

ITT / RFA

ITT 3030

SERVICE MANUAL

MK 3886 COMBO

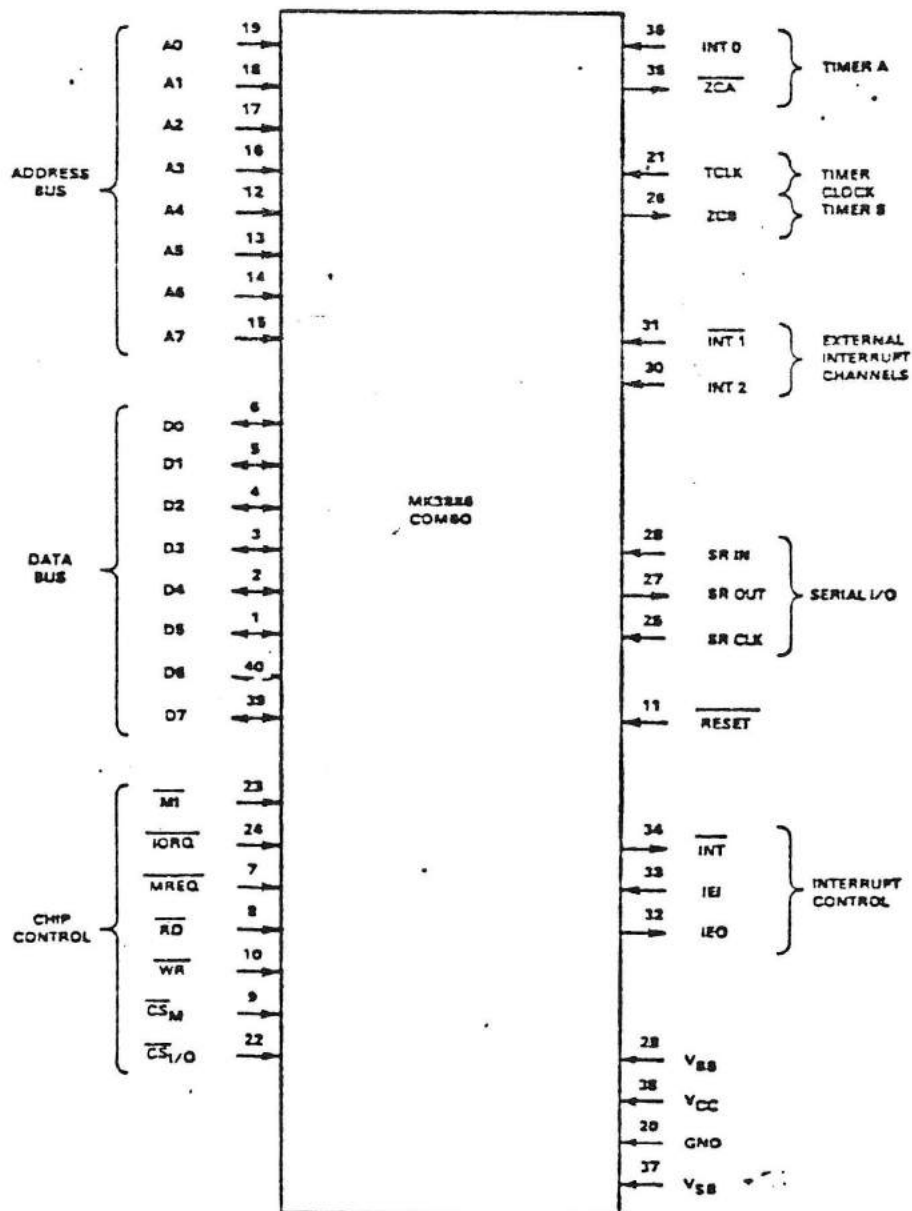
1.0 INTRODUCTION

1.1 The MK3886 Combo chip is a Z80 microprocessor peripheral containing a combination of features that enables a user to have great flexibility with a single component. The chip contains 256 bytes of RAM, two timers, a serial Input/Output port and three external interrupt inputs. Additionally, the interrupt vector and priority circuitry can be software programmable to configure the chip for special user requirements. The Combo chip utilizes N-channel silicon gate depletion load technology and is packaged in a 40-pin DIP. Major features of the MK3886 are:

- 256 x 8 static RAM - 64 bytes of which can operate in write protected or low power standby mode
- Two programmable timers which operate from an independent clock source
- Three external interrupt channels with programmable vector for each channel
- Serial I/O port - synchronous or asynchronous operation with end of word interrupt
- Z80 compatible daisy chain interrupt structure
- Single 5 volt supply ($\pm 5\%$)

MK3886 PIN CONFIGURATION

Figure 3.0-1

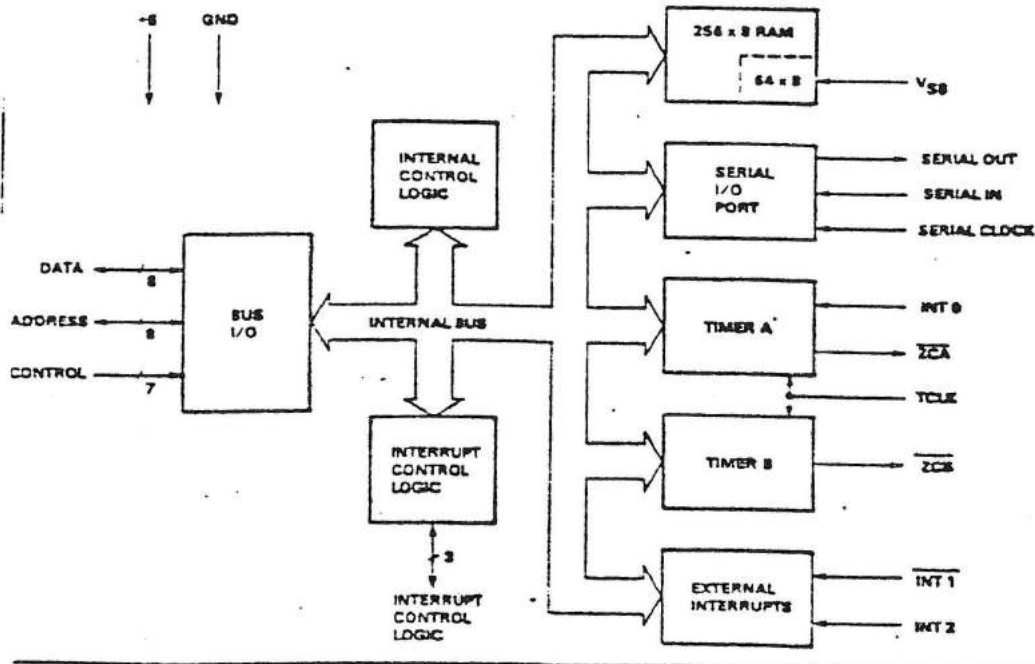


2.0 MK3886 ARCHITECTURE

2.1 A block diagram of the MK3886 Combo chip is shown in Figure 2.0-1. The internal structure of the Combo chip consists of a latched CPU bus interface, internal control logic, 256 bytes of random access memory, serial I/O port, two timers, and three external interrupt channels. This part is not restricted for use only in Z80 CPU systems but can easily be adapted to other bus oriented systems.

MK3886 BLOCK DIAGRAM

Figure 2.0-1



The internal control logic of the Combo Chip receives and decodes the control sequences to be performed from the CPU bus. Three types of access cycles may occur: memory read/write, I/O read/write, and interrupt acknowledge. None of these operations are dependent upon the timer clock, TCLK, but rather on timing conditions present on the control input lines. The TCLK input is only used to drive the two timers. The control logic and I/O ports can be addressed via ten registers. These internal registers serve to configure the chip for proper operation and provide a means for exchange of control and data information between the CPU and Combo chip.

Four individual microcomputer component features can be identified within the Combo Chip. These features are Read/Write memory, counting and timing channels, serial input/output, and external interrupt inputs. These combinations enable the MK3886 to function as a key element in a minimum component Z80 system.

The memory consists of 256 bytes of static RAM. The lower 64 bytes have two special features: Write protection and standby power. The write protection protects this memory area from undesired write operations. The low power standby RAM provides a method to preserve important data during a loss of system power.

Two versatile software programmable timers are provided. The Timer clock (TCLK) input is used by both timers to provide an accurate time base. In addition to the zero count output on both timers, Timer A has an external interrupt input linked directly to its control circuitry which provides two additional timer modes.

The serial port allows input and output of serial data in either asynchronous or synchronous modes. The port is basically a 16 bit shift register that can be read from or written to while the data is being shifted at a rate determined by the external serial clock. This port can be used to provide serial data communications or to interface to external serial logic such as shift registers or serial memories (CCD).

Three external interrupt lines are provided in order to allow prioritized, vectored, maskable, edge triggered external interrupt inputs. All interrupt lines are TTL compatible with Schmitt trigger buffered input circuits.

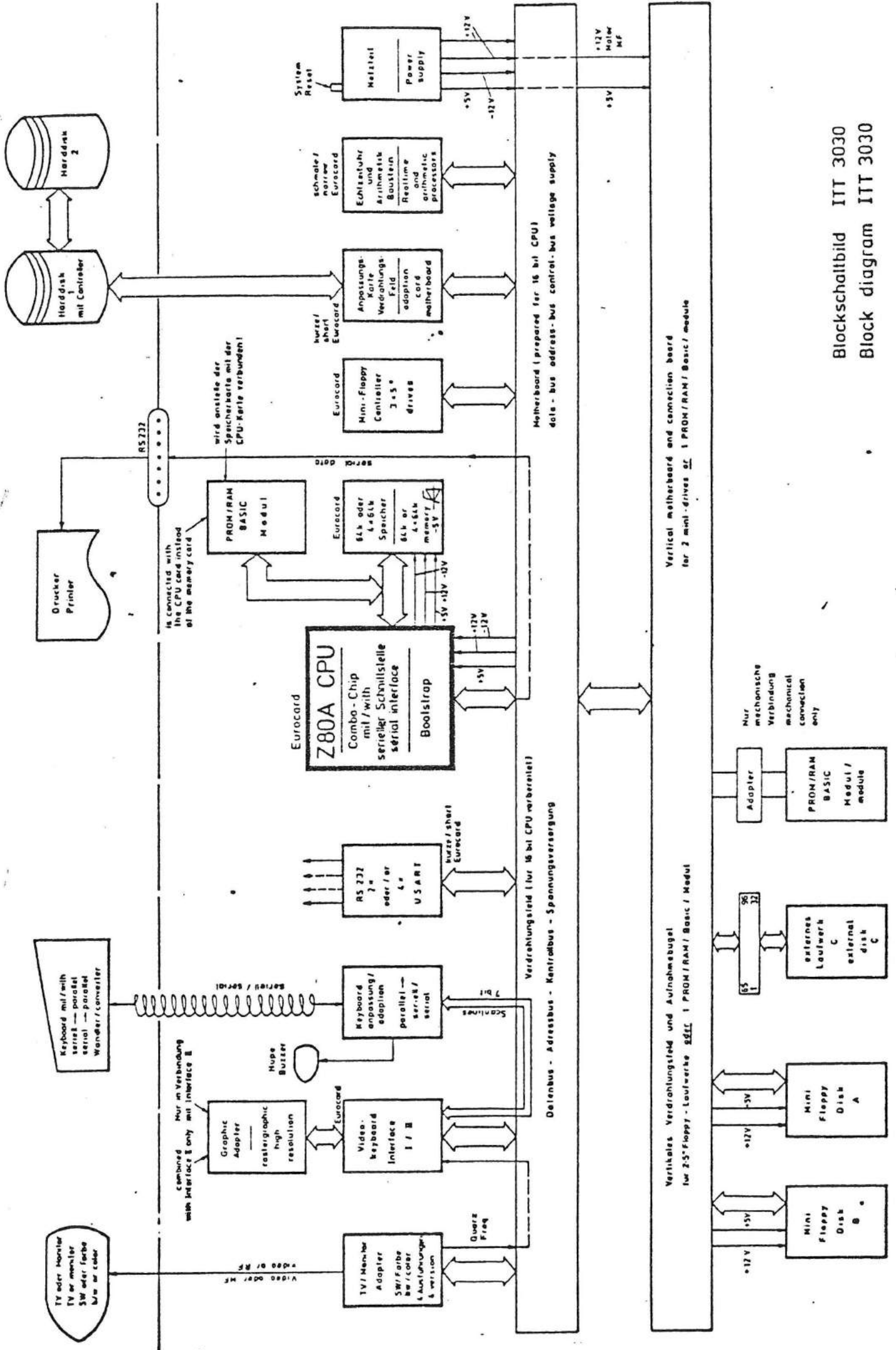
The interrupt control logic section handles all CPU interrupt protocol for nested priority interrupt structures. Priority is determined in two ways. First, an internal priority has been assigned for the 7 channels capable of generating an interrupt. This priority is listed in section 7. Secondly, priority of any component device is determined by its physical location in a daisy chain configuration. Two lines are provided in each Combo chip to form this daisy chain with the device closest to the CPU having the highest priority.

The MK3886 requires only a single +5 volt supply. However a second input, V_{SB}, is provided to supply power to the low power standby RAM. This supply input can be tied either to a battery back up supply or simply to V_{CC} if low power standby is not required.

2 JUL 1982

VORABVERSION

SYSTEMHANDBUCH

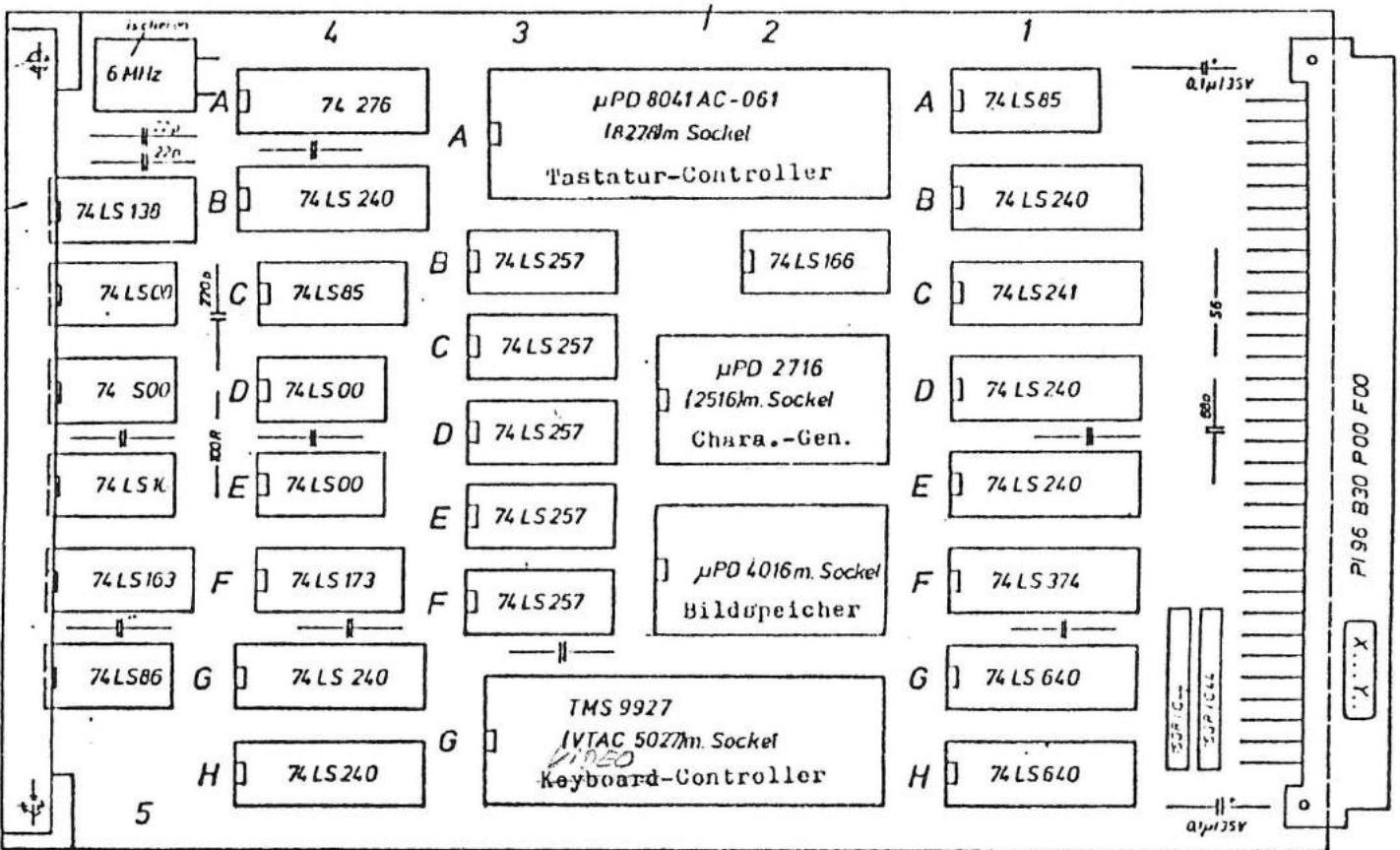


Blockschaltbild IIT 3030
Block diagram IIT 3030

TTT 3030

auf Wachstum programmiert

Video/Keyboard-Controller

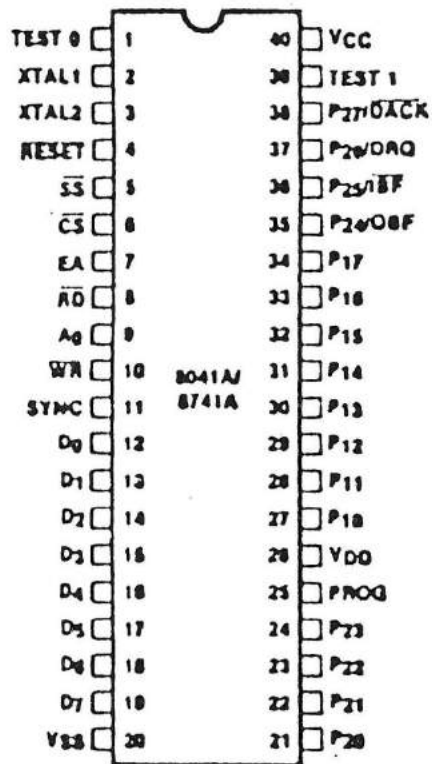


Technik der Welt

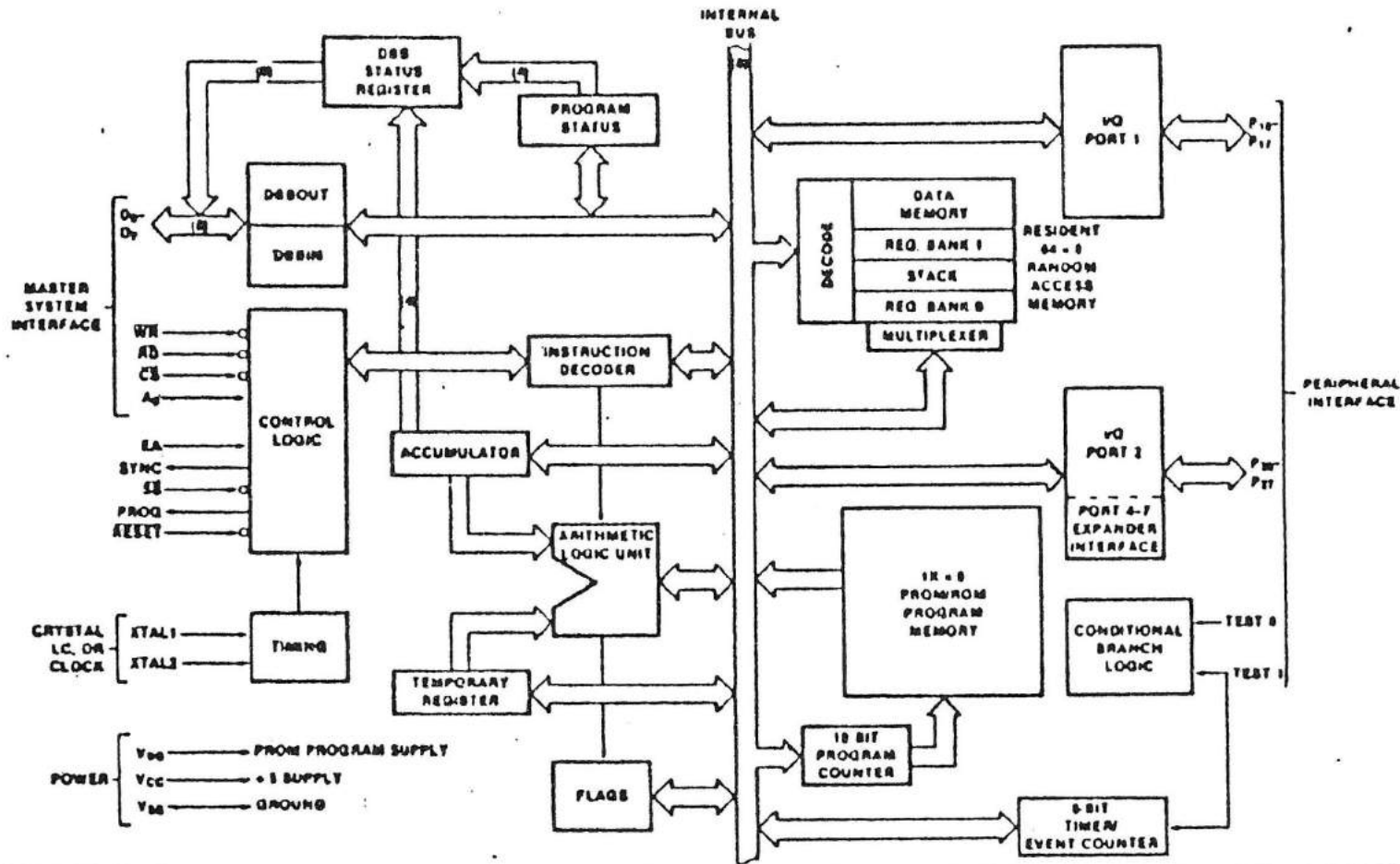


KEYBOARD - CONTROLLER 8741 A

PIN CONFIGURATION



BLOCK DIAGRAM



1. CRT - Controller

1.1 Allgemeines

Der CRT-Controller ist aufgeteilt in die Funktionsgruppen:

- Erkennung Bildspeicher
- Adreßerkennung und Adreßauswahl
- Adreßmodifizierung
- programmierbare Zeitsteuerung
- Bildwiederholpeicher mit Ausgabelatch
- Zeichengenerator
- Videosignalerzeugung
- Taktgenerator

- Erkennung Bildspeicher

Bei Betrieb mit dem 64K-Speicher wird mit dem Signal SR/\overline{KOMBI} zwischen 64K-Speicher und Bildspeicher umgeschaltet.

- Adreßerkennung und Adreßauswahl

Der Bildspeicher verhält sich zur BUS-Schnittstelle wie ein 4K-RAM-Speicher. Die oberen 4 Bit der auf der $3000H$ liegenden Basisadressen generieren zusammen mit MEMR oder MEMW die Signale WEM oder REM sowie EXT.

EXT schaltet die 3 Adreßmultiplexer von interner Refreshadresse auf externe Rechneradresse um, sollte eine Anforderung auf externen Zugriff vorliegen. Die unteren 12 Bit sind im Bildwiederholpeicher so aufgeteilt, daß jedes auf dem Bildschirm dargestellte Zeichen Teil eines x-y-Koordinatensystems ist. Dabei bilden die Adreßbit $A_0 - A_6$ die Zeichenposition innerhalb der Zeile, die Adreßbit $A_7 - A_{11}$ die Zeilennummer.

Das Lupensignal wird auf der Adresse $34H$ mit dem Datenbit 0 übergeben.

- Adreßmodifizierung.


Über einen MUX werden die über den BUS kommenden 12 Zeichenadreßbit zu einer 11 Bit-Adresse reduziert, um so den Speicherplatz dem tatsächlich benötigten Bedarf anzupassen.

Die Zuordnung zwischen BUS-Adresse und der sich daraus ergebenden internen Speicheradresse zeigt die folgende Tabelle:

Adresse 32 steuert eine externe Hupe

Adresse 33 steuert eine Anzeige-LED in der Tastatur

BUS-Adressen						
Zusatz- Basis- adresse	Zeilen Nr.	Zeichen				
		00 - 3F	40 - 4F	50 - 5F	60 - 6F	70 - 7F
		Zeichen 1 - 64	Zeichen 65 - 80	verbotener Zeichenadress- Bereich		
0 0 0	1	000 - 03F	600 - 60F	600 - 60F	600 - 60F	600 - 60F
0 8 0	2	040 - 07F	640 - 64F	640 - 64F	640 - 64F	640 - 64F
...
7 8 0	16	3C0 - 3FF	7D0 - 7DF	7D0 - 7DF	7D0 - 7DF	Interrupt Request
8 0 0	17	400 - 43F	620 - 62F	620 - 62F	620 - 62F	620 - 62F
8 8 0	18	480 - 47F	660 - 66F	660 - 66F	660 - 66F	660 - 66F
...
B 8 0	24	5C0 - 5FF	7E0 - 7EF	7E0 - 7EF	7E0 - 7EF	7E0 - 7EF
C 0 0	25	600 - 63F	630 - 63F	630 - 63F	630 - 63F	630 - 63F
...
F 8 0	32	7C0 - 7FF	7F0 - 7FF	7F0 - 7FF	7F0 - 7FF	Zeitablauf- Steuerung

-  Speicherbereich 1 K

- Bei den Steueradressen "Interrupt Request" und Zeitablaufsteuerung tritt kein Speicher-Chip-Select auf, so daß eine Doppelbeschreibung des Speichers vermieden wird.

- Der Speicherbereich für die Zeichen 65-80 in Zeile 1-24 und Zeichen 1-64 in Zeile 25-32 ist identisch, so daß die größtmöglichen Formate 32 Zeilen - 64 Zeichen bzw. 24 Zeilen - 80 Zeichen sind.

Der Interrupt-Request dient zur Synchronisierung des Rechnerzugriffes auf die vertikale Austastlücke, so daß der externe Zugriff während der Dunkeltastung des Bildschirms erfolgt. Dadurch wird eine Störung des Bildes beim Rechnerzugriff vermieden.

Programmierbare Zeitsteuerung

Die programmierbare Zeitsteuerung erfolgt durch den Controller Chip CRT 5027. Seine Steuerregister werden durch die unteren 4 Adreßbits adressiert und über den Datenbus geladen. Dazu wird auf der Karte, außer dem Chip Select auch ein Data-Strobe erzeugt. Der Controller Chip erzeugt dann über die Ausgänge $H_0 - H_6$ (Zeichenposition) und $DR_0 - DR_4$ (Zeilennummer) die interne Refresh-Adresse für den Bildwiederholungspeicher und über $R_0 - R_3$ (Row Scan) die aktuelle Rasterzeilennummer für den Zeichengenerator.

Der Baustein erzeugt außerdem den horizontalen und vertikalen Synchronisationsimpuls, sowie ein Gemisch dieser Impulse für das BAS-Signal. Ferner liefert der Baustein das Cursor-Signal und das Blank-Signal für die Austastlücken.

- Video-Signal-Erzeugung

Die serielle Videoinformation des Schieberegisters wird mit dem aus Synchronisierungsgründen zwischengespeicherten Invers- bzw. Cursorbit verknüpft, das ebenso wie das Blanksignal (Horizontale, vertikale Austastlücke) vom Controller Chip erzeugt wird.

- Taktgenerator

Die Signale DC und DCC werden von einem Quarzgenerator bzw. von dem nachgeschalteten Synchronzähler geliefert. Dieser wird mit dem Videodot Clock getaktet und erzeugt bei Überlauf das Signal DCC.

Mit der nächsten Taktflanke setzt er sich auf den Anfangszustand 8.

- Bildwiederholpeicher mit Ausgabelatch

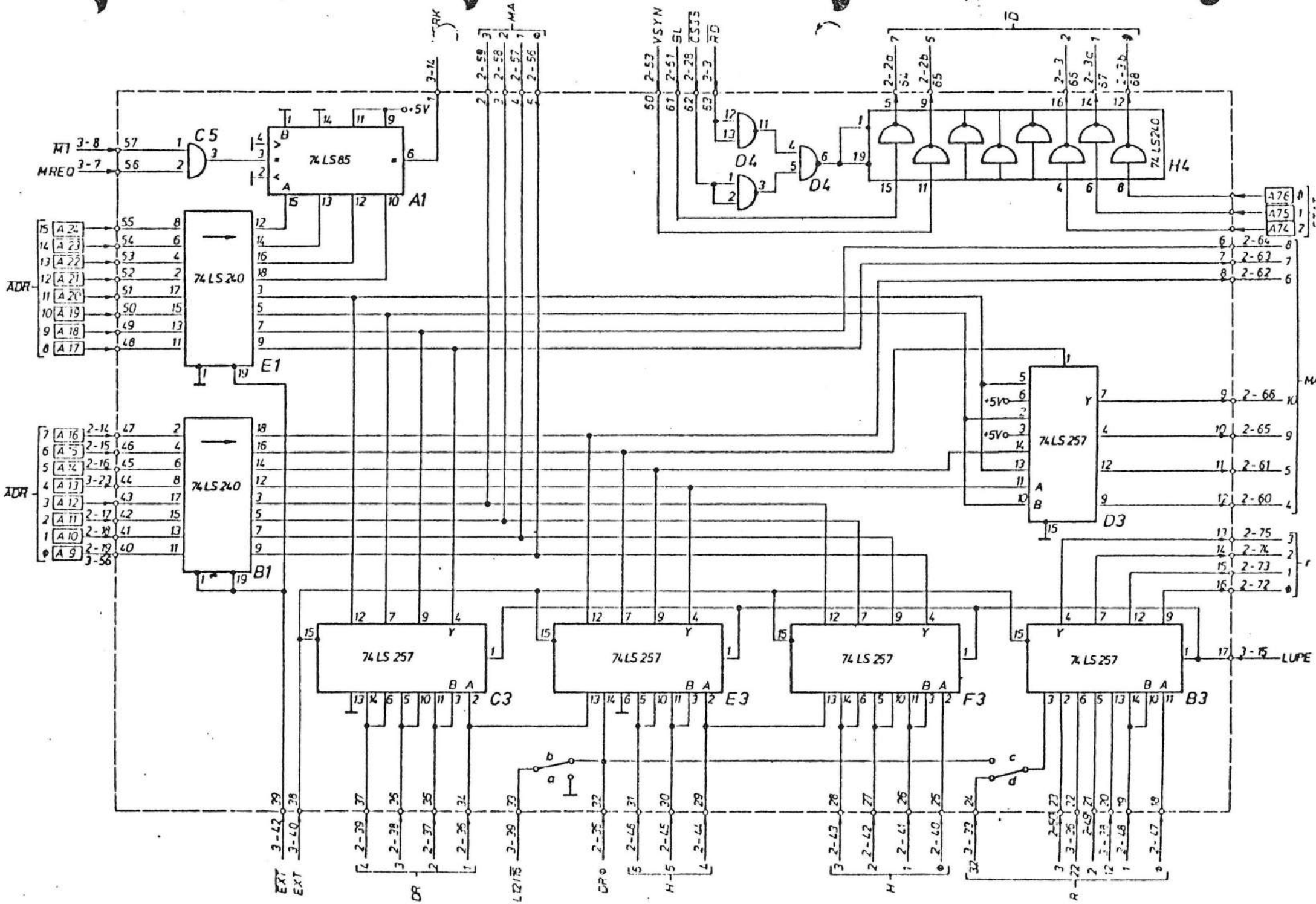
Die CRT-Controllerplatine benötigt je nach gewünschtem Darstellungsformat (16 x 32, 16 x 64, 24 x 80, 12x40 Zeilen x Zeichen) einen 1 K bzw. 2 K großen Bildwiederholpeicher, der durch die Multiplexer der Adressenauswahl adressiert wird.

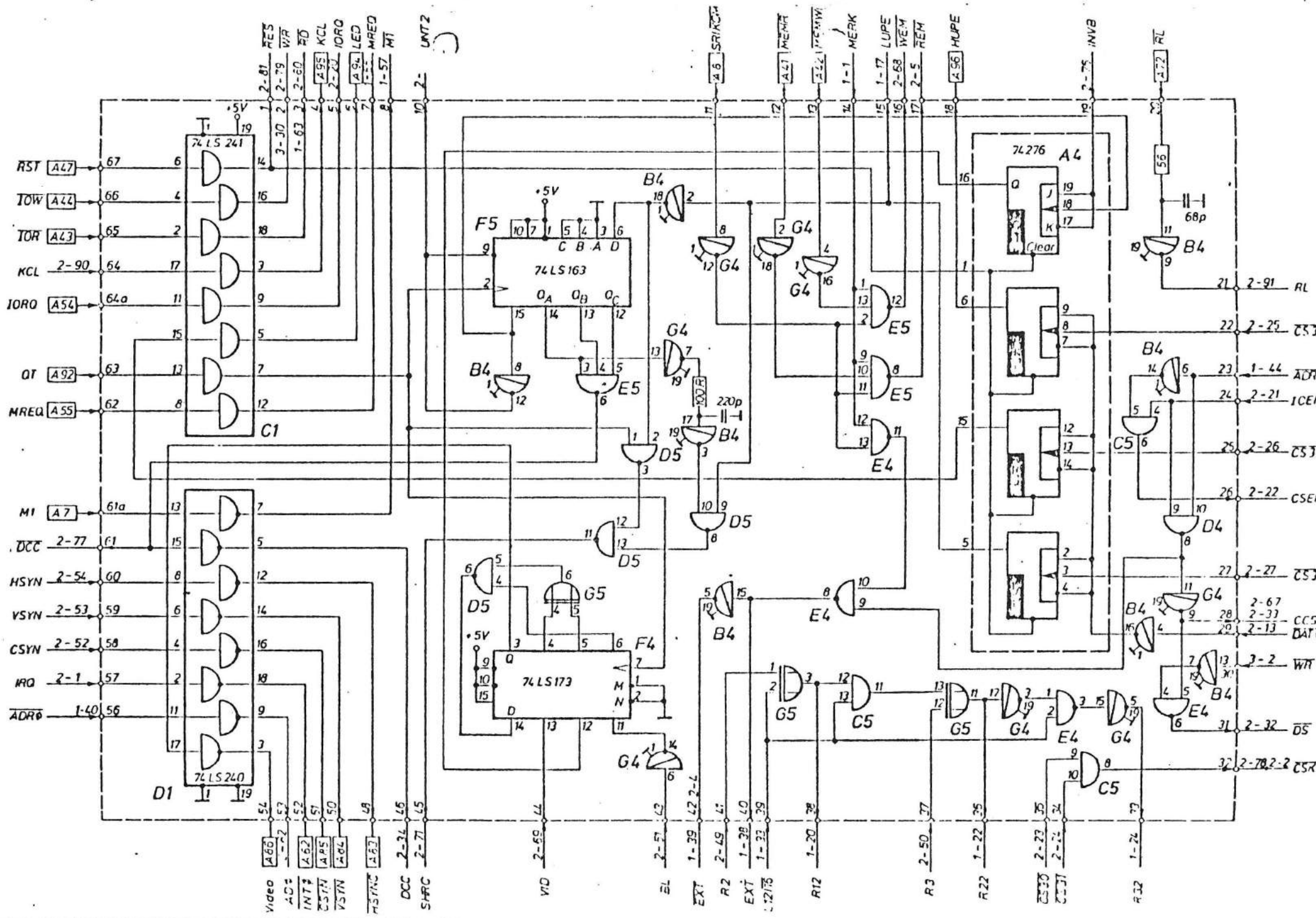
Die 8-Bit Datenausgänge dieses Speichers werden für einen Zeichenzyklus mit dem Signal DCC zwischengespeichert. Die unteren 7-Bit entsprechen dem ASCII-Code, während das 8. Bit zu einer zeichenbegleitenden Inversdarstellung benutzt wird.

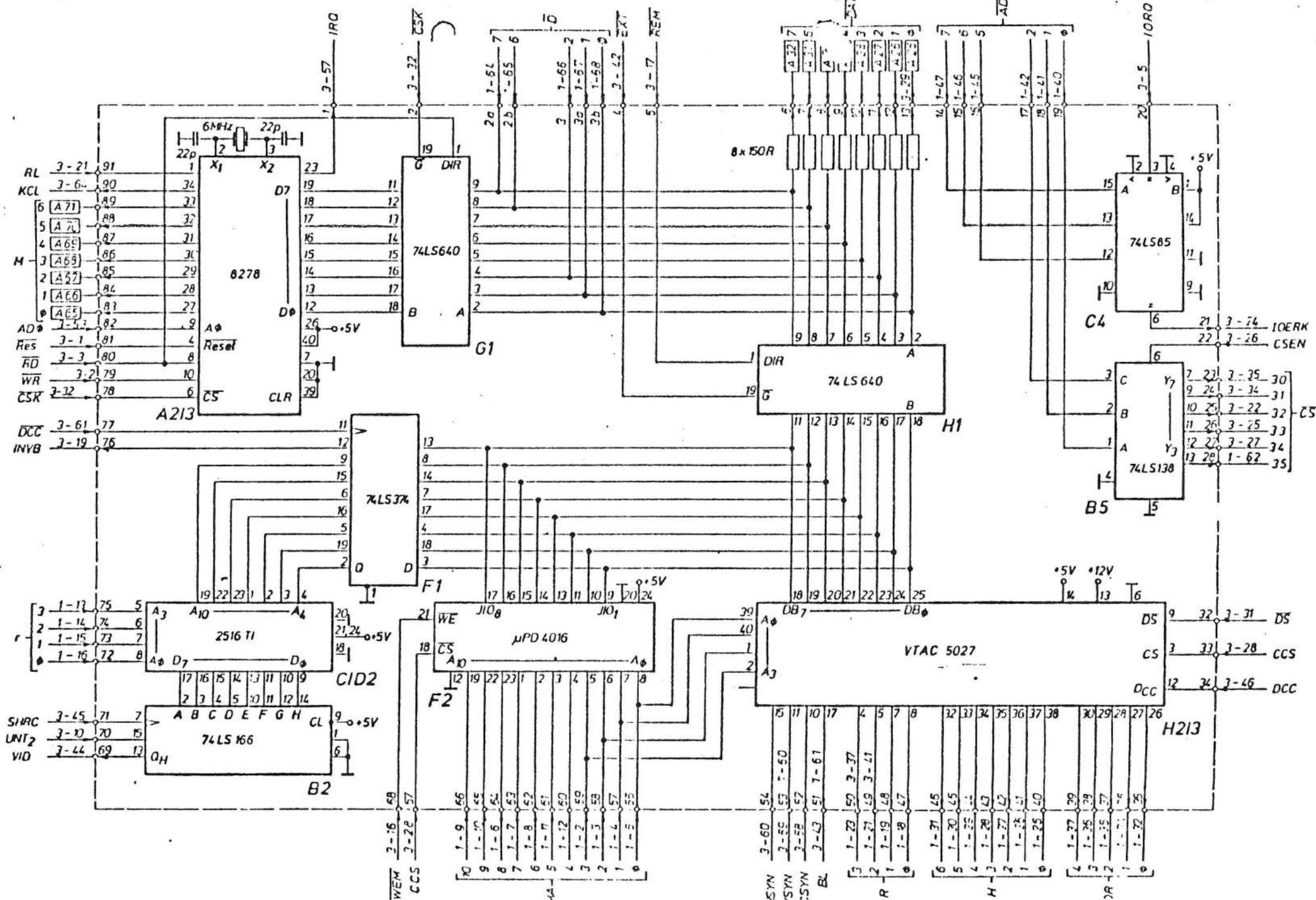
- Zeichengenerator

Der ASCII-Code des Bildwiederholpeichers bildet zusammen mit den 4 Bit ($R_7 - R_3$) des Controller Chip die Adresse für den Zeichengenerator (2 K Byte Prom mit max. 128 Zeichen).

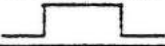
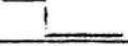
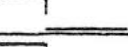
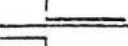
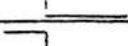
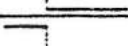


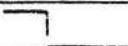
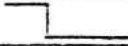
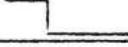
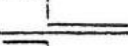
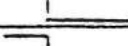
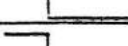
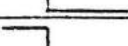
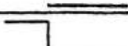





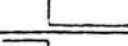
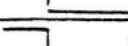



Mit dem Takt DCC werden die Daten dieses PROM'S parallel in ein Schieberegister übernommen und mit dem Takt DC in eine serielle Form gebracht.

