

GRAF computer

KEY3_{r3}

KEY3

- Tastaturschnittstelle für den NDR-Computer und das 19"-Ausbildungssystem.
- Anschlußmöglichkeit für PC/XT-Tastatur, für ASCII-Tastatur und Atari-Maus



The following information is provided for your information.
 The information is for your information only.
 The information is for your information only.
 The information is for your information only.

Inhaltsverzeichnis	Seite
1. Einführung	4
1.1. Zum NDR Computer	4
1.2. Wozu dient die Baugruppe	5
2. Technische Daten	5
3. Funktionsprinipien von Tastaturen	6
3.1. ASCII-Code	6
3.2. ASCII-Tastatur	6
3.3. PC/XT- Tastatur	7
4. Prinzipbeschreibung	9
4.1. Tastaturteil	9
4.1.1. Grundsätzlicher Aufbau	9
4.1.2. Blockschaltbild	10
4.1.3. Beschreibung des Blockschaltbildes	11
4.1.4. Flußdiagramm	12
4.1.4.1. Hauptprogramm	
4.1.4.2. INTProgramm	
4.1.4.3. NMIProgramm	
4.1.5. Beschreibung des Flußdiagramms	13
4.2. Mausteil	15
5. Aufbauanleitung	16
5.1. CMOS-Warnung	16
5.2. Stückliste	16
5.3. Löten und Lötzinn	17
5.4. Aufbau Schritt für Schritt	17
6. Testanleitung	23
6.1. Erste Prüfung ohne IC's	23
6.2. Einsetzen der kompletten Baugruppe	24
6.3. Tastaturanschluß	25
6.4. Testen mit dem Z80-Grundprogramm	26
6.5. Test mit dem FLOMON	26
6.6. Amerikanisch-deutscher Zeichensatz beim FLOMON	26
6.7. Testen mit dem 680XX Grundprogramm	26
6.8. Testen mit der CPU 8088	26
7. Fehlersuchanleitung	28
7.1. Sichtprüfung	28
7.2. Messung	28

	Seite
8. Schaltungsbeschreibung	30
8.1. Tastaturteil	30
8.1.1. Mikroprozessor - Herz der Baugruppe	30
8.1.2. Adressdekodierungen	31
8.1.3. Ein- und Ausgänge	32
8.1.4. Interrupts	33
8.2. Mausteil	34
9. Bedienungsanleitung	35
9.1. Grundfunktionen	35
9.2. Ebenen	35
9.3. Funktionstasten	36
9.4. Befehls- und Wortwiederholung	38
9.5. Reset	38
9.6. ASCII-Tastatur	38
9.7. At-Tastatur	39
10. Kommunikation	39
11. Bauelemente	40
11.1 TTL-Bausteine	40
11.1.1 7404	40
11.1.1 74LS05	41
11.1.2 74LS32	42
11.1.3 74LS125	42
11.1.4 74LS138	43
11.1.5 74LS164	44
11.1.6 74LS245	45
11.1.7 74LS234	46
11.1.8 74LS590	47
11.2 LM556	48
11.3 CMOS STATIC RAM 6264	49
11.4 EPROM 2764	50
11.5 Microprozessor Z80	51
Anhang A: Schaltplan (Tastaturteil)	60
Anhang B: Schaltplan (Mausteil)	61
Anhang C: Bestückungsplan	62
Anhang D: Layout Bestückungsseite	63
Anhang E: Layout Lötseite	64
Anhang F: Listing	65

1. Einführung

1.1 Zum NDR-Computer

Der NDR-Computer wird in der Fernsehserie "Rechner modular" aufgebaut, erklärt und in Betrieb genommen. Diese Serie wird vom Norddeutschen Rundfunk, vom Sender Freies Berlin, vom Bayerischen Rundfunk und von Radio Bremen ausgestrahlt. Der NDR-Computer wurde für diese Sendereihe entwickelt. Von Anfang an wurde bei diesem Computer das modulare Konzept in den Vordergrund gestellt. Mittlerweile laufen auf dem NDR-Computer nicht weniger als sechs CPUs und fünf Betriebssysteme. Kein anderes Computersystem kann eine solche Universalität aufweisen. Nähere Angaben zum modularen Konzept und zu den verfügbaren Baugruppen zum NDR-Computer finden Sie in unserem über 200-seitigen Farbkatalog.

Zur Serie gibt es einige Begleitmaterialien, daher ist es nicht unbedingt notwendig, die Fernsehserie gesehen zu haben, um den NDR-Computer zu bauen und zu begreifen:

- **Buch:** Rolf-Dieter Klein: *"Rechner modular"*
ISBN 3-7723-8721-7, Franzis-Verlag, München
Bestellnummer: 10991
- **Buch:** Rolf-Dieter Klein: *"Die Prozessoren 68000 und 68008"*
ISBN 3-7723-7651-7, Franzis-Verlag, München
Bestellnummer: 10588
- **Buch:** Helmut Ostermann:
"Maschinen- und Assemblersprache des MC 68000"
ISBN 3-8023-0174-9, Vogel-Verlag, Würzburg
Bestellnummer: 11003
- **Zeitschrift:** "LOOP" der Firma Graf Elektronik Systeme GmbH
Die Zeitschrift LOOP ist eine Kundenzeitschrift und enthält
Neuerungen, Änderungen, Tips und Tricks, Software usw.
zum NDR-Computer und auch speziell zu dieser Baugruppe.
- **Zeitschriften:** "mc" und "ELO" des Francis-Verlags
- **Christiani:** Für zahlreiche Anwendungen mit dem NDR-Computer werden
von dem renommierten Fernlehrinstitut Christiani Kurse und
Fernlehrgänge angeboten: z.B. SPS, Maschinensprache Z80,
Assembler Z80.
- **Katalog:** Der Katalog enthält alle Baugruppen, Systemzusammenstel-
lungen, Software, Teachware, usw. zum NDR-Computer. Er
enthält auch eine Einführung in die Rechnertechnik.
Farbkatalog 208 Seiten, Bestellnr.: 10834

1.2 Wozu dient die Baugruppe

Die Baugruppe KEY 3 ist in erster Linie das Bindeglied zwischen dem Mikrocomputer (SBC2, SBC3, CPU Z80 oder CPU680xx) und einer Tastatur. Mit ihr ist es möglich, sowohl eine herkömmliche ASCII-Tastatur (parallele Tastatur), als auch eine modernere PC oder MF-Tastatur (serielle Tastatur) anzuschließen.

Dabei kommen die Vorteile der KEY3 vor allem mit einer seriellen Tastatur zur Geltung, da sie die Möglichkeiten einer solchen Tastatur voll unterstützt.

Aber auch mit einer parallelen Tastatur bietet sie gegenüber den älteren KEY Baugruppen Verbesserungen, wie z. B. den Tastaturpuffer.

Des Weiteren dient die KEY3 zur Aufnahme einer SMART Watch. Dies wurde vorgesehen, da die SMART Watch auf einem 8K RAM sitzt, diese RAM-Bausteine aber auf der neueren ROA256/1M von 32k od. 256k Bausteinen abgelöst wurden. So bestand keine Möglichkeit mehr, die SMART Watch zu betreiben, obwohl sie aber gerade im neuen Multitasking Betriebssystem unbedingt benötigt wird.

Auch bietet das akkugepufferte RAM auf der KEY3 die Möglichkeit wichtige Daten, wie z. B. Setupdaten (Welche Konfiguration, was für Laufwerke, Festplatte ...) auch beim Abschalten des Computers zu erhalten.

2. Technische Daten

- Tastatur: 8-Bit seriell (Pc/XT/MF-Tastatur)
7-Bit parallel (ASCII-Tastatur)
- 256 Zeichen Ringpuffer
- 40 programmierbare Funktionstasten
- Belegung der Funktionstasten auch über das Betriebssystem oder das Grundprogramm möglich
- Wiederholungsmöglichkeit für die letzten 7 Befehle
- Tastentauschen durch Software möglich
- Bedienung der SMART Watch wird komplett übernommen
- Verwendung als Timer möglich (auch über Interrupt)

3. Funktionsprinzipien von Tastaturen

3.1 ASCII-Code

Da ein Computer keine Buchstaben oder Zeichen, sondern nur Zahlen versteht, ist es notwendig, das ganze Alphabet, alle Sonderzeichen und die Ziffern zu codieren, also für jedes Zeichen eine bestimmte Zahl festzulegen, die im Computer dann anstatt des Zeichens verwendet wird.

Bei den Mikrocomputern hat sich für diese Codierung der ASCII-Code durchgesetzt. ASCII bedeutet American Standard Code for Information Interchange (= Amerikanischer Standardcode für Informationsaustausch)

Dabei wurde von 7 Bit Daten und dem 8ten Bit als Überprüfungsbit (Paritybit), ob die anderen 7 Bit richtig übertragen wurden, ausgegangen. Es stehen so also 128 Codes zur Verfügung. Davon wurden die ersten 32 Codes für Steuerzeichen, wie z.B. CR (Wagenrücklauf) oder LF (line feed) usw. verwendet.

In neuerer Zeit wird das Überprüfungsbit (Paritybit) kaum noch verwendet, da die Übertragung so sicher ist, daß sich eine Überprüfung erübrigt. Von einigen Computerherstellern wird das so freiwerdende 8. Bit mit zur Codierung dazugenommen, so daß 256 Zeichen codiert werden können. Dabei werden die zusätzlichen Codes praktisch ausschließlich für Graphiksonderzeichen verwendet, die nicht unbedingt benötigt werden.

hex	dez.	ASCII	hex	dez.	ASCII	hex	dez.	ASCII	hex	dez.	ASCII
00	0	NUL	20	32	Space	40	64	@	60	96	^
01	1	SOH	21	33	!	41	65	A	61	97	a
02	2	STX	22	34	"	42	66	B	62	98	b
03	3	ETX	23	35	#	43	67	C	63	99	c
04	4	EOT	24	36	\$	44	68	D	64	100	d
05	5	ENQ	25	37	%	45	69	E	65	101	e
06	6	ACK	26	38	&	46	70	F	66	102	f
07	7	BEL	27	39	'	47	71	G	67	103	g
08	8	BS	28	40	(48	72	H	68	104	h
09	9	HT	29	41)	49	73	I	69	105	i
0A	10	LF	2A	42	*	4A	74	J	6A	106	j
0B	11	VT	2B	43	+	4B	75	K	6B	107	k
0C	12	FF	2C	44	,	4C	76	L	6C	108	l
0D	13	CR	2D	45	-	4D	77	M	6D	109	m
0E	14	SO	2E	46	.	4E	78	N	6E	110	n
0F	15	SI	2F	47	/	4F	79	O	6F	111	o
10	16	DLE	30	48	0	50	80	P	70	112	p
11	17	DC1	31	49	1	51	81	Q	71	113	q
12	18	DC2	32	50	2	52	82	R	72	114	r
13	19	DC3	33	51	3	53	83	S	73	115	s
14	20	DC4	34	52	4	54	84	T	74	116	t
15	21	NAK	35	53	5	55	85	U	75	117	u
16	22	SYN	36	54	6	56	86	V	76	118	v
17	23	ETB	37	55	7	57	87	W	77	119	w
18	24	CAN	38	56	8	58	88	X	78	120	x
19	25	EM	39	57	9	59	89	Y	79	121	y
1A	26	SUB	3A	58		5A	90	Z	7A	122	z
1B	27	ESC	3B	59	.	5B	91	[7B	123	{
1C	28	FS	3C	60	<	5C	92	\	7C	124	
1D	29	GS	3D	61	=	5D	93]	7D	125	}
1E	30	RS	3E	62	>	5E	94	^	7E	126	~
1F	31	US	3F	63	?	5F	95	_	7F	127	DEL

Bild: ASCII Tabelle

3.2 ASCII-Tastatur

Die ASCII-Tastatur war bis vor einiger Zeit die am häufigsten benutzte Mikrocomputertastatur. Sie war daher auch die preisgünstigste und wurde deshalb auch im NDR-Computer (mit den Baugruppen KEY und KEY2) verwendet.

Bei einer ASCII-Tastatur werden die Tasten schon innerhalb der Tastatur dekodiert und jeder Taste der entsprechende ASCII-Code zugeordnet. Dieser Code wird dann über die Datenleitung zum Computer gesendet. Bei den meisten ASCII-Tastaturen wird diese Übertragung parallel vorgenommen, d.h. für jedes der 7 Bit gibt es eine eigene Leitung. Aber es ist noch eine Leitung notwendig, um dem Computer zu melden, wenn neue Daten da sind. Die Tastatur legt dann auf diese Leitung einen kurzen Impuls (Strobe) an dem der Computer erkennt, daß jetzt eine Taste gedrückt wurde.

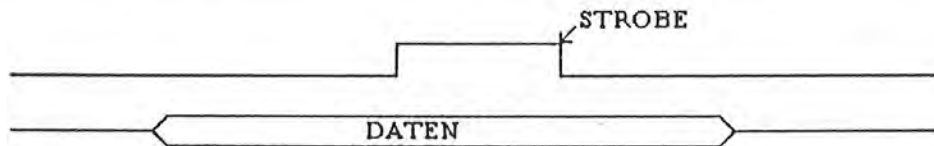


Bild: Strobe Timing

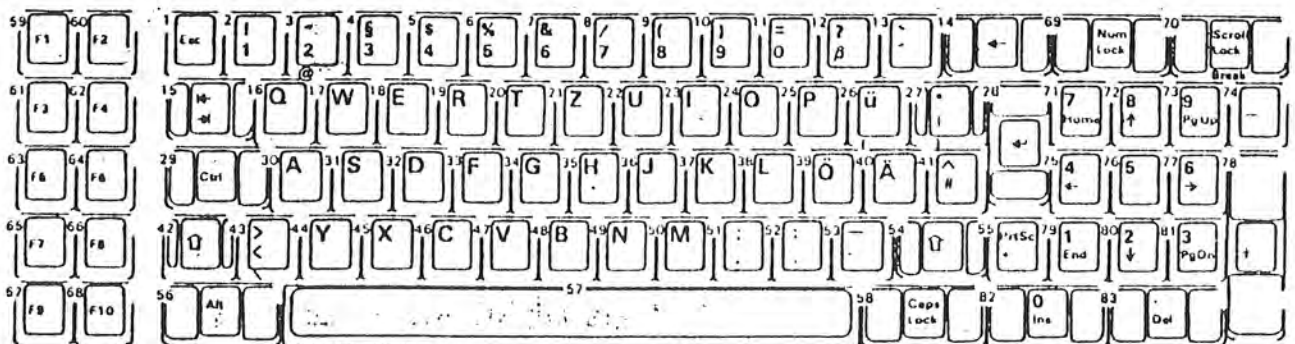
Bei älteren Tastaturen wurde diese Übertragung von der Tastatur zum Rechner auch seriell vorgenommen, d.h. auf einer Leitung wurden die 7 Bit hintereinander übertragen.

3.3 PC / XT Tastatur

In letzter Zeit hat sich immer mehr die Tastaturart der Personal Computer etabliert. So kam es zu einem Preisverfall dieser Tastaturen, wodurch sie inzwischen günstiger als ASCII-Tastaturen sind.

Bei diesen Tastaturen wird kein Code für jedes Zeichen (z.B. ASCII-Code), sondern ein Positionscode für jede Taste gesendet.

Positionscode heißt, daß eine Taste nicht nach ihrer Bedeutung, sondern nach ihrer Position codiert wird. Dabei wird einfach die Taste links oben auf der Tastatur (ESC) mit 1 bezeichnet und dann vorlaufend durchnummeriert.



Dieser Positionscode wird als 8 Bit Datenbyte gesendet und zwar, wenn eine Taste gedrückt wurde (Bit 7 ist dann Null) und nochmal, wenn die Taste wieder losgelassen wurde (Bit 7 ist dann Eins).

Wenn z.B. die Taste "4" gedrückt wird, so sendet die Tastatur den Code 5d (d=deziamal). Beim Loslassen der Taste wird der Code 133d (oder 85h h=hexadezimal) gesendet.

Auch Sondertasten wie "SHIFT" oder "CONTROL" werden genauso behandelt. Dies hat zur Folge, daß z.B. die Unterscheidungen, zwischen Klein- und Großbuchstaben (SHIFT gedrückt) oder zwischen Normal- und Controlbene (Control gedrückt), vom Computer selber übernommen werden müssen und nicht durch die Tastatur erledigt werden, wie bei der ASCII-Tastatur. Dies hört sich zwar im ersten Moment wie ein Rückschritt an, aber dadurch, daß sich viel mehr Möglichkeiten für die Tastatur (z.B. viele Ebenen, verschiedene Belegungen, Funktionstasten ...) ergeben und diese Möglichkeiten auch vom Rechner selber gesteuert werden können, fällt der Nachteil des Mehraufwands für den Computer kaum mehr ins Gewicht.

Auch wird bei einer PC/XT-Tastatur die Übertragung nicht parallel, sondern seriell vorgenommen. Dazu werden zwei Leitungen verwendet. Auf einer Leitung werden die Daten übertragen, auf der anderen Leitung ein Takt, der zur Synchronisation der Daten dient. Bei jeder steigenden Taktflanke wird der momentane Zustand der Datenleitung als ein Bit des Codes übernommen. Dabei werden nicht nur die 8 Datenbit gesendet, sondern immer vor einem Code noch ein Startbit, das immer High ist. So ergibt sich ein 9 Bitwort, dessen erstes Bit allerdings immer eins ist.

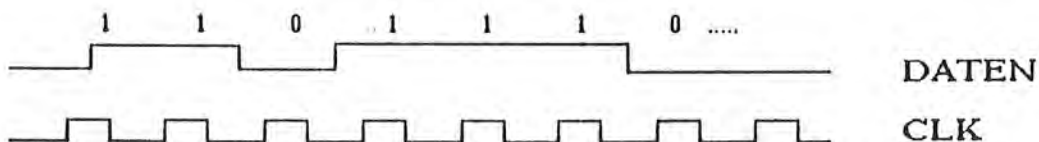


Bild: Timing einer Sendung

Da der NDR-Computer aber nicht für die Verwendung einer solchen Tastatur konstruiert wurde, kann diese aufwendige Tastaturkontrolle nicht vom Hauptprozessor übernommen werden. Um aber trotzdem die Vorteile einer PC/XT-Tastatur (vor allem den Preisvorteil) nutzen zu können, wurde auf die neue Tastaturbaugruppe, KEY3, eine eigener Prozessor installiert, der die komplette Tastaturverwaltung übernimmt. Dadurch fällt sogar noch der Mehraufwand für den Hauptprozessor, den eine PC/XT-Tastatur mit sich bringt, weg. So entstehen durch Verwendung dieser Tastatur keine Geschwindigkeitseinbußen, bei kompletter Nutzung der Vorteile solch einer Tastatur.

4. Prinzipbeschreibung

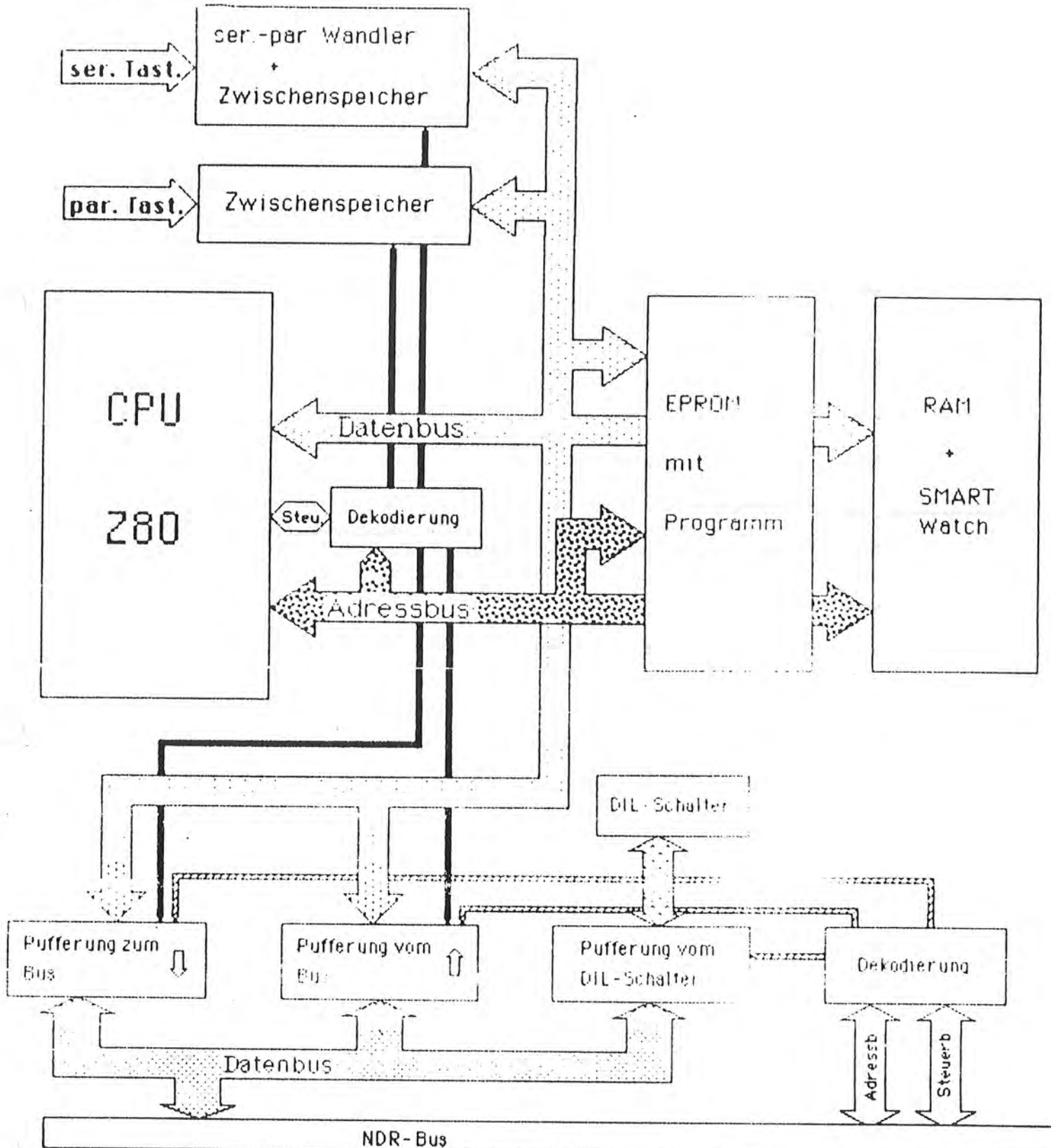
4.1 Tastaturteil

4.1.1 Grundsätzlicher Aufbau

Die KEY3 ist eigentlich ein kleiner Einplatinencomputer. Sie besitzt einen eigenen Prozessor, hat einen eigenen Speicherbereich, hat ihre eigenen Ein- und Ausgabeports und benötigt natürlich auch ihr eigenes Programm. Der Prozessor steuert mit Hilfe des Programms alle Vorgänge auf der Baugruppe.

Und wie beim Hauptcomputer auch, sind, um Daten auszutauschen und Adressen anzugeben, hier natürlich auch ein Daten- und Adressbus vorhanden (Baugruppeninterne Busse). Daneben gibt es für die Ports noch Pufferbausteine, die über Steuerleitungen angesprochen werden. Zum besseren Verständnis der Baugruppe tragen Kenntnisse über eine andere Prozessorplatine (SBC2, SBC3, CPU680XX) bei.

4.1.2 Blockschaltbild



4.1.3 Beschreibung des Blockschaltbildes

Links oben im Blockschaltbild sind die Schnittstellen zu den Tastaturen als Pfeile dargestellt. Der obere Eingang dient zum Anschluß einer seriellen PC/XT-Tastatur. Die ankommenden seriellen Daten werden hier mit Hilfe eines Schieberegisters in parallele Daten umgewandelt und zwischengespeichert. Beim Eingang für die parallele ASCII-Tastatur entfällt natürlich die seriell-parallel Wandlung, aber die Daten werden genauso zwischengespeichert.

Die in den Zwischenspeichern liegenden Daten können nun vom Prozessor (CPU Z80) abgeholt und weiterverarbeitet werden. Um die Daten zu holen, wird vom Prozessor über die Dekodierung der entsprechende Zwischenspeicher angesprochen. Der angesprochene Zwischenspeicher legt nun seine Daten auf den Datenbus und sie werden dort vom Z80 abgeholt.

Damit der Prozessor überhaupt weiß, daß er Daten holen soll, benötigt er ein Programm. Dieses Programm steht im Eprom.

Das Programm ist auch zuständig für die Weiterverarbeitung der eingelesenen Daten. Die Daten vom seriellen Eingang werden zuerst vom Positionscodex in den ASCII-Code umgewandelt; dabei werden Sondertasten erkannt und gesondert behandelt (z.B. SHIFT, ALT oder auch die Funktionstasten usw.). Die nun erhaltenen Daten werden im RAM zwischengespeichert. Bei den Daten vom seriellen Port ist eine Umwandlung oder gesonderte Interpretation nicht notwendig. Sie werden unverändert zwischengespeichert.

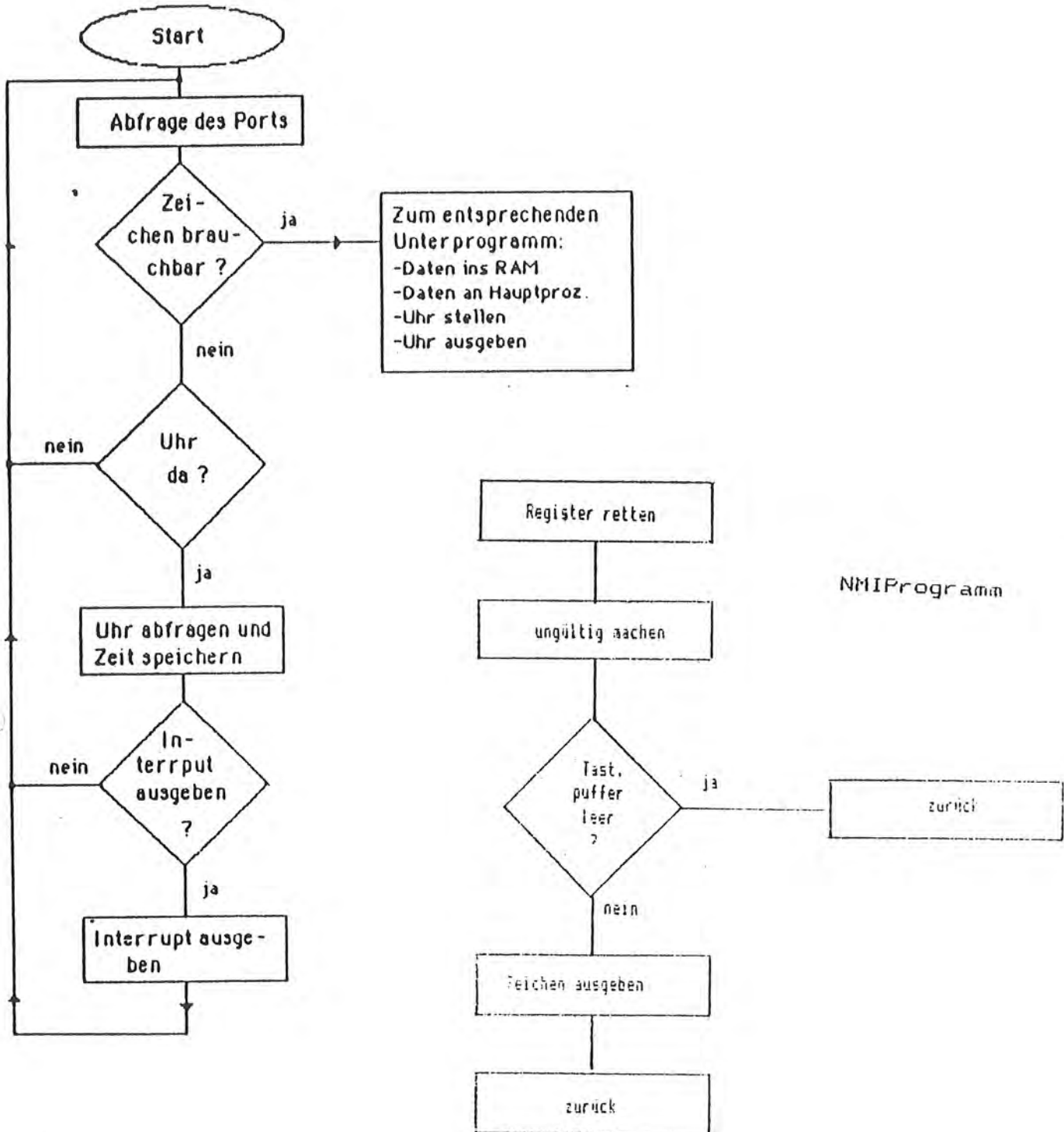
Die im RAM zwischengespeicherten Daten werden vom Z80 nacheinander immer dann an die Pufferung zum Bus weitergegeben, wenn der Hauptprozessor zugreift. Mit diesem Zugriff bestätigt der Hauptprozessor, daß er die letzten Daten übernommen hat und bereit ist für neue Daten. Diese Art der "Mitteilung" ist auch bei den alten Tastaturbaugruppen KEY1 und KEY2 so vorgesehen und wurde, um kompatibel zu sein, übernommen.

Die Pufferung vom Bus ist notwendig, daß der Hauptprozessor mit dem KEYProzessor kommunizieren kann.

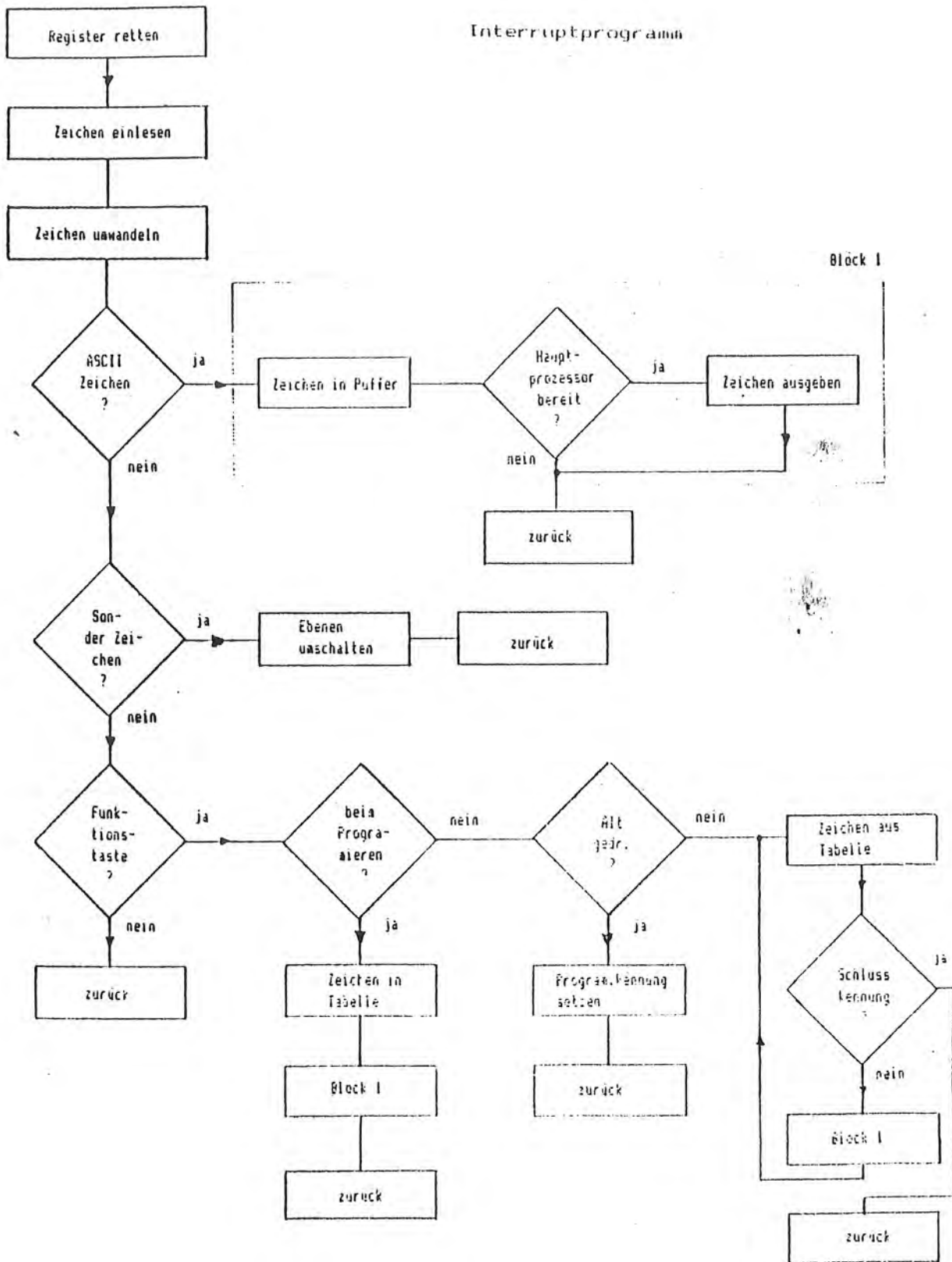
Die zweite Dekodierung ist für den Hauptprozessor, damit er über den BUS auf die Pufferungsbausteine zugreifen kann und so Daten holen oder abliefern kann.

4.1.4 Flußdiagramme des Programms

Hauptprogramm



Interruptprogramm



4.1.5 Beschreibung des Flußdiagramms

Das Flußdiagramm gliedert sich in drei voneinander unabhängige Teile. Die drei Programmteile Hauptprogramm, Interruptprogramm und NMI-Programm sind deshalb unabhängig voneinander, da sie sich nicht gegenseitig aufrufen. Normalerweise befindet sich der Prozessor im Hauptprogramm, in dem sich kein Sprung in ein Unterprogramm befindet. Die Unterprogramme können nur durch einen Interrupt aufgerufen werden. Das Interruptprogramm durch den normalen, sperrbaren Interrupt (INT) und das NMI-Programm durch den nicht sperrbaren Interrupt (NMI = Non Maskable Interrupt).

Das Flußdiagramm des Hauptprogramms ist einfach zu durchschauen. Es werden hier eigentlich nur der Eingangsport (im Blockdiagramm: Pufferung vom Bus) abgefragt. Wenn hier ein bestimmtes Zeichen anliegt, wird zum entsprechenden Unterprogramm verzweigt. So z.B. zum Unterprogramm "Daten ins RAM einlesen".

Falls eine SMART Watch vorhanden ist, wird auch diese im Hauptprogramm laufend abgefragt, um immer auf dem aktuellen Stand zu sein und immer, nach einer einstellbaren Zeit, einen Interrupt auf den Bus zu legen (wenn freigegeben).

Etwas komplizierter ist da schon das Flußdiagramm des Interruptprogramms. Hier wird, nachdem die Register gesichert wurden, das Zeichen vom Zwischenpuffer eingelesen und über die Tabelle in ein ASCII-Zeichen umgewandelt. Danach wird abgefragt, ob es ein normales ASCII-Zeichen ist. Wenn ja, wird es gleich in den Puffer geschrieben und falls der Hauptprozessor gerade bereit ist, auch gleich ausgegeben. Ist es kein normales ASCII-Zeichen, wird abgefragt, was für ein Sonderzeichen es ist.

Bei den Sonderzeichen, die in eine andere Ebene umschalten (z.B. SHIFT od. CONTROL), wird der Zeiger auf die Umwandlungstabelle verändert, so daß in der Tabelle eine andere Ebene angesprochen wird. Auch werden die Sonderzeichen beim Loslassen beachtet und der Zeiger wieder auf den normalen Wert gebracht.

Die dritte Möglichkeit ist, daß eine Funktionstaste gedrückt wurde. Hierbei wird noch unterschieden, ob sie nur ausgegeben werden soll, oder ob sie programmiert wird.

Beim Ausgeben wird einfach aus einer anderen Tabelle ein Zeichen nach dem anderen geholt und genauso weiterbehandelt, als ob dieses Zeichen gedrückt worden wäre.

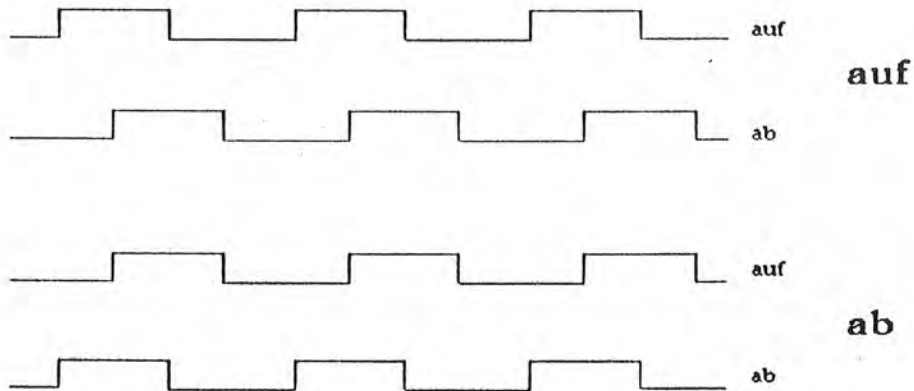
Beim Programmieren werden alle Zeichen, die von der Tastatur kommen, solange die Funktionstaste gedrückt bleibt, in die Tabelle geschrieben und stehen so auch später zur Verfügung.

Der letzte Teil ist das NMI-Programm. Es wird aufgerufen, wenn der Hauptprozessor das Gültigkeitsbit, das Bit 7, ungültig machen will. Dies ist notwendig, daß er dann ein neues Zeichen als neu erkennt. (Sonst würde er nicht erkennen, wenn mehrmals hintereinander dieselbe Taste gedrückt wurde.)

Das Bit 7 wird durch einen Zugriff des Prozessors auf den Dil-Schalterport, sprich Port 69H ungültig gemacht. Dann wird überprüft, ob sich im Tastaturpuffer noch nicht ausgegebene Zeichen befinden. Ist dies der Fall, wird gleich das nächste Zeichen ausgegeben.

4.2 Mausteil

Eine einfache Maus oder ein einfacher Trackball besitzt 4 TTL-Ausgänge, entsprechend den vier möglichen Bewegungsrichtungen (rechts, links, auf und ab). Bei einer Aufwärtsbewegung der Maus erscheinen Rechtecksignale an den beiden Ausgängen für auf und ab. Die Anzahl der ausgesandten Impulse wächst proportional mit dem zurückgelegten Weg. Bei einer Abwärtsbewegung der Maus erscheinen an den beiden genannten Eingängen ebenfalls Rechteckimpulse. Wie läßt sich nun die Bewegungsrichtung ermitteln? Die beiden Signale einer Bewegungsrichtung (auf und ab bzw. links und rechts) weisen eine Phasenverschiebung zueinander auf. Anhand dieser Phasenverschiebung kann zum Beispiel die Unterscheidung einer Auf- oder Abwärtsbewegung erfolgen. Bewegt sich nun die Maus nicht rein waage- oder senkrecht, so kann aus der Zahl der empfangenen Impulse in X- und Y-Richtung die Bewegungsrichtung ermittelt werden und somit eine Positionsbestimmung erfolgen.



5. Aufbauanleitung

5.1. CMOS-Warnung

CMOS-Bausteine sind hochempfindlich gegen elektrostatische Aufladung! Verwenden Sie zur Aufbewahrung und zum Transport von CMOS-Bausteinen immer den leitenden Schaumstoff! Alle Pins müssen kurzgeschlossen sein. Bei der KEY3 wird der CMOS-Baustein 6264 (RAM-Speicher) verwendet.

