALL XOTSTR
09C0 0A0D0214     3110        DEFB #0A,#0D,REP,20,"*,#0A,#0D
     2A0A0D
09C7 4C203D4C     3120        DEFM 'L =Lade PAL File'
     61646520
     50414C20
     46696C65
09D7 0A0D         3130        DEFB #0A,#0D
09D9 53203D53     3140        DEFM 'S =Sichere PAL File'
     69636865
     72652050
     414C2046
     696C65
09EC 0A0D         3150        DEFB #0A,#0D
09EE 4E203D4E     3160        DEFM 'N =Neues PAL'
     65756573
     2050414C
09FA 0A0D         3170        DEFB #0A,#0D
09FC 50203D50     3180        DEFM 'P =Prog. PAL'
     726F672E
     2050414C
0A08 0A0D         3190        DEFB #0A,#0D
0A0A 46203D46     3200        DEFM 'F =Fuse Check'
     75736520
     43686563
     6B
0A17 0A0D         3210        DEFB #0A,#0D
0A19 45203D45     3220        DEFM 'E =Editiere PRL.'
     64697469
     65726520
     50524C2E
0A29 0A0D0214     3230        DEFB #0A,#0D,REP,20,"*,#0A,#0D,END
     2A0A0D00
0A31 C9           3240        RET
```

**** PAL-Betriebssystem ***

```
                3250 ;
                3260 ;
0A32 0A32       3270 NEW     EQU  $
                3280 ; Funktion: Eroffnen eines neuen PAL Files
                3290 ;           Eingabe uber Terminal
                3300 ;
                3310 ; CALLED: XOTSTR,CRLF,EDIT
                3320 ;
0A32 CD0301     3330         CALL XOTSTR     ;Bringe Sicherheitsabfrage
0A35 C0         3340         DEFB #0D
0A36 47616E7A   3350         DEFM 'Ganz neue Eingabe? (J/N):   '
     206E6575
     65204569
     6E676162
     653F2028
     4A2F4E29
     3A2020
0A51 080800     3360         DEFB CUL,CUL,END
0A54 CD1201     3370         CALL INCHR
0A57 CD8A0B     3380         CALL CRLF
0A5A FE4A       3390         CP   "J
0A5C C0         3400         RET  NZ         ;=>keine neue Eingabe
                3410 ;
0A5D CD0301     3420 PT      CALL XOTSTR
0A60 50414C20   3430         DEFM 'PAL Type bitte :     '
     54797065
     20626974
     7465203A
     20202020
0A74 020408     3440         DEFB REP,4,CUL
0A77 00         3450         DEFB END
0A78 212020     3460         LD   HL,TBUFS ;Textbuffer Start laden
0A7B CD2701     3470         CALL EDIT
0A7E CD850A     3480         CALL SEARCH   ;Suche nach PAL Type
0A81 38DA       3490         JR   C,PT     ;=>PAL unbekannt
0A83 185F       3500         JR   PALASM   ;Weitere Eingaben
```

**** PAL-Betriebssystem ***

```
                3510 ;
0A85 0A85       3520 SEARCH EQU  $
                3530 ; Funktion: Durchsuchen der Pal Tabelle
                3540 ;           nach Eingegebenen Type
                3550 ;
                3560 ; ENT: (TST)=Textbuffer Start
                3570 ; RET: Bei nicht vorhanden ,setze Carry
                3580 ;      und bringe Meldung
                3590 ; CALLED: XOTSTR
                3600 ;
0A85 0618       3610        LD   B,TLEN    ;Anzahl der Eintraege
0A87 DD21910B   3620        LD   IX,TABST  ;Tabellen Start
0A8B FD2A0220   3630        LD   IY,(TST)  ;Textbuffer Start
                3640 ;
0A8F FD7E00     3650 L2     LD   A,(IY+0)
0A92 FE20       3660        CP   #20
0A94 2004       3670        JR   NZ,L1
0A96 FD23       3680        INC  IY   ;overjump blanks
0A98 18F5       3690        JR   L2
                3700 ;
0A9A DD7E00     3710 L1     LD   A,(IX+0)  ;Teste 1. Zeichen
0A9D FDBE00     3720        CP   (IY+0)
0AA0 2018       3730        JR   NZ,NEXT
0AA2 DD7E01     3740        LD   A,(IX+1)  ;Test 2. Zeichen
0AA5 FDBE01     3750        CP   (IY+1)
0AA8 2010       3760        JR   NZ,NEXT
0AAA DD7E02     3770        LD   A,(IX+2)  ; 3. Zeichen
0AAD FDBE02     3780        CP   (IY+2)
0AB0 2008       3790        JR   NZ,NEXT
0AB2 DD7E03     3800        LD   A,(IX+3)  ; 4. Zeichen
0AB5 FDBE03     3810        CP   (IY+3)
0AB8 2824       3820        JR   Z,FOUND
0ABA 05         3830 NEXT   DEC  B
0ABB 2807       3840        JR   Z,NOPAL ;PAL-Type nicht gefunden
0ABD 110B00     3850        LD   DE,DLEN ;Datensatzlaenge
0AC0 DD19       3860        ADD  IX,DE   ;Neuer  D.-satz
0AC2 18D6       3870        JR   L1
```

**** PAL-Betriebssystem ***

```
                3880 ;
0AC4 CD0301     3890 NOPAL   CALL XOTSTR
0AC7 0D         3900         DEFB #0D
0AC8 50414C20   3910         DEFM 'PAL Type unbekannt '
     54797065
     20756E62
     656B616E
     6E7420
0ADB 00         3920         DEFB END
0ADC 37         3930         SCF
0ADD C9         3940         RET
                3950 ;
0ADE DD220020   3960 FOUND   LD   (PALT),IX ;Setzt PAL Zeiger
0AE2 B7         3970         OR   A          ;Losche Carry
0AE3 C9         3980         RET
                3990 ;
                4000 ;
0AE4 0AE4       4010 PALASM EQU  $
                4020 ; Funktion: Eingabe der Kreutzungspunkte
                4030 ;           Bedienung des Editores
                4040 ;
                4050 ; ENT: NON
                4060 ; CALLED: XOTSTR,CRLF,MSKTS,EDIT
                4070 ;
0AE4 DD2A0020   4080         LD   IX,(PALT)     ;Zeiger auf Pal Liste
                4090 ;
0AE8 CD8A0B     4100         CALL CRLF          ;Bild Aufbauen
0AEB CD0301     4110         CALL XOTSTR
0AEE 0D02112D   4120         DEFB #0D,REP,17,"-","/",REP,10,"-
     2F020A2D
0AF6 496E7075   4130         DEFM 'Input line bitte'
     74206C69
     6E652062
     69747465
0B06 020A2D00   4140         DEFB REP,10,"-",END
                4150 ;
0B0A 3E80       4160         LD   A,128       ;Zahler vorsetzen
0B0C 112020     4170         LD   DE,TBUFS
                4180 ;
0B0F DD6E07     4190 PRO1    LD   L,(IX+7)    ;HL=PRL Tabelle start
0B12 DD6608     4200         LD   H,(IX+8)
0B15 CD2A01     4210         CALL MSKTS
0B18 3007       4220         JR   NC,DISP   ;Gultige PRL
0B1A 3C         4230 PRO2    INC  A
0B1B 20F2       4240         JR   NZ,PRO1   ;Bearbeiten? Nein=>
0B1D CDD006     4250         CALL #06D0     ;Ja,Matrix erzeugen
0B20 C9         4260         RET
```

**** PAL-Betriebssystem ***

```
                  4270 ;
0B21 F5           4280 DISP   PUSH AF        ;Ausgabe von Meldung
0B22 D5           4290        PUSH DE        ;und PRL Nr.
0B23 E67F         4300        AND  #7F   ;7 bit Zahl
0B25 CD0301       4310        CALL XOTSTR
0B28 0A0D         4320        DEFB #0A,#0D
0B2A 50726F64     4330 PRLM   DEFM 'Produkt line :'
     756B7420
     6C696E65
     203A
0B38 00           4340        DEFB END
0B39 CD2101       4350        CALL OUTA
0B3C CD0301       4360        CALL XOTSTR
0B3F 203E00       4370        DEFB " ,">,END
                  4380 ;
0B42 EB           4390        EX   DE,HL    ;Warten auf Eingabe
0B43 D1           4400        POP  DE       ;String ,benutze den
0B44 CD2701       4410        CALL EDIT     ;Editor
0B47 EB           4420        EX   DE,HL
0B48 13           4430        INC  DE
0B49 F1           4440        POP  AF
0B4A 18CE         4450        JR   PRO2
                  4460 ;
                  4470 ;
0B4C CD0301       4480 LEND   CALL XOTSTR
0B4F 0A0D         4490        DEFB #0A,#0D
0B51 2A2A2A50     4500        DEFM '***PAL wird Programmiert***'
     414C2077
     69726420
     50726F67
     72616D6D
     69657274
     2A2A2A
0B6C 00           4510        DEFB END
0B6D CD600D       4520        CALL PALPRG
0B70 C9           4530        RET
                  4540 ;
                  4550 ;
0B71 49687220     4560 INMELD DEFM 'Ihr Befehl >'
     42656665
     686C203E
0B7D 00           4570        DEFB END
                  4580 ;
                  4590 ;
                  4600 ; LOSCHE BILDSCHIRM
0B7E 0618         4610 CSC    LD   B,24
0B80 CD0301       4620 C1     CALL XOTSTR
0B83 02502000     4630        DEFB REP,80,#20,END
0B87 10F7         4640        DJNZ C1
0B89 C9           4650        RET
                  4660 ;
                  4670 ;
0B8A CD0301       4680 CRLF   CALL XOTSTR
0B8D 0D0A00       4690        DEFB #0D,#0A,END
0B90 C9           4700        RET
```