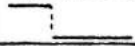
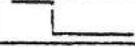
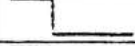
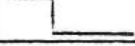

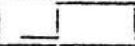
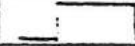



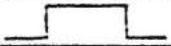
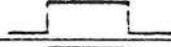
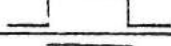
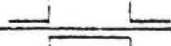

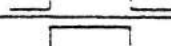


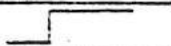
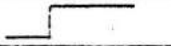
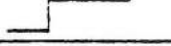

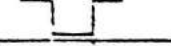
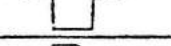
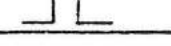





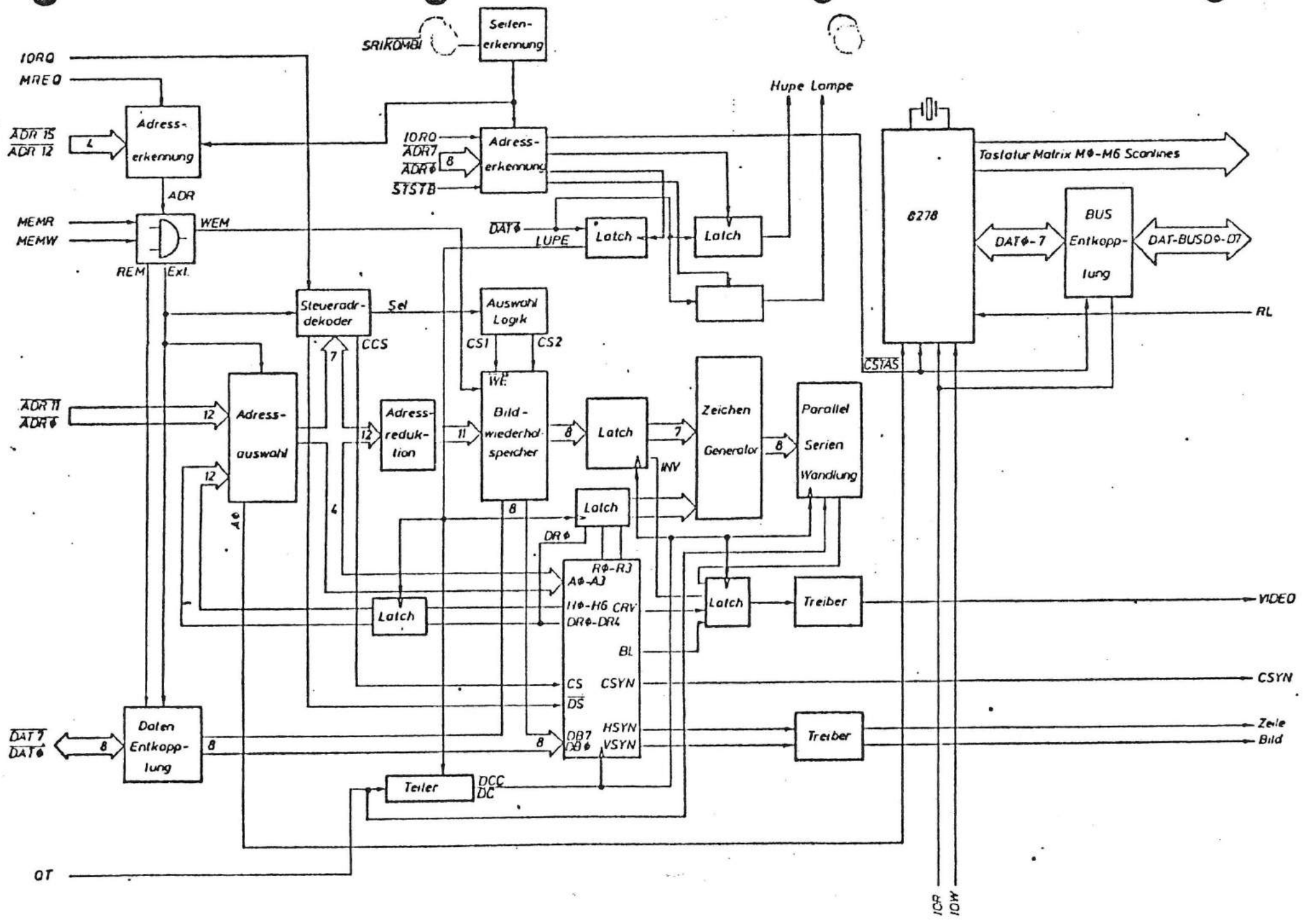


STECKERBELEGUNGSLISTE

PIN	Bezeichnung	Diagramm	Erläuterungen	Blatt Nr.
01	+ 5V			
02	0V			
03	+ 12V			
04	0V			
05	- 12V			
06	0V			
07	M1		Machine Cycle One	
08	SR/KOMBI		Freigabe Speicher/Interface	
09	ADR 0		Adreß-Bit 0	
10	ADR 1		Adreß-Bit 1	
11	ADR 2		Adreß-Bit 2	
12	ADR 3		Adreß-Bit 3	
13	ADR 4		Adreß-Bit 4	
14	ADR 5		Adreß-Bit 5	
15	ADR 6		Adreß-Bit 6	
16	ADR 7		Adreß-Bit 7	
17	ADR 8		Adreß-Bit 8	
18	ADR 9		Adreß-Bit 9	
19	ADR 10		Adreß-Bit 10	
20	ADR 11		Adreß-Bit 11	
21	ADR 12		Adreß-Bit 12	
22	ADR 13		Adreß-Bit 13	
23	ADR 14		Adreß-Bit 14	
24	ADR 15		Adreß-Bit 15	
25	DAT 0		Daten-Bit 0	
26	DAT 1		Daten-Bit 1	
27	DAT 2		Daten-Bit 2	
28	DAT 3		Daten-Bit 3	
29	DAT 4		Daten-Bit 4	
30	DAT 5		Daten-Bit 5	
31	DAT 6		Daten-Bit 6	
32	DAT 7		Daten-Bit 7	

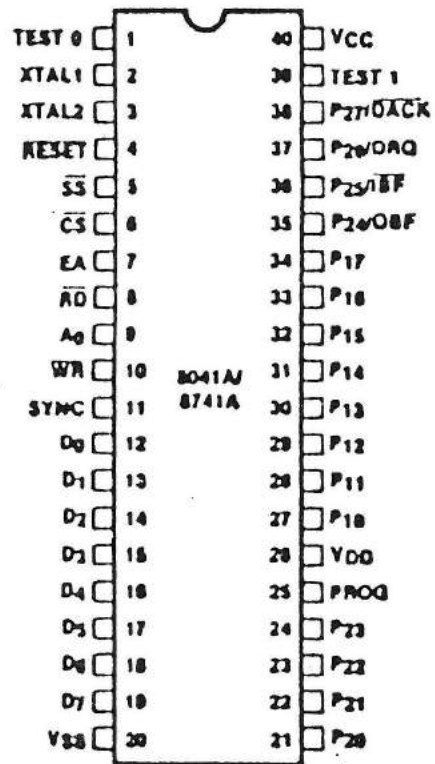
PIN	Bezeichnung	Diagramm	Erläuterungen	Blatt Nr.
33	+ 5V			
34	0V			
35	+ 12V			
36	0V			
37	- 12V			
38	0V			
39				
40				
41	$\overline{\text{MEMR}}$		Memory Read	
42	$\overline{\text{MEMW}}$		Memory Write	
43	$\overline{\text{IOR}}$		IN/OUT Read	
44	$\overline{\text{IOW}}$		IN/OUT Write	
45				
46				
47	$\overline{\text{RST}}$		Reset intern	
48				
49				
50				
51				
52				
53				
54	IORQ		Input/Output Request	
55	MREQ		Memory Request	
56				
57				
58				
59				
60				
61				
62	$\text{INT } \emptyset$		External Interrupt \emptyset	
63				
64				

PIN	Bezeichnung	Diagramm	Erläuterungen	Blatt Nr.
65	M \emptyset		Matrix Scan Ausgang \emptyset	
66	M 1		Matrix Scan Ausgang 1	
67	M 2		Matrix Scan Ausgang 2	
68	M 3		Matrix Scan Ausgang 3	
69	M 4		Matrix Scan Ausgang 4	
70	M 5		Matrix Scan Ausgang 5	
71	M 6		Matrix Scan Ausgang 6	
72	RL		Tastatur Rückleitung	
73				
74	STAT 2		Statussignal 2	
75	STAT 1		Statussignal 1	
75	STAT \emptyset		Statussignal \emptyset	
77				
78				
79				
80				
81				
82				
83	HSYN		Zeilenimpuls	
84	VSYN		Bildimpuls	
85	CSYN		Composite Synchronisier Signal	
86	Video		Video Signal	
87				
88				
89				
90				
91				
92	QT		Quarztakt	
93				
94	LED		Lampenansteuerung	
95	KCL		Key Clock	
96	Hupe		Hupe	

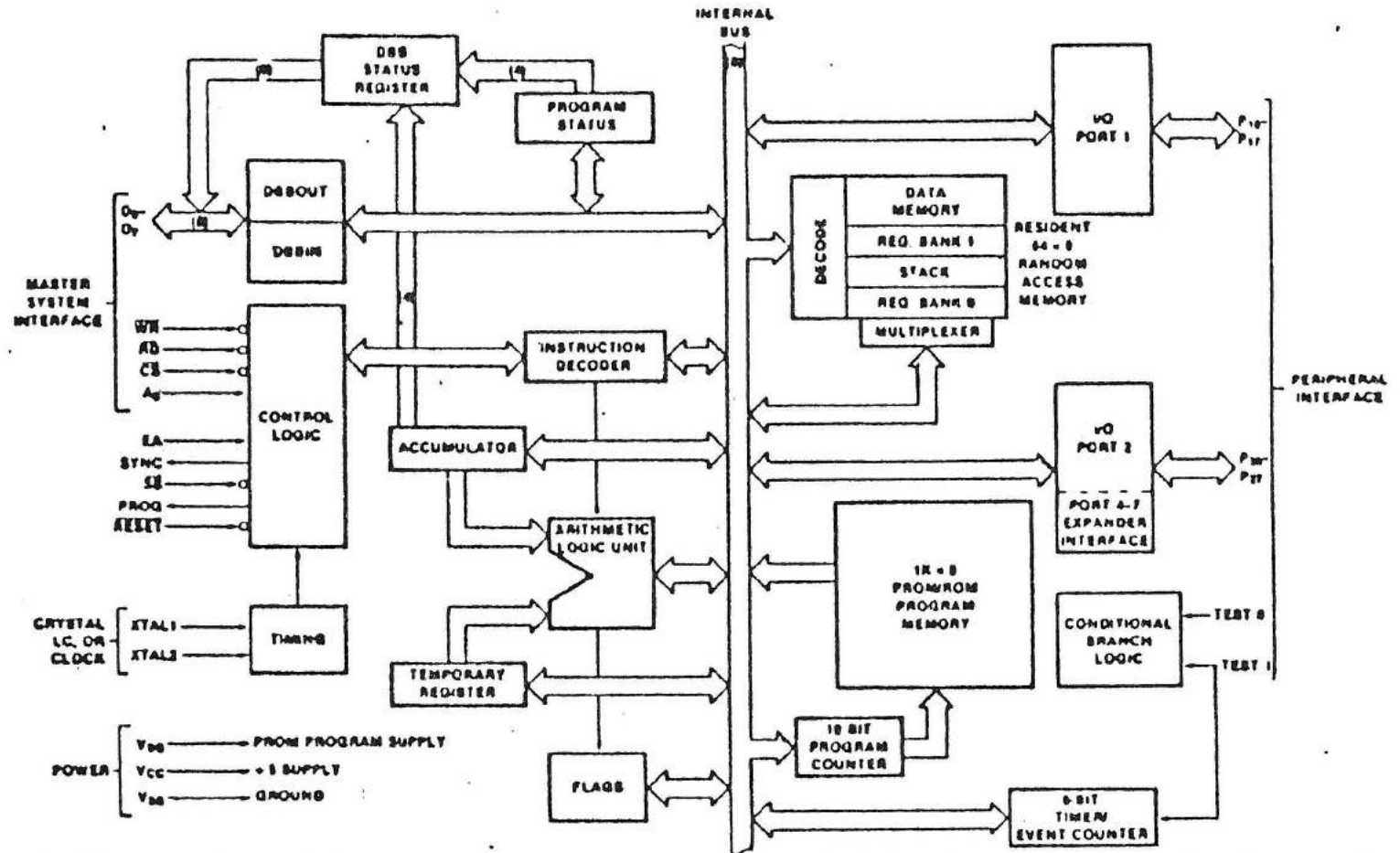


KEYBOARD - CONTROLLER 8741 A

PIN CONFIGURATION



BLOCK DIAGRAM



1. CRT - Controller

1.1 Allgemeines

Der CRT-Controller ist aufgeteilt in die Funktionsgruppen:

- Erkennung Bildspeicher
- Adreßerkennung und Adreßauswahl
- Adreßmodifizierung
- programmierbare Zeitsteuerung
- Bildwiederholungspeicher mit Ausgabelatch
- Zeichengenerator
- Videosignalerzeugung
- Taktgenerator

- Erkennung Bildspeicher

Bei Betrieb mit dem 64K-Speicher wird mit dem Signal $\overline{\text{SR/KOMBI}}$ zwischen 64K-Speicher und Bildspeicher umgeschaltet.

- Adreßerkennung und Adreßauswahl

Der Bildspeicher verhält sich zur BUS-Schnittstelle wie ein 4K-RAM-Speicher. Die oberen 4 Bit der auf der 3000H liegenden Basisadressen generieren zusammen mit MEMR oder MEMW die Signale WEM oder REM sowie EXT.

EXT schaltet die 3 Adreßmultiplexer von interner Refreshadresse auf externe Rechneradresse um, sollte eine Anforderung auf externen Zugriff vorliegen. Die unteren 12 Bit sind im Bildwiederholungspeicher so aufgeteilt, daß jedes auf dem Bildschirm dargestellte Zeichen Teil eines x-y-Koordinatensystems ist. Dabei bilden die Adreßbit $A_0 - A_6$ die Zeichenposition innerhalb der Zeile, die Adreßbit $A_7 - A_{11}$ die Zeilennummer.

Das Lupensignal wird auf der Adresse 34H mit dem Datenbit 0 übergeben.

- Adreßmodifizierung


Über einen MUX werden die über den BUS kommenden 12 Zeichenadreibit zu einer 11 Bit-Adresse reduziert, um so den Speicherplatz dem tatsächlich benötigten Bedarf anzupassen.

Die Zuordnung zwischen BUS-Adresse und der sich daraus ergebenden internen Speicheradresse zeigt die folgende Tabelle:

Adresse 32 steuert eine externe Hupe

Adresse 33 steuert eine Anzeige-LED in der Tastatur

BUS-Adressen						
Basis- adresse	Zusatz- adresse	00 - 3F	40 - 4F	50 - 5F	60 - 6F	70 - 7F
	Zeilen Nr.	Zeichen 1 - 64	Zeichen 65 - 80	verbotener Zeichenadress- Bereich		
0 0 0	1	000 - 03F	600 - 60F	600 - 60F	600 - 60F	600 - 60F
0 8 0	2	040 - 07F	640 - 64F	640 - 64F	640 - 64F	640 - 64F
...
7 8 0	16	3C0 - 3FF	7D0 - 7DF	7D0 - 7DF	7D0 - 7DF	Interrupt Request
8 0 0	17	400 - 43F	620 - 62F	620 - 62F	620 - 62F	620 - 62F
8 8 0	18	480 - 47F	660 - 66F	660 - 66F	660 - 66F	660 - 66F
...
B 8 0	24	5C0 - 5FF	7E0 - 7EF	7E0 - 7EF	7E0 - 7EF	7E0 - 7EF
C 0 0	25	600 - 63F	630 - 63F	630 - 63F	630 - 63F	630 - 63F
...
F 8 0	32	7C0 - 7FF	7F0 - 7FF	7F0 - 7FF	7F0 - 7FF	Zeitablauf- Steuerung

-  Speicherbereich 1 K
- Bei den Steueradressen "Interrupt Request" und Zeitablaufsteuerung tritt kein Speicher-Chip-Select auf, so daß eine Doppelbeschreibung des Speichers vermieden wird.
- Der Speicherbereich für die Zeichen 65-80 in Zeile 1-24 und Zeichen 1-64 in Zeile 25-32 ist identisch, da daß die größtmöglichen Formate 32 Zeilen - 64 Zeichen bzw. 24 Zeilen - 80 Zeichen sind.

Der Interrupt-Request dient zur Synchronisierung des Rechnerzugriffes auf die vertikale Austastlücke, so daß der externe Zugriff während der Dunkelastung des Bildschirmes erfolgt. Dadurch wird eine Störung des Bildes beim Rechnerzugriff vermieden.

Programmierbare Zeitsteuerung

Die programmierbare Zeitsteuerung erfolgt durch den Controller Chip CRT 5027. Seine Steuerregister werden durch die unteren 4 Adreßbits adressiert und über den Datenbus geladen. Dazu wird auf der Karte, außer dem Chip Select auch ein Data-Strobe erzeugt. Der Controller Chip erzeugt dann über die Ausgänge $H_0 - H_6$ (Zeichenposition) und $DR_0 - DR_4$ (Zeilennummer) die interne Refresh-Adresse für den Bildwiederholungspeicher und über $R_0 - R_3$ (Row Scan) die aktuelle Rasterzeilennummer für den Zeichengenerator.

Der Baustein erzeugt außerdem den horizontalen und vertikalen Synchronisationsimpuls, sowie ein Gemisch dieser Impulse für das BAS-Signal. Ferner liefert der Baustein das Cursor-Signal und das Blank-Signal für die Austastlücken.

- Video-Signal-Erzeugung

Die serielle Videoinformation des Schieberegisters wird mit dem aus Synchronisierungsgründen zwischengespeicherten Invers- bzw. Cursorbit verknüpft, das ebenso wie das Blanksignal (Horizontale, vertikale Austastlücke) vom Controller Chip erzeugt wird.

- Taktgenerator

Die Signale DC und DCC werden von einem Quarzgenerator bzw. von dem nachgeschalteten Synchronzähler geliefert. Dieser wird mit dem Videodot Clock getaktet und erzeugt bei Überlauf das Signal DCC. Mit der nächsten Taktflanke setzt er sich auf den Anfangszustand 8.

- Bildwiederholpeicher mit Ausgabelatch

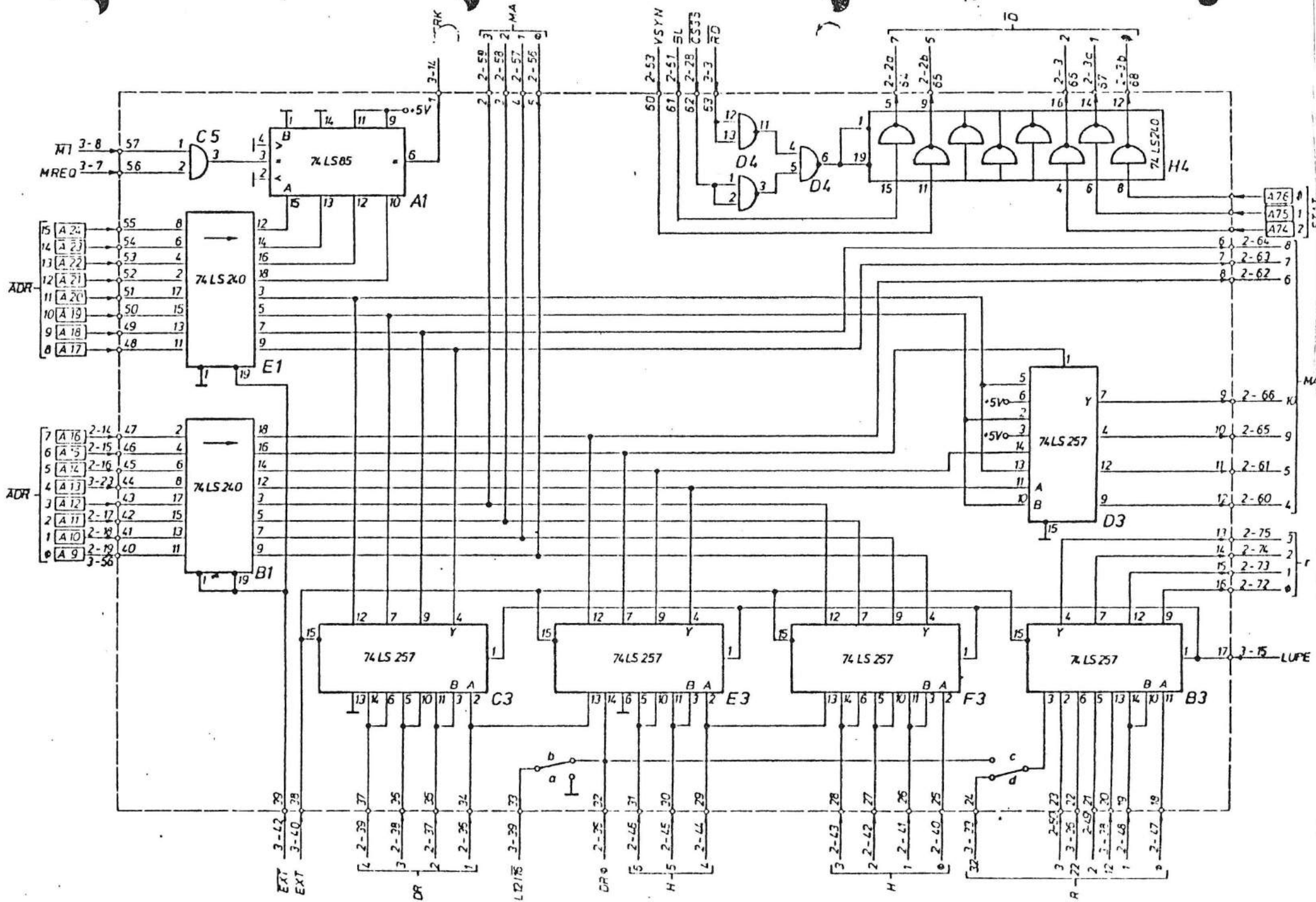
Die CRT-Controllerplatine benötigt je nach gewünschtem Darstellungsformat (16 x 32, 16 x 64, 24 x 80, 12x40 Zeilen x Zeichen) einen 1 K bzw. 2 K großen Bildwiederholpeicher, der durch die Multiplexer der Adressenauswahl adressiert wird.

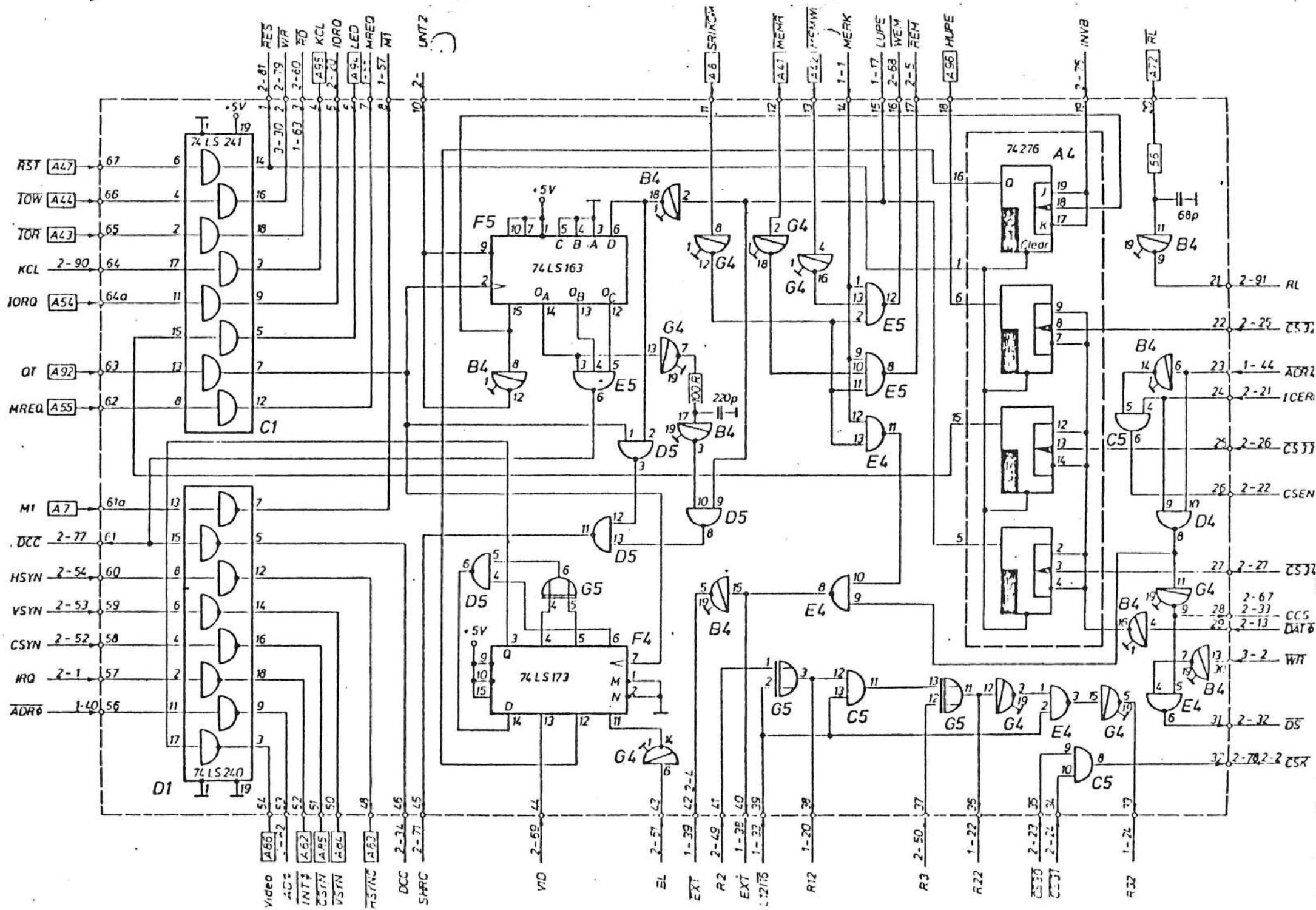
Die 8-Bit Datenausgänge dieses Speichers werden für einen Zeichenzyklus mit dem Signal DCC zwischengespeichert. Die unteren 7-Bit entsprechen dem ASCII-Code, während das 8. Bit zu einer zeichenbegleitenden Inversdarstellung benutzt wird.

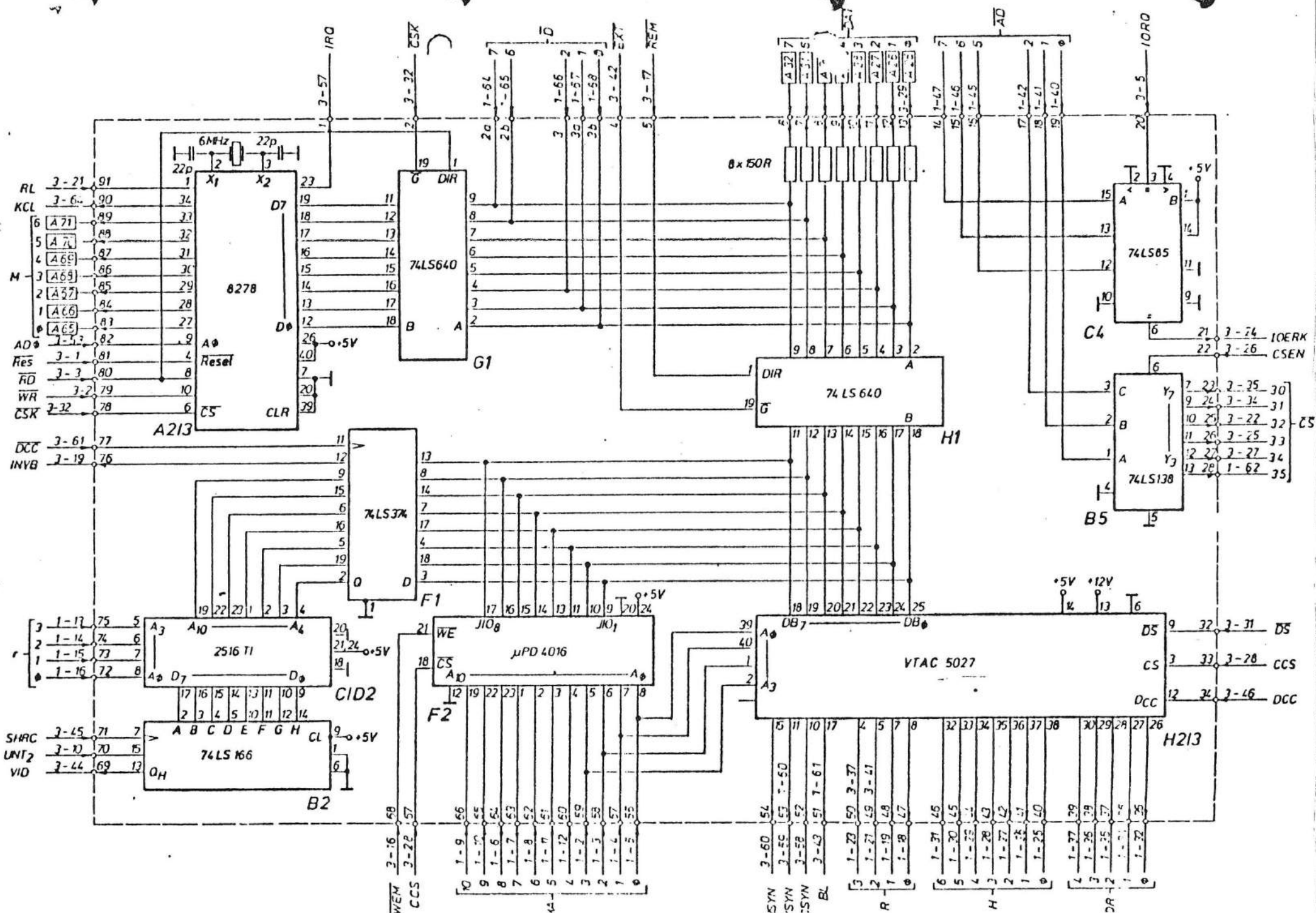
- Zeichengenerator

Der ASCII-Code des Bildwiederholspeichers bildet zusammen mit den 4 Bit ($R_7 - R_3$) des Controller Chip die Adresse für den Zeichengenerator (2 K Byte Prom mit max. 128 Zeichen).

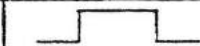
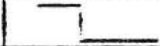
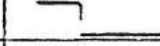
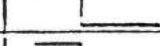
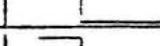
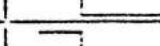



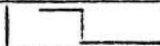

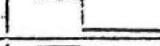
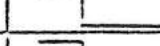
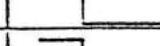
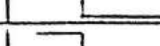
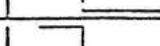


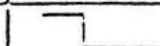
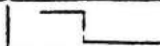
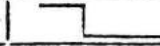
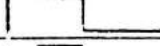
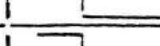



Mit dem Takt DCC werden die Daten dieses PROM'S parallel in ein Schieberegister übernommen und mit dem Takt DC in eine serielle Form gebracht.

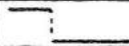
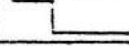
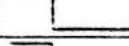
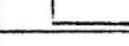


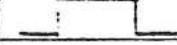
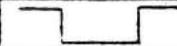



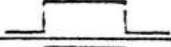
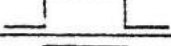
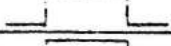
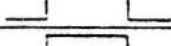





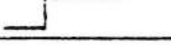


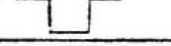
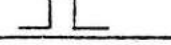
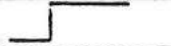




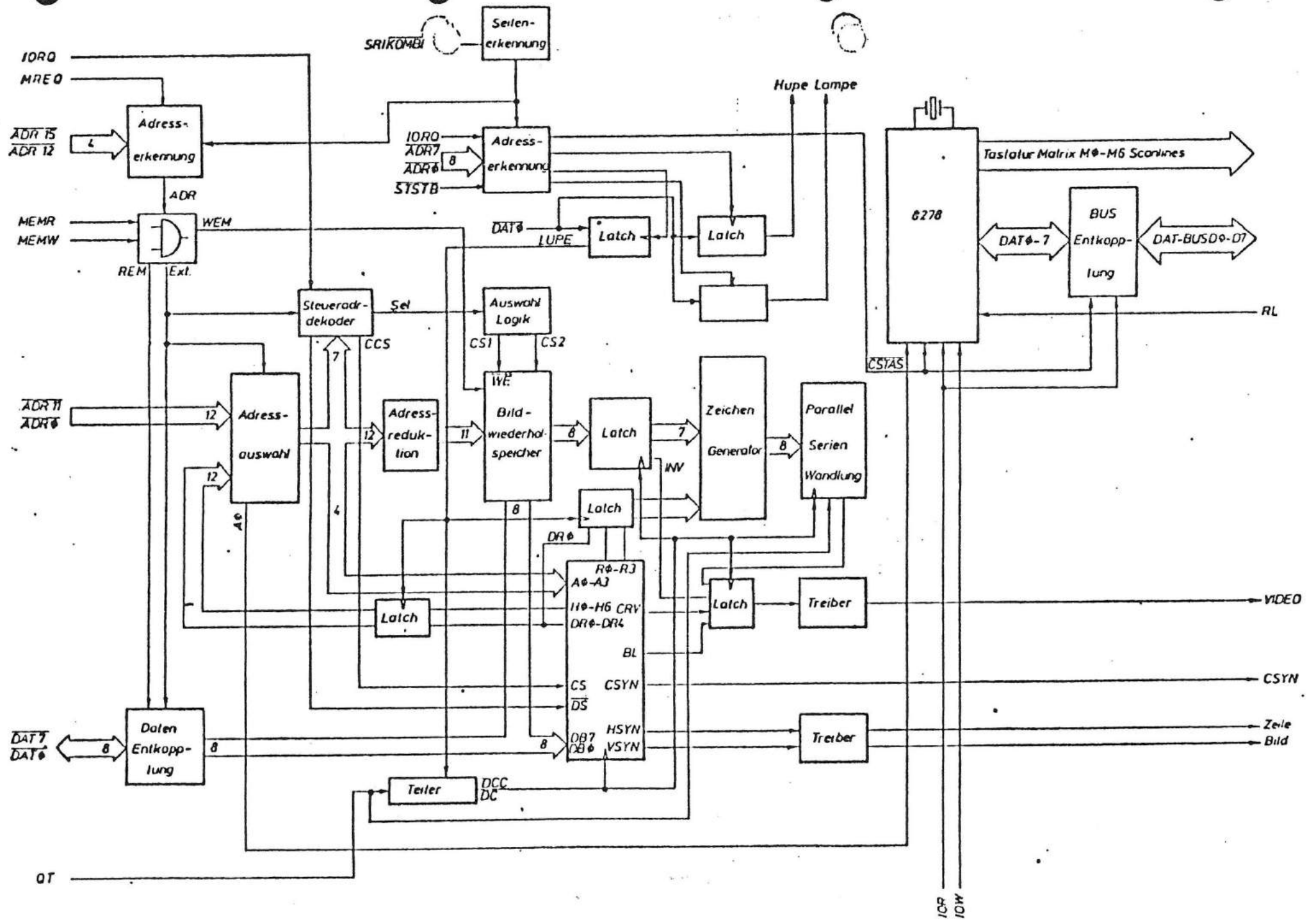


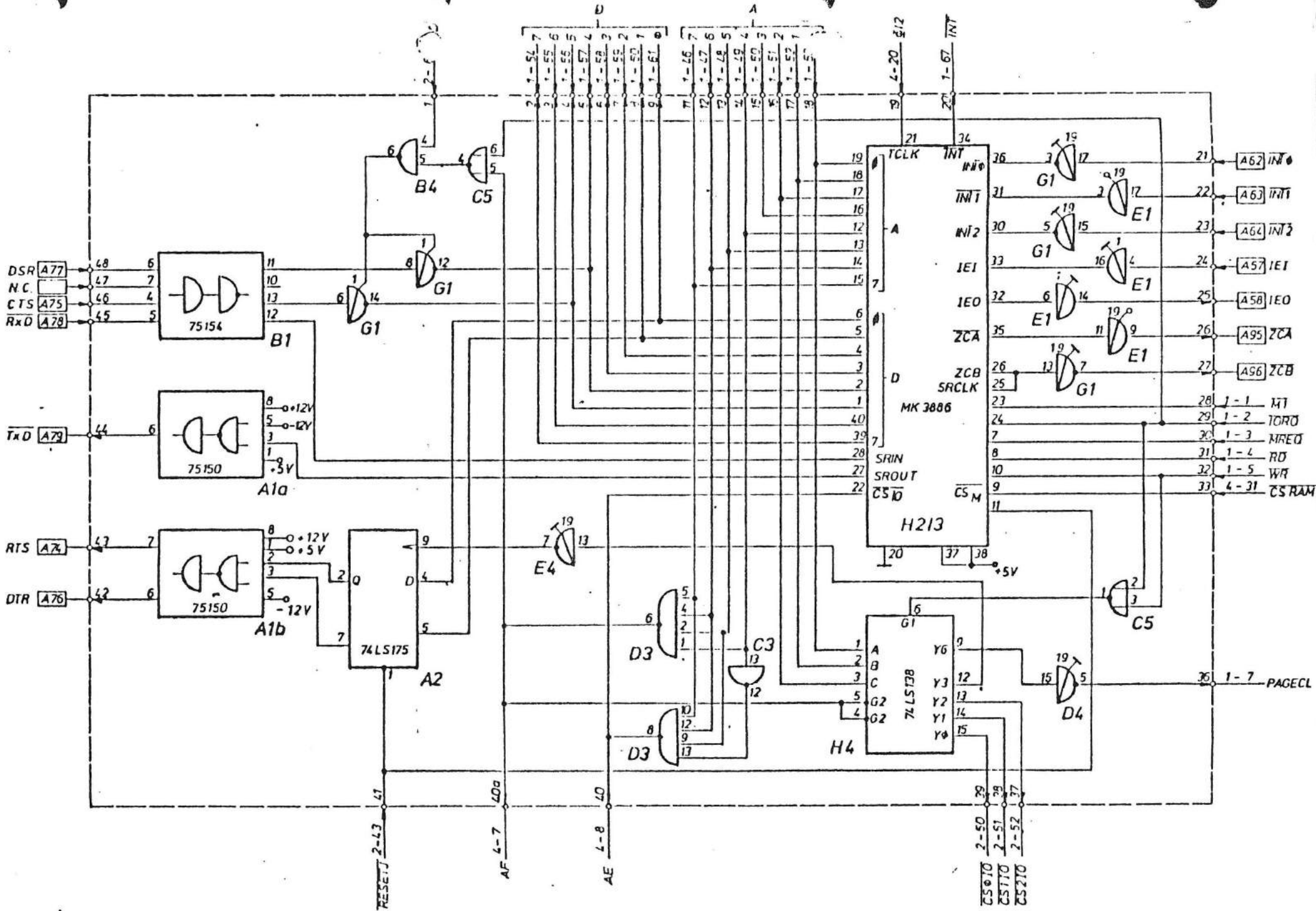
STECKERBELEGUNGSLISTE

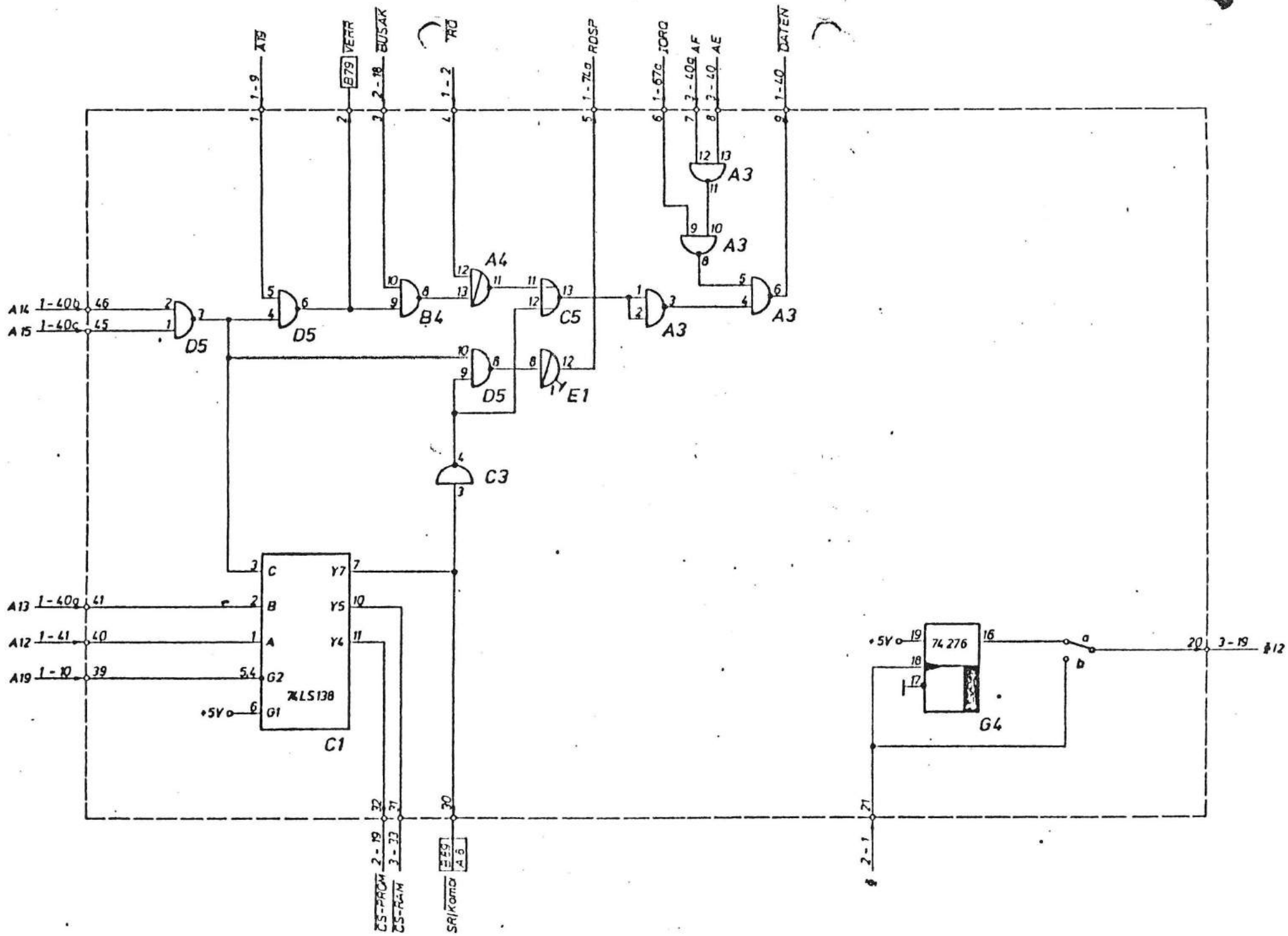
PIN	Bezeichnung	Diagramm	Erläuterungen	Blatt Nr.
01	+ 5V			
02	0V			
03	+ 12V			
04	0V			
05	- 12V			
06	0V			
07	M1		Machine Cycle One	
08	SR/KOMBI		Freigabe Speicher/Interface	
09	ADR 0		Adreß-Bit 0	
10	ADR 1		Adreß-Bit 1	
11	ADR 2		Adreß-Bit 2	
12	ADR 3		Adreß-Bit 3	
13	ADR 4		Adreß-Bit 4	
14	ADR 5		Adreß-Bit 5	
15	ADR 6		Adreß-Bit 6	
16	ADR 7		Adreß-Bit 7	
17	ADR 8		Adreß-Bit 8	
18	ADR 9		Adreß-Bit 9	
19	ADR 10		Adreß-Bit 10	
20	ADR 11		Adreß-Bit 11	
21	ADR 12		Adreß-Bit 12	
22	ADR 13		Adreß-Bit 13	
23	ADR 14		Adreß-Bit 14	
24	ADR 15		Adreß-Bit 15	
25	DAT 0		Daten-Bit 0	
26	DAT 1		Daten-Bit 1	
27	DAT 2		Daten-Bit 2	
28	DAT 3		Daten-Bit 3	
29	DAT 4		Daten-Bit 4	
30	DAT 5		Daten-Bit 5	
31	DAT 6		Daten-Bit 6	
32	DAT 7		Daten-Bit 7	

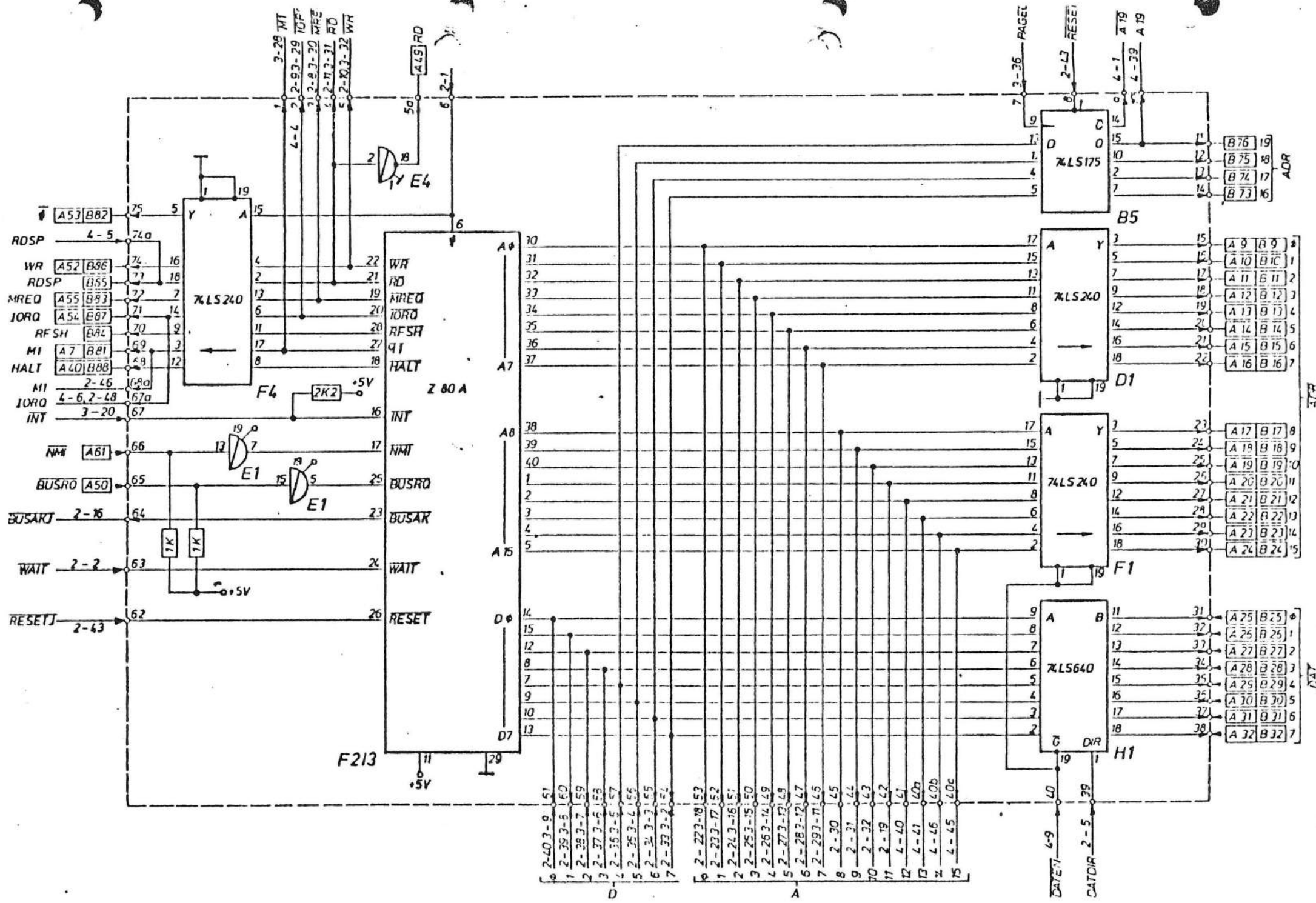
PIN	Bezeichnung	Diagramm	Erläuterungen	Blatt Nr.
33	+ 5V			
34	0V			
35	+ 12V			
36	0V			
37	- 12V			
38	0V			
39				
40				
41	$\overline{\text{MEMR}}$		Memory Read	
42	$\overline{\text{MEMW}}$		Memory Write	
43	$\overline{\text{IOR}}$		IN/OUT Read	
44	$\overline{\text{IOW}}$		IN/OUT Write	
45				
46				
47	$\overline{\text{RST}}$		Reset intern	
48				
49				
50				
51				
52				
53				
54	IORQ		Input/Output Request	
55	MREQ		Memory Request	
56				
57				
58				
59				
60				
61				
62	INT \emptyset		External Interrupt \emptyset	
63				
64				