Tip: Fassen Sie an ein geerdetes Teil (z.B. Heizung, Wasserleitung) bevor Sie einen Baustein berühren.

Bitte beachten Sie hierzu auch den Artikel "Schutzmaßnahmen für MOS-Schaltungen" in unserer Zeitschrift LOOP3.

5.2. Stückliste

Prüfen Sie zunächst den Bausatz auf Vollständigkeit .

Stückliste KEY3 Ausgabe 1

1 Original GES-Leiterplatte mit Lötstoplack r1

1 Handbuch KEY3 Ausgabe 1

7 Sockel 14-polig

9 Sockel 16-polig

6 Sockel 20-polig

2 Sockel 28-polig

1 Sockel 40-polig

2 556

J12, J16

1 7404

J13

1 74LS05

J3

1 74LS32

J2

1 74LS125

J25

5 74LS138

J8, J10, J11, J14, J15

1 74LS164

J9

3 74LS245

J4, J19, J20

3 74LS374

J5, J17, J18

4 74LS590

J21-J24

1 RAM 8K NDR statisch (...64)

J6

1 EPROM KEY3 V1.0

J7

1 CPU Z80A 4MHz

J1

14 Con. 100nF Keramik

C1, C2, C5-C7, C10, C16, C17, C19,
C21, C22, C25-C27

8 Con. 10nF

C4, C8, C9, C11, C14, C15, C18,
C20

1	Con. 10uF Tantal	C28
5	Con. 22nF Wickel	C3, C12, C13, C23, C24
4	10KOhm Widerstand	R1-R4
3	4*1KOhm Netzwerkwiderstand	RN1, RN3, RN4
1	8*1KOhm Netzwerkwiderstand	RN5
1	4*10KOhm Netzwerkwiderstand	RN2
1	Stiftleiste gew. 36-polig	ST5
1	Stiftleiste gew. 18-polig	ST5
1	Canon-Buchse 15-polig gew. lötlbar/Sc	ST4
1	DSUB 9-polig gew. Stecker m. Halter	ST3
1	2*3 polige Stiftleiste gerade	JMP1
1	2*8 polige Stiftleiste gerade	ST2
2	3 polige Stiftleiste gerade	JMP2, JMP3
3	2 poligen Shuntstecker	
1	Quarz 4.000 MHz	X1
1	Dilschalter 8	S1

5.3. Löten und Lötzinn

Die Baugruppe ist sehr eng bestückt. Zum Aufbau ist daher etwas **Sorgfalt und ein wenig Erfahrung** nötig.

Die Leistung des verwendeten LötKolbens sollte 50 W nicht überschreiten, da sonst **Bauteile und die Leiterplatte** zu heiß und damit zerstört werden könnten.

Beim Lötzinne ist darauf zu achten, daß es unbedingt ein **Flußmittel enthält und nicht dicker als 1 mm** ist (z. B. SN 60 0.75 mm Durchmesser). Das richtige Lötzinn kann viel **Ärger mit Nachlöten** und Fehlersuche ersparen.

5.4 Aufbau Schritt für Schritt

Zum Aufbau der Baugruppe benötigen Sie folgendes Werkzeug:

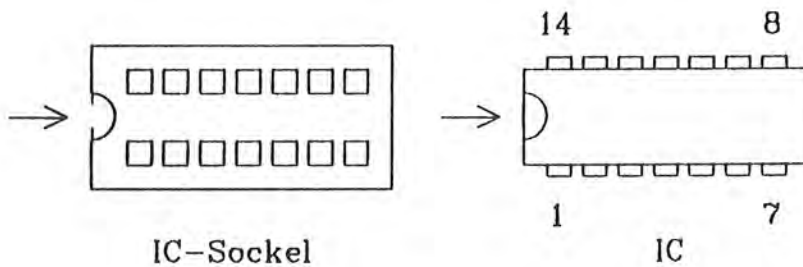
- LötKolben mit temperatureregelter Spitze
- Lötzinn, säurefrei, mit kolophonium-Seele
- Pinzette
- Elektroniker-Seitenschneider

Beim Aufbau der Baugruppe sollten Sie sehr sorgfältig arbeiten und sich **Zeit lassen**. Gehen Sie dabei genau nach der Aufbauanleitung vor. Am besten ist es, wenn Sie die **Anleitung vorher genau durchlesen**, um sich einen Überblick über die Reihenfolge der Bestückung zu verschaffen.

Zu Ihrer Orientierung sind bei jedem Abschnitt Kontrollpunkte angegeben, die Sie abhaken können, wenn Sie einen Abschnitt bearbeitet haben.

Die Bauteile werden auf der Bestückungsseite bestückt. Auf der anderen Seite der Leiterplatte, der Lötseite, wird ausschließlich gelötet. Sie erkennen diese Seite an der Aufschrift "Lötseite".

Beginnen Sie mit dem Einlöten der IC-Sockel. Bestücken Sie die Leiterplatte zuerst mit allen Sockeln von der Bauteilseite her. Schauen Sie schon beim Bestücken nach, ob die Lage der Sockel richtig ist (Kerbe in der Fassung muß in dieselbe Richtung weisen, wie die Nase auf dem Bestückungsplan (Siehe Skizze).



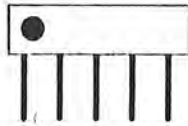
Markierung an IC und Sockel

Beachten Sie auch, daß der Platz für den DIL-Schalter (S1) freibleibt. Wenn Sie alle Sockel eingesetzt haben, decken Sie die Seite mit einem Karton ab und drehen die Leiterplatte um. Der Karton hilft Ihnen, die Sockel beim Umdrehen der Leiterplatte in der gewünschten Position zu halten. Verlöten Sie nun von jedem Sockel, am besten diagonal zueinander, zwei Beinchen. Bevor Sie nun die restlichen Beinchen verlöten, drehen Sie die Leiterplatte um und kontrollieren sie nach den folgenden Gesichtspunkten.

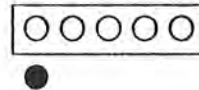
- () Schauen Sie noch einmal nach, ob die IC-Sockel sich auf ihrem richtigen Platz befinden.
- () Überprüfen Sie, ob die Kerben der Sockel in die gleiche Richtung weisen, wie auf dem Bestückungsplan.
- () Die Sockel sollten auf der Leiterplattenoberfläche aufliegen.
- () Wenn Sie Wert auf eine sehr gute Optik legen, dann können Sie die Sockel so ausrichten, daß sie parallel zu den Seitenkanten verlaufen.

Wenn Sie sich von der richtigen Lage der Sockel überzeugt haben, können Sie diese jetzt vollständig einlöten.

Nun folgt die Bestückung mit den Netzwerkwiderständen RN1..RN4. Ein Netzwerkwiderstand ist eine Ansammlung von Widerständen, die alle den gleichen Wert und einen gleichen Bezugspunkt haben. Dieser Punkt ist sowohl auf dem Netzwerkwiderstand, als auch auf dem Bestückungsplan mit einem kleinen Punkt gekennzeichnet (Siehe Skizze).



Netzwerkwiderstand



Leiterplatte

Markierung des gemeinsamen Anschlusses bei Netzwerkwiderständen
Achten Sie beim Einlöten darauf, daß der Punkt des Netzwerkwiderstandes an der gleichen Stelle liegt, wie auf dem Bestückungsplan, da sonst falsche Widerstandswerte entstehen. Dies führt dazu, daß die Baugruppe gar nicht oder nicht richtig funktioniert.

Der Wert der Netzwerkwiderstände ist aus der Beschriftung zu erkennen. Das letzte Zeichen der Aufschrift ist immer ein Buchstabe, der uns aber zur Bestimmung des Wertes nicht weiter interessiert. Wichtig sind dagegen die drei Ziffern, die vor diesem Buchstaben stehen. Die ersten zwei Ziffern geben den Grundwert an, die dritte Ziffer die dazugehörige Zehnerpotenz. So hat der RN1 z.B. den Wert 1KOhm. Die Aufschrift lautet entsprechend "102". Der Wert errechnet sich folgendermaßen:

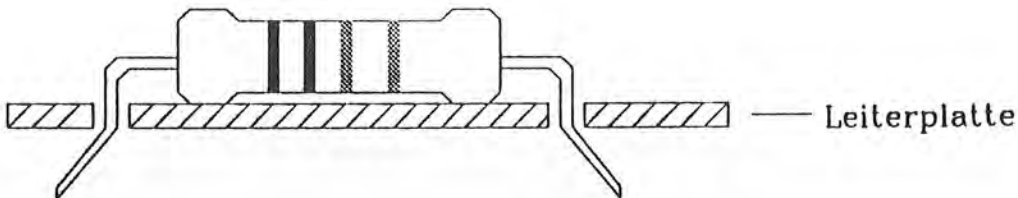
$$10 \text{ Ohm} * 10^2 = 10 \text{ Ohm} * 100 = 1000 \text{ Ohm} = 1\text{KOhm}.$$

Gehen Sie beim Einlöten ähnlich wie bei den Sockeln vor. Löten sie zuerst nur ein Beinchen des Netzwerkwiderstandes fest. Drehen Sie die Leiterplatte dann wieder um, und kontrollieren Sie sie nach folgenden Gesichtspunkten.

- () Überprüfen Sie, ob alle 5 Netzwerkwiderstände so eingelötet sind, daß sich der Punkt auf dem Bauteil an der selben Stelle befindet, wie auf dem Bestückungsplan.
- () Kontrollieren Sie bitte, ob die Werte der Netzwerkwiderstände mit den Werten auf dem Bestückungsplan übereinstimmen.
- () Richten Sie die Bauteile entsprechend aus (z.B. horizontale Lage).

Fahren Sie fort mit dem Einlöten der Widerstände R1..R4. Biegen Sie die Beine so, daß der Nun können Sie die Netzwerkwiderstände vollständig einlöten.

Widerstand liegend auf der Platinenoberfläche aufgebracht werden kann (Siehe Skizze).



Richtiges Bestücken eines Widerstandes

Sobald dies geschehen ist, winkeln sie die Beine leicht ab, um ein Verrutschen der Bauteile zu verhindern. Da die vier Widerstände den gleichen Wert besitzen, müssen sie vor dem Einlöten nur darauf achten, daß die Widerstände sich an der richtigen Position laut Bestückungsplan befinden.

Um eine Fehlbestückung bei den Kondensatoren C1..C28 zu verhindern, löten sie immer einen Kondensatortyp (z.B. alle Keramik-Condensatoren) vollständig ein. Gehen sie hierbei wie bei den Widerständen vor. Biegen sie zuerst die Beinchen so, daß der Bauteilkörper auf der Platineoberfläche aufliegt. Winkeln Sie zur ersten Fixierung die Beinchen dann ab. Wenn Sie sich davon überzeugt haben, daß jeder Kondensator sich auf seinem richtigen Platz befindet, können Sie mit dem Verlöten beginnen.

ACHTUNG: Der Tantal-Kondensator C28 (10 uF) ist ein Elko. Bei diesem Kondensator müssen Sie auf die Polung achten, d.h. das Beinchen, das mit "+" gekennzeichnet ist, muß mit dem "+" Zeichen auf dem Bestückungsplan übereinstimmen. Sollte er falsch eingelötet und die Baugruppe in Betrieb genommen werden, würden sie diesen Kondensator zerstören.

Vorsicht: Im schlimmsten Fall kann es passieren, daß der Elko explodiert. Bei den anderen Kondensatoren C1..C27 brauchen Sie nicht auf die Polung achten.

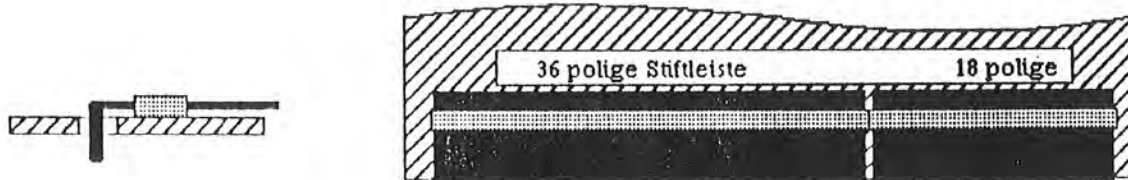
Für die nachstehenden Kontrollabschnitt wurden folgende Abkürzungen gewählt:

- Ker. = Keramik-Kondensator
- tan. = Tantal-Kondensator
- Wic. = Wickelkondensator
- Con. = 10 nF-Condensator

	Ker.	Tan.	Wic.	Con.
Stimmt die Position(en) des/der Kondensators/Kondensatoren?	()	()	()	()
Kontrollieren Sie nochmals die Polung des Tantalkondensators!		()		

Jetzt können sie damit beginnen die Steckleisten einzulöten. Beginnen Sie mit den Jumperleisten (2*3polig (JMP1) und 1*3Polig (JMP2/JMP3) und dem Anschluß für die ASCII-Tastatur (ST2). Verfahren Sie hierbei ebenso wie beim Einlöten der IC-Sockel.

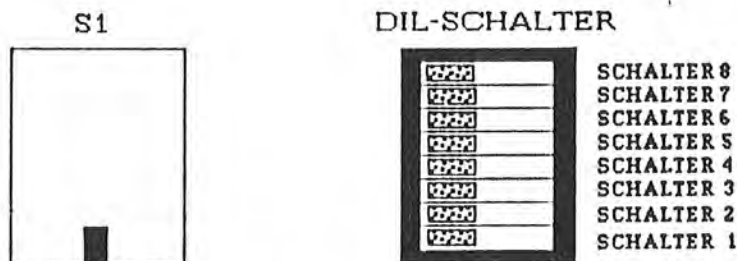
Wenn Sie dies erledigt haben, können Sie die NDR-Leiste aufbringen. Gehen Sie ähnlich vor wie bei den IC-Sockeln. Die NDR-Leiste setzt sich aus einer 18 poligen und einer 36 poligen gewinkelten Stiftleiste zu einer 54 poligen Leiste zusammen (Siehe Skizze).



AUFBAU NDR-LEISTE

Setzen Sie diese Leisten ein und löten sie am besten nur drei Beine je Leiste fest. Kontrollieren Sie, ob die Leiste nun parallel zur Kante verläuft. Wenn nicht, korrekieren Sie dies, indem sie die Leiste leicht in die entsprechende Richtung drücken und den günstigsten Pol festlöten. Wenn die Leiste die gewünschte Parallelität erreicht hat, können Sie die restlichen Beine verlöten.

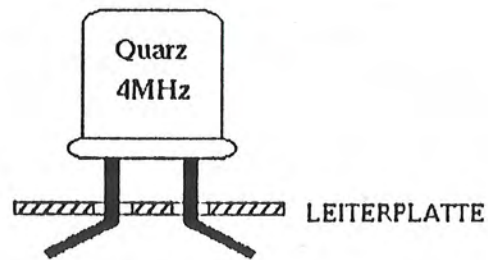
Jetzt können Sie die großen Bauteile einlöten. Bei DII-Schalter verfahren Sie genauso, wie bei den IC-Sockeln. Wie Sie sehen können ist der DIL-Schalter ein Bauelement, der mehrere Schalter beinhaltet. Diese Schalter sind nummeriert. Der Schalter 1 muß auf der Seite liegen, auf der die Kerbe im Bestückungsplan eingezeichnet ist (Siehe Skizze).



Der PC/XT-Anschluß (ST4) wird mit dem Aufbringen auf der Platine fest justiert und kann sofort verlötet werden.

Für den Mausanschluß (ST3) ist es ratsam, die Canon-Buchse mittels der Schrauben ebenfalls auszurichten und fest zu justieren, und ihn dann erst zu verlöten.

Als letztes Bauelement wird nun der Quarz aufgebracht. Dies geschieht zur Sicherheit des Quarzes, da dieser durch das häufige Umdrehen der Leiterplatte beschädigt werden könnte. Beim Quarz müssen Sie nicht auf die Polung achten. Er wird direkt stehend auf der Leiterplatte aufgebracht (Siehe Skizze).



EINLÖTEN DES QUARZES

6. Testanleitung

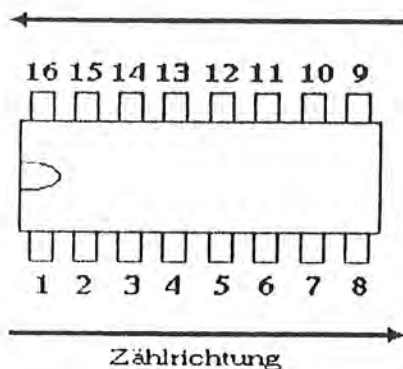
6.1 Erste Prüfung ohne IC's

Der erste Test der Baugruppe sollte immer ohne IC's ausgeführt werden, um eine Zerstörung derselben zu verhindern. Die Leiterplatte ist demnach nur mit den IC-Sockeln und den passiven Elementen bestückt. Sollten sie die IC-Bausteine schon eingesetzt haben, bitten wir sie, diese wieder zu entfernen. Überprüfen sie zunächst den ohmschen Widerstand zwischen Masse (NDR-Leiste/Pin 6, 7, 52, 53) und der Versorgungsspannung +5V (NDR-Leiste/Pin 4, 5). Hier darf der gemessene Wert keinesfalls niederohmig sein. Werte um die 1KOhm sind normal. Wenn sie einen abweichenden Wert feststellen, fahren sie mit dem Kapitel 7 (Fehlersuche) fort.

Sollten sie ein lauffähiges System haben, d.h. nach dem Einschalten erhalten sie eine Meldung auf dem Bildschirm, stecken sie die halbfertige KEY3 dazu (**vorher ausschalten !!!**). Wenn sie jetzt das System wieder einschalten, muß dieses noch weiterhin problemlos funktionieren (Test der Daten- und Adressleitungen auf Kurzschluß). Wenn nicht, weiter mit Kapitel 7.

Nun beginnen sie damit, die Spannungsversorgung +5V der einzelnen IC-Bausteine zu messen. Bei den Standard TTL-Bausteinen liegt diese jeweils auf dem letzten Pin der Fassung (z.B. bei 14 poligen an Pin 14).

Zu zählen beginnen sie auf der rechten Seite der Einkerbung, und zwar der Reihe nach auf dieser Seite hoch (Siehe Skizze). Wenn sie am letzten Pin auf der rechten Seite angekommen sind, haben sie bei den Standard TTL-Bausteinen den Massepin erreicht. Wenn sie nun weiterzählen müssen, springen sie direkt auf die andere Seite des Sockels und zählen dann auf dieser Seite wieder in Richtung Kerbe. Links der Kerbe sitzt nun der Letzte Pin der Fassung. Hier liegt die 5V-Versorgungsspannung an.



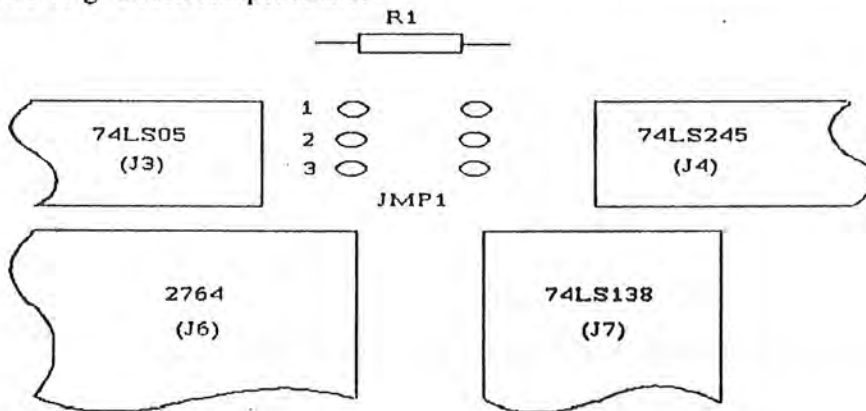
Dieselbe Anordnung gilt für den Speicherbaustein 6264 (J6) und die Timmer 556 (J12, J16). Beim Prozessorbaustein Z80A (J1) liegt die Versorgungsspannung an Pin 11 an, die Masseverbindung an Pin 29. Beim EPROM muß die Versorgungsspannung +5V am Pin 28 des Sockels liegen, die Masseverbindung an Pin 14 des Sockels.

Wenn sie alle Spannungen überprüft haben, können sie das System abschalten, die KEY3 herausnehmen und die IC-Bausteine einsetzen.

VORSICHT: Stecken sie die IC's richtig herum (d.h. IC-Kerbe über Sockel-Kerbe) auf die entsprechenden Sockel. Kontrollieren sie nachdem sie alle eingesteckt haben, die Lage der IC's lieber noch einmal nach. Schauen sie auch, ob alle IC-Beinchen in den Sockeln stecken. Herausstehende Beinchen bewirken, daß die Baugruppe nicht funktioniert und eine eventuelle Fehlersuche erschwert wird.

6.2. Einsetzen der kompletten Baugruppe

Bevor sie die Baugruppe wieder einsetzen, müssen sie den Jumper JMP1 richtig setzen. Mit diesem Jumper entscheiden sie, ob sie eine PC/XT oder eine ASCII-Tastatur verwenden. Das folgende Bild zeigt diesen Jumper JMP1:



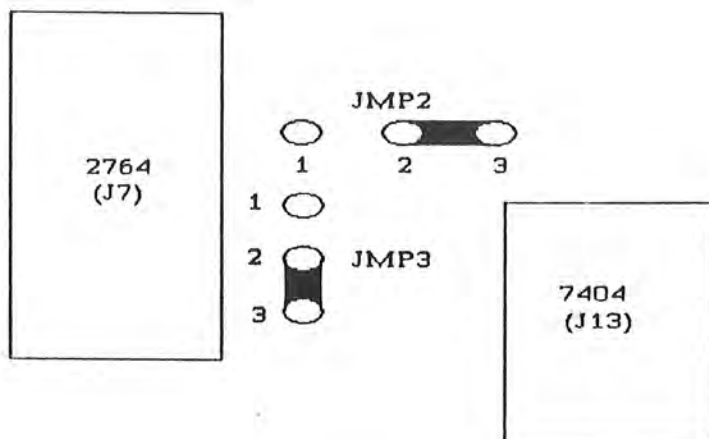
*Achtung! Jumperung entspricht nicht Version r3
siehe übernächste Seite*

In der Stellung 1 können sie eine ASCII-Tastatur betreiben. Hierbei ist die Voraussetzung, daß die Tastatur ein STROBE-Signal mit LOW-Pegel aussendet.

Sendet ihre Tastatur das Strobe-Signal mit einem HIGH-Pegel, müssen sie die Jumperstellung 2 wählen.

Für den Fall, daß sie eine PC/XT Tastatur verwenden, müssen sie die Jumperstellung 3 wählen.

Wie sie wissen, gibt es Speicher mit unterschiedlicher Speicherkapazität. Damit die KEY3 nicht nur mit dem standardmässigen RAM 6264 betrieben werden kann, sondern auch mit dem nächsthöheren RAM, dem 62256, wird der Jumper JMP3 benötigt. Die gleiche Funktion besitzt der Jumper JMP2 für das EPROM. Standardmässig verwenden wir den EPROM-Typ 2764. Mit Hilfe des JMP2 können auch die Typen 27128 und 27256 eingesetzt werden. Für den Fall, daß Sie einen höherwertigen Speicher oder ein höherwertiges EPROM einsetzen, müssen Sie die Voreinstellung auf der Lötseite der Platine unterbrechen. Auf der folgenden Seite sehen Sie die Standardeinstellung der Jumper und eine Tabelle, wie Sie die Jumper für den jeweiligen Typ setzen müssen.



JUMPERSTELLUNGEN JMP2:

SPEICHER	TYP	STELLUNG
8K * 8	6264	VERBINDUNG 2-3
32K * 8	62256	VERBINDUNG 1-2

JUMPERSTELLUNGEN JMP3:

SPEICHER	TYP	STELLUNG
8K * 8	2764	VERBINDUNG 1-2
16K * 8	27128	VERBINDUNG 1-2
32K * 8	27256	VERBINDUNG 2-3

Sobald die Jumper richtig eingestellt sind, können sie die Baugruppe einstecken. Wenn sie ihr System einschalten, muß die Anzeige nach wie vor problemlos erscheinen. Sollte dies nicht der Fall sein, schalten sie das System sofort ab und gehen sie zum Kapitel 7.

6.3. Tastaturanschluß

Bevor sie die Tastatur anschliessen, sollten sie zuerst das System wieder ausschalten. Die PC-Tastatur wird an der Steckerleiste ST 4 angeschlossen. Sobald der Stecker eingesteckt wurde, schalten sie das System ein. Wenn sie nun die Tasten CAPS-Lock (s.Kap.3.3) oder NUM-Lock drücken, sehen sie, sofern die Kontrolleuchten rechts oben auf der Tastatur aufleuchten, ob die Tastatur die erforderliche Spannungsversorgung hat.

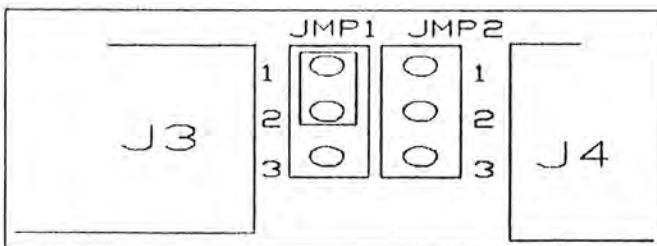
Wenn sie eine ASCII-Tastatur anschließen wollen, müssen sie die Verbindung zur Steckerleiste ST2 herstellen. Hierbei müssen sie darauf achten, daß der Stecker richtig herum aufgesteckt wird.

ACHTUNG: REVISIONSÄNDERUNG KEY3 (R2) - KEY3 (R3)

Durch Modifikationen der Tastatur/Mausschnittstelle KEY3 (R2), die der Erhöhung der Zuverlässigkeit und Ausräumung von Timingproblemen dienten, ergaben sich auf der modifizierten Version KEY3 (R3) einige Änderungen, die im folgenden dokumentiert sind.

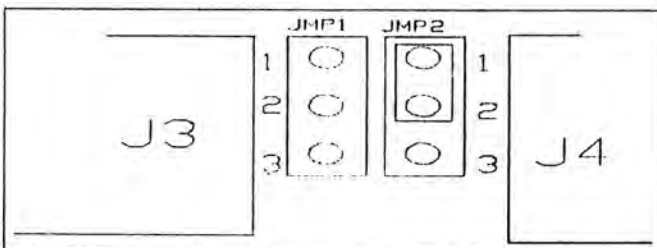
1. Änderungen in der Jumperstellung.

JMP1:



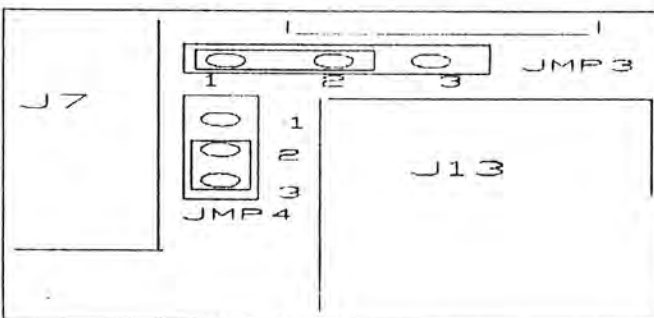
Jmp1 dient der Einstellung des Strobes. In der gezeichneten Position ist der Strobe invertiert. (pos) z.B. für die ASCII-Tastatur. In der alternativen Stellung ist er nicht invertiert.

JMP2:



Jmp2 dient der Spezifikation der Tastatur. In der gezeichneten Position wird die ASCII - Tastatur betrieben, in der anderen die IBM - Tastatur.

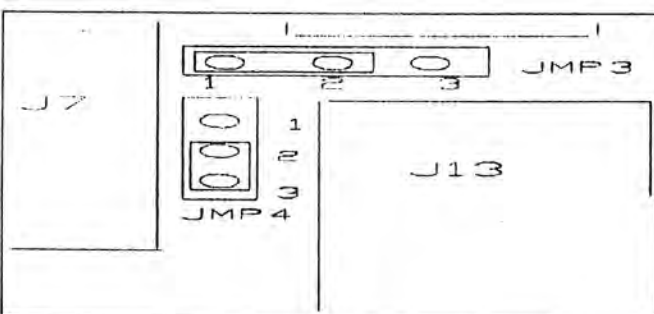
JMP3:



Jmp3 nimmt die Position des Jmp2 auf Version R2 ein und dient der Auswahl der Speichergröße.

Verbindung 1/2: 64 KB
Verbindung 1/3: 256 KB

JMP4:



Jmp4 entspricht Jmp3 auf der alten Version und dient ebenfalls der Auswahl der Speichergröße.

Verbindung 1/3: 256 KB
Verbindung 2/3: 64/128 KB
Voreinstellung: 2/3

2. Änderungen in der Bestückung

Da in der Version Key3 (R3) ein Oszillator verwendet wurde, müssen die Bauteile R4, R5, C16 und X1 (siehe Bestückungsplan) nicht mehr bestückt werden, da diese im Oszillator integriert sind.

ACHTUNG! Wenn die Key3 mit einer ASCII-Tastatur verwendet wird, muß während des Einschaltens des Rechners und während jedes Resets die Leertaste gedrückt werden, damit die Key3 die ASCII-Tastatur erkennt.

6.4 Testen mit dem Z80-Grundprogramm

Wenn Sie das System gestartet haben, wartet der Rechner auf eine Eingabe von Ihnen. Diese Eingabe erscheint unten auf dem Bildschirm in einem kleinen Feld. In diesem Feld erscheint maximal 1 Zeichen. Sobald Sie ein neues Zeichen eingeben, verschwindet das alte Zeichen und der Rechner erwartet eine neue Eingabe. Geben Sie nun das nächste Zeichen ein und kontrollieren Sie, ob das von Ihnen eingegebene Zeichen auf dem Monitor erscheint. Gehen Sie auf diese Weise alle Tasten der Tastatur durch.

6.5. Test mit dem FLOMON

Der Tastaturtest mit dem FLOMON verläuft ziemlich einfach. Sie müssen nur die Control-Taste und die C-Taste gemeinsam drücken. Sie befinden sich dann im Testmodus, und können alle Tasten ihrer Tastatur drücken und sehen, ob das entsprechende Zeichen auf dem Monitor erscheint. Verwenden sie eine PC/XT Tastatur und gibt der Rechner Zeichen aus, die nicht dem gedrückten Zeichen auf der Tastatur entsprechen, betrachten sie den Punkt 6.6..

6.6. Amerikanischer-Deutscher Zeichensatz beim FLOMON

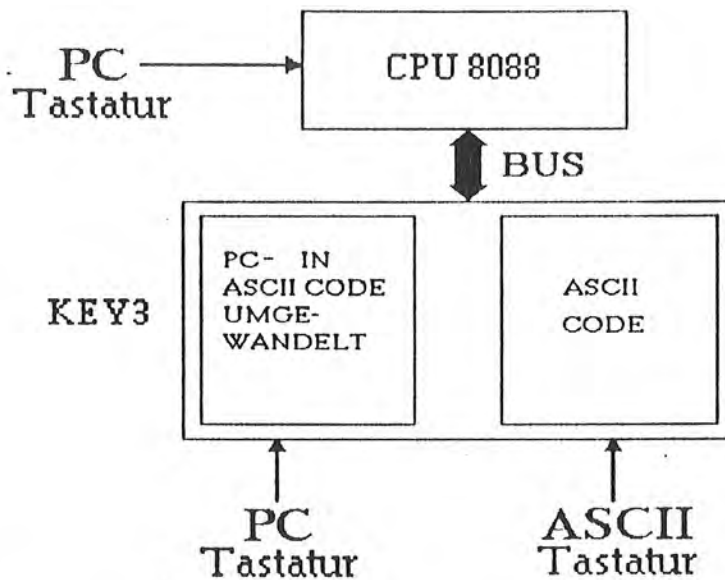
Sollte auf dem Monitor falsche Zeichen erscheinen und sie benutzen eine PC/XT Tastatur, besteht die Möglichkeit, daß diese Tastatur den amerikanischen Zeichensatz anwendet. Um den deutschen Zeichensatz einzustellen, drücken sie nochmals die Control und die C-Taste. Anschließend drücken sie die Escape ("ESC")-Taste. Dann betätigen sie die "z"-Taste und anschließend die "1". Nun ist der deutsche Zeichensatz eingegeben. Wenn sie nun wiederum den Testmodus mit Control C aufrufen, müßen alle Tasten ihr richtiges Signal ergeben. Um den amerikanischen Zeichensatz einzugeben, gehen sie den oben gezeigten Weg, nur anstelle des "1" geben sie den Wert "0" ein.

Zusammenfassung:

deutsch	ESC z 1
amerikanisch	ESC z 0

6.7 Testen mit dem 680xx Grundprogramm

Wenn Sie das Grundprogramm 4.3 besitzen, verläuft der Test wie beim Z80 Grundprogramm. Besitzen Sie jedoch die neuere Version 6.0, gehen Sie folgendermaßen vor. Drücken Sie die Taste "E". Mit dieser Taste wählen Sie den Editor aus, in welchem Sie sich nun befinden müssten. Der Editor ist ein einfaches Textverarbeitungsprogramm, mit dessen Hilfe Sie nun alle Tasten der Tastatur durcharbeiten und kontrollieren können.



6.8 Testen mit der CPU 8088

Wenn Sie eine CPU 8088 besitzen, benötigen Sie die KEY3, wenn Sie eine ASCII-Tastatur verwenden. Sie haben nun aber zwei Möglichkeiten eine PC-Tastatur anzuschließen, da die CPU einen eigenen Tastaturanschluß besitzt. Die KEY3 funktioniert hier zwar auch, sinnvoller ist es jedoch, die PC-Tastatur direkt an die CPU anzuschließen.

Wie Sie sehen, greift die PC-Tastatur im Falle eines Anschlusses an die CPU8088 direkt auf diese zu. Wenn Sie die Tastatur an die KEY3 anschließen, wandelt diese die Eingabesignale in den entsprechenden ASCII-Wert um. Dieser Wert wird zur CPU übertragen, und muß von der CPU wieder in den PC-Code umgewandelt werden.

Wenn Sie die Tastatur testen wollen, starten Sie das System. Schon im Grundmenue können Sie alle Tasten durcharbeiten und die Zeichen am Monitor überprüfen.

7. Fehlersuchanleitung

Sollte Ihre KEY3-Baugruppe bei den im Kapitel "Test der Baugruppe" beschriebenen Tests nicht funktionieren, so heißt es jetzt auf systematische Fehlersuche zu gehen. Wir wollen Ihnen nun ein paar Vorschläge machen, wie eine systematische Fehlersuche vorsich gehen kann:

- Sind die bisher verwendeten Baugruppen in Ordnung ?
(Funktioniert das System ohne Baugruppe ?)
- Sind die Jumpers richtig gesteckt ?

7.1. Sichtprüfung

1. Machen Sie zunächst eine Sichtprobe. Können Sie irgendwo auf der Platine **unsaubere** Lötstellen (zuviel Lötzinn, manchmal zieht das Lötzinn auch Fäden) erkennen, die eventuell einen Kurzschluß verursachen könnten? Dann müssen Sie diese Lötstellen nachlöten und jede unzulässige Verbindung beseitigen.
2. Haben Sie alle IC's richtig herum am richtigen Platz aufgesteckt? Manchmal können beim Einstecken der IC's einzelne Pins weggebogen sein. Da Sie dies durch reine Sichtkontrolle oft nicht erkennen können, sollten Sie jeden Baustein noch einmal herausziehen, kontrollieren und dann erneut einsetzen.
3. Haben Sie den gepolten Kondensator auch richtig eingesetzt?
4. Sind die Netzwerkwiderstände richtig eingelötet ?
5. Haben Sie auch keine Lötstelle vergessen ? (Sehen Sie lieber noch einmal nach.)
6. Sehen Sie irgendwo "kalte" Lötstellen ?
Kalte Lötstellen erkennt man daran, daß sie nicht glänzen. Sie sind im Vergleich mit richtig gelöteten Lötstellen trübe.
7. Haben Sie auch nicht zu heiß gelötet ?
Wenn der LötKolben zu heiß eingestellt ist und (oder) Sie zu lange auf der Lötstelle bleiben, dann kann es passieren, daß sich die Leiterbahnen von der Platine lösen, daß Durchkontaktierungen unterbrochen werden oder, daß Bauteile durch zu heißes Löten zerstört werden.

Sollten Sie nach der Sichtprüfung noch keinen Fehler entdeckt haben, so wird es notwendig, daß man ein Meßgerät (Multimeter, Prüftstift, Oszilloskop etc.) zur Hand nimmt.

7.2. Messungen

Nehmen Sie alle IC's aus ihren Fassungen. Nehmen Sie sich die Layouts zur Hand und kontrollieren Sie alle Leiterbahnen mit einem Durchgangsprüfer oder einem Ohmmeter auf Durchgang. Bereits kontrollierte Bahnen können Sie auf dem Layout mit Bleistift markieren.

Wenn Sie alle Leiterbahnen kontrolliert und nichts gefunden haben, dann ist die Wahrscheinlichkeit groß, daß ein Bauteil defekt ist.

Wenn Sie einen Prüfstift oder ein Oszilloskop haben, dann können Sie jetzt überprüfen, ob Sie an den jeweiligen Ausgängen die richtigen Signale haben. Welche Signale wo anliegen müssen, können Sie der Schaltungsbeschreibung und dem Schaltplan entnehmen.

Falls Sie keine Meßgeräte haben, dann müssen Sie alle Bausteine systematisch austauschen, bis sie das defekte Teil gefunden haben. Verwenden Sie dazu eventuell eine zweite Baugruppe (die eines Freundes oder eines Bekannten).

Sollten Sie gar nicht zu Rande kommen, hilft Ihnen unser Pauschal-Reparatur-Service, dessen Bedingungen Sie der Preisliste entnehmen können.

8. Schaltungsbeschreibung

8.1 Tastaturteil

8.1.1 Z80 - Mikroprozessor - Herz der Baugruppe

Wie schon in der Prinzipbeschreibung gesagt, ist diese Baugruppe eigentlich ein eigener kleiner Computer. Wie bei allen Computern gibt es auch hier einen Prozessor, den Z80 von Zilog (IC1, Z80).

Dieser Prozessor übernimmt auf der KEY 3 den Löwenanteil der Arbeit.

Er liest von der Tastatur kommende Daten, entweder vom parallelen Port (IC5, 74LS374) oder vom seriellen Port (IC4, 74LS245) ein, wandelt die Daten (wenn notwendig) in, für den Computer verständliche, ASCII-Zeichen um (siehe 3.), speichert sie im Tastaturpuffer zwischen, erkennt Funktionstasten und Sondertasten und gibt die Zeichen über einen Puffer (IC 17, 74LS374) an den Bus des Hauptprozessors weiter. Nebenbei kontrolliert er noch, ob der Hauptprozessor ihm (über einen weiteren Puffer, IC18, 74S374) etwas mitteilen will, und reagiert darauf, indem er z.B. die Belegung der Funktionstasten ändert.

Mit zum Prozessorteil gehören auch die Taktlogik und die Speicherbausteine.

Die Taktlogik besteht aus IC13 74LS04, dem Quarz (X1, 4MHz), zwei Widerständen (R3,R4, 1K) und einem Kondensator (C3, 100nF). Durch beschalten (siehe Schaltplan) des Quarzes mit den anderen Bauteilen wird der Quarz in Schwingung versetzt. Da er nur mit seiner eingepprägten Frequenz von 4MHz schwingen kann, liefert er so den benötigten Takt von 4MHz an den Prozessor (IC1, Z80, Pin6).