```
                                  **** PAL-Betriebssystem ***

                    4710 ;
0B91 0B91           4720 TABST  EQU  $
                    4730 ; Liste der PALs
                    4740 ;
                    4750 ; Aufbau :
                    4760 ; PINs,Eingange,Ausgange
                    4770 ;  Adr. Prod. Line,Adr. Input Line
                    4780 ;
0B91 31365234 4790          DEFM '16R4'
0B95 141008   4800          DEFB 20,16,8
0B98 920C950C 4810          DEFW P16R4,I16R4
0B9C 31365236 4820          DEFM '16R6'
0BA0 141008   4830          DEFB 20,16,8
0BA3 920C950C 4840          DEFW P16R6,I16R6
0BA7 31365238 4850          DEFM '16R8'
0BAB 141008   4860          DEFB 20,16,8
0BAE 920C950C 4870          DEFW P16R8,I16R8
0BB2 32305234 4880          DEFM '20R4'
0BB6 181408   4890          DEFB 24,20,8
0BB9 00000000 4900          DEFW 0,0
0BBD 32305236 4910          DEFM '20R6'
0BC1 181408   4920          DEFB 24,20,8
0BC4 00000000 4930          DEFW 0,0
0BC8 32305238 4940          DEFM '20R8'
0BCC 181408   4950          DEFB 24,20,8
0BCF 00000000 4960          DEFW 0,0
0BD3 31304838 4970          DEFM '10H8'
0BD7 140A08   4980          DEFB 20,10,8
0BDA 980CA90C 4990          DEFW P10H8,I10H8
0BDE 31304C38 5000          DEFM '10L8'
0BE2 140A08   5010          DEFB 20,10,8
0BE5 980CA90C 5020          DEFW P10L8,I10L8
0BE9 31324836 5030          DEFM '12H6'
0BED 00000000 5040          DEFW 0,0
0BF1 31324C36 5050          DEFM '12L6'
0BF5 140C06   5060          DEFB 20,12,6
0BF8 00000000 5070          DEFW 0,0
0BFC 140E04   5080          DEFB 20,14,4
0BFF 00000000 5090          DEFW 0,0
0C03 31344C38 5100          DEFM '14L8'
0C07 140E04   5110          DEFB 20,14,4
0C0A 00000000 5120          DEFW 0,0
0C0E 31364832 5130          DEFM '16H2'
0C12 141002   5140          DEFB 20,16,2
0C15 00000000 5150          DEFW 0,0
0C19 31364C32 5160          DEFM '16L2'
0C1D 141002   5170          DEFB 20,16,2
0C20 00000000 5180          DEFW 0,0
0C24 31364331 5190          DEFM '16C1'
0C28 141002   5200          DEFB 20,16,2
0C2B 00000000 5210          DEFW 0,0
0C2F 32304331 5220          DEFM '20C1'
0C33 181402   5230          DEFB 24,20,2
0C36 00000000 5240          DEFW 0,0
0C3A 31344C38 5250          DEFM '14L8'
0C3E 140E08   5260          DEFB 20,14,8
0C41 00000000 5270          DEFW 0,0
```

```
                    **** PAL-Betriebssystem ***

0C45 31364C36 5280          DEFM '16L6'
0C49 141006   5290          DEFB 20,16,6
0C4C 00000000 5300          DEFW 0,0
0C50 31384C34 5310          DEFM '18L4'
0C54 141204   5320          DEFB 20,18,4
0C57 00000000 5330          DEFW 0,0
0C5B 32304C32 5340          DEFM '20L2'
0C5F 141402   5350          DEFB 20,20,2
0C62 00000000 5360          DEFW 0,0
0C66 31364C38 5370          DEFM '16L8'
0C6A 141008   5380          DEFB 20,16,8
0C6D 00000000 5390          DEFW 0,0
0C71 32304C38 5400          DEFM '20L8'
0C75 181408   5410          DEFB 24,20,8
0C78 00000000 5420          DEFW 0,0
0C7C 31365834 5430          DEFM '16X4'
0C80 141008   5440          DEFB 20,16,8
0C83 00000000 5450          DEFW 0,0
0C87 31364134 5460          DEFM '16A4'
0C8B 141008   5470          DEFB 20,16,8
0C8E 00000000 5480          DEFW 0,0
              5490 ;
              5499 ;
              5500 ; Produkt Line und Input Line Listen
              5501 ; Aufbau: von,bis,von,bis.......,END
              5502 ;         Innerhalb der Liste ist Bit 7
              5503 ;         immmer gesetzt
              5504 ;
0C92 0C92     5510 P16R4  EQU  $
0C92 0C92     5520 P16R6  EQU  $
0C92 0C92     5530 P16R8  EQU  $
0C92 80BF     5540        DEFB #80,#BF
0C94 00       5550        DEFB END
              5560 ;
              5600 ;
0C95 0C95     5610 I16R4  EQU  $
0C95 0C95     5620 I16R6  EQU  $
0C95 0C95     5630 I16R8  EQU  $
0C95 809F     5640        DEFB #80,#9F
0C97 00       5650        DEFB END
              5660 ;
              5670 ;
0C98 0C98     5680 P10H8  EQU  $
0C98 0C98     5690 P10L8  EQU  $
0C98 80818889 5700        DEFB #80,#81,#88,#89,#90,#91,#98,#99
     90919899
0CA0 A0A1A8A9 5710        DEFB #A0,#A1,#A8,#A9,#B0,#B1,#B8,#B9
     B0B1B8B9
0CA8 00       5720        DEFB END
              5730 ;
0CA9 0CA9     5740 I10H8  EQU  $
0CA9 0CA9     5750 I10L8  EQU  $
0CA9 80858889 5760        DEFB #80,#85,#88,#89,#8C,#8D,#90,#91
     8C8D9091
0CB1 94959899 5770        DEFB #94,#95,#98,#99,#9C,#9F
     9C9F
0CB7 00       5780        DEFB END
```

**** PAL-Betriebssystem ***

ZEAP Z80 Assembler - Symbol Table

2010H	0390	BBUFS	0007H	0490	BEL
200CH	0770	BEND	08A1H	1450	BERROR
0990H	2670	BLOWN	2008H	0750	BST
200AH	0760	BZ	08BFH	1600	BZ1
0B80H	4620	C1	0004H	0550	CDEL
08B7H	1520	CHBUFZ	000EH	0540	CINS
0013H	0620	CL	0B8AH	4680	CRLF
0B7EH	4610	CSC	000BH	0600	CUD
0008H	0580	CUL	000CH	0590	CUP
0010H	0570	CUR	007FH	0520	DEL
0B21H	4280	DISP	000BH	0660	DLEN
0127H	0280	EDIT	0000H	0460	END
001BH	0500	ESC	08D8H	1940	F1
0ADEH	3960	FOUND	08CDH	1830	FUSE
09BDH	3020	HELP	0012H	0610	HOME
0D60H	0115	HPACK	0CA9H	5740	I10H8
0CA9H	5750	I10L8	0C95H	5610	I16R4
0C95H	5620	I16R6	0C95H	5630	I16R8
0112H	0210	INCHR	0001H	0470	IND
0115H	0220	INDAT	0100H	0150	INIT
0B71H	4560	INMELD	000BH	0530	INS
0100H	0140	IOPACK	09A9H	2820	IPLT
08C9H	1790	JMON	0A9AH	3710	L1
0A8FH	3650	L2	0B4CH	4480	LEND
08C9H	1750	LOAD	011BH	0240	MOD
0400H	0330	MON	012AH	0290	MSKTS
0A32H	3270	NEW	0909H	2040	NEWPRL
0ABAH	3830	NEXT	093EH	2170	NIL
0957H	2300	NIL0	095DH	2340	NIL1
0971H	2470	NIL3	096EH	2440	NIL4
0980H	2560	NIL5	098AH	2630	NIL6
099CH	2750	NIL7	0AC4H	3890	NOPAL
0999H	2740	NPRLC	0106H	0170	OTCHR
010CH	0190	OTSTR	0121H	0260	OUTA
0124H	0270	OUTHL	0C98H	5680	P10H8
0C98H	5690	P10L8	0C92H	5510	P16R4
0C92H	5520	P16R6	0C92H	5530	P16R8
0AE4H	4010	PALASM	0D60H	0120	PALPRG
2000H	0710	PALT	08C9H	1730	PLE
0B2AH	4330	PRLM	0B0FH	4190	PRO1
0B1AH	4230	PRO2	0A5DH	3420	PT
23FFH	0380	RAM2E	2000H	0370	RAM2S
237FH	0440	RAMTOP	0002H	0480	REP
08C9H	1740	SAVE	01 8H	0230	SCAN
0A85H	3520	SEARCH	0080H	0430	STACK
0800H	0890	START	0977H	2510	STERN
0B91H	4720	TABST	2020H	0400	TBUFS
010FH	0200	TDEL	2006H	0740	TEND
0018H	0650	TLEN	09ABH	2900	TMIL
2002H	0720	TST	2004H	0730	TZ
0861H	1060	WS1	0813H	0980	WSM
011EH	0250	XFUNCT	0109H	0180	XOTCHR
0103H	0160	XOTSTR			

*** PALPGM- Listing ***

ZEAP Z80 Assembler - Source Listing