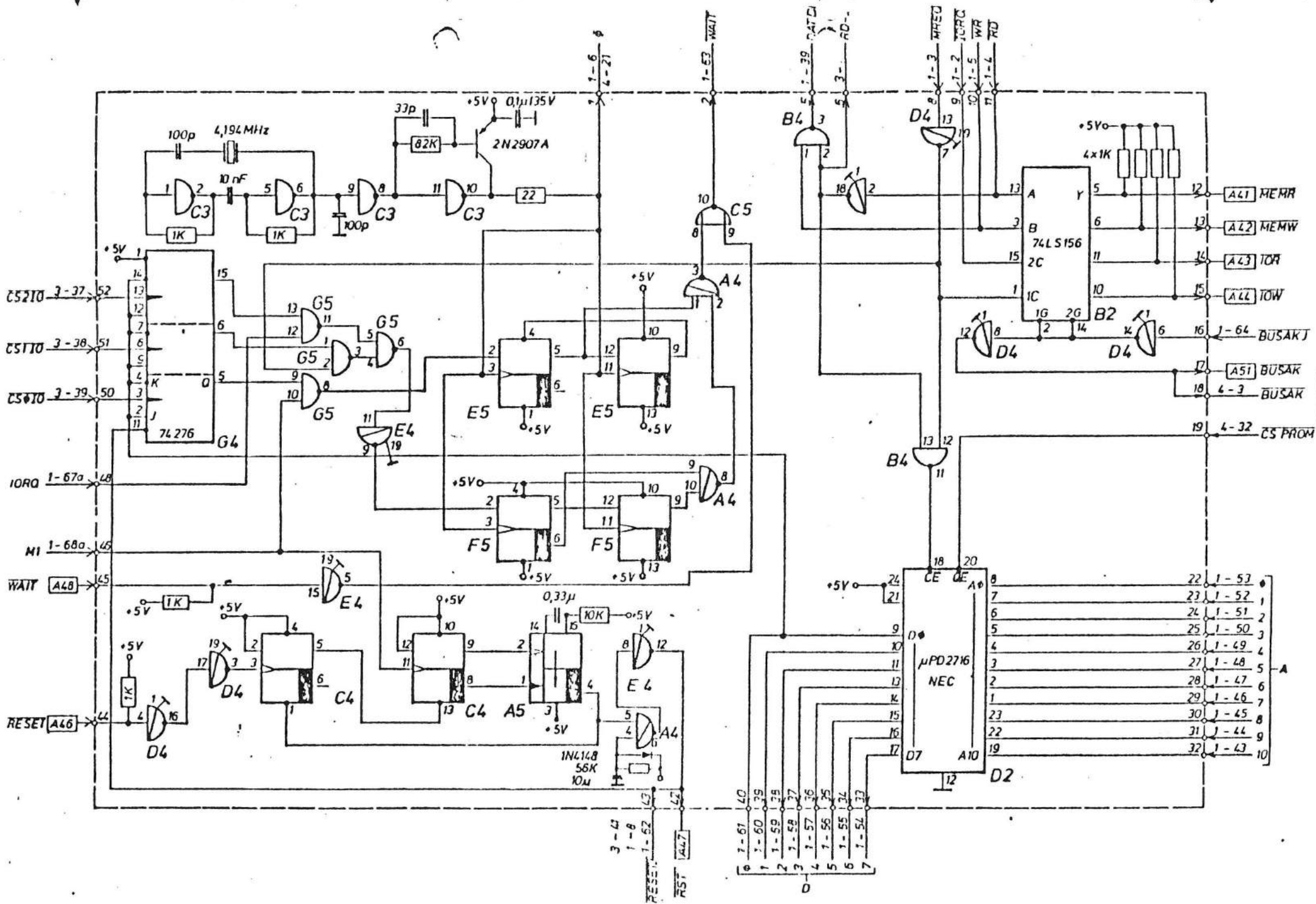
PIN	Bezeichnung	Diagramm	Erläuterungen	Blatt Nr.
65	M \emptyset		Matrix Scan Ausgang \emptyset	
66	M 1		Matrix Scan Ausgang 1	
67	M 2		Matrix Scan Ausgang 2	
68	M 3		Matrix Scan Ausgang 3	
69	M 4		Matrix Scan Ausgang 4	
70	M 5		Matrix Scan Ausgang 5	
71	M 6		Matrix Scan Ausgang 6	
72	\overline{RL}		Tastatur Rückleitung	
73				
74	STAT 2		Statussignal 2	
75	STAT 1		Statussignal 1	
75	STAT \emptyset		Statussignal \emptyset	
77				
78				
79				
80				
81				
82				
83	\overline{HSYN}		Zeilenimpuls	
84	\overline{VSYN}		Bildimpuls	
85	\overline{CSYN}		Composite Synchronisier Signal	
86	Video		Video Signal	
87				
88				
89				
90				
91				
92	QT		Quarztakt	
93				
94	LED		Lampenansteuerung	
95	KCL		Key Clock	
96	Hupe		Hupe	

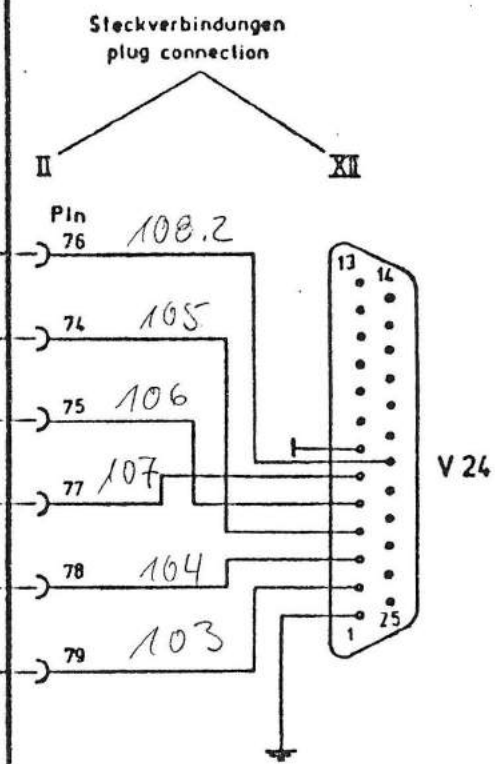
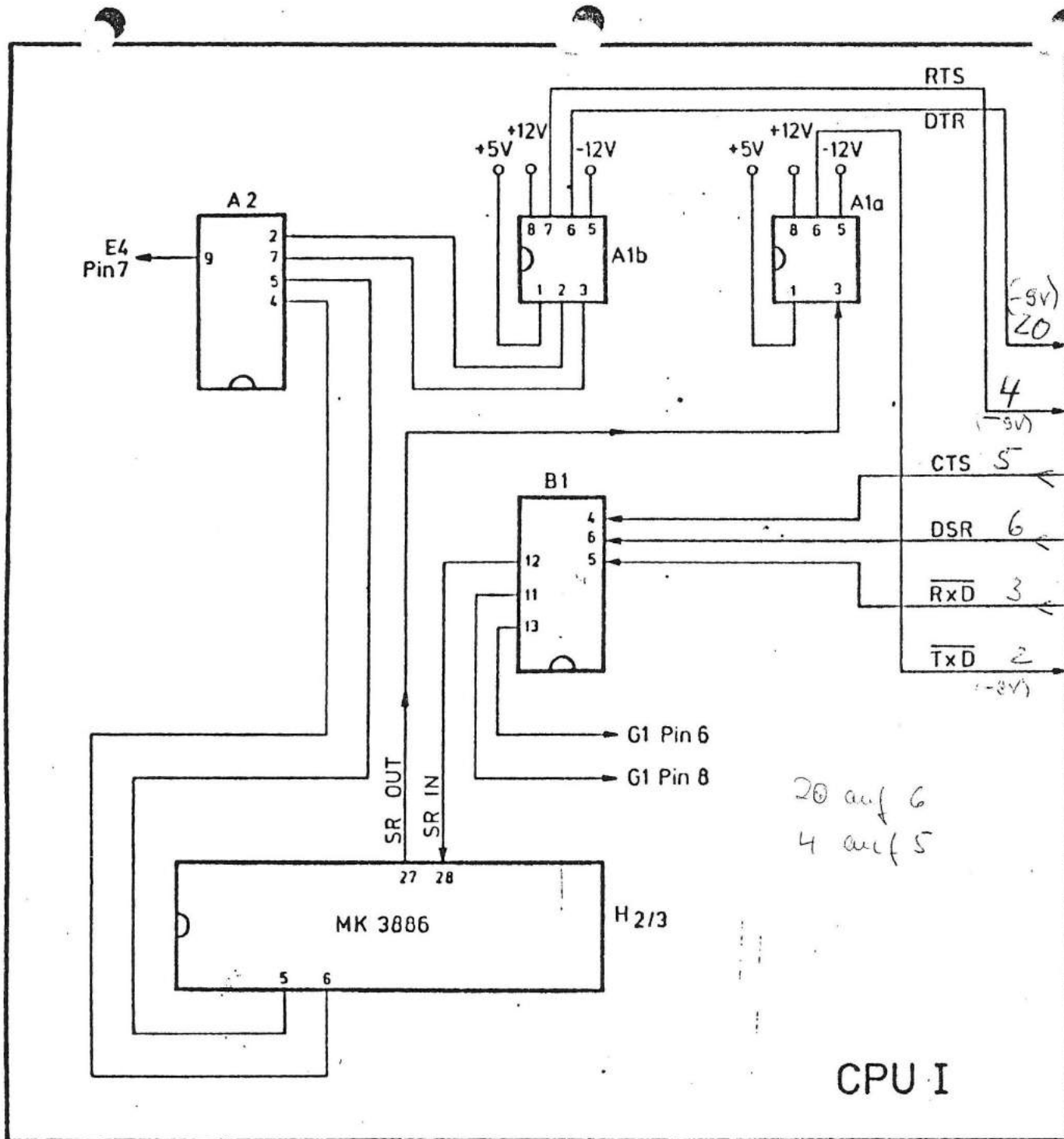








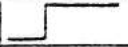
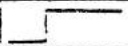
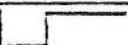
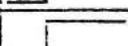
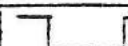

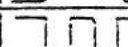

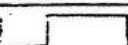
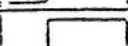

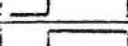
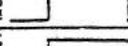
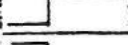




20 auf 6
4 auf 5



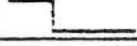
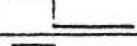
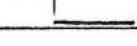

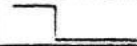
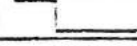
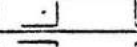
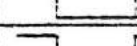

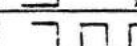

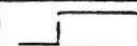

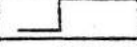
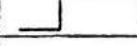
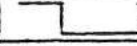
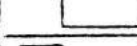
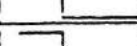
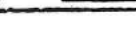
- A1a/A1b = 75150
- A2 = 74LS175
- B1 = 75154
- H2/3 = MK 3886


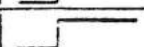
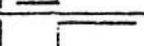
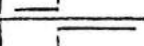
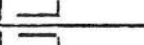
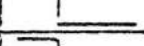

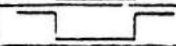
CPU I

PIN	Bezeichnung	Diagramm	Erläuterungen	Statt Nr.
65	+ 5V			
66	0V			
67	+ 12V			
68	0V			
69	- 12V			
70	0V			
71				
72				
73	ADR 16		Adreß-Bit 16	
74	ADR 17		Adreß-Bit 17	
75	ADR 18		Adreß-Bit 18	
76	ADR 19		Adreß-Bit 19	
77				
78				
79	\overline{VERR}		Verriegelung Speicher	
80				
81	M1		Machine Cycle One	
82	\emptyset		Systemtakt	
83	MREQ		Memory Request	
84	RFSH		Refresh	
85	RDSP		Read-Speicher	
86	WR		Write	
87	IORQ		Input/Output Request	
88	HALT		Halt State	
89	SR/ $\overline{\text{Kombi}}$		Freigabe Speicher/Interface	

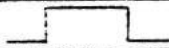
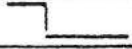

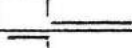
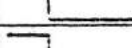
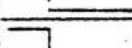

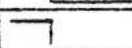
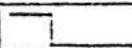
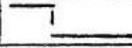
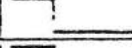
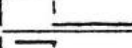
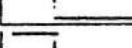
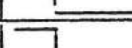
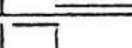
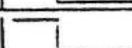
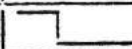


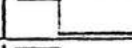
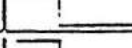
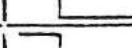
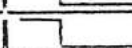



STECKERBELEGUNGSLISTE

PIN	Bezeichnung	Diagramm	Erläuterungen	Blatt Nr.
01	+ 5V			
02	0V			
03	+ 12V			
04	0V			
05	- 12V			
06	0V			
07				
08				
09	$\overline{\text{ADR } 0}$		Adreß-Bit 0	
10	$\overline{\text{ADR } 1}$		Adreß-Bit 1	
11	$\overline{\text{ADR } 2}$		Adreß-Bit 2	
12	$\overline{\text{ADR } 3}$		Adreß-Bit 3	
13	$\overline{\text{ADR } 4}$		Adreß-Bit 4	
14	$\overline{\text{ADR } 5}$		Adreß-Bit 5	
15	$\overline{\text{ADR } 6}$		Adreß-Bit 6	
16	$\overline{\text{ADR } 7}$		Adreß-Bit 7	
17	$\overline{\text{ADR } 8}$		Adreß-Bit 8	
18	$\overline{\text{ADR } 9}$		Adreß-Bit 9	
19	$\overline{\text{ADR } 10}$		Adreß-Bit 10	
20	$\overline{\text{ADR } 11}$		Adreß-Bit 11	
21	$\overline{\text{ADR } 12}$		Adreß-Bit 12	
22	$\overline{\text{ADR } 13}$		Adreß-Bit 13	
23	$\overline{\text{ADR } 14}$		Adreß-Bit 14	
24	$\overline{\text{ADR } 15}$		Adreß-Bit 15	
25	$\overline{\text{DAT } 0}$		Daten-Bit 0	
26	$\overline{\text{DAT } 1}$		Daten-Bit 1	
27	$\overline{\text{DAT } 2}$		Daten-Bit 2	
28	$\overline{\text{DAT } 3}$		Daten-Bit 3	
29	$\overline{\text{DAT } 4}$		Daten-Bit 4	
30	$\overline{\text{DAT } 5}$		Daten-Bit 5	
31	$\overline{\text{DAT } 6}$		Daten-Bit 6	
32	$\overline{\text{DAT } 7}$		Daten-Bit 7	

PIN	Bezeichnung	Diagramm	Erläuterungen	Blatt Nr.
33	+ 5V			
34	0V			
35	+ 12V			
36	0V			
37	- 12V			
38	0V			
39				
40	HALT		Halt state	
41	$\overline{\text{MEMR}}$		Memory Read	
42	$\overline{\text{MEMW}}$		Memory Write	
43	$\overline{\text{IOR}}$		In/Out Read	
44	$\overline{\text{IOW}}$		In/Out Write	
45				
46	$\overline{\text{Reset}}$		Reset	
47	$\overline{\text{RST}}$		Reset intern	
48	$\overline{\text{WAIT}}$		Wait	
49	$\overline{\text{RD}}$		Read	
50	$\overline{\text{BUSRQ}}$		Bus Request	
51	$\overline{\text{BUSAK}}$		Bus Acknowledge	
52	$\overline{\text{WR}}$		Write	
53	\emptyset		Systemtakt	
54	$\overline{\text{IORQ}}$		Input/Output Request	
55	$\overline{\text{MREQ}}$		Memory Request	
56				
57	$\overline{\text{IEI}}$		Interrupt Enable In	
58	$\overline{\text{IEO}}$		Interrupt Enable Out	
59				
60				
61	$\overline{\text{NMI}}$		Non Maskable Interrupt	
62	$\overline{\text{INT } \emptyset}$		External Interrupt \emptyset	
63	$\overline{\text{INT } 1}$		External Interrupt 1	
64	$\overline{\text{INT } 2}$		External Interrupt 2	

PIN	Bezeichnung	Diagramm	Erläuterungen	Blatt Nr.
65	+ 5V			
66	0V			
67	+ 12V			
68	0V			
69	- 12V			
70	0V			
71				
72				
73				
74	RTS		Request to Send	
75	CTS		Clear to Send	
76	DTR		Data Terminal Ready	
77	DSR		Data Set Ready	
78	$\overline{\text{RxD}}$		Received Data	
79	$\overline{\text{TxD}}$		Transmitted Data	
80				
81				
82				
83				
84				
85				
86				
87				
88				
89				
90				
91				
92				
93				
94				
95	$\overline{\text{ZCA}}$		Zähler A = Null	
96	$\overline{\text{ZCB}}$		Zähler B = Null	

STECKERBELEGUNGSLISTE

PIN	Bezeichnung	Diagramm	Erläuterungen	Blatt Nr.
01	+ 5V			
02	0V			
03	+ 12V			
04	0V			
05	- 12V			
06	0V			
07	M1		Machine Cycle one	
08	SR/KOMBI		Freigabe Speicher/Interface	
09	$\overline{\text{ADR } 0}$		Adreß-Bit 0	
10	$\overline{\text{ADR } 1}$		Adreß-Bit 1	
11	$\overline{\text{ADR } 2}$		Adreß-Bit 2	
12	$\overline{\text{ADR } 3}$		Adreß-Bit 3	
13	$\overline{\text{ADR } 4}$		Adreß-Bit 4	
14	$\overline{\text{ADR } 5}$		Adreß-Bit 5	
15	$\overline{\text{ADR } 6}$		Adreß-Bit 6	
16	$\overline{\text{ADR } 7}$		Adreß-Bit 7	
17	$\overline{\text{ADR } 8}$		Adreß-Bit 8	
18	$\overline{\text{ADR } 9}$		Adreß-Bit 9	
19	$\overline{\text{ADR } 10}$		Adreß-Bit 10	
20	$\overline{\text{ADR } 11}$		Adreß-Bit 11	
21	$\overline{\text{ADR } 12}$		Adreß-Bit 12	
22	$\overline{\text{ADR } 13}$		Adreß-Bit 13	
23	$\overline{\text{ADR } 14}$		Adreß-Bit 14	
24	$\overline{\text{ADR } 15}$		Adreß-Bit 15	
25	$\overline{\text{DAT } 0}$		Daten-Bit 0	
26	$\overline{\text{DAT } 1}$		Daten-Bit 1	
27	$\overline{\text{DAT } 2}$		Daten-Bit 2	
28	$\overline{\text{DAT } 3}$		Daten-Bit 3	
29	$\overline{\text{DAT } 4}$		Daten-Bit 4	
30	$\overline{\text{DAT } 5}$		Daten-Bit 5	
31	$\overline{\text{DAT } 6}$		Daten-Bit 6	
32	$\overline{\text{DAT } 7}$		Daten-Bit 7	

Die Baugruppe CPU V basiert auf dem Mikroprozessor Z 80 A und unterteilt sich in folgende Funktionsblöcke:

1. Prozessor
2. Promspeicher
3. Wait-Logik, Reset-Logik
4. Serielle Schnittstelle (Kombo-Chip)
5. Adreßlogik

1. Prozessor

Der Mikroprozessor Z 80 A arbeitet mit einem Takt von 4,194 MHz. Durch eine spezielle Schaltung wird gewährleistet, daß die Anstiegszeiten und Abfallzeiten der Taktflanke kleiner 30 Nanosekunden sind. Eine interne Reset-Logik (R-C Glied 56KR/10 μ F) bringt den Prozessor nach dem Einschalten der Spannung in Grundstellung. Die 16 Bit Adresse werden über Treiber (LS 240) auf den A- und B-Stecker geführt. Über ein zusätzliches Register werden aus den Daten die Adreßbits 16 bis 19 generiert. Der Datenbus geht über einen bidirektionalen Treiber (LS 640) zum Stecker. Ein Dekoder erzeugt aus den Signalen RD, WR, IORQ und MREQ die Schreib- Lese-Signale für Speicher und Peripherie.

2. Promspeicher

Auf der CPU-Karte ist ein Sockel vorgesehen für die Aufnahme eines 2K x 8 Prom vom Typ 2716. Die Verknüpfung der Signale RD und MREQ erzeugen einen Chipenable. Die Leitung CS Prom vom Dekoder C1 gibt die Daten des Proms auf den Datenbus.

3. Wait-Logik, Reset-Logik

Es gibt drei Arten des Prozessorzugriffs:

- a) OP-Code lesen
- b) Speicher schreiben oder lesen
- c) I/O schreiben oder lesen

Für jeden dieser Zugriffe ist ein Wait-Zyklus einschaltbar. Die Flip-Flops E5 und F5 erzeugen synchronisiert mit dem Takt \emptyset einen Puls von der Länge eines Taktzyklus, der dem Prozessor am Wait-Eingang zugeführt wird.

Die Reset-Logik übernimmt die Synchronisierung eines externen Resetsignals mit dem Signal M1. Dies ist notwendig, damit ein Auffrischen von dynamischen Rams nicht unterbrochen wird.

4. Serielle Schnittstelle

Der Kombo-Chip vom Typ 3886 enthält außer einem Ramspeicher von 256 Byte einen seriellen Sende- und Empfangskanal. Der Takt für den Kombochip ist der Prozessortakt geteilt durch 2. Über einen internen Zähler wird die Baudrate für die serielle Übertragung eingestellt. Über Treiber und Empfänger nach RS 232 Spezifikation ist die serielle Steuereinheit mit dem Stecker verbunden. Mit dem Register A2 (LS 175) werden zwei Steuersignale (DTR, RTS) erzeugt. Über Tri-State-Treiber (LS 240) werden die ankommenden Steuersignale (DSR, CTS) auf den Datenbus geführt.

5. Adreßlogik

Die Adreßlogik der CPU V steuert, in Abhängigkeit der Adreßbits 14, 15 und 19, die Zugriffe auf den internen bzw. externen Speicher. Der Dekoder C1 (LS 138) ist dann aktiv, wenn über Adreßbit 19 die Seite 8 (intern) angewählt ist. Die Bits A12, A13 erzeugen die Chipselekt's für Ram und Prom, sowie das Signal SR/Kombi. Um aus der Seite 8 auf den Hauptspeicher zugreifen zu können, wird bei der Adresse größer $C000_H$ der Datenbustreiber H1 freigegeben, der ansonsten über die Adreßlogik, wenn A19 aktiv ist, gesperrt ist.

Ebenso gibt das Signal IORQ und das Signal SR/ $\overline{\text{Kombi}}$ den Bustreiber frei. Die Adressen A14, A15 und A19 erzeugen über das Gatter D5 ein Speichersperrsignal (VERR), das in der internen Seite bei Adressen kleiner $C000_H$ aktiv ist. Eine Adreßdekodierung (Gatter D3) erzeugt die Steuersignale für die internen Register und den Kombochip.

Adreßbelegung I/O:

Ausgabe

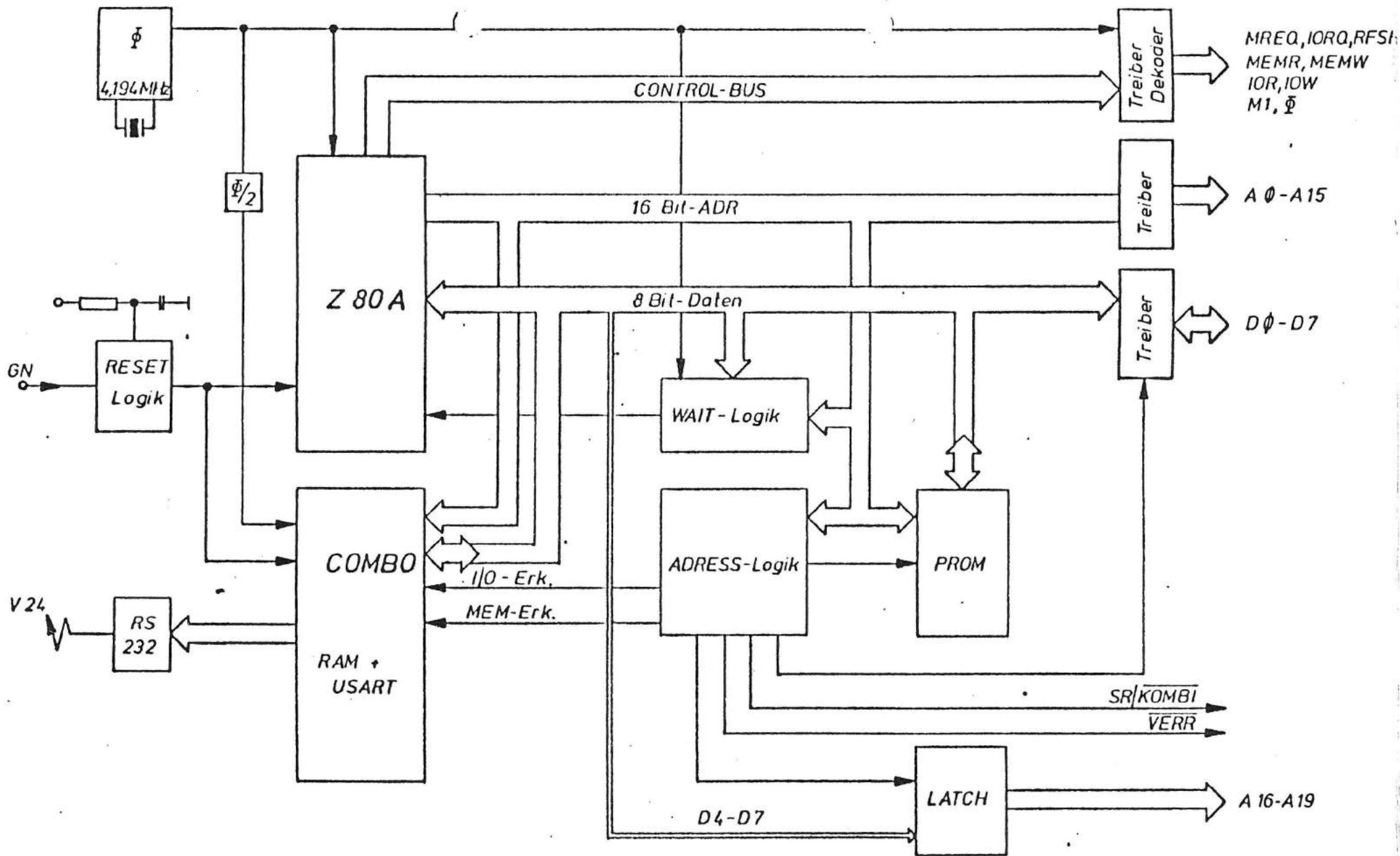
<u>1 1 1 1</u>	0 0 0 0 - CS \emptyset	<u>1 1 1 0 0 0 0 0</u> - CS I/O Combo
	0 0 0 1 - CS1	
F	0 0 1 0 - CS2	E
	0 0 1 1 - CSV24	
	0 1 1 0 - CSPage	

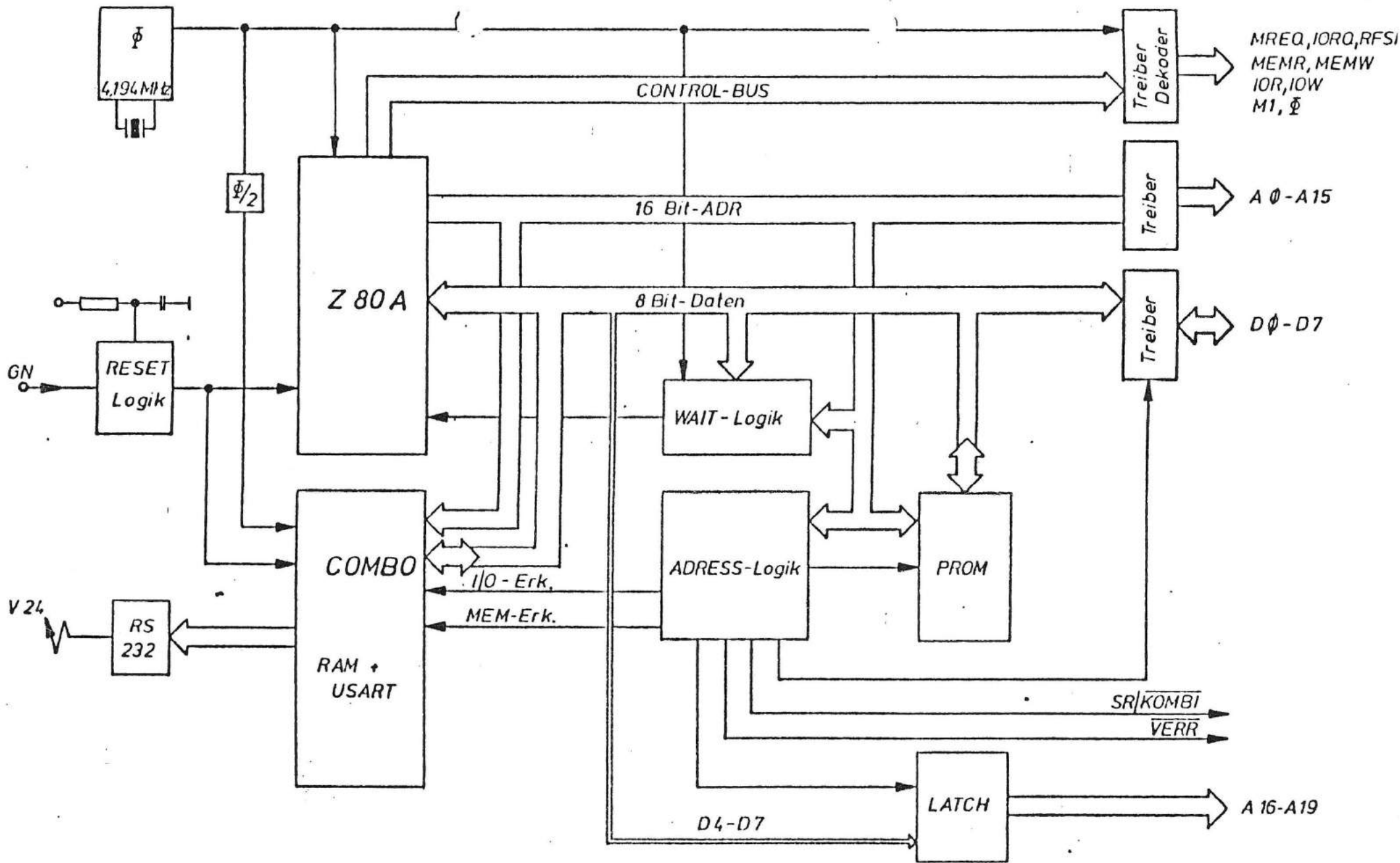
CS \emptyset → Waitzyklus OP-Code	} D \emptyset = 0 Wait aus D \emptyset = 1 Wait ein
CS1 → Waitzyklus Memory	
CS2 → Waitzyklus I/O	

CSV24: D \emptyset → RTS	CSPage: D7 → Adreßbit 16
D1 → DTR	D6 → Adreßbit 17
	D5 → Adreßbit 18
	D4 → Adreßbit 19

Eingabe F \emptyset_H : D4 → DSR
D5 → CTS

Adresse $\emptyset \emptyset \emptyset \emptyset$: Prom (2K x 8)
Adresse 1 $\emptyset \emptyset \emptyset$: RAM (256 x 8 Combo)
Adresse 3 $\emptyset \emptyset \emptyset$: Kombi (Bildspeicher)

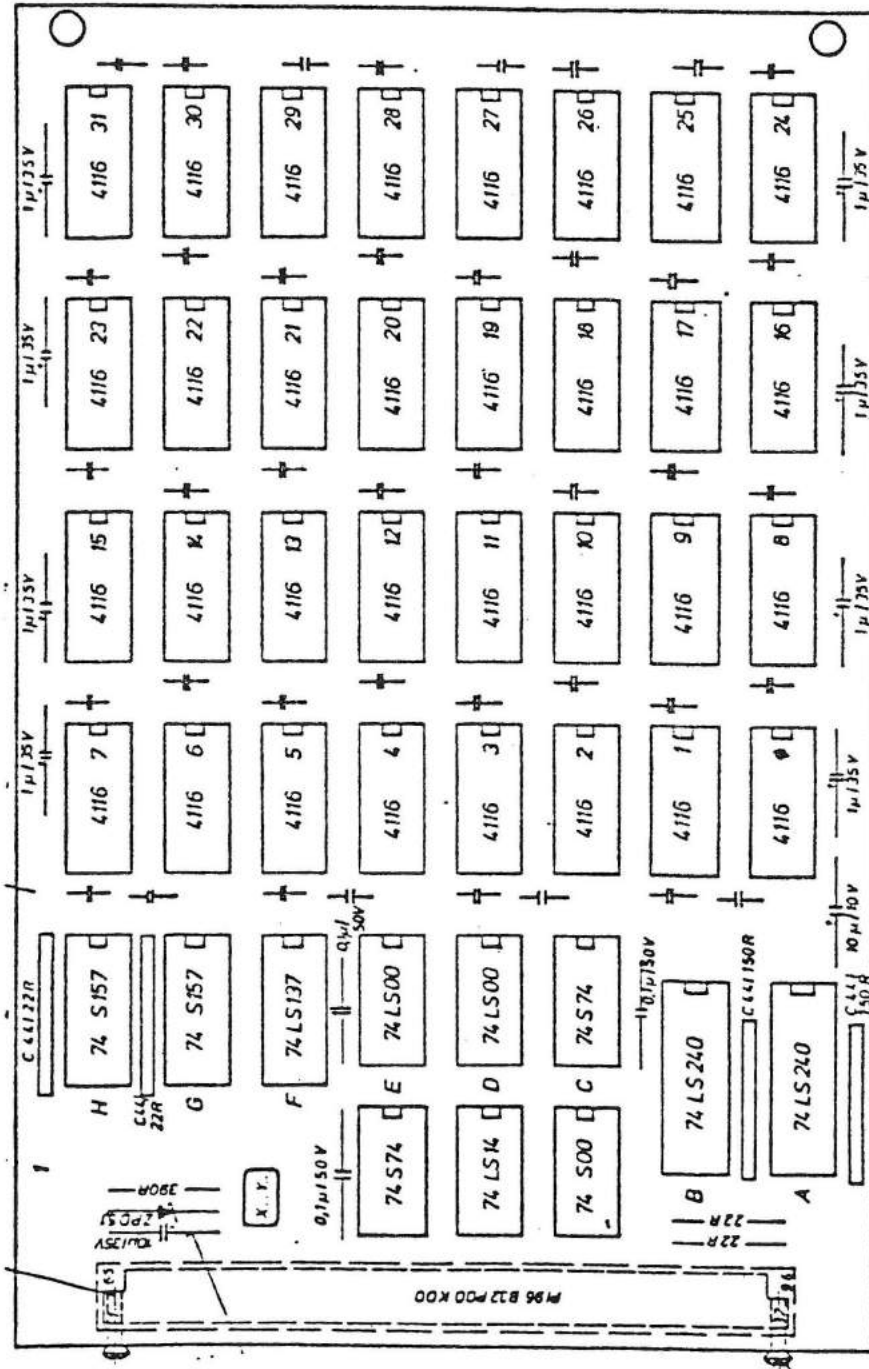




ITT 3030

auf Wachstum programmiert

64 KB - Speicherzusatz



Technik der Welt

ITT

Beschreibung 64K - RAM

Die Baugruppe 64K-RAM adaptiert über einen speziellen Stecker direkt die CPU.

An diesem Stecker sind nur die a- und c-Reihe belegt. Die a-Reihe entspricht dem MC 80-BUS, die c-Reihe beinhaltet die BUS-Steuersignale und spezielle Signale, die der Speicher benötigt.

Der 64K-RAM-Speicher ist in 4 Bänke à 16K organisiert. Die RAM's selbst sind 16K x 1 dyn. Chips mit 200 ns Zugriffszeit.

Am Adreßmultiplexer liegen die unteren 14 Bit Adressen an, aus denen die RAS- und CAS-Adresse erzeugt werden.

Nach Anlegen von MREQ und den Adressen 14 und 15 wird am Dekoder F2 eines von vier RAS-Signalen erzeugt und eine Hälfte der Adressen A0 - A13 als RAS-Adresse übernommen. Über 2 Flip-Flops (E1) wird ein Umschaltsignal (UMUX) erzeugt, das den Multiplexer auf die anderen 7 Bit umschaltet und danach über Gatter C1 den CAS-Strobe erzeugt.

Das Signal VERR sperrt die Adreßerkennung.

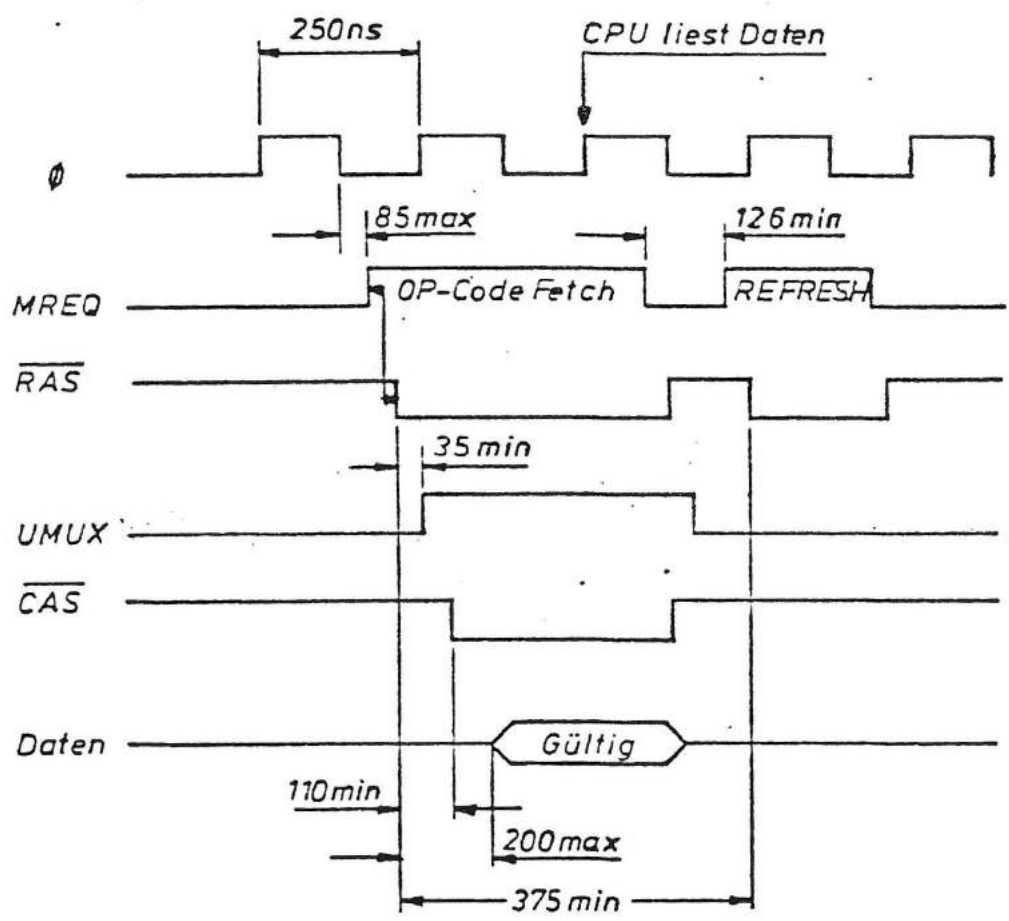
Das Signal RFSH startet den Auffrischvorgang über einen "RAS only Refresh"; es werden alle RAS-Signale aktiv und CAS gesperrt. Das Signal SR/\overline{Kombi} verhindert das Schreiben in den RAM, wenn der Bildspeicher angesprochen wird.

Die Auftrennung der Daten von bidirektionalen in den unidirektionalen BUS geschieht über zwei LS 240, die speziell geschaltet sind.

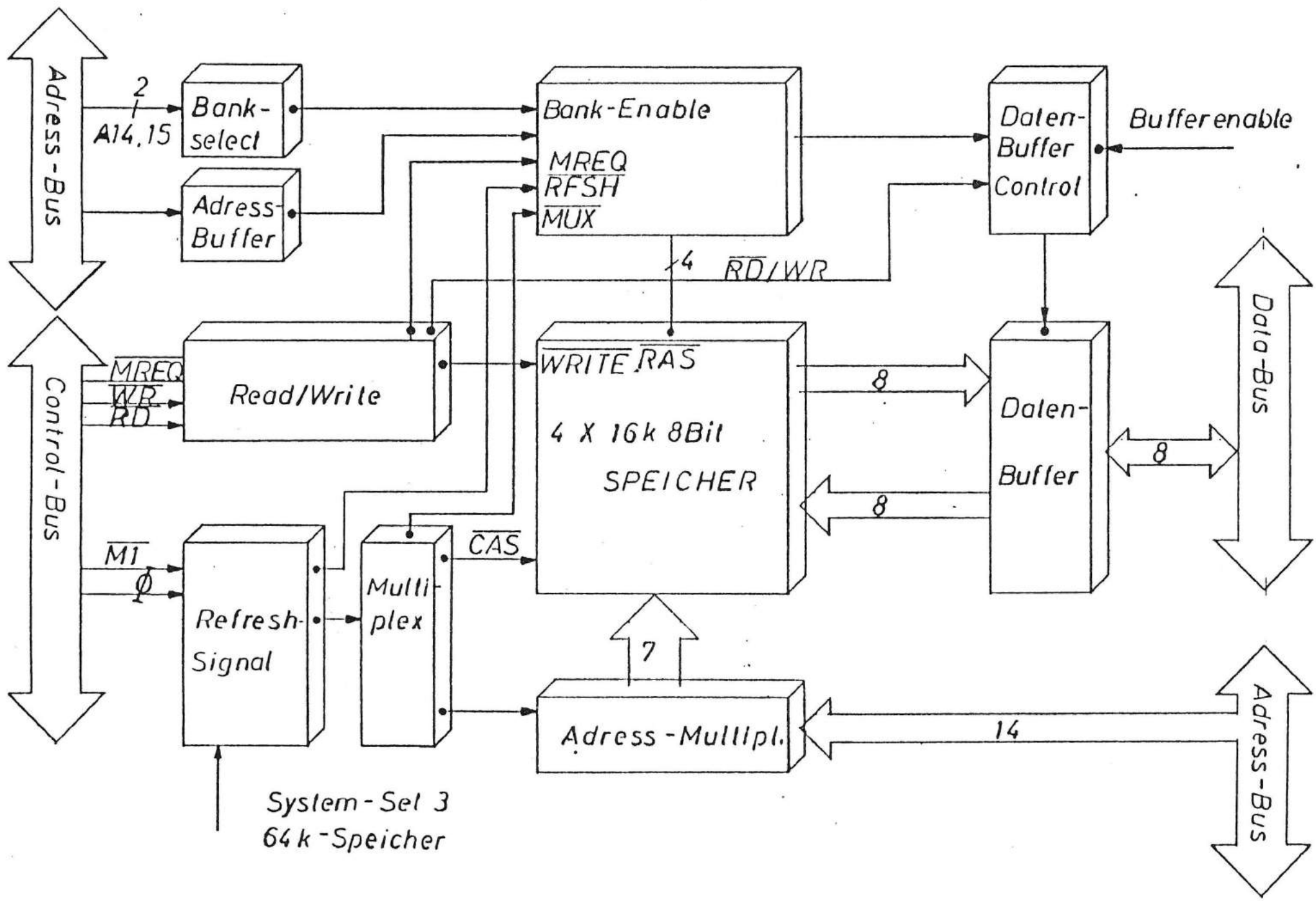
Das Refreshen erfolgt alle 2 ms in 128 Zyklen. Den Refreshzyklus steuert die CPU-280 selbständig. Zur Dämpfung von negativen Überschwingungen auf Grund der kapazitiven Belastung von 32 Chips sind verschiedene Signale mit 22 Ohm-Längswiderständen versehen.