Zum Betrieb eines Prozessors wird natürlich auch Speicher benötigt. Im Speicher steht einmal das Programm, ohne das der Prozessor nicht weiß was er tun soll. Außerdem wird der Speicher zur Zwischenspeicherung von Daten (z.B. für Tastaturpuffer) benötigt.

Um das Programm beim Ausschalten des Computers nicht zu verlieren, muß es in einem EPROM gespeichert sein. In einem EPROM gehen Daten auch ohne Versorgungsspannung nicht verloren, aber dafür können in ein EPROM im normalen Betrieb (der hier gegeben ist) keine Daten geschrieben werden, sondern nur Daten (die in einem besonderen Verfahren hineinprogrammiert wurden) ausgelesen werden. Im diesem EPROM (IC7, 2764) ist das Programm gespeichert.

Da wie gesagt in ein EPROM nichts geschrieben werden kann, wird noch ein RAM (IC6, 8464) benötigt. Beim RAM kann geschrieben und gelesen werden, aber beim Abschalten gehen alle gespeicherten Daten verloren. Die KEY3 wird mit den 8KByte-Typen 2764 und 8464 geliefert. Durch Auftrennen und Umstecken der Jumper 2 u. 3 können aber auch 32K-Typen (27256 u. 43256) verwendet werden.

Die beiden Speicherbausteine liegen direkt am internen Datenbus der Baugruppe. Vom internen Adressbus sind die Adressen A0-A13 direkt verbunden. Das RAM fühlt sich nur angesprochen, wenn A15 auf high liegt (über IC3, 74LS05). Es ergeben sich damit für das EPROM der Adressbereich von 0 - 1FFFH (- 7FFFFH beim 27256) und für das RAM von 8000H - 9FFFFH (- FFFFFH beim 43256).

Die beiden OR-Gatter (IC2 74LS32) sorgen dafür, daß die beiden Bausteine nur auf Speicherzugriffe des Prozessors (IC1, Z80, Pin19 MERQ ist dann aktiv = low) reagieren.

8.1.2 Adressdekodierungen

Da die Baugruppe sowohl einen eigenen Prozessor hat, als auch über den Bus mit dem Hauptprozessor verbunden ist, sind zwei Adressdekodierungen notwendig.

Durch die interne Dekodierung (IC8, 74LS138) werden die Adressen der verschiedenen Puffer und Ports für den internen Prozessor festgelegt.

Durch die zweite Dekodierung (IC14,15,10,11 alle 74S138) werden die Adressen der Baugruppe (einschließlich Mausteil) aus Sicht des Hauptprozessors festgelegt.

Tabelle:

	Port	Adresse	Eingang/Ausgang
Intern:	parallele Tastatur	5	*
	serielle Tastatur	1	*
	serielle Tast. löschen	0	*
	Ausgabe an BUS	2	*
	Einlesen von BUS	3	*
	Ausgaben von INT an BUS	4	*

Als Baugruppe aus Sicht des Hauptprozessors:

Daten von KEY3 holen	68H	*	
Daten an KEY3 geben	8CH		*
Daten von DIL-Schalter	69H	*	
Mausteil			
Taste gedrückt	8BH	*	
Steps nach oben	8FH	*	
	8EH	*	
	8DH	*	
	8CH	*	
Vorbelegen	8DH		*
Löschen	8EH		*

Etwas verwirrend ist vielleicht die Tatsache, daß die Bausteine IC17 u. IC18 (beide 74LS374) sowohl intern, als auch vom BUS aus angesprochen werden können, und zwar unter verschiedenen Adressen.

Der Baustein IC17 dient als Puffer zur Zeichenausgabe der Baugruppe an den BUS. So kann der interne Prozessor ein Byte, das er ausgeben will, in den Baustein schreiben. Dabei hat der Baustein die Adresse 2. für den internen Prozessor (interne Dekodierung).

Der Hauptprozessor kann ein Byte, das in diesem IC zwischengespeichert wird, abholen, und zwar unter der Adresse 68h (Baugruppendekodierung).

Beim Baustein IC18 ist es genau umgekehrt. Hier wird das Zeichen vom Hauptprozessor

Beim Baustein IC18 ist es genau umgekehrt. Hier wird das Zeichen vom Hauptprozessor hineingeschrieben (in die Adresse 8Ch) und kann vom internen Prozessor (unter der Adresse 3) abgeholt werden.

8.1.3 Ein- und Ausgänge

Um die Verbindung mit der "Außenwelt", sprich zur Tastatur und zum Hauptprozessor zu bekommen, braucht der Prozessor einige Ein- und Ausgänge, sogenannte Ports. Jeder dieser Ports wird über eine bestimmte Adresse angesprochen (siehe 8.1.2).

Der wichtigste Port ist der Eingabeport von der Seriellen Tastatur. Hier werden die seriell ankommenden Daten (siehe 3.3) zuerst durch ein Schieberegister (IC9, 74S164) in parallele Daten umgewandelt.

Beim 74LS164 handelt es sich um ein 8 Bit-Schieberegister, das die an Pin 1 (oder Pin 2) anliegenden Daten bei jeder steigenden Taktflanke übernimmt. Da die Daten aber mit einem Takt geliefert werden, der bei jeder fallenden Flanke gültig ist (siehe 3.3), wird dieser Takt zuerst durch IC3 (74LS05) invertiert.

Obwohl von der Tastatur 9 Bit gesendet werden, ist hier ein 8 Bit-Schieberegister ausreichend, da das erste der gesendeten Bits, das Startbit, nicht zu den Daten gehört. Es wird durch das ganze Schieberegister durchgeschoben und fällt, wenn das letzte Bit ankommt, wieder heraus. So, daß dann genau die 8 Datenbit im Schieberegister stehen. Da nach 9 Bit auch kein Takt mehr gesendet wird, bleiben die Daten stehen, bis das Schieberegister vom internen Prozessor gelöscht wird, oder bis neue Daten von der Tastatur kommen (was aber solange dauert, daß der Prozessor das Schieberegister längst gelöscht hat). Dieses Löschen, das durch ansprechen (schreiben oder lesen) der Adresse 0 geschieht, ist notwendig, da, falls an Ausgang H (Pin 13) des Schieberegister ein High-Pegel liegt, sonst die Interruptleitung des Prozessors blockiert wäre. Doch zurück zum Startbit. Ganz nutzlos ist es in der Schaltung nicht, denn wenn es an der letzten Stelle (Ausgang H, Pin 13) des Schieberegisters angekommen ist, löst es beim Z80 einen Interrupt aus (siehe 8.1.4). Dem Prozessor wird so gemeldet, daß Daten da sind und er diese bei Adresse 1 abholen kann. Dabei wird der Baustein IC4 (74LS245) angesprochen, der dann die Daten auf den internen Datenbus durchschaltet.

Etwas einfacher geht die Sache am Port für die parallele Tastatur vor sich. Hier gibt es den Baustein 74S374 (IC5), der am Eingang anliegenden Daten aufnimmt und zwischenspeichert sobald am CLK-Eingang (Pin 11) eine steigende Flanke auftritt. Diese Flanke wird von der parallelen Tastatur als Strobe (siehe 3.2) mitgeliefert. Außerdem erzeugt dieser Strobe gleich auch noch den Interrupt für den Prozessor.

Dieser kann die Daten jetzt abholen, und zwar bei Adresse 5. Dabei schaltet IC 5 seine zwischenspeicherten Daten auf den Datenbus durch.

Ein Löschen der Daten ist nicht notwendig. Wenn wieder eine Taste gedrückt wird, werden die alten Daten einfach überschrieben.

Dann gibt es noch zwei Ports (IC17, IC18, beide 74LS374) mit dem der interne Prozessor mit dem Hauptprozessor in Verbindung treten kann. Dies ist notwendig da z.B. eine gedrückte Taste dem Hauptprozessor mitgeteilt werden muß.

Es sind zwei Ports notwendig, da beide nur in einer Richtung betrieben werden können und zwar aus der Sicht des internen Prozessors IC17 als Ausgang und IC18 als Eingang.

Da es hier, im Gegensatz zu den Tastaturen, keine Meldeleitung (Interrupt o.ä.) gibt, wissen die Prozessoren voneinander nicht, wann der andere etwas will.

Beim Ausgang (aus der Sicht des internen Prozessors) ist das Problem so gelöst: Der Hauptprozessor fragt immer wieder den Port ab, d.h. er liest die zwischengespeicherten Daten ein und schaut nach ob sie brauchbar sind (ASCII-Zeichen). Wenn ja, merkt er sich die Daten, die ja für eine gedrückte Taste stehen. Wenn er jetzt aber gleich wieder mit dem Abfragen beginnt, erhält er sofort wieder brauchbare Daten, da sich am Port noch gar nichts geändert hat. Für den Hauptprozessor sieht es aber so aus, als sei die gleiche Taste schon wieder gedrückt worden. Um dies zu vermeiden bedient er sich eines Tricks. Er liest die Stellung des DIL-Schalters ein. Dabei interessiert ihn die Stellung der DIL-Schalter überhaupt nicht, aber durch das Ansprechen des DIL-Schalterports wird über die Gatter IC2 (74LS32) u. IC13 (74LS04) das höchstwertige Bit am Ausgangsport (IC17) gesetzt.

Wenn der Hauptprozessor jetzt wieder mit dem Abfragen beginnt, erhält er nicht sofort wieder brauchbare Daten, da jetzt das höchstwertige Bit gesetzt ist, was einer Zahl von größer 127 entspricht. (Brauchbare Zeichen sind ASCII-Zeichen, die nur im Bereich von 0-127 liegen). Außerdem wird durch den Zugriff auf den DIL-Schalter noch dem internen Prozessor über die NMI-Leitung (siehe 8.4) mitgeteilt, daß der Hauptprozessor das ASCII-Zeichen abgeholt hat. Der interne Prozessor kann dann gleich das nächste Zeichen aus dem Tastaturpuffer (sofern eines darin ist) ausgeben. Für den Hauptprozessor ist dann wieder ein gültiges Zeichen da und der Kreislauf beginnt von vorne.

Beim Eingangsport (aus der Sicht des internen Prozessors) ist das Problem anders gelöst. Es wird in Kapitel 10 genau beschrieben.

Auch für den Hauptprozessor enthält die Schaltung einige Ports, und zwar die gerade beschriebenen Verbindungsports zum internen Prozessor. Sie haben aus Sicht des Hauptprozessors nur andere Adressen. Über den DIL-Schalter Port können bestimmte Zahlen, die z.B. Aussagen über die vorhandene Computerkonfiguration machen eingestellt und vom Hauptprozessor eingelesen werden.

Und natürlich gibt es auch für den Mausteil eigene Ports, und zwar gleich sieben (siehe 8.1.2 und 8.2).

8.1.4. Interrupts

Wie in 8.3 beschrieben werden dem internen Prozessor Mitteilungen über die INT-Leitung bzw. über NMI-Leitung gemacht. Wenn auf einer dieser Leitungen ein Signal ankommt, hört das Programm sofort mit der Abarbeitung seines momentan laufenden Programms auf und macht an einer bestimmten Adresse (unterschiedliche Adresse bei NMI und INT) weiter.

Während INT einfach die Abkürzung von Interrupt ist, bedeutet NMI Non Maskable Interrupt, auf deutsch: Nicht sperrbarer Interrupt.

Dies bedeutet, daß der NMI nicht per Software gesperrt, also verhindert werden kann. Dagegen ist es beim INT möglich, ihn durch einen Befehl im Programm zu sperren und ihn durch einen anderen Befehl wieder zuzulassen.

8.2 Mausteil

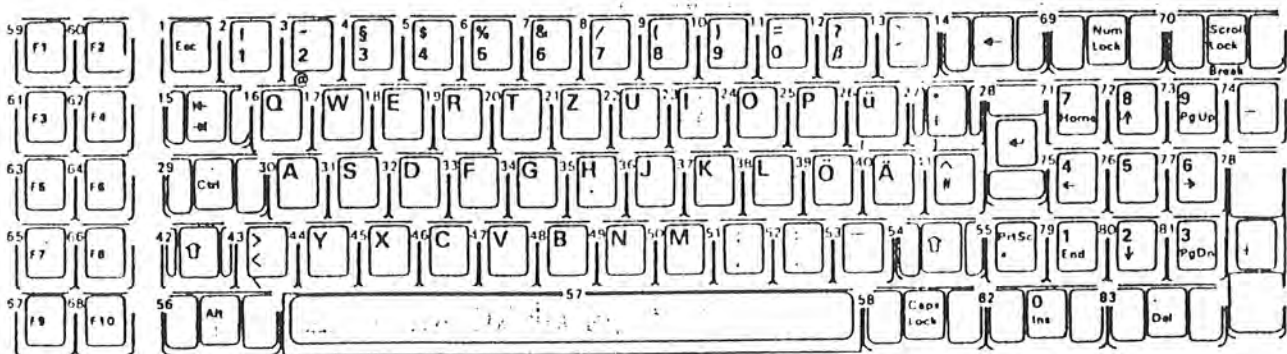
Wie bereits erwähnt, besitzt die Maus für die horizontale und vertikale Bewegung jeweils zwei Ausgänge, die zueinander phasenverschobene Rechtecksignale liefern. Die vier Ausgangssignale der Maus führen zum Triggereingang des Timerbausteins NE556. Entsprechend seiner externen Beschaltung arbeitet der Baustein als monostabiler Impulsgeber. Jede negative Flanke am Triggereingang bewirkt einen kurzen positiven Impuls am Ausgang. Diese Ausgangssignale dienen als Takt für die 8 Bit Zähler 74LS590. Da bei einer Aufwärtsbewegung beide Zähler für die Horizontalbewegung, wegen der Phasenverschiebung, einen Taktimpuls erhalten, jedoch nur einer zählen darf, muß der andere Zähler gesperrt werden. Durch Zuführung der paarweise vertauschten Rechtecksignale auf die Freigabeingänge des Taktes wird entsprechend der Bewegung nur ein Zähler freigegeben. Zur Erfassung der Bewegung der Maus müssen die Inhalte der vier Zähler regelmäßig ausgelesen werden.

Da die Zählerbausteine mit Ausgaberegistern arbeiten, bedarf es vor dem Auslesen der Zählerinhalte eines Schreibzugriffs (beliebiges Datenwort) auf die Adresse 8DH, um eine Übernahme des aktuellen Zählerstandes in das Ausgaberegister zu veranlassen. Der Zugriff auf die Adresse 8DH bewirkt gleichzeitig eine Speicherung der Zählerstände bei allen vier Zählern. Ebenso bewirkt ein Schreibzugriff (beliebiges Datenwort) auf die Adresse 8EH ein gleichzeitiges Löschen aller vier Zähler. Über den Baustein 74LS125 besteht die Möglichkeit, den Zustand von maximal vier Tasten der Maus abzufragen. Bei der standardmäßigen Belegung führen allerdings nur zwei Eingänge zum 9-poligen Maus-Stecker.

9. Bedienungsanleitung

9.1 Grundfunktionen

Im Grundzustand (z.B. nach dem Einschalten) verhält sich eine, an die (funktionstüchtige) Baugruppe, angeschlossene XT-Tastatur genauso wie eine ganz normale Schreibmaschine. Es erscheint immer das Zeichen auf dem Bildschirm, das auch auf der gedrückten Taste aufgedruckt ist. Um diesen und die folgenden Test durchzuführen sollten Sie sich am Besten in einem Editor (z.B. 680xx-Grundprogramm Editor) oder in einem Textverarbeitungsprogramm (z.B. Wordstar) befinden, da auf Menü- oder Betriebssystemebene manche Zeichen (z.B. ÜÖÄü ...) vom Computer nicht beachtet werden.



Solange Sie jetzt keine Sondertaste (im Bild hervorgehoben) drücken bleiben Sie in der Grunde-ebene, d.h. alle Buchstaben erscheinen (wie bei der Schreibmaschine) als Kleinbuchstaben. Bei den Zahlen und beim Zehnerblock erscheinen die Ziffern und nicht die zusätzlich aufgedruckten Zeichen (wie z.B. !"\$%).

9.2 Ebenen

Wenn Sie jetzt z.B. einen Großbuchstaben erzeugen wollen, müssen Sie dazu die 'Ebene' wechseln. Dies hört sich vielleicht etwas verwirrend an, geschieht aber ganz einfach durch Drücken einer der zwei SHIFT-Tasten. Auch dies ist genau wie bei einer Schreibmaschine. Durch das Drücken der SHIFT-Taste haben jetzt alle (fast) Tasten eine andere Bedeutung, und zwar nur solange wie die SHIFT-Taste gedrückt bleibt.

Genauso wird die Ebene durch die Tasten < CTRL > und < ALT > gewechselt. Auch hier ergeben sich für die Tasten andere Zeichen oder Funktionen. Die CTRL-Ebene wird z.B. oft als Befehlsebene (z.B. im Grundpgm.-Editor oder in Wordstar) verwendet, d.h. eine Taste erzeugt auf dem Bildschirm keine Zeichen, sondern löst eine Funktion (wie z.B. 'gehe eine Zeile nach oben'...) aus.

Die ALT-Ebene wird bei der KEY3 auch zur Befehlswiederholung verwendet (siehe 9.4). Auch verschiedene Kombinationen der Sondertasten (z.B. SHIFT und CTRL gedrückt) ergeben verschiedene Ebenen.

Die Zeichen, die zu einer Ebene gehören, können hier nicht alle aufgeführt werden, aber sie sind leicht durch Ausprobieren zu erhalten und lassen sich auch dem Programm (im Teil Datentabelle) leicht entnehmen.

Wichtig sind auch noch die Tasten < CAPSLOCK > und < NUMLOCK >. 'LOCK' bedeutet hier soviel wie 'eingeklinkt', d.h. ihre Wirkung bleibt auch nach dem Loslassen der Taste erhalten.

Die CAPSLOCK-Taste hat die gleiche Wirkung wie die SHIFT-Taste, nur eben auch nach dem Loslassen. Sie kann durch nochmaliges Drücken oder durch Drücken einer SHIFT-TASTE wieder gelöst werden.

Die NUMLOCK-Taste bewirkt ein Umschalten des Zehnerblocks von Zahlentasten zu Steuertasten. Dabei sind nun die aufgedruckten Pfeile als Cursorsteuerung wirksam (funktioniert mit den meisten Textverarbeitungsprogrammen). Alle der so erhaltenen Funktionen stehen in der CTRL-Ebene auch auf dem normalen Tastenfeld zur Verfügung. Im einzelnen gilt:

7 = CTRL -q 8 = CTRL -e 9 = CTRL -r + = CTRL -a
 4 = CTRL -s 5 = CTRL -a 6 = CTRL -d
 1 = CTRL -k 2 = CTRL -x 3 = CTRL -c - = CTRL -f
 0 = CTRL -v . = CTRL -g

Zurück zur Grundebene kommt man durch erneutes Drücken von NUMLOCK.

9.3 Funktionstasten

Es werden von der KEY3 auch die auf jeder XT- oder MF-Tastatur vorhandenen Funktionstasten unterstützt.

Die Funktionstasten sind auf der Tastatur mit F1 - F10 bezeichnet. Bei verschiedenen Tastaturen gibt es auch noch F11 und F12, welche aber von der KEY3 nicht unterstützt wird.

Die zehn vorhanden Funktionstasten stehen in vier Ebenen zur Verfügung (Grundebene, SHIFT gedrückt, CTRL gedrückt, SHIFT und CTRL gedrückt), so daß insgesamt 40 Funktionstasten zur Verfügung stehen. Dabei sind die Funktionstasten vorbelegt, können aber alle ganz einfach selber programmiert werden.

Vorbelegungstabelle:

Grundebene:		CTRL-Ebene:			
F1	=	'dir'	F1	=	'ws'
F2	=	'pip'	F2	=	'm80'
F3	=	'era'	F3	=	
F4	=	'ren'	F4	=	
F5	=	'stat'	F5	=	
F6	=	'type'	F6	=	
F7	=	'a.'	F7	=	
F8	=	'b.'	F8	=	
F9	=	'*.'	F9	=	
F10	=	'*.'	F10	=	

SHIFT-Ebene:

F1 = 'dir'
F2 = 'pip'
F3 = 'era'
F4 = 'ren'
F5 = 'stat'
F6 = 'type'
F7 = 'a.'
F8 = 'b.'
F9 = '*.'
F10 = '*.'

SHIFT-CTRL-Ebene:

F1 = 'ws'
F2 = 'm80'
F3 =
F4 =
F5 =
F6 =
F7 =
F8 =
F9 =
F10 =

Um eine Funktionstaste zu programmieren, also einer beliebigen Zeichenfolge gleichzusetzen, gibt es zwei Möglichkeiten: Erstens direkt von der Tastatur aus durch bestimmte Tastenfolgen (wird im folgenden beschrieben) oder durch ein Programm vom Hauptprozessor aus (wird im Kapitel 10 'Kommunikation' beschrieben).

Um eine Funktionstaste (z.B. F5) von der Tastatur aus zu programmieren geht man wie folgt vor:

- < ALT > drücken und gedrückt halten
- Ebene wählen und gedrückt halten (entfällt bei Grundebene)
- < F5 > drücken und gedrückt halten
- < ALT > loslassen
- gewählte Ebene loslassen
- Zeichenfolge die programmiert werden soll eingeben
- < F5 > loslassen

Jetzt ist F5 neu programmiert. Immer wenn jetzt F5 gedrückt wird, gibt die KEY3 die eingegebene Zeichenfolge aus.

Als Länge der Zeichenfolge sind für jede Funktionstaste 16 Zeichen vorgeben. Es ist zwar möglich, die Tasten mit mehr als 16 Zeichen zu programmieren, nur reicht dann der zum Speichern vorgesehene Platz nicht mehr aus und die nächste Funktionstaste wird überschrieben. Wenn z.B. in F6 'Donaudampfschiffahrt' programmiert werden soll, so erscheint beim Drücken von F6 richtig 'Donaudampfschiffahrt' und beim Drücken von F7 erscheint dann der Teil, der über die vorgesehenen 16 Zeichen hinausragt, hier: 'fahrt'. Das, was vorher in F7 gespeichert war ist gelöscht.

Eine Ausnahme bildet F10 (in allen Ebenen). Da nach F10 keine weitere Funktionstaste mehr kommt, können hier die Zeichenfolgen bis zu 80 Zeichen lang sein.

Zu Definition des Begriffs "Zeichenfolge":

Hierzu zählen nicht nur Buchstaben und Ziffern, sondern auch Steuerzeichen (z.B. < ENTER > oder < CTRL - c > ...) und sogar andere Funktionstasten. Nur < ALT > darf während des Programmierens nicht gedrückt werden.

Dadurch, daß auch andere Funktionstasten zu einer Zeichenfolge zählen, können die Funktionstasten auch ineinander programmiert werden.

Eigene Versuche schaffen hier schnell Klarheit.

9.4 Befehls- und Wortwiederholung

Die KEY3 hat eine eingebaute Befehls- und Wortwiederholung. Dabei können auf Tastendruck einer(s) der letzten sieben Befehle (Worte) wiederholt werden.

Diese Möglichkeit erweist sich vor allem auf Betriebssystemebene als sehr bequem und zeitsparend.

Die Erklärung ist an einem Beispiel am verständlichsten:

Folgende Eingaben wurden gemacht (Beispiel auf CP/M)

```
A> dir
A> dir *.com
A> dir *.exe
A> B:
B>
```

Jetzt möchte man auf Laufwerk B "dir *.com" aufrufen. Dazu drückt man jetzt < Alt > und < 3 > (3, da "dir *.com" der drittletzte Befehl war). Sofort erscheint "dir *.com". Genauso kann man bis zu sieben Befehle zurück alle Befehle durch < Alt > und < 1 - 7 > wieder aufrufen. Wenn jetzt in unserem Beispiel "dir *.exe" aufgerufen werden soll, so geschieht dies wieder durch < Alt > und < 3 >, da durch das Wiederholen von "dir *.com", "dir *.exe" jetzt der drittletzte Befehl ist.

Auch hier erhält man durch eigene Versuche schnell Klarheit.

Genauso, wie beschrieben, verhält es sich bei der Wortwiederholung, nur daß hier nicht die letzten sieben Befehle sondern die letzten sieben Worte wiederholt werden können. Der Aufruf der Wortwiederholung erfolgt durch die Tastenkombination < Alt > und < "Taste unter der Zahl" >, also z.B. beim vorletzten Wort < Alt > und < w >.

Die Wortwiederholung ist in Version 1.0 noch nicht implementiert.

9.5 Reset

Genauso, wie ein Computer, kann sich auch die Tastaturkarte einmal "aufhängen", z.B. durch elektromagnetische Störungen (Lichteinschalten, o.ä.) oder durch nicht definierte Tastenkombinationen. Dies wirkt sich so aus, daß von der KEY3 kein Zeichen mehr an den Computer weitergegeben wird. Um dann nicht einen kompletten Reset des ganzen Computers machen zu müssen, was den Verlust der gerade bearbeiteten Daten nach sich zieht, gibt es auf der KEY3 eine Resettaste: Dies ist die Taste < SCROLL LOCK >.

Dabei wird der Reset nur von der Tastaturbaugruppe durchgeführt, der restliche Computer bleibt davon unbeeindruckt.

Wenn also einmal Nichts mehr geht, zuerst die < SCROLL LOCK > Taste betätigen. Sofern der Fehler von der KEY3 kam, kann dann ganz normal weitergearbeitet werden.

9.6 ASCII-Tastatur

Beim Betrieb mit einer ASCII-Tastatur ist folgendes zu beachten:

Während des Einschaltens des Rechners und während jedes Resets muß die Leertaste der

Tastatur gedrückt werden, da sonst das Key3-Programm nicht erkennt, daß eine ASCII-Tastatur angeschlossen ist.
Ein Tastaturpuffer für die ASCII-Tastatur ist bei Version 1.0 noch nicht vorgesehen.

9.7 AT-Tastatur

Wenn die Key3 mit einer umschaltbaren AT-Tastatur betrieben werden, so ist es je nach Tastatur möglich, daß einige Zeichen eine andere als die aufgedruckte Funktion haben (es sind aber immer alle Zeichen vorhanden).
Die Steuertasten wie z.B. -> oder 'Entf' stimmen im XT-Betrieb (der eingestellt sein muß!) immer mit den entsprechenden Tasten des Zehnerblocks überein.

10. Kommunikation

Wie schon mehrmals angedeutet, ist es auch möglich, vom Hauptprozessor aus mit dem KEY3-Prozessor in Verbindung zu treten. Dabei ist ein Datenaustausch in beide Richtungen möglich. Der Austausch erfolgt, wie schon unter 8.1.3 beschrieben, über die Ein-, Ausgabeports. Jeder Prozessor hat dabei einen Port, in den er schreibt und einen Port, vom dem er liest. Der Schreibport des einen ist der Leseport des anderen Prozessors.

Die Ports haben die Adressen:

KEY3-Prozessor:	Schreibport:	2
	Leseport:	3
Hauptprozessor:	Schreibport:	8Ch
	Leseport:	68H

Ein Datenaustausch kann nur vom Hauptprozessor aus begonnen werden. In der jetzigen Programmversion gibt es vier verschiedene Arten des Datenaustausches. Jede dieser Möglichkeiten wird durch Anlegen eines bestimmten Codes vom Hauptprozessor an Port 8Ch aktiviert.

Die Codes sind:

"01"	Daten von der KEY3 auslesen
"02"	Daten in die KEY3 einlesen
"03"	Uhr auf der KEY3 stellen
"04"	Uhr von der KEY3 auslesen

Der Datenaustausch der HOST-CPU mit der KEY3 ist bei der jetzt ausgelieferten Version 1.0 noch nicht implementiert.

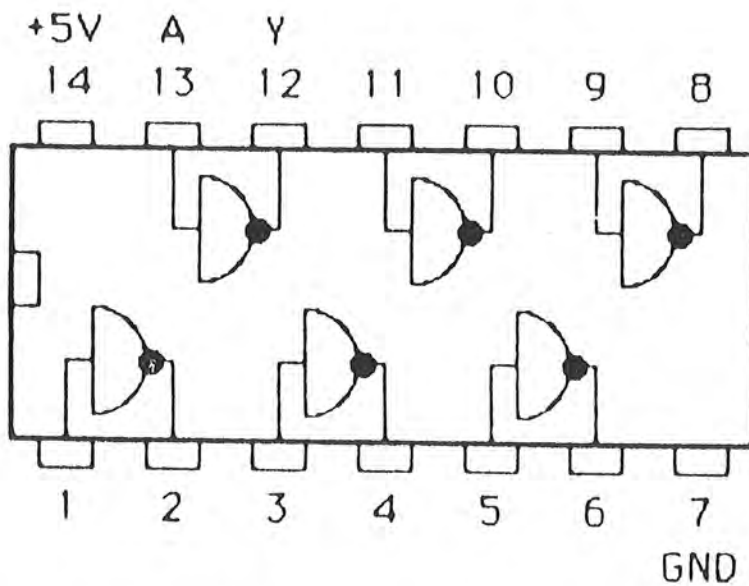
11. Bauelemente

11.1 TTL-Bausteine

11.1.1. 7404

7404

6 Inverter



Logiktablelle

A	Y
0	1
1	0

Typ Impuls-
Verzögerungszeit: 9 ns

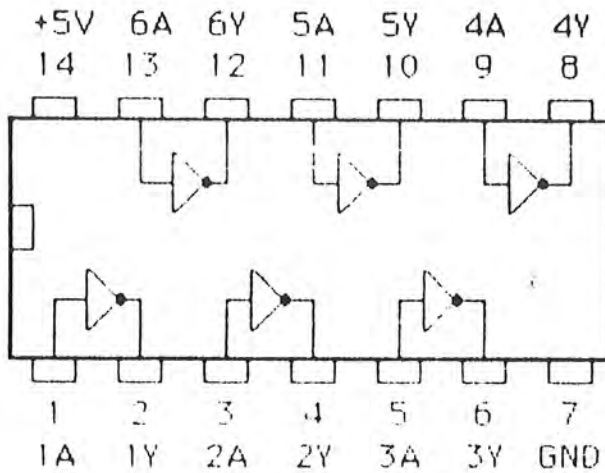
Typ Versor-
gungsstrom: 25 mA

positive Logik:
 $Y = \bar{A}$

11.1.2 74LS05

74LS 05

Sechs Inverter (open Collector)



Positive Logik $Y = \bar{A}$

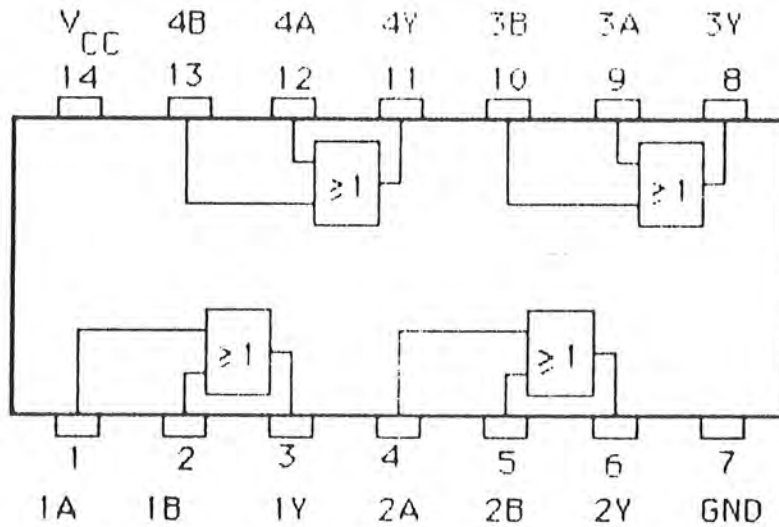
Typ Impuls-
Verzögerungszeit 22 ns

Typ Leistungs-
aufnahme 60 mW

11.1.3 74LS32

74LS32

Vier Or-Gatter mit je 2 Eingängen



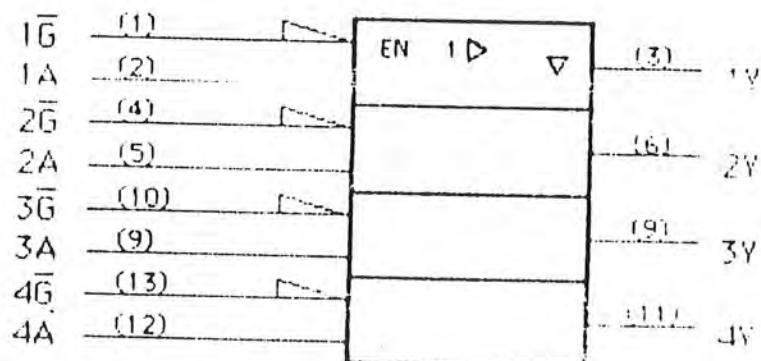
Typ. Impulsverzögerungszeit: 12 ns

Typ. Versorgungsstrom 4 mA

11.1.4 74LS125

74LS125

4 Bus-Leitungstreiber



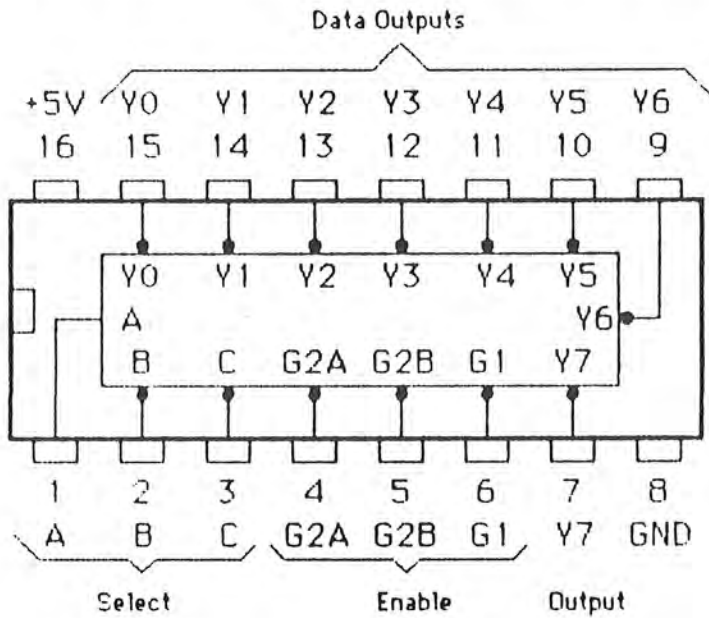
Typ. Impulsverzögerungszeit 10 ns

Typ. Leistungsaufnahme: 55 mW

11.1.5 74LS138

74LS138

3-Bit Binärdekode/Demultiplexer (3 zu 8)



Logiktablelle

Inputs					Outputs							
Enable		Select			Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
G1	G2*	C	B	A								
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	H	L	L	H	H	H	L	H	H	H	H
H	L	H	L	H	H	H	H	H	L	H	H	H
H	L	H	H	L	H	H	H	H	H	L	H	H
H	L	H	H	H	H	H	H	H	H	H	L	H

Positive Logik

*G2 = G2A + G2B

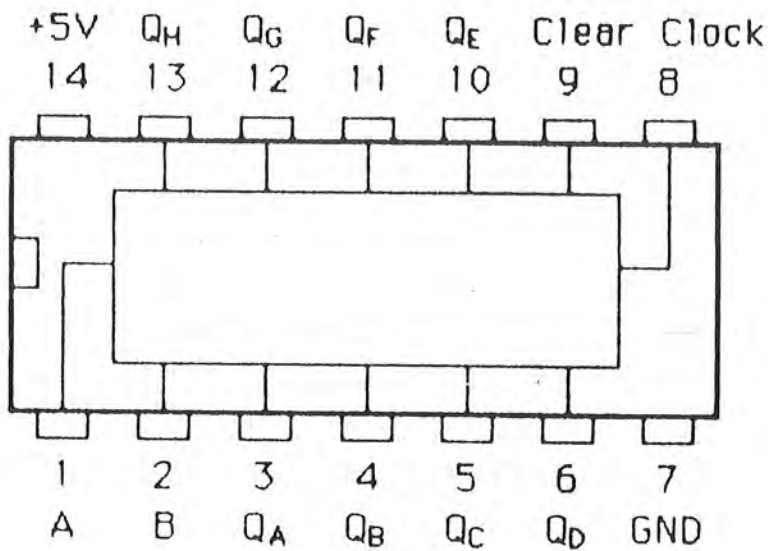
Typ. Impulsverzögerungszeit 22 ns

Typ. Versorgungsstrom 7 mA

11.1.6 74LS164

74LS164

Schieberegister mit 8-Bit paralleler Ausgabe



Function Table:

INPUTS				OUTPUTS		
Clear	Clock	A	B	QA	QB	QH
L	x	x	x	L	L	L
H	L	x	x	QA0	QB0	QH0
H	↑	H	H	H	QAn	QGn
H	↑	L	x	L	QAn	QGn
H	↑	x	L	L	QAn	QGn

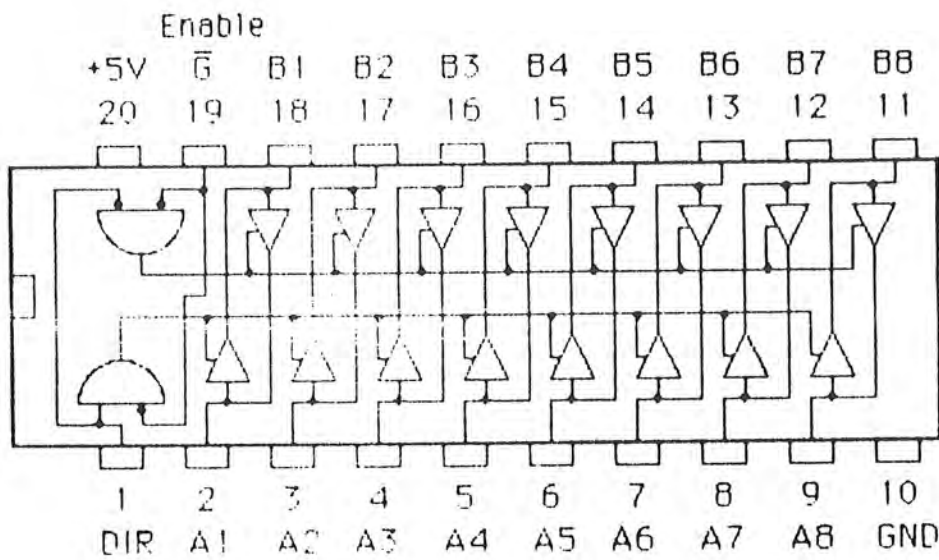
Typ. Impulsverzögerungszeit: 15 ns

Typ. Versorgungsstrom: 20 mA

11.1.7 74LS245

74LS245

8-fach Bus-Transceiver mit 3-state Ausgängen



Function Table

ENABLE	DIRECTION CONTROL	OPERATION
\bar{B}	DIP	
L	H	B data to A bus
H	L	A data to B bus
		Isolation