```
0D20            0010 START  ORG  #0D20
                0020 ;*****************************************
                0030 ;*                    === PALPGM ===     *
                0040 ;* Steuerung  der Hardware nach Daten    *
                0050 ;* aus der Frogrammiermatrix             *
                0060 ;*                                       *
                0070 ;*****************************************
                0080 ;* Abschlußarbeit M.Boretius P.Zecherle  *
                0090 ;*                                       *
                0100 ;* Autor: M.Boretius          30.05.83   *
                0110 ;*****************************************
                0120 ;
                0130 ;      Merkzellen
0D20 1000       0140 VERCNT EQU  #1000; Zähler für progr.versuch
0D20 1001       0150 VERADR EQU  #1001 ;akt.L oder akt.H adresse
0D20 1003       0160 AKLTBT EQU  #1003 ; Testbyte für aktiv L
0D20 1004       0170 AKHTBT EQU  #1004 ; Testbyte für aktiv H
0D20 1005       0180 TABA   EQU  #1005; 3 Bytes Arbeitbereich
                0190 ;
0D20 2000       0200 PALT   EQU  #2000; Konstantentabelle
                0210 ;
                0220 ;
                0230 ;
                0240 ; Doppelregister IX zeigt auf PRL-tab.
                0250 ; Doppelregister IY zeigt auf IPL-tab.
                0260 ;
                0270 ;
                0280 ;
                0290 ;
                0300 ;
                0310 ;  Byte B3 zum Schalten der Spannungen:
                0320 ;   00    alle V ein
                0330 ;   01    V2,V3  ein
                0340 ;   02    V1,V3  ein
                0350 ;   03    V3     ein
                0360 ;   04    V1,V2  ein
                0370 ;   05    V2     ein
                0380 ;   06    V1     ein
                0390 ;   07    alle V aus
                0400 ;  Bytes B1,B2 zum setzen der V-pegel:
                0410 ;  B1 : 50H select   V1 = Vcc
                0420 ;       60H select   V2 = rechte Seite
                0430 ;       70H select   V3 = linke Seite& P13
                0440 ;       80H deselect V1-V3
                0450 ;  B2 : Werte von 00H - FFH
                0460 ;
                0470 ;
                0480 ;-----------------------------------------
```

*** PALPGM- listing ***

```
                 0480 ;-----------------------------------------------------
                 0490 ;        VORBEREITUNG
                 0500 ;-----------------------------------------------------
0D20 DD2A0020    0510          LD   IX,(PALT); Wird akt.H oder akt.L prog?
0D24 DD7E02      0520          LD   A,(IX+2) ; H lesen aus (PALT)
0D27 FE48        0530          CP   #48   ; nur aktiv H-PAL entscheid
0D29 2805        0540          JR   Z,HTYP
0D2B 210310      0550          LD   HL,AKLTBT;Merkzellenadresse laden
0D2E 1803        0560          JR   LV
0D30 210410      0570 HTYP     LD   HL,AKHTBT;Merkzellenadresse laden
0D33 220110      0580 LV       LD   (VERADR),HL
                 0590 ;
0D36 DD210810    0600          LD   IX,#1008  ;Start der PRL-Tabelle
0D3A FD214810    0610          LD   IY,#1048  ;Start der IPL-Tabelle
0D3E AF          0620          XOR  A
0D3F DD77F8      0630          LD   (IX-8),A  ; Zähler = 0
0D42 CD630E      0640          CALL INITA        ; Portinitialisierung
0D45 CD5A0E      0650          CALL REEX         ; Reset expander
0D48 CD150E      0660          CALL VOLT
0D4B 600007      0670          DEFB #60,#00,#07 ; alle spannungen aus.
0D4E C9          0680          RET  ; Systemmeldung : ***Vpp Ein***
                 0690 ;-----------------------------------------------------
                 0700 ;          START MIT Vpp EIN
                 0710 ;-----------------------------------------------------
0D4F CD150E      0720          CALL VOLT        ; Versorgungsspannung EIN
0D52 500B06      0730          DEFB #50,#0B,#06 ; V1 = 5V
                 0740 ;
                 0750 ; Hauptprogrammschleife
0D55 CD420E      0760 HLOOP    CALL OUT10        ; Deselct all
0D58 BB0F        0770          DEFW DESTAB       ; OD/CLK=>L,I=>HH,
0D5A 1E10        0780          DEFB #1E,#10      ; O,L/R,A=>Z
0D5C CD150E      0790          CALL VOLT
0D5F 708002      0800          DEFB #70,#80,#02 ; V3 = 11,5V
0D62 CD150E      0810          CALL VOLT
0D65 60BB00      0820          DEFB #60,#BB,#00 ; V2 = 11,5V
                 0830 ;
                 0840 ; Programmendeabfrage
0D68 DD7E00      0850          LD   A,(IX)
0D6B FE00        0860          CP   #00;alle Prod.lines abgearbeitet?
0D6D 2810        0870          JR   Z,END; programmierung abgeschlossen
0D6F FD7E00      0880          LD   A,(IY)
0D72 FE00        0890          CP   #00;erste Prod.line abgearbeitet?
0D74 CA790D      0900          JP   Z,ZEIGER ; beide Zeiger neu setzen
0D77 180D        0910          JR   TAKT ; Schmelzen einer Sicherung
0D79 DD23        0920 ZEIGER   INC  IX ; Neuer PRL-Wert
0D7B FD23        0930          INC  IY ; Neuer IPL-Wert
0D7D 18D6        0940          JR   HLOOP
                 0950 ;
0D7F CD150E      0960 END      CALL VOLT
0D82 600007      0970          DEFB #60,#00,#07 ; Alle Spannungen AUS
0D85 C9          0980          RET  ; Systemmeldung:*** Progr.beendet***
                 0990 ;        Systemmeldung:***    FEHLER    ***
                 1000 ;
                 1010 ;-----------------------------------------------------
```

*** PALPGM- Listing ***

```
                1010 ;-----------------------------------------------------
                1020 ;
                1030 ; Hauptgruppenentscheidung
0D86 08         1040 TAKT    EX    AF,AF'          ; registertausch
0D87 DD7E00     1050         LD    A,(IX)          ; teste PRL-Nr.
0D8A FE9F       1060         CP    #9F             ; letzte PRL in 1.HG
0D8C 08         1070         EX    AF,AF'          ; C & Z - flag gerettet
                1080 ;-----------------------------------------------------
                1090 ; Start Taktdiagramm
                1100 ;-----------------------------------------------------
0D8D CDF60D     1110         CALL  0D              ; select Hauptgruppe
0D90 CD6A0E     1120         CALL  IXSEL           ; select Inputline
0D93 CD990E     1130         CALL  LR              ;   II        II
0D96 CDC00E     1140         CALL  AXSEL           ; select Productline
0D99 CD150E     1150         CALL  VOLT
0D9C 508000     1160         DEFB  #50,#80,#00 ; V1 = 11,5V
0D9F CDF10E     1170         CALL  PGM             ; Programmierpuls
0DA2 CD150E     1180         CALL  VOLT
0DA5 500400     1190         DEFB  #50,#04,#00 ; V1 = 6,5V
0DA8 CD280F     1200         CALL  VERIFY          ;Kehrt mit 1003/1004
                1210 ;                    zurück
0DAB 210110     1220         LD    HL,VERADR
0DAE 7E         1230         LD    A,(HL)     ; verifyergebnis
0DAF CB45       1240         BIT   0,L      ; (L)steht der progr.Baustein
                1250 ;
0DB1 2806       1260         JR    Z,HPAL
0DB3 FE00       1270 LPAL    CP    #00 ;LPAL korrekt programmiert?
0DB5 2811       1280         JR    Z,VERFY2; Ja
0DB7 1804       1290         JR    VERSL1
0DB9 FEC0       1300 HPAL    CP    #C0 ;HPAL korrekt programmiert?
0DBB 280B       1310         JR    Z,VERFY2; Ja
0DBD 2B         1320 VERSL1  DEC   HL       ; nein, versuchszähler
0DBE 34         1330         INC   (HL)     ; erhöhen
0DBF 3E05       1340         LD    A,#5
0DC1 BE         1350         CP    (HL)       ; schon 5mal versucht?
0DC2 2091       1360         JR    NZ,HLOOP; neuer versuch an glei.Stelle
0DC4 CBFE       1370         SET   7,(HL) ; Fehlerbit in 1000
0DC6 18B7       1380         JR    END
                1390 ;
0DC8 CD150E     1400 VERFY2  CALL  VOLT
0DCB 500400     1410         DEFB  #50,#04,#00 ; V1 = 4,5V
0DCE CD280F     1420         CALL  VERIFY
                1430 ;
0DD1 210110     1440         LD    HL,VERADR
0DD4 7E         1450         LD    A,(HL)     ; verifyergebnis
0DD5 CB45       1460         BIT   0,L      ; (L)steht der progr.Baustein
                1470 ;
```

*** PALPGM- listing ***