1. Stromaufnahme:
- | | | |
|-------|--------|--------------------------|
| + 5V | ± 5 % | ≲ 300 mA |
| + 12V | ± 5 % | ≲ 200 mA |
| - 12V | ± 20 % | ≲ 50 mA (- 5V Erzeugung) |

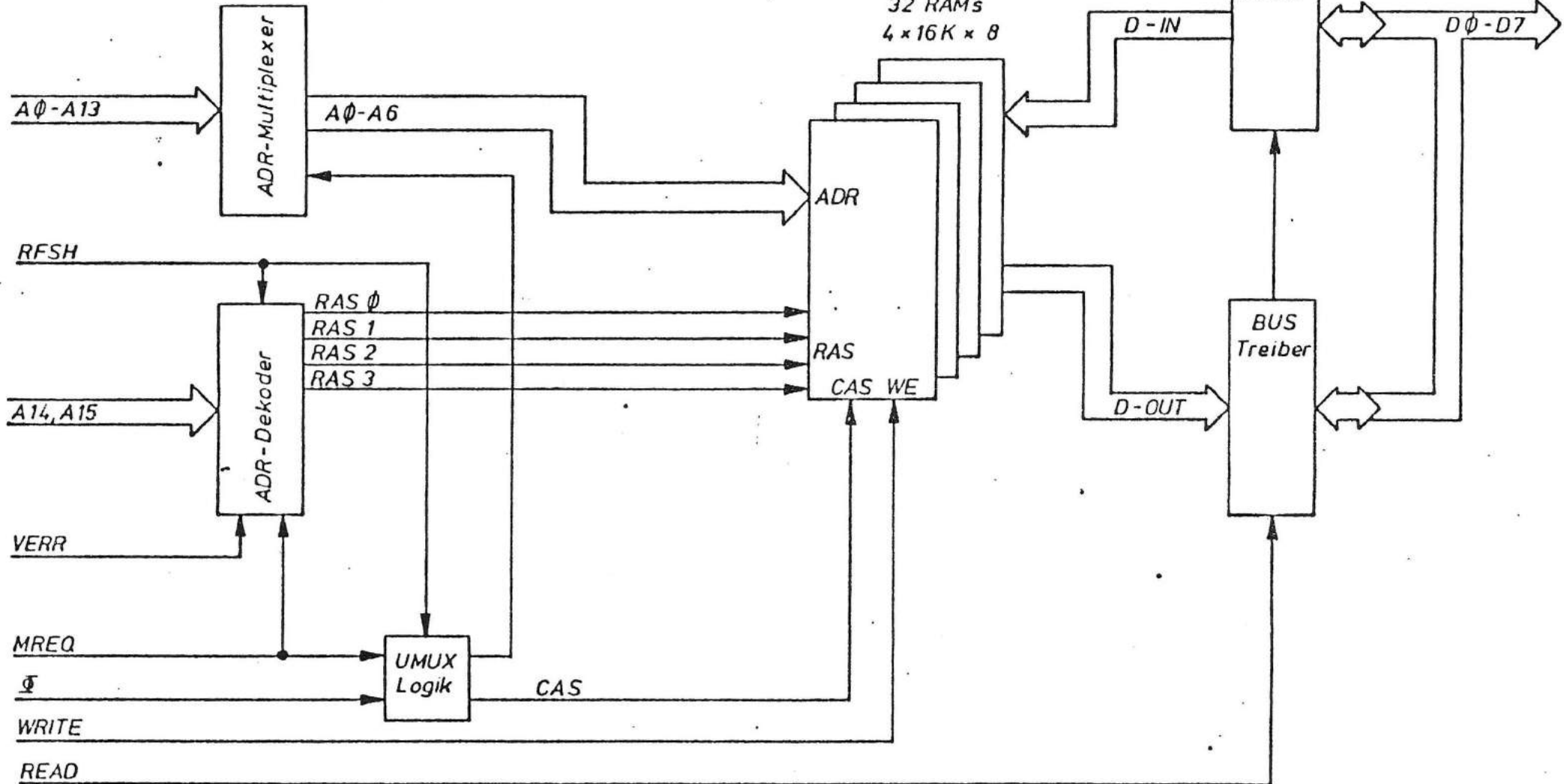
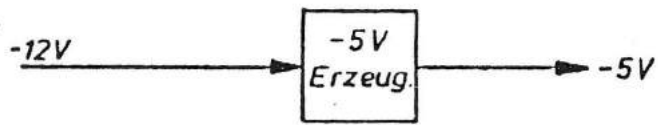
2. Timing RAS, CAS, UMUX



alle Zeiten in Nanosekunden



System - Set 3
64k-Speicher



TYPE 3080

SOFTWARE

1

Schnittstellen - Beschreibung von Bildschirm- und
Tastatur - Treiber

DATA COM
Ingenieurgesellschaft mbH
Altenwoogstraße 31
6750 KAISERSLAUTERN
Telefon 0631/40806 - Fs 45918

This book must not, in whole or in part, be copied,
printed or reproduced in any material form without the
permission of Standard Elektrik Lorenz AG, Pforzheim
The publication of information in this document does not imply
freedom from patent and other protective rights of

Für Bildschirm und Tastatur gibt es folgende Eingänge:

- I. Betrieb mit Funktionsnummer (Adresse FE9AH)
- II. Zeichenweiser Betrieb mit/ohne Steuerzeichenerkennung (Adresse FE09H/FE76H)
- III. Tastaturstatus abfragen (Adresse FE12H)
- IV. Warten auf Taste (Adresse FE73H)

I. Betrieb mit Funktionsnummer

Mit dieser Schnittstelle können folgende Funktionen ausgeführt werden:

0. Rückgabe des Parameters X von Funktion 1.

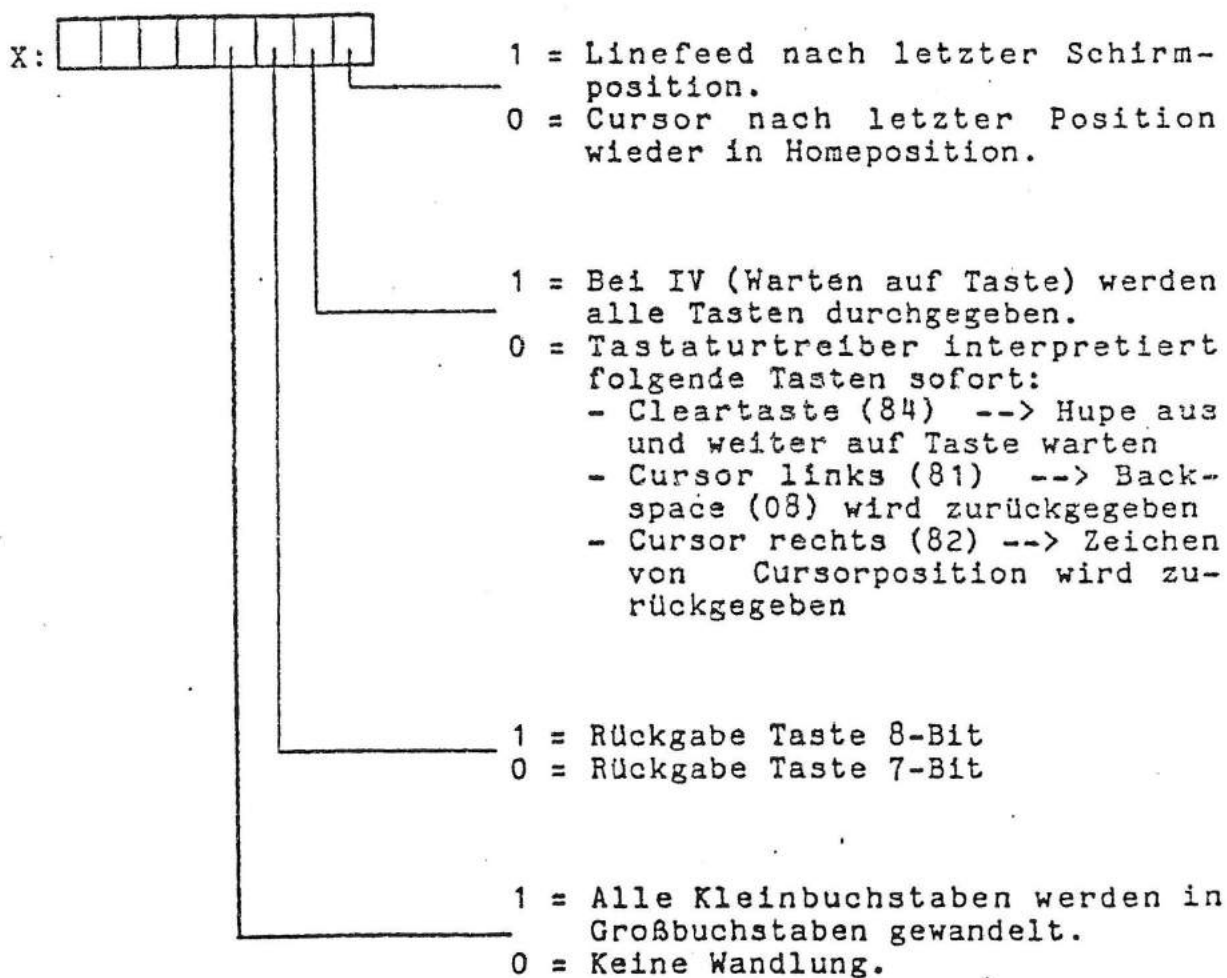
Aufruf: Reg. A = 0
Rückgabe: Reg. C = Parameter X

1. Generelle Initialisierung:

Aufruf: Reg. A = 1
Reg. E = Parameter X (Bedeutung siehe unten)

Der Bildschirm wird initialisiert, der Treiber wird grundgestellt (auch der Zentrier-Mode wird abgeschaltet, siehe Zentrieren), der Bildschirm wird gelöscht und der Cursor wird in die linke untere Ecke positioniert (letzte Zeile, erste Spalte).

Bedeutung des Parameters X:



Rückgabe: keine

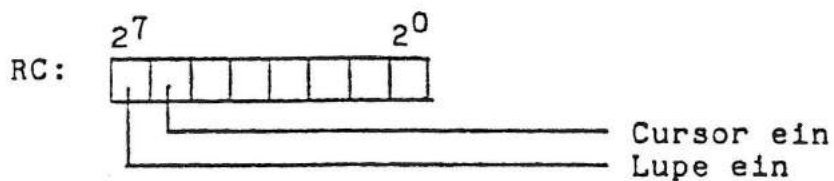
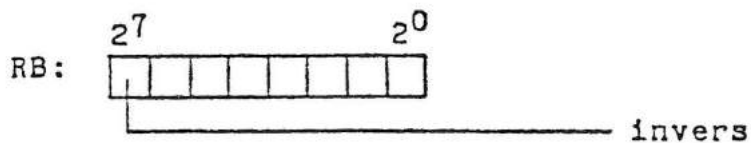
2. Display Status lesen

Es wird der Status des Display-Treibers, sowie die aktuelle Zeilen- und Spaltenanzahl zurückgegeben.

Aufruf: RA = 2

Rückgabe: RD = Anzahl Zeilen
RE = Anzahl Spalten

RB/RC = Status



Nur RH/RL sind unverändert.

3. Zeichen Lesen vom Bildschirm

Es wird das Zeichen der momentanen Cursorposition gelesen und der Cursor um eine Position weiterbewegt.

Aufruf: RA = 4

Rückgabe: RE = ASCII-Zeichen
RD = begleitende Bits (siehe Status).

4. Zeichen Schreiben auf Bildschirm

Das übergebene Zeichen wird an die momentane Cursorposition geschrieben und der Cursor um eine Position weiterbewegt.

Aufruf: RA = 5
RE = ASCII-Zeichen
RD = begleitende Bits (siehe Status).

Rückgabe: keine

5. Cursor-Position Lesen

Es wird die momentane Cursorposition zurückgegeben.

Aufruf: RA = 6

Rückgabe: RD = Zeilennummer
RE = Position in Zeile

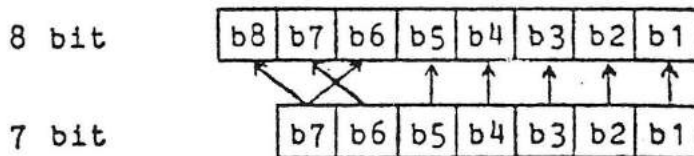
6. Tastaturtabelle laden

Es ist möglich, statt der im Betriebssystem implementierten Tastaturtabelle eigene Tastencodes zu verwenden. Die von der Taste gelieferte Matrix (siehe Layout-Schaubild) ist gleichzeitig der Index für die Tabelle. Dabei muß der Abstand zwischen Nicht-Shift-Code und Shift-Code 128 betragen.

Aufruf: Reg. A = 7
 Reg. H/L = Adresse Tastaturtabelle.

Nach dem Aufruf der Funktion steht der RAM - Bereich der neuen Tastaturtabelle wieder zur Verfügung. Wird in Register H/L 0000 übergeben, erfolgt eine Rückschaltung auf die im Betriebssystem implementierte Tastaturtabelle.

Die Tastaturtabellen sind im 8 Bit ASCII-Code abgelegt.



7. Umsetztabelle anhängen

Der Anwender kann der zeichenweisen Schnittstelle eine Umsetztabelle vorschalten. Dadurch ist es möglich, jedem beliebigen Code eine bestimmte Bildschirmfunktion zuzuordnen. Die Tabelle sieht folgendermassen aus:

	Code 1 Byte	Länge 1 Byte	String 6 Byte

insgesamt
16 Einträge

Aufruf: Reg. A = 8
 Reg. E = Anzahl der Einträge (maximal 16)
 Reg. H/L = Adresse der Tabelle

Nach dem Aufruf steht der RAM-Bereich der Tabelle wieder zur Verfügung.
 Wird in Register H/L 0000 übergeben, erfolgt ein "Abhängen" der Tabelle.

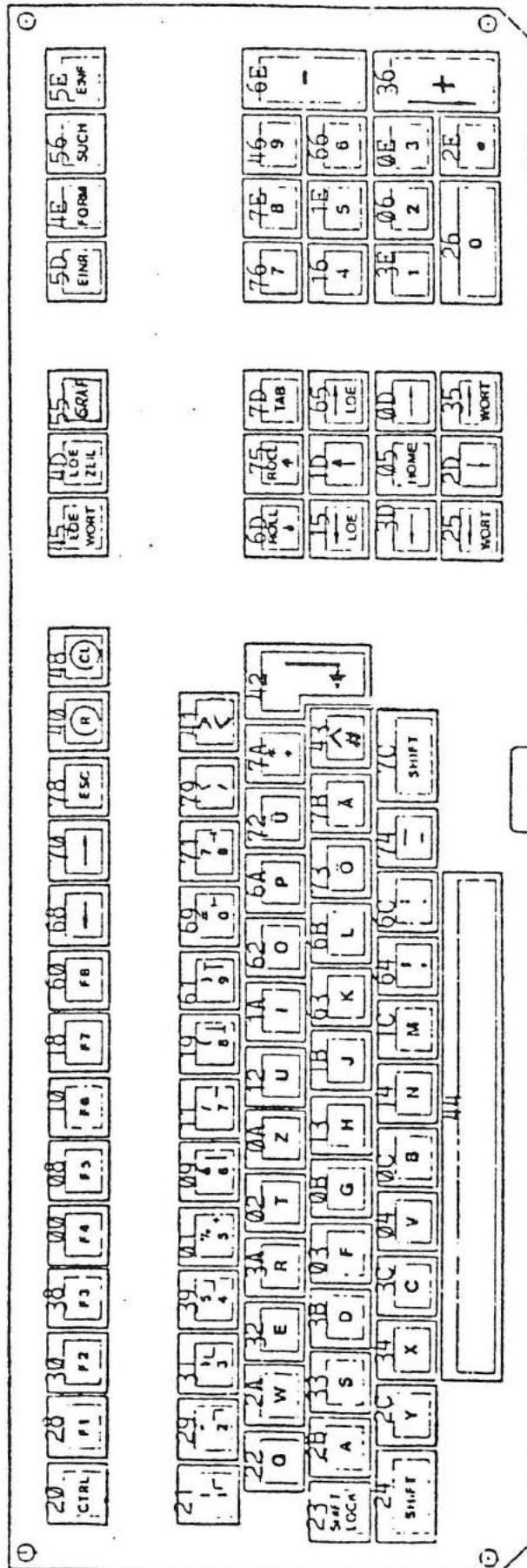
Für Sondertasten, die nicht vom Tastatur-Treiber durchgegeben sondern intern verarbeitet werden, sind folgende Codes reserviert:

SHIFT-Taste	C1H
SHIFT-Lock-Taste	COH
Wiederholtaste	C2H
Control-Taste	C5H

TET 3030

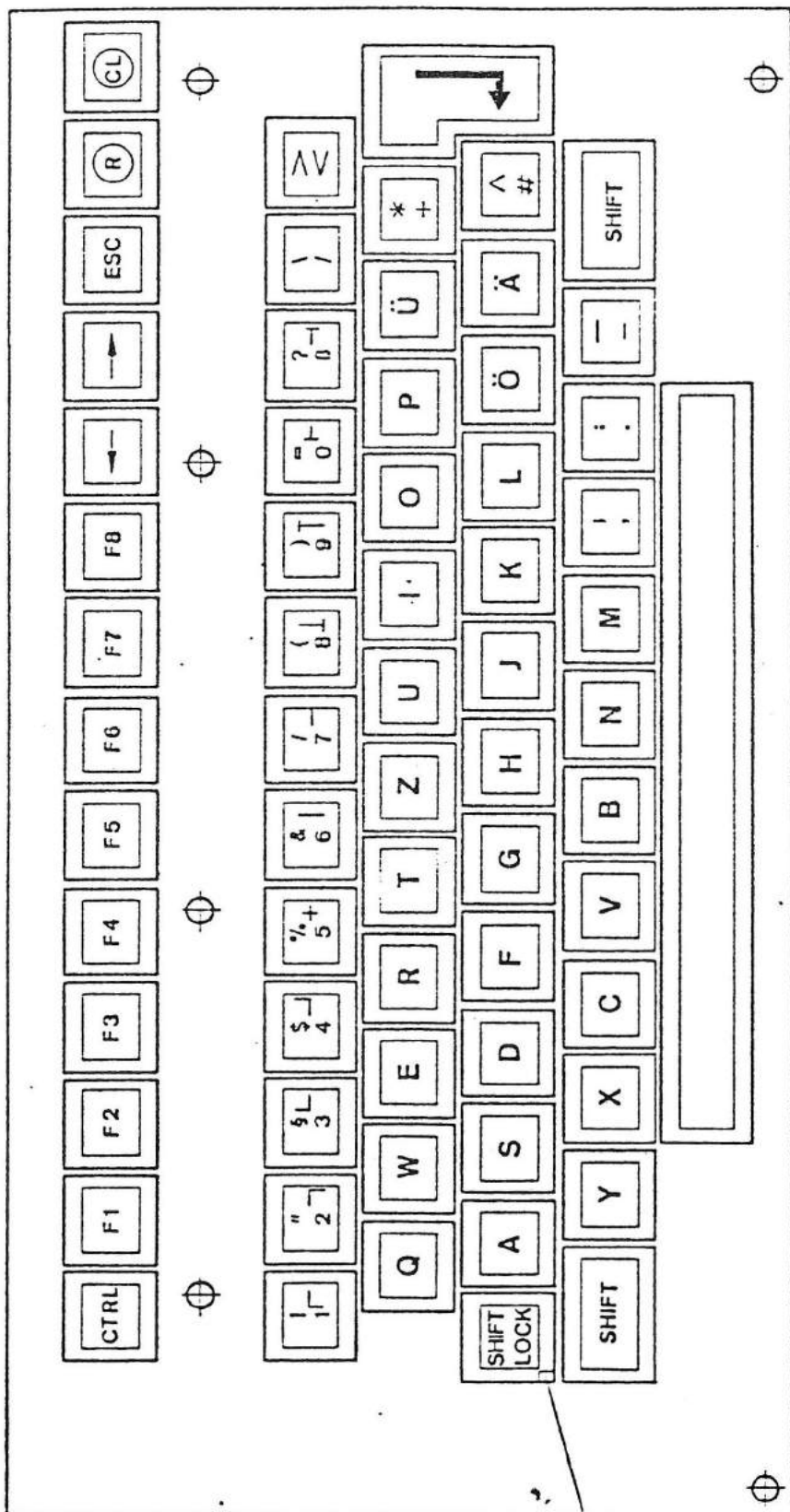
AUF WACHSTUM PROGRAMMIERT!

Matrix - Werte der Tasten



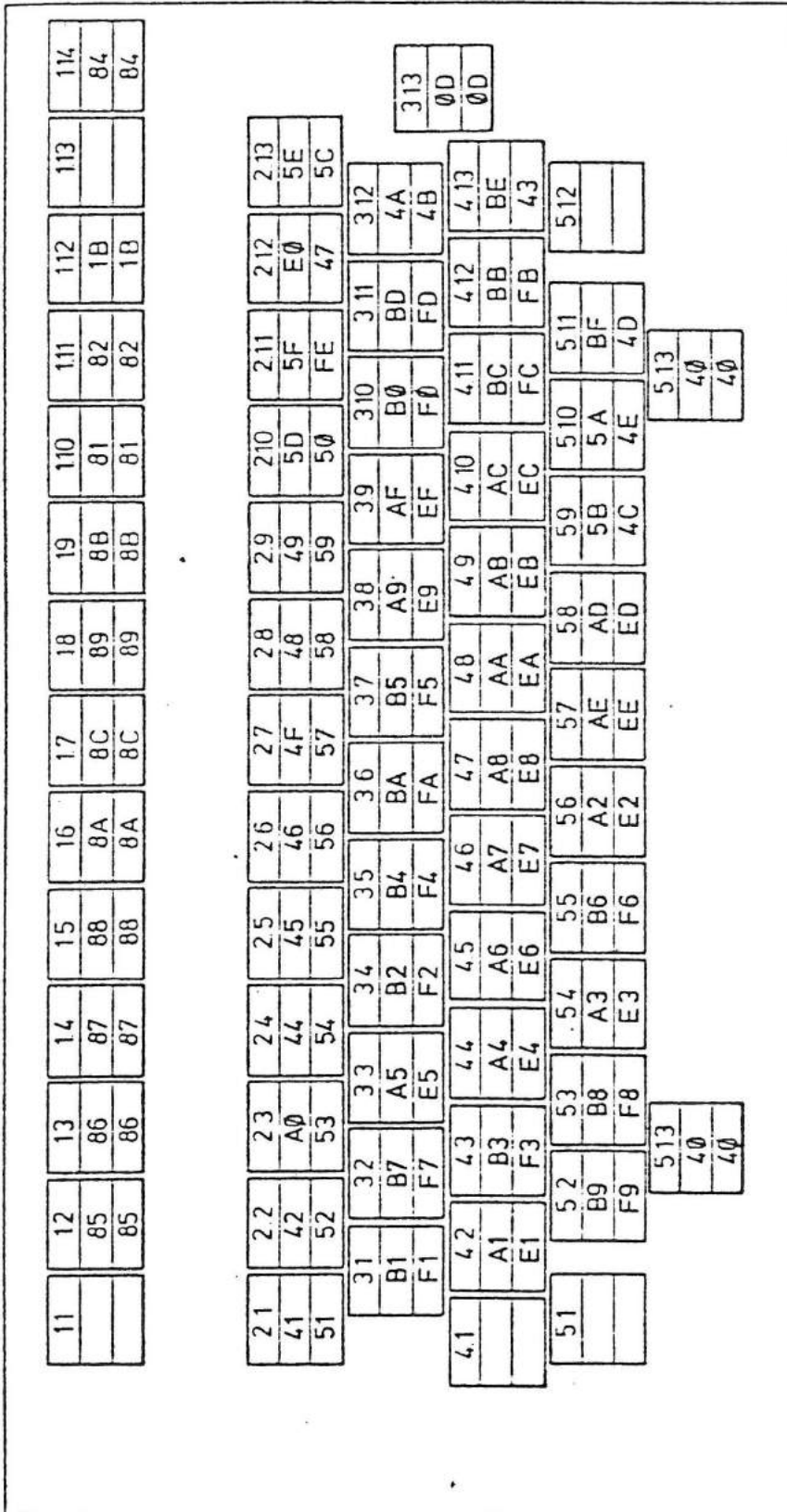
ITT 3030

AUF WACHSTUM PROGRAMMIERT !



TTE 3030

AUF WACHSTUM PROGRAMMIERT !

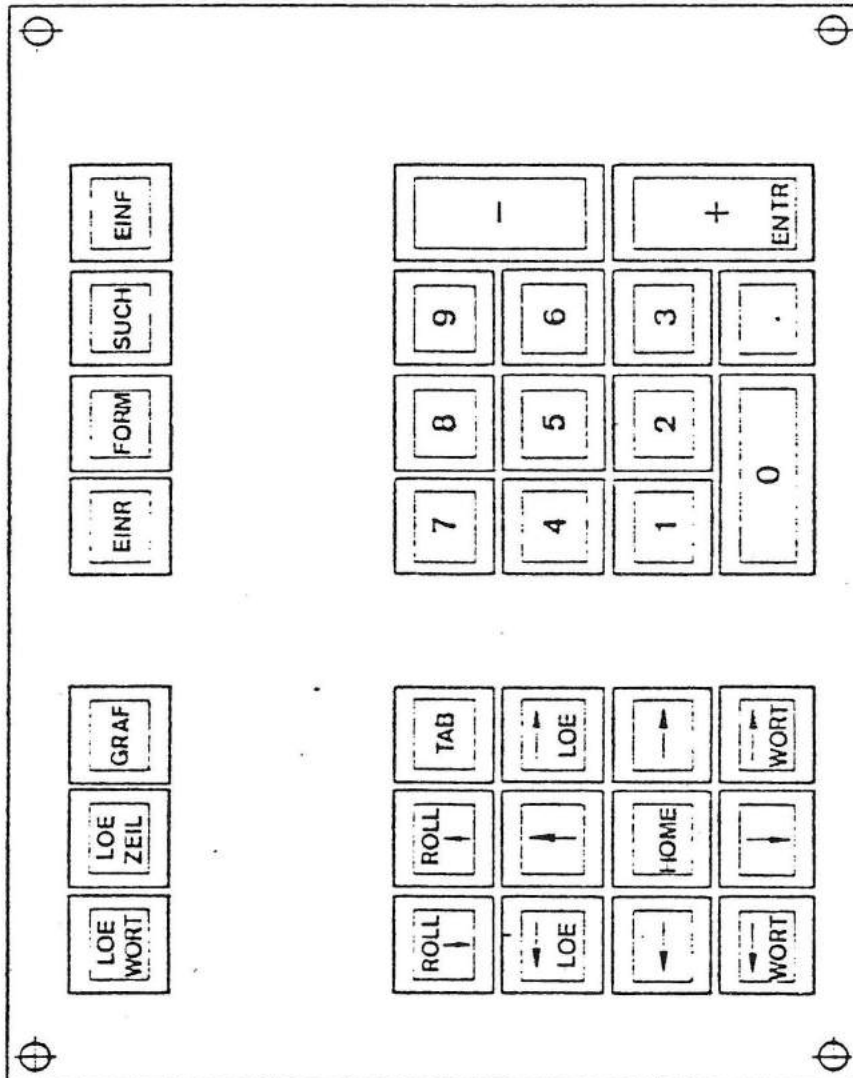


Tastennummern
Code (shift)
Code (unshift)



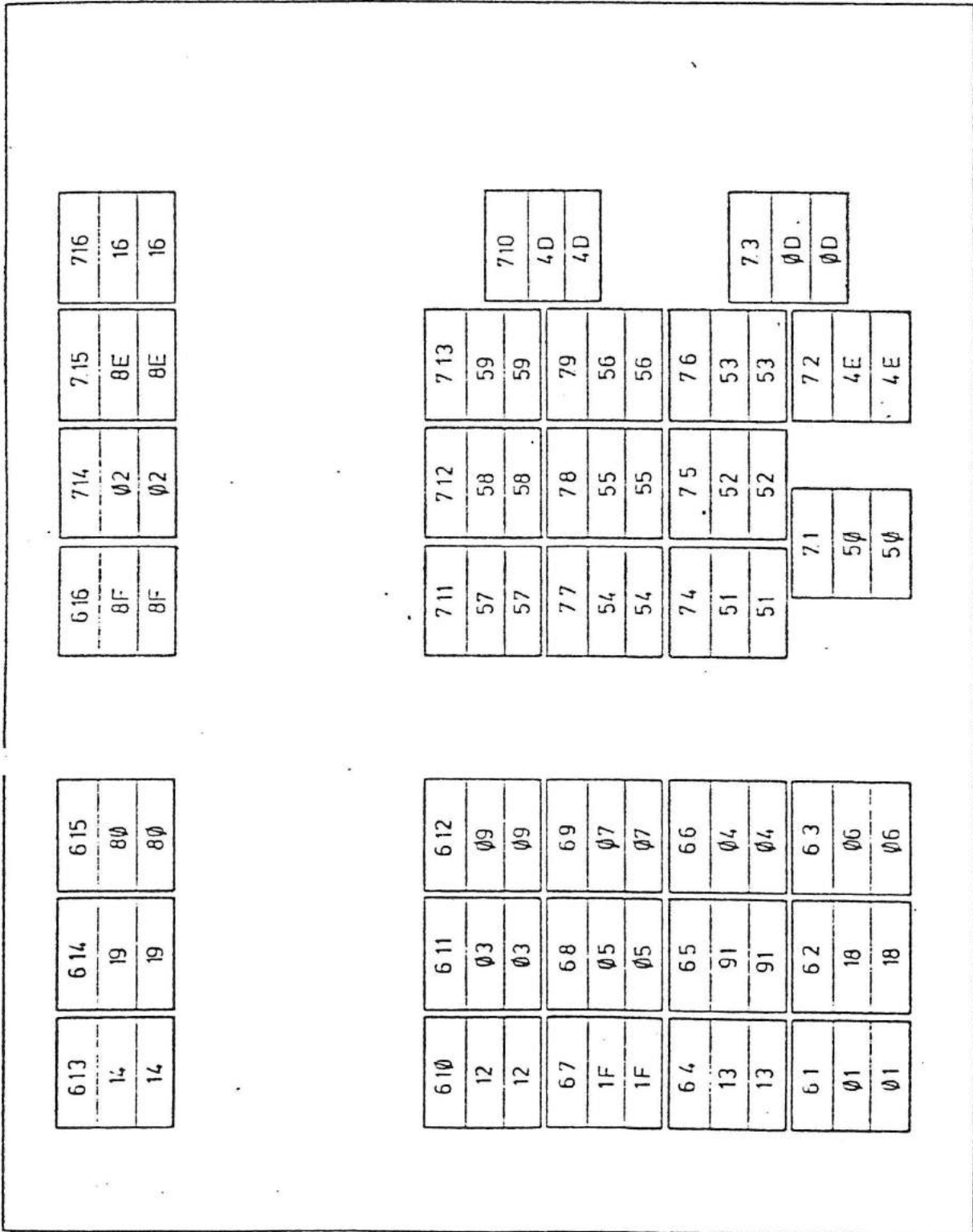
ITT 3030

AUF WACHSTUM PROGRAMMIERT !



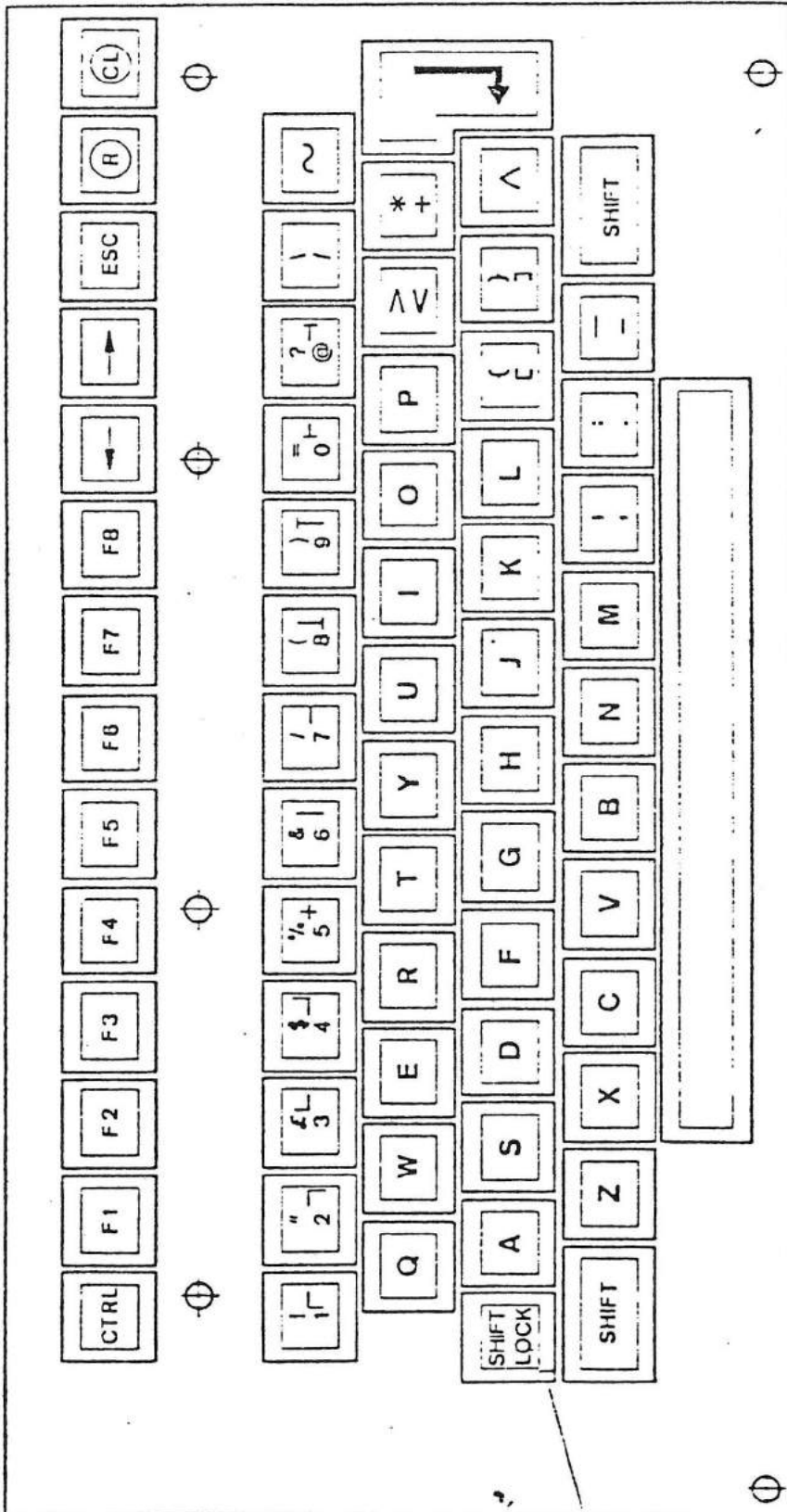
ITT 3030

AUF WACHSTUM PROGRAMMIERT !



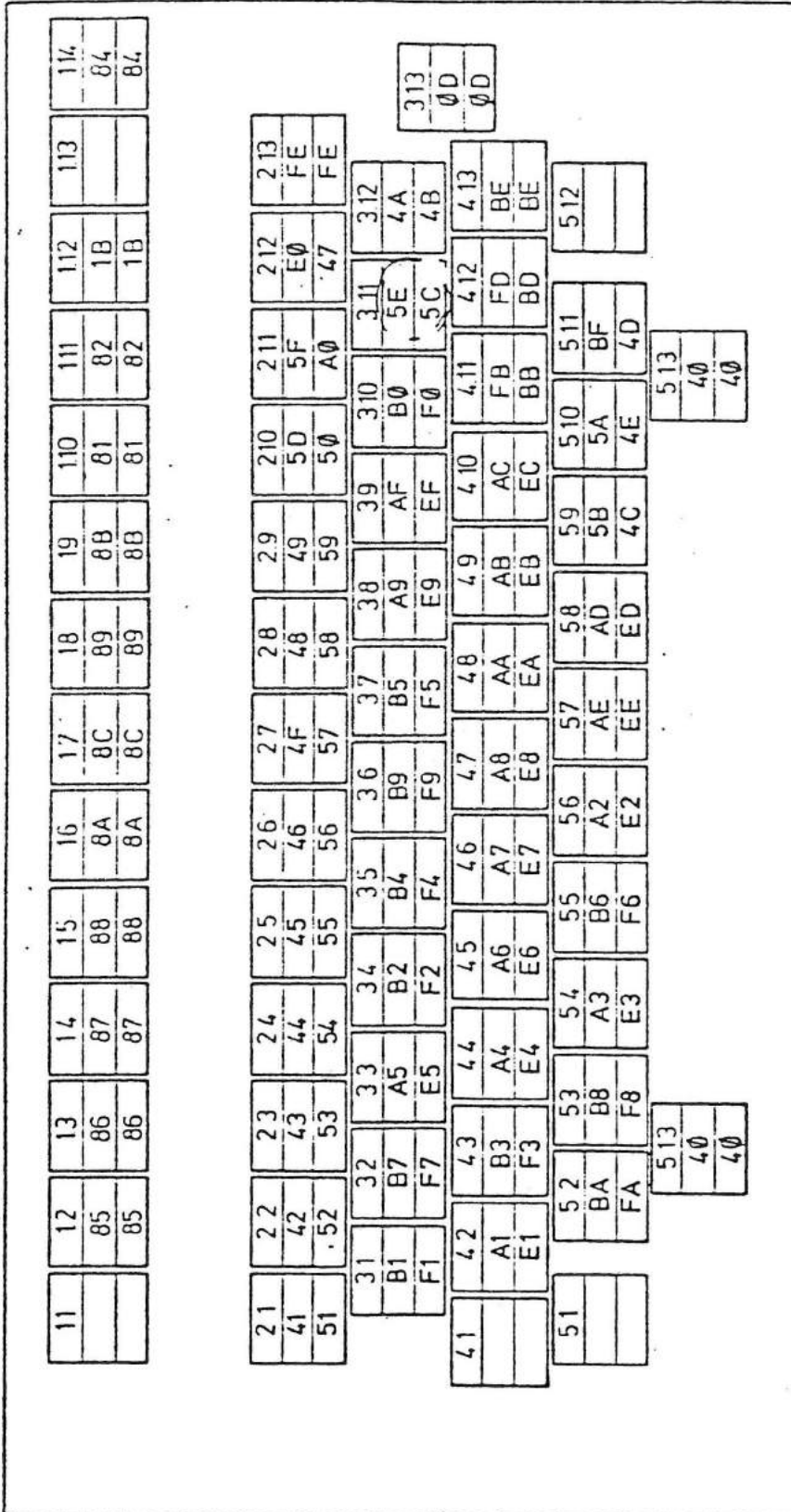
ITT 3030

AUF WACHSTUM PROGRAMMIERT !



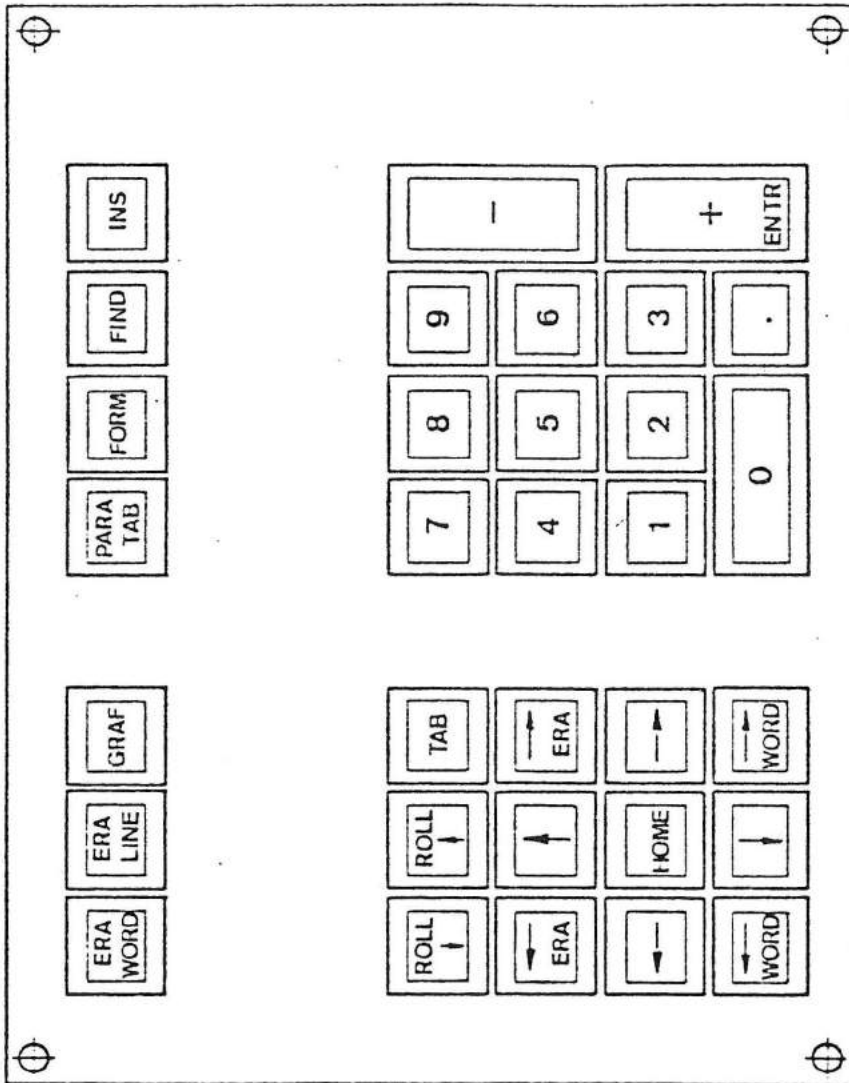
TET 3030

AUF WACHSTUM PROGRAMMIERT !