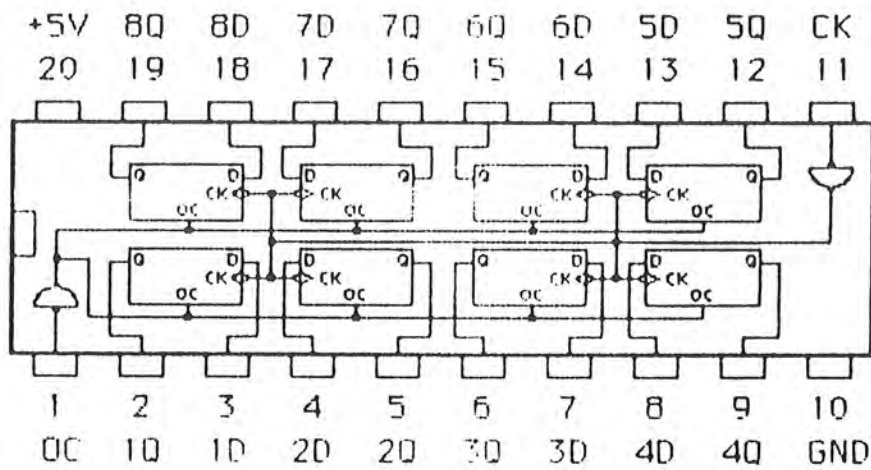
Typ Impuls-
Verzögerungszeit: 20 ns

Typ Versor-
gungsstrom: 75 mA

11.1.8 74LS374

74LS374

8-Bit D Register mit 3-state-Ausgängen



Logiktablelle:

OC	CK	D	Q
0	↑	1	1
0	↑	0	0
0	0	X	Q _o
1	X	X	Z

Typ. Impuls-
Verzögerungszeit: 16 ns

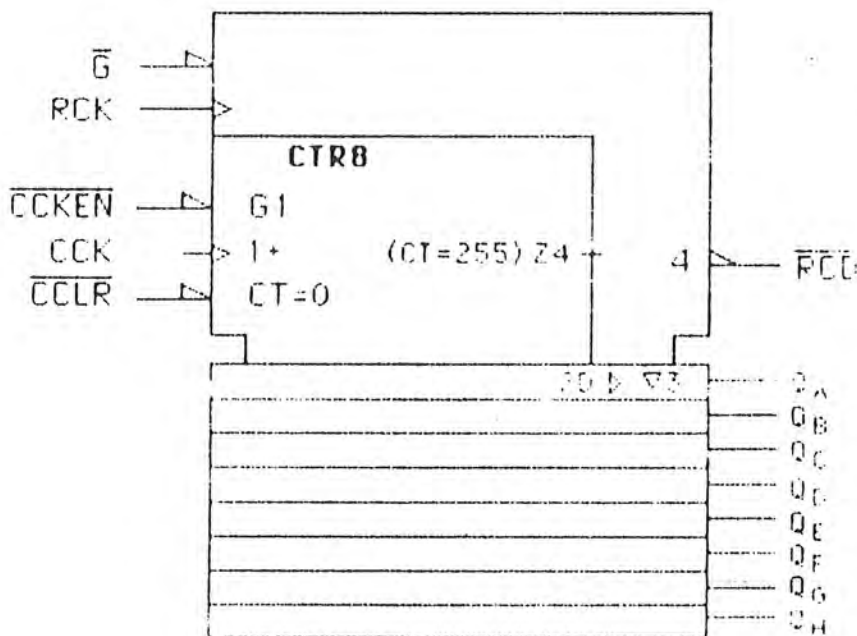
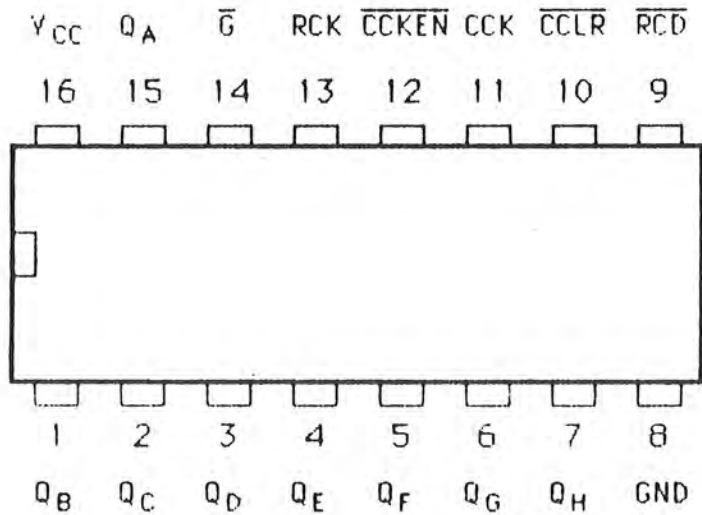
Typ. Versor-
gungsstrom: 2.6 mA

positive Logik ja

11.1.9 74LS590

74LS590

8 Bit Binärzähler mit Ausgangsregister

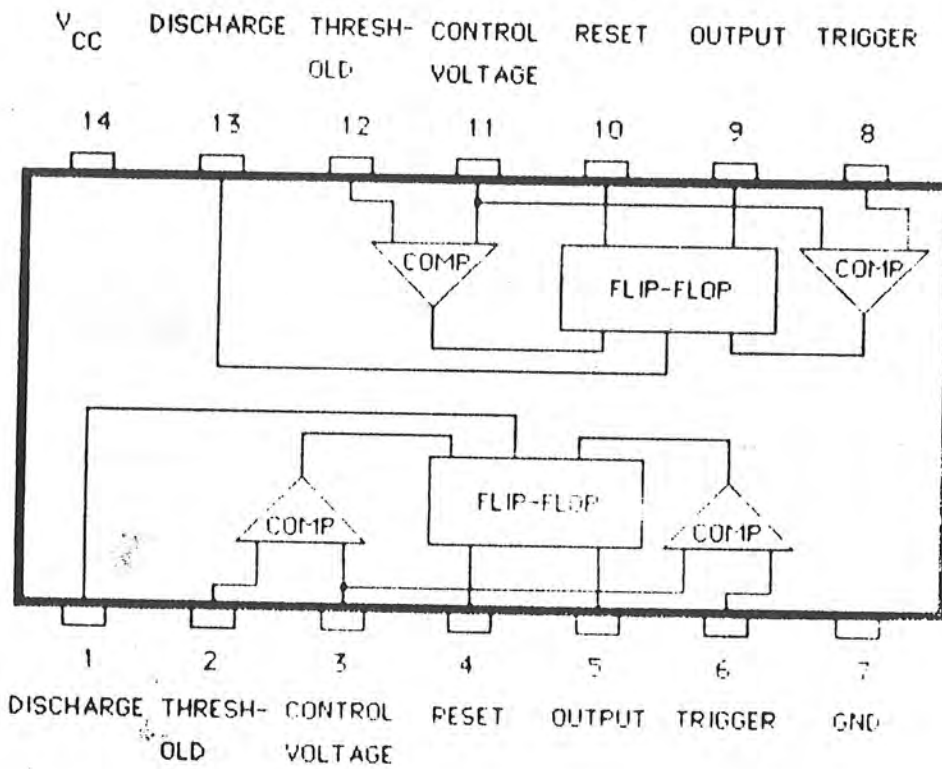


Min. garantierte Zählfrequenz: 20 MHz
 Typ. Leistungsaufnahme: 205 mW

11.2 LM556

LM556

Doppel-Timer



Maximaler Versorgungsstrom ca 30 mA

Maximale Versorgungsspannung 18 V

11.3 CMOS STATC RAM 6264

8192-word x 8-bit High Speed CMOS Static RAM

■ **FEATURES**

- Fast access Time 100ns/120ns/150ns (max.)
- Low Power Standby Standby 0.1mW (typ.)
10μW (typ.) L-/LL-version
- Low Power Operation Operating: 200mW/MHz (typ.)
- Single +5V Supply
- Completely Static Memory No clock or Timing Strobe Required
- Equal Access and Cycle Time
- Common Data Input and Output, Three State Output
- Directly TTL Compatible: All Input and Output
- Standard 28pin Package Configuration
- Pin Out Compatible with 64K EPROM HN482764
- Capability of Battery Back Up Operation (L-/LL-version)

■ **ORDERING INFORMATION**

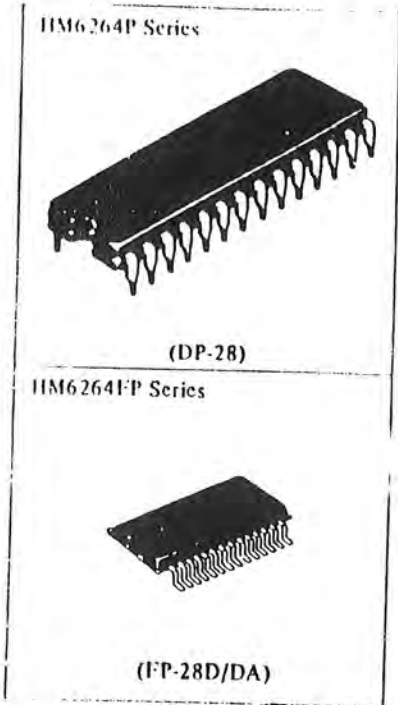
Type No.	Access Time	Package	
HM6264P-10	100ns	600 mil 28 pin Plastic DIP	
HM6264P-12	120ns		
HM6264P-15	150ns		
HM6264LP-10	100ns		
HM6264LP-12	120ns		
HM6264LP-15	150ns		
HM6264LP-10L	100ns		28 pin Plastic SOP (Note)
HM6264LP-12L	120ns		
HM6264LP-15L	150ns		
HM6264FP-10	100ns		
HM6264FP-12	120ns		
HM6264FP-15	150ns		
HM6264LFP-10	100ns		
HM6264LFP-12	120ns		
HM6264LFP-15	150ns		
HM6264LFP-10L	100ns		
HM6264LFP-12L	120ns		
HM6264LFP-15L	150ns		

Note) A character L is added to the end of type No. for SOP of 1.00 mm (max.) thickness

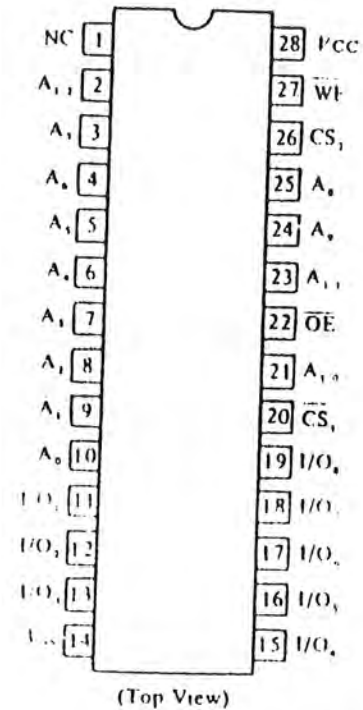
■ **ABSOLUTE MAXIMUM RATINGS**

Item	Symbol	Rating	Unit
Terminal Voltage *1	V _T	-0.5 ⁺² to +7.0	V
Power Dissipation	P _T	1.0	W
Operating Temperature	T _{opr}	0 to +70	°C
Storage Temperature	T _{stg}	-55 to +125	°C
Storage Temperature Under Bias	T _{bias}	-10 to +85	°C

Notes) *1. With respect to V_{SS}
*2. -3.0V for pulse width ≤ 50ns



■ **PIN ARRANGEMENT**



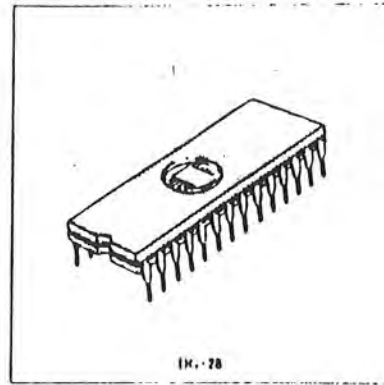
11.4 EPROM 2764

8192-word x 8-bit UV Erasable and Programmable Read Only Memory

The HN482764G is a 8192 word by 8 bit erasable and electrically programmable ROM. This device is packaged in a 28 pin dual-in-line package with transparent lid. The transparent lid on the package allows the memory content to be erased with ultraviolet light.

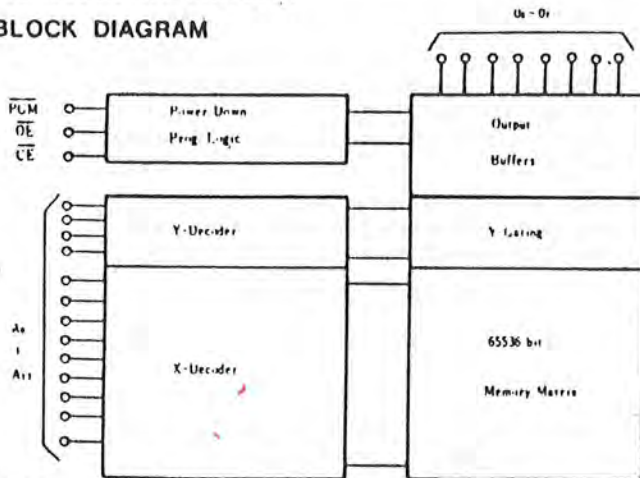
■ FEATURES

- Single Power Supply +5V ± 5%
- Simple Programming Program Voltage: +21V D.C.
Program with one 50ms Pulse
- Static No Clocks Required
- Inputs and Outputs TTL Compatible During Both Read and Program Mode.
- Access Time HN482764G-2 200ns max
HN482764G 250ns max
HN482764G-3 300ns max
- High Performance Programming Available
- Low Standby Current 35mA max.
- Compatible with Intel 2764

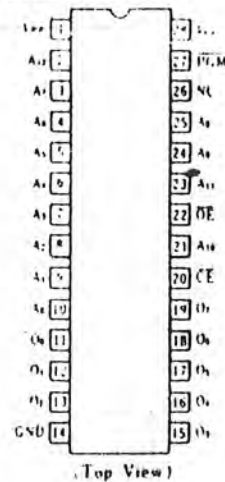


HN-78

■ BLOCK DIAGRAM



■ PIN ARRANGEMENT



(Top View)

■ MODE SELECTION

Mode	Pins	CE (20)	OE (22)	PGM (27)	V _{PP} (1)	V _{IL} (28)	Outputs (11-13, 15-19)
Read		V _{IL}	V _{IL}	V _{IN}	V _{CC}	V _{CC}	Dout
Stand-by		V _{IN}	x	x	V _{CC}	V _{CC}	High Z
Program		V _{IL}	x	V _{IL}	V _{PP}	V _{CC}	Din
Program Verify		V _{IL}	V _{IL}	V _{IN}	V _{PP}	V _{CC}	Dout
Program Inhibit		V _{IN}	x	x	V _{PP}	V _{CC}	High Z

x : don't care

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Operating Temperature Range	T _{op}	0 to +70	°C
Storage Temperature Range	T _{stg}	-65 to +125	°C
All Input and Output Voltage*	V _I	-0.6 to +7	V
V _{PP} Voltage*	V _{PP}	-0.6 to +26.5	V

* : with respect to GND

Note: This device is not available for new application.

11.5 Microprozessor Z80

Z8400
Z80[®] CPU Central
Processing Unit

Zilog

Product
Specification

April 1985

FEATURES

- The instruction set contains 158 instructions. The 78 instructions of the 8080A are included as a subset; 8080A software compatibility is maintained.
- Eight MHz, 6 MHz, 4 MHz, and 2.5 MHz clocks for the Z80H, Z80B, Z80A, and Z80 CPU result in rapid instruction execution with consequent high data throughput.
- The extensive instruction set includes string, bit, byte, and word operations. Block searches and block transfers, together with indexed and relative addressing, result in the most powerful data handling capabilities in the microcomputer industry.
- The Z80 microprocessors and associated family of peripheral controllers are linked by a vectored interrupt

system. This system may be daisy-chained to allow implementation of a priority interrupt scheme. Little, if any, additional logic is required for daisy-chaining.

- Duplicate sets of both general-purpose and flag registers are provided, easing the design and operation of system software through single-context switching, background-foreground programming, and single level interrupt processing. In addition, two 16-bit index registers facilitate program processing of tables and arrays.
- There are three modes of high speed interrupt processing: 8080 similar, non-Z80 peripheral device, and Z80 Family peripheral with or without daisy chain.
- On-chip dynamic memory refresh counter

Z80 CPU

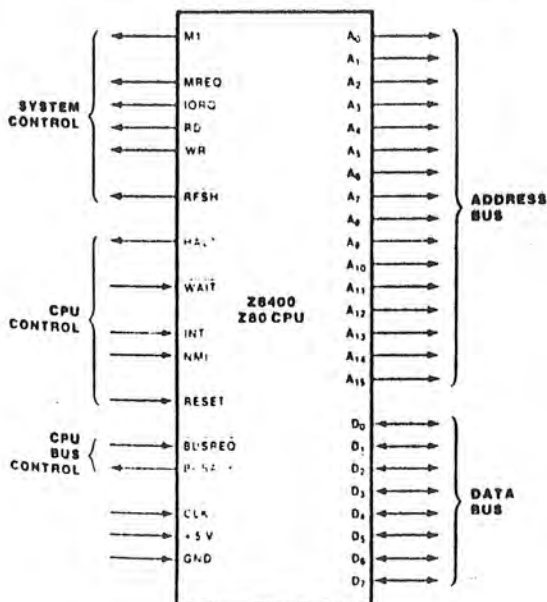


Figure 1. Pin Functions



Figure 2a. 40-Pin Dual-In-Line Package (DIP) Pin Assignments

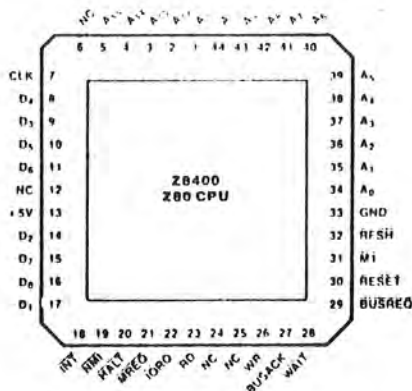


Figure 2b. 44-Pin Chip Carrier Pin Assignments

GENERAL DESCRIPTION

The Z80, Z80A, Z80B, and Z80H CPUs are third-generation single-chip microprocessors with exceptional computational power. They offer higher system throughput and more efficient memory utilization than comparable second- and third-generation microprocessors. The internal registers contain 208 bits of read/write memory that are accessible to the programmer. These registers include two sets of six general-purpose registers which may be used individually as either 8-bit registers or as 16-bit register pairs. In addition, there are two sets of accumulator and flag registers. A group of "Exchange" instructions makes either set of main or alternate registers accessible to the programmer. The alternate set allows operation in foreground-background mode or it may be reserved for very fast interrupt response.

The Z80 also contains a Stack Pointer, Program Counter, two index registers, a Refresh register (counter), and an Interrupt register. The CPU is easy to incorporate into a system since it requires only a single +5V power source. All output signals are fully decoded and timed to control standard memory or peripheral circuits; the CPU is supported by an extensive family of peripheral controllers. The internal block diagram (Figure 3) shows the primary functions of the Z80 processors. Subsequent text provides more detail on the Z80 I/O controller family, registers, instruction set, interrupts and daisy chaining, and CPU timing.

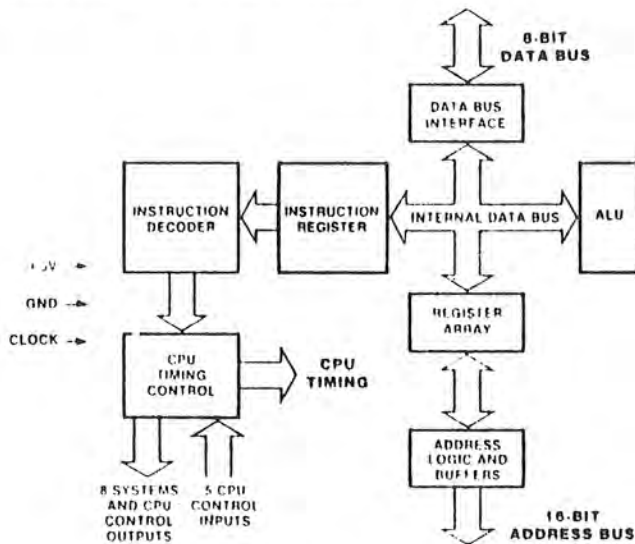


Figure 3. Z80 CPU Block Diagram

Z80 MICROPROCESSOR FAMILY

The Zilog Z80 microprocessor is the central element of a comprehensive microprocessor product family. This family works together in most applications with minimum requirements for additional logic, facilitating the design of efficient and cost effective microcomputer-based systems.

Zilog has designed five components to provide extensive support for the Z80 microprocessor. These are:

- The PIO (Parallel Input/Output) operates in both data-byte I/O transfer mode (with handshaking) and in bit mode (without handshaking). The PIO may be configured to interface with standard parallel peripheral devices such as printers, tape punches, and keyboards.
- The CIC (Counter/Timer Circuit) features four programmable 8 bit counter/timers, each of which has an

8 bit prescaler. Each of the four channels may be configured to operate in either counter or timer mode.

- The DMA (Direct Memory Access) controller provides dual port data transfer operations and the ability to terminate data transfer as a result of a pattern match.
- The SIO (Serial Input/Output) controller offers two channels. It is capable of operating in a variety of programmable modes for both synchronous and asynchronous communication, including Bi-Synch and SDLC.
- The DART (Dual Asynchronous Receiver/Transmitter) device provides low cost asynchronous serial communication. It has two channels and a full modem control interface.

Z80 CPU REGISTERS

Figure 4 shows three groups of registers within the Z80 CPU. The first group consists of duplicate sets of 8 bit registers: a principal set and an alternate set (designated by 'prime', e.g., A'). Both sets consist of the Accumulator Register, the Flag Register, and six general-purpose registers. Transfer of data between these duplicate sets of registers is accomplished by use of "Exchange" instructions. The result is faster response to interrupts and easy, efficient implementation of such versatile pro-

gramming techniques as background-foreground data processing. The second set of registers consists of six registers with assigned functions. These are the I (Interrupt Register), the R (Refresh Register), the IX and IY (Index Registers), the SP (Stack Pointer), and the PC (Program Counter). The third group consists of two interrupt status flip-flops, plus an additional pair of flip-flops which assists in identifying the interrupt mode at any particular time. Table 1 provides further information on these registers.

Z80 CPU

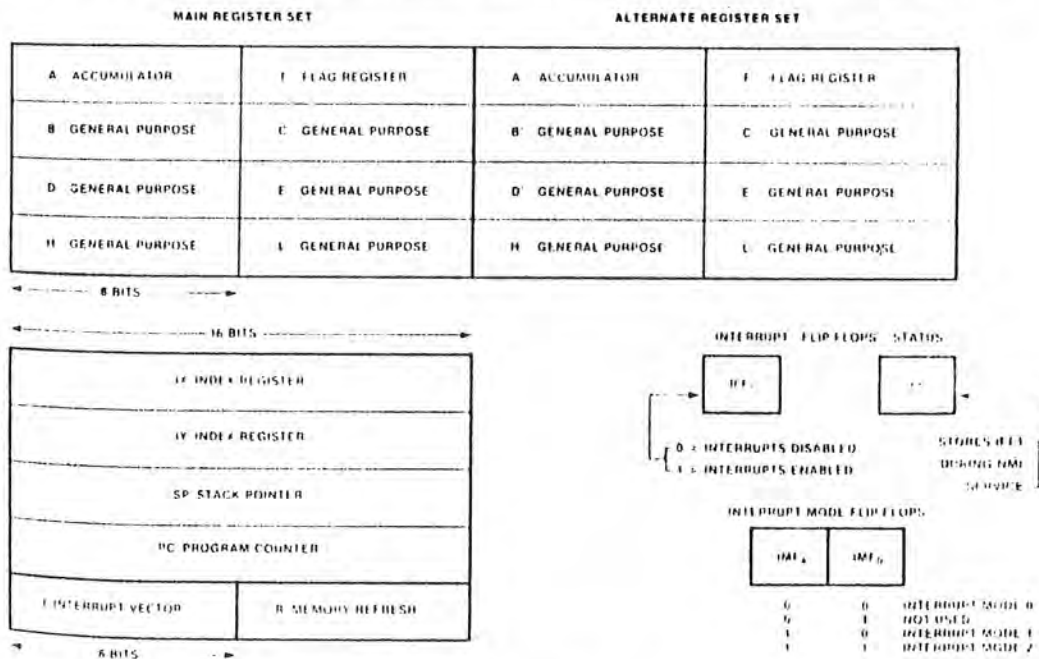


Figure 4. CPU Registers

Z80 CPU REGISTERS (Continued)

Table 1. Z80 CPU Registers

Register	Size (Bits)	Remarks	
A, A'	Accumulator	8	Stores an operand or the results of an operation
F, F'	Flags	8	See Instruction Set
B, B'	General Purpose	8	Can be used separately or as a 16 bit register with C
C, C'	General Purpose	8	See B, above.
D, D'	General Purpose	8	Can be used separately or as a 16 bit register with E
E, E'	General Purpose	8	See D, above.
H, H'	General Purpose	8	Can be used separately or as a 16 bit register with L
L, L'	General Purpose	8	See H, above.
<p>Note: The (B,C), (D,E), and (H,L) sets are combined as follows:</p> <p>B — High byte C — Low byte D — High byte E — Low byte H — High byte L — Low byte</p>			
I	Interrupt Register	8	Stores upper eight bits of memory address for vectored interrupt processing.
R	Refresh Register	8	Provides user-transparent dynamic memory refresh. Automatically incremented and placed on the address bus during each instruction fetch cycle.
IX	Index Register	16	Used for indexed addressing
IY	Index Register	16	Used for indexed addressing
SP	Stack Pointer	16	Holds address of the top of the stack. See Push or Pop in instruction set.
PC	Program Counter	16	Holds address of next instruction
IFF ₁ -IFF ₂	Interrupt Enable	Flip-Flops	Set or reset to indicate interrupt status (see Figure 4)
IMFa-IMFb	Interrupt Mode	Flip-Flops	Reflect Interrupt mode (see Figure 4)

INTERRUPTS: GENERAL OPERATION

The CPU accepts two interrupt input signals, $\overline{\text{NMI}}$ and $\overline{\text{INT}}$. The $\overline{\text{NMI}}$ is a non-maskable interrupt and has the highest priority. $\overline{\text{INT}}$ is a lower priority interrupt and it requires that interrupts be enabled in software in order to operate. $\overline{\text{INT}}$ can be connected to multiple peripheral devices in a wired OR configuration.

The Z80 has a single response mode for interrupt service for the non-maskable interrupt. The maskable interrupt, $\overline{\text{INT}}$, has three programmable response modes available. These are:

- Mode 0 — similar to the 8080 microprocessor
- Mode 1 — Peripheral Interrupt service for use with non-8080/Z80 systems
- Mode 2 — a vectored interrupt scheme, usually daisy-chained, for use with Z80 Family and compatible peripheral devices

The CPU services interrupts by sampling the $\overline{\text{NMI}}$ and $\overline{\text{INT}}$ signals at the rising edge of the last clock of an instruction. Further interrupt service processing depends upon the type of interrupt that was detected. Details on interrupt responses are shown in the CPU Timing Section.

Non-Maskable Interrupt ($\overline{\text{NMI}}$). The nonmaskable interrupt cannot be disabled by program control and therefore will be accepted at all times by the CPU. $\overline{\text{NMI}}$ is usually reserved for servicing only the highest priority type interrupts, such as that for orderly shutdown after power failure has been detected. After recognition of the $\overline{\text{NMI}}$ signal (providing $\overline{\text{BUSREQ}}$ is not active), the CPU jumps to restart location #000H. Normally, software starting at this address contains the interrupt service routine.

Maskable Interrupt ($\overline{\text{INT}}$). Regardless of the interrupt mode set by the user, the Z80 response to a maskable interrupt input follows a common timing cycle. After the

interrupt has been detected by the CPU (provided that interrupts are enabled and $\overline{\text{BUSREQ}}$ is not active) a special interrupt processing cycle begins. This is a special fetch ($\overline{\text{M1}}$) cycle in which $\overline{\text{IORQ}}$ becomes active rather than $\overline{\text{MREQ}}$, as in a normal $\overline{\text{M1}}$ cycle. In addition, this special $\overline{\text{M1}}$ cycle is automatically extended by two $\overline{\text{WAIT}}$ states, to allow for the time required to acknowledge the interrupt request.

Mode 0 Interrupt Operation. This mode is similar to the 8080 microprocessor interrupt service procedures. The interrupting device places an instruction on the data bus. This is normally a Restart instruction, which will initiate a call to the selected one of eight restart locations in page zero of memory. Unlike the 8080, the Z80 CPU responds to the Call instruction with only one interrupt acknowledge cycle followed by two memory read cycles.

Mode 1 Interrupt Operation. Mode 1 operation is very similar to that for the NMI. The principal difference is that the Mode 1 interrupt has only one restart location, 0038H.

Mode 2 Interrupt Operation. This interrupt mode has been designed to utilize most effectively the capabilities of the Z80 microprocessor and its associated peripheral family. The interrupting peripheral device selects the starting address of the interrupt service routine. It does this by placing an 8-bit vector on the data bus during the interrupt acknowledge cycle. The CPU forms a pointer using this byte as the lower 8 bits and the contents of the I register as the upper 8 bits. This points to an entry in a table of addresses for interrupt service routines. The CPU then jumps to the routine at that address. This flexibility in selecting the interrupt service routine address allows the peripheral device to use several different types of service routines. These routines may be located at any available location in memory. Since the interrupting device supplies the low-order byte of the 2-byte vector, bit 0 (A_0) must be a zero.

Interrupt Priority (Daisy Chaining and Nested Interrupts). The interrupt priority of each peripheral device is determined by its physical location within a daisy-chain configuration. Each device in the chain has an interrupt enable input line (IEI) and an interrupt enable output line (IEO), which is fed to the next lower priority device. The first device in the daisy chain has its IEI input hardwired to a High

level. The first device has highest priority, while each succeeding device has a corresponding lower priority. This arrangement permits the CPU to select the highest priority interrupt from several simultaneously interrupting peripherals.

The interrupting device disables its IEO line to the next lower priority peripheral until it has been serviced. After servicing, its IEO line is raised, allowing lower priority peripherals to demand interrupt servicing.

The Z80 CPU will nest (queue) any pending interrupts or interrupts received while a selected peripheral is being serviced.

Interrupt Enable/Disable Operation. Two flip-flops, IFF₁ and IFF₂, referred to in the register description, are used to signal the CPU interrupt status. Operation of the two flip-flops is described in Table 2. For more details, refer to the *Z80 CPU Technical Manual* (03-0029 01) and *Z80 Assembly Language Programming Manual* (03 0002 01).

Table 2. State of Flip-Flops

Action	IFF ₁	IFF ₂	Comments
CPU Reset	0	0	Maskable interrupt INT disabled
DI instruction execution	0	0	Maskable interrupt INT disabled
EI instruction execution	1	1	Maskable interrupt INT enabled
LD A,I instruction execution	•	•	IFF ₂ → Parity flag
LD A,R instruction execution	•	•	IFF ₂ → Parity flag
Accept NMI	0	IFF ₁	IFF ₁ → IFF ₂ (Maskable interrupt INT disabled)
RETN instruction execution	IFF ₂	•	IFF ₂ → IFF ₁ at completion of an NMI service routine

PIN DESCRIPTIONS

A₀-A₁₅. *Address Bus* (output, active High, 3-state). A₀-A₁₅ form a 16 bit address bus. The Address Bus provides the address for memory data bus exchanges (up to 64K bytes) and for I/O device exchanges.

BUSACK. *Bus Acknowledge* (output, active Low). Bus Acknowledge indicates to the requesting device that the CPU address bus, data bus, and control signals MREQ, IORQ, RD, and WR have entered their high-impedance states. The external circuitry can now control these lines.

BUSREQ. *Bus Request* (input, active Low). Bus Request has a higher priority than IORQ and is always recognized at the end of the current machine cycle. BUSREQ forces the CPU address bus, data bus, and control signals MREQ, IORQ, RD, and WR to go to a high-impedance state so that other devices can control these lines. BUSREQ is normally wired-OR and requires an external pullup for these applications. Extended BUSREQ periods due to extensive DMA operations can prevent the CPU from properly refreshing dynamic RAMs.

D₀-D₇. *Data Bus* (input/output, active High, 3-state). D₀-D₇ constitute an 8-bit bidirectional data bus, used for data exchanges with memory and I/O.

Halt. *Halt State* (output, active Low). HALT indicates that the CPU has executed a Halt instruction and is awaiting either a nonmaskable or a maskable interrupt (with the mask enabled) before operation can resume. While halted, the CPU executes NOPs to maintain memory refresh.

INT. *Interrupt Request* (input, active Low). Interrupt Request is generated by I/O devices. The CPU honors a request at the end of the current instruction if the internal software-controlled interrupt enable flip-flop (IFF) is enabled. INT is normally wired-OR and requires an external pullup for these applications.

IORQ. *Input/Output Request* (output, active Low, 3-state). IORQ indicates that the lower half of the address bus holds a valid I/O address for an I/O read or write operation. IORQ is also generated concurrently with M1 during an interrupt acknowledge cycle to indicate that an interrupt response vector can be placed on the data bus.

M1. *Machine Cycle One* (output, active Low). M1, together with MREQ, indicates that the current machine cycle is the opcode fetch cycle of an instruction execution. M1, together with IORQ, indicates an interrupt acknowledge cycle.

MREQ. *Memory Request* (output, active Low, 3-state). MREQ indicates that the address bus holds a valid address for a memory read or memory write operation.

NMI. *Non-Maskable Interrupt* (input, negative edge triggered). NMI has a higher priority than INT. NMI is always recognized at the end of the current instruction, independent of the status of the interrupt enable flip-flop, and automatically forces the CPU to restart at location 0066H.

RD. *Read* (output, active Low, 3-state). RD indicates that the CPU wants to read data from memory or an I/O device. The addressed I/O device or memory should use this signal to gate data onto the CPU data bus.

RESET. *Reset* (input, active Low). RESET initializes the CPU as follows: it resets the interrupt enable flip-flop, clears the PC and Registers I and R, and sets the interrupt status to Mode 0. During reset time, the address and data bus go to a high-impedance state, and all control output signals go to the inactive state. Note that RESET must be active for a minimum of three full clock cycles before the reset operation is complete.

RFSH. *Refresh* (output, active Low). RFSH, together with MREQ, indicates that the lower seven bits of the system's address bus can be used as a refresh address to the system's dynamic memories.

WAIT. *Wait* (input, active Low). WAIT indicates to the CPU that the addressed memory or I/O devices are not ready for a data transfer. The CPU continues to enter a Wait state as long as this signal is active. Extended WAIT periods can prevent the CPU from refreshing dynamic memory properly.

WR. *Write* (output, active Low, 3-state). WR indicates that the CPU data bus holds valid data to be stored at the addressed memory or I/O location.

Z80 CPU

AC CHARACTERISTICS†

Number	Symbol	Parameter	Z80 CPU		Z80A CPU		Z80B CPU		Z80H CPU	
			Min	Max	Min	Max	Min	Max	Min	Max
1	TcC	Clock Cycle Time	400*		250*		165*		125*	
2	TwCh	Clock Pulse Width (High)	180	2000	110	2000	65	2000	55	2000
3	TwCl	Clock Pulse Width (Low)	180	2000	110	2000	65	2000	55	2000
4	TfC	Clock Fall Time		30		30		20		10
5	TrC	Clock Rise Time		30		30		20		10
6	TdCr(A)	Clock ↑ to Address Valid Delay		145		110		90		80
7	TdA(MREQ)	Address Valid to $\overline{\text{MREQ}}$ ↓ Delay	125*		65*		35*		20*	
8	TdCl(MREQ)	Clock ↓ to $\overline{\text{MREQ}}$ ↓ Delay		100		85		70		60
9	TdCr(MREQ)	Clock ↑ to $\overline{\text{MREQ}}$ ↑ Delay		100		85		70		60
10	TwMREQh	$\overline{\text{MREQ}}$ Pulse Width (High)	170*		110*		65*		45*	
11	TwMREQl	$\overline{\text{MREQ}}$ Pulse Width (Low)	360*		220*		135*		100*	
12	TdCl(MREQ)	Clock ↓ to $\overline{\text{MREQ}}$ ↑ Delay		100		85		70		60
13	TdCl(RD)	Clock ↓ to $\overline{\text{RD}}$ ↓ Delay		130		95		80		70
14	TdCr(RD)	Clock ↑ to $\overline{\text{RD}}$ ↑ Delay		100		85		70		60
15	TsD(Cr)	Data Setup Time to Clock ↑	50		35		30		30	
16	ThD(RD)	Data Hold Time to $\overline{\text{RD}}$ ↑		0		0		0		0
17	TsWAIT(Cl)	$\overline{\text{WAIT}}$ Setup Time to Clock ↓	70		70		60		50	
18	ThWAIT(Cl)	$\overline{\text{WAIT}}$ Hold Time after Clock ↓		0		0		0		0
19	TdCr(M $\overline{\text{I}}$)	Clock ↑ to $\overline{\text{M}}$ ↑ ↓ Delay		130		100		80		70
20	TdCr(M $\overline{\text{I}}$)	Clock ↑ to $\overline{\text{M}}$ ↑ ↑ Delay		130		100		80		70
21	TdCr(RFSH)	Clock ↑ to $\overline{\text{RFSH}}$ ↓ Delay		180		130		110		95
22	TdCr(RFSHr)	Clock ↑ to $\overline{\text{RFSH}}$ ↑ Delay		150		120		100		85
23	TdCl(RD)	Clock ↓ to $\overline{\text{RD}}$ ↑ Delay		110		85		70		60
24	TdCr(RD)	Clock ↑ to $\overline{\text{RD}}$ ↓ Delay		100		85		70		60
25	TsD(Cl)	Data Setup to Clock ↓ during M ₂ , M ₃ , M ₄ , or M ₅ Cycles	60		50		40		30	
26	TdA(IORQ)	Address Stable prior to $\overline{\text{IORQ}}$ ↓	320*		180*		110*		75*	
27	TdCr(IORQ)	Clock ↑ to $\overline{\text{IORQ}}$ ↓ Delay		90		75		65		55
28	TdCl(IORQ)	Clock ↓ to $\overline{\text{IORQ}}$ ↑ Delay		110		85		70		60
29	TdD(WR)	Data Stable prior to $\overline{\text{WR}}$ ↓	190*		80*		25*		5*	
30	TdCl(WR)	Clock ↓ to $\overline{\text{WR}}$ ↓ Delay		90		80		70		60
31	TwWR	$\overline{\text{WR}}$ Pulse Width	360*		220*		135*		100*	
32	TdCl(WR)	Clock ↓ to $\overline{\text{WR}}$ ↑ Delay		100		80		70		60
33	TdD(WR)	Data Stable prior to $\overline{\text{WR}}$ ↑	20*		-10*		-55*		55*	
34	TdCr(WR)	Clock ↑ to $\overline{\text{WR}}$ ↓ Delay		80		65		60		55
35	TdWR(D)	Data Stable from $\overline{\text{WR}}$ ↑	120*		60*		30*		15*	
36	TdCl(HALT)	Clock ↓ to $\overline{\text{HALT}}$ ↑ or ↓		300		300		260		225
37	TwNMI	$\overline{\text{NMI}}$ Pulse Width	80		80		70		50*	
38	TsBUSREQ(Cr)	$\overline{\text{BUSREQ}}$ Setup Time to Clock ↑	80		50		50		40	

Z80 CPU

*For clock periods other than the minimums shown, calculate parameters using the table on the following page. Calculated values above assumed T_C = T_{IC} = 20 ns
†Units in nanoseconds (ns).