```
0DD7 2806      1480          JR   Z,HPAL2
0DD9 FE00      1490 LPAL2    CP   #00 ;LPAL korrekt programmiert?
0DDB 2812      1500          JR   Z,NEWIPL; Ja
0DDD 1804      1510          JR   VERSL2
0DDF FEC0      1520 HPAL2    CP   #C0 ;HPAL korrekt programmiert?
0DE1 280C      1530          JR   Z,NEWIPL; Ja
0DE3 2B        1540 VERSL2   DEC  HL      ; nein, versuchszähler
0DE4 34        1550          INC  (HL)    ; erhöhen
0DE5 3E05      1560          LD   A,#5
0DE7 BE        1570          CP   (HL)      ; schon 5mal versucht?
0DE8 C2550D    1580          JP   NZ,HLOOP; neuer versuch an glei.Stelle
0DEB CBFE      1590          SET  7,(HL) ; Fehlerbit in 1000
0DED 1890      1600          JR   END
               1610 ;
0DEF AF        1620 NEWIPL   XOR  A
0DF0 77        1630          LD   (HL),A
0DF1 FD23      1640          INC  IY
0DF3 C3550D    1650          JP   HLOOP
               1660 ;-----------------------------------------------------
               1670 ;
               1680 ;
               1690 ;
               1700 ;
               1710 ;-----------------------------------------------------
               1720 ;  UNTERPROGRAMME
               1730 ;-----------------------------------------------------
               1740 ; Unterprogramm OD : RA,RB
0DF6 08        1750 OD       EX   AF,AF'
0DF7 3814      1760          JR   C,OD1HG ; Hauptgruppenentscheidung
0DF9 2812      1770          JR   Z,OD1HG
0DFB 08        1780 OD2HG    EX   AF,AF'
0DFC 3E3B      1790          LD   A,#3B ;
0DFE 47        1800          LD   B,A   ; 2 Steuercodes für VHH an
0DFF 3E13      1810          LD   A,#13 ; Pin 11 des 20-pol.PAL
0E01 D310      1820 ODHG     OUT  (#10),A
0E03 CD350E    1830          CALL STROBE
0E06 78        1840          LD   A,B
0E07 D310      1850          OUT  (#10),A ; senden
0E09 CD350E    1860          CALL STROBE
0E0C C9        1870          RET
0E0D 08        1880 OD1HG    EX   AF,AF'
0E0E 3E48      1890          LD   A,#48 ; 2 Steuercodes für Pin 1
0E10 47        1900          LD   B,A   ; laden
0E11 3E20      1910          LD   A,#20
0E13 18EC      1920          JR   ODHG
               1930 ;-----------------------------------------------------
```

*** PALPGM- Listing ***

```
                1930 ;
                1940 ; Unterprogramm VOLT : RA, RB, RC, RHL
0E15 E3         1950 VOLT   EX   (SP),HL      ;-----------------------------
0E16 7E         1960        LD   A,(HL)       ;      hier werden
0E17 D310       1970        OUT  (#10),A      ;      alle drei
0E19 23         1980        INC  HL           ;      Parameter
0E1A 7E         1990        LD   A,(HL)       ;      zur steuerung
0E1B D311       2000        OUT  (#11),A      ;      der Spannungs-
0E1D 23         2010        INC  HL           ;      quellen ausge-
0E1E 7E         2020        LD   A,(HL)       ;      geben.
0E1F 23         2030        INC  HL           ;
0E20 E3         2040        EX   (SP),HL      ;
0E21 CD350E     2050        CALL STROBE
0E24 07         2060        RLCA ; Bitposition zum
0E25 07         2070        RLCA ; umschalten setzen
0E26 07         2080        RLCA
0E27 C5         2090        PUSH BC
0E28 47         2100        LD   B,A
0E29 3EC7       2110        LD   A,#C7        ; vorbereitung der
0E2B B0         2120        OR   B            ; Masken
0E2C 4F         2130        LD   C,A
0E2D DB02       2140        IN   A,(#02)      ; alten Steuerwert
0E2F B0         2150        OR   B            ; lesen und maskieren
0E30 A1         2160        AND  C
0E31 C1         2170        POP  BC
0E32 D302       2180        OUT  (#02),A      ; nur die Bits 3,4,5
0E34 C9         2190        RET  ; werden verändert
                2200 ;-----------------------------------------------------
                2210 ; Unterprogramm STROBE
0E35 F5         2220 STROBE PUSH AF
0E36 DB02       2230        IN   A,(#02)      ; alten Steuerportwert
0E38 E6FE       2240        AND  #FE          ; lesen und maskieren
0E3A D302       2250        OUT  (#02),A      ; Bit 0 ist aktive Flanke
0E3C CBC7       2260        SET  0,A          ; LS 155 sperren
0E3E D302       2270        OUT  (#02),A
0E40 F1         2280        POP  AF
0E41 C9         2290        RET
                2300 ;-----------------------------------------------------
                2310 ; Unterprogramm OUT10: R A,BC,DE,HL
0E42 00         2320 OUT10  NOP
0E43 E3         2330        EX   (SP),HL
0E44 7E         2340        LD   A,(HL)
0E45 5F         2350        LD   E,A      ; H- Adressbyte
0E46 23         2360        INC  HL
0E47 7E         2370        LD   A,(HL)
0E48 57         2380        LD   D,A      ; L- Adressbyte
0E49 23         2390        INC  HL
0E4A 7E         2400        LD   A,(HL)
0E4B 47         2410        LD   B,A      ; Tabellenlänge
```

*** PALPGM- listing ***

```
0E4C 23        2420          INC  HL
0E4D 7E        2430          LD   A,(HL)
0E4E 4F        2440          LD   C,A      ; Portadresse
0E4F 23        2450          INC  HL
0E50 E3        2460          EX   (SP),HL
0E51 EB        2470          EX   DE,HL
0E52 EDA3      2480 OUTL     OUTI ; Einzelbyteausgabe
0E54 CD350E    2490          CALL STROBE
0E57 20F9      2500          JR   NZ,OUTL
0E59 C9        2510          RET
               2520 ;-----------------------------------------------------
               2530 ; Unterprogramm INIT
0E5A 3E39      2540 REEX     LD   A,#39     ; reset Expander
0E5C D302      2550          OUT  (#02),A
0E5E 3E3D      2560          LD   A,#3D
0E60 D302      2570          OUT  (#02),A
0E62 C9        2580          RET
0E63 3E80      2590 INITA    LD   A,#80     ; alles Ausgänge
0E65 D313      2600          OUT  (#13),A
0E67 D303      2610          OUT  (#03),A      ; Zweiter Portbaustein
0E69 C9        2620          RET
               2630 ;-----------------------------------------------------
               2640 ; Unterprogramm IXSEL: R A,BC,DE,HL
0E6A 21D90F    2650 IXSEL    LD   HL,IOI9
0E6D 0E10      2660          LD   C,#10    ; Portadresse
0E6F CD7C0E    2670          CALL HLPOS ; Zeiger HL positionieren
0E72 0602      2680          LD   B,#2     ; 2 Bytes ausgeben
0E74 EDA3      2690 LIXSEL   OUTI ; einzelbyteausgabe
0E76 CD350E    2700          CALL STROBE
0E79 C8        2710          RET  Z
0E7A 18F8      2720          JR   LIXSEL
               2730 ;
0E7C 1600      2740 HLPOS    LD   D,00 ; Initialisieren Summand*
0E7E 3E03      2750          LD   A,#3       ; Ausblendmaske
0E80 FD5E00    2760          LD   E,(IY)     ;*register DE
0E83 CBBB      2770          RES  7,E
0E85 A3        2780          AND  E
0E86 19        2790          ADD  HL,DE     ; Zeiger in Unterruppe
0E87 FE03      2800          CP   #3         ; Testen bits 0 und 1
0E89 280C      2810          JR   Z,DECHL
0E8B FE02      2820          CP   #2
0E8D 2807      2830          JR   Z,DECHL2
0E8F FE01      2840          CP   #1
0E91 2801      2850          JR   Z,INCHL
0E93 C9        2860          RET  ; bei RET ist der
0E94 23        2870 INCHL    INC  HL
0E95 C9        2880          RET  ; Zeiger HL korrekt
0E96 2B        2890 DECHL2   DEC  HL
0E97 2B        2900 DECHL    DEC  HL
0E98 C9        2910          RET  ; positioniert
               2920 ;-----------------------------------
```

*** PALPGM- Listing ***