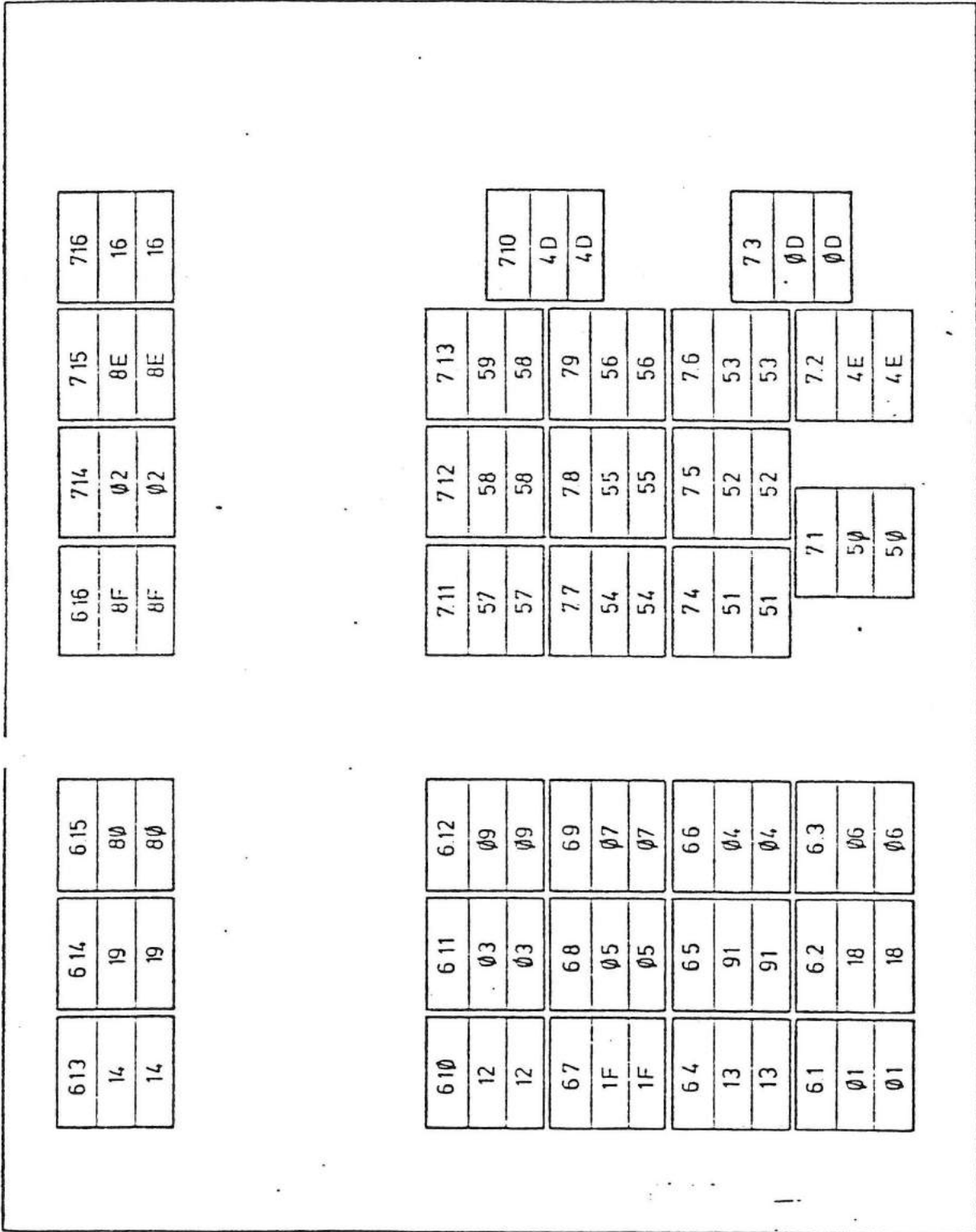
ITT 3030

AUF WACHSTUM PROGRAMMIERT !



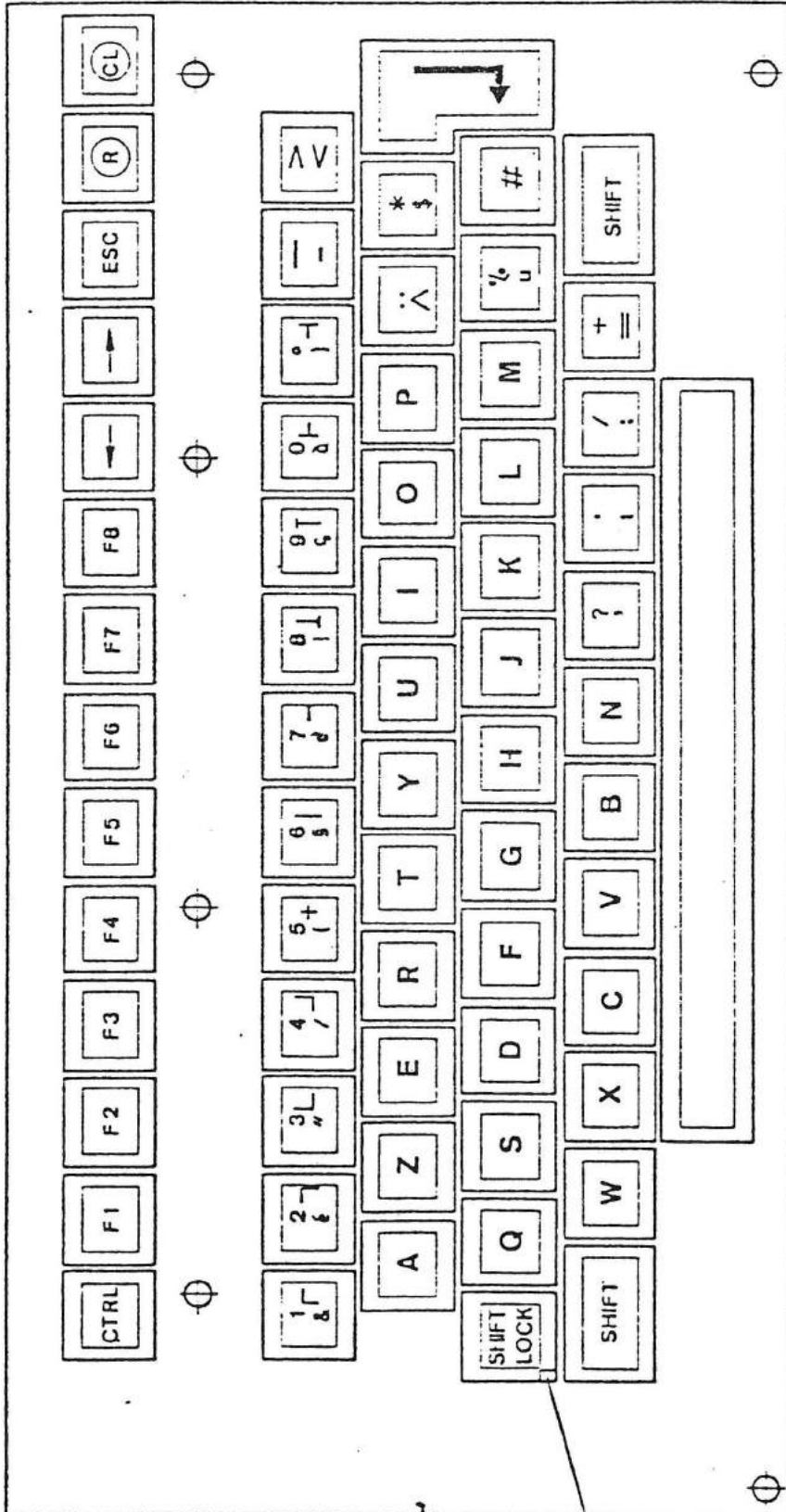
ITT 3030

AUF WACHSTUM PROGRAMMIERT !



TEF 3030

AUF WACHSTUM PROGRAMMIERT!

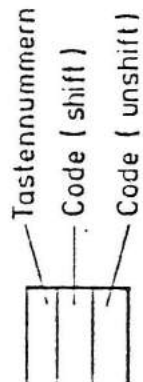
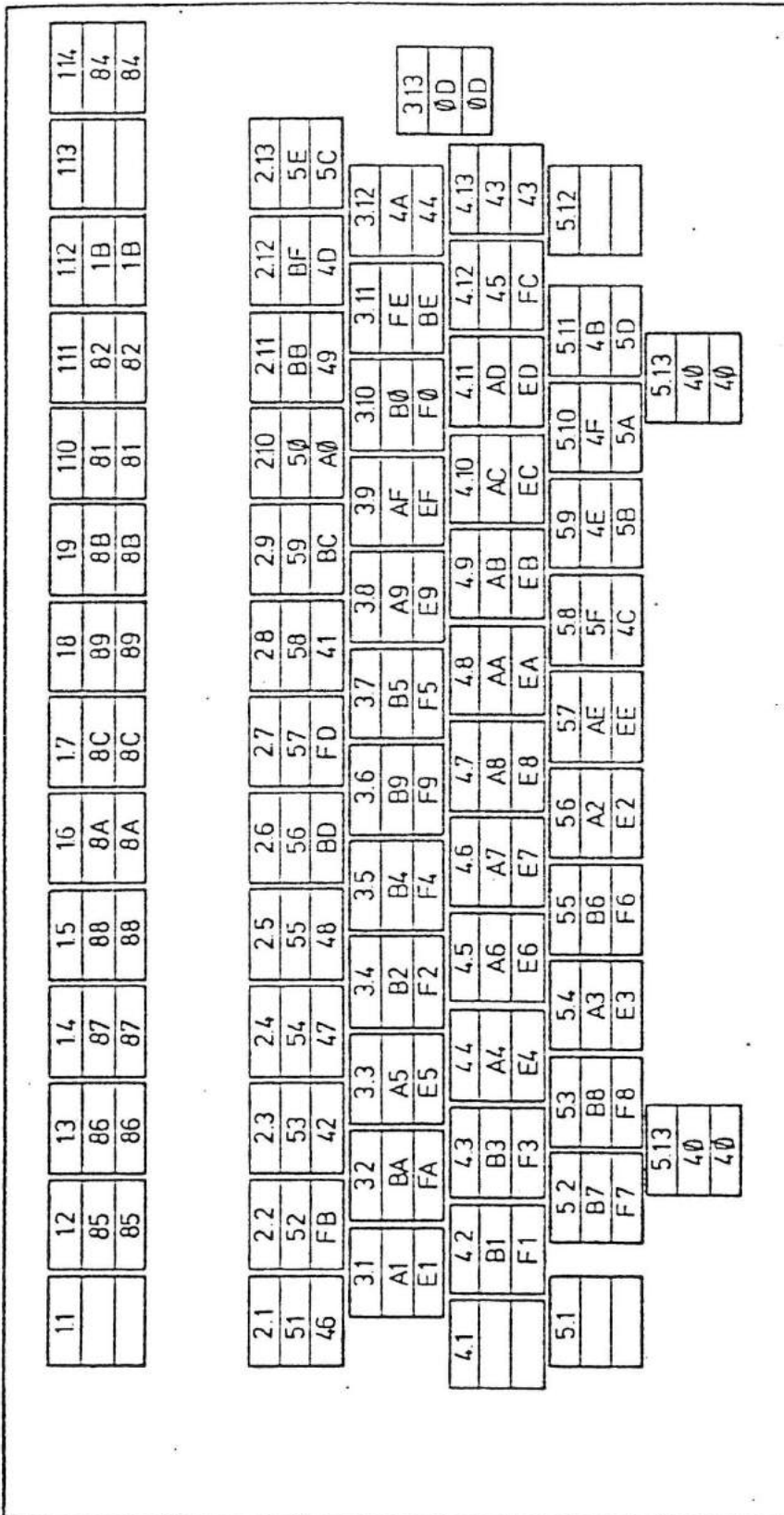


Technik der Welt



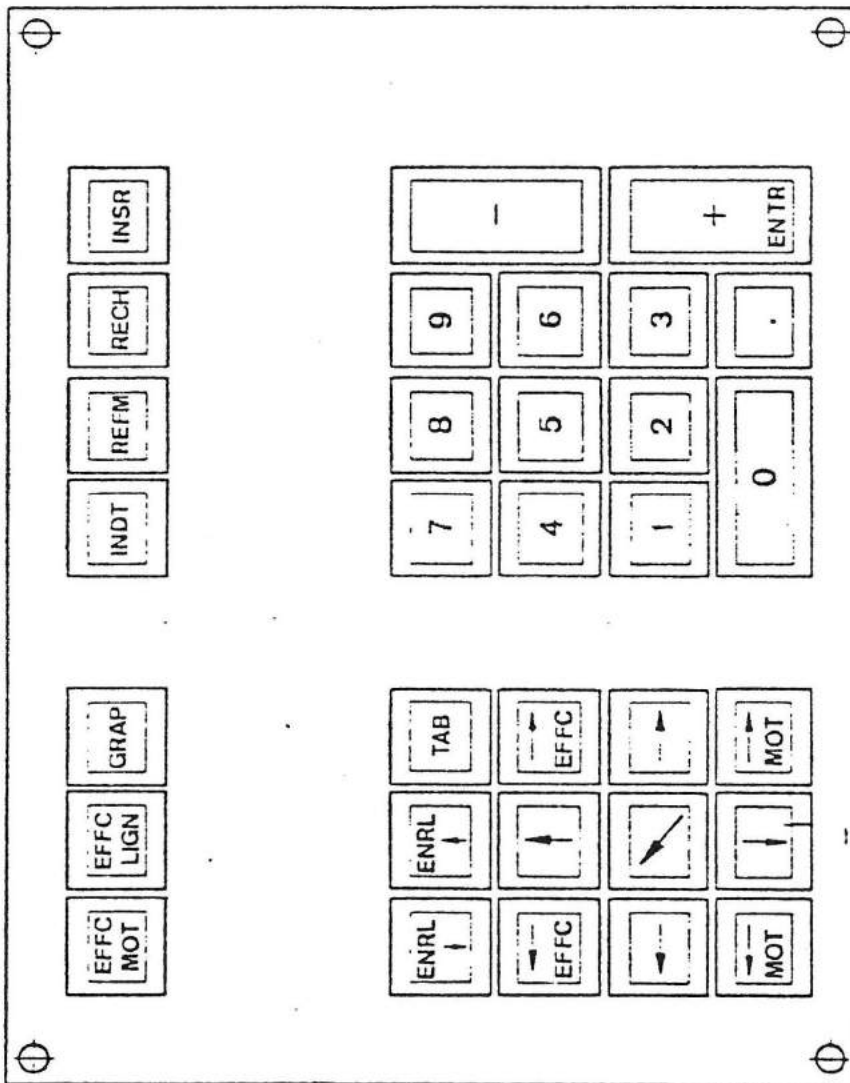
ITT 3030

AUF WACHSTUM PROGRAMMIERT !



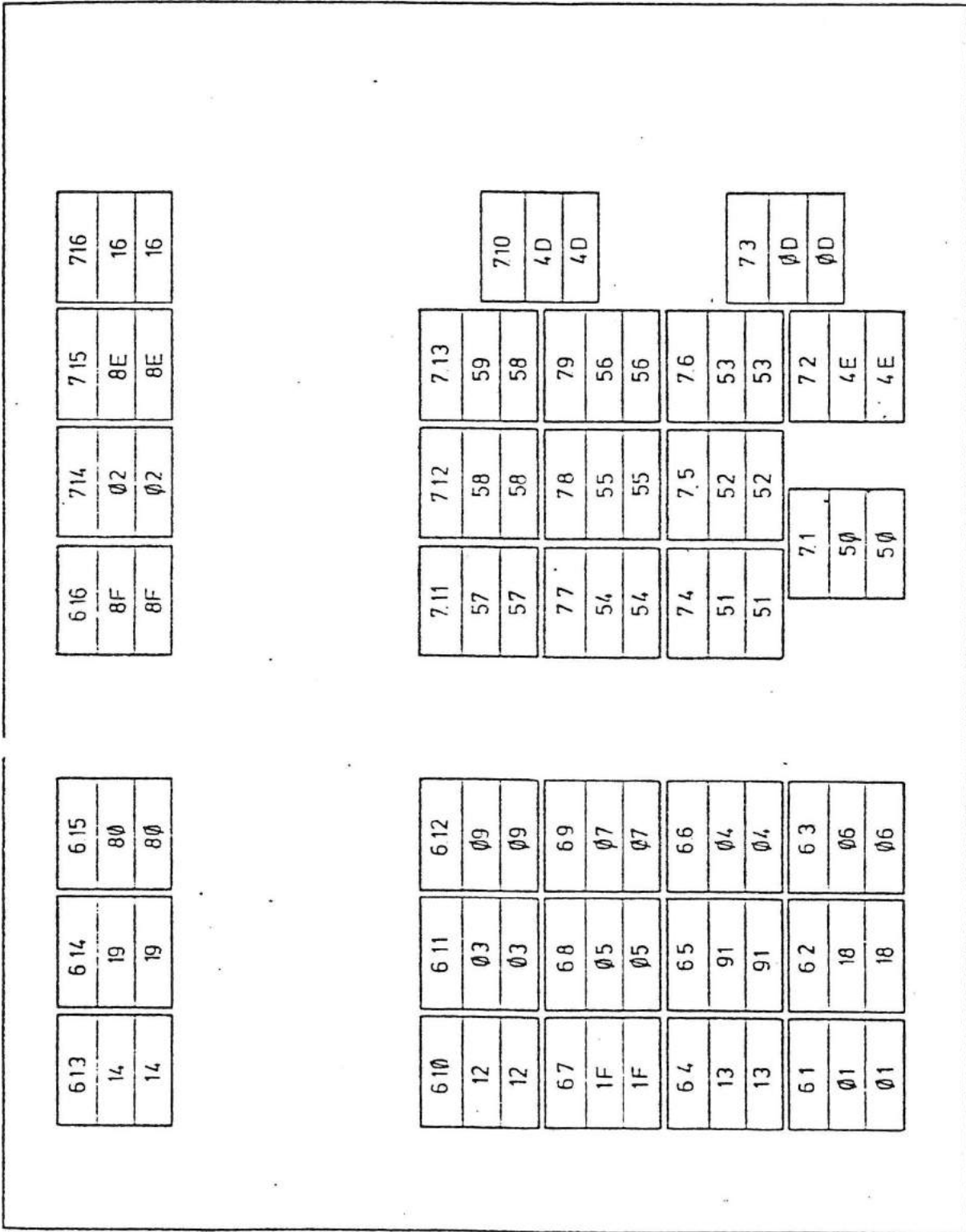
ITT 3030

AUF WACHSTUM PROGRAMMIERT !



TECH 3030

AUF WACHSTUM PROGRAMMIERT !



Bei allen Funktionen werden alle nicht genannten Register, mit Ausnahme des Registers A und des Bedingungsregisters, unverändert zurückgegeben.

II. Zeichenweiser Betrieb mit Steuerzeichenerkennung

Es wird jeweils 1 Zeichen übergeben (in Register C), wobei die unten aufgeführten Steuerzeichen interpretiert werden. Alle Register, mit Ausnahme des Registers A und des Bedingungsregisters, werden unverändert zurückgegeben. Bei einigen Steuerfunktionen muß ein Escape-Zeichen (1BH) vorausgehen.

Folgende Steuerzeichen werden wie beschrieben interpretiert:

07 BELL (BEL)

Die Clear-Lampe leuchtet auf und die Hupe ertönt. Beide Signale dauern ca. 1 Sek. an.

08 Cursor links (BS)

Der Cursor wandert um eine Stelle nach links.

0A Zeilenschaltung (LF)

Der Cursor wandert um eine Zeile nach unten. Ist der Cursor in der untersten Zeile wird das Bild um eine Zeile nach oben gerollt und die unterste Zeile gelöscht. Die Spaltenposition des Cursors bleibt dabei unverändert.

0C Grundstellung ohne Lupe

Der Bildschirmcontroller wird initialisiert, der Bildschirm-Treiber wird grundgestellt. Die Cursorposition ist in der linken unteren Ecke (letzte Zeile, erste Spalte). Der Bildschirm wird gelöscht. War Lupe eingeschaltet, bleibt dieser Mode erhalten, ebenso Zentriermode.

0D Ende Zeile

Der Cursor wird an den Anfang der Zeile gestellt.

11 Lupe einschalten

Lupe wird eingeschaltet, Bildschirm gelöscht, Cursor steht in der linken unteren Ecke. War Zentriermode eingeschaltet, wird dieser zurückgesetzt. Dergleichen alle Hintergrund-Modifikationen.

12 Endecode Zeichenmodifikation

Hintergrundbits werden grundgestellt.

13 Lupe ausschalten

Lupe wird ausgeschaltet, Bildschirm gelöscht, Cursor steht in der linken unteren Ecke. War Zentriermode eingeschaltet, wird dieser zurückgesetzt. Desgleichen alle Hintergrund-Modifikationen.

1C Invers einschalten

Alle folgenden Zeichen werden invers dargestellt.

Steuerzeichen mit Escape:

1B, 10, XX Blanks ausgeben

Es werden XX Blanks ausgegeben, der Cursor wandert entsprechend weiter.

1B, 11 Cursor home

Der Cursor wird in die obere linke Ecke gestellt (erste Zeile, erste Spalte).

1B, 12 Cursor ausschalten

Der Cursor wird ausgeschaltet, die Position jedoch weiterverwaltet.

1B, 13 Cursor einschalten

Der Cursor wird eingeschaltet.

1B, 14 ROLL-UP

Das Bild wird um eine Zeile nach oben gerollt. Die oberste Zeile wird unten angehängt. Die relative Cursorposition bleibt unverändert (Zeilen- und Spaltenposition bleiben gleich).

Diese Funktion sollte nicht in Verbindung mit der Zentrierfunktion angewendet werden.

1B, 15, XX, YY Zentrieren

Mit dieser Funktion kann eine erwünschte Schreibbreite mit XX (ungerade Zahl, da Spalten bei 0 beginnen), bzw. Schreibhöhe mit YY (gerade Zahl, da Zeilen mit 1 beginnen) eingestellt werden. Das Bild wird automatisch in die neue Bildmitte zentriert. Der Cursor wird in die neue Home-Position gebracht (nach wie vor Zeile 1, Spalte 0). Der Bildschirmteil außerhalb des gewählten "Fensters" ist dann nicht mehr zugänglich. Die Funktion Grundstellen (Formfeed 0C) läßt den zentrierten Zustand bestehen, d. h. der Cursor steht in der linken unteren Ecke des gewählten Bereiches. Es wird allerdings der gesamte Bildschirm gelöscht.

1B, 16, XX, YY Cursor positionieren

Der Cursor wird auf Zeile XX, Spalte YY positioniert.

1B, 17 Rest des Bildschirm löschen

Der Bildschirm wird ab der momentanen Cursorposition gelöscht. Die Cursorposition bleibt unverändert.

1B, 18 Rest der Zeile löschen

Die Zeile wird ab der momentanen Cursorposition gelöscht. Die Cursorposition verändert sich nicht.

1B, 19 Rest der Zeile löschen

Die Zeile wird ab der momentanen Cursorposition gelöscht, der Cursor wird auf den Anfang der nächsten Zeile positioniert.

- 1B, 1A Cursor rechts
Der Cursor wird um eine Position nach rechts bewegt.
- 1B, 1C Cursor up
Der Cursor wird um eine Zeile nach oben positioniert. Am oberen Bildrand erfolgt kein Roll-down.
- 1B, 1D, XX, Y \dot{Y}
Es werden XX Zeichen mit dem Code YY ausgegeben.
- 1B, 1E, XX Hintergrund setzen
Es wird ab sofort jedes Zeichen mit dem angegebenen Hintergrund ausgegeben. Die Belegung ist identisch mit RB bei der Funktion Display-Status lesen.
- 1B, 1F, XX, YY Cursor positionieren mit Offset
Wie 1B, 16, XX, YY, jedoch müssen Zeile und Spalte mit einem Offset von 20 H übergeben werden.

Alle anderen Zeichen werden auf dem Bildschirm ausgegeben und der Cursor wird um eine Position weiterbewegt.

Die Schnittstelle ohne Steuerzeichenerkennung (Adresse FE76H) gibt alle Codes direkt auf den Bildschirm.

III. Tastaturstatus abfragen

Mit dieser Routine kann ermittelt werden, ob eine Taste angeschlagen wurde oder nicht.

Aufruf: keine Parameter

Rückgabe: RA = 0, ZERO = 1 => keine Taste
RA = FF, ZERO = 0 => Taste angeschlagen
CARRY = 1 => Übertragungsratenfehler
CARRY = 0 => kein Übertragungsratenfehler

IV. Warten auf Taste

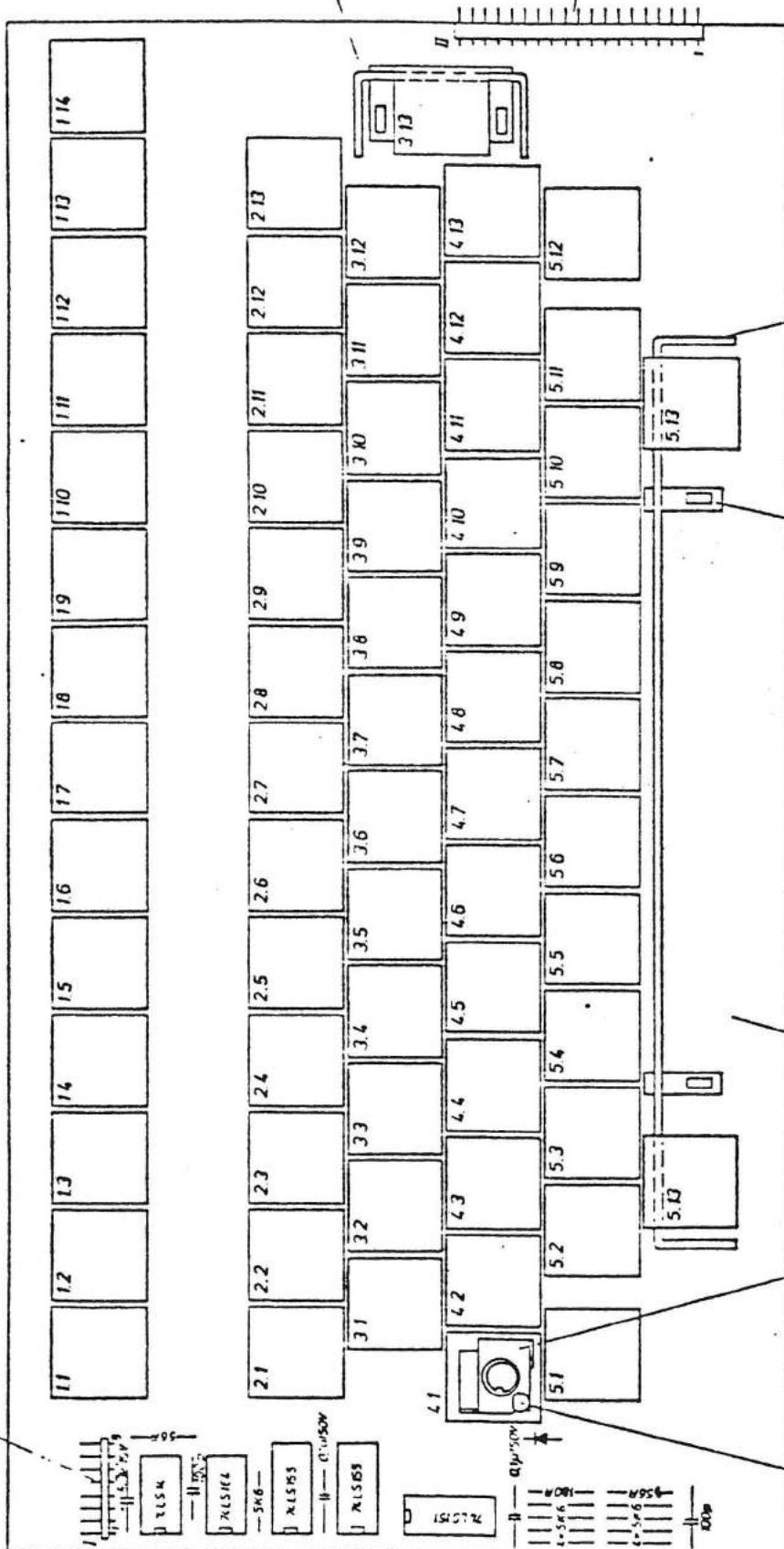
Diese Routine wartet bis eine Taste angeschlagen wird, und gibt dann den Tastencode zurück (Besonderheiten siehe Generelle Initialisierung, Bedeutung Parameter X).

Aufruf: keine Parameter

Rückgabe: RA = Tastencode

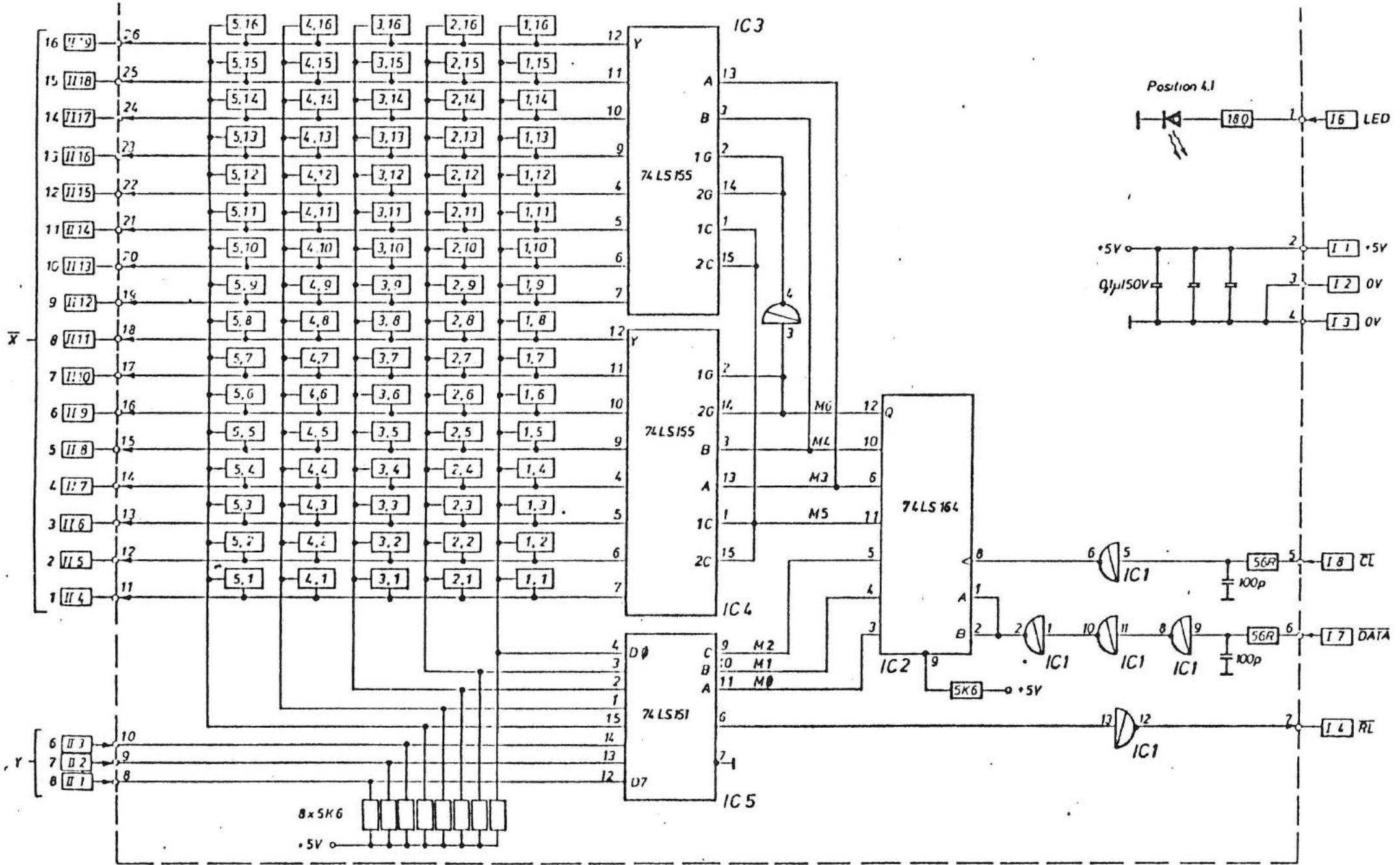
ITT 3030

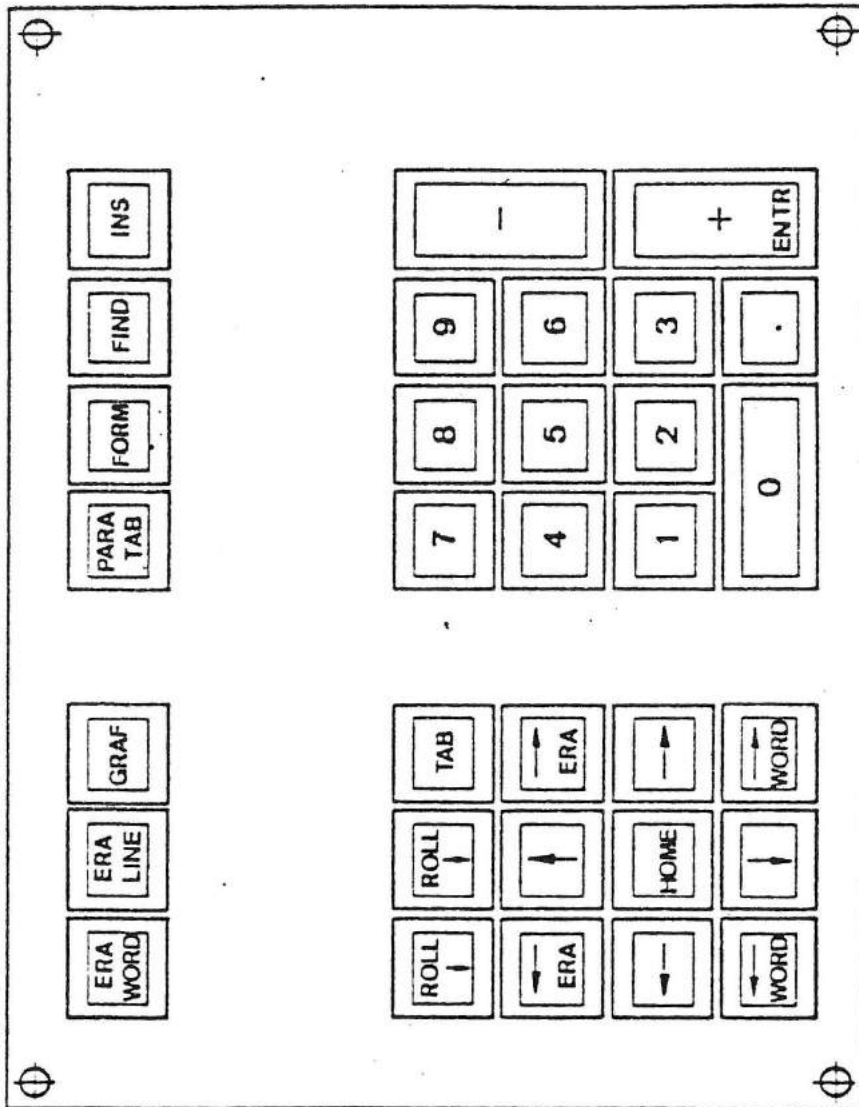
auf Wachstum programmiert



Technik der Welt

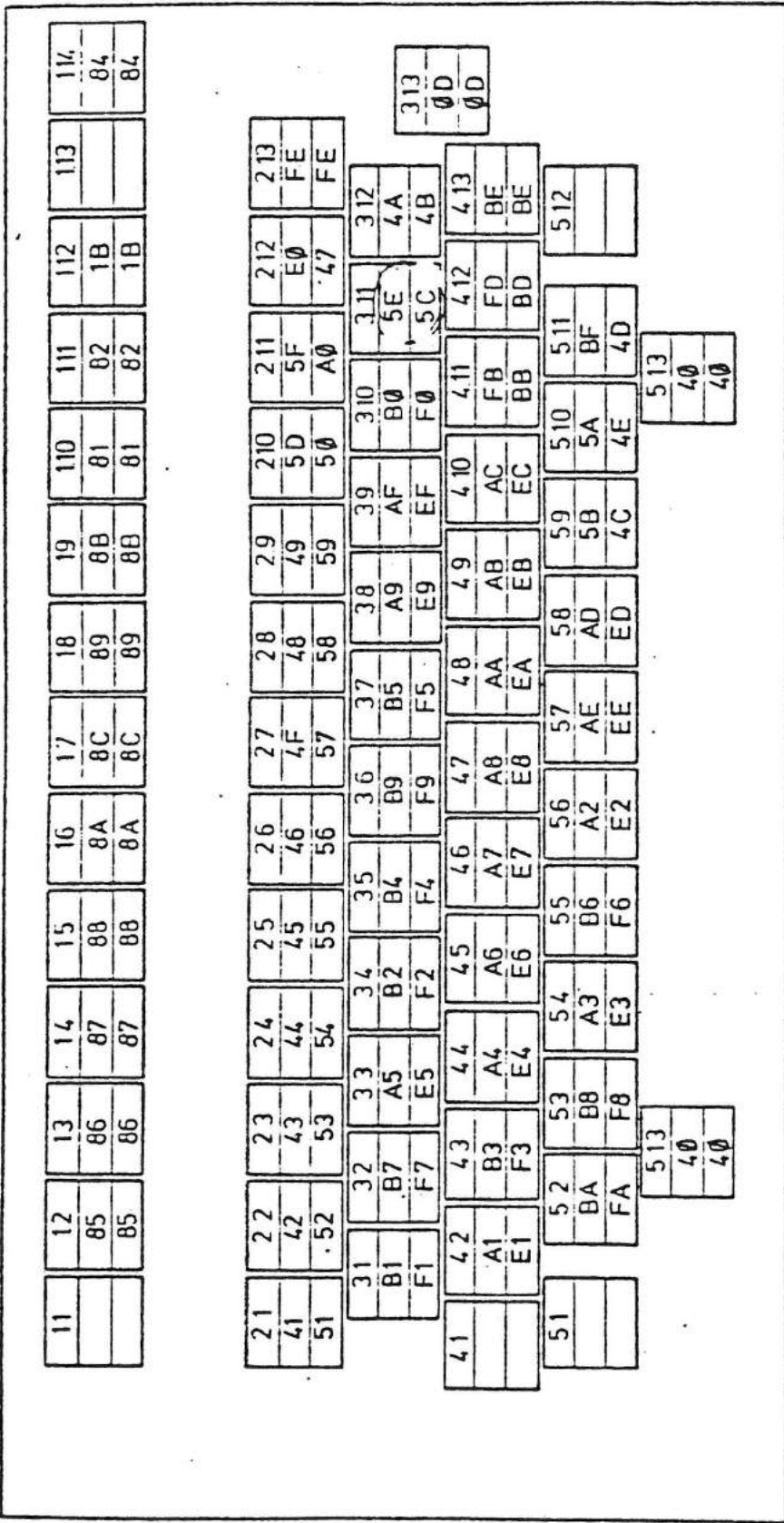
ITT



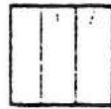


Die Anweisung darf nur verwendet werden
 Personen in der Abteilung für die
 alle Rechte vorbehalten

Ausgabe tag Monat Jahr	22.02 1982	82 Danz Bepr. Harm	18.2 Bols	Items Bols	Zusatztafelur GB Layout	Maßstab 1:1	Arbeitsnr. / Blatt / von
---------------------------------	---------------	-----------------------------	--------------	---------------	----------------------------	----------------	--------------------------------



Tastennummern
 Code (shift)
 Code (unshift)



ASCII - Tabelle

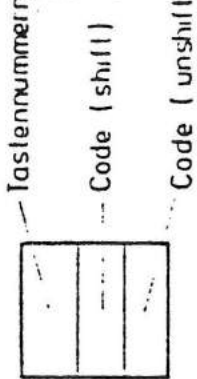
Code Liste 7 Bit USASCII Code : ITT 3030 - GB

Bit 6	#	0	!	0	#	0	!	0	#	!	!	#	!	!	!	#	
Bit 5	#	0	!	0	#	L	!	L	#	0	!	0	#	L	!	L	#
Bit 4	#	0	!	L	#	0	!	L	#	0	!	L	#	0	!	L	#
Dez. Äqui.	#	0	1	#	2	3	#	4	5	#	6	7	#				
Bit (3..0)	0000	#	0	#	.	.	SP	0	@	2	'	0	#				
	0001	#	1	#	.	.	!	1	A	Q	h	o	#				
	0010	#	2	#	.	.	"	2	B	R	o	i	#				
	0011	#	3	#	.	.	#	3	C	S	o	l	#				
	0100	#	4	#	.	.	\$	4	D	T	o	o	#				
	0101	#	5	#	.	.	%	5	E	U	e	u	#				
	0110	#	6	#	.	.	&	6	F	V	r	v	#				
	0111	#	7	#	.	.	'	7	G	W	s	w	#				
	1000	#	8	#	.	.	(8	H	X	S	x	#				
	1001	#	9	#	.	.)	9	I	Y	i	y	#				
	1010	#	10	#	.	.	#	:	J	Z	j	z	#				
	1011	#	11	#	.	.	+	;	K	[k	[#				
	1100	#	12	#	.	.	,	<	L	\	l	\	#				
	1101	#	13	#	.	.	-	=	M]	m]~	#				
	1110	#	14	#	.	.	.	>	N	^	n	^	#				
	1111	#	15	#	.	.	/	?	O	_	o	_	#				

Das Diagramm zeigt die Anordnung der Tasten auf dem Tastaturfeld. Die Tasten sind in Gruppen angeordnet und sind durch Linien verbunden. Die Tasten sind in Gruppen angeordnet und sind durch Linien verbunden.

613	614	615	716
14	19	80	16
14	19	80	16

610	611	612	711	712	713	710
12	03	09	57	58	59	40
12	03	09	57	58	58	40
67	68	69	77	78	79	
1F	05	07	54	55	56	
1F	05	07	54	55	56	
64	65	66	74	75	76	
13	91	04	51	52	53	
13	91	04	51	52	53	
61	62	63	71	72	73	
01	18	06	50	4E	00	
01	18	06	50	4E	00	



ITT 3030

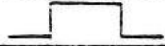
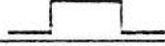

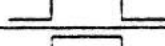
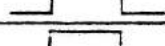


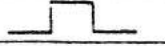

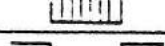
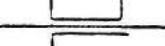

auf Wachstum programmiert

TASTATUR-ANPASSUNG

Die Tastatur-Anpassung dient zur Parallel-Seriell-Wandlung der sieben SCAN-Adressen des Bausteins 8278 im Kombi-Interface.

Als Startsignal dient KCL, das vom 8278 erzeugt wird. Ein Zähler 74 LS 193 wird damit auf den Wert "6" geladen und gleichzeitig ein Oszillator (770 KHz) gestartet. Der erste Takt übernimmt die parallel anstehende Information am 74 LS 165 und kippt mit seiner Rückflanke ein Flip-Flop, so daß die nächsten 8 Takte die geladene Information seriell herausschieben. Nach diesen 9 Takten stoppt der Zähler beim Übergang von 15 nach \emptyset den Oszillator durch das Zurücksetzen des FF D2. Mit dem nächsten KCL-Signal läuft der Vorgang von vorne ab.

Ein zweiter Oszillator dient zum Ansteuern einer Hupe. Dieses Signal wird per Software vom Kombi-Interface erzeugt.