AC CHARACTERISTICS† (Continued)

Number	Symbol	Parameter	Z80 CPU		Z80A CPU		Z80B CPU		Z80H CPU	
			Min	Max	Min	Max	Min	Max	Min	Max
39	T _H BUSREQ(Cr)	BUSREQ Hold Time after Clock ↑	0		0		0		0	
40	T _d Cr(BUSACK↓)	Clock ↑ to BUSACK ↓ Delay		120		100		90		80
41	T _d Cl(BUSACK↑)	Clock ↓ to BUSACK ↑ Delay		110		100		90		80
42	T _d Cr(Dz)	Clock ↑ to Data Float Delay		90		90		80		70
43	T _d Cr(CTz)	Clock ↑ to Control Outputs Float Delay (\overline{MREQ} , \overline{IORQ} , \overline{RD} , and \overline{WR})		110		80		70		60
44	T _d Cr(Az)	Clock ↑ to Address Float Delay		110		90		80		70
45	T _d Cl(A)	\overline{MREQ} ↑, \overline{IORQ} ↑, \overline{RD} ↑, and \overline{WR} ↑ to Address Hold Time	160*		80*		35*		20*	
46	T _s RESET(Cr)	\overline{RESET} to Clock ↑ Setup Time	90		60		60		45	
47	T _H RESET(Cr)	\overline{RESET} to Clock ↑ Hold Time		0		0		0		0
48	T _s INT(Cr)	\overline{INT} to Clock ↑ Setup Time	80		80		70		55	
49	T _H INT(Cr)	\overline{INT} to Clock ↑ Hold Time		0		0		0		0
50	T _d Cl(\overline{IORQ})	$\overline{M1}$ ↑ to \overline{IORQ} ↓ Delay	920*		565*		365*		270*	
51	T _d Cl(\overline{IORQ} ↑)	Clock ↑ to \overline{IORQ} ↓ Delay		110		85		70		60
52	T _d Cl(\overline{IORQ} ↑)	Clock ↑ to \overline{IORQ} ↑ Delay		100		85		70		60
53	T _d Cl(D)	Clock ↑ to Data Valid Delay		230		150		130		115

*For clock periods other than the minimums shown, calculate parameters using the following table. Calculated values above

assumed $f_{IC} = f_{IC} - 20$ ns

†Units in nanoseconds (ns)

FOOTNOTES TO AC CHARACTERISTICS

Number	Symbol	General Parameter	Z80	Z80A	Z80B	Z80H
1	T _c C	$t_{wCh} + t_{wCl} + t_{IC} + t_{IC}$				
7	T _s AMBLEDQ	$t_{wCh} + t_{IC}$	- 75	- 65	- 50	- 45
10	T _s MREQh	$t_{wCh} + t_{IC}$	- 30	- 20	- 20	- 20
11	T _s MREQl	t_{wCl}	- 40	- 30	- 30	- 25
26	T _s AMBLEDQ	t_{IC}	- 80	- 70	- 55	- 45
29	T _s DWRh	t_{IC}	- 210	- 170	- 140	- 120
31	t_{wWP}	t_{IC}	- 40	- 30	- 30	- 25
33	T _s DWRh	$t_{wCl} + t_{IC}$	- 180	- 140	- 140	- 120
35	T _s INT	$t_{wCl} + t_{IC}$	- 80	- 70	- 55	- 45
45	T _s Cl(A)	$t_{wCl} + t_{IC}$	- 40	- 50	- 50	- 45
50	T _s M1(\overline{IORQ})	$2t_{IC} + t_{wCh} + t_{IC}$	- 80	- 65	- 50	- 45

AC Test Conditions

V_{DD} = 2.0V V_{DD} = 1.5V
V_{IL} = 0.8V V_{OH} = 1.5V
V_{DDC} = V_{CC} = 0.6V F_{LOAD} = 0.5V
V_{ILC} = 0.45V

ABSOLUTE MAXIMUM RATINGS

Voltages on all pins with respect to ground: 0.3V to +7V
 Operating Ambient Temperature: See Ordering Information
 Storage temperature: 65°C to +150°C

Stresses in excess of the Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at the condition above those indicated in the operating section of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

STANDARD TEST CONDITIONS

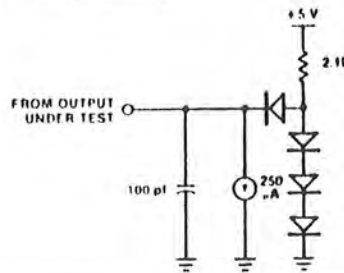
The DC Characteristics and Capacitance sections below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND (0V). Positive current flows into the referenced pin.

All ac parameters assume a load capacitance of 100 pF. Add 15 ns delay for each 50 pF increase in load up to a maximum of 200 pF for the data bus. AC timing measurements are referenced to 1.5 volts (except for clock, which is referenced to the 10% and 90% points).

Available operating temperature ranges are:

- S = 0°C to +70°C, +4.75V ≤ V_{CC} ≤ +5.25V
- E = -40°C to +85°C, +4.75V ≤ V_{CC} ≤ +5.25V
- M = -55°C to +125°C, +4.5V ≤ V_{CC} ≤ +5.25V

The Ordering Information section lists temperature ranges and product numbers. Package drawings are in the Package Information section in this book. Refer to the Literature List for additional documentation.



780 CPU

DC CHARACTERISTICS

All parameters are tested unless otherwise noted.

Symbol	Parameter	Min	Max	Unit	Test Condition
V _{ILC}	Clock Input Low Voltage	0.3	0.45	V	
V _{IHC}	Clock Input High Voltage	V _{CC} - 6	V _{CC} + 3	V	
V _{IL}	Input Low Voltage	0.3	0.8	V	
V _{IH}	Input High Voltage	2.0 ¹	V _{CC}	V	
V _{OL}	Output Low Voltage		0.4	V	I _{OL} = 2.0 mA
V _{OH}	Output High Voltage	2.4 ¹		V	I _{OH} = -250 μA
I _{CC}	Power Supply Current		100	mA	Note 3
I _{LI}	Input Leakage Current		10	μA	V _{IN} = 0 to V _{CC}
I _{LO}	3 State Output Leakage Current in Float	10	10 ⁻²	μA	V _{OUT} = 0.4 to V _{CC}

¹ For military grade parts refer to the 780 Military Electrical Specification
² A₁, A₂, D, D₁, (B, B₁), (B₂), (B₃), (B₄) and (B₅)
³ Maximum current is limited by the power supply.

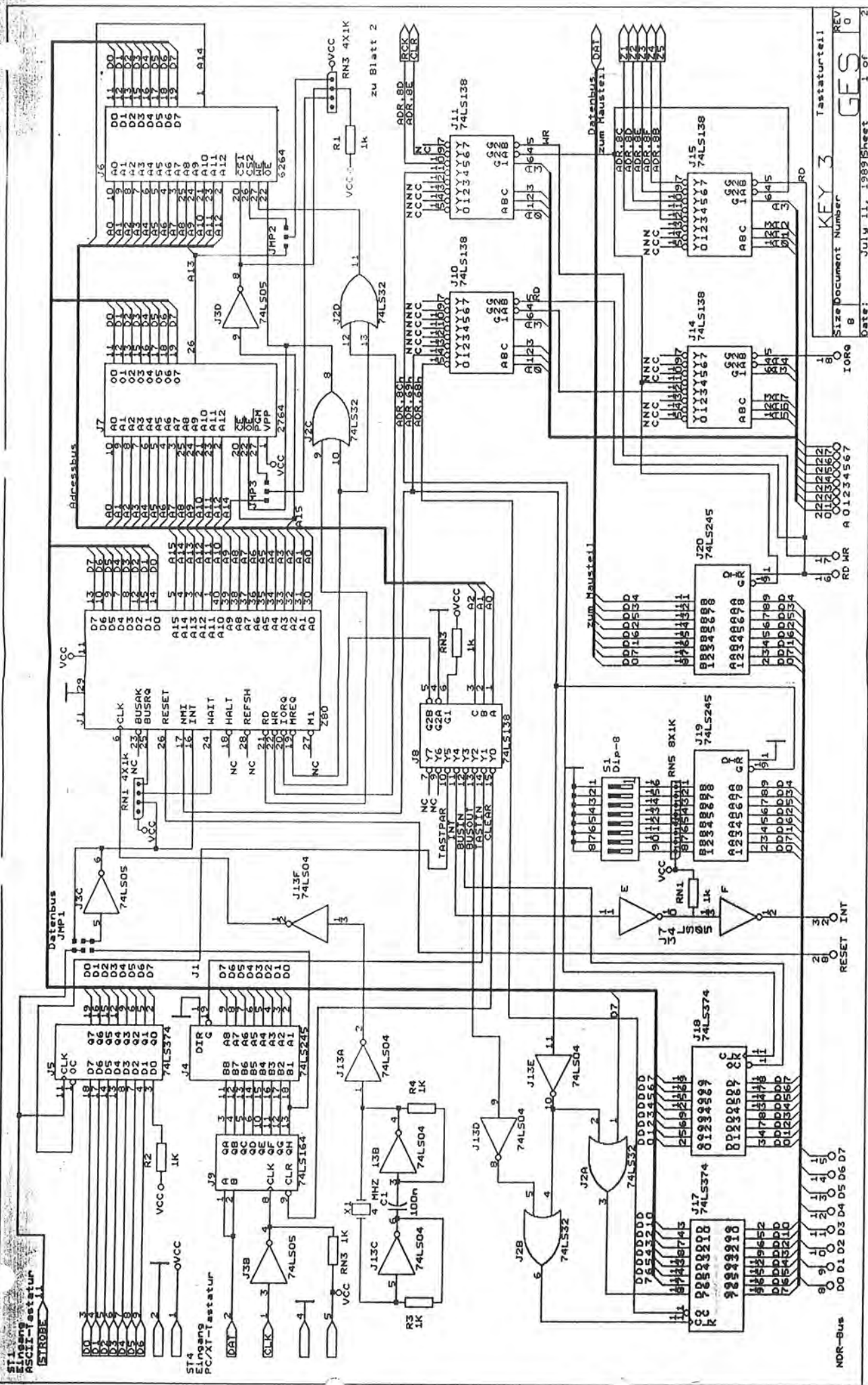
CAPACITANCE

Guaranteed by design and characterization

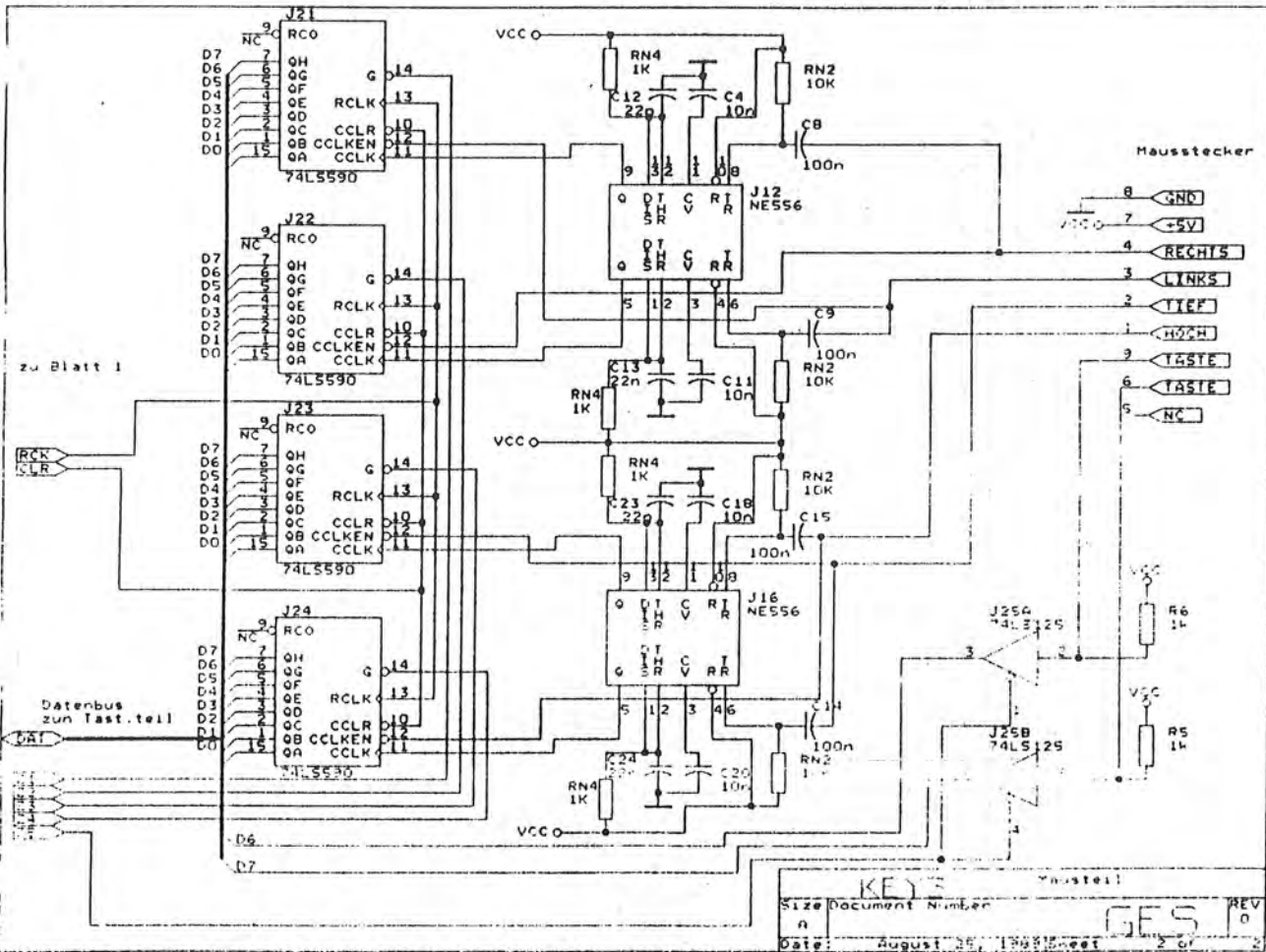
Symbol	Parameter	Min	Max	Unit
C _{clock}	Clock Capacitance		35	pF
C _{in}	Input Capacitance		5	pF
C _{out}	Output Capacitance		15	pF

NOTES

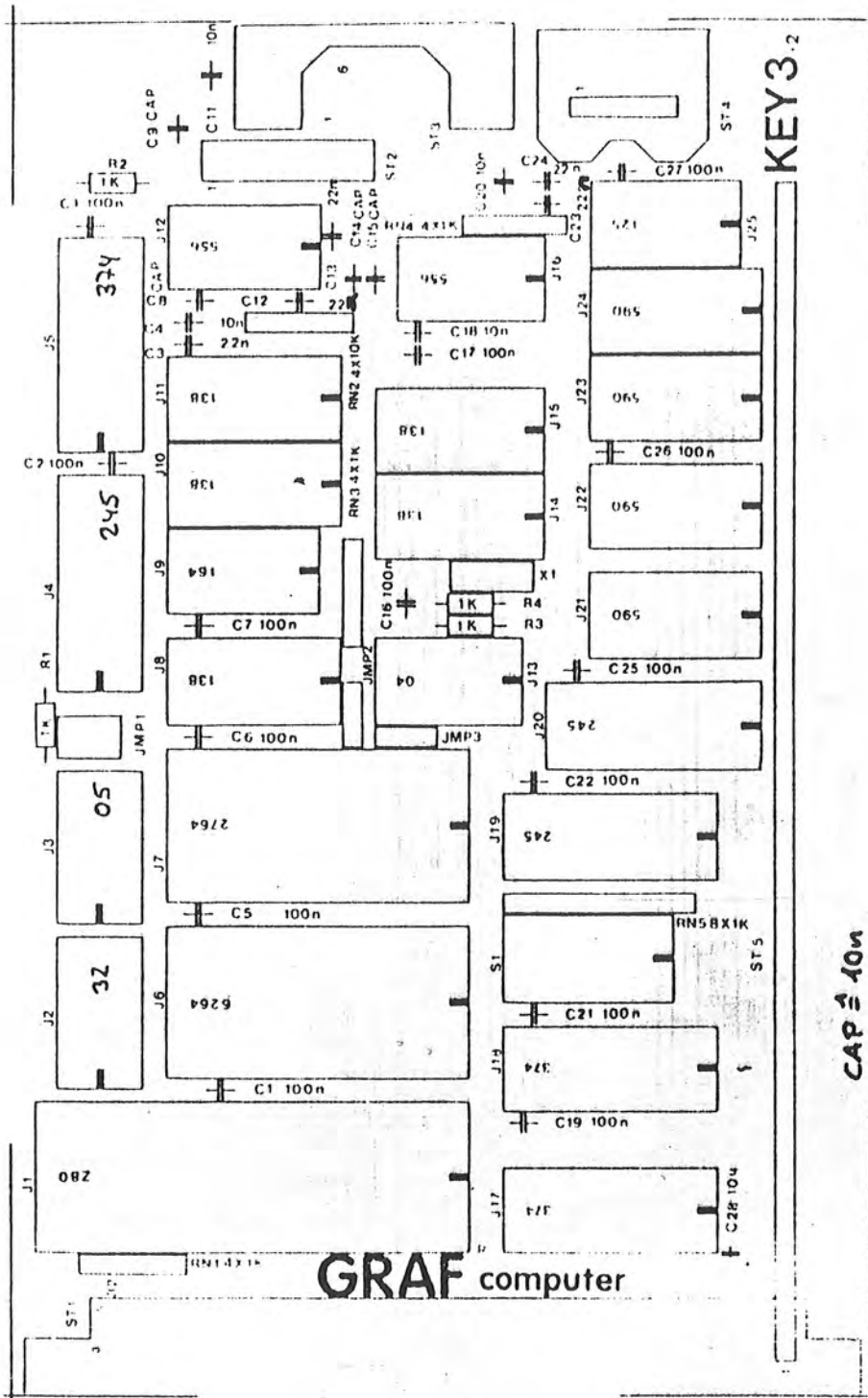
T_a = 25°C ± 1 MB
 Unmeasured pins returned to ground



ANHANG B: SCHALTPLAN (MAUSTEIL)

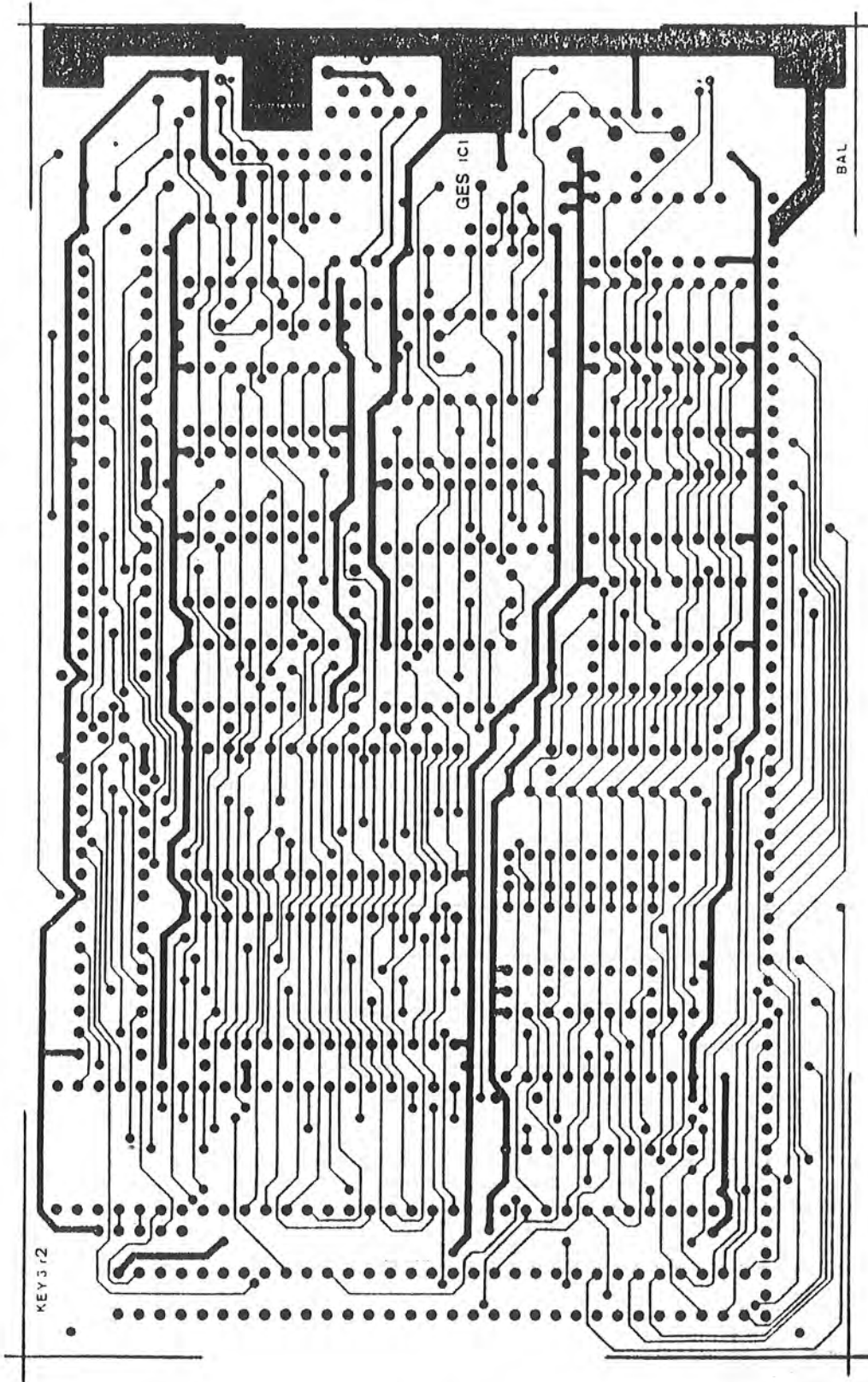


ANHANG C: BESTÜCKUNGSPLAN

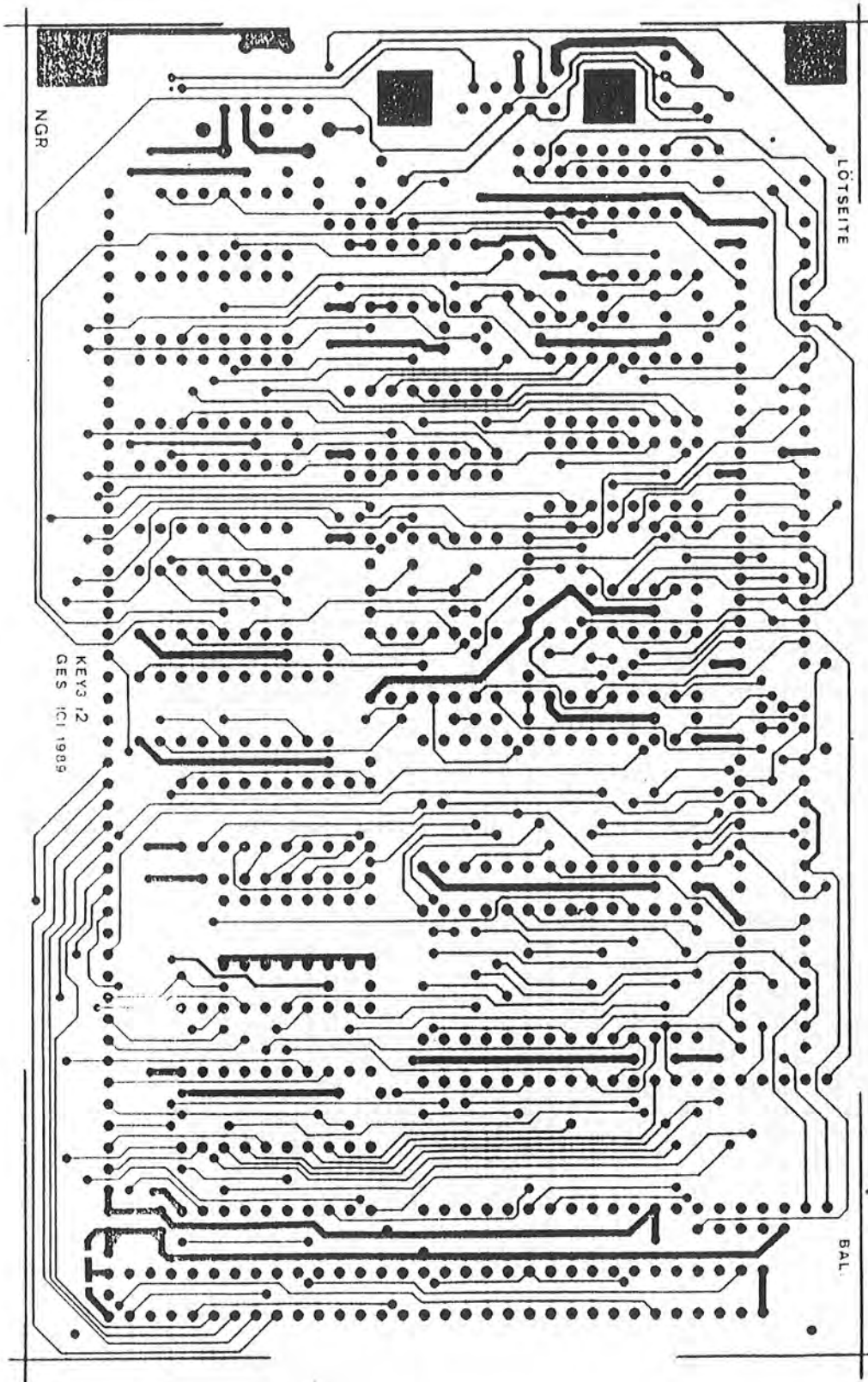


KEY3

ANHANG D: LAYOUT (BESTÜCKUNGSSEITE)



ANHANG E: LAYOUT LÖTSEITE



ANHANG F: LISTING DES KEY3-PROGRAMMS

```

*****
;*
;* Programm für Eprom auf der Key 3 Baugruppe *
;*
;*          Version 1.0          *
;*
;* (C) GRAF COMPUTER , Norbert Grotz *
;*
;*
;* Programmgeschichte:          *
;* Grundversion                8.12.88 *
;* Tastaturpuffer              14.12.88 *
;* Funktionstastenvorbele      15.12.88 *
;* Funktionsta. programmierbar 20.12.88 *
;* Funktionstasteneben (4)     21.12.88 *
;* Tastaturpuffer rücksetzen   27.12.88 *
;* Befehlswiederholung         28.12.88 *
;* Befehlswiederholung -7 Bef. 29.12.88 *
;* Kommunikation 1              1. 2.89 *
;* Pufferbehandlung neu        31. 3.89 *
;* INT-Pgm Verbesserung        5. 4.89 *
;* Tastbelegung neu            8. 4.89 *
;* Tastbelegung verbessert    10.4.89 *
;* Kommunikation 2             11.4.89 *
;* NMI-Pgm Verbesserung       3. 8.89 *
;* ASCII-Tastatur              4. 8.89 *
;* Verschiedene Verbesserungen 7. 8.89 *
*****

.z80

; Vereinbarungen

0000 port1 equ 0h ;Clear
0001 port2 equ 1h ;Tastein
0002 port3 equ 2h ;Zeiaus
0005 port5 equ 5h ;ASCIIPort

org 000

0000' F3 di
0001' C3 0200' jp haupt ;Sprung zum Hauptpgm

org 038h

0038' F3 di ;Interrupt sperren
0039' C3 006A' jp intv ;Sprung zum INT-pgm über Vektor

org 066h

0066' F3 di ;Interrupt sperren
0067' C3 006F' jp nmiv ;Sprung zum NMI-Pgm über Vektor

```

```

006A' E5          intv:  push hl
006B' 2A 0011"    ld hl,(ivektor) ;Interruptvektor holen
006E' E9          jp (hl)      ;anspringen

006F' E5          nmiv:  push hl
0070' 2A 0013"    ld hl,(nvektor) ;Nmivektor holen
0073' E9          jp (hl)      ;anspringen

0074' 43 6F 70 79  defm 'Copyright GRAF COMPUTER, Kempten'
007B' 72 69 67 68
007C' 74 20 47 52
0080' 41 46 20 43
0084' 4F 4D 50 55
0088' 54 45 52 2C
008C' 20 4B 65 6D
0090' 70 74 65 6E
0094' 50 67 6D 20  defm 'Pgm von Norbert Grotz, Kempten '
0098' 76 6F 6E 20
009C' 4E 6F 72 62
00A0' 65 72 74 20
00A4' 47 72 6F 74
00A8' 7A 2C 20 4B
00AC' 65 6D 70 74
00B0' 65 6E 20 20

org 200h

; Hauptprogramm -----

0200' 31 9FFF      haupt:  ld sp,9fffh    ;Stackpointer initialisieren
0203' ED 56        im 1      ;Interruptmode 1 einschalten

0205' 3E FF        ld a,0ffh    ;Port gleich
0207' D3 02        out (port3),a ;ungültig machen

0209' AF          xor a
020A' D3 00        out (port1),a ;Schieberegister loeschen u. sperren

                                ;Variablen vorbelegen
020C' 3E 10        ld a,10h    ;Voreinstellung des Offset
020E' 32 0000"     ld (off),a
0211' AF          xor a
0212' 32 0002"     ld (pufe),a ;Puffer Eingabezeiger
0215' 32 0003"     ld (pufa),a ;Puffer Ausgabezeiger
0218' 32 0004"     ld (pufz),a ;Zahl der Zeichen im Puffer
021B' 32 0008"     ld (pgm),a  ;Programmierererkennung

                                ;Verschiebung der Funktionstasten-
                                ;vorbelegung ins RAM
021E' 01 03FF     ld bc,03ffh ;Zahl der zu Übertragenden Bytes

```

```

0221' 11 8C00          ld de,8c00h      ;Zieladresse
0224' 21 0C00          ld hl,0c00h     ;Quelladresse
0227' ED B0           ldir            ;Übertrage alle Bytes
                                ;Ende der Blockverschiebung

0229' DB 05           in a,(port5)    ;Zeichen vom ASCII-Port holen
022B' E6 7F           and 127
022D' FE 20           cp ' '          ;wenn nicht Space, dann
022F' C2 0243'        jp nz,haupt1    ;gleich weiter
0232' 01 0400          ld bc,400h      ;Intpgm für ASCII bei 400h
0235' ED 43 0011''    ld (ivektor),bc ;400h in ivektor
0239' 01 0900          ld bc,900h      ;Nmipgm für ASCII bei 900h
023C' ED 43 0013''    ld (nvektor),bc ;900h in nvektor
0240' C3 0251'        jp haupt2
0243' 01 0500          haupt1: ld bc,0500h ;Intpgm für PC bei 500h
0246' ED 43 0011''    ld (ivektor),bc ;500h in ivektor
024A' 01 0A00          ld bc,0A00h     ;Nmipgm für pc bei A00h
024D' ED 43 0013''    ld (nvektor),bc ;A00h in nvektor

0251' FB             haupt2: ei      ;Interrupt freigeben

                                ; hier kommt später das eigentliche Hauptpgm hin

0252' 76             intwart:halt
0253' C3 0252'        jp intwart

                                ; Ende Hauptprogramm -----

org 400h

                                ;INTASCIIprogramm-----

0400' F5             inta:  push af      ;Register AF retten
0401' DB 05           in a,(port5)    ;Zeichen vom Port holen
0403' E6 7F           and 127         ;Bit 7 auf null setzen
0405' D3 02           out (port3),a   ;Zeichen an Bus ausgeben
0407' F1             pop af
0408' E1             pop hl         ;Register wieder zurück
0409' FB             ei          ;Intrrupt wieder freigeben
040A' ED 4D           reti

                                ; Ende INTASCIIprogramm-----

org 500h

                                ; INTPCprogramm -----

0500' F5             int:   push af      ;Register af retten
0501' 3E 30           ld a,030h      ;Warteschleife
0503' D6 01           warte: sub 1     ;bis Zeichen ganz da
0505' 00             nop            ;Schleife notwendig da erst
0506' 20 FB           jr nz,warte    ;8 Bit des Zeichens da
0508' DB 01           in a,(port2)   ;Port einlesen

```



```

050A' 6F          ld l,a          ;Tastencode in REG L
050B' AF          xor a
050C' D3 00       out (port1),a    ;Schieberegister löschen
050E' 3A 0000''   ld a,(off)
0511' 67          ld h,a          ;Offset in REG H
0512' 7E          ld a,(hl)       ;Zeichen aus Tabelle holen
0513' 32 0001''   ld (zei),a      ;Zeichen sichern
0516' 3A 0000''   ld a,(off)      ;Offset holen
0519' E6 04       and 4
051B' FE 04       cp 4            ; ALT ???
051D' C2 052A'    jp nz,int1      ;wenn nicht weiter
0520' 3A 0001''   ld a,(zei)      ;Zeichen holen
0523' E6 30       and 30h
0525' FE 30       cp 30h         ;wenn Zahl (0-9) dann zu
0527' CA 06C1'    jp z,wied       ;Befhls wiederholung
052A' 3A 0001''   int1: ld a,(zei)
052D' E6 80       and 128        ; ASCII ???
052F' CA 05E0'    jp z,pufpgm     ;dann sofort zum Pufferpgm
0532' 3A 0001''   ld a,(zei)
0535' FE 8F       cp 8fh         ; ASCII (losgelassen) ???
0537' CA 0748'    jp z,intend     ;dann zum Ende
053A' E6 90       and 144        ; Funktionstaste ???
053C' FE 90       cp 144         ; (Bit 4 u. 7 gesetzt)
053E' CA 062C'    jp z,funkt      ;dann zu Funktionstastenpgm
0541' 3A 0001''   ld a,(zei)
0544' E6 A0       and 160        ; Funktionstaste (losgelassen) ???
0546' FE A0       cp 160         ; (Bit 5 u.7 gesetzt)
0548' CA 06A7'    jp z,funklos    ;dann zu Funktionst.losgelassen

; Sondertastenteil -----
054B' 3A 0001''   ld a,(zei)
054E' FE 80       sod1: cp 80h    ; Control (gedrückt) ???
0550' C2 055E'    jp nz,sod2
0553' 3A 0000''   ld a,(off)      ;Offset holen
0556' F6 01       or 1            ;um 1 erhöhen
0558' 32 0000''   ld (off),a      ;Offset wieder sichern
055B' C3 0748'    jp intend
055E' FE 81       sod2: cp 81h    ; Control (losgelassen) ???
0560' C2 056E'    jp nz,sod3      ;wenn nicht dann zum nächsten
0563' 3A 0000''   ld a,(off)
0566' E6 FE       and 254        ;Offset um 1 verkleinern
0568' 32 0000''   ld (off),a
056B' C3 0748'    jp intend
056E' FE 82       sod3: cp 82h    ; Shift ???
0570' C2 057E'    jp nz,sod4      ;wenn nicht weiter
0573' 3A 0000''   ld a,(off)
0576' F6 02       or 2            ;um 2 erhöhen
0578' 32 0000''   ld (off),a
057B' C3 0748'    jp intend
057E' FE 83       sod4: cp 83h    ; Shift (losge.) ???
0580' C2 058E'    jp nz,sod5      ;wenn nicht weiter
0583' 3A 0000''   ld a,(off)
0586' E6 FD       and 253        ;um 2 verkleinern
0588' 32 0000''   ld (off),a
058B' C3 0748'    jp intend
058E' FE 84       sod5: cp 84h    ; Alt ???