```
                  2930 ; Unterprogramm LR:  RA, RB
0E99 FD7E00       2940 LR      LD    A,(IY)   ; IPL
0E9C 47           2950         LD    B,A
0E9D 08           2960         EX    AF,AF'     ; Hauptgruppentest
0E9E 380F         2970         JR    C,LRHG1    ; es werden die ent-
0EA0 280D         2980 LRHG2   JR    Z,LRHG1    ; sprechenden Steuer-
0EA2 08           2990         EX    AF,AF'     ;codes geladen und
0EA3 CB48         3000         BIT   1,B       ; mit SWLR ausgegeben
0EA5 2804         3010         JR    Z,LRZ2
0EA7 3E0B         3020 LRHH2   LD    A,#0B ; PIN 19 => HH
0EA9 180F         3030         JR    SWLR
0EAB 3E03         3040 LRZ2    LD    A,#03  ; PIN 19 => Z
0EAD 180B         3050         JR    SWLR
                  3060 ;
0EAF 08           3070 LRHG1   EX    AF,AF'
0EB0 CB48         3080         BIT   1,B
0EB2 2804         3090         JR    Z,LRZ1
0EB4 3E3C         3100 LRHH1   LD    A,#3C ; PIN 12 => HH
0EB6 1802         3110         JR    SWLR
0EB8 3E34         3120 LRZ1    LD    A,#34  ; PIN 12 => Z
0EBA D310         3130 SWLR    OUT   (#10),A
0EBC CD350E       3140         CALL  STROBE
0EBF C9           3150         RET
                  3160 ;-----------------------------------------------------------
                  3170 ; Unterprogramm AXSEL: RA,RB,RC,RHL
0EC0 08           3180 AXSEL   EX    AF,AF'
0EC1 380F         3190         JR    C,AXHG1      ; Hauptgruppentest
0EC3 280D         3200 AXHG2   JR    Z,AXHG1
0EC5 08           3210         EX    AF,AF'       ;
0EC6 AF           3220         XOR   A            ; anfangswert für 2. HG
0EC7 CDD70E       3230 AXLOP   CALL  ATAB ; 3 Steuerbytes errechnen
0ECA CD420E       3240         CALL  OUT10       ; 3 errechnete Bytes
0ECD 0510         3250         DEFW  TABA        ; ausgeben
0ECF 0310         3260         DEFB  #03,#10
0ED1 C9           3270         RET
0ED2 08           3280 AXHG1   EX    AF,AF'
0ED3 3E35         3290         LD    A,#35        ; anfangswert für 1.HG
0ED5 18F0         3300         JR    AXLOP
                  3310 ;
0ED7 210510       3320 ATAB    LD    HL,TABA ; Arbeitsbereich
0EDA 0603         3330         LD    B,3   ; drei bits testen und setzen
0EDC DD4E00       3340         LD    C,(IX)  ;testbyte FRL
0EDF CB51         3350 AXLOP1  BIT   2,C
0EE1 2804         3360         JR    Z,AXNUL
0EE3 CBDF         3370         SET   3,A       ; AX => HH
0EE5 1802         3380         JR    AXNXT      ; nächstes Byte errechnen
0EE7 CB9F         3390 AXNUL   RES   3,A ; AX => L
                  3400 ;
0EE9 77           3410 AXNXT   LD    (HL),A ; erst einmal abspeichern
0EEA 23           3420         INC   HL           ; nächster freier Platz
0EEB CB01         3430         RLC   C            ; Testbyte bitposition 2
0EED 3C           3440         INC   A
0EEE 10EF         3450         DJNZ  AXLOP1
0EF0 C9           3460         RET
                  3470 ;-----------------------------------------------------------
```

*** PALPGM- Listing ***

```
                3470 ;
                3480 ; Unterprogramm PGM: RA,RBC,RDE,RHL
0EF1 08         3490 PGM     EX   AF,AF'
0EF2 3807       3500         JR   C,PGM1HG   ; Hauptgruppentest
0EF4 2805       3510 PGM2HG  JR   Z,PGM1HG
0EF6 08         3520         EX   AF,AF'
0EF7 063C       3530         LD   B,#3C      ; Startwert, Bit 0 & 1 = 0
0EF9 1803       3540         JR   PGM0
0EFB 08         3550 PGM1HG  EX   AF,AF'
0EFC 0608       3560         LD   B,#08      ; Startwert, Bit 0 & 1 = 0
0EFE DD7E00     3570 PGM0    LD   A,(IX); Testbits 2 & 3 von PRL-Nr
0F01 2F         3580         CPL
0F02 1F         3590         RRA
0F03 1F         3600         RRA
0F04 1F         3610         RRA
0F05 E603       3620         AND  #03
0F07 80         3630         ADD  A,B   ; 0-selectbyte für HH
0F08 47         3640         LD   B,A    ; vorbereitendes Registerladen
0F09 CB9F       3650         RES  3,A
0F0B 4F         3660         LD   C,A
0F0C DB12       3670         IN   A,(#12)
0F0E E6FE       3680         AND  #FE
0F10 57         3690         LD   D,A
0F11 CBC7       3700         SET  0,A
0F13 5F         3710         LD   E,A
                3720 ;  ***  Start programzykus ***
0F14 78         3730         LD   A,B
0F15 D310       3740         OUT  (#10),A  ; sel.byte für HH
0F17 7A         3750         LD   A,D       ;-----------------------
0F18 D302       3760         OUT  (#02),A  ; aktiver Strobe  I
0F1A 7B         3770         LD   A,E
0F1B D302       3780         OUT  (#02),A
0F1D 79         3790         LD   A,C
0F1E D310       3800         OUT  (#10),A  ; deselect => L
0F20 7A         3810         LD   A,D
0F21 D302       3820         OUT  (#02),A  ; aktiver Strobe  I
0F23 7B         3830         LD   A,E       ;-----------------------
0F24 D302       3840         OUT  (#02),A
0F26 48         3850         LD   C,B   ; selectbyte retten
0F27 C9         3860         RET
                3870 ;-----------------------------------------------
                3880 ; Unterprogramm VERYFY
0F28 CD300F     3890 VERIFY  CALL INPEG
0F2B 3E80       3900         LD   A,#80
0F2D D313       3910         OUT  (#13),A   ; Port B => ausgabe
0F2F C9         3920         RET
                3930 ;
```

*** PALPGM- listing ***

```
0F30 1605      3940 INPEG  LD   D,#5; shiftzähler Startwert
0F32 CD760F    3950        CALL TEST  ;setztRE,benötigt RC(PGM-Byte)
0F35 3E06      3960        LD   A,#06     ;Vref = 2,4 V
0F37 D310      3970        OUT  (#10),A
0F39 CD350E    3980        CALL STROBE
0F3C CD8E0F    3990        CALL CLCK;schaltet den aktivierungsimpuls
0F3F 3E82      4000        LD   A,#82 ;  "eingabe"-code
0F41 D313      4010        OUT  (#13),A   ; Port B => eingang
0F43 DB11      4020        IN   A,(#11)
0F45 1002      4030 INPGL1 DJNZ INPGL2 ;TEST setzte den Zähler
0F47 1803      4040        JR   INPGL3
0F49 07        4050 INPGL2 RLCA ; positionierung auf Bit 7
0F4A 18F9      4060        JR   INPGL1
0F4C 5F        4070 INPGL3 LD   E,A  ; Bit 7 retten
               4080 ;
0F4D 1606      4090        LD   D,#6      ; Shiftzähler startwert
0F4F CD760F    4100        CALL TEST  ;RC, RE
0F52 3E0E      4110        LD   A,#0E     ; Vref = 0,4V
0F54 D310      4120        OUT  (#10),A
0F56 CD350E    4130        CALL STROBE
0F59 CD8E0F    4140        CALL CLCK ; aktivierungstakt
0F5C DB11      4150        IN   A,(#11)
0F5E 1002      4160 INPGL4 DJNZ INPGL5;TEST setzte den Zähler
0F60 1803      4170        JR   INPGL6
0F62 07        4180 INPGL5 RLCA ; Positionierung in CARRY
0F63 18F9      4190        JR   INPGL4
0F65 3805      4200 INPGL6 JR   C,SETB6
0F67 7B        4210 RESB6  LD   A,E   ; Bit 7 holen
0F68 CBB7      4220        RES  6,A       ; Verifizierungsergebnis
0F6A 1803      4230        JR   LDTBT      ; abhängig von akt L oder H
               4240 ;
0F6C 7B        4250 SETB6  LD   A,E  ; Bit 7 holen
0F6D CBF7      4260        SET  6,A       ; Verifizierungsergebnis
0F6F E6C0      4270 LDTBT  AND  #C0; abhängig von akt L oder H
0F71 210110    4280        LD   HL,(VERADR)
0F74 77        4290        LD   (HL),A
0F75 C9        4300        RET  ; vom Pegellesezyklus
               4310 ;                 mit Ergebnis in (VERADR)
               4320 ;
0F76 63        4330 TEST   LD   H,E     ;save
0F77 1E0F      4340        LD   E,#0F     ; ausblendmaske
0F79 3E3F      4350        LD   A,#3F     ; vergleichsstartwert
0F7B 0604      4360        LD   B,#4      ; schleifenzähler
0F7D B9        4370 TESTL  CP   C
0F7E 280B      4380        JR   Z,TESTNX; gefunden
0F80 3D        4390        DEC  A         ; nicht gefunden
0F81 14        4400        INC  D         ; nachher eins weiter shiften
0F82 10F9      4410        DJNZ TESTL
```

*** PALPGM- Listing ***

```
0F84 A3          4420          AND  E        ; weiter mit testwert #0B
0F85 0604        4430          LD   B,#4      ; wieder 4mal versuchen
0F87 CB9A        4440          RES  3,D      ; shifter = 1,-> nicht shiften
0F89 18F2        4450          JR   TESTL
0F8B 42          4460 TESTNX   LD   B,D;Shiftzählerendwert=Schlei-
0F8C 5C          4470          LD   E,H      ; reload///fenzähler für DJNZ
0F8D C9          4480          RET  ; ***DJNZ steht vor shift****
                 4490 ;
0F8E 08          4500 CLCK     EX   AF,AF'; Ausgabe der Tabelle zum
0F8F 3816        4510          JR   C,CLKHG1  ; schalten des aktivieruns-
0F91 2814        4520          JR   Z,CLKHG1  ; taktes in Abhängigkeit der
0F93 08          4530 CLKHG2   EX   AF,AF'; selektierten Hauptgruppe
0F94 1805        4540          JR   CKHG2
0F96 40282040    4550 CK2TAB   DEFB #40,#28,#20,#40,#28
     28
0F9B C5          4560 CKHG2    PUSH BC
0F9C D5          4570          PUSH DE
0F9D CD420E      4580          CALL OUT10
0FA0 960F        4590          DEFW CK2TAB
0FA2 0510        4600          DEFB #05,#10
0FA4 D1          4610          POP  DE
0FA5 C1          4620          POP  BC
0FA6 C9          4630          RET
0FA7 08          4640 CLKHG1   EX   AF,AF'
0FA8 1805        4650          JR   CKHG1
0FAA 331B1333    4660 CK1TAB   DEFB #33,#1B,#13,#33,#1B
     1B
0FAF C5          4670 CKHG1    PUSH BC
0FB0 D5          4680          PUSH DE
0FB1 CD420E      4690          CALL OUT10
0FB4 AA0F        4700          DEFW CK1TAB
0FB6 0510        4710          DEFB #05,#10
0FB8 D1          4720          POP  DE
0FB9 C1          4730          POP  BC
0FBA C9          4740          RET
                 4750 ;-----------------
```

*** PALPGM- listing ***