PIN	Bezeichnung	Diagramm	Erläuterungen	Blatt Nr.
01	+ 5 V			
02	0V			
03	M \emptyset		Matrix Scan Ausgang \emptyset	
04	M1		Matrix Scan Ausgang 1	
05	M2		Matrix Scan Ausgang 2	
06	M3		Matrix Scan Ausgang 3	
07	M4		Matrix Scan Ausgang 4	
08	M5		Matrix Scan Ausgang 5	
09	M6		Matrix Scan Ausgang 6	
10				
11	KCL		Key Clock	
12	DATA		Serielle Daten (M \emptyset -M6)	
13	CL		Takt-Schieberegister	
14	CN		General Null	
15	Hupe		Hupe	



00



01



02



03



04



05



06



07



08



09



0A



0B



0C



0D



0E



0F



10



11



12



13



14



15



16



17



18



19



1A



1B



1C



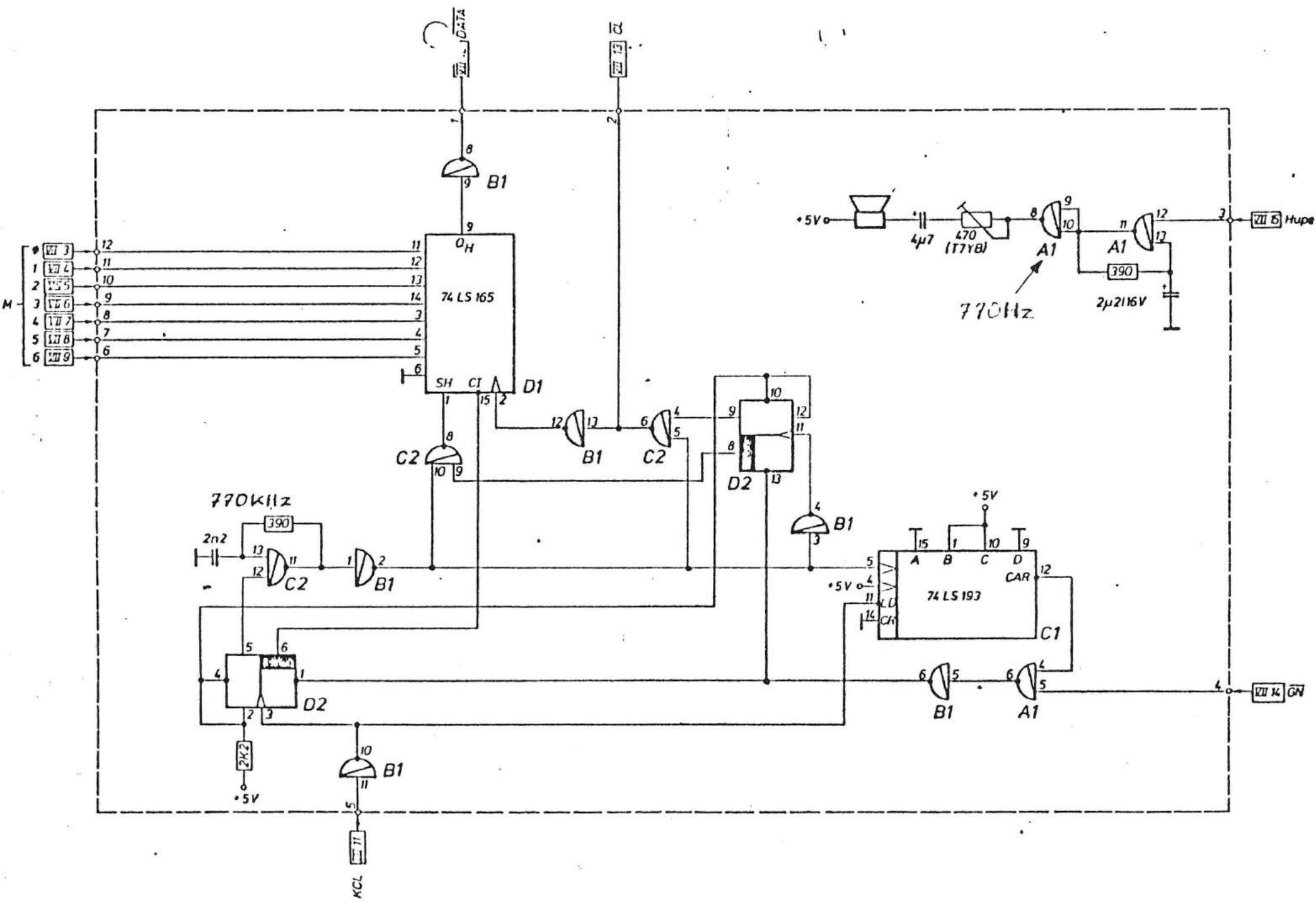
1D

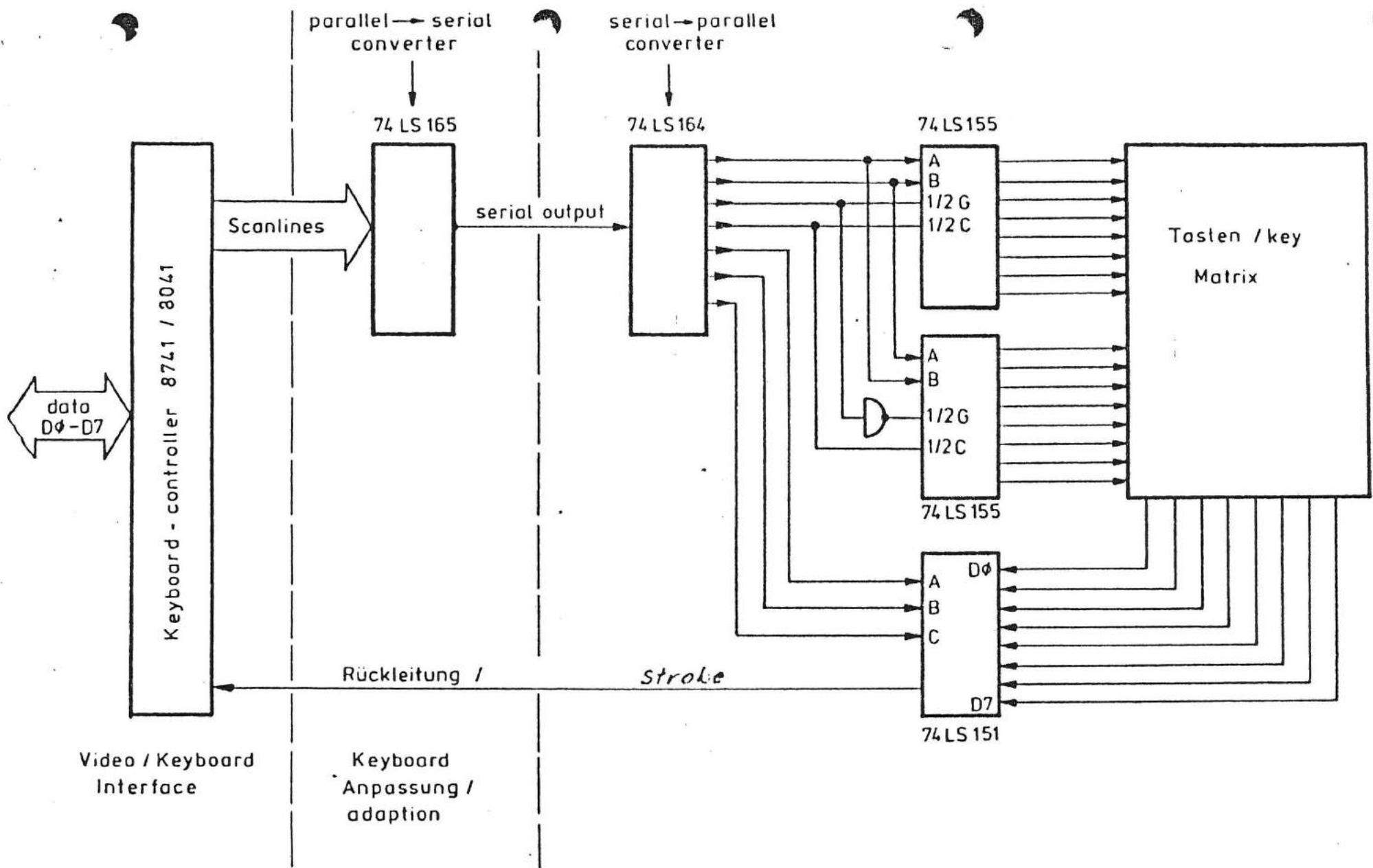


1E



1F



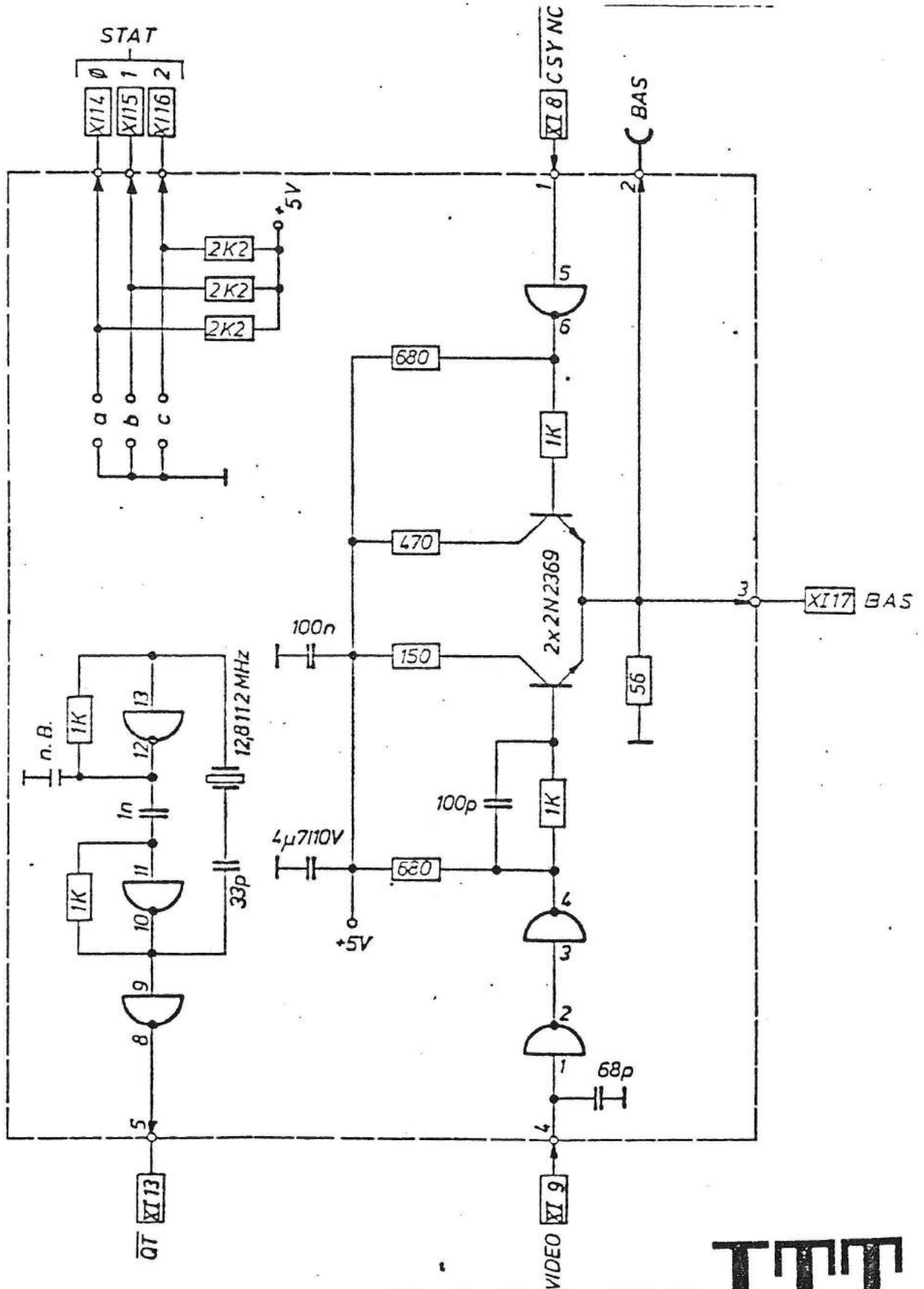


Tastatur - Abfrage ITT 3030
 Keyboard-scanning ITT 3030

ITT 3030

auf Wachstum programmiert

Monitor-Adapter

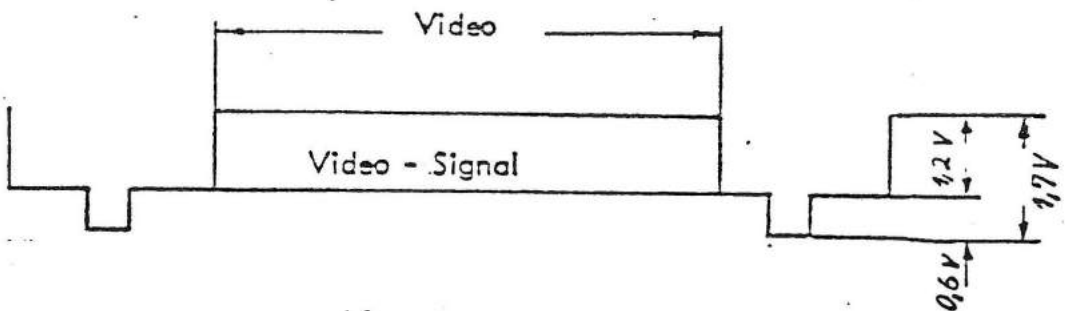


CSYNC		Composite-Synchronisier-Signal
Video		Video-Signal
\overline{QT}		Quarztakt
STAT \emptyset		Statussignal \emptyset
STAT 1		Statussignal 1
STAT 2		Statussignal 2
BAS		Bildaustastensynchronisierungssignal

1. Stromversorgung: + 5V \leq 30 mA

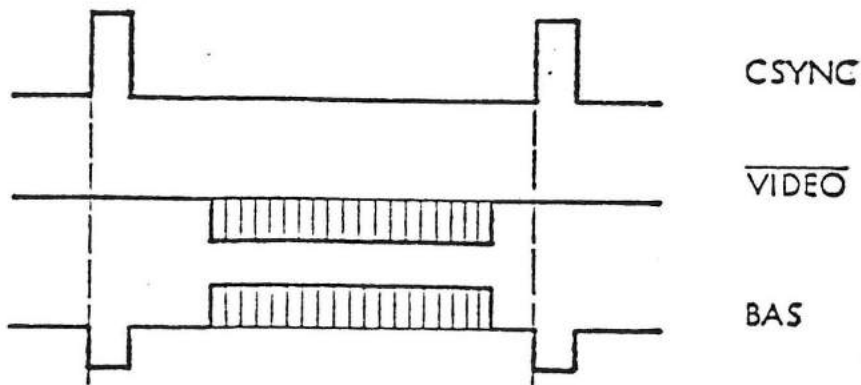
2. Zeitzentrale: 12,8112 MHz \pm 10⁻⁴ .

3. Spezifikation BAS-Signal (Leerlauf)



Die Ausgangsspannung des BAS-Signals ist durch Festwiderstände eingestellt auf 1,7 Vss. Dabei ergibt sich folgende Amplituden-Aufteilung:

- 0 - 0,6 V Synchronsignal \pm 15 %
- 0,6 - 1,7 V Video-Signal \pm 15 %



ITT 3030

auf Wachstum programmiert

Spezifikation der Stromversorgung für ITT 3030

I. Allgemeine Eigenschaften:

- alle Strecken kurzschlußfest gegen OV ✓
- Überspannungsschutz bei +5V
- Funkentstörgrad N-12 dB
- VDE 0730 - 2P gerecht
- Uhrentakt 50/100 Hz, umschaltbar, netzsynchron
- Einschaltnullstellung (General Null)
- alle Signale TTL-kompatibel / aktiv low
- Netzausfall-Frühwarnung als Option

II. Elektrische Daten:

Eingangsspannungen: 220V + 10 % / - 15 % - 50 Hz

110V + 10 % / - 15 % - 50 Hz

Aufgenommene Leistung: bei 220V - *min 75 VA*
max 150 VA

Netzsicherung: 1,25 AT / 250 V

Ausgangsspannungen: Die allgemeinen elektrischen Daten der Sekundärkreise sind der beiliegenden Tabelle zu entnehmen.

Einstellvorschrift

Abgleich der +5V Strecke auf

$$5,1 \text{ V} \pm 0,02 \text{ V}$$

bei 220 V Eingangsspannung und 7A Last

Leistungsdaten:

Nach erfolgtem Abgleich gelten folgende Leistungsdaten:

Regel- Strecke V	Strom A	Toleranzbereich		Restwel- ligkeit mVss max.	Ausregelzeit Sprung 10-90% msec	Strombegren- zung A	
		von V	bis A			min.	max.
+ 5	7,0	5,0	5,25	80	3,0		10,0
+ 12	0,8	11,5	12,5	50	0,1		2,25
- 12	0,4	-11,5	-12,5	50	0,1		1,2
+ 12M	2,5	11,5	12,5	100	1,5	3,0	4,0

Überspannungsschutz: + 5V Logikstrecke - 5V-Strecke schaltet ab

Einschalt-Nullstellung: General Null > 80 msec nach Hoch-
laufen der +5V Strecke

Aufbau: Geschlossenes Metallgehäuse; in das
die Elektronik-Platine (1 Leiterplatte)
eingebaut wird.

Lüfter und Kühlflächen sind innerhalb
des Gehäuses untergebracht.

220V-Eingang (Kaltgerätebuchse) über
Netzschalter zu schalten.

Sekundäre Abgänge über 48-pol. Steck-
verbinder nach DIN 41612

Abmessungen siehe Maßbild

Umgebungstemperatur und
Belüftung:

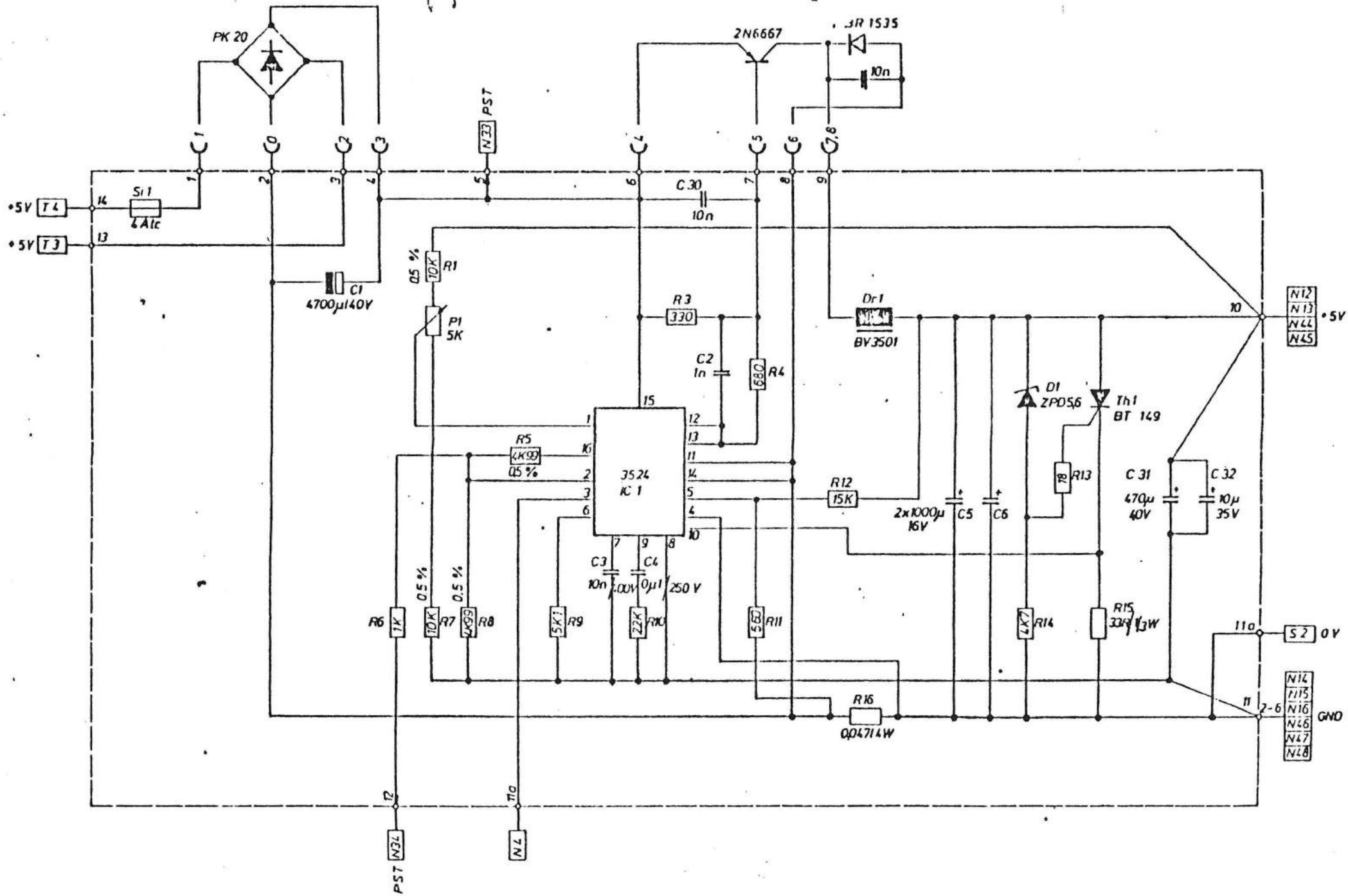
bei 20 m^3 pro Stunde Eintrittstempe-
ratur 60° C

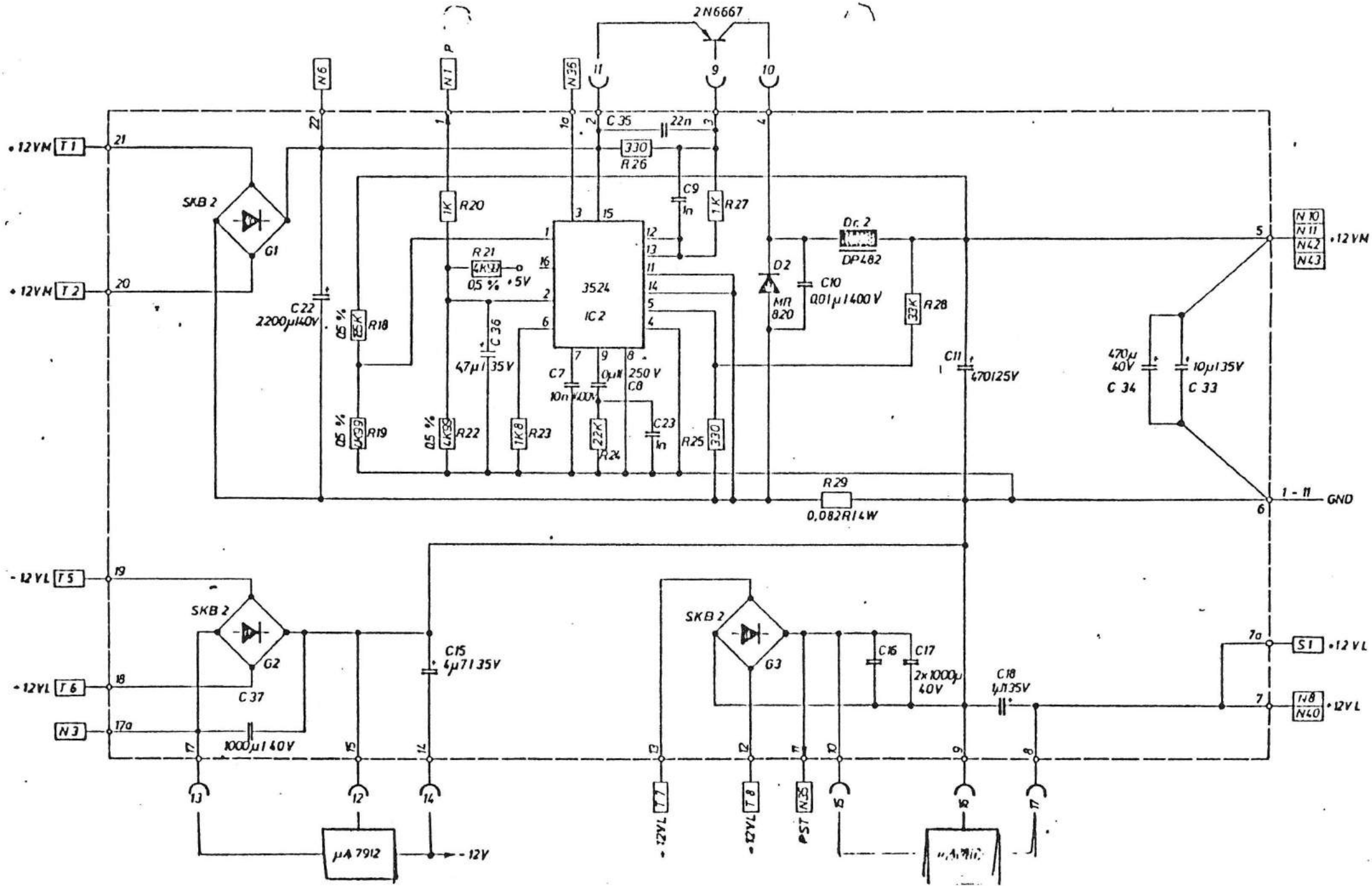
Lagertemperatur:

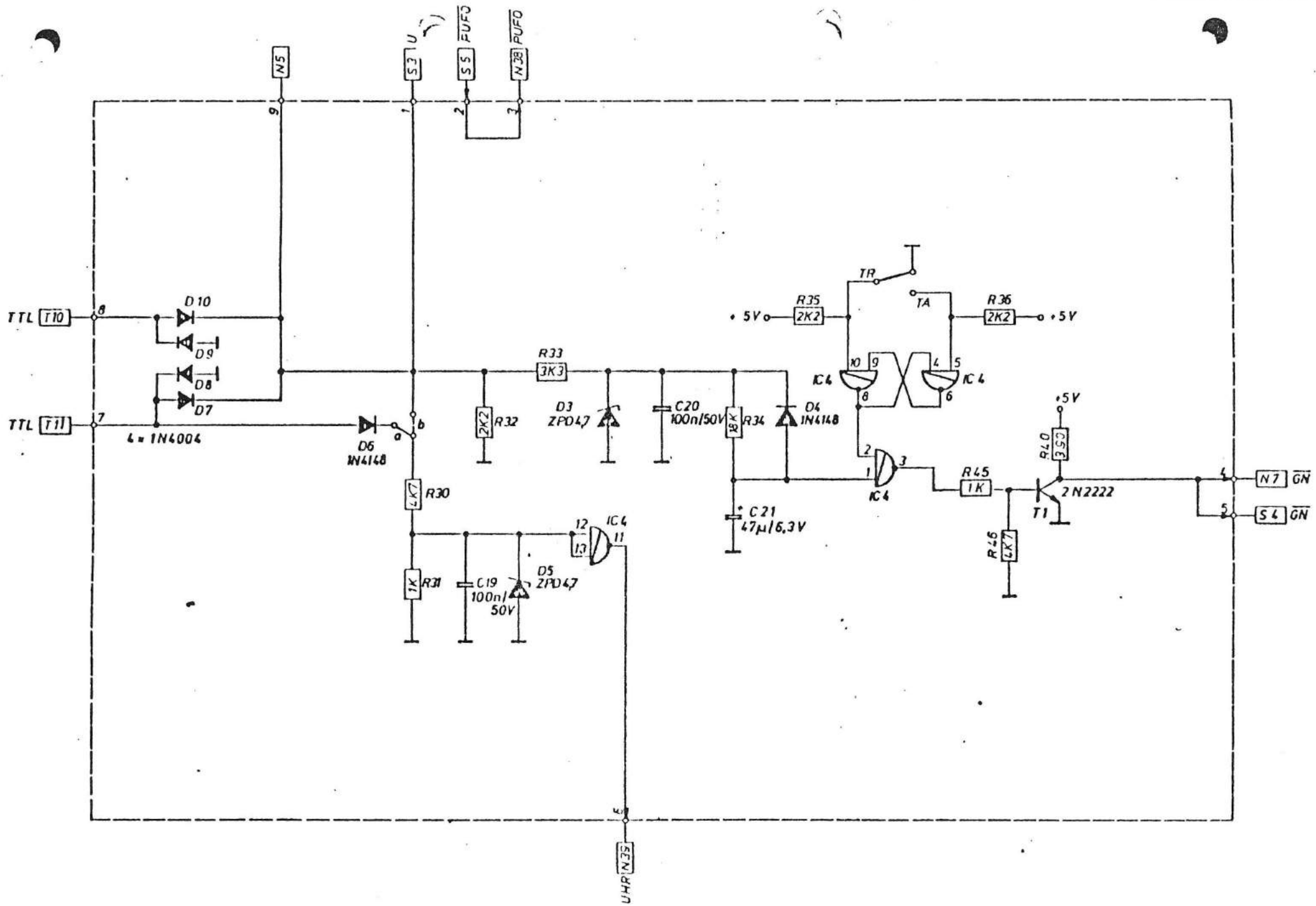
- 25° C bis + 80° C

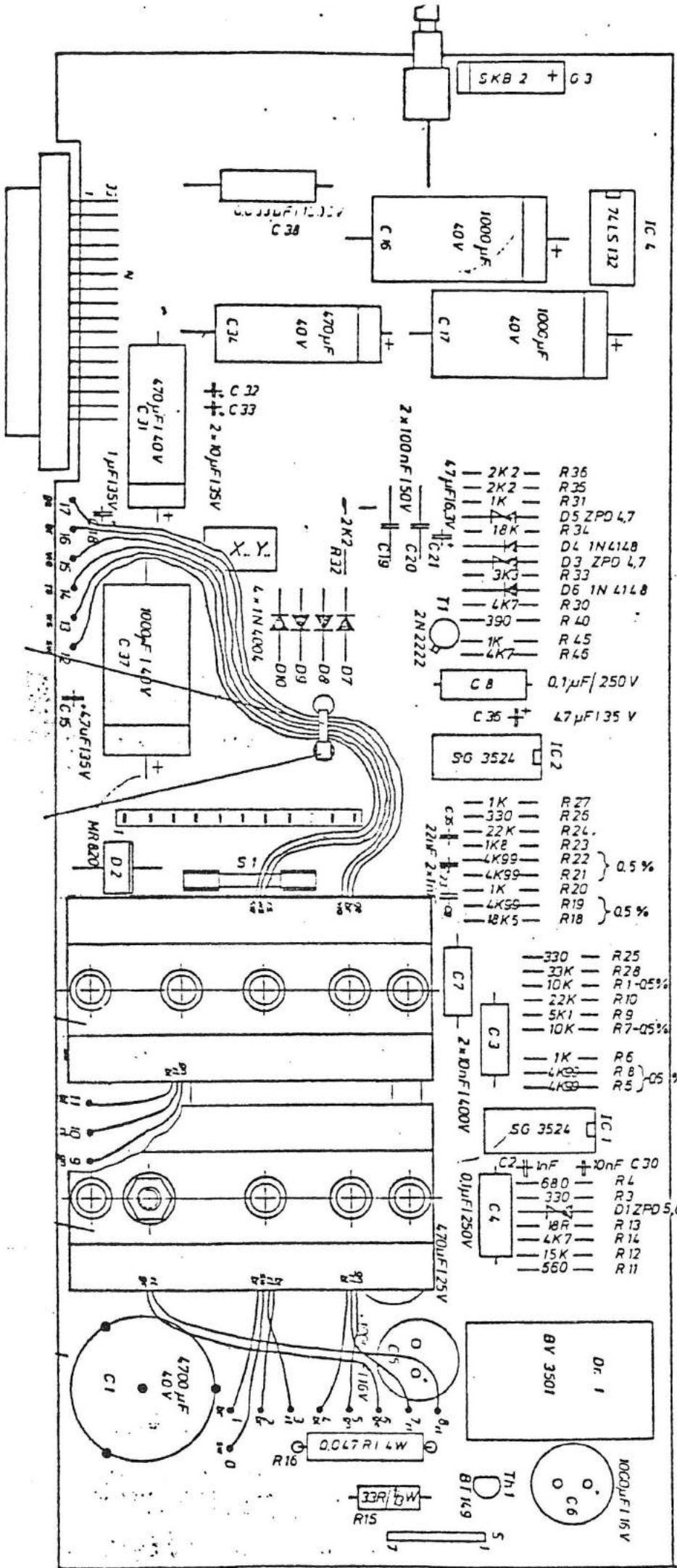
Rel. Luftfeuchte:

10 - 80 % bei 25° C









ITT 3030

AUF WACHSTUM PROGRAMMIERT !

1. Stromaufnahme:

+ 5V \pm 5 %: \leq 300 mA


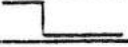
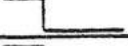
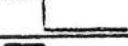


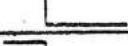
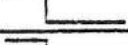
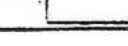
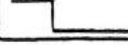
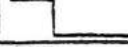

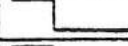
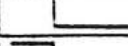
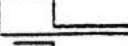
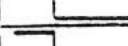

- 12V \pm 5 %: \leq 60 mA

+ 12V \pm 5 %: \leq 60 mA

2. Oszillator: 7,3728 MHz \pm 10^{-4}

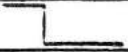


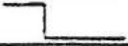

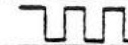

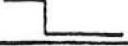
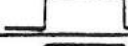
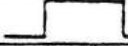
ITT 3030

AUF WACHSTUM PROGRAMMIERT !

STECKERBELEGUNGSLISTE			STECKER : A	
PIN	Bezeichnung	Diagramm	Erläuterungen	Blatt Nr.
01	+ 5V			
02	0V			
03	+ 12V			
04	0V			
05	- 12V			
06	0V			
07	M1		Machine Cycle one	
08				
09	$\overline{\text{ADR } 0}$		Adreß-Bit 0	
10	$\overline{\text{ADR } 1}$		Adreß-Bit 1	
11	$\overline{\text{ADR } 2}$		Adreß-Bit 2	
12	$\overline{\text{ADR } 3}$		Adreß-Bit 3	
13	$\overline{\text{ADR } 4}$		Adreß-Bit 4	
14	$\overline{\text{ADR } 5}$		Adreß-Bit 5	
15	$\overline{\text{ADR } 6}$		Adreß-Bit 6	
16	$\overline{\text{ADR } 7}$		Adreß-Bit 7	
17				
18				
19				
20				
21				
22				
23				
24				
25	$\overline{\text{DAT } 0}$		Daten-Bit 0	
26	$\overline{\text{DAT } 1}$		Daten-Bit 1	
27	$\overline{\text{DAT } 2}$		Daten-Bit 2	
28	$\overline{\text{DAT } 3}$		Daten-Bit 3	
29	$\overline{\text{DAT } 4}$		Daten-Bit 4	
30	$\overline{\text{DAT } 5}$		Daten-Bit 5	
31	$\overline{\text{DAT } 6}$		Daten-Bit 6	
32	$\overline{\text{DAT } 7}$		Daten-Bit 7	

ITT 3030

AUF WACHSTUM PROGRAMMIERT !

STECKERBELEGUNGSLISTE			STECKER: A	
PIN	Bezeichnung	Diagramm	Erläuterungen	Blatt Nr.
33	+ 5V			
34	0V			
35	+ 12V			
36	0V			
37	- 12V			
38	0V			
39				
40				
41				
42				
43	$\overline{\text{IOR}}$		IN/OUT Read	
44	$\overline{\text{IOW}}$		IN/OUT Write	
45	$\overline{\text{INTV}}$		Interrupt Vector	
46				
47	$\overline{\text{RST}}$		Reset intern	
48				
49				
50				
51	$\overline{\text{BUSAK}}$		Bus Acknowledge	
52				
53	\emptyset		Systemtakt	
54	IORQ		Input/Output Request	
55				
56	$\overline{\text{INT}}$		Interrupt	
57	IEI		Interrupt Enable In	
58	IEO		Interrupt Enable Out	

ITT 3030

AUF WACHSTUM PROGRAMMIERT !

STECKERBELEGUNGSLISTE			STECKER : B	
PIN	Bezeichnung	Diagramm	Erläuterungen	Blatt Nr.
01	$\overline{\text{TXD1}}$		Transmitted Data 1	
02	DTR1		Data Terminal Ready 1	
03	RTS1		Request to Send 1	
04	CTS1		Clear to Send 1	
05	$\overline{\text{RXD1}}$		Received Data 1	
06	DSR1		Data Set Ready 1	
07	OV			
08	OV			
09	M5/1		M5/1 (Modemsignal)	
10	S4/1		S4/1 (Modemsignal)	
11	RXC1		Receiver Clock 1	
12	TXC1		Transmitter Clock 1	
13	$\overline{\text{TXD2}}$		Transmitted Data 2	
14	DTR2		Data Terminal Ready 2	
15	RTS2		Request to Send 2	
16	CTS2		Clear to Send 2	
17	$\overline{\text{RXD2}}$		Received Data 2	
18	DSR2		Data Set Ready 2	
19	OV			
20	OV			

ITT 3030

AUF WACHSTUM PROGRAMMIERT !

STECKERBELEGUNGSLISTE			STECKER : B	
PIN	Bezeichnung	Diagramm	Erläuterungen	Blatt Nr.
21	$\overline{\text{TXD3}}$		Transmitted Data 3	
22	DTR3		Data Terminal Ready 3	
23	RTS3		Request to Send 3	
24	CTS3		Clear to Send 3	
25	$\overline{\text{RXD3}}$		Received Data 3	
26	DSR3		Data Set Ready 3	
27	OV			
28	OV			
29	M5/3		M5/3 (Modemsignal)	
30	S4/3		S4/3 (Modemsignal)	
31	RXC3		Receiver Clock 3	
32	TXC3		Transmitter Clock 3	
33	$\overline{\text{TXD4}}$		Transmitted Data 4	
34	DTR4		Data Terminal Ready 4	
35	RTS4		Request to Send 4	
36	CTS4		Clear to Send 4	
37	$\overline{\text{RXD4}}$		Received Data 4	
38	DSR4		Data Set Ready 4	
39	OV			
40	OV			

SIGNALBENENNUNGSLISTE		
Kurzbezeichnung	Erläuterung interner Signale	Blatt Nr.
IORQ	Input/Output Request	
M1	Machine Cycle one	
ϕ	Systemtakt	
D ϕ -7	Internes Datenbit ϕ - 7	
CS ϕ -3	Chipselect ϕ - 3	
T	Takt für Zähler 8253	
WR	Write	
RD	Read	
A ϕ /1	Internes Adreß-Bit ϕ /1	
RESI	Reset intern	
TXD	Transmitted Data	
DTR	Data Terminal Ready	
RTS	Request to Send	
CTS	Clear to Send	
RXD	Received Data	
DSR	Data Set Ready	
M5	Modemsignal	
S4	Modemsignal	
RXC	Receiver Clock	
TXC	Transmitter Clock	

ITT 3030

AUF WACHSTUM PROGRAMMIERT I

Brücken		offen	geschlossen	Erläuterung
Kanal 1	Kanal 3			
1, 3	5, 7		x	Kanal 1 und 3 interner Takt
2, 4	6, 8	x		
1, 3	5, 7	x		Kanal 1 und 3 externer Takt
2, 4	6, 8		x	

ITT 3030

AUF WACHSTUM PROGRAMMIERT!

Die Baugruppe RS 232 - Adapter steuert maximal 4 V24-Kanäle nach RS 232 Spezifikation.

Kanal 1 und 3 sind voll modemfähig.

Die 4 Kanäle werden mit 2 Z80A-SIO realisiert, die auch SDLC-fähig sind.

Zum Einstellen der Baudrate ist ein programmierbarer Zähler vom Typ 8253 vorhanden. Zwei der drei internen Zähler versorgen die SIO's mit dem Sende- und Empfangstakt, während der dritte Zähler für Softwareroutinen frei ist.

Über Brücken können die Kanäle 1 und 3 entweder mit internen oder externen Takten betrieben werden. Die Eingangsfrequenz des 8253 wird mit einem Quarzoszillator und nachgeschaltetem Teiler erzeugt.

Beispiel für Einstellen der Baudrate auf 4800 Baud:

Quarzfrequenz: 7,3728 MHz

Teiler : $\div 2 \rightarrow 3,6864 \text{ MHz}$ (Eing. Freq. 8253)

Zähler : $\div 48 \rightarrow 76,8 \text{ KHz}$ (Eing. Freq. SIO)

SIO intern : $\div 16 \rightarrow 4800 \text{ Hz}$

Mögliche Teilerverhältnisse SIO:

$\div 1, \div 16, \div 32, \div 64$

Mögliche Teilerverhältnisse 8253:

1 bis 65536 (16 Bit-Zähler)

Wird die Baugruppe nur als 2-fach Multiplexer betrieben, so sind nur Kanal 1 und 2 vorhanden.

Der 40-pol. Abgangsstecker geht über ein spezielles Kabel auf vier 25-pol. Cannonstecker.

ITT 3030

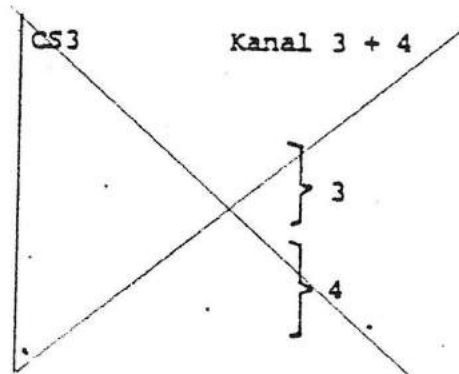
AUF WACHSTUM PROGRAMMIERT!

ADRESSBELEGUNG

ADR BIT	7	6	5	4	3	2	1	0	
	Basisadr.								
	0	0	X	X					→ CS0
	0	1	X	X					→ CS1
	1	0	X	X					→ CS2
	1	1	X	X					→ CS3

Handwritten: 1000, 0110

CS0	Kanal 1 + 2		
ADR BIT	2 ¹	2 ⁰	USART
	0	0	A Daten
	0	1	A Befehle
	1	0	B Daten
	1	1	B Befehle
CS1	TIMER		



Schreiben

ADR BIT	2 ¹	2 ⁰	
	0	0	laden Zähler 1
	0	1	laden Zähler 2
	1	0	laden Zähler 3
	1	1	Mode

Lesen

ADR BIT	2 ¹	2 ⁰	
	0	0	lesen Zähler 1
	0	1	lesen Zähler 2
	1	0	lesen Zähler 3
	1	1	Tristate

CS2	Lesen		
M5	Kanal 1	→	Datenbit 2 ³
M5	Kanal 3	→	Datenbit 2 ⁴

ITT 3030

AUF WACHSTUM PROGRAMMIERT !

Z8440 Z80[®] SIO Serial Input/Output Controller



Product Specification

March 1981

- Features**
- Two independent full-duplex channels, with separate control and status lines for modems or other devices.
 - Data rates of 0 to 500K bits/second in the x1 clock mode with a 2.5 MHz clock (Z-80 SIO), or 0 to 800K bits/second with a 4.0 MHz clock (Z-80A SIO).
 - Asynchronous protocols: everything necessary for complete messages in 5, 6, 7 or 8 bits/character. Includes variable stop bits and several clock-rate multipliers; break generation and detection; parity; overrun and framing error detection.

- Synchronous protocols: everything necessary for complete bit- or byte-oriented messages in 5, 6, 7 or 8 bits/character, including IBM Bisync, SDLC, HDLC, CCITT-X.25 and others. Automatic CRC generation/checking, sync character and zero insertion/deletion, abort generation/detection and flag insertion.
- Receiver data registers quadruply buffered, transmitter registers doubly buffered.
- Highly sophisticated and flexible daisy-chain interrupt vectoring for interrupts without external logic.