```

```

0590' C2 059E'      jp nz,sod6        ;wen nicht weiter
0593' 3A 0000"     ld a,(off)
0596' F6 04        or 4              ;um 4 erhöhen
0598' 32 0000"     ld (off),a
059B' C3 0748'     jp intend
059E' FE 85        sod6: cp 85h      ; Alt (losge.) ???
05A0' C2 05AE'     jp nz,sod7       ;wenn nicht weiter
05A3' 3A 0000"     ld a,(off)
05A6' E6 FB        and 251         ;um 4 verkleinern
05A8' 32 0000"     ld (off),a
05AB' C3 0748'     jp intend
05AE' FE 86        sod7: cp 86h      ; Capslock ???
05B0' C2 05BE'     jp nz,sod8       ;wenn nicht weiter
05B3' 3A 0000"     ld a,(off)
05B6' EE 02        xor 2           ;ein- od. ausschalten
05B8' 32 0000"     ld (off),a
05BB' C3 0748'     jp intend
05BE' FE 87        sod8: cp 87h      ; Numlock ???
05C0' C2 05CE'     jp nz,sod9       ;wenn nicht weiter
05C3' 3A 0000"     ld a,(off)
05C6' EE 08        xor 8           ;ein- od. ausschalten
05C8' 32 0000"     ld (off),a
05CB' C3 0748'     jp intend
05CE' FE 88        sod9: cp 88h      ; Pufferreset ???
05D0' C2 0748'     jp nz,intend     ;wenn nicht weiter
05D3' AF          xor a
05D4' 32 0004"     ld (pufz),a      ;Pufferzahl = 0
05D7' 32 0003"     ld (pufa),a      ;Pufferauszeiger = 0
05DA' 32 0002"     ld (pufe),a      ;Puffereinzeiger = 0
05DD' C3 0748'     jp intend

; Ende Sondertastenteil -----

; Pufferpgm -----

05E0' CD 05E6'     pufpgm: call pufupgm ;Pufferunterprogramm
05E3' C3 0748'     jp intend        ;aufrufen

; Pufferpgm Ende -----

;Pufferunterprogramm -----

05E6' AF          pufupgm:xor a
05E7' 32 0010"     ld (intstop),a   ;Intstop auf null setzen
05EA' 3A 0002"     ld a,(pufe)
05ED' 6F          ld l,a           ;Einzeiger in L
05EE' 3C          inc a           ;und gleich um 1 erhöhen
05EF' 32 0002"     ld (pufe),a
05F2' 26 81       ld h,81h        ;Puffer höherwertig in H
05F4' 3A 0001"     ld a,(zei)
05F7' 77          ld (hl),a       ;Zeichen in Puffer
05FB' 3A 0004"     ld a,(pufz)
05FB' 3C          inc a           ;Pufferzahl um 1 erhöhen
05FC' 32 0004"     ld (pufz),a

```

```

05FF' E6 FE          and 1111110b
0601' C2 0618'      jp nz,pufuend      ;wenn Zahl grösser 1 zum Ende
0604' 32 0010"     ld (intstop),a
0607' E6 FF          and 255
0609' C2 0618'      jp nz,pufuend      ;wenn intstop <> 0 zum Ende

060C' 3A 0001"     ld a,(zei)
060F' D3 02          out (port3),a      ;Zeichen an Port
0611' 3A 0003"     ld a,(pufa)
0614' 3C             inc a              ;Pufferauszeiger um 1 erhöhen
0615' 32 0003"     ld (pufa),a

0618' 3A 0008"     pufuend:ld a,(pgm) ; Beim Programmieren ???
0618' FE 01         cp 1
061D' C2 062B'      jp nz,pufuret      ; wenn nicht, dann zum Schluss
0620' 3A 0001"     ld a,(zei)         ;sonst
0623' 2A 000A"     ld hl,(pgma)       ;Programmierziger in HL
0626' 77            ld (hl),a           ;Zeichen in Tabelle abspeichern
0627' 23            inc hl              ;Zeiger um 1 erhöhen
0628' 22 000A"     ld (pgma),hl      ;und auch wieder abspeichern
0628' C9            pufuret:ret          ;zurück

                                ;Pufferunterprogramm Ende

                                ; Funktionstastenpgm -----
062C' 3A 0008"     funkt: ld a,(pgm) ; beim Programmieren ???
062F' FE 01         cp 1
0631' C2 0642'      jp nz,funkt1      ;dann alle Tasten möglich
0634' 3A 0009"     ld a,(funpr)      ;zu programmierende Taste holen
0637' 47            ld b,a
0638' 3A 0001"     ld a,(zei)         ;Zeichen holen
0638' 88            cp b
063C' CA 06A4'      jp z,funkend       ;wenn gleich, dann zum Ende
063F' C3 0672'      jp funkt2
0642' 3A 0000"     funkt1: ld a,(off) ;Offset holen
0645' E6 04         and 4
0647' FE 04         cp 4
0649' C2 0672'      jp nz,funkt2      ;wenn nicht, dann zum Funkt.auslesen
064C' 3E 01         ld a,1
064E' 32 0008"     ld (pgm),a        ;sonst Pgmerkennung auf 1 setzten
0651' 3A 0001"     ld a,(zei)
0654' 32 0009"     ld (funpr),a      ;Funktionstaste merken
0657' E6 0F         and 15
0659' CB 27         sla a
065B' CB 27         sla a
065D' CB 27         sla a
065F' CB 27         sla a
0661' 6F            ld l,a            ;Zeiger in L laden
0662' 3A 0000"     ld a,(off)        ;Offset laden
0665' E6 03         and 3
0667' 47            ld b,a            ;in B
0669' 3E 8C         ld a,8Ch          ;Grundadresse holen
066B' 80            add a,b          ;Offset + Grundadresse = Zeiger
066E'              ld h,a            ;Zeiger höherwertig in H laden
066C' 2A 000A"     ld (pgma),hl      ;Programmieranfangzeiger sichern
066F' 47            jp intend         ;und zum Schluss

```

```

0672' 3A 0001"      funkt2: ld a,(zei)
0675' E6 0F                and 15                ;für Tabelle anpassen
0677' CB 27                sla a                ;(Low nibble wird zu High nibble)
0679' CB 27                sla a
067B' CB 27                sla a
067D' CB 27                sla a
067F' 6F                ld l,a              ;und in L laden
0680' 32 0006"            ld (funz),a         ;und Zeiger sichern
0683' 3A 0000"      fein:  ld a,(off)   ;Offset laden
0686' E6 03                and 3               ;oberen Teil abschneiden
0688' 47                ld b,a              ;in B
0689' 3E 8C                ld a,8Ch            ;Grundadresse holen
068B' 80                add a,b             ;Offset + Grundadresse = Zeiger
068C' 67                ld h,a              ;höherwertigen Teil in H
068D' 7E                ld a,(hl)           ;Zeichen aus Tabelle holen
068E' 32 0001"            ld (zei),a         ;und sichern
0691' FE FF                cp 0ffh             ;wenn Zeichen = FF
0693' CA 06A4'            jp z,funkend        ;dann fertig mit Funktionstasten
0696' CD 05E6'            call pufupgm        ;Zeichen in Puffer schreiben
0699' 3A 0006"            ld a,(funz)
069C' 3C                inc a                ;Zeiger um 1 erhöhen
069D' 32 0006"            ld (funz),a
06A0' 6F                ld l,a
06A1' C3 0683'            jp fein              ;naechstes Zeichen holen

```

```

06A4' C3 0748'      funkend:jp intend

```

; Funktionstastenpgm Ende -----

;Funktionstaste losgelassen -----

```

06A7' 3A 0009"      funklos:ld a,(funpr) ;gerade programmierte ftaste holen
06AA' 47                ld b,a
06AB' 3A 0001"            ld a,(zei)         ;Zeichen holen
06AE' D6 10                sub 16              ;notwendig da losgelassen
06B0' B8                cp b                ;vergleichen
06B1' C2 0748'            jp nz,intend        ;wenn nicht gleich zum Ende
06B4' AF                xor a
06B5' 32 0008"            ld (pgm),a         ;sonst Pgmerkennung löschen
06BB' 3E FF                ld a,0ffh          ;und Endemarkierung
06BA' 2A 000A"            ld hl,(pgma)       ;ans Ende der programmierten
06BD' 77                ld (hl),a          ;Funktionstaste setzen
06BE' C3 0748'            jp intend          ;und zum Ende

```

;Funktlosgelassen Ende -----

; Befehlswiederholung -----

```

06C1' 3A 0002"      wied:  ld a,(pufe)   ;aktuellen Puffereinzeiger holen
06C4' 3D                dec a                ; -1
06C5' 32 000E"            ld (wieda),a       ;= Wiederholungsanfang
06C8' 6F                ld l,a
06C9' AF                xor a
06CA' 32 000C"            ld (wiedz),a       ;Zähler löschen

```

06CD'	3A 0001"	ld a,(zei)	;Zeichen holen
06D0'	E6 07	and 07h	;in Ziffer (0-7) umsetzen
06D2'	32 000D"	ld (wiedh),a	;und sichern
06D5'	3E 81	ld a,81h	;Pufzeiger höherwertig
06D7'	67	ld h,a	;im H laden
06D8'	7E	wied1: ld a,(hl)	;Zeichen aus Puffer holen -Suchschleife
06D9'	32 0001"	ld (zei),a	;sichern
06DC'	3A 000E"	ld a,(wieda)	
06DF'	3D	dec a	;Wiederholungsanfang - 1
06E0'	32 000E"	ld (wieda),a	
06E3'	6F	ld l,a	;und in L übernehmen
06E4'	3A 000C"	ld a,(wiedz)	
06E7'	3C	inc a	
06E8'	32 000C"	ld (wiedz),a	
06EB'	FE FF	cp 0ffh	;wenn schon 255 dann Puffer durch
06ED'	CA 0748'	jp z,intend	;und deshalb zum Ende
06F0'	3A 0001"	ld a,(zei)	
06F3'	FE 0D	cp 0dh	;wenn CR
06F5'	CA 06FB'	jp z,wgef	;dann zu gefunden
06F8'	C3 06D8'	jp wied1	;sonst weitersuchen
06FB'	3A 000D"	wgef: ld a,(wiedh)	;Hilfsvariable holen (enthält Ziffer)
06FE'	FE 01	cp 1	;wenn 1 dann
0700'	CA 070F'	jp z,whed	;zum Endemerken
0703'	FE 00	cp 0	;wenn 0 dann
0705'	CA 0720'	jp z,whan	;zum Anfangmerken
0708'	3D	dec a	; - 1
0709'	32 000D"	ld (wiedh),a	;wieder sichern
070C'	C3 06D8'	jp wied1	;sonst weitersuchen
070F'	3A 000E"	whed: ld a,(wieda)	;momentaner Zeiger
0712'	3C	inc a	; + 1
0713'	32 000F"	ld (wiede),a	; = Wiederholungsende
0716'	3A 000D"	ld a,(wiedh)	;Hilfsvar holen
0719'	3D	dec a	; -1
071A'	32 000D"	ld (wiedh),a	;wieder sichern
071D'	C3 06D8'	jp wied1	;jetzt weitersuchen
0720'	3A 000E"	whan: ld a,(wieda)	;Wiederholungsende holen
0723'	3C	inc a	; + 2
0724'	3C	inc a	
0725'	32 000E"	ld (wieda),a	;sichern
0728'	6F	wied2: ld l,a	;Zeiger in L -Ausgabeschleife
0729'	3E 81	ld a,81h	;Zeiger höherwertig in
072B'	67	ld h,a	; H
072C'	7E	ld a,(hl)	;Zeichen holen
072D'	32 0001"	ld (zei),a	;und sichern
0730'	CD 05E6'	call pufupgm	;und ausgeben
0733'	3A 000E"	ld a,(wieda)	
0736'	3C	inc a	; +1
0737'	32 000E"	ld (wieda),a	
073A'	47	ld b,a	
073B'	3A 000F"	ld a,(wiede)	;wenn Anfang = Ende
073E'	BB	cp b	
073F'	CA 0748'	jp z,intend	;dann zum Ende
0742'	3A 000E"	ld a,(wieda)	
0745'	C3 0728'	jp wied2	;sonst weiter ausgeben

; Ende Befehlswiederholung -----

```

0748' F1          intend: pop af          ;Register AF zurück
0749' E1          pop hl                ;Register HL zurück
074A' FB          ei                    ;Interrupt freigeben

074B' ED 4D       reti                  ;zurück zum Hauptprogramm

```

; Ende Interruptprogramm -----

org 900h

; NMIASCIprogramm -----

```

0900' F5          nmia: push af
0901' 3E FF       ld a,0ffh          ;Zeichen am Port ungültig
0903' D3 02       out (port3),a      ;machen
0905' F1          pop af
0906' E1          pop hl
0907' FB          ei
0908' ED 45       retn                ;zurück

```

; Ende NMIASCIprogramm -----

org 0A00h

; NMIProgramm -----

```

0A00' F5          nmi:   push af          ;Register AF retten

0A01' 3E 80       ld a,80h          ;damit Daten ungültig
0A03' D3 02       out (port3),a      ;Bit 7 am Port setzen

0A05' 3A 0004"    ld a,(pufz)
0A08' E6 FF       and 255          ;wenn Puffzahl = 0
0A0A' CA 0A26'    jp z,nmiend     ;dann gleich zurück
0A0D' 3D          dec a            ;Pufferzahl um 1 verringern
0A0E' 32 0004"    ld (pufz),a
0A11' CA 0A26'    jp z,nmiue     ;wenn Puffer leer nichts ausgeben
0A14' 3A 0003"    ld a,(pufa)
0A17' 6F          ld l,a          ;Puffauszeiger in L
0A18' 3C          inc a
0A19' 32 0003"    ld (pufa),a    ;gleich noch erhöhen
0A1C' 26 81       ld h,81h        ;Puffhöherwertig in H
0A1E' 7E          ld a,(hl)       ;Zeichen aus Puffer in a
0A1F' D3 02       out (port3),a    ;und ausgeben
0A21' 3E 01       ld a,1
0A23' 32 0010"    ld (intstop),a ;Intstop einschalten
0A26'          nmiue:
0A26' F1          nmiend: pop af    ;Register zurückholen
0A27' E1          pop hl
0A28' FB          ei

```


0A29' ED 45

retn

; zurück zum Hauptprogramm

; Ende NMIprogramm -----

;Funktionstastentabelle

org 0C00h

;Grundebene

0C00'	64 69 72	defm 'dir'	;F1 = 'dir'
0C03'	FF FF	defb 0ffh,0ffh	;Enderkennung
		org 0C10h	
0C10'	70 69 70	defm 'pip'	;F2 = 'pip'
0C13'	FF FF	defb 0ffh,0ffh	
		org 0C20h	
0C20'	65 72 61	defm 'era'	;F3 = 'era'
0C23'	FF FF	defb 0ffh,0ffh	
		org 0C30h	
0C30'	72 65 6E	defm 'ren'	;F4 = 'ren'
0C33'	FF FF	defb 0ffh,0ffh	
		org 0C40h	
0C40'	73 74 61 74	defm 'stat'	;F5 = 'stat'
0C44'	FF FF	defb 0ffh,0ffh	
		org 0C50h	
0C50'	74 79 70 65	defm 'type'	;F6 = 'type'
0C54'	FF FF	defb 0ffh,0ffh	
		org 0C60h	
0C60'	61 3A	defm 'a:'	;F7 = 'a:'
0C62'	FF FF	defb 0ffh,0ffh	
		org 0C70h	
0C70'	62 3A	defm 'b:'	;F8 = 'b:'
0C72'	FF FF	defb 0ffh,0ffh	
		org 0C80h	
0C80'	2A	defm '*'	;F9 = '*'
0C81'	FF FF	defb 0ffh,0ffh	
		org 0C90h	
0C90'	2A 2E	defm '*.'	;F10 = '*.'
0C92'	FF FF	defb 0ffh,0ffh	

org 0D00h

;Controlebene

0D00'	77 73	defm 'ws'	;F1 = 'ws'
0D02'	0D FF	defb 00dh,0ffh	;Return + Enderkennung
		org 0D10h	
0D10'	6D 38 30	defm 'm80'	;F2 = 'm80'
0D13'	0D FF	defb 00dh,0ffh	
		org 0D20h	
0D20'	6C 33 34	defm 'l34'	;F3 = 'l34'
0D23'	0D FF	defb 00dh,0ffh	
		org 0D30h	
0D30'	2F 70 3A 30	defm '/p:0000/d:8000'	;F4 = '/p:0000/d:8000'
0D34'	30 30 30 2F		
0D38'	64 3A 38 30		
0D3C'	30 30		

```

0D3E' 0D FF          defb 00dh,0ffh
                                org 0D40h
0D40' 73 74 61 74    defm 'stat'                ;F5 = 'stat'
0D44' FF FF          defb 0ffh,0ffh
                                org 0D50h
0D50' 74 79 70 65    defm 'type'                ;F6 = 'type'
0D54' FF FF          defb 0ffh,0ffh
                                org 0D60h
0D60' 61 3A          defm 'a:'                  ;F7 = 'a:'
0D62' FF FF          defb 0ffh,0ffh
                                org 0D70h
0D70' 62 3A          defm 'b:'                  ;F8 = 'b:'
0D72' FF FF          defb 0ffh,0ffh
                                org 0D80h
0D80' 2A             defm '*'                  ;F9 = '*'
0D81' FF FF          defb 0ffh,0ffh
                                org 0D90h
0D90' 2A 2E          defm '*.'                ;F10 = '*.'
0D92' FF FF          defb 0ffh,0ffh

                                org 0E00h                ;Shiftebene

0E00' 64 69 72        defm 'dir'                ;F1 = 'dir'
0E03' FF FF          defb 0ffh,0ffh                ;Enderkennung
                                org 0E10h
0E10' 70 69 70        defm 'pip'                ;F2 = 'pip'
0E13' FF FF          defb 0ffh,0ffh
                                org 0E20h
0E20' 65 72 61        defm 'era'                ;F3 = 'era'
0E23' FF FF          defb 0ffh,0ffh
                                org 0E30h
0E30' 72 65 6E        defm 'ren'                ;F4 = 'ren'
0E33' FF FF          defb 0ffh,0ffh

                                org 0E40h
0E40' 73 74 61 74    defm 'stat'                ;F5 = 'stat'
0E44' FF FF          defb 0ffh,0ffh
                                org 0E50h
0E50' 74 79 70 65    defm 'type'                ;F6 = 'type'
0E54' FF FF          defb 0ffh,0ffh
                                org 0E60h
0E60' 61 3A          defm 'a:'                  ;F7 = 'a:'
0E62' FF FF          defb 0ffh,0ffh
                                org 0E70h
0E70' 62 3A          defm 'b:'                  ;F8 = 'b:'
0E72' FF FF          defb 0ffh,0ffh
                                org 0E80h
0E80' 2A             defm '*'                  ;F9 = '*'
0E81' FF FF          defb 0ffh,0ffh
                                org 0E90h
0E90' 2A 2E          defm '*.'                ;F10 = '*.'
0E92' FF FF          defb 0ffh,0ffh

                                org 0F00h                ;Control- und Shiftebene

0F00' 64 69 72        defm 'dir'                ;F1 = 'dir'

```

```

0F03'  FF FF          defb 0ffh,0ffh          ;Enderkennung
                                org 0F10h
0F10'  70 69 70      defm 'pip'              ;F2 = 'pip'
0F13'  FF FF          defb 0ffh,0ffh
                                org 0F20h
0F20'  65 72 61      defm 'era'              ;F3 = 'era'
0F23'  FF FF          defb 0ffh,0ffh
                                org 0F30h
0F30'  72 65 6E      defm 'ren'              ;F4 = 'ren'
0F33'  FF FF          defb 0ffh,0ffh
                                org 0F40h
0F40'  73 74 61 74   defm 'stat'           ;F5 = 'stat'
0F44'  FF FF          defb 0ffh,0ffh
                                org 0F50h
0F50'  74 79 70 65   defm 'type'          ;F6 = 'type'
0F54'  FF FF          defb 0ffh,0ffh
                                org 0F60h
0F60'  61 3A         defm 'a:'            ;F7 = 'a:'
0F62'  FF FF          defb 0ffh,0ffh
                                org 0F70h
0F70'  62 3A         defm 'b:'            ;F8 = 'b:'
0F72'  FF FF          defb 0ffh,0ffh
                                org 0F80h
0F80'  2A            defm '*'             ;F9 = '*'
0F81'  FF FF          defb 0ffh,0ffh
                                org 0F90h
0F90'  2A 2E         defm '*.'           ;F10 = '*.'
0F92'  FF FF          defb 0ffh,0ffh

```

;Datentabelle (wird durch Linker ab Adresse 1000 gelegt)

org 1000h ;Grundebene (keine Sondertaste gedrückt)

```

; x0 x1 x2 x3 x4 x5 x6 x7      Adresse niederwertig (0-7)
; x8 x9 xA xB xC xD xE xF      Adresse niederwertig (8-F)
;                               x Adresse höherwertig (nach unten)

```

```

..... 8F 1B 31 32      db 8Fh,1Bh,31h,32h,33h,34h,35h,36h ;0  NB ESC 1 2 3 4 5 6 7
1004'  33 34 35 36
1008'  37 38 39 30      db 37h,38h,39h,30h,7Eh,27h,08h,09h ;0  7 8 9 0 B ` BS TAB
100C'  7E 27 08 09
1010'  71 77 65 72      db 71h,77h,65h,72h,74h,7Ah,75h,69h ;1  q w e r t z u i
1014'  74 7A 75 69
1018'  6F 70 7D 2B      db 6Fh,70h,7Dh,2Bh,0Dh,80h,61h,73h ;1  o p ü + CR CTRL a s
101C'  0D 80 61 73
1020'  64 66 67 68      db 64h,66h,67h,68h,6Ah,6Bh,6Ch,7Ch ;2  d f g h j k l ö
1024'  6A 6B 6C 7C
1028'  7B 23 82 3C      db 7Bh,23h,82h,3Ch,79h,78h,63h,76h ;2  ä # SHIFT < y x c v
102C'  79 78 63 76
1030'  62 6E 6D 2C      db 62h,6Eh,6Dh,2Ch,2Eh,2Dh,82h,2Ah ;3  b n m , . - SHIFT *
1034'  2E 2D 82 2A
1038'  84 20 86 90      db 84h,20h,86h,90h,91h,92h,93h,94h ;3  ALT " " CAPSL. F1 F2 F3 F4 F5
103C'  91 92 93 94
1040'  95 96 97 98      db 95h,96h,97h,98h,99h,87h,88h,37h ;4  F6 F7 F8 F9 F10 NUML. NB 7

```

1044'	99 87 88 37		
1048'	38 39 2D 34	db 38h,39h,2Dh,34h,35h,36h,28h,31h	;4 8 9 - 4 5 6 + 1
104C'	35 36 2B 31		
1050'	32 33 30 2E	db 32h,33h,30h,2Eh,8Fh,8Fh,8Fh,8Fh	;5 2 3 0 . NB NB NB NB
1054'	8F 8F 8F 8F		
1058'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;5 NB NB NB NB NB NB NB NB
105C'	8F 8F 8F 8F		
1060'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;6 NB NB NB NB NB NB NB NB
1064'	8F 8F 8F 8F		
1068'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;6 NB NB NB NB NB NB NB NB
106C'	8F 8F 8F 8F		
1070'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;7 NB NB NB NB NB NB NB NB
1074'	8F 8F 8F 8F		
1078'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;7 NB NB NB NB NB NB NB NB
107C'	8F 8F 8F 8F		
1080'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;8 NB NB NB NB NB NB NB NB
1084'	8F 8F 8F 8F		
1088'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;8 NB NB NB NB NB NB NB NB
108C'	8F 8F 8F 8F		
1090'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;9 NB NB NB NB NB NB NB NB
1094'	8F 8F 8F 8F		
1098'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,81h,8Fh,8Fh	;9 NB NB NB NB NB CTRL E NB NB
109C'	8F 81 8F 8F		
10A0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;A NB NB NB NB NB NB NB NB
10A4'	8F 8F 8F 8F		
10A8'	8F 8F 83 8F	db 8Fh,8Fh,83h,8Fh,8Fh,8Fh,8Fh,8Fh	;A NB NB SHIFTE NB NB NB NB NB
10AC'	8F 8F 8F 8F		
10B0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,83h,8Fh	;B NB NB NB NB NB NB SHIFTE NB
10B4'	8F 8F 83 8F		
10B8'	85 8F 8F A0	db 85h,8Fh,8Fh,0A0h,0A1h,0A2h,0A3h,0A4h	;B ALTE NB NB F1 F2 F3 F4 F5
10BC'	A1 A2 A3 A4		
10C0'	A5 A6 A7 A8	db 0A5h,0A6h,0A7h,0A8h,0A9h,8Fh,8Fh,8Fh	;C F6 F7 F8 F9 F10 NB NB NB
10C4'	A9 8F 8F 8F		
10C8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;C NB NB NB NB NB NB NB NB
10CC'	8F 8F 8F 8F		
10D0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;D NB NB NB NB NB NB NB NB
10D4'	8F 8F 8F 8F		
10D8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;D NB NB NB NB NB NB NB NB
10DC'	8F 8F 8F 8F		
10E0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;E NB NB NB NB NB NB NB NB
10E4'	8F 8F 8F 8F		
10E8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;E NB NB NB NB NB NB NB NB
10EC'	8F 8F 8F 8F		
10F0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;F NB NB NB NB NB NB NB NB
10F4'	8F 8F 8F 8F		
10F8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;F NB NB NB NB NB NB NB NB
10FC'	8F 8F 8F 8F		

org 1100h ;Controlebene (Control gedrückt)

; x0 x1 x2 x3 x4 x5 x6 x7 Adresse niederwertig (0-7)
; x8 x9 xA xB xC xD xE xF Adresse niederwertig (8-F)

				x	Adresse höherwertig (nach unten)
1100'	8F 1B 31 32	db 8Fh, 1Bh, 31h, 32h, 33h, 34h, 35h, 36h	;0		NB ESC 1 2 3 4 5 6 7
1104'	33 34 35 36				
1108'	37 38 39 30	db 37h, 38h, 39h, 30h, 7Eh, 27h, 08h, 09h	;0		7 8 9 0 B ` BS TAB
110C'	7E 27 08 09				
1110'	11 17 05 12	db 11h, 17h, 05h, 12h, 14h, 1Ah, 15h, 09h	;1		CTRL- -Q -W -E -R -T -Z -U -I
1114'	14 1A 15 09				
1118'	0F 10 1D 2B	db 0Fh, 10h, 1Dh, 2Bh, 0Dh, 80h, 01h, 13h	;1		CTRL- -O -P -Ü + CR CTRL -A -S
111C'	0D 80 01 13				
1120'	04 06 07 08	db 04h, 06h, 07h, 08h, 0Ah, 0Bh, 0Ch, 1Ch	;2		CTRL- -D -F -G -H -J -K -L -\
1124'	0A 0B 0C 1C				
1128'	1B 00 82 3C	db 1Bh, 00h, 82h, 3Ch, 19h, 18h, 03h, 16h	;2		CTRL- -Ä -ß SHIFT < -Y -X -C -V
112C'	19 18 03 16				
1130'	02 0E 0D 2C	db 02h, 0Eh, 0Dh, 2Ch, 2Eh, 2Dh, 82h, 00h	;3		CTRL- -B -N -M , . - SHIFT -ß
1134'	2E 2D 82 00				
1138'	84 20 86 90	db 84h, 20h, 86h, 90h, 91h, 92h, 93h, 94h	;3		ALT " " CAPSL. F1 F2 F3 F4 F5
113C'	91 92 93 94				
1140'	95 96 97 98	db 95h, 96h, 97h, 98h, 99h, 87h, 88h, 11h	;4		F6 F7 F8 F9 F10 NUML. NB CTRL-Q
1144'	99 87 88 11				
1148'	05 12 01 13	db 05h, 12h, 01h, 13h, 01h, 04h, 06h, 0Bh	;4		CTRL- -E -R -A -S -A -D -F -K
114C'	01 04 06 0B				
1150'	18 03 16 07	db 18h, 03h, 16h, 07h, 8Fh, 8Fh, 8Fh, 8Fh	;5		CTRL- -X -C -V -G
1154'	8F 8F 8F 8F				
1158'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;5		NB NB NB NB NB NB NB NB
115C'	8F 8F 8F 8F				
1160'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;6		NB NB NB NB NB NB NB NB
1164'	8F 8F 8F 8F				
1168'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;6		NB NB NB NB NB NB NB NB
116C'	8F 8F 8F 8F				
1170'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;7		NB NB NB NB NB NB NB NB
1174'	8F 8F 8F 8F				
1178'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;7		NB NB NB NB NB NB NB NB
117C'	8F 8F 8F 8F				
1180'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;8		NB NB NB NB NB NB NB NB
1184'	8F 8F 8F 8F				
1188'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;8		NB NB NB NB NB NB NB NB
118C'	8F 8F 8F 8F				
1190'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;9		NB NB NB NB NB NB NB NB
1194'	8F 8F 8F 8F				
1198'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 81h, 8Fh, 8Fh	;9		NB NB NB NB NB CTRL-E NB NB
119C'	8F 81 8F 8F				
11A0'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;A		NB NB NB NB NB NB NB NB
11A4'	8F 8F 8F 8F				
11A8'	8F 8F 83 8F	db 8Fh, 8Fh, 83h, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;A		NB NB SHIFTE NB NB NB NB NB
11AC'	8F 8F 8F 8F				
11B0'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 83h, 8Fh	;B		NB NB NB NB NB NB SHIFTE NB
11B4'	8F 8F 83 8F				
11B8'	85 8F 8F A0	db 85h, 8Fh, 8Fh, 0A0h, 0A1h, 0A2h, 0A3h, 0A4h	;B		ALTE NB NB NB NB NB NB NB
11BC'	A1 A2 A3 A4				
11C0'	A5 A6 A7 A8	db 0A5h, 0A6h, 0A7h, 0A8h, 0A9h, 8Fh, 8Fh, 8Fh	;C		NB NB NB NB NB NB NB NB
11C4'	A9 8F 8F 8F				
11C8'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;C		NB NB NB NB NB NB NB NB
11CC'	8F 8F 8F 8F				
11D0'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;D		NB NB NB NB NB NB NB NB
11D4'	8F 8F 8F 8F				

1108'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;D	NB NB NB NB NB NB NB NB
110C'	8F 8F 8F 8F			
11E0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;E	NB NB NB NB NB NB NB NB
11E4'	8F 8F 8F 8F			
11E8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;E	NB NB NB NB NB NB NB NB
11EC'	8F 8F 8F 8F			
11F0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;F	NB NB NB NB NB NB NB NB
11F4'	8F 8F 8F 8F			
11F8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;F	NB NB NB NB NB NB NB NB
11FC'	8F 8F 8F 8F			

org 1200h ;Shiftebene (Shift gedrückt oder Capslock eingeschaltet)

; x0 x1 x2 x3 x4 x5 x6 x7 Adresse niederwertig (0-7)
; x8 x9 xA xB xC xD xE xF Adresse niederwertig (8-F)
; x Adresse höherwertig (nach unten)

1200'	8F 1B 21 22	db 8Fh,1Bh,21h,22h,40h,24h,25h,26h	;0	NB ESC ! " \$ % &
1204'	40 24 25 26			
1208'	2F 28 29 3D	db 2Fh,28h,29h,3Dh,3Fh,60h,08h,09h	;0	/ () = ? ` BS TAB
120C'	3F 60 08 09			
1210'	51 57 45 52	db 51h,57h,45h,52h,54h,5Ah,55h,49h	;1	Q W E R T Z U I
1214'	54 5A 55 49			
1218'	4F 50 5D 2A	db 4Fh,50h,5Dh,2Ah,0Dh,80h,41h,53h	;1	O P Ü * CR CTRL A S
121C'	0D 80 41 53			
1220'	44 46 47 4B	db 44h,46h,47h,48h,4Ah,4Bh,4Ch,5Ch	;2	D F G H J K L \
1224'	4A 4B 4C 5C			
1228'	5B 5E 82 3E	db 5Bh,5Eh,82h,3Eh,59h,58h,43h,56h	;2	Ä ` SHIFT > Y X C V
122C'	59 58 43 56			
1230'	42 4E 4D 3B	db 42h,4Eh,4Dh,3Bh,3Ah,5Fh,82h,00h	;3	B N M ; : _ SHIFT CTRL-B
1234'	3A 5F 82 00			
1238'	84 20 86 90	db 84h,20h,86h,90h,91h,92h,93h,94h	;3	ALT " " CAPSL. F1 F2 F3 F4 F5
123C'	91 92 93 94			
1240'	95 96 97 98	db 95h,96h,97h,98h,99h,87h,88h,37h	;4	F6 F7 F8 F9 F10 NUML. NB 7
1244'	99 87 88 37			
1248'	38 39 2D 34	db 38h,39h,2Dh,34h,35h,36h,2Bh,31h	;4	8 9 - 4 5 6 + 1
124C'	35 36 2B 31			
1250'	32 33 30 2E	db 32h,33h,30h,2Eh,8Fh,8Fh,8Fh,8Fh	;5	2 3 0 . NB NB NB NB
1254'	8F 8F 8F 8F			
1258'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;5	NB NB NB NB NB NB NB NB
125C'	8F 8F 8F 8F			
1260'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;6	NB NB NB NB NB NB NB NB
1264'	8F 8F 8F 8F			
1268'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;6	NB NB NB NB NB NB NB NB
126C'	8F 8F 8F 8F			
1270'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;7	NB NB NB NB NB NB NB NB
1274'	8F 8F 8F 8F			
1278'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;7	NB NB NB NB NB NB NB NB
127C'	8F 8F 8F 8F			
1280'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;8	NB NB NB NB NB NB NB NB
1284'	8F 8F 8F 8F			
1288'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;8	NB NB NB NB NB NB NB NB
128C'	8F 8F 8F 8F			


```

1290' 8F 8F 8F 8F      db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;9  NB NB NB NB NB NB NB NB
1294' 8F 8F 8F 8F      db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;9  NB NB NB NB NB CTRL E NB NB
1298' 8F 8F 8F 8F      db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;A  NB NB NB NB NB NB NB NB
129C' 8F 81 8F 8F      db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;A  NB NB SHIFTE NB NB NB NB NB
12A0' 8F 8F 8F 8F      db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;B  NB NB NB NB NB NB SHIFTE NB
12A4' 8F 8F 8F 8F      db 85h,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;B  ALTE NB NB NB NB NB NB NB NB
12A8' 8F 8F 83 8F      db 0A5h,0A6h,0A7h,0A8h,0A9h,8Fh,8Fh,8Fh ;C  NB NB NB NB NB NB NB NB
12AC' 8F 8F 8F 8F      db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;C  NB NB NB NB NB NB NB NB
12B0' 8F 8F 8F 8F      db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;D  NB NB NB NB NB NB NB NB
12B4' 8F 8F 83 8F      db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;D  NB NB NB NB NB NB NB NB
12B8' 85 8F 8F A0      db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;E  NB NB NB NB NB NB NB NB
12BC' A1 A2 A3 A4      db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;E  NB NB NB NB NB NB NB NB
12C0' A5 A6 A7 A8      db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;F  NB NB NB NB NB NB NB NB
12C4' A9 8F 8F 8F      db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;F  NB NB NB NB NB NB NB NB
12C8' 8F 8F 8F 8F      ;
12CC' 8F 8F 8F 8F      ;
12D0' 8F 8F 8F 8F      ;
12D4' 8F 8F 8F 8F      ;
12D8' 8F 8F 8F 8F      ;
12DC' 8F 8F 8F 8F      ;
12E0' 8F 8F 8F 8F      ;
12E4' 8F 8F 8F 8F      ;
12E8' 8F 8F 8F 8F      ;
12EC' 8F 8F 8F 8F      ;
12F0' 8F 8F 8F 8F      ;
12F4' 8F 8F 8F 8F      ;
12F8' 8F 8F 8F 8F      ;
12FC' 8F 8F 8F 8F      ;

```

```

org 1300h      ;Control-Shiftebene (Control+Shift gedrückt)
               ; (=Controlebene)

```

```

; x0 x1 x2 x3 x4 x5 x6 x7      Adresse niederwertig (0-7)
; x8 x9 xA xB xC xD xE xF      Adresse niederwertig (8-F)
;                                x Adresse höherwertig (nach unten)

```

```

1300' 8F 1B 31 32      db 8Fh,1Bh,31h,32h,33h,34h,35h,36h ;0  NB ESC 1 2 3 4 5 6 7
1304' 33 34 35 36      ;
1308' 37 38 39 30      db 37h,38h,39h,30h,7Eh,27h,08h,09h ;0  7 8 9 0 B ` BS TAB
130C' 7E 27 08 09      ;
1310' 11 17 05 12      db 11h,17h,05h,12h,14h,1Ah,15h,09h ;1  CTRL- -Q -W -E -R -T -Z -U -I
1314' 14 1A 15 09      ;
1318' 0F 10 1D 2B      db 0Fh,10h,1Dh,2Bh,0Dh,80h,01h,13h ;1  CTRL- -O -P -Ü + CR CTRL -A -S
131C' 0D 80 01 13      ;
1320' 04 06 07 08      db 04h,06h,07h,08h,0Ah,0Bh,0Ch,1Ch ;2  CTRL- -D -F -G -H -J -K -L -\
1324' 0A 0B 0C 1C      ;
1328' 1B 00 82 3C      db 1Bh,00h,82h,3Ch,19h,18h,03h,16h ;2  CTRL- -Å -ß SHIFT < -Y -X -C -V
132C' 19 1B 03 16      ;
1330' 02 0E 0D 2C      db 02h,0Eh,0Dh,2Ch,2Eh,2Dh,82h,00h ;3  CTRL- -B -N -M , . - SHIFT -B
1334' 2E 2D 82 00      ;
1338' 84 20 86 90      db 84h,20h,86h,90h,91h,92h,93h,94h ;3  ALT " " CAPSL. F1 F2 F3 F4 F5
133C' 91 92 93 94      ;
1340' 95 96 97 98      db 95h,96h,97h,98h,99h,87h,88h,11h ;4  F6 F7 F8 F9 F10 NUML. NB CTRL-Q

```

1344'	99 87 88 11				
1348'	05 12 01 13	db 05h, 12h, 01h, 13h, 01h, 04h, 06h, 08h	;4	CTRL- -E -R -A -S -A -D -F -K	
134C'	01 04 06 08				
1350'	18 03 16 07	db 18h, 03h, 16h, 07h, 8Fh, 8Fh, 8Fh, 8Fh	;5	CTRL- -X -C -V -G	
1354'	8F 8F 8F 8F				
1358'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;5	NB NB NB NB NB NB NB NB	
135C'	8F 8F 8F 8F				
1360'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;6	NB NB NB NB NB NB NB NB	
1364'	8F 8F 8F 8F				
1368'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;6	NB NB NB NB NB NB NB NB	
136C'	8F 8F 8F 8F				
1370'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;7	NB NB NB NB NB NB NB NB	
1374'	8F 8F 8F 8F				
1378'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;7	NB NB NB NB NB NB NB NB	
137C'	8F 8F 8F 8F				
1380'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;8	NB NB NB NB NB NB NB NB	
1384'	8F 8F 8F 8F				
1388'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;8	NB NB NB NB NB NB NB NB	
138C'	8F 8F 8F 8F				
1390'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;9	NB NB NB NB NB NB NB NB	
1394'	8F 8F 8F 8F				
1398'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 81h, 8Fh, 8Fh	;9	NB NB NB NB NB NB CTRL E NB NB	
139C'	8F 81 8F 8F				
13A0'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;A	NB NB NB NB NB NB NB NB	
13A4'	8F 8F 8F 8F				
13A8'	8F 8F 83 8F	db 8Fh, 8Fh, 83h, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;A	NB NB SHIFTE NB NB NB NB NB	
13AC'	8F 8F 8F 8F				
13B0'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 83h, 8Fh	;B	NB NB NB NB NB NB SHIFTE NB	
13B4'	8F 8F 83 8F				
13B8'	85 8F 8F A0	db 85h, 8Fh, 8Fh, 0A0h, 0A1h, 0A2h, 0A3h, 0A4h	;B	ALTE NB NB NB NB NB NB NB	
13BC'	A1 A2 A3 A4				
13C0'	A5 A6 A7 A8	db 0A5h, 0A6h, 0A7h, 0A8h, 0A9h, 8Fh, 8Fh, 8Fh	;C	NB NB NB NB NB NB NB NB	
13C4'	A9 8F 8F 8F				
13C8'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;C	NB NB NB NB NB NB NB NB	
13CC'	8F 8F 8F 8F				
13D0'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;D	NB NB NB NB NB NB NB NB	
13D4'	8F 8F 8F 8F				
13D8'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;D	NB NB NB NB NB NB NB NB	
13DC'	8F 8F 8F 8F				
13E0'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;E	NB NB NB NB NB NB NB NB	
13E4'	8F 8F 8F 8F				
13E8'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;E	NB NB NB NB NB NB NB NB	
13EC'	8F 8F 8F 8F				
13F0'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;F	NB NB NB NB NB NB NB NB	
13F4'	8F 8F 8F 8F				
13F8'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;F	NB NB NB NB NB NB NB NB	
13FC'	8F 8F 8F 8F				

org 1400h ;Altebene (Alt gedrückt)
; (=Shiftebene)

; x0 x1 x2 x3 x4 x5 x6 x7 Adresse niederwertig (0-7)