```
                4750 ;
                4760 ;          TABELLEN
                4770 ;-----------------------------------------------------------
                4780 ; Deselecttabelle DESTAB
0FBB 4028       4790 DESTAB DEFB #40,#28          ;CLK/OD => L
0FBD 2149       4800        DEFB #21,#49          ; I-------------
0FBF 224A       4810        DEFB #22,#4A          ; I
0FC1 234B       4820        DEFB #23,#4B          ; I
0FC3 244C       4830        DEFB #24,#4C          ; I  I0-I7 =>HH
0FC5 254D       4840        DEFB #25,#4D          ; I
0FC7 264E       4850        DEFB #26,#4E          ; I
0FC9 274F       4860        DEFB #27,#4F          ; I
0FCB 1038       4870        DEFB #10,#38          ; I-------------
0FCD 331B       4880        DEFB #33,#1B          ;CLK/OD => L
0FCF 3435       4890        DEFB #34,#35          ; I-------------
0FD1 3637       4900        DEFB #36,#37          ; I
0FD3 0001       4910        DEFB #00,#01          ; I  P14-P23 =>Z
0FD5 0203       4920        DEFB #02,#03          ; I
0FD7 0405       4930        DEFB #04,#05          ; I-------------
                4940 ;-----------------------------------------------------------
                4950 ; Selecttabelle IOI9
0FD9 4129       4960 IOI9   DEFB #41,#29            ; 1.OG => L
0FDB 4121       4970        DEFB #41,#21          ;       => H
0FDD 422A       4980        DEFB #42,#2A          ; 2.OG => L
0FDF 4222       4990        DEFB #42,#22          ;       => H
0FE1 432B       5000        DEFB #43,#2B          ; 3.OG => L
0FE3 4323       5010        DEFB #43,#23          ;       => H
0FE5 442C       5020        DEFB #44,#2C          ; 4.OG => L
0FE7 4424       5030        DEFB #44,#24          ;       => H
0FE9 452D       5040        DEFB #45,#2D          ; 5.OG => L
0FEB 4525       5050        DEFB #45,#25          ;       => H
0FED 462E       5060        DEFB #46,#2E          ; 6.OG => L
0FEF 4626       5070        DEFB #46,#26          ;       => H
0FF1 472F       5080        DEFB #47,#2F          ; 7.OG => L
0FF3 4727       5090        DEFB #47,#27          ;       => H
0FF5 3018       5100        DEFB #30,#18          ; 8.OG => L
0FF7 3010       5110        DEFB #30,#10          ;       => H
                5120 ;-----------------------------------------------------------
```

*** PALPGM- listing ***

ZEAP Z80 Assembler - Symbol Table

1004H	0170	AKHTBT	1003H	0160	AKLTBT
0ED7H	3320	ATAB	0ED2H	3280	AXHG1
0EC3H	3200	AXHG2	0EC7H	3230	AXLOP
0EDFH	3350	AXLOP1	0EE7H	3390	AXNUL
0EE9H	3410	AXNXT	0EC0H	3180	AXSEL
0FAAH	4660	CK1TAB	0F96H	4550	CK2TAB
0FAFH	4670	CKHG1	0F9BH	4560	CKHG2
0F8EH	4500	CLCK	0FA7H	4640	CLKHG1
0F93H	4530	CLKHG2	0E97H	2900	DECHL
0E96H	2890	DECHL2	0FBBH	4790	DESTAB
0D7FH	0960	END	0D55H	0760	HLOOP
0E7CH	2740	HLPOS	0DB9H	1300	HPAL
0DDFH	1520	HPAL2	0D30H	0570	HTYP
0FD9H	4960	IOI9	0E94H	2870	INCHL
0E63H	2590	INITA	0F30H	3940	INPEG
0F45H	4030	INPGL1	0F49H	4050	INPGL2
0F4CH	4070	INPGL3	0F5EH	4160	INPGL4
0F62H	4180	INPGL5	0F65H	4200	INPGL6
0E6AH	2650	IXSEL	0F6FH	4270	LDTBT
0E74H	2690	LIXSEL	0DB3H	1270	LPAL
0DD9H	1490	LPAL2	0E99H	2940	LR
0EAFH	3070	LRHG1	0EA0H	2980	LRHG2
0EB4H	3100	LRHH1	0EA7H	3020	LRHH2
0EB8H	3120	LRZ1	0EABH	3040	LRZ2
0D33H	0580	LV	0DEFH	1620	NEWIPL
0DF6H	1750	OD	0E0DH	1880	OD1HG
0DFBH	1780	OD2HG	0E01H	1820	ODHG
0E42H	2320	OUT10	0E52H	2480	OUTL
2000H	0200	PALT	0EF1H	3490	PGM
0EFBH	3550	PGM1HG	0EF4H	3510	PGM2HG
0EFEH	3570	PGM0	0E5AH	2540	REEX
0F67H	4210	RESB6	0F6CH	4250	SETB6
0D20H	0010	START	0E35H	2220	STROBE
0EBAH	3130	SWLR	1005H	0180	TABA
0D86H	1040	TAKT	0F76H	4330	TEST
0F7DH	4370	TESTL	0F8BH	4460	TESTNX
1001H	0150	VERADR	1000H	0140	VERCNT
0DC8H	1400	VERFY2	0F28H	3890	VERIFY
0DBDH	1320	VERSL1	0DE3H	1540	VERSL2
0E15H	1950	VOLT	0D79H	0920	ZEIGER

*** MATRIX- listing ***

ZEAP Z80 Assembler - Source Listing

```
06D0            0010 START  ORG  #06D0
                0020 ;*****************************************
                0030 ;*                  === MATRIX ===       *
                0040 ;* Erstellen der Programmiermatrix       *
                0050 ;*                                       *
                0060 ;*****************************************
                0070 ;* Abschlußarbeit M.Boretius F.Zecherle  *
                0080 ;*                                       *
                0090 ;* Autor: M.Boretius          25.05.83   *
                0100 ;*****************************************
                0110 ;
                0120 ; Das Programm erzeugt zwei Werte (IPL&
                0130 ; PRL), die vorher auf Vorhandensein in
                0140 ; der Konstantentabelle untersucht wer-
                0150 ; den (MASK-routine). Der IPL-wert wird
                0160 ; dann  über  die  hex-variablentabelle
                0170 ; geschickt (HL), die im Folgenden CPL.
                0180 ; genannt wird. Eine CPL ( Compareline)
                0190 ; ist immer der Teil zwischen zwei Ende-
                0200 ; Kennzeichen und steht für die auszu-
                0210 ; schließenden IPLs einer zugehörigen
                0220 ; Produktlinenummer.
                0230 ; Wurde Übereinstimmung festgestellt,
                0240 ; wird der erzeugte Wert mit einem neuen
                0250 ; überschrieben und die CPL von Vorn
                0260 ; durchsucht, bis keine gültigen IPLs
                0270 ; mehr vorhanden sind ( Test Bit 2,B)
                0280 ;
                0290 ;-----------------------------------------
                0300 ; Konstantendefinitionen
06D0 1000       0310 MZ     EQU  #1000  ; Anfangsadr. aktuelle CPL
                0320 ;       #1002    Endadresse aktuelle CPL
06D0 2000       0330 PALT   EQU  #2000 ; steht die adr.akt.PAL
06D0 0100       0340 IOPACK EQU  #100
06D0 012A       0350 MASK   EQU  IOPACK+#2A
                0360 ;#1004 wird der umzuwand.ASCII gesichert
                0370 ;
                0380 ; Initialisierung der Tabellenzeiger
                0390 ;
06D0 212020     0400        LD   HL,#2020 ; Variablentabelle
06D3 FD214810   0410        LD   IY,#1048 ; Inputlinetabelle
06D7 DD210810   0420        LD   IX,#1008 ; Produktlinetabelle
                0430 ;
06DB 1E80       0440        LD   E,#80     ; PRL-zähler-anfangswert
06DD 0600       0450        LD   B,#00    ; Flagregister
                0460 ;
```

*** MATRIX- listing ***