**General
Description**

The Z-80 SIO Serial Input/Output Controller is a dual-channel data communication interface with extraordinary versatility and capability. Its basic functions as a serial-to-parallel, parallel-to-serial converter/controller can be programmed by a CPU for a broad range of serial communication applications. The device supports all common asynchronous and synchronous protocols, byte-

bit-oriented, and performs all of the functions traditionally done by UARTs, USARTs and synchronous communication controllers combined, plus additional functions traditionally performed by the CPU. Moreover, it does this on two fully-independent channels, with an exceptionally sophisticated interrupt structure that allows very fast transfers. Full interlacing is provided for CPU or DMA

Z80 SIO
018 082

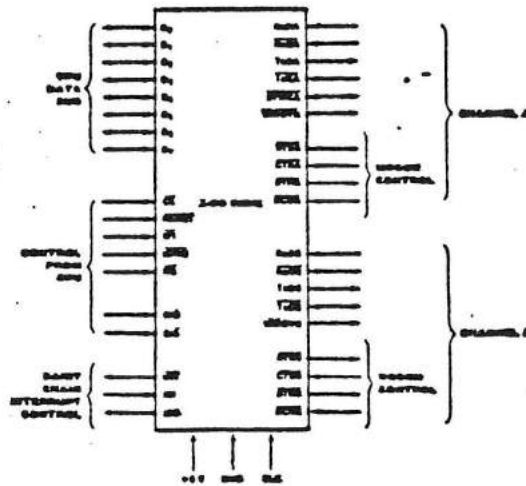


Figure 1. Z-80 SIO/2 Pin Functions

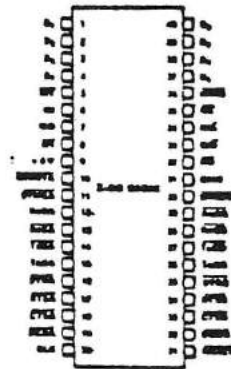


Figure 2. Z-80 SIO/2 Pin Assignments

ITT 3030

AUF WACHSTUM PROGRAMMIERT !

General Description (Continued)

control. In addition to data communication, the circuit can handle virtually all types of serial I/O with fast (or slow) peripheral devices. While designed primarily as a member of the Z-80 family, its versatility makes it well suited to many other CPUs.

The Z-80 SIO is an n-channel silicon-gate depletion-load device packaged in a 40-pin plastic or ceramic DIP. It uses a single +5 V power supply and the standard Z-80 family single-phase clock.

Pin Description

Figures 1 through 6 illustrate the three pin configurations (bonding options) available in the SIO. The constraints of a 40-pin package make it impossible to bring out the Receive Clock (RxC), Transmit Clock (TxC), Data Terminal Ready (DTR) and Sync (SYN) signals for both channels. Therefore, either Channel B lacks a signal or two signals are bonded together in the three bonding options offered:

- Z-80 SIO/2 lacks SYN_B
- Z-80 SIO/1 lacks DTR_B
- Z-80 SIO/0 has all four signals, but TxC_B and RxC_B are bonded together

The first bonding option above (SIO/2) is the preferred version for most applications. The pin descriptions are as follows:

B/A. Channel A Or B Select (input, High selects Channel B). This input defines which channel is accessed during a data transfer between the CPU and the SIO. Address bit A₀ from the CPU is often used for the selection function.

C/D. Control Or Data Select (input, High selects Control). This input defines the type of information transfer performed between the CPU and the SIO. A High at this input during a CPU write to the SIO causes the information on the data bus to be interpreted as a command for the channel selected by B/A. A Low at C/D means that the information on the data bus is data. Address bit A₁ is often used for this function.

CE. Chip Enable (input, active Low). A Low level at this input enables the SIO to accept command or data input from the CPU during a write cycle or to transmit data to the CPU during a read cycle.

CLK. System Clock (input). The SIO uses the standard Z-80 System Clock to synchronize internal signals. This is a single-phase clock.

CTS_A, CTS_B. Clear To Send (inputs, active Low). When programmed as Auto Enables, a Low on these inputs enables the respective transmitter. If not programmed as Auto Enables, these inputs may be programmed as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow-risetime signals. The SIO detects pulses on these inputs and interrupts the CPU on both logic level transitions. The Schmitt-trigger buffering does not guarantee a specified noise-level margin.

D₀-D₇. System Data Bus (bidirectional, 3-state). The system data bus transfers data and commands between the CPU and the Z-80 SIO. D₀ is the least significant bit.

DCD_A, DCD_B. Data Carrier Detect (inputs, active Low). These pins function as receiver enables if the SIO is programmed for Auto Enables; otherwise they may be used as general-purpose input pins. Both pins are Schmitt-trigger buffered to accommodate slow-risetime signals. The SIO detects pulses on these pins and interrupts the CPU on both logic level transitions. Schmitt-trigger buffer-

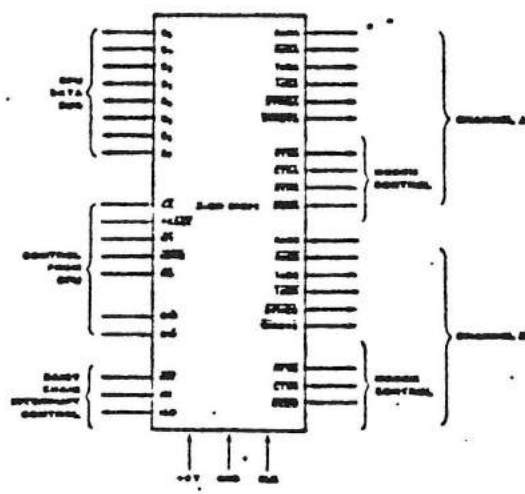


Figure 1. Z-80 SIO/1 Pin Functions

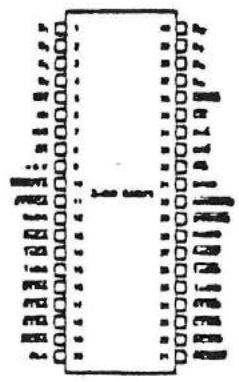


Figure 4. Z-80 SIO/1 Pin Assignments

ITT 3030

AUF WACHSTUM PROGRAMMIERT !

Pin Description
(Continued)

ing does not guarantee a specific noise-level margin.

DTR_A, DTR_B. *Data Terminal Ready* (outputs, active Low). These outputs follow the state programmed into Z-80 SIO. They can also be programmed as general-purpose outputs.

In the Z-80 SIO/1 bonding option, DTR_B is omitted.

IEL. *Interrupt Enable In* (input, active High). This signal is used with IEO to form a priority daisy chain when there is more than one interrupt-driven device. A High on this line indicates that no other device of higher priority is being serviced by a CPU interrupt service routine.

IEO. *Interrupt Enable Out* (output, active High). IEO is High only if IEL is High and the CPU is not servicing an interrupt from this SIO. Thus, this signal blocks lower priority devices from interrupting while a higher priority device is being serviced by its CPU interrupt service routine.

INT. *Interrupt Request* (output, open drain, active Low). When the SIO is requesting an interrupt, it pulls INT Low.

IORQ. *Input/Output Request* (input from CPU, active Low). IORQ is used in conjunction with B/A, C/D, CE and RD to transfer commands and data between the CPU and the SIO. When CE, RD and IORQ are all active, the channel selected by B/A transfers data to the CPU (a read operation). When CE and IORQ are active but RD is inactive, the channel selected by B/A is written to by the CPU with either data or control information as specified by C/D. If IORQ and MI are active simultane-

ously, the CPU is acknowledging an interrupt and the SIO automatically places its interrupt vector on the CPU data bus if it is the highest priority device requesting an interrupt.

MI. *Machine Cycle* (input from Z-80 CPU, active Low). When MI is active and RD is also active, the Z-80 CPU is fetching an instruction from memory; when MI is active while IORQ is active, the SIO accepts MI and IORQ as an interrupt acknowledge if the SIO is the highest priority device that has interrupted the Z-80 CPU.

RxCA, RxCB. *Receiver Clocks* (inputs). Receive data is sampled on the rising edge of RxC. The Receive Clocks may be 1, 16, 32 or 64 times the data rate in asynchronous modes. These clocks may be driven by the Z-80 CTC Counter Timer Circuit for programmable baud rate generation. Both inputs are Schmitt-trigger buffered (no noise level margin is specified).

In the Z-80 SIO/0 bonding option, RxCB is bonded together with TxCB.

RD. *Read Cycle Status* (input from CPU, active Low). If RD is active, a memory or I/O read operation is in progress. RD is used with B/A, CE and IORQ to transfer data from the SIO to the CPU.

RxDA, RxDB. *Receive Data* (inputs, active High). Serial data at TTL levels.

RESET. *Reset* (input, active Low). A Low RESET disables both receivers and transmitters, forces TxDA and TxDB marking, forces the modem controls High and disables all interrupts. The control registers must be

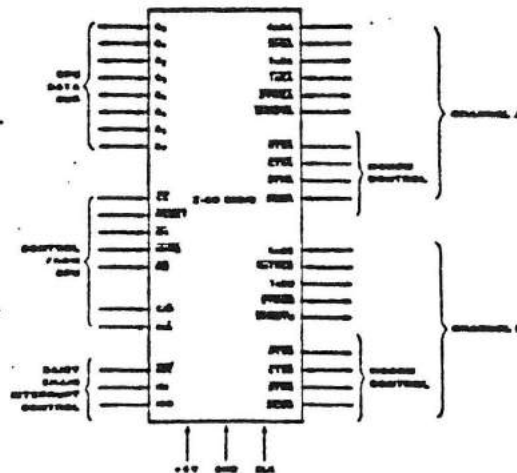


Figure 4. Z-80 SIO/0 Pin Functions

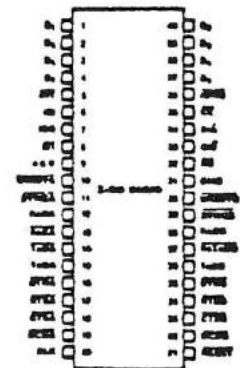


Figure 5. Z-80 SIO/0 Pin Assignments

ITT 3030

AUF WACHSTUM PROGRAMMIERT !

Pin Description
(Continued)

rewritten after the SIO is reset and before data is transmitted or received.

RTSA, RTSB. *Request To Send* (outputs, active Low). When the RTS bit in Write Register 5 (Figure 14) is set, the RTS output goes Low. When the RTS bit is reset in the Asynchronous mode, the output goes High after the transmitter is empty. In Synchronous modes, the RTS pin strictly follows the state of the RTS bit. Both pins can be used as general-purpose outputs.

SYNCA, SYNCB. *Synchronization* (inputs/outputs, active Low). These pins can act either as inputs or outputs. In the asynchronous receive mode, they are inputs similar to CTS and DCD. In this mode, the transitions on these lines affect the state of the Sync/Hunt status bits in Read Register 0 (Figure 13), but have no other function. In the External Sync mode, these lines also act as inputs. When external synchronization is achieved, SYNC must be driven Low on the second rising edge of RxC after that rising edge of RxC on which the last bit of the sync character was received. In other words, after the sync pattern is detected, the external logic must wait for two full Receive Clock cycles to activate the SYNC input. Once SYNC is forced Low, it should be kept Low until the CPU informs the external synchronization detect logic that synchronization has been lost or a new message is about to start. Character assembly begins on the rising edge of RxC that immediately precedes the falling edge of SYNC in the External Sync mode.

In the internal synchronization mode (Monosync and Bisync), these pins act as outputs that are active during the part of the receive clock (RxC) cycle in which sync characters are recognized. The sync condition is not latched, so these outputs are active each time a sync pattern is recognized, regardless of character boundaries.

In the Z-80 SIO/2 bonding option, SYNC is omitted.

TxCA, TxCB. *Transmitter Clocks* (inputs). In asynchronous modes, the Transmitter Clocks may be 1, 16, 32 or 64 times the data rate; however, the clock multiplier for the transmitter and the receiver must be the same. The Transmit Clock inputs are Schmitt-trigger buffered for relaxed rise- and fall-time requirements (no noise level margin is specified). Transmitter Clocks may be driven by the Z-80 CTC Counter Timer Circuit for programmable baud rate generation.

In the Z-80 SIO/0 bonding option, TxCB is bonded together with RxCB.

TxDA, TxDB. *Transmit Data* (outputs, active High). Serial data at TTL levels. TxD changes from the falling edge of TxC.

W/RDYA, W/RDYB. *Wait/Ready A, Wait/Ready B* (outputs, open drain when programmed for Wait function, driven High and Low when programmed for Ready function). These dual-purpose outputs may be programmed as Ready lines for a DMA controller or as Wait lines that synchronize the CPU to the SIO data rate. The reset state is open drain.

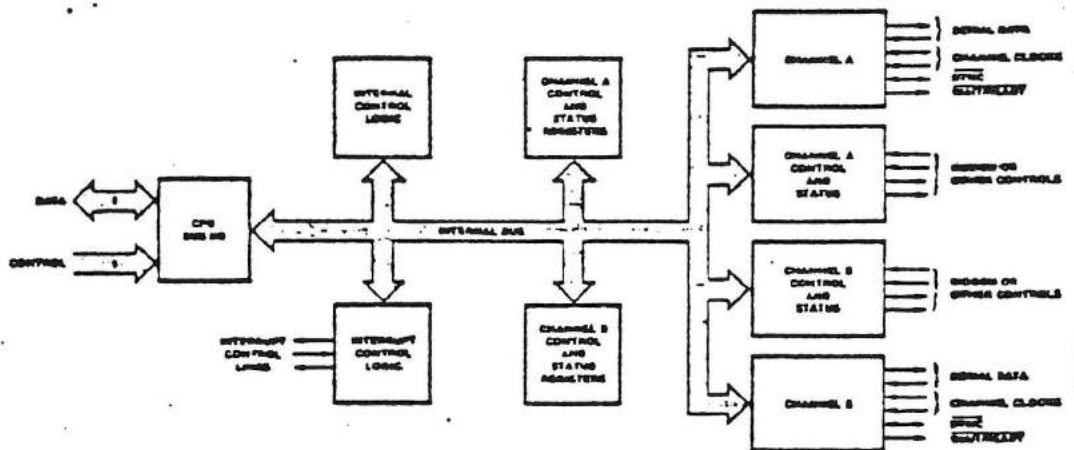


Figure 7. Block Diagram

Functional Description

The functional capabilities of the Z-80 SIO can be described from two different points of view: as a data communications device, it transmits and receives serial data in a wide variety of data-communication protocols; as a Z-80 family peripheral, it interacts with the Z-80 CPU and other peripheral circuits, sharing the data, address and control buses, as well as being a part of the Z-80 interrupt structure. As a peripheral to other microprocessors,

the SIO offers valuable features such as non-vectorized interrupts, polling and simple handshake capability.

Figure 8 illustrates the conventional devices that the SIO replaces.

The first part of the following discussion covers SIO data-communication capabilities; the second part describes interactions between the CPU and the SIO.

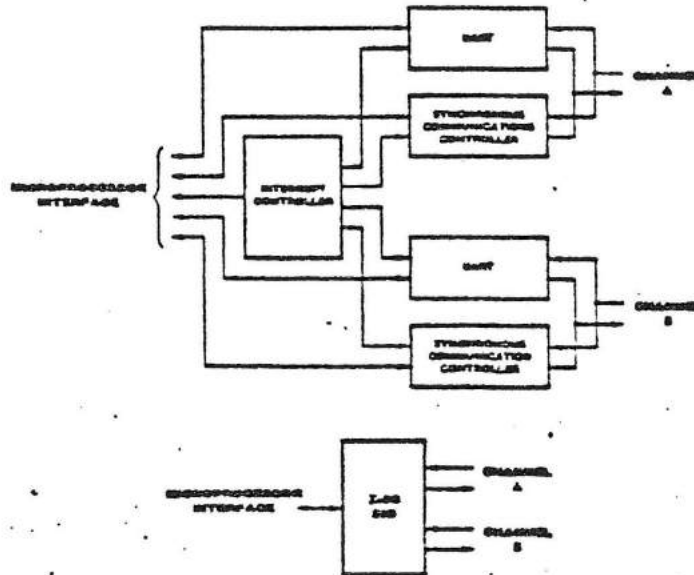


Figure 8. Conventional Devices Replaced by the Z-80 SIO

Data Communication Capabilities

The SIO provides two independent full-duplex channels that can be programmed for use in any common asynchronous or synchronous data-communication protocol. Figure 9 illustrates some of these protocols. The following is a short description of them. A more detailed explanation of these modes can be found in the *Z-80 SIO Technical Manual*.

Asynchronous Modes. Transmission and reception can be done independently on each channel with five to eight bits per character, plus optional even or odd parity. The transmitters can supply one, one-and-a-half or two stop bits per character and can provide a break output at any time. The receiver break-detection logic interrupts the CPU both at the start and end of a received break. Reception is protected from spikes by a transient spike-rejection mechanism that checks the signal one-half a bit time after a Low level is detected on the receive data input (RxDA or RxDB in Figure 5). If the Low does not persist—as in the case of a transient—the character assembly process is not started.

Framing errors and overrun errors are detected and buffered together with the partial character on which they occurred. Vectored

interrupts allow fast servicing of error conditions using dedicated routines. Furthermore, a built-in checking process avoids interpreting a framing error as a new start bit; a framing error results in the addition of one-half a bit time to the point at which the search for the next start bit is begun.

The SIO does not require symmetric transmit and receive clock signals—a feature that allows it to be used with a Z-80 CTC or many other clock sources. The transmitter and receiver can handle data at a rate of 1, 1/16, 1/32 or 1/64 of the clock rate supplied to the receive and transmit clock inputs.

In asynchronous modes, the SYNC pin may be programmed as an input that can be used for functions such as monitoring a ring indicator.

Synchronous Modes. The SIO supports both byte-oriented and bit-oriented synchronous communication.

Synchronous byte-oriented protocols can be handled in several modes that allow character synchronization with an 8-bit sync character (Monosync), any 16-bit sync pattern (Bisync), or with an external sync signal. Leading sync

ITT 3030

AUF WACHSTUM PROGRAMMIERT!

**Data
Communi-
cation
Capabilities
(Continued)**

characters can be removed without interrupting the CPU.

Five-, six- or seven-bit sync characters are detected with 8- or 16-bit patterns in the SIO by overlapping the larger pattern across multiple in-coming sync characters, as shown in Figure 10.

CRC checking for synchronous byte-oriented modes is delayed by one character time so the CPU may disable CRC checking on specific characters. This permits implementation of protocols such as IBM Bisync.

Both CRC-16 ($X^{16} + X^{15} + X^2 + 1$) and CCITT ($X^{16} + X^{12} + X^5 + 1$) error checking polynomials are supported. In all non-SDLC modes, the CRC generator is initialized to 0's; in SDLC modes, it is initialized to 1's. The SIO can be used for interfacing to peripherals such as hard-sectored floppy disk, but it cannot generate or check CRC for IBM-compatible soft-sectored disks. The SIO also provides a feature that automatically transmits CRC data when no other data is available for transmission. This allows very high-speed transmissions under DMA control with no need for CPU intervention at the end of a message. When there is no data or CRC to send in synchronous modes, the transmitter inserts 8- or 16-bit sync characters regardless of the programmed character length.

The SIO supports synchronous bit-oriented protocols such as SDLC and HDLC by performing automatic flag sending, zero insertion and CRC generation. A special command can be used to abort a frame in transmission. At the end of a message the SIO automatically transmits the CRC and trailing flag when the transmit buffer becomes empty. If a transmit

underrun occurs in the middle of a message, an external/status interrupt warns the CPU of this status change so that an abort may be issued. One to eight bits per character can be sent, which allows reception of a message with no prior information about the character structure in the information field of a frame.

The receiver automatically synchronizes on the leading flag of a frame in SDLC or HDLC, and provides a synchronization signal on the SYNC pin; an interrupt can also be programmed. The receiver can be programmed to search for frames addressed by a single byte to only a specified user-selected address or to a global broadcast address. In this mode, frames that do not match either the user-selected or broadcast address are ignored. The number of address bytes can be extended under software control. For transmitting data, an interrupt on the first received character or on every character can be selected. The receiver automatically deletes all zeroes inserted by the transmitter during character assembly. It also calculates and automatically checks the CRC to validate frame transmission. At the end of transmission, the status of a received frame is available in the status registers.

The SIO can be conveniently used under DMA control to provide high-speed reception or transmission. In reception, for example, the SIO can interrupt the CPU when the first character of a message is received. The CPU then enables the DMA to transfer the message to memory. The SIO then issues an end-of-frame interrupt and the CPU can check the status of the received message. Thus, the CPU is freed for other service while the message is being received.

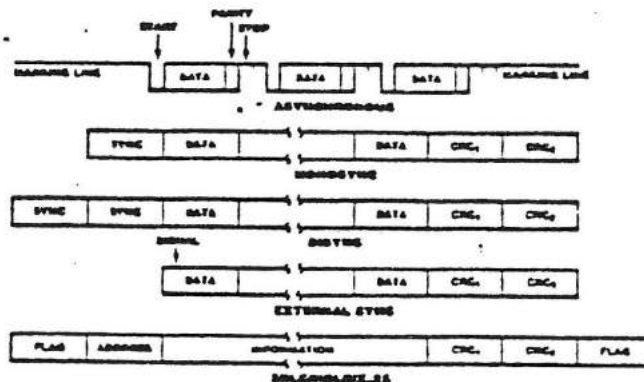


Figure 8. Some Z-80 SIO Protocols

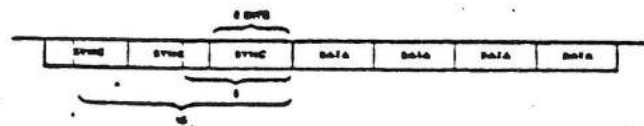


Figure 10.

ITT 3030

AUF WACHSTUM PROGRAMMIERT !

I/O Interface Capabilities

The SIO offers the choice of polling, interrupt (vectored or non-vectored) and block-transfer modes to transfer data, status and control information to and from the CPU. The block-transfer mode can also be implemented under DMA control.

Polling. Two status registers are updated at appropriate times for each function being performed (for example, CRC error-status valid at the end of a message). When the CPU is operated in a polling fashion, one of the SIO's two status registers is used to indicate whether the SIO has some data or needs some data. Depending on the contents of this register, the CPU will either write data, read data, or just go on. Two bits in the register indicate that a data transfer is needed. In addition, error and other conditions are indicated. The second status register (special receive conditions) does not have to be read in a polling sequence, until a character has been received. All interrupt modes are disabled when operating the device in a polled environment.

Interrupts. The SIO has an elaborate interrupt scheme to provide fast interrupt service in real-time applications. A control register and a status register in Channel B contain the interrupt vector. When programmed to do so, the SIO can modify three bits of the interrupt vector in the status register so that it points directly to one of eight interrupt service routines in memory, thereby servicing conditions in both channels and eliminating most of the needs for a status-analysis routine.

Transmit interrupts, receive interrupts and external/status interrupts are the main sources of interrupts. Each interrupt source is enabled under program control, with Channel A having a higher priority than Channel B, and with receive, transmit and external/status interrupts prioritized in that order within each channel. When the transmit interrupt is enabled, the

CPU is interrupted by the transmit buffer becoming empty. (This implies that the transmitter must have had a data character written into it so it can become empty.) The receiver can interrupt the CPU in one of two ways:

- Interrupt on first received character
- Interrupt on all received characters

Interrupt-on-first-received-character is typically used with the block-transfer mode. Interrupt-on-all-received-characters has the option of modifying the interrupt vector in the event of a parity error. Both of these interrupt modes will also interrupt under special receive conditions on a character or message basis (end-of-frame interrupt in SDLC, for example). This means that the special-receive condition can cause an interrupt only if the interrupt-on-first-received-character or interrupt-on-all-received-characters mode is selected. In interrupt-on-first-received-character, an interrupt can occur from special-receive conditions (except parity error) after the first-received-character interrupt (example: receive-overflow interrupt).

The main function of the external/status interrupt is to monitor the signal transitions of the Clear To Send (CTS), Data Carrier Detect (DCD) and Synchronization (SYNC) pins (Figures 1 through 6). In addition, an external/status interrupt is also caused by a CRC-sending condition or by the detection of a break sequence (asynchronous mode) or abort sequence (SDLC mode) in the data stream. The interrupt caused by the break/abort sequence allows the SIO to interrupt when the break/abort sequence is detected or terminated. This feature facilitates the proper termination of the current message, correct initialization of the next message, and the accurate timing of the break/abort condition in external logic.

018 02Z

ITT 3030

AUF WACHSTUM PROGRAMMIERT!

I/O Interface Capabilities (Continued)

In a Z-80 CPU environment (Figure 11), SIO interrupt vectoring is "automatic": the SIO passes its internally-modifiable 8-bit interrupt vector to the CPU, which adds an additional 8 bits from its interrupt-vector (I) register to form the memory address of the interrupt-routine table. This table contains the address of the beginning of the interrupt routine itself. The process entails an indirect transfer of CPU control to the interrupt routine, so that the next instruction executed after an interrupt acknowledge by the CPU is the first instruction of the interrupt routine itself.

CPU/DMA Block Transfer. The SIO's block-transfer mode accommodates both CPU block transfers and DMA controllers (Z-80 DMA or other designs). The block-transfer mode uses the Wait/Ready output signal, which is selected with three bits in an internal control register. The Wait/Ready output signal can be programmed as a WAIT line in the CPU block-transfer mode or as a READY line in the DMA block-transfer mode.

To a DMA controller, the SIO READY output indicates that the SIO is ready to transfer data to or from memory. To the CPU, the WAIT output indicates that the SIO is not ready to transfer data, thereby requesting the CPU to extend the I/O cycle.

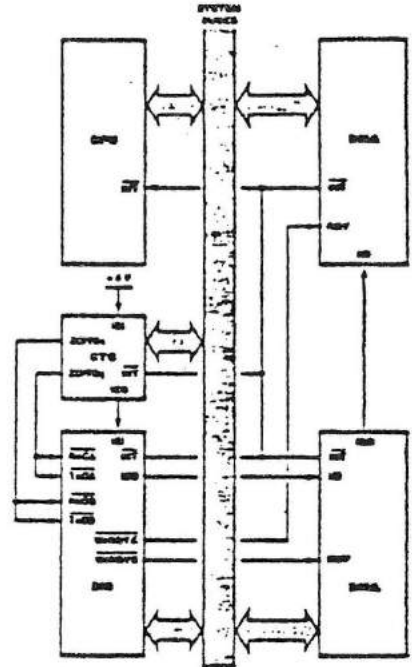


Figure 11. Typical Z-80 Environment

Internal Structure

The internal structure of the device includes a Z-80 CPU interface, internal control and interrupt logic, and two full-duplex channels. Each channel contains its own set of control and status (write and read) registers, and control and status logic that provides the interface to modems or other external devices.

The registers for each channel are designated as follows:

- WR0-WR7 — Write Registers 0 through 7
- RR0-RR2 — Read Registers 0 through 2

The register group includes five 8-bit control registers, two sync-character registers and two status registers. The interrupt vector is written into an additional 8-bit register (Write Register 2) in Channel B that may be read through another 8-bit register (Read Register 2) in Channel B. The bit assignment and functional grouping of each register is configured to simplify and organize the programming process. Table 1 lists the functions assigned to each read or write register.

Read Register Functions

RR0	Transmit/Receive buffer status, interrupt status and external status
RR1	Special Receive Condition status
RR2	Modified interrupt vector (Channel B only)

Write Register Functions

WR0	Register pointers, CRC initialize, initialization commands for the various modes, etc.
WR1	Transmit/Receive interrupt and data transfer mode definition.
WR2	Interrupt vector (Channel B only)
WR3	Receive parameters and control
WR4	Transmit/Receive miscellaneous parameters and modes
WR5	Transmit parameters and controls
WR6	Sync character or SDLC address field
WR7	Sync character or SDLC flag

Internal Structure
(Continued)

The logic for both channels provides formats, synchronization and validation for data transferred to and from the channel interface. The modem control inputs, Clear To Send (CTS) and Data Carrier Detect (DCD), are monitored by the external control and status logic under program control. All external control-and-status-logic signals are general-purpose in nature and can be used for functions other than modem control.

Data Path. The transmit and receive data path illustrated for Channel A in Figure 12 is identical for both channels. The receiver has three 8-bit buffer registers in a FIFO arrangement, in addition to the 8-bit receive shift register. This scheme creates additional time for the

CPU to service an interrupt at the beginning of a block of high-speed data. Incoming data is routed through one of several paths (data or CRC) depending on the selected mode and—in asynchronous modes—the character length.

The transmitter has an 8-bit transmit data buffer register that is loaded from the internal data bus, and a 20-bit transmit shift register that can be loaded from the sync-character buffers or from the transmit data register. Depending on the operational mode, outgoing data is routed through one of four main paths before it is transmitted from the Transmit Data output (TxD).

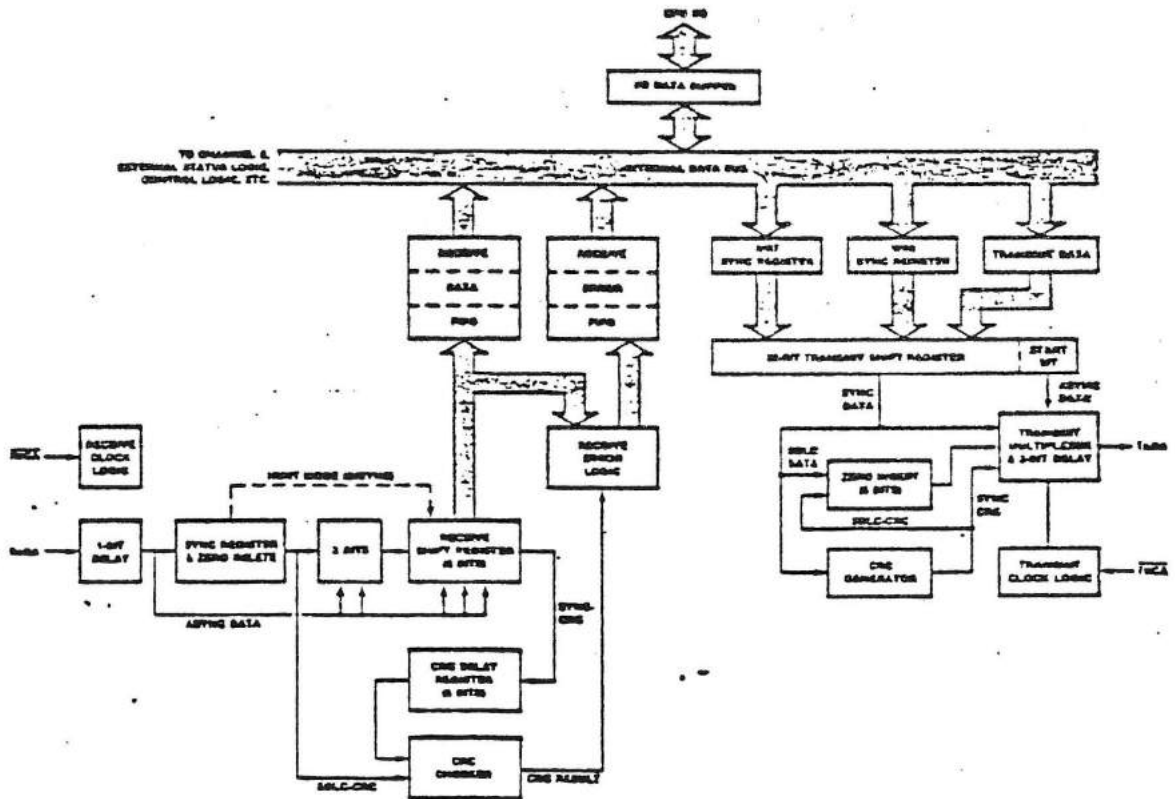


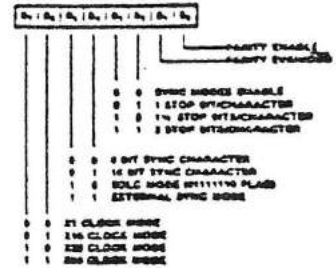
Figure 12. Transmit and Receive Data Path (Channel A)

Programming
(Continued)

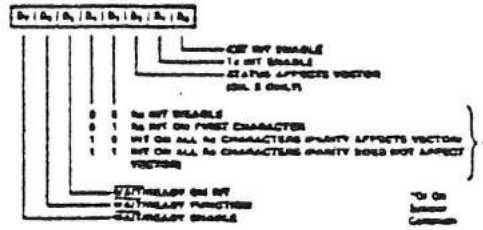
WRITE REGISTER 0



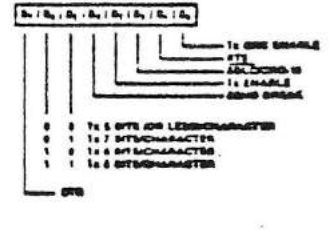
WRITE REGISTER 4



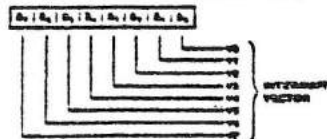
WRITE REGISTER 1



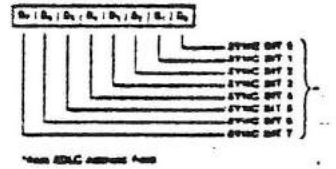
WRITE REGISTER 5



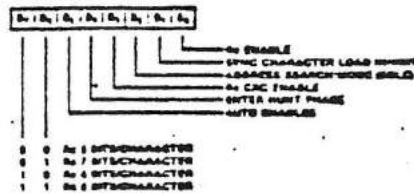
WRITE REGISTER 2 (CHANNEL 3 ONLY)



WRITE REGISTER 6



WRITE REGISTER 3



WRITE REGISTER 7

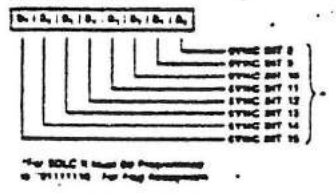


Figure 14. Write Register Bit Functions

Programming The system program first issues a series of commands that initialize the basic mode of operation and then other commands that qualify conditions within the selected mode. For example, the asynchronous mode, character length, clock rate, number of stop bits, even or odd parity might be set first; then the interrupt mode; and finally, receiver or transmitter enable.

Both channels contain registers that must be programmed via the system program prior to operation. The channel-select input (B/\bar{A}) and the control/data input (C/\bar{D}) are the command-structure addressing controls, and are normally controlled by the CPU address bus. Figures 15 and 16 illustrate the timing relationships for programming the write registers and transferring data and status.

Read Registers. The SIO contains three read registers for Channel B and two read registers for Channel A (RR0-RR2 in Figure 13) that can be read to obtain the status information: RR2 contains the internally-modifiable interrupt vector and is only in the Channel B register set. The status information includes error conditions, interrupt vector and standard communications-interface signals.

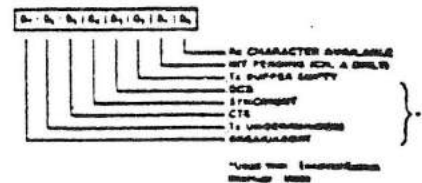
To read the contents of a selected read register other than RR0, the system program must first write the pointer byte to WR0 in exactly the same way as a write register operation. Then, by executing a read instruction, the contents of the addressed read register can be read by the CPU.

The status bits of RR0 and RR1 are carefully grouped to simplify status monitoring. For example, when the interrupt vector indicates that a Special Receive Condition interrupt has occurred, all the appropriate error bits can be read from a single register (RR1).

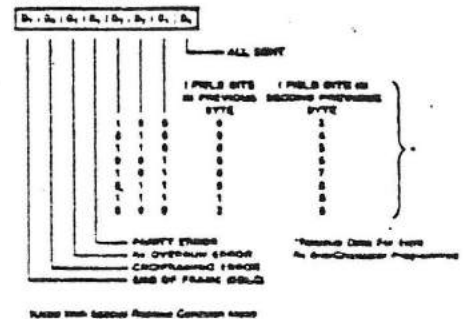
Write Registers. The SIO contains eight write registers for Channel B and seven write registers for Channel A (WR0-WR7 in Figure 14) that are programmed separately to configure the functional personality of the channels; WR2 contains the interrupt vector for both channels and is only in the Channel B register set. With the exception of WR0, programming the write registers requires two bytes. The first byte is to WR0 and contains three bits (D_0 - D_2) that point to the selected register; the second byte is the actual control word that is written into the register to configure the SIO.

WR0 is a special case in that all of the basic commands can be written to it with a single byte. Reset (internal or external) initializes the pointer bits D_0 - D_2 to point to WR0. This implies that a channel reset must not be combined with the pointing to any register.

READ REGISTER 0



READ REGISTER 1*



READ REGISTER 2

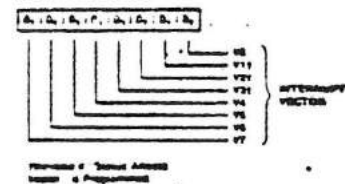


Figure 13. Read Register Bit Functions

Timing

The SIO must have the same clock as the CPU (same phase and frequency relationship, not necessarily the same driver).

Read Cycle. The timing signals generated by a Z-80 CPU input instruction to read a data or status byte from the SIO are illustrated in Figure 15.

Write Cycle. Figure 16 illustrates the timing and data signals generated by a Z-80 CPU output instruction to write a data or control byte into the SIO.

Interrupt-Acknowledge Cycle. After receiving an interrupt-request signal from an SIO (INT pulled Low), the Z-80 CPU sends an interrupt-acknowledge sequence (MI Low, and IORQ Low a few cycles later) as in Figure 17.

The SIO contains an internal daisy-chained interrupt structure for prioritizing nested interrupts for the various functions of its two channels, and this structure can be used within an external user-defined daisy chain that prioritizes several peripheral circuits.

The IEI of the highest-priority device is terminated High. A device that has an interrupt pending or under service forces its IEO Low. For devices with no interrupt pending or under service, IEO = IEI.

To insure stable conditions in the daisy chain, all interrupt status signals are prevented from changing while MI is Low. When IORQ is Low, the highest priority interrupt requestor (the one with IEI High) places its interrupt vector on the data bus and sets its

internal interrupt-under-service latch.

Return From Interrupt Cycle. Figure 18 illustrates the return from interrupt cycle. Normally, the Z-80 CPU issues a RETI (Return From Interrupt) instruction at the end of an interrupt service routine. RETI is a 2-byte opcode (ED-4D) that resets the interrupt-under-service latch in the SIO to terminate the interrupt that has just been processed. This is accomplished by manipulating the daisy chain in the following way.

The normal daisy-chain operation can be used to detect a pending interrupt; however, it cannot distinguish between an interrupt under service and a pending unacknowledged interrupt of a higher priority. Whenever "ED" is decoded, the daisy chain is modified by forcing High the IEO of any interrupt that has not yet been acknowledged. Thus the daisy chain identifies the device presently under service as the only one with an IEI High and an IEO Low. If the next opcode byte is "4D," the interrupt-under-service latch is reset.

The ripple time of the interrupt daisy chain (both the High-to-Low and the Low-to-High transitions) limits the number of devices that can be placed in the daisy chain. Ripple time can be improved with carry-look-ahead, or by extending the interrupt-acknowledge cycle. For further information about techniques for increasing the number of daisy-chained devices, refer to the Z-80 CPU Product Specification.

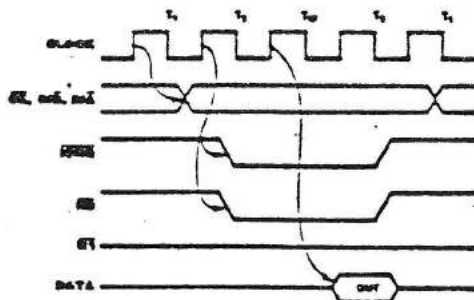


Figure 15. Read Cycle

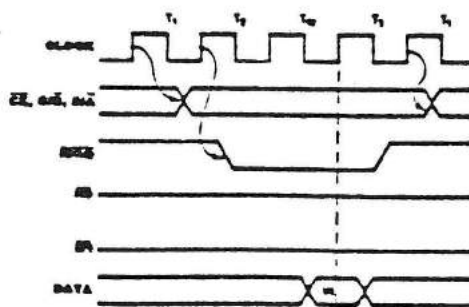


Figure 16. Write Cycle

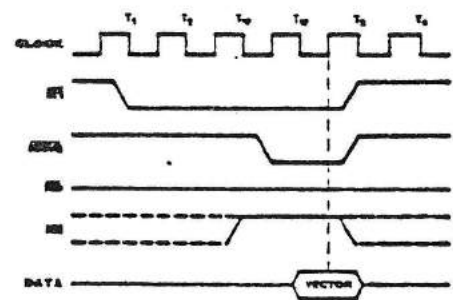


Figure 17. Interrupt Acknowledge Cycle

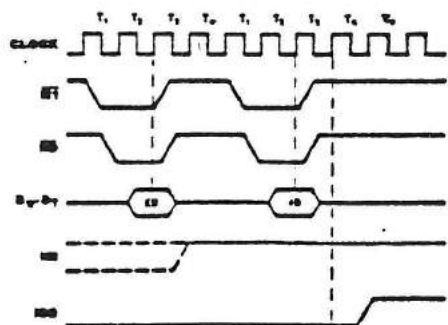


Figure 18. Return from Interrupt Cycle

ITT 3030

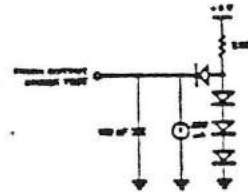
AUF WACHSTUM PROGRAMMIERT!

Absolute Maximum Ratings	Voltages on all inputs and outputs with respect to GND.....	-0.3 V to +7.0 V	Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.
	Operating Ambient Temperature	As Specified in Ordering Information	
	Storage Temperature.....	-65°C to +150°C	

Test Conditions The characteristics below apply for the following test conditions, unless otherwise noted. All voltages are referenced to GND (0 V). Positive current flows into the referenced pin. Available operating temperature ranges are:

The product number for each operating temperature range may be found in the ordering information section.

- 0°C to +70°C,
+4.75 V ≤ V_{CC} ≤ +5.25 V
- -40°C to +85°C,
+4.75 V ≤ V_{CC} ≤ +5.25 V
- -55°C to +125°C,
+4.5 V ≤ V_{CC} ≤ +5.5 V



DC Characteristics	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} -0.6	+5.5	V	
	V _{IL}	Input Low Voltage	-0.3	+0.8	V	
	V _{IH}	Input High Voltage	+2.0	+5.5	V	
	V _{OL}	Output Low Voltage		+0.4	V	I _{OL} = 2.0 mA
	V _{OH}	Output High Voltage	+2.4		V	I _{OH} = -250 μA
	I _{I1}	Input Leakage Current	-10	+10	μA	0 < V _{IH} < V _{CC}
	I _{I2}	3-State Output/Data Bus Input Leakage Current	-10	+10	μA	0 < V _{IH} < V _{CC}
	I _{I(LST)}	SYNC Pin Leakage Current	-40	+10	μA	0 < V _{IH} < V _{CC}
	I _{CC}	Power Supply Current		100	mA	

Over specified temperature and voltage range.