		; xB x9 xA xB xC xD xE xF	Adresse niederwertig (8-F)
		;	x Adresse höherwertig (nach unten)
1400'	8F 1B 31 32	db 8Fh, 1Bh, 31h, 32h, 33h, 34h, 35h, 36h	;0 NB ESC 1 2 3 4 5 6
1404'	33 34 35 36		
1408'	37 38 39 30	db 37h, 38h, 39h, 30h, 3Fh, 60h, 08h, 09h	;0 7 8 9 0 ? ` BS TAB
140C'	3F 60 08 09		
1410'	51 57 45 52	db 51h, 57h, 45h, 52h, 54h, 5Ah, 55h, 49h	;1 Q W E R T Z U I
1414'	54 5A 55 49		
1418'	4F 50 5D 2A	db 4Fh, 50h, 5Dh, 2Ah, 0Dh, 80h, 41h, 53h	;1 O P Ü * CR CTRL A S
141C'	0D 80 41 53		
1420'	44 46 47 48	db 44h, 46h, 47h, 48h, 4Ah, 48h, 4Ch, 5Ch	;2 D F G H J K L \
1424'	4A 4B 4C 5C		
1428'	5B 5E 82 3E	db 5Bh, 5Eh, 82h, 3Eh, 59h, 58h, 43h, 56h	;2 Å ^ SHIFT > Y X C V
142C'	59 58 43 56		
1430'	42 4E 4D 3B	db 42h, 4Eh, 4Dh, 3Bh, 3Ah, 5Fh, 82h, 00h	;3 B N M ; : _ SHIFT CTRL-B
1434'	3A 5F 82 00		
1438'	84 20 86 90	db 84h, 20h, 86h, 90h, 91h, 92h, 93h, 94h	;3 ALT " " CAPSL. F1 F2 F3 F4 F5
143C'	91 92 93 94		
1440'	95 96 97 98	db 95h, 96h, 97h, 98h, 99h, 87h, 88h, 37h	;4 F6 F7 F8 F9 F10 NUML. NB 7
1444'	99 87 88 37		
1448'	38 39 2D 34	db 38h, 39h, 2Dh, 34h, 35h, 36h, 2Bh, 31h	;4 8 9 - 4 5 6 + 1
144C'	35 36 2B 31		
1450'	32 33 30 2E	db 32h, 33h, 30h, 2Eh, 8Fh, 8Fh, 8Fh, 8Fh	;5 2 3 0 . NB NB NB NB
1454'	8F 8F 8F 8F		
1458'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;5 NB NB NB NB NB NB NB NB
145C'	8F 8F 8F 8F		
1460'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;6 NB NB NB NB NB NB NB NB
1464'	8F 8F 8F 8F		
1468'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;6 NB NB NB NB NB NB NB NB
146C'	8F 8F 8F 8F		
1470'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;7 NB NB NB NB NB NB NB NB
1474'	8F 8F 8F 8F		
1478'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;7 NB NB NB NB NB NB NB NB
147C'	8F 8F 8F 8F		
1480'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;8 NB NB NB NB NB NB NB NB
1484'	8F 8F 8F 8F		
1488'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;8 NB NB NB NB NB NB NB NB
148C'	8F 8F 8F 8F		
1490'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;9 NB NB NB NB NB NB NB NB
1494'	8F 8F 8F 8F		
1498'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 81h, 8Fh, 8Fh	;9 NB NB NB NB NB NB CTRL E NB NB
149C'	8F 81 8F 8F		
14A0'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;A NB NB NB NB NB NB NB NB
14A4'	8F 8F 8F 8F		
14A8'	8F 8F 83 8F	db 8Fh, 8Fh, 83h, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;A NB NB SHIFTE NB NB NB NB NB
14AC'	8F 8F 8F 8F		
14B0'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 83h, 8Fh	;B NB NB NB NB NB NB SHIFTE NB
14B4'	8F 8F 83 8F		
14B8'	85 8F 8F A0	db 85h, 8Fh, 8Fh, 0A0h, 0A1h, 0A2h, 0A3h, 0A4h	;B ALTE NB NB NB NB NB NB NB
14BC'	A1 A2 A3 A4		
14C0'	A5 A6 A7 A8	db 0A5h, 0A6h, 0A7h, 0A8h, 0A9h, 8Fh, 8Fh, 8Fh	;C NB NB NB NB NB NB NB NB
14C4'	A9 8F 8F 8F		
14C8'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;C NB NB NB NB NB NB NB NB
14CC'	8F 8F 8F 8F		
14D0'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;D NB NB NB NB NB NB NB NB

14D4'	8F 8F 8F 8F			
14D8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;D	NB NB NB NB NB NB NB NB
14DC'	8F 8F 8F 8F			
14E0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;E	NB NB NB NB NB NB NB NB
14E4'	8F 8F 8F 8F			
14E8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;E	NB NB NB NB NB NB NB NB
14EC'	8F 8F 8F 8F			
14F0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;F	NB NB NB NB NB NB NB NB
14F4'	8F 8F 8F 8F			
14F8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;F	NB NB NB NB NB NB NB NB
14FC'	8F 8F 8F 8F			

org 1500h ;Control-Altebene (Control+ Alt gedrückt)
; (=Controlebene)

; x0 x1 x2 x3 x4 x5 x6 x7 Adresse niederwertig (0-7)
; x8 x9 xA xB xC xD xE xF Adresse niederwertig (8-F)
; x Adresse höherwertig (nach unten)

1500'	8F 1B 31 32	db 8Fh,1Bh,31h,32h,33h,34h,35h,36h	;0	NB ESC 1 2 3 4 5 6 7
1504'	33 34 35 36			
1508'	37 38 39 30	db 37h,38h,39h,30h,7Eh,27h,08h,09h	;0	7 8 9 0 B ` BS TAB
150C'	7E 27 08 09			
1510'	11 17 05 12	db 11h,17h,05h,12h,14h,1Ah,15h,09h	;1	CTRL- -Q -W -E -R -T -2 -U -I
1514'	14 1A 15 09			
1518'	0F 10 1D 2B	db 0Fh,10h,1Dh,2Bh,0Dh,80h,01h,13h	;1	CTRL- -O -P -Ü + CR CTRL -A -S
151C'	0D 80 01 13			
1520'	04 06 07 08	db 04h,06h,07h,08h,0Ah,0Bh,0Ch,1Ch	;2	CTRL- -D -F -G -H -J -K -L -\
1524'	0A 0B 0C 1C			
1528'	1B 00 82 3C	db 1Bh,00h,82h,3Ch,19h,18h,03h,16h	;2	CTRL- -Ä -B SHIFT < -Y -X -C -V
152C'	19 18 03 16			
1530'	02 0E 0D 2C	db 02h,0Eh,0Dh,2Ch,2Eh,2Dh,82h,00h	;3	CTRL- -B -N -M , . - SHIFT -S
1534'	2E 2D 82 00			
1538'	84 20 86 90	db 84h,20h,86h,90h,91h,92h,93h,94h	;3	ALT " " CAPSL. F1 F2 F3 F4 F5
153C'	91 92 93 94			
1540'	95 96 97 98	db 95h,96h,97h,98h,99h,87h,88h,11h	;4	F6 F7 F8 F9 F10 NUML. NB CTRL-Q
1544'	99 87 88 11			
1548'	05 12 01 13	db 05h,12h,01h,13h,01h,04h,06h,0Bh	;4	CTRL- -E -R -A -S -A -D -F -K
154C'	01 04 06 0B			
1550'	18 03 16 07	db 18h,03h,16h,07h,8Fh,8Fh,8Fh,8Fh	;5	CTRL- -X -C -V -G
1554'	8F 8F 8F 8F			
1558'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;5	NB NB NB NB NB NB NB NB
155C'	8F 8F 8F 8F			
1560'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;6	NB NB NB NB NB NB NB NB
1564'	8F 8F 8F 8F			
1568'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;6	NB NB NB NB NB NB NB NB
156C'	8F 8F 8F 8F			
1570'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;7	NB NB NB NB NB NB NB NB
1574'	8F 8F 8F 8F			
1578'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;7	NB NB NB NB NB NB NB NB
157C'	8F 8F 8F 8F			
1580'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;8	NB NB NB NB NB NB NB NB

1584'	8F 8F 8F 8F								
1588'	8F 8F 8F 8F	db	8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;B	NB NB NB NB NB NB NB NB				
158C'	8F 8F 8F 8F								
1590'	8F 8F 8F 8F	db	8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;9	NB NB NB NB NB NB NB NB				
1594'	8F 8F 8F 8F								
1598'	8F 8F 8F 8F	db	8Fh,8Fh,8Fh,8Fh,8Fh,81h,8Fh,8Fh	;9	NB NB NB NB NB CTRL E NB NB				
159C'	8F 81 8F 8F								
15A0'	8F 8F 8F 8F	db	8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;A	NB NB NB NB NB NB NB NB				
15A4'	8F 8F 8F 8F								
15A8'	8F 8F 83 8F	db	8Fh,8Fh,83h,8Fh,8Fh,8Fh,8Fh,8Fh	;A	NB NB SHIFTE NB NB NB NB NB				
15AC'	8F 8F 8F 8F								
15B0'	8F 8F 8F 8F	db	8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,83h,8Fh	;B	NB NB NB NB NB NB SHIFTE NB				
15B4'	8F 8F 83 8F								
15B8'	85 8F 8F A0	db	85h,8Fh,8Fh,0A0h,0A1h,0A2h,0A3h,0A4h	;B	ALTE NB NB NB NB NB NB NB				
15BC'	A1 A2 A3 A4								
15C0'	A5 A6 A7 8F	db	0A5h,0A6h,0A7h,08Fh,0A9h,8Fh,8Fh,8Fh	;C	NB NB NB NB NB NB NB NB				
15C4'	A9 8F 8F 8F								
15C8'	8F 8F 8F 8F	db	8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;C	NB NB NB NB NB NB NB NB				
15CC'	8F 8F 8F 8F								
15D0'	8F 8F 8F 8F	db	8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;D	NB NB NB NB NB NB NB NB				
15D4'	8F 8F 8F 8F								
15D8'	8F 8F 8F 8F	db	8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;D	NB NB NB NB NB NB NB NB				
15DC'	8F 8F 8F 8F								
15E0'	8F 8F 8F 8F	db	8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;E	NB NB NB NB NB NB NB NB				
15E4'	8F 8F 8F 8F								
15E8'	8F 8F 8F 8F	db	8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;E	NB NB NB NB NB NB NB NB				
15EC'	8F 8F 8F 8F								
15F0'	8F 8F 8F 8F	db	8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;F	NB NB NB NB NB NB NB NB				
15F4'	8F 8F 8F 8F								
15F8'	8F 8F 8F 8F	db	8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;F	NB NB NB NB NB NB NB NB				
15FC'	8F 8F 8F 8F								

org 1600h ;Shift-Altebene (Shift+Alt gedrückt)
; (=Shiftebene)

; x0 x1 x2 x3 x4 x5 x6 x7 Adresse niederwertig (0-7)
; x8 x9 xA xB xC xD xE xF Adresse niederwertig (8-F)
; x Adresse höherwertig (nach unten)

1600'	8F 1B 21 22	db	8Fh,1Bh,21h,22h,40h,24h,25h,26h	;0	NB ESC I " @ \$ % &				
1604'	40 24 25 26								
1608'	2F 28 29 3D	db	2Fh,28h,29h,3Dh,3Fh,60h,08h,09h	;0	/ () = ? ` BS TAB				
160C'	3F 60 08 09								
1610'	51 57 45 52	db	51h,57h,45h,52h,54h,5Ah,55h,49h	;1	Q W E R T Z U I				
1614'	54 5A 55 49								
1618'	4F 50 5D 2A	db	4Fh,50h,5Dh,2Ah,0Dh,80h,41h,53h	;1	O P Ü * CR CTRL A S				
161C'	0D 80 41 53								
1620'	44 46 47 48	db	44h,46h,47h,48h,4Ah,4Bh,4Ch,5Ch	;2	D F G H J K L \				
1624'	4A 4B 4C 5C								
1628'	5B 5E 82 3E	db	5Bh,5Eh,82h,3Eh,59h,58h,43h,56h	;2	Ä ^ SHIFT > Y X C V				
162C'	59 58 43 56								
1630'	42 4E 4D 3B	db	42h,4Eh,4Dh,3Bh,3Ah,5Fh,82h,00h	;3	B N M ; : _ SHIFT CTRL-B				
1634'	3A 5F 82 00								
1638'	84 20 86 90	db	84h,20h,86h,90h,91h,92h,93h,94h	;3	ALT " " CAPSL. F1 F2 F3 F4 F5				

163C'	91 92 93 94		
1640'	95 96 97 98	db 95h,96h,97h,98h,99h,87h,88h,37h	;4 F6 F7 F8 F9 F10 NUML. NB 7
1644'	99 87 88 37		
1648'	38 39 2D 34	db 38h,39h,2Dh,34h,35h,36h,28h,31h	;4 8 9 - 4 5 6 + 1
164C'	35 36 2B 31		
1650'	32 33 30 2E	db 32h,33h,30h,2Eh,8Fh,8Fh,8Fh,8Fh	;5 2 3 0 . NB NB NB NB
1654'	8F 8F 8F 8F		
1658'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;5 NB NB NB NB NB NB NB NB
165C'	8F 8F 8F 8F		
1660'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;6 NB NB NB NB NB NB NB NB
1664'	8F 8F 8F 8F		
1668'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;6 NB NB NB NB NB NB NB NB
166C'	8F 8F 8F 8F		
1670'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;7 NB NB NB NB NB NB NB NB
1674'	8F 8F 8F 8F		
1678'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;7 NB NB NB NB NB NB NB NB
167C'	8F 8F 8F 8F		
1680'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;8 NB NB NB NB NB NB NB NB
1684'	8F 8F 8F 8F		
1688'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;8 NB NB NB NB NB NB NB NB
168C'	8F 8F 8F 8F		
1690'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;9 NB NB NB NB NB NB NB NB
1694'	8F 8F 8F 8F		
1698'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;9 NB NB NB NB NB CTRL NB NB
169C'	8F 81 8F 8F		
16A0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;A NB NB NB NB NB NB NB NB
16A4'	8F 8F 8F 8F		
16A8'	8F 8F 83 8F	db 8Fh,8Fh,83h,8Fh,8Fh,8Fh,8Fh,8Fh	;A NB NB SHIFTE NB NB NB NB NB
16AC'	8F 8F 8F 8F		
16B0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,83h,8Fh	;B NB NB NB NB NB NB SHIFTE NB
16B4'	8F 8F 83 8F		
16B8'	85 8F 8F A0	db 85h,8Fh,8Fh,0A0h,0A1h,0A2h,0A3h,0A4h	;B ALTE NB NB NB NB NB NB NB
16BC'	A1 A2 A3 A4		
16C0'	A5 A6 A7 A8	db 0A5h,0A6h,0A7h,0A8h,0A9h,8Fh,8Fh,8Fh	;C NB NB NB NB NB NB NB NB
16C4'	A9 8F 8F 8F		
16C8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;C NB NB NB NB NB NB NB NB
16CC'	8F 8F 8F 8F		
16D0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;D NB NB NB NB NB NB NB NB
16D4'	8F 8F 8F 8F		
16D8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;D NB NB NB NB NB NB NB NB
16DC'	8F 8F 8F 8F		
16E0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;E NB NB NB NB NB NB NB NB
16E4'	8F 8F 8F 8F		
16E8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;E NB NB NB NB NB NB NB NB
16EC'	8F 8F 8F 8F		
16F0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;F NB NB NB NB NB NB NB NB
16F4'	8F 8F 8F 8F		
16F8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;F NB NB NB NB NB NB NB NB
16FC'	8F 8F 8F 8F		

org 1700h ;Control-Shift-Altebene (Control+ Alt+ Shift gedrückt)
; (=Controlebene)

; x0 x1 x2 x3 x4 x5 x6 x7 Adresse niederwertig (0-7)

		; x8 x9 xA xB xC xD xE xF	Adresse niederwertig (8-F)
		;	x Adresse höherwertig (nach unten)
1700'	8F 18 31 32	db 8Fh, 18h, 31h, 32h, 33h, 34h, 35h, 36h	;0 NB ESC 1 2 3 4 5 6 7
1704'	33 34 35 36		
1708'	37 38 39 30	db 37h, 38h, 39h, 30h, 7Eh, 27h, 08h, 09h	;0 7 8 9 0 B ` BS TAB
170C'	7E 27 08 09		
1710'	11 17 05 12	db 11h, 17h, 05h, 12h, 14h, 1Ah, 15h, 09h	;1 CTRL- -Q -W -E -R -T -Z -U -I
1714'	14 1A 15 09		
1718'	0F 10 1D 2B	db 0Fh, 10h, 1Dh, 2Bh, 0Dh, 80h, 01h, 13h	;1 CTRL- -O -P -Ü + CR CTRL -A -S
171C'	0D 80 01 13		
1720'	04 06 07 08	db 04h, 06h, 07h, 08h, 0Ah, 0Bh, 0Ch, 1Ch	;2 CTRL- -D -F -G -H -J -K -L -\
1724'	0A 0B 0C 1C		
1728'	1B 00 82 3C	db 1Bh, 00h, 82h, 3Ch, 19h, 18h, 03h, 16h	;2 CTRL- -Ä -B SHIFT < -Y -X -C -V
172C'	19 18 03 16		
1730'	02 0E 0D 2C	db 02h, 0Eh, 0Dh, 2Ch, 2Eh, 2Dh, 82h, 00h	;3 CTRL- -B -N -M , . - SHIFT -B
1734'	2E 2D 82 00		
1738'	84 20 86 90	db 84h, 20h, 86h, 90h, 91h, 92h, 93h, 94h	;3 ALT " " CAPSL. F1 F2 F3 F4 F5
173C'	91 92 93 94		
1740'	95 96 97 98	db 95h, 96h, 97h, 98h, 99h, 87h, 88h, 11h	;4 F6 F7 F8 F9 F10 NUML. NB CTRL-Q
1744'	99 87 88 11		
1748'	05 12 01 13	db 05h, 12h, 01h, 13h, 01h, 04h, 06h, 0Bh	;4 CTRL- -E -R -A -S -A -D -F -K
174C'	01 04 06 0B		
1750'	18 03 16 07	db 18h, 03h, 16h, 07h, 8Fh, 8Fh, 8Fh, 8Fh	;5 CTRL- -X -C -V -G
1754'	8F 8F 8F 8F		
1758'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;5 NB NB NB NB NB NB NB NB
175C'	8F 8F 8F 8F		
1760'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;6 NB NB NB NB NB NB NB NB
1764'	8F 8F 8F 8F		
1768'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;6 NB NB NB NB NB NB NB NB
176C'	8F 8F 8F 8F		
1770'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;7 NB NB NB NB NB NB NB NB
1774'	8F 8F 8F 8F		
1778'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;7 NB NB NB NB NB NB NB NB
177C'	8F 8F 8F 8F		
1780'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;8 NB NB NB NB NB NB NB NB
1784'	8F 8F 8F 8F		
1788'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;8 NB NB NB NB NB NB NB NB
178C'	8F 8F 8F 8F		
1790'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;9 NB NB NB NB NB NB NB NB
1794'	8F 8F 8F 8F		
1798'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 81h, 8Fh, 8Fh	;9 NB NB NB NB NB CTRL E NB NB
179C'	8F 81 8F 8F		
17A0'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;A NB NB NB NB NB NB NB NB
17A4'	8F 8F 8F 8F		
17A8'	8F 8F 83 8F	db 8Fh, 8Fh, 83h, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;A NB NB SHIFTE NB NB NB NB NB
17AC'	8F 8F 8F 8F		
17B0'	8F 8F 8F A0	db 8Fh, 8Fh, 8Fh, 0A0h, 0A1h, 0A2h, 0A3h, 0A4h	;B NB NB NB NB NB NB SHIFTE NB
17B4'	A1 A2 A3 A4		
17B8'	A5 A6 A7 A8	db 0A5h, 0A6h, 0A7h, 0A8h, 0A9h, 8Fh, 8Fh, 8Fh	;B ALTE NB NB NB NB NB NB NB
17BC'	A9 8F 8F 8F		
17C0'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;C NB NB NB NB NB NB NB NB
17C4'	8F 8F 8F 8F		
17C8'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;C NB NB NB NB NB NB NB NB
17CC'	8F 8F 8F 8F		
17D0'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;D NB NB NB NB NB NB NB NB

17D4' 8F 8F 8F 8F
 17D8' 8F 8F 8F 8F
 17DC' 8F 8F 8F 8F
 17E0' 8F 8F 8F 8F
 17E4' 8F 8F 8F 8F
 17E8' 8F 8F 8F 8F
 17EC' 8F 8F 8F 8F
 17F0' 8F 8F 8F 8F
 17F4' 8F 8F 8F 8F
 17F8' 8F 8F 8F 8F
 17FC' 8F 8F 8F 8F

db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;D NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;E NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;E NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;F NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;F NB NB NB NB NB NB NB NB

org 1800h

;Numlockebene (Numlock eingeschaltet)

; x0 x1 x2 x3 x4 x5 x6 x7 Adresse niederwertig (0-7)
 ; x8 x9 xA xB xC xD xE xF Adresse niederwertig (8-F)
 ; x Adresse höherwertig (nach unten)

1800' 8F 1B 31 32
 1804' 33 34 35 36
 1808' 37 38 39 30
 180C' 7E 27 08 09
 1810' 71 77 65 72
 1814' 74 7A 75 69
 1818' 6F 70 7D 2B
 181C' 0D 80 61 73
 1820' 64 66 67 68
 1824' 6A 6B 6C 7C
 1828' 7B 23 82 3C
 182C' 79 78 63 76
 1830' 62 6E 6D 2C
 1834' 2E 2D 82 2A
 1838' 84 20 86 90
 183C' 91 92 93 94
 1840' 95 96 97 98
 1844' 99 87 88 11
 1848' 05 12 01 13
 184C' 01 04 06 0B
 1850' 18 03 16 07
 1854' 8F 8F 8F 8F
 1858' 8F 8F 8F 8F
 185C' 8F 8F 8F 8F
 1860' 8F 8F 8F 8F
 1864' 8F 8F 8F 8F
 1868' 8F 8F 8F 8F
 186C' 8F 8F 8F 8F
 1870' 8F 8F 8F 8F
 1874' 8F 8F 8F 8F
 1878' 8F 8F 8F 8F
 187C' 8F 8F 8F 8F
 1880' 8F 8F 8F 8F
 1884' 8F 8F 8F 8F
 1888' 8F 8F 8F 8F
 188C' 8F 8F 8F 8F

db 8Fh,1Bh,31h,32h,33h,34h,35h,36h ;0 NB ESC 1 2 3 4 5 6 7
 db 37h,38h,39h,30h,7Eh,27h,08h,09h ;0 7 8 9 0 B ` BS TAB
 db 71h,77h,65h,72h,74h,7Ah,75h,69h ;1 q u e r t z u i
 db 6Fh,70h,7Dh,2Bh,0Dh,80h,61h,73h ;1 o p ü + CR CTRL a s
 db 64h,66h,67h,68h,6Ah,6Bh,6Ch,7Ch ;2 d f g h j k l ö
 db 7Bh,23h,82h,3Ch,79h,78h,63h,76h ;2 ß # SHIFT < y x c v
 db 62h,6Eh,6Dh,2Ch,2Eh,2Dh,82h,2Ah ;3 b n m , . - SHIFT *
 db 84h,20h,86h,90h,91h,92h,93h,94h ;3 ALT " " CAPSL. F1 F2 F3 F4 F5
 db 95h,96h,97h,98h,99h,87h,88h,11h ;4 F6 F7 F8 F9 F10 NUML. NB CTRL-Q
 db 05h,12h,01h,13h,04h,06h,0Bh ;4 CTRL- -E -R --A -S -A -D -F -K
 db 18h,03h,16h,07h,8Fh,8Fh,8Fh,8Fh ;5 CTRL- -X -C -V -G NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;5 NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;6 NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;6 NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;7 NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;7 NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;8 NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;8 NB NB NB NB NB NB NB NB

1890'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;9	NB NB NB NB NB NB NB NB
1894'	8F 8F 8F 8F			
1898'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,81h,8Fh,8Fh	;9	NB NB NB NB NB CTRL E NB NB
189C'	8F 81 8F 8F			
18A0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;A	NB NB NB NB NB NB NB NB
18A4'	8F 8F 8F 8F			
18AB'	8F 8F 83 8F	db 8Fh,8Fh,83h,8Fh,8Fh,8Fh,8Fh,8Fh	;A	NB NB SHIFTE NB NB NB NB NB
18AC'	8F 8F 8F 8F			
18B0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,83h,8Fh	;B	NB NB NB NB NB NB SHIFTE NB
18B4'	8F 8F 83 8F			
18B8'	85 8F 8F A0	db 85h,8Fh,8Fh,0A0h,0A1h,0A2h,0A3h,0A4h	;B	ALTE NB NB NB NB NB NB NB
18BC'	A1 A2 A3 A4			
18C0'	A5 A6 A7 A8	db 0A5h,0A6h,0A7h,0A8h,0A9h,8Fh,8Fh,8Fh	;C	NB NB NB NB NB NB NB NB
18C4'	A9 8F 8F 8F			
18C8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;C	NB NB NB NB NB NB NB NB
18CC'	8F 8F 8F 8F			
18D0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;D	NB NB NB NB NB NB NB NB
18D4'	8F 8F 8F 8F			
18D8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;D	NB NB NB NB NB NB NB NB
18DC'	8F 8F 8F 8F			
18E0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;E	NB NB NB NB NB NB NB NB
18E4'	8F 8F 8F 8F			
18E8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;E	NB NB NB NB NB NB NB NB
18EC'	8F 8F 8F 8F			
18F0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;F	NB NB NB NB NB NB NB NB
18F4'	8F 8F 8F 8F			
18F8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;F	NB NB NB NB NB NB NB NB
18FC'	8F 8F 8F 8F			

org 1900h ;Control-Numebene (Control+Numlock gedrückt)
; (=Controlebene)

; x0 x1 x2 x3 x4 x5 x6 x7 Adresse niederwertig (0-7)
; x8 x9 xA xB xC xD xE xF Adresse niederwertig (8-F)
; x Adresse höherwertig (nach unten)

1900'	8F 1B 31 32	db 8Fh,1Bh,31h,32h,33h,34h,35h,36h	;0	NB ESC 1 2 3 4 5 6 7
1904'	33 34 35 36			
1908'	37 38 39 30	db 37h,38h,39h,30h,7Eh,27h,08h,09h	;0	7 8 9 0 B ` BS TAB
190C'	7E 27 08 09			
1910'	11 17 05 12	db 11h,17h,05h,12h,14h,1Ah,15h,09h	;1	CTRL- -Q -W -E -R -T -Z -U -I
1914'	14 1A 15 09			
1918'	0F 10 1D 2B	db 0Fh,10h,1Dh,2Bh,0Dh,80h,01h,13h	;1	CTRL- -O -P -Ü + CR CTRL -A -S
191C'	0D 80 01 13			
1920'	04 06 07 08	db 04h,06h,07h,08h,0Ah,08h,0Ch,1Ch	;2	CTRL- -D -F -G -H -J -K -L -\
1924'	0A 0B 0C 1C			
1928'	1B 00 82 3C	db 1Bh,00h,82h,3Ch,19h,18h,03h,16h	;2	CTRL- -Ä -ß SHIFT < -Y -X -C -V
192C'	19 18 03 16			
1930'	02 0E 0D 2C	db 02h,0Eh,0Dh,2Ch,2Eh,2Dh,82h,00h	;3	CTRL- -B -M -M , . + SHIFT -B
1934'	2E 2D 82 00			
1938'	84 20 86 90	db 84h,20h,86h,90h,91h,92h,93h,94h	;3	ALT " " CAPSL. F1 F2 F3 F4 F5
193C'	91 92 93 94			
1940'	95 96 97 98	db 95h,96h,97h,98h,99h,87h,88h,11h	;4	F6 F7 F8 F9 F10 NUML. NB CTRL-Q
1944'	99 87 88 11			

1948'	05 12 01 13	db 05h, 12h, 01h, 13h, 01h, 04h, 06h, 08h	;4	CTRL- -E -R -A -S -A -D -F -K
194C'	01 04 06 0B			
1950'	18 03 16 07	db 18h, 03h, 16h, 07h, 8Fh, 8Fh, 8Fh, 8Fh	;5	CTRL- -X -C -V -G
1954'	8F 8F 8F 8F			
1958'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;5	NB NB NB NB NB NB NB NB
195C'	8F 8F 8F 8F			
1960'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;6	NB NB NB NB NB NB NB NB
1964'	8F 8F 8F 8F			
1968'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;6	NB NB NB NB NB NB NB NB
196C'	8F 8F 8F 8F			
1970'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;7	NB NB NB NB NB NB NB NB
1974'	8F 8F 8F 8F			
1978'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;7	NB NB NB NB NB NB NB NB
197C'	8F 8F 8F 8F			
1980'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;8	NB NB NB NB NB NB NB NB
1984'	8F 8F 8F 8F			
1988'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;8	NB NB NB NB NB NB NB NB
198C'	8F 8F 8F 8F			
1990'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;9	NB NB NB NB NB NB NB NB
1994'	8F 8F 8F 8F			
1998'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 81h, 8Fh, 8Fh	;9	NB NB NB NB NB CTRL E NB NB
199C'	8F 81 8F 8F			
19A0'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;A	NB NB NB NB NB NB NB NB
19A4'	8F 8F 8F 8F			
19A8'	8F 8F 83 8F	db 8Fh, 8Fh, 83h, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;A	NB NB SHIFTE NB NB NB NB NB
19AC'	8F 8F 8F 8F			
19B0'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 83h, 8Fh	;B	NB NB NB NB NB NB SHIFTE NB
19B4'	8F 8F 83 8F			
19B8'	85 8F 8F A0	db 85h, 8Fh, 8Fh, 0A0h, 0A1h, 0A2h, 0A3h, 0A4h	;B	ALTE NB NB NB NB NB NB NB
19BC'	A1 A2 A3 A4			
19C0'	A5 A6 A7 A8	db 0A5h, 0A6h, 0A7h, 0A8h, 0A9h, 8Fh, 8Fh, 8Fh	;C	NB NB NB NB NB NB NB NB
19C4'	A9 8F 8F 8F			
19C8'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;C	NB NB NB NB NB NB NB NB
19CC'	8F 8F 8F 8F			
19D0'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;D	NB NB NB NB NB NB NB NB
19D4'	8F 8F 8F 8F			
19D8'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;D	NB NB NB NB NB NB NB NB
19DC'	8F 8F 8F 8F			
19E0'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;E	NB NB NB NB NB NB NB NB
19E4'	8F 8F 8F 8F			
19E8'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;E	NB NB NB NB NB NB NB NB
19EC'	8F 8F 8F 8F			
19F0'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;F	NB NB NB NB NB NB NB NB
19F4'	8F 8F 8F 8F			
19F8'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;F	NB NB NB NB NB NB NB NB
19FC'	8F 8F 8F 8F			

org 1A00h ;Shift-Numebene (Shift/Caps und Num gedrückt)

; x0 x1 x2 x3 x4 x5 x6 x7 Adresse niederwertig (0-7)
 ; x8 x9 xA xB xC xD xE xF Adresse niederwertig (8-F)
 ; x Adresse höherwertig (nach unten)

1A00'	8F 1B 21 22	db 8Fh, 18h, 21h, 22h, 40h, 24h, 25h, 26h	;0	NB ESC I " B \$ % &
1A04'	40 24 25 26			
1A08'	2F 28 29 3D	db 2Fh, 28h, 29h, 3Dh, 3Fh, 60h, 08h, 09h	;0	/ () = ? ` BS TAB
1A0C'	3F 60 08 09			
1A10'	51 57 45 52	db 51h, 57h, 45h, 52h, 54h, 5Ah, 55h, 49h	;1	Q W E R T Z U I
1A14'	54 5A 55 49			
1A18'	4F 50 5D 2A	db 4Fh, 50h, 5Dh, 2Ah, 0Dh, 80h, 41h, 53h	;1	O P U * CR CTRL A S
1A1C'	0D 80 41 53			
1A20'	44 46 47 48	db 44h, 46h, 47h, 48h, 4Ah, 48h, 4Ch, 5Ch	;2	D F G H J K L \
1A24'	4A 4B 4C 5C			
1A28'	5B 5E 82 3E	db 5Bh, 5Eh, 82h, 3Eh, 59h, 58h, 43h, 56h	;2	Å ^ SHIFT > Y X C V
1A2C'	59 58 43 56			
1A30'	42 4E 4D 3B	db 42h, 4Eh, 4Dh, 3Bh, 3Ah, 5Fh, 82h, 00h	;3	B N M ; : _ SHIFT CTRL-S
1A34'	3A 5F 82 00			
1A38'	84 20 86 90	db 84h, 20h, 86h, 90h, 91h, 92h, 93h, 94h	;3	Alt " " CAPSL. F1 F2 F3 F4 F5
1A3C'	91 92 93 94			
1A40'	95 96 97 98	db 95h, 96h, 97h, 98h, 99h, 87h, 8Fh, 11h	;4	F6 F7 F8 F9 F10 NUML. NB CTRL-Q
1A44'	99 87 8F 11			
1A48'	05 12 01 13	db 05h, 12h, 01h, 13h, 01h, 04h, 06h, 0Bh	;4	CTRL- -E -R -A -S -A -D -F -K
1A4C'	01 04 06 0B			
1A50'	18 03 16 07	db 18h, 03h, 16h, 07h, 8Fh, 8Fh, 8Bh, 8Fh	;5	CTRL- -X -C -V -G NB NB NB NB
1A54'	8F 8F 8B 8F			
1A58'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;5	NB NB NB NB NB NB NB NB
1A5C'	8F 8F 8F 8F			
1A60'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;6	NB NB NB NB NB NB NB NB
1A64'	8F 8F 8F 8F			
1A68'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;6	NB NB NB NB NB NB NB NB
1A6C'	8F 8F 8F 8F			
1A70'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;7	NB NB NB NB NB NB NB NB
1A74'	8F 8F 8F 8F			
1A78'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;7	NB NB NB NB NB NB NB NB
1A7C'	8F 8F 8F 8F			
1A80'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;8	NB NB NB NB NB NB NB NB
1A84'	8F 8F 8F 8F			
1A88'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;8	NB NB NB NB NB NB NB NB
1A8C'	8F 8F 8F 8F			
1A90'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;9	NB NB NB NB NB NB NB NB
1A94'	8F 8F 8F 8F			
1A98'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 81h, 8Fh, 8Fh	;9	NB NB NB NB NB NB CTRL-E NB NB
1A9C'	8F 81 8F 8F			
1AA0'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;A	NB NB NB NB NB NB NB NB
1AA4'	8F 8F 8F 8F			
1AA8'	8F 8F 83 8F	db 8Fh, 8Fh, 83h, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;A	NB NB SHIFTE NB NB NB NB NB
1AAC'	8F 8F 8F 8F			
1AB0'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 83h, 8Fh	;B	NB NB NB NB NB NB SHIFTE NB
1AB4'	8F 8F 83 8F			
1AB8'	85 8F 8F A0	db 85h, 8Fh, 8Fh, 0A0h, 0A1h, 0A2h, 0A3h, 0A4h	;B	ALTE NB NB F1 F2 F3 F4 F5
1ABC'	A1 A2 A3 A4			
1AC0'	A5 A6 A7 A8	db 0A5h, 0A6h, 0A7h, 0A8h, 0A9h, 8Fh, 8Fh, 8Fh	;C	F6 F7 F8 F9 NB NB NB NB
1AC4'	A9 8F 8F 8F			
1AC8'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;C	NB NB NB NB NB NB NB NB
1ACC'	8F 8F 8F 8F			
1AD0'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;D	NB NB NB NB NB NB NB NB
1AD4'	8F 8F 8F 8F			
1AD8'	8F 8F 8F 8F	db 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh, 8Fh	;D	NB NB NB NB NB NB NB NB

1ADC' 8F 8F 8F 8F
 1AEO' 8F 8F 8F 8F
 1AE4' 8F 8F 8F 8F
 1AEB' 8F 8F 8F 8F
 1AEC' 8F 8F 8F 8F
 1AFO' 8F 8F 8F 8F
 1AF4' 8F 8F 8F 8F
 1AFB' 8F 8F 8F 8F
 1AFC' 8F 8F 8F 8F

db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;E NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;E NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;F NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;F NB NB NB NB NB NB NB NB

org 1800h ;Control-Shift-Numbene (Control+ Shift+ Numlock
 ; gedrückt) (= Controlebene)

; x0 x1 x2 x3 x4 x5 x6 x7 Adresse niederwertig (0-7)
 ; x8 x9 xA xB xC xD xE xF Adresse niederwertig (8-F)
 ; x Adresse höherwertig (nach unten)

1800' 8F 1B 31 32
 1804' 33 34 35 36
 1808' 37 38 39 30
 180C' 7E 27 08 09
 1810' 11 17 05 12
 1814' 14 1A 15 09
 1818' 0F 10 1D 2B
 181C' 0D 80 01 13
 1820' 04 06 07 08
 1824' 0A 0B 0C 1C
 1828' 1B 00 82 3C
 182C' 19 18 03 16
 1830' 02 0E 0D 2C
 1834' 2E 2D 82 00
 1838' 84 20 86 90
 183C' 91 92 93 94
 1840' 95 96 97 98
 1844' 99 87 88 11
 1848' 05 12 01 13
 184C' 01 04 06 0B
 1850' 18 03 16 07
 1854' 8F 8F 8F 8F
 1858' 8F 8F 8F 8F
 185C' 8F 8F 8F 8F
 1860' 8F 8F 8F 8F
 1864' 8F 8F 8F 8F
 1868' 8F 8F 8F 8F
 186C' 8F 8F 8F 8F
 1870' 8F 8F 8F 8F
 1874' 8F 8F 8F 8F
 1878' 8F 8F 8F 8F
 187C' 8F 8F 8F 8F
 1880' 8F 8F 8F 8F
 1884' 8F 8F 8F 8F
 1888' 8F 8F 8F 8F
 188C' 8F 8F 8F 8F
 1890' 8F 8F 8F 8F

db 8Fh,1Bh,31h,32h,33h,34h,35h,36h ;0 NB ESC 1 2 3 4 5 6 7
 db 37h,38h,39h,30h,7Eh,27h,08h,09h ;0 7 8 9 0 B ` BS TAB
 db 11h,17h,05h,12h,14h,1Ah,15h,09h ;1 CTRL- -Q -W -E -R -T -Z -U -I
 db 0Fh,10h,1Dh,2Bh,0Dh,80h,01h,13h ;1 CTRL- -O -P -Ü + CR CTRL -A -S
 db 04h,06h,07h,08h,0Ah,0Bh,0Ch,1Ch ;2 CTRL- -D -F -G -H -J -K -L -\
 db 1Bh,00h,82h,3Ch,19h,18h,03h,16h ;2 CTRL- -Ä -B SHIFT < -Y -X -C -V
 db 02h,0Eh,0Dh,2Ch,2Eh,2Dh,82h,00h ;3 CTRL- -B -N -M , . - SHIFT -B
 db 84h,20h,86h,90h,91h,92h,93h,94h ;3 ALT " " CAPSL. F1 F2 F3 F4 F5
 db 95h,96h,97h,98h,99h,87h,88h,11h ;4 F6 F7 F8 F9 F10 NUML. NB CTRL-Q
 db 05h,12h,01h,13h,01h,04h,06h,0Bh ;4 CTRL- -E -R -A -S -A -D -F -K
 db 18h,03h,16h,07h,8Fh,8Fh,8Fh,8Fh ;5 CTRL- -X -C -V -G
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;5 NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;6 NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;6 NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;7 NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;7 NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;8 NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;8 NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;9 NB NB NB NB NB NB NB NB

1C4C'	01 04 06 0B			
1C50'	1B 03 16 07	db 18h,03h,16h,07h,8Fh,8Fh,8Fh,8Fh	;5	CTRL- -X -C -V -G NB NB NB NB
1C54'	8F 8F 8F 8F			
1C58'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;5	NB NB NB NB NB NB NB NB
1C5C'	8F 8F 8F 8F			
1C60'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;6	NB NB NB NB NB NB NB NB
1C64'	8F 8F 8F 8F			
1C68'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;6	NB NB NB NB NB NB NB NB
1C6C'	8F 8F 8F 8F			
1C70'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;7	NB NB NB NB NB NB NB NB
1C74'	8F 8F 8F 8F			
1C78'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;7	NB NB NB NB NB NB NB NB
1C7C'	8F 8F 8F 8F			
1C80'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;8	NB NB NB NB NB NB NB NB
1C84'	8F 8F 8F 8F			
1C88'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;8	NB NB NB NB NB NB NB NB
1C8C'	8F 8F 8F 8F			
1C90'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;9	NB NB NB NB NB NB NB NB
1C94'	8F 8F 8F 8F			
1C98'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;9	NB NB NB NB NB CTRL E NB NB
1C9C'	8F 81 8F 8F			
1CA0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;A	NB NB NB NB NB NB NB NB
1CA4'	8F 8F 8F 8F			
1CA8'	8F 8F 83 8F	db 8Fh,8Fh,83h,8Fh,8Fh,8Fh,8Fh,8Fh	;A	NB NB SHIFTE NB NB NB NB NB
1CAC'	8F 8F 8F 8F			
1CB0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,83h,8Fh	;B	NB NB NB NB NB NB SHIFTE NB
1CB4'	8F 8F 83 8F			
1CB8'	85 8F 8F A0	db 85h,8Fh,8Fh,0A0h,0A1h,0A2h,0A3h,0A4h	;B	ALTE NB NB NB NB NB NB NB
1CBC'	A1 A2 A3 A4			
1CC0'	A5 A6 A7 A8	db 0A5h,0A6h,0A7h,0A8h,0A9h,8Fh,8Fh,8Fh	;C	NB NB NB NB NB NB NB NB
1CC4'	A9 8F 8F 8F			
1CC8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;C	NB NB NB NB NB NB NB NB
1CCC'	8F 8F 8F 8F			
1CD0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;D	NB NB NB NB NB NB NB NB
1CD4'	8F 8F 8F 8F			
1CD8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;D	NB NB NB NB NB NB NB NB
1CDC'	8F 8F 8F 8F			
1CE0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;E	NB NB NB NB NB NB NB NB
1CE4'	8F 8F 8F 8F			
1CE8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;E	NB NB NB NB NB NB NB NB
1CEC'	8F 8F 8F 8F			
1CF0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;F	NB NB NB NB NB NB NB NB
1CF4'	8F 8F 8F 8F			
1CF8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;F	NB NB NB NB NB NB NB NB
1CFC'	8F 8F 8F 8F			

org 1d00h ;Control-Alt-Numebene (Control+ Alt+ Numlock
; gedrückt) (= Controlebene)