```
                0470 ;-----------------------------------------------------
                0480 ; Hauptprogramm
                0490 ;-----------------------------------------------------
06DF CD7E07     0500 LOOP   CALL PRL; erste PRL-Nr.produzieren
06E2 CB58       0510        BIT  3,B      ; keine mehr da? (MASK)
06E4 C2F907     0520        JP   NZ,END    ; Matrix ist erzeugt
06E7 CB90       0530        RES  2,B      ; Endeflag aus IPL-erzeugung
06E9 1680       0540        LD   D,#80     ; IPL-zähler-anfangswert
06EB CD5607     0550        CALL IPL     ; erste IPL-Nr. produzieren
                0560 ;
06EE 3E00       0570 HLOOP  LD   A,#00
06F0 BE         0580        CP   (HL)
06F1 284F       0590        JR   Z,ALL   ;erste Null, alle IPLs erzeugen
06F3 3E2A       0600        LD   A,#2A   ;ASCII für Kurzkreuz
06F5 BE         0610        CP   (HL)
06F6 2846       0620        JR   Z,CLEAR; diese PRL ignorieren
06F8 3E20       0630        LD   A,#20   ; Blank einfach überspringen
06FA BE         0640        CP   (HL)
06FB 2003       0650        JR   NZ,MPROC; jetzt gültiger Wert
06FD 23         0660        INC  HL
06FE 18EE       0670        JR   HLOOP
                0680 ;
0700 220010     0690 MPROC  LD   (MZ),HL; Startadresse CPL
0703 FD7E00     0700 PROC   LD   A,(IY); produzierten Wert holen
0706 F5         0710 CPROC  PUSH AF
0707 CDAA07     0720        CALL EX ;an dieser Stelle (HL) ASCII->Hex
070A F1         0730        POP  AF
070B BE         0740        CP   (HL) ; HL zeiger auf gewandelten Wert
070C 2005       0750        JR   NZ,LPROC;nächsten Wert aus CPL holen
                0760 ;          oder   Endekennzeichen CPL ?
070E CDF307     0770        CALL LOAD ;Wandlung rückgängig machen
0711 1819       0780        JR   NPROC ;CPLwert zum Ausschließen aus IY
                0790 ;
0713 CDF307     0800 LPROC  CALL LOAD;Wandlung rückgängig machen
0716 23         0810        INC  HL
0717 4F         0820        LD   C,A
0718 AF         0830        XOR  A    ; Endekennzeichen?
0719 BE         0840        CP   (HL)
071A 280B       0850        JR   Z,HPROC; gesamte CPL durchsucht
071C 3E20       0860        LD   A,#20   ; Blanks überspringen
071E BE         0870 XPROC  CP   (HL)
071F 2003       0880        JR   NZ,SPROC; nächster CPL-wert
0721 23         0890        INC  HL
0722 18FA       0900        JR   XPROC
0724 79         0910 SPROC  LD   A,C; mit erzeugtem identisch?
0725 18DF       0920        JR   CPROC
```

*** MATRIX- Listing ***

```
                0930 ;
0727 220210     0940 HPROC  LD   (MZ+2),HL;Start der neuen CPL für
                0950 ;                     eine neue PRL
072A FD23       0960        INC  IY
072C 00         0970 NPROC  NOP  ; neue IPL produzieren
072D CD5607     0980        CALL IPL
0730 CB50       0990        BIT  2,B            ; keine mehr da?
0732 2005       1000        JR   NZ,NEWPRL       ; neue PRL produzieren
0734 2A0010     1010        LD   HL,(MZ)         ; mit alter CPL für die
                1020 ;                     nächste  gültige IPL
0737 18CA       1030        JR   PROC
                1040 ;
0739 2A0210     1050 NEWPRL LD   HL,(MZ+2); alte CPL abgearbeitet
                1060 ;          und Startadresse neuer CPL laden
073C 180D       1070        JR   NEWIY
                1080 ;
073E 23         1090 CLEAR  INC  HL
073F 23         1100        INC  HL
0740 189D       1110        JR   LOOP
                1120 ;
0742 FD23       1130 ALL    INC  IY
0744 CD5607     1140        CALL IPL
0747 CB50       1150        BIT  2,B
0749 28F7       1160        JR   Z,ALL
                1170 ;
074B 23         1180 NEWIY  INC  HL; Variablenzeiger neu
074C DD23       1190        INC  IX      ; PRL-zeiger neu
074E AF         1200        XOR  A
074F FD7700     1210        LD   (IY),A ; Endekennzeichen für die IPLs
                1220 ;                     einer PRL
0752 FD23       1230        INC  IY     ; IPL-zeiger neu
0754 1889       1240        JR   LOOP
                1250 ;
                1260 ;-----------------------------------------------------
                1270 ; Unterprogramm IPL
                1280 ; Arbeitet mit   Startwert       die MASK-
                1290 ; routine ab.Ist der Wert innerhalb,also
                1300 ; gültig, wird er in (IY) abgelegt.
                1310 ; Dieser Wert wird in RD gesichert und
                1320 ; bei Neuaufruf von IPL und zum Weiter-
                1330 ; verarbeiten herangezogen,bis er Null
                1340 ; geworden ist. (Die IPL-nummern in der
                1350 ; Konstantentabelle, die ja MASK benutzt
                1360 ; gehen von #80 für IPL 00 bis #FF für
                1370 ; IPL 7F).                          Wenn
                1380 ; nicht, wird der Startwert incrementiert
                1390 ; und von Neuem begonnen.
                1400 ; Diese Routine "produziert" also den Wert
                1410 ; der nachher über die Compareline CPL ge-
                1420 ; schickt wird.
```

*** MATRIX- Listing ***

```
0756 7A        1430 IPL     LD   A,D
0757 DDE5      1440         PUSH IX
0759 E5        1450         PUSH HL
075A DD2A0020  1460         LD   IX,(PALT)
075E CD6707    1470         CALL IPLL1
0761 57        1480         LD   D,A
0762 E1        1490         POP  HL
0763 DDE1      1500         POP  IX
0765 14        1510         INC  D
0766 C9        1520         RET
               1530 ;
0767 DD6E09    1540 IPLL1   LD   L,(IX+9)
076A DD660A    1550         LD   H,(IX+10)
076D CD2A01    1560         CALL MASK
0770 3005      1570         JR   NC,CPYI; gültigen Wert gefunden
0772 3C        1580         INC  A      ; nicht gefunden, neuer Versuc
0773 2806      1590         JR   Z,ENDI ; jetzt am Ende
0775 18F0      1600         JR   IPLL1
               1610 ;
0777 FD7700    1620 CPYI    LD   (IY),A
077A C9        1630         RET
               1640 ;
077B CBD0      1650 ENDI    SET  2,B ; keine mehr zu produzieren
077D C9        1660         RET
               1670 ;-----------------------------------------------------
               1680 ; Unterprogramm PRL     ---  Wie IPL ---
077E 7B        1690 PRL     LD   A,E
077F FDE5      1700         PUSH IY
0781 E5        1710         PUSH HL
0782 DDE5      1720         PUSH IX
0784 DD2A0020  1730         LD   IX,(PALT)
               1740 ;
0788 CD9607    1750         CALL PRLL1
078B 5F        1760         LD   E,A
078C DDE1      1770         POP  IX
078E DD7700    1780         LD   (IX),A
0791 E1        1790         POP  HL
0792 FDE1      1800         POP  IY
0794 1C        1810         INC  E
0795 C9        1820         RET
               1830 ;
0796 DD6E07    1840 PRLL1   LD   L,(IX+7)
0799 DD6608    1850         LD   H,(IX+8)
079C CD2A01    1860         CALL MASK
079F 3005      1870         JR   NC,CPYP
07A1 3C        1880         INC  A
07A2 2803      1890         JR   Z,ENDP
07A4 18F0      1900         JR   PRLL1
               1910 ;
07A6 C9        1920 CPYP    RET
               1930 ;
07A7 CBD8      1940 ENDP    SET  3,B
07A9 C9        1950         RET
```

*** MATRIX- listing ***

```
                1960 ;-----------------------------------------------------------
                1970 ; Unterprogramm EX
                1980 ; wandelt ein- oder zweistellige ASCII
                1990 ; erlaubt formatfreie Variableneingabe
07AA 23         2000 EX      INC  HL
07AB 3E00       2010         LD   A,#00; einstelliger ASCII-wert?
07AD BE         2020         CP   (HL)
07AE 281E       2030         JR   Z,EX1; ja
07B0 3E20       2040         LD   A,#20
07B2 BE         2050         CP   (HL)
07B3 2819       2060         JR   Z,EX1; auch einstellig
                2070 ;
07B5 CDEC07     2080         CALL SAVE; den zu wandelnden Wert retten
07B8 2B         2090         DEC  HL    ; erste ASCII-stelle
07B9 7E         2100         LD   A,(HL); wandeln
07BA 17         2110         RLA
07BB 17         2120         RLA
07BC 17         2130         RLA
07BD 17         2140         RLA
07BE 23         2150         INC  HL  ; zweite ASCII-stelle
07BF CB66       2160         BIT  4,(HL)
07C1 2005       2170         JR   NZ,ZIF; 0 - 9
07C3 F5         2180         PUSH AF
07C4 CDDA07     2190         CALL ADHL ; A - F wandeln
07C7 F1         2200         POP  AF
07C8 ED67       2210 ZIF     RRD  ; 0 - 9 wandeln
07CA CBFF       2220         SET  7,A
07CC 77         2230         LD   (HL),A
07CD C9         2240         RET  ; zweistellige Zahl
                2250 ;
07CE 2B         2260 EX1     DEC  HL
07CF CDEC07     2270         CALL SAVE
07D2 CDD607     2280         CALL EX3 ;eine wandl.
07D5 C9         2290         RET  ; einstelliger Wert in Vari.tab
                2300 ;
07D6 CB66       2310 EX3     BIT  4,(HL); 0 - 9 oder A - F ?
07D8 200B       2320         JR   NZ,RES45
07DA 3E01       2330 ADHL    LD   A,#01 ; A - F wandeln
07DC 86         2340         ADD  A,(HL)
07DD CBDF       2350         SET  3,A
07DF E60F       2360         AND  #0F
07E1 CBFF       2370         SET  7,A
07E3 77         2380         LD   (HL),A
07E4 C9         2390         RET
07E5 3E0F       2400 RES45   LD   A,#0F ; einstelligen Wert wandeln
07E7 A6         2410         AND  (HL)
07E8 CBFF       2420         SET  7,A
07EA 77         2430         LD   (HL),A
07EB C9         2440         RET
```

*** MATRIX- Listing ***