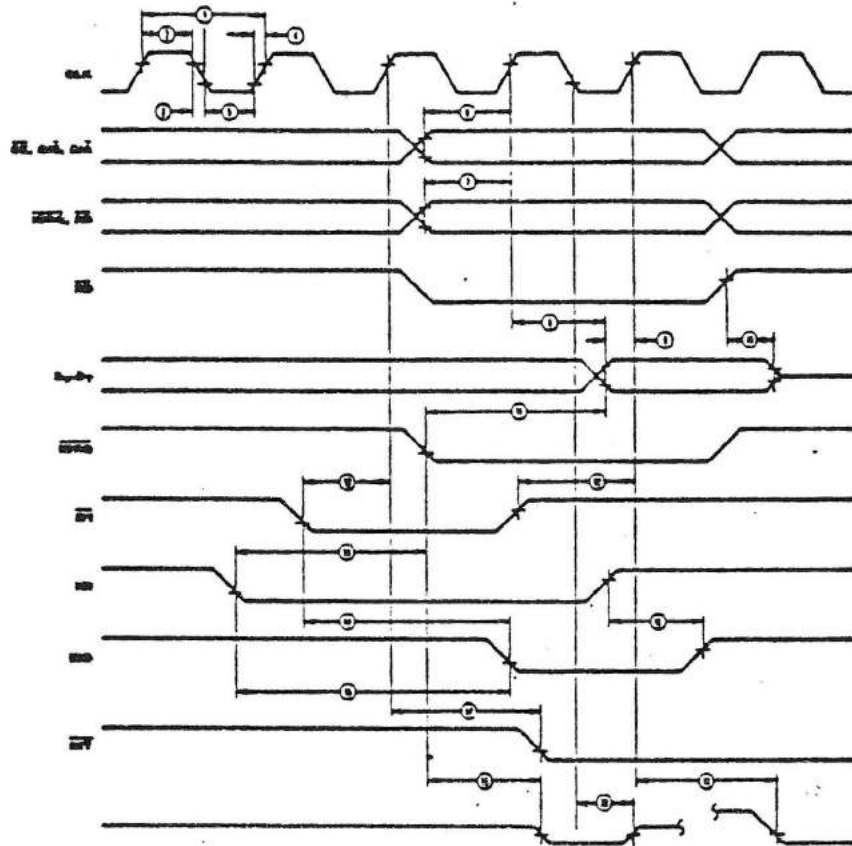
Capacitance	Symbol	Parameter	Min	Max	Unit	Test Condition
	C	Clock Capacitance		40	pF	Unmeasured
	C _{IN}	Input Capacitance		5	pF	pins returned
	C _{OUT}	Output Capacitance		10	pF	to ground

Over specified-temperature range: f = 1MHz

ITT 3030

AUF WACHSTUM PROGRAMMIERT!

AC
Electrical
Character-
istics

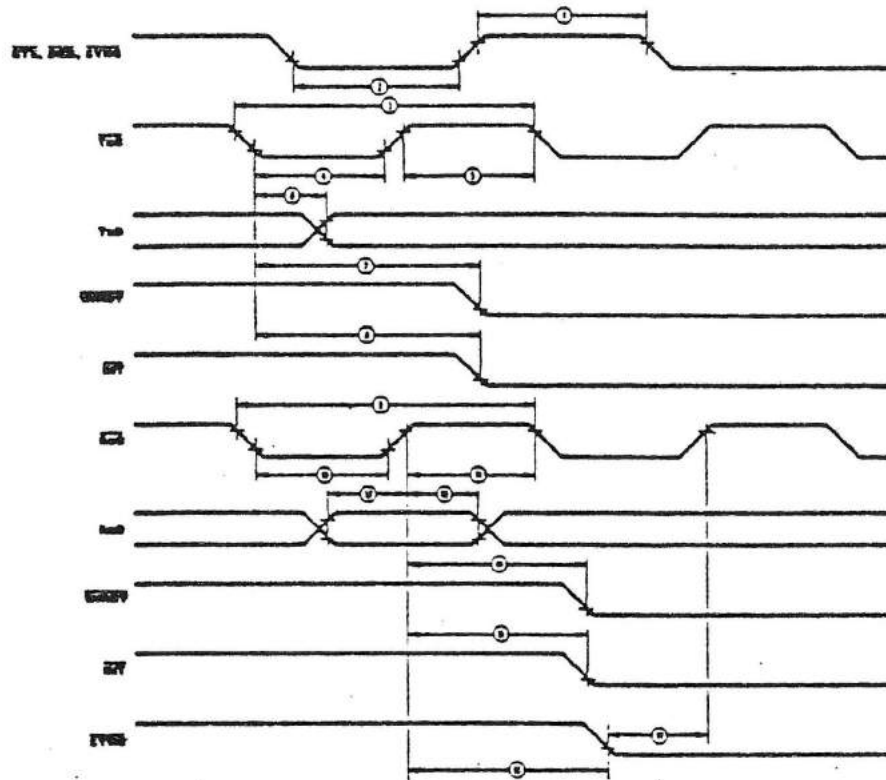


Number	Symbol	Parameter	Z-80 SIO		Z-80A SIO		Z-80B SIO		Unit
			Min	Max	Min	Max	Min	Max	
1	T _c	Clock Cycle Time	400	4000	250	4000	165	4000	ns
2	T _{wCh}	Clock Width (High)	170	2000	105	2000	70	2000	ns
3	T _{fC}	Clock Fall Time		30		30		15	ns
4	T _{rC}	Clock Rise Time		30		30		15	ns
5	T _{wCl}	Clock Width (Low)	170	2000	105	2000	70	2000	ns
6	T _{sAD(C)}	\overline{CE} , C/D, B/ \overline{A} to Clock Setup Time	160		145		80		ns
7	T _{sCS(C)}	\overline{IORQ} , RD to Clock Setup Time	240		115		80		ns
8	T _{dC(DO)}	Clock to Data Out Delay		240		220		150	ns
9	T _{sDI(C)}	Data In to Clock Setup (Write or \overline{M} Cycle)	50		50		30		ns
10	T _{dRD(DOz)}	\overline{RD} to Data Out Float Delay		230		110		90	ns
11	T _{dIO(DOI)}	\overline{IORQ} to Data Out Delay (INTACK Cycle)		340		160		100	ns
12	T _{sMI(C)}	\overline{M} to Clock Setup Time	210		90		75		ns
13	T _{sEI(IO)}	\overline{IEI} to \overline{IORQ} Setup Time (INTACK Cycle)	200		140		120		ns
14	T _{dMI(IEO)}	\overline{M} to \overline{IEO} Delay (interrupt before \overline{M})		300		190		160	ns
15	T _{dEI(IEOz)}	\overline{IEI} to \overline{IEO} Delay (after ED decode)		150		100		70	ns
16	T _{dEI(IEO)}	\overline{IEI} to \overline{IEO} Delay		150		100		70	ns
17	T _{dC(INT)}	Clock to INT Delay		200		200		150	ns
18	T _{dIO(W/RW)}	\overline{IORQ} or \overline{CE} to $\overline{W/RDY}$ Delay (Wait Mode)		300		210		175	ns
19	T _{dC(W/R)}	Clock to $\overline{W/RDY}$ Delay (Ready Mode)		120		120		100	ns
20	T _{dC(W/RWz)}	Clock to $\overline{W/RDY}$ Float Delay (Wait Mode)		150		130		110	ns
21	T _h	Any unspecified Hold when Setup is specified	0		0		0		ns

ITT 3030

AUF WACHSTUM PROGRAMMIERT !

AC
Electrical
Character-
istics
(Continued)

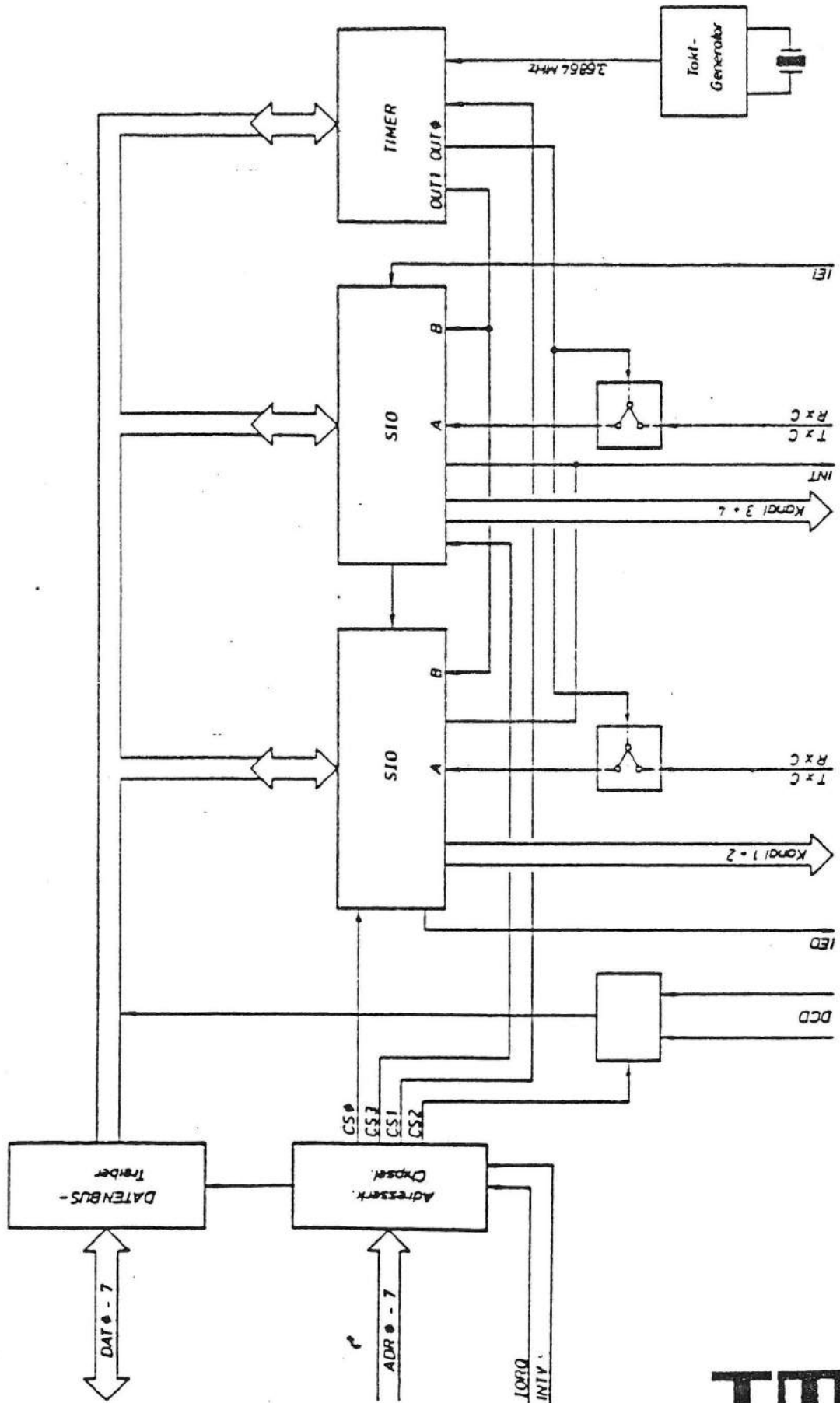


Number	Symbol	Parameter	Z-80 SIO		Z-80A SIO		Z-80B SIO		Unit
			Min	Max	Min	Max	Min	Max	
1	TwPh	Pulse Width (High)	200		200		200		ns
2	TwPl	Pulse Width (Low)	200		200		200		ns
3	TcTxC	TxC Cycle Time	400	=	400	=	330	=	ns
4	TwTxCl	TxC Width (Low)	180	=	180	=	100	=	ns
5	TwTxCh	TxC Width (High)	180	=	180	=	100	=	ns
6	TdTxC(TxD)	TxC to TxD Delay (xl Mode)		400		300		220	ns
7	TdTxC(W/RD)	TxC to W/RDY Delay (Ready Mode)	5	9	5	9	5	9	Clk Periods*
8	TdTxC(INT)	TxC to INT Delay	5	9	5	9	5	9	Clk Periods*
9	TcRxC	RxC Cycle Time	400	=	400	=	330	=	ns
10	TwRxCl	RxC Width (Low)	180	=	180	=	100	=	ns
11	TwRxCh	RxC Width (High)	180	=	180	=	100	=	ns
12	TsRxD(RxC)	RxD to RxC Setup Time (xl Mode)	0		0		0		ns
13	ThRxD(RxC)	RxC to RxD Hold Time (xl Mode)	140		140		100		ns
14	TdRxC(W/RD)	RxC to W/RDY Delay (Ready Mode)	10	13	10	13	10	13	Clk Periods*
15	TdRxC(INT)	RxC to INT Delay	10	13	10	13	10	13	Clk Periods*
16	TdRxC(SYNC)	RxC to SYNC Delay (Output Modes)	4	7	4	7	4	7	Clk Periods*
17	TsSYNC(RxC)	SYNC to RxC Setup (External Sync Modes)	-100		-100		100		ns

*All values, the System Clock rate must be at least five times the maximum data rate.
*INT must arrive a minimum of one complete Clock Cycle.
*Sync Clock

ITT 3030

AUF WACHSTUM PROGRAMMIERT !



Die Baugruppe V24 Multiplexer ~~steuert~~ steuert maximum 4 V24-Kanäle nach RS 232 Spezifikation.

Kanal 1 und 3 sind voll modemfähig.

Die 4 Kanäle werden mit 2 Z80-SIO realisiert, die auch SDLC-fähig sind.

Zum Einstellen der Baudrate ist ein programmierbarer Zähler vom Typ 8253 vorhanden. Zwei der drei ~~den~~ internen Zähler versorgen die SIO's mit dem Send- und Empfangstakt, während der dritte Zähler für Software-routinen frei ist.

Über Brücken können die Kanäle 1 und 3 entweder mit internem oder externen Takten betrieben werden.

Die ~~Send- und Empfangstakte~~ ^{Eingangsfrequenz des 8253} ~~werden~~ ^{wird} mit einem Quarzoszillator und nachgeschaltetem Teiler erzeugt.

Beispiel für Einstellen der Baudrate auf 4800 Baud:

Quarzfreq.: 7,3728 MHz

Teiler : $\div 2 \rightarrow 3,6864$ (Eing.Freq. 8253)

Zähler : $\div 48 \rightarrow 76,8$ KHz (Eing.Freq. SIO)

SIO intern : $\div 16 \rightarrow 4800$ Hz

Mögliche Teilverhältnisse SIO:

$\div 1, \div 16, \div 32, \div 64$

Mögliche Teilverhältnisse 8253

1 bis 65536 (16 Bit-Zähler)

Die Z80A-SIO's sind ~~weiter~~ auch SDLC-fähig

Wird die Baugruppe nur als 2-fach Multiplexer
betrieben, so sind nur Kanal 1 und 2 vorhanden.
Über ^{Brücken} Jumper kann der Interruptausgang des S10 zur
CPU geschaltet werden.

40-pol. Der Abgangsstecker geht über ein spezielles Kabel auf
ein vier 25-pol. Cammerstecker. (2)

ADR BIT	7	6	5	4	3	2	1	0	
	Basisadr.				0	0	x	x	→ CS4
					0	1	x	x	→ 1
					1	0	x	x	→ 2
					1	1	x	x	→ 3
	8								
	0	1	1	0					
	1	0	0	0					
	6								

CS4 Kanal 1+2

CS3 Kanal 3+4

ADR BIT	2 ¹	2 ⁰	USART		
	0	0	A	Daten	} 1
	0	1	A	Befehle	
	1	0	B	Daten	} 2
	1	1	B	Befehle	
					} 3
					} 4

CS1 TIMER

Schreiben

ADR BIT	2 ¹	2 ⁰		
	0	0	Zadern Zähler	1
	0	1	"	2
	1	0	"	3
	1	1	Mode	

Lesen

ADR BIT	2 ¹	2 ⁰		
	0	0	Lesen Zähler	1
	0	1	"	2
	1	0	"	3
	1	1	Tri state	

CS 2 Lesen

MS Kanal 1 → Datenbit 2³
 MS Kanal 3 → — 4 — 2⁴

3

BRÜCKENBELEGUNGSLISTE

1.) Externe, bzw. interne Fahrt

Brücken		offen	geschlossen	Erklärung
Kanal 1	Kanal 3			
1, 3	5, 7		X	Kanal 1 und 3 interne Fahrt
2, 4	6, 8	X		
1, 3	5, 7	X		Kanal 1 und 3 externe Fahrt
2, 4	6, 8		X	

2.) Inkrupt

Durch Schließen der Brücken 9 bis 14 wird der entsprechende Inkrupt erreicht

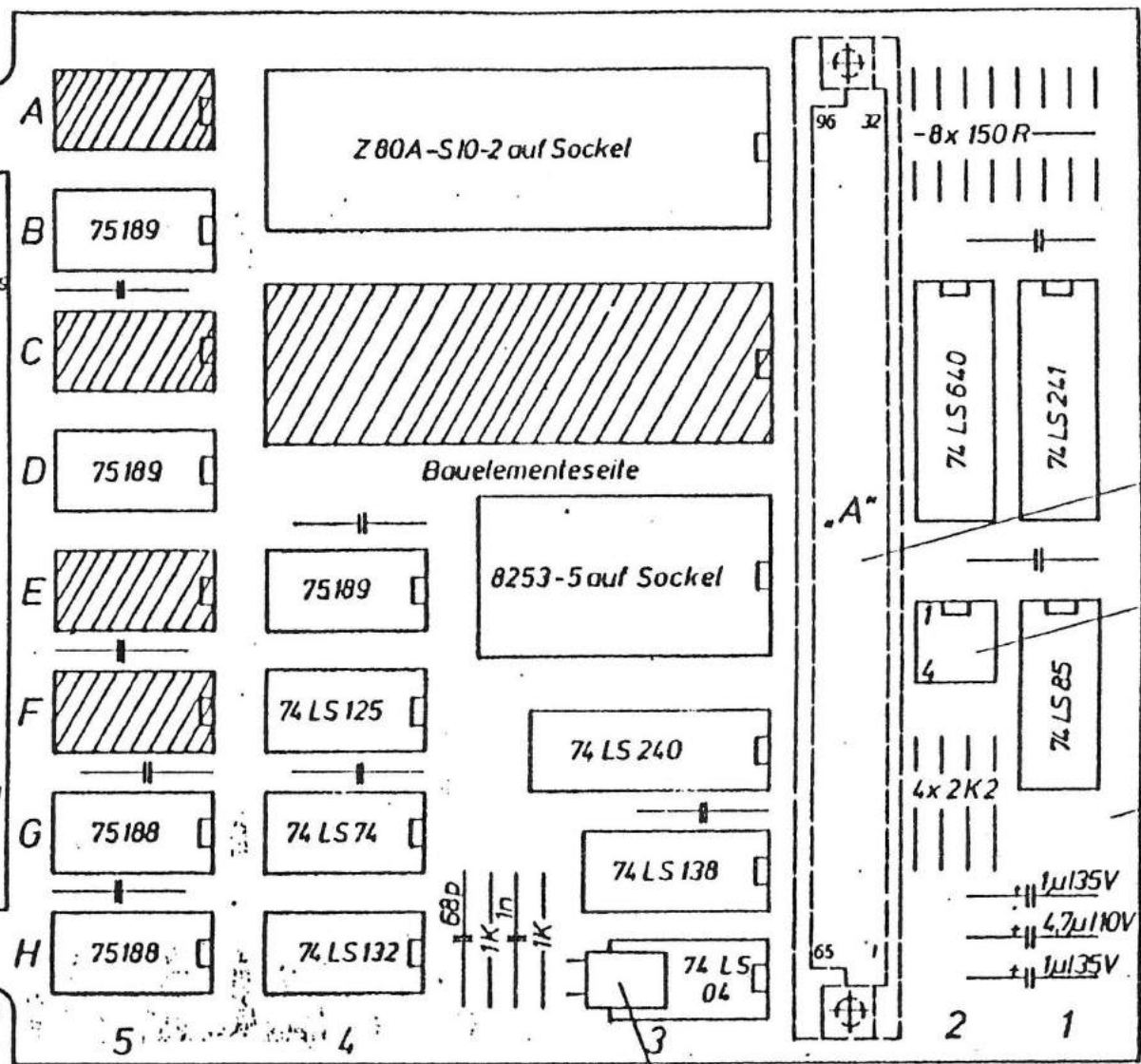
Kanal 1 u. 2	Brücke 9	geschlossen	entspricht Inkrupt	INT 2
	" 10	"	"	INT 1
Kanal 3 u. 4	" 11	"	"	INT \emptyset
	" 12	"	"	INT 2
	" 13	"	"	INT 1
	" 14	"	"	INT \emptyset

			Esca		Unterlagen Nr:	von
			Doku.	2	2 NIC 583 E	1
Nr. Änderung	TTMMJJ	Name			Referenz-Baugruppe:	Seite
					2 NIC 519B/520B	1

RS 232
2-fach

TELE 3030

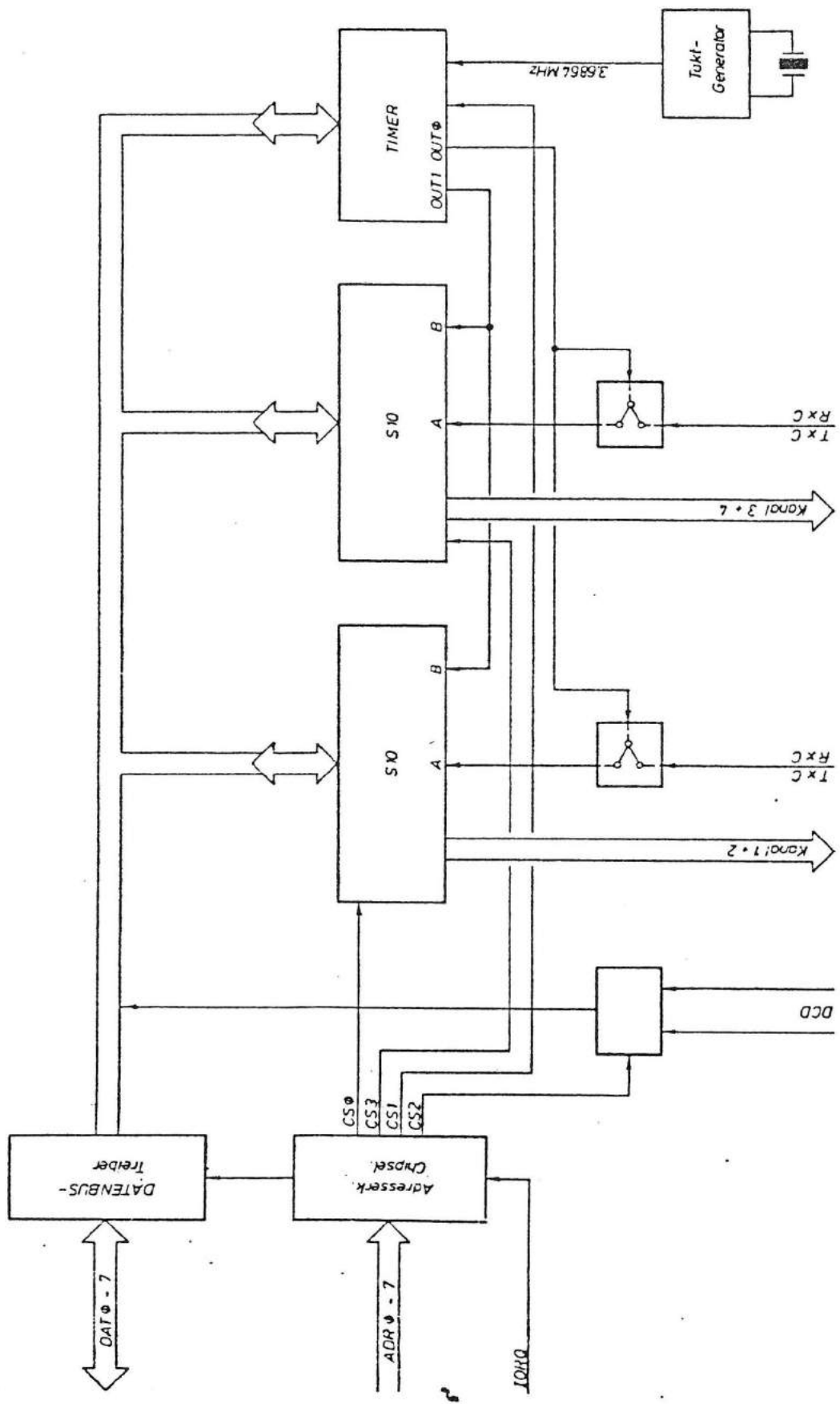
AUF WACHSTUM PROGRAMMIERT !



7,3728 MHz

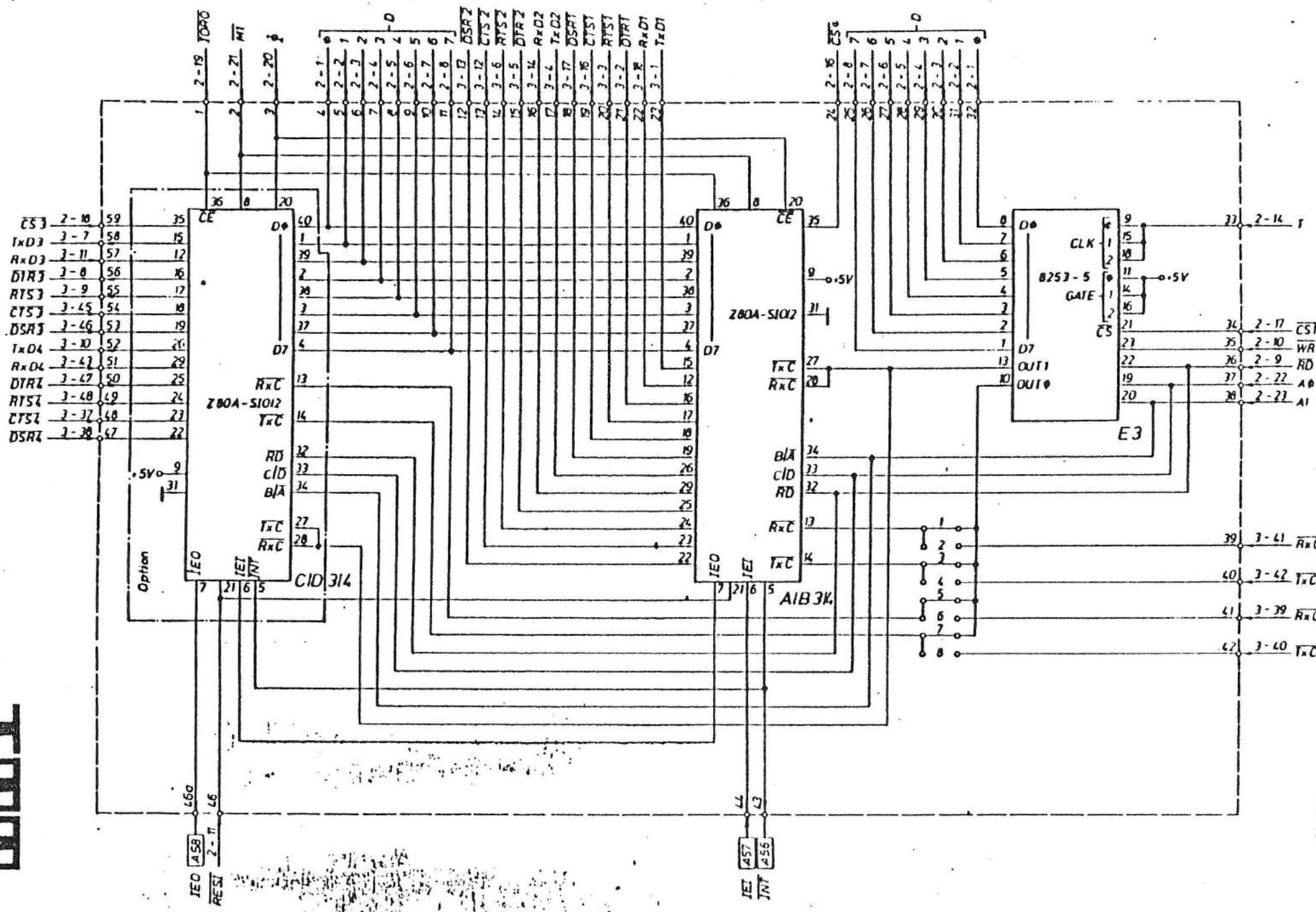
Technik der Welt

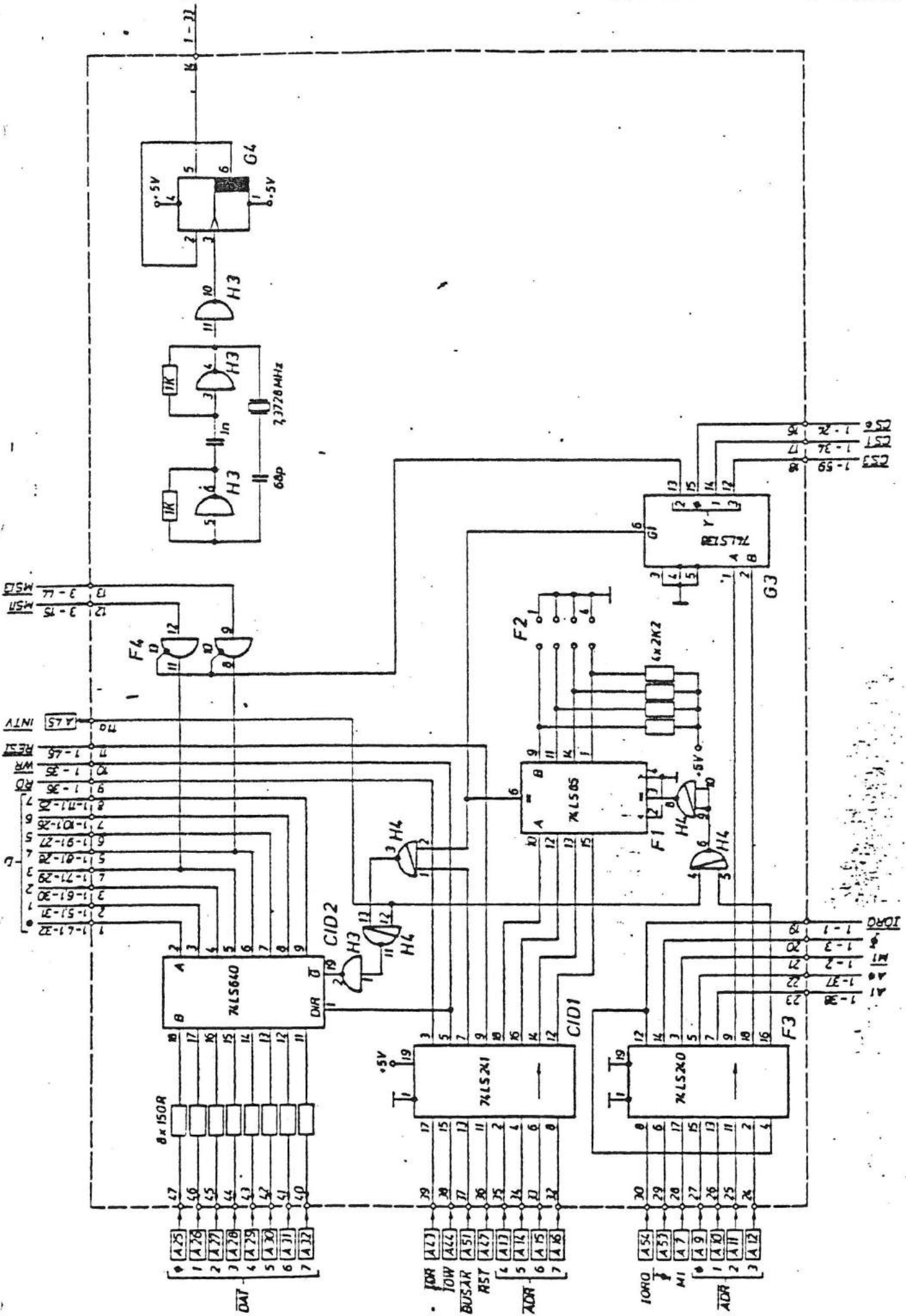


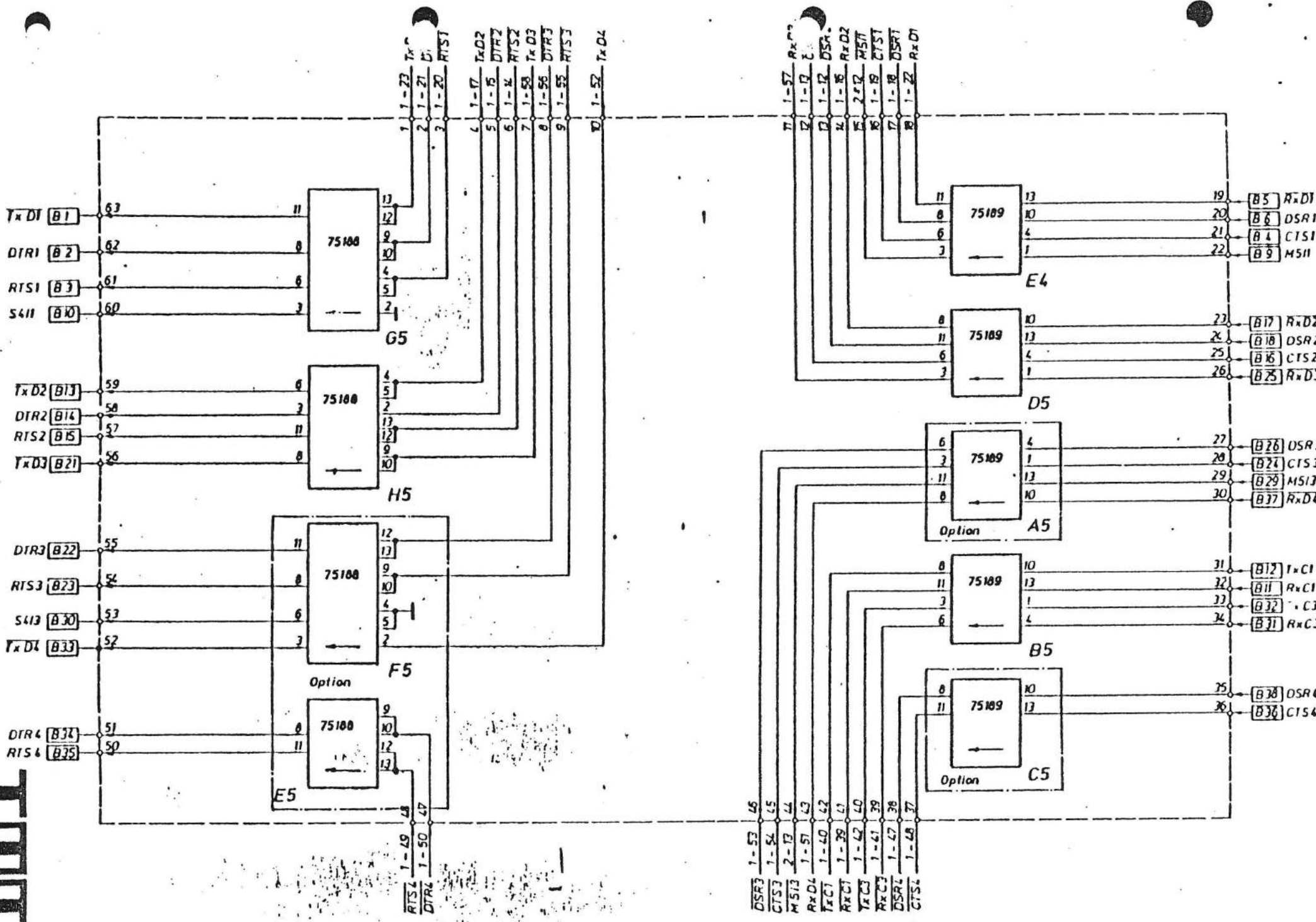


Diese Unterlage ist ausschließlich unser Eigentum.
 Ohne Zustimmung darf sie nicht veröffentlicht werden.
 Alle Rechte vorbehalten.

Ausgabe Ist	Name	1982 Jahr	Tag 26.3	Name	Maßstab 1/1	Blatt 7	Bestell-Nr. 2 MIC 586 E
Name	Obj.						Blattschaltbild
Aut.							







ITT 3030

SOFTWARE

Schnittstellenbeschreibung

MOS - Anteile

für CP/M und BOS Betriebssysteme

Zeichenweiser Betrieb

Der Strom-Routine wird das jeweilige Zeichen übergeben bzw. von ihr übernommen. Um die Organisation der Daten im eigentlichen Gerät braucht sich der Benutzer nicht zu kümmern (z. B. Blockung usw.). Der zeichenweise Betrieb ist standardmäßig mit folgenden Treibern/Geräten möglich:

- System Ein/Ausgabegerät
- Tastatur
- Display
- Drucker

Bei Fehlern in den Routinen wird CARRY = 1 gesetzt. Die jeweilige Fehlerrückmeldung wird in den einzelnen Routinen genauer angegeben.

Schnittstelle für Command-Output und Lister

Funktion CO, LO

Aufruf: Ausgabezeichen RC
 mit Interpretation Sonderzeichen

Rückgabe: Meldung Fehler CARRY = 1
 Treiberidentifikation RA wenn CARRY = 1
 Status RB / RC wenn CARRY = 1

Bemerkung: Register RH / RL und RD / RE sind unverändert.
 Rückgabe Status siehe DROPEN

Schnittstelle für Command-Input

Funktion CI

Aufruf: -

Rückgabe: Eingabezeichen RA
 Meldung Fehler CARRY = 1
 Treiberidentifikation RA wenn CARRY = 1
 Status RB / RC wenn CARRY = 1

Bemerkung: Register RH / RL und RD / RE sind unverändert.

Man beachte, daß Status und Treiberidentifikation nur gültig sind, wenn CARRY gesetzt ist. Andernfalls sind beide undefiniert. Eine Randbedingungs-Fehlerbehandlung muß deshalb immer mit der CARRY - Abfrage ansetzen.

Schnittstelle für Reader-Input

Funktion RI

Aufruf: -

Rückgabe: Empfangenes Zeichen RA

Bemerkung: Alle Register sind verändert.
 Einstellung Baudrate über
 Funktion V24T Initialisieren.

Direkte Eingänge Kanaltreiber

Funktionen: DISPE, TASTE, DRUCE

Die Funktionen haben die gleiche Schnittstelle wie CO, CI, LO, nur werden immer folgende Funktionen ausgeführt:

- DISPE ausgeben auf Display ohne Interpretation von Sonderzeichen
- TASTE Taste von Tastatur abholen
- DRUCE Zeichen auf Drucker ausgeben

Stelle Drucker ein:

Funktion: DROPEN

Aufruf: Baud-Rate (Zählrate) RD / RE
 Druckersteuerbytes RB / RC
 Steuerwort RL
Protokolllänge RH

Steuerwort: RL

- Bit 2⁰ 1 = Wandlung 7 Bit nach 8 Bit USASCII
 0 = Wandlung 8 Bit nach 7 Bit USASCII
- Bit 2¹ 0 = 8 Bit ohne Parität
 1 = 8 Bit mit gerader Parität
- Bit 2² 0 = kein Protokoll
 1 = Protokoll: ETX/ACK
- Bit 2⁴ 0 = warten bis Drucker bereit
 1 = sofort zurück, wenn Drucker nicht bereit
- Bit 2⁷ 0 = ./.. 16 Mode
 1 = ./.. 1 Mode

Rückgabe: Meldung Fehler CARRY = 1
 Status in B: Bit 2⁰ = 1 keine Verbindung,
 Drucker aus
 Bit 2¹ = 1 Busy, wenn Bit 2⁴ = 1
 Bit 2³ = 1 Handshake Fehler

Bemerkung: Alle Register und Flags sind verändert.
 Werte für Baud-Rate (siehe Funktion V24T)
 Protokollänge nur massgeblich, wenn Bit 2² = 1.

Schnittstelle V24-Treiber

Funktion: V24T

1. Funktion Initialisieren (01)

Aufruf:	Funktionsnummer	RA
	Baudrate (Zählrate)	RD/RE
	Steuerbyte	RC

Rückgabe: keine

Tabelle Zählrate für Baudrate:

<u>Mode</u>	<u>Baud-Rate</u>	<u>Werte (Register RD/RE)</u>	
	4800	0807	7
	2400	080E	14
./ .16.	1200	081B	27
senden u.	600	0837	55
empfangen	300	086D	109
	110	4877	
./ . 1	19200	081B	
nur senden	9600	0837	

Steuerbyte RC:

Bit 2⁰ 0 = Wandlung 8 --> 7 Bit USASCII
 1 = Wandlung 7 --> 8 Bit USASCII

 Bit 2¹ 0 = ohne Parität
 1 = mit Parität

 Bit 2² 0 = mit gerader Parität
 1 = mit ungerader Parität

 Bit 2³ 0 = 1 Stopbit
 1 = 2 Stopbits

 Bit 2⁷ 0 = ./.. 16 Mode
 1 = ./.. 1 Mode

2. Funktion Empfangen (80)

Aufruf:	Funktionsnummer	RA
	Pufferadresse	RB/RC
	Länge	RD/RE
Rückgabe:	Länge	RD/RE

3. Funktion Senden (81)

Aufruf:	Funktionsnummer	RA
	Pufferadresse	RB/RC
	Länge	RD/RE
Rückgabe:	Länge	RD/RE

Switch-Byte

Es besteht die Möglichkeit alle CO - Ausgaben parallel auf dem Drucker auszugeben (List command output). Diese Funktion wird mit dem Switch-Byte gesteuert.

Bit 2^0 = 0 kein Ausdruck

Bit 2^0 = 1 Ausdruck

Funktion:	SWGET	Switch-Byte holen
	SWPUT	Switch-Byte ablegen

Aufruf:	bei SWPUT	Switch-Byte	RA
	bei SWGET	Switch-Byte	RA

Bemerkung: Alle anderen Register bleiben unverändert.

Tastatureingabe, Bildschirmanzeige und Statusabfrage

Funktion: CSTS

Aufruf: -

Rückgabe: keine Taste RA = 0, CARRY=0, ZERO=1
 Taste RA = OFFH, CARRY=0, ZERO=0
 Übertragungsratenfehler RA = OFFH, CARRY=1

Bemerkung: Bis auf RA sind alle Register unverändert. Die Taste wird nicht verarbeitet und muß mit CI abgeholt werden.

Eingabe mit gleichzeitiger Anzeige auf dem Bildschirm

Funktion: TI

Es wird auf ein Zeichen vom CI-Strom gewartet und übergeben mit gleichzeitigem Ausgeben des Zeichens auf den CO-Strom.

Aufruf: -

Rückgabe: Eingabezeichen RA
 Meldung CARRY=1 falls Fehler
 Treiberidentifikation RA wenn CARRY=1
 Status RB / RC wenn CARRY=1

Textstring ausgeben

Funktion: TXCO

Aufruf: Adresse String RH / RL

Rückgabe: -

Bemerkung: RH/RL und RD/RE sind unverändert. Es wird ein String über CO ausgegeben. Der String muß folgenden Aufbau haben:

1. Byte Länge der Ausgabe
2. Byte n Byte Ausgabedaten

Spezialfunktionen Display

Funktion: DISC

Es werden Spezialfunktionen des Displays aufgerufen. Die genaue Schnittstelle und die Bedeutung ist der Beschreibung über den Display-Kanaltreiber zu entnehmen.

Eingabe Hexadezimalwerte

Funktion: EXSTD

Mit Hilfe einer Steuertabelle wird eine Eingabe gesteuert.

Aufruf: Adresse Steuertabelle RH / RL

Aufbau Steuertabelle

1.	Byte	Anzahl einzugebende Werte
2. + 3.	Byte	Standardbelegung erster Eingabewert
4. + 5.	Byte	Standardbelegung zweiter Eingabewert
usw.		

Rückgabe: Im Stack eingegebene Werte, letzter Wert als oberster Wert.

Bemerkung: Alle Register sind unverändert.

Es werden sovielen Werte eingegeben wie angegeben sind. Die einzelnen Werte sind durch Komma getrennt. Soll bei einem Wert der Standardwert genommen werden, so wird nichts eingegeben.

Beispiel:

Tabelle:	5, 1234H, 1, 2, 5678H, 8
Eingabe:	5, 6, 3
Übergebene Werte:	