; x0 x1 x2 x3 x4 x5 x6 x7 Adresse niederwertig (0-7)
; x8 x9 xA xB xC xD xE xF Adresse niederwertig (8-F)
; x Adresse höherwertig (nach unten)

1D00' 8F 1B 31 32 db 8Fh,1Bh,31h,32h,33h,34h,35h,36h ;0 NB ESC 1 2 3 4 5 6 7

1D04'	33 34 35 36				
1D08'	37 38 39 30	db 37h,38h,39h,30h,7Eh,27h,08h,09h	;0	7 8 9 0 8 ` BS TAB	
1D0C'	7E 27 08 09				
1D10'	11 17 05 12	db 11h,17h,05h,12h,14h,1Ah,15h,09h	;1	CTRL- -Q -W -E -R -T -Z -U -I	
1D14'	14 1A 15 09				
1D18'	0F 10 1D 2B	db 0Fh,10h,1Dh,2Bh,0Dh,80h,01h,13h	;1	CTRL- -O -P -Ü + CR CTRL -A -S	
1D1C'	0D 80 01 13				
1D20'	04 06 07 08	db 04h,06h,07h,08h,0Ah,0Bh,0Ch,1Ch	;2	CTRL- -D -F -G -H -J -K -L -\	
1D24'	0A 0B 0C 1C				
1D28'	1B 00 82 3C	db 1Bh,00h,82h,3Ch,19h,18h,03h,16h	;2	CTRL- -Ä -ß SHIFT < -Y -X -C -V	
1D2C'	19 18 03 16				
1D30'	02 0E 0D 2C	db 02h,0Eh,0Dh,2Ch,2Eh,2Dh,82h,00h	;3	CTRL- -B -N -M , . - SHIFT -ß	
1D34'	2E 2D 82 00				
1D38'	84 20 86 90	db 84h,20h,86h,90h,91h,92h,93h,94h	;3	ALT " " CAPSL. F1 F2 F3 F4 F5	
1D3C'	91 92 93 94				
1D40'	95 96 97 98	db 95h,96h,97h,98h,99h,87h,88h,11h	;4	F6 F7 F8 F9 F10 NUML. NB CTRL-Q	
1D44'	99 87 88 11				
1D48'	05 12 01 13	db 05h,12h,01h,13h,01h,04h,06h,0Bh	;4	CTRL- -E -R -A -S -A -D -F -K	
1D4C'	01 04 06 0B				
1D50'	18 03 16 07	db 18h,03h,16h,07h,8Fh,8Fh,8Fh,8Fh	;5	CTRL- -X -C -V -G	
1D54'	8F 8F 8F 8F				
1D58'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;5	NB NB NB NB NB NB NB NB	
1D5C'	8F 8F 8F 8F				
1D60'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;6	NB NB NB NB NB NB NB NB	
1D64'	8F 8F 8F 8F				
1D68'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;6	NB NB NB NB NB NB NB NB	
1D6C'	8F 8F 8F 8F				
1D70'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;7	NB NB NB NB NB NB NB NB	
1D74'	8F 8F 8F 8F				
1D78'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;7	NB NB NB NB NB NB NB NB	
1D7C'	8F 8F 8F 8F				
1D80'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;8	NB NB NB NB NB NB NB NB	
1D84'	8F 8F 8F 8F				
1D88'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;8	NB NB NB NB NB NB NB NB	
1D8C'	8F 8F 8F 8F				
1D90'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;9	NB NB NB NB NB NB NB NB	
1D94'	8F 8F 8F 8F				
1D98'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,81h,8Fh,8Fh	;9	NB NB NB NB NB CTRL-E NB NB	
1D9C'	8F 81 8F 8F				
1DA0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;A	NB NB NB NB NB NB NB NB	
1DA4'	8F 8F 8F 8F				
1DAB'	8F 8F 83 8F	db 8Fh,8Fh,83h,8Fh,8Fh,8Fh,8Fh,8Fh	;A	NB NB SHIFTE NB NB NB NB NB	
1DAC'	8F 8F 8F 8F				
1DB0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,83h,8Fh	;B	NB NB NB NB NB NB SHIFTE NB	
1DB4'	8F 8F 83 8F				
1DB8'	85 8F 8F A0	db 85h,8Fh,8Fh,0A0h,0A1h,0A2h,0A3h,0A4h	;B	ALTE NB NB NB NB NB NB NB	
1DBC'	A1 A2 A3 A4				
1DC0'	A5 A6 A7 A8	db 0A5h,0A6h,0A7h,0A8h,0A9h,8Fh,8Fh,8Fh	;C	NB NB NB NB NB NB NB NB	
1DC4'	A9 8F 8F 8F				
1DC8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;C	NB NB NB NB NB NB NB NB	
1DCC'	8F 8F 8F 8F				
1DD0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;D	NB NB NB NB NB NB NB NB	
1DD4'	8F 8F 8F 8F				
1DD8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;D	NB NB NB NB NB NB NB NB	
1DDC'	8F 8F 8F 8F				
1DE0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;E	NB NB NB NB NB NB NB NB	

1DE4' 8F 8F 8F 8F
 1DE8' 8F 8F 8F 8F
 1DEC' 8F 8F 8F 8F
 1DF0' 8F 8F 8F 8F
 1DF4' 8F 8F 8F 8F
 1DF8' 8F 8F 8F 8F
 1DFC' 8F 8F 8F 8F

db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;E NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;F NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;F NB NB NB NB NB NB NB NB

org 1E00h ;Shift-Alt-Numebene (Shift/Caps + Num +Alt
 ; gedrückt) (=Num-Shiftebene)
 ; x0 x1 x2 x3 x4 x5 x6 x7 Adresse niederwertig (0-7)
 ; x8 x9 xA xB xC xD xE xF Adresse niederwertig (8-F)
 ; x Adresse höherwertig (nach unten)

1E00' 8F 1B 21 22
 1E04' 40 24 25 26
 1E08' 2F 28 29 3D
 1E0C' 3F 60 08 09
 1E10' 51 57 45 52
 1E14' 54 5A 55 49
 1E18' 4F 50 5D 2A
 1E1C' 0D 80 41 53
 1E20' 44 46 47 48
 1E24' 4A 4B 4C 5C
 1E28' 5B 5E 82 3E
 1E2C' 59 58 43 56
 1E30' 42 4E 4D 3B
 1E34' 3A 5F 82 00
 1E38' 84 20 86 90
 1E3C' 91 92 93 94
 1E40' 95 96 97 98
 1E44' 99 87 88 11
 1E48' 05 12 01 13
 1E4C' 01 04 06 0B
 1E50' 1B 03 16 07
 1E54' 8F 8F 8F 8F
 1E58' 8F 8F 8F 8F
 1E5C' 8F 8F 8F 8F
 1E60' 8F 8F 8F 8F
 1E64' 8F 8F 8F 8F
 1E68' 8F 8F 8F 8F
 1E6C' 8F 8F 8F 8F
 1E70' 8F 8F 8F 8F
 1E74' 8F 8F 8F 8F
 1E78' 8F 8F 8F 8F
 1E7C' 8F 8F 8F 8F
 1E80' 8F 8F 8F 8F
 1E84' 8F 8F 8F 8F
 1E88' 8F 8F 8F 8F
 1E8C' 8F 8F 8F 8F
 1E90' 8F 8F 8F 8F
 1E94' 8F 8F 8F 8F

db 8Fh,1Bh,21h,22h,40h,24h,25h,26h ;0 NB ESC ! " # \$ % &
 db 2Fh,28h,29h,3Dh,3Fh,60h,08h,09h ;0 / () = ? ` BS TAB
 db 51h,57h,45h,52h,54h,5Ah,55h,49h ;1 Q W E R T Z U I
 db 4Fh,50h,5Dh,2Ah,0Dh,80h,41h,53h ;1 O P Ü * CR CTRL A S
 db 44h,46h,47h,48h,4Ah,4Bh,4Ch,5Ch ;2 D F G H J K L \
 db 5Bh,5Eh,82h,3Eh,59h,58h,43h,56h ;2 Å ^ SHIFT > Y X C V
 db 42h,4Eh,4Dh,3Bh,3Ah,5Fh,82h,00h ;3 B N M ; : _ SHIFT CTRL-S
 db 84h,20h,86h,90h,91h,92h,93h,94h ;3 Alt " " CAPSL. F1 F2 F3 F4 F5
 db 95h,96h,97h,98h,99h,87h,88h,11h ;4 F6 F7 F8 F9 F10 NUML. NB CTRL-O
 db 05h,12h,01h,13h,01h,04h,06h,0Bh ;4 CTRL- -E -R, -A -S -A -D -F -K
 db 1Bh,03h,16h,07h,8Fh,8Fh,8Fh,8Fh ;5 CTRL- -X -C -V -G NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;5 NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;6 NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;6 NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;7 NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;7 NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;8 NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;8 NB NB NB NB NB NB NB NB
 db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh ;9 NB NB NB NB NB NB NB NB

1E98'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,81h,8Fh,8Fh	;9	NB NB NB NB NB CTRL NB NB
1E9C'	8F 81 8F 8F			
1EA0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;A	NB NB NB NB NB NB NB NB
1EA4'	8F 8F 8F 8F			
1EA8'	8F 8F 83 8F	db 8Fh,8Fh,83h,8Fh,8Fh,8Fh,8Fh,8Fh	;A	NB NB SHIFTE NB NB NB NB NB
1EAC'	8F 8F 8F 8F			
1EB0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,83h,8Fh	;B	NB NB NB NB NB NB SHIFTE NB
1EB4'	8F 8F 83 8F			
1EB8'	85 8F 8F A0	db 85h,8Fh,8Fh,0A0h,0A1h,0A2h,0A3h,0A4h	;B	ALTE NB NB NB NB NB NB NB
1EBC'	A1 A2 A3 A4			
1EC0'	A5 A6 A7 AB	db 0A5h,0A6h,0A7h,0A8h,0A9h,8Fh,8Fh,8Fh	;C	NB NB NB NB NB NB NB NB
1EC4'	A9 8F 8F 8F			
1EC8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;C	NB NB NB NB NB NB NB NB
1ECC'	8F 8F 8F 8F			
1ED0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;D	NB NB NB NB NB NB NB NB
1ED4'	8F 8F 8F 8F			
1ED8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;D	NB NB NB NB NB NB NB NB
1EDC'	8F 8F 8F 8F			
1EE0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;E	NB NB NB NB NB NB NB NB
1EE4'	8F 8F 8F 8F			
1EE8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;E	NB NB NB NB NB NB NB NB
1EEC'	8F 8F 8F 8F			
1EFO'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;F	NB NB NB NB NB NB NB NB
1EF4'	8F 8F 8F 8F			
1EF8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;F	NB NB NB NB NB NB NB NB
1EFC'	8F 8F 8F 8F			

org 1F00h ;Control-Alt-Shift-Numebene (Control+ Alt+ Numlock+
; Shift gedrückt) (= Controlebene)

; x0 x1 x2 x3 x4 x5 x6 x7 Adresse niederwertig (0-7)
; x8 x9 xA xB xC xD xE xF Adresse niederwertig (8-F)
; x Adresse höherwertig (nach unten)

1F00'	8F 1B 31 32	db 8Fh,1Bh,31h,32h,33h,34h,35h,36h	;0	NB ESC 1 2 3 4 5 6 7
1F04'	33 34 35 36			
1F08'	37 38 39 30	db 37h,38h,39h,30h,7Eh,27h,08h,09h	;0	7 8 9 0 B BS TAB
1F0C'	7E 27 08 09			
1F10'	11 17 05 12	db 11h,17h,05h,12h,14h,1Ah,15h,09h	;1	CTRL- -Q -W -E -R -T -Z -U -I
1F14'	14 1A 15 09			
1F18'	0F 10 1D 2B	db 0Fh,10h,1Dh,2Bh,0Dh,80h,01h,13h	;1	CTRL- -O -P -Ü + CR CTRL -A -S
1F1C'	0D 80 01 13			
1F20'	04 06 07 08	db 04h,06h,07h,08h,0Ah,08h,0Ch,1Ch	;2	CTRL- -D -F -G -H -J -K -L -\
1F24'	0A 0B 0C 1C			
1F28'	1B 00 82 3C	db 1Bh,00h,82h,3Ch,19h,18h,03h,16h	;2	CTRL- -Ä -B SHIFT < -Y -X -C -V
1F2C'	19 18 03 16			
1F30'	02 0E 0D 2C	db 02h,0Eh,0Dh,2Ch,2Eh,2Dh,82h,00h	;3	CTRL- -B -M -M , . - SHIFT -B
1F34'	2E 2D 82 00			
1F38'	84 20 86 90	db 84h,20h,86h,90h,91h,92h,93h,94h	;3	ALT " " CAPSL. F1 F2 F3 F4 F5
1F3C'	91 92 93 94			
1F40'	95 96 97 98	db 95h,96h,97h,98h,99h,87h,88h,11h	;4	F6 F7 F8 F9 F10 NUML. NB CTRL-Q
1F44'	99 87 88 11			

1F48'	05 12 01 13	db 05h,12h,01h,13h,01h,04h,06h,08h	;4	CTRL- -E -R -A -S -A -D -F -K
1F4C'	01 04 06 0B			
1F50'	18 03 16 07	db 18h,03h,16h,07h,8Fh,8Fh,8Fh,8Fh	;5	CTRL- -X -C -V -G
1F54'	8F 8F 8F 8F			
1F58'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;5	NB NB NB NB NB NB NB NB
1F5C'	8F 8F 8F 8F			
1F60'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;6	NB NB NB NB NB NB NB NB
1F64'	8F 8F 8F 8F			
1F68'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;6	NB NB NB NB NB NB NB NB
1F6C'	8F 8F 8F 8F			
1F70'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;7	NB NB NB NB NB NB NB NB
1F74'	8F 8F 8F 8F			
1F78'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;7	NB NB NB NB NB NB NB NB
1F7C'	8F 8F 8F 8F			
1F80'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;8	NB NB NB NB NB NB NB NB
1F84'	8F 8F 8F 8F			
1F88'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;8	NB NB NB NB NB NB NB NB
1F8C'	8F 8F 8F 8F			
1F90'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;9	NB NB NB NB NB NB NB NB
1F94'	8F 8F 8F 8F			
1F98'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;9	NB NB NB NB NB NB CTRL NB NB
1F9C'	8F 8F 8F 8F			
1FA0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;A	NB NB NB NB NB NB NB NB
1FA4'	8F 8F 8F 8F			
1FAB'	8F 8F 83 8F	db 8Fh,8Fh,83h,8Fh,8Fh,8Fh,8Fh,8Fh	;A	NB NB SHIFTE NB NB NB NB NB
1FAC'	8F 8F 8F 8F			
1FB0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,83h,8Fh	;B	NB NB NB NB NB NB SHIFTE NB
1FB4'	8F 8F 83 8F			
1FB8'	85 8F 8F A0	db 85h,8Fh,8Fh,0A0h,0A1h,0A2h,0A3h,0A4h	;B	ALTE NB NB NB NB NB NB NB
1FBC'	A1 A2 A3 A4			
1FC0'	A5 A6 A7 A8	db 0A5h,0A6h,0A7h,0A8h,0A9h,8Fh,8Fh,8Fh	;C	NB NB NB NB NB NB NB NB
1FC4'	A9 8F 8F 8F			
1FC8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;C	NB NB NB NB NB NB NB NB
1FCC'	8F 8F 8F 8F			
1FD0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;D	NB NB NB NB NB NB NB NB
1FD4'	8F 8F 8F 8F			
1FD8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;D	NB NB NB NB NB NB NB NB
1FDC'	8F 8F 8F 8F			
1FE0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;E	NB NB NB NB NB NB NB NB
1FE4'	8F 8F 8F 8F			
1FEB'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;E	NB NB NB NB NB NB NB NB
1FEC'	8F 8F 8F 8F			
1FF0'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;F	NB NB NB NB NB NB NB NB
1FF4'	8F 8F 8F 8F			
1FF8'	8F 8F 8F 8F	db 8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh,8Fh	;F	NB NB NB NB NB NB NB NB
1FFC'	8F 8F 8F 8F			

2000' dseg

0000" 00 off: defb 0 ;Offsetvariable (Voreinst. = 80h)
 0001" 00 ze: defb 0 ;Zeichenplatzhalter


```

0002" 00      pufe:  defb 0      ;Pufferzeiger beim Einlesen
0003" 00      pufa:  defb 0      ;Pufferzeiger beim Auslesen
0004" 00      pufz:  defb 0      ;Zahl der Zeichen im Puffer
0005" 00      pufh:  defb 0      ;Pufferhilfszeiger
0006" 00      funz:  defb 0      ;Zeiger für Funktionstasteneinlesen
0007" 00      nmie:  defb 0      ;NMI-Enable
0008" 00      pgm:   defb 0      ;Programmierererkennung
0009" 00      funpr: defb 0      ;zu programmierende Funktionstaste
000A" 00 00    pgma:  defb 0,0    ;Programmieranfang (2 Byte)
000C" 00      wiedz: defb 0      ;Zähler für Befehlswiederholung
000D" 00      wiedh: defb 0      ;Hilfsvar für Befehlswiederholung
000E" 00      wieda: defb 0      ;Wiederholungsanfang im Puffer
000F" 00      wiede: defb 0      ;Wiederholungsende im Puffer
0010" 00      intstop: defb 0    ;Hilfsvar. zur Verhinderung Ausgabefehlern
0011" 00 00    ivektor: defb 0,0 ;Vektor für Adresse des Intpgms
0013" 00 00    nvektor: defb 0,0 ;Vektor für Adresse des Nmipgms

```

Macros:

Symbols:

0683'	FEIN	06A4'	FUNKEND	06A7'	FUNKLOS
062C'	FUNKT	0642'	FUNKT1	0672'	FUNKT2
0009"	FUNPR	0006"	FUNZ	0200'	HAUPT
0243'	HAUPT1	0251'	HAUPT2	0500'	INT
052A'	INT1	0400'	INTA	0748'	INTEND
0010"	INTSTOP	006A'	INTV	0252'	INTWART
0011"	IVEKTOR	0A00'	NMI	0900'	NMIA
0007"	NMIE	0A26'	NMIEND	0A26'	NMIUE
006F'	NMIV	0013"	NVEKTOR	0000"	OFF
0008"	PGM	000A"	PGMA	0000	PORT1
0001	PORT2	0002	PORT3	0005	PORT5
0003"	PUFA	0002"	PUFE	0005"	PUFH
05E0'	PUFPGM	0618'	PUFUEND	05E6'	PUFUPGM
062B'	PUFURET	0004"	PUFZ	054E'	SOD1
055E'	SOD2	056E'	SOD3	057E'	SOD4
058E'	SOD5	059E'	SOD6	05AE'	SOD7
058E'	SOD8	05CE'	SOD9	0503'	WARTE
06FB'	WGEF	0720'	WHAN	070F'	WHED
06C1'	WIED	06D8'	WIED1	0728'	WIED2
000E"	WIEDA	000F"	WIEDE	000D"	WIEDH
000C"	WIEDZ	0001"	ZEI		

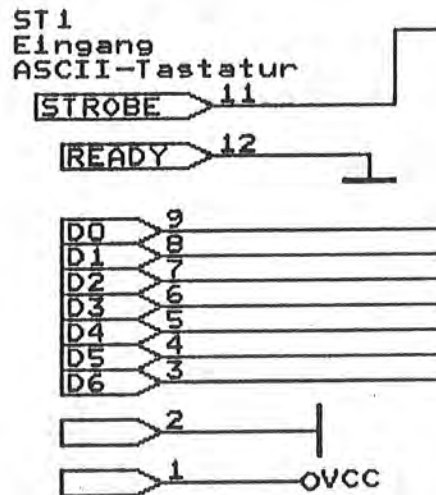
No Fatal error(s)

Änderungen KEY3 r2

Die erste Serie der KEY3 Baugruppe weißt noch kleine Probleme im Bezug auf die ASCII-Tastatur auf. Diese kleinen Fehler können aber problemlos behoben werden

1. Das READY-Signal

Das READY-Signal auf ASCII-Tastaturen wird dazu benutzt um festzustellen, ob der Rechner bereit ist von der Tastatur ein Zeichen zu empfangen. Da dieses Signal bei der KEY nicht verwendet wird (die KEY ist immer bereit) wird dieses Signal fest auf Masse gelegt. Bei älteren Tastaturen liegt dieses Signal oft schon intern auf Masse. In einem solchen Fall muß die KEY3 nicht geändert werden. Sollte dies aber nicht der Fall sein, muß Pin 12 von ST1 auf Masse gelegt werden (Skizze).

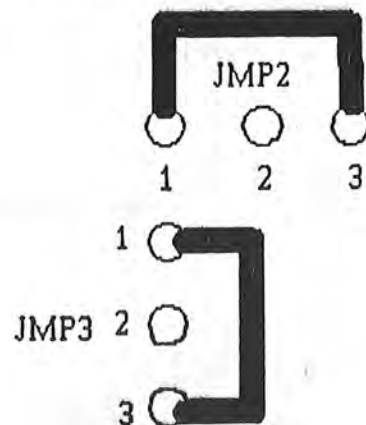


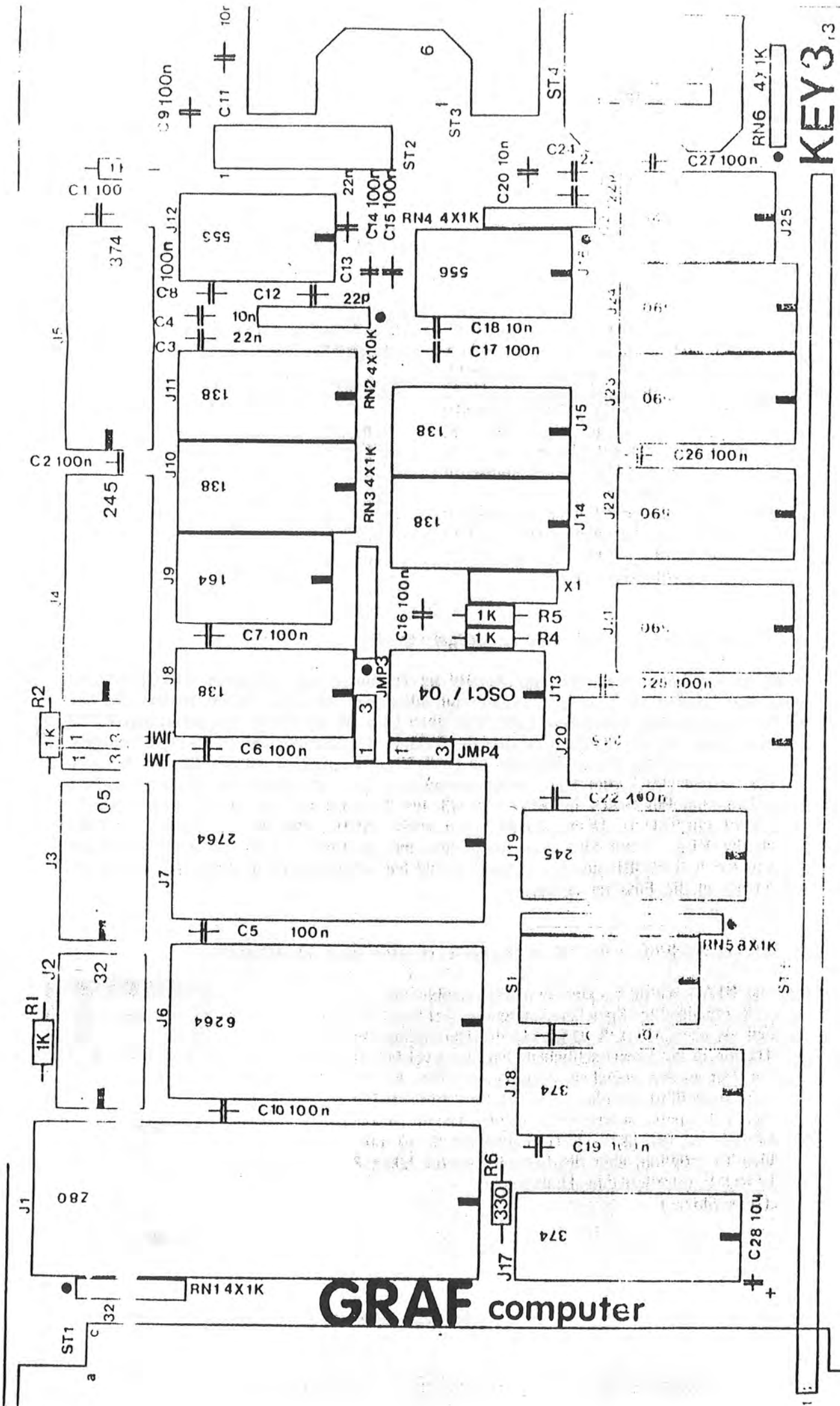
2. Umschaltung positiver und negativer Strobe

Beim NDR-Computer wurden von Anfang an Tastaturen mit positivem STROBE-Signal eingesetzt. Sollten Sie aber eine Tastatur mit negativem STROBE haben, müssen Sie den in der Beschreibung erwähnten JMP1 umlegen. Dies ist allerdings bei der neuen KEY3 nicht möglich, da das OPEN COLLEKTOR Gatter J3C ständig ein LOW-Signal ausgibt, und damit ständig ein Zeichen erkannt wird. Um dies zu umgehen, müssen Sie den Ausgang des Gatters (J3C) vom INT des Prozessors trennen. Am einfachsten ist dies zu erreichen, indem man Pin 6 des J3 hochbiegt. (J3 aus der Fassung nehmen, Pin 6 hochbiegen, J3 wieder einstecken). Damit funktionieren auch ASCII-Tastaturen mit negativem STROBE an der KEY3. Wenn Sie später allerdings eine PC-Tastatur oder eine ASCII-Tastatur mit negativem STROBE anschließen wollen, dürfen Sie allerdings nicht vergessen, diesen Pin wieder in die Fassung zu stecken.

3. Verwendung von 32k x 8 Speicher (RAMs und EPROMs)

Die KEY3 wurde so konzipiert, daß problemlos 8k x 8 organisierte Speicher verwendet werden können als auch 32k x 8. Doch bei der Umsetzung der Theorie in die Praxis schlichen sich hier zwei Fehler ein. Der an den Speichern umschaltbare Pin, der auf Jumper geführt wurde, ist nicht der mittlere Pin dieses Jumpers, sondern der rechte. Damit ist die Einstellung für 8k x 8 (Default-Einstellung) problemlos möglich, aber die Einstellung von 32k x 8 EPROMs erfordert eine Drahtbrücke (siehe Skizze).



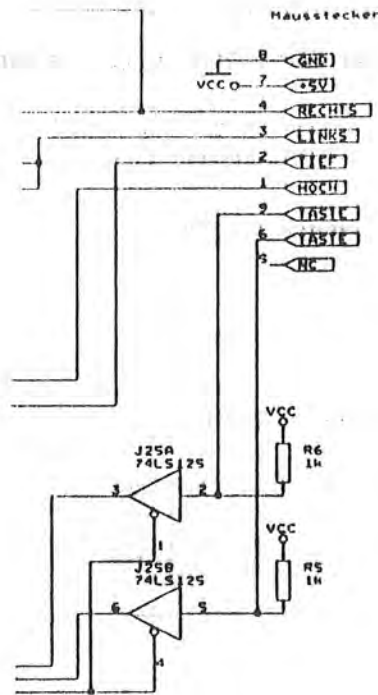


GRAF computer

KEY3

4. Prellen der Tasten bei der Maus

Die KEY3 beinhaltet auch die Maus-Schnittstelle. Sind die Tasten der Maus nicht gedrückt, hängen die Eingänge für die Tasten an der KEY3 in der Luft. Bedingt durch das doch relativ lange Mausekabel und einer eventuell HF-versuchten Umgebung, kann es passieren, daß sich das hier LOW-Signale auftreten, obwohl keine Taste gedrückt wurde. Dieser Effekt kann durch zwei Pull-Up-Widerstände von 1k gegen +5V beseitigt werden (siehe Skizze).



5. Verlängerung des Reset-Impulses

Beim Betrieb der KEY3-Baugruppe ist ab und zu ein kleines Problem aufgetaucht. Beim Einschalten oder bei einem Reset 'kommt die Tastatur manchmal nicht hoch'. Dies liegt daran, daß der Reset-Impuls von der jeweiligen CPU für die KEY3 zu kurz ist.

Falls bei Ihnen dieses Problem auch auftritt, können sie dies folgendermaßen beheben. Sie müssen nur die Zeitkonstante des R-C-Glieds, das den Reset auslöst, verlängern. Dies erreichen Sie am besten, indem Sie auf der CPU-Karte, parallel zu dem Kondensator im Reset-Schwingkreis, nochmal den gleichen Kondensator einlöten.

Bei der CPU68000

Parallel zu C1 (10 μ F) noch einen Kondensator mit 10 μ F einlöten.
Oder statt R7 (100 kOhm) einen 200 kOhm Widerstand einlöten.

Bei der CPU68K

Parallel zu C1 (10 μ F) noch einen Kondensator mit 10 μ F einlöten.
Oder statt R8 (100 kOhm) einen 200 kOhm Widerstand einlöten.

Bei der CPU Z80.

Parallel zu C2 (10 μ F) noch einen Kondensator mit 10 μ F einlöten.
Oder statt R6 (10 kOhm) einen 100 kOhm Widerstand einlöten.

Bei der SBC3

Parallel zu C1 (10 μ F) noch einen Kondensator mit 10 μ F einlöten.
Oder statt R10 (10 kOhm) einen 100 kOhm Widerstand einlöten.

6. Änderungen der Speicherkonfiguration auf der Key3

An der Erklärung der Jumperbelegung bei der Speicherkonfiguration (Eprom und Ram) hatte sich ebenfalls ein Fehler eingeschlichen. Nachfolgend nun die korrekte Anweisung.

Jumperstellungen JMP2

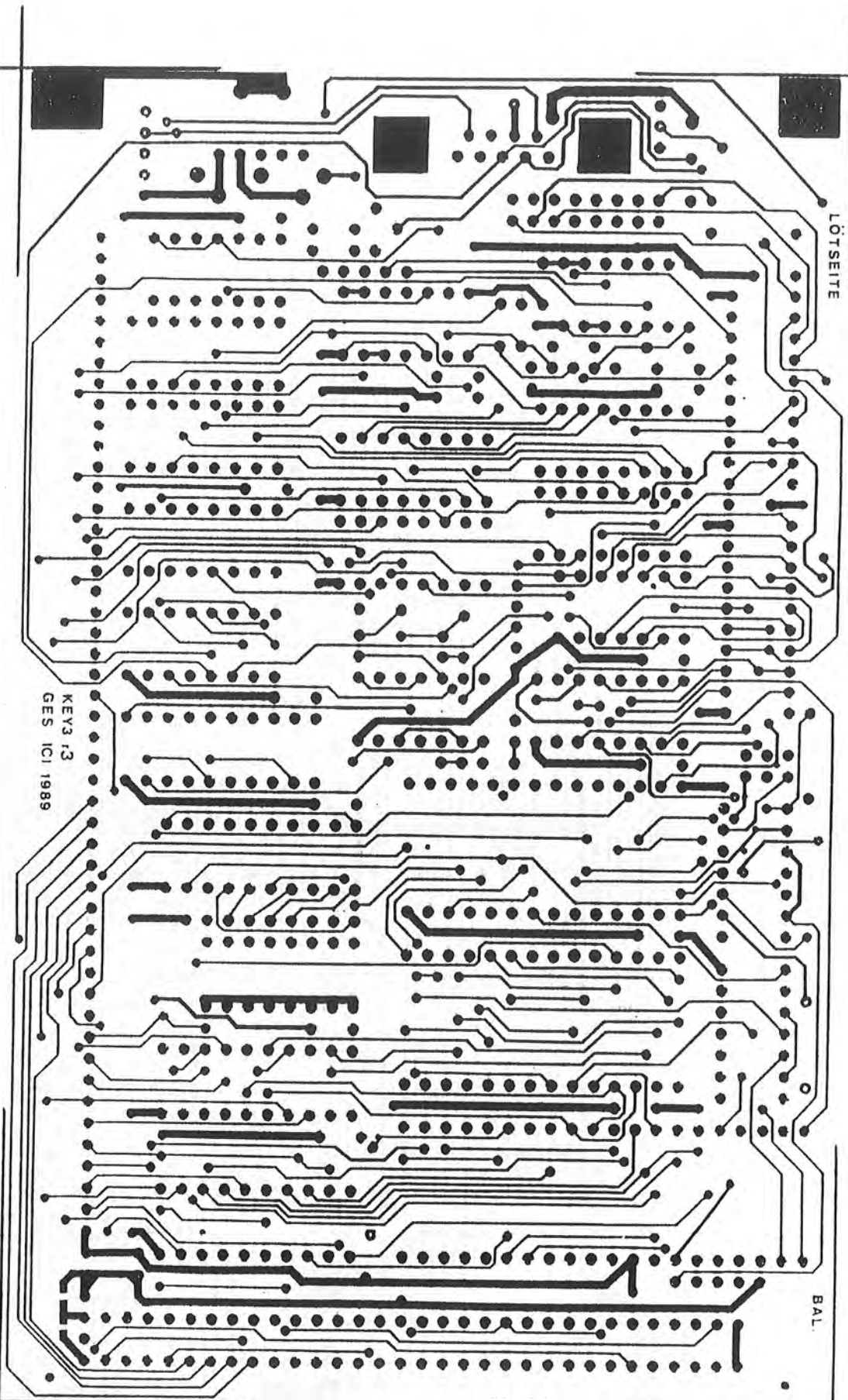
Speicher	Type	Stellung
8K*8	6264	Verbindung 1-2
32K*8	62256	Verbindung 1-3

Jumperstellungen JMP3

Speicher	Type	Stellung
8K*8	2764	Verbindung 2-3
16K*8	27128	Verbindung 1-3
32K*8	27256	Verbindung 1-3

7. CLK des Z80

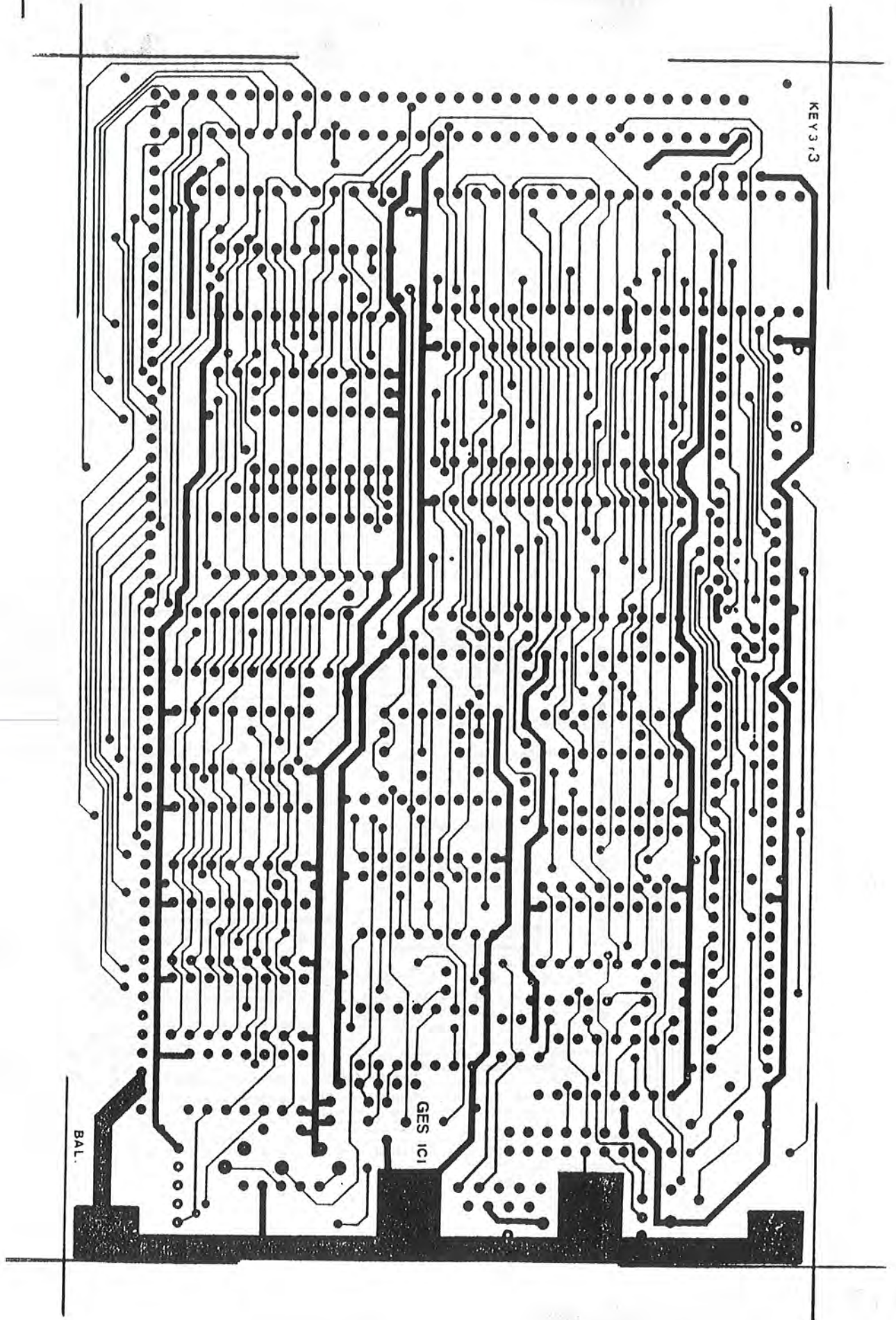
Der Oszillator mit der Frequenz von 4 MHz schwingt nicht immer sicher an. Um dies zu vermeiden, muß unbedingt ein 74LS04 auf dem Platz von J13 stecken. Um eine stabile Funktion des Z80 zu gewährleisten, muß über dem CLK des Z80 ein 330 Ohm Widerstand gegen +5V hängen. Dieser Widerstand kann sehr einfach von Pin 6 des Z80 zum Pin 11 des Z80 geführt werden.



LÖTSEITE

KEY3 P3
GES ICI 1989

BAL



KEY 3 13

GES 1C1

BAL.