```
                2450 ;-----------------------------------------------------------
                2460 ; Unterprogramm SAVE sichert zu wandelnden
07EC F5         2470 SAVE   PUSH AF ;        Wert
07ED 7E         2480        LD   A,(HL)
07EE 320410     2490        LD   (MZ+4),A
07F1 F1         2500        POP  AF
07F2 C9         2510        RET
                2520 ;
07F3 F5         2530 LOAD   PUSH AF ; läd ihn an alte Stelle
07F4 3A0410     2540        LD   A,(MZ+4) ; zurück
07F7 77         2550        LD   (HL),A
07F8 F1         2560        POP  AF
07F9 C9         2570 END    RET
                2580 ;-----------------------------------------------------------
```

ZEAP Z80 Assembler - Symbol Table

```
07DAH 2330 ADHL          0742H 1130 ALL
073EH 1090 CLEAR         0706H 0710 CPROC
0777H 1620 CPYI          07A6H 1920 CPYP
07F9H 2570 END           077BH 1650 ENDI
07A7H 1940 ENDP          07AAH 2000 EX
07CEH 2260 EX1           07D6H 2310 EX3
06EEH 0570 HLOOP         0727H 0940 HPROC
0100H 0340 IOPACK        0756H 1430 IPL
0767H 1540 IPLL1         07F3H 2530 LOAD
06DFH 0500 LOOP          0713H 0800 LPROC
012AH 0350 MASK          0700H 0690 MPROC
1000H 0310 MZ            074BH 1180 NEWIY
0739H 1050 NEWPRL        072CH 0970 NPROC
2000H 0330 PALT          077EH 1690 PRL
0796H 1840 PRLL1         0703H 0700 PROC
07E5H 2400 RES45         07ECH 2470 SAVE
0724H 0910 SPROC         06D0H 0010 START
071EH 0870 XPROC         07C8H 2210 ZIF
```

ANHANG *** Q ***

ü

Quellennachweis :

Q1 ELCOMP
Z 80 Assembler- Handbuch
Hofacker-Verlag 1. Aufl 1980

Q2 Zaks Rodnay
Programmierung des Z 80
SYBEX - Verlag 1. Aufl 1982

Q3 INTEL
Peripheral Design Handbook
August 1980

Q4 NATIONAL SEMICONDUCTOR
Linear Databook
1982

Q5 Monolithic Memories
PAL
Programmable Array Logic
Handbook
1981

Q6 NATIONAL SEMICONDUCTOR
PAL Data pack
März 1982

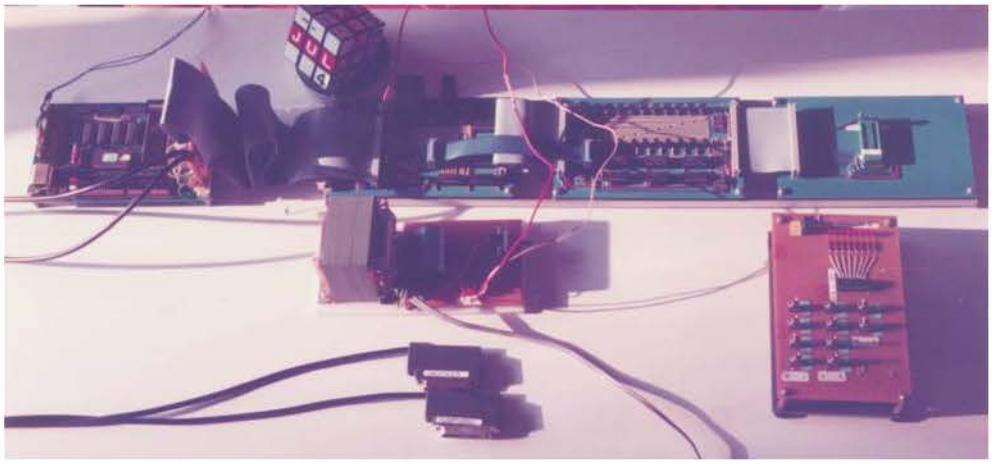
Q7 Texas Instruments
The TTL Databook
1982

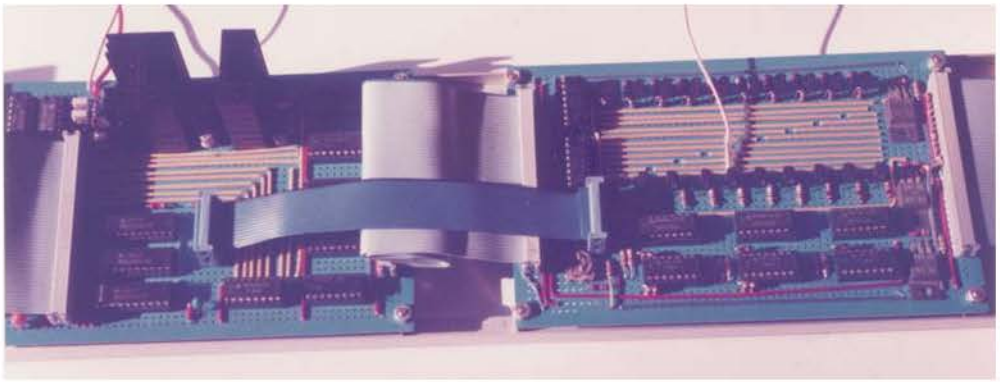
Q8 Texas Instruments
The Linear Databook
1977

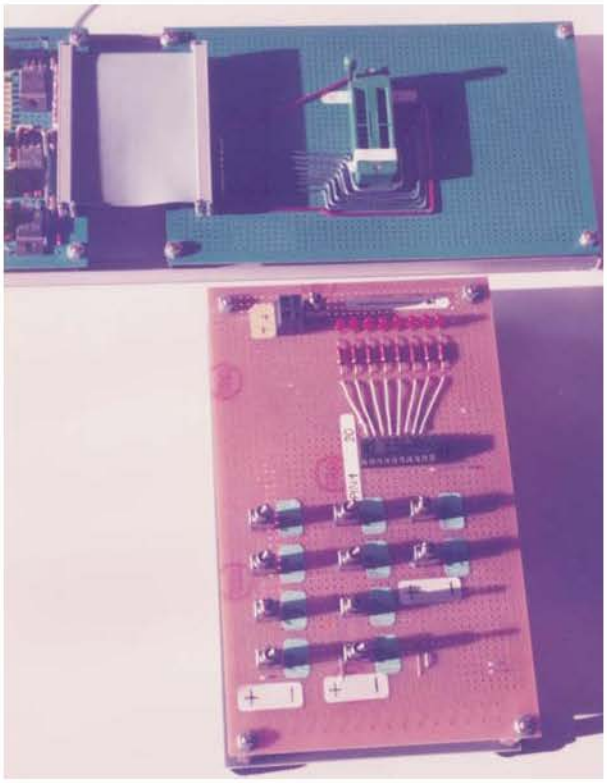
Q9 SGS ATES
Applikationsbericht L 200
Januar 1980

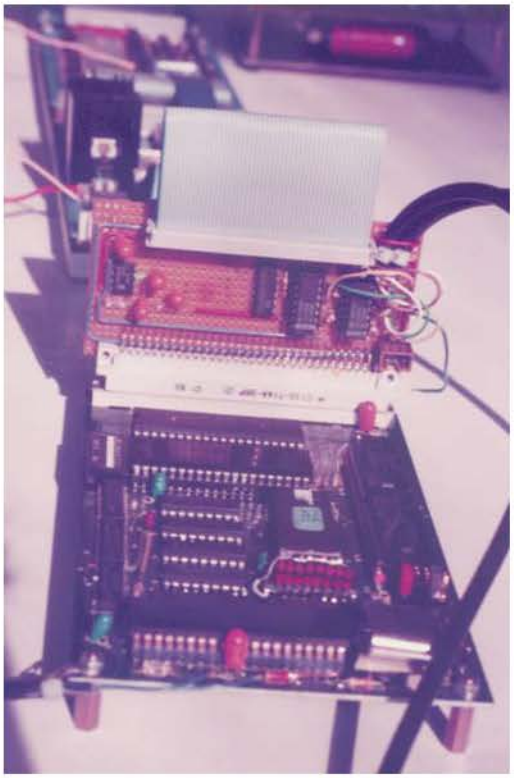
Q10 VALVO
Halbleiterbauelemente
1981/82

Q11 Klein, Rolf Dieter
Mikrocomputer Hard- und Softwarepraxis
Franzis Verlag 1981
Seite 108, Line 0037-005B









Name: Boretius, Michael

Ort: 6100 Darmstadt

Straße: Mühlstr. 46

, den 07.07.83

An die

Fachhochschule Darmstadt
Fachbereich Elektrotechnik

6100 D a r m s t a d t

Schöfferstr. 3

R e c h n u n g

Folgendes Gerät wurde von mir im Rahmen meiner Ingenieurarbeit angefertigt und an Sie geliefert:

Bezeichnung	Preis
autonomes bildschirmorientiertes PAL-Programmiergerät	1000,87 DM

(Unterschrift)

Das Gerät wird vom FB Elektrotechnik für Unterrichtszwecke verwendet und dafür angekauft.

Prüfungsvermerk des Dozenten:

i.O.

(Unterschrift)

Inventar Nr.: .....

sachlich richtig:

Konto-Nr.: [redacted]

bei: Darmstadter Volksbank

BLZ: .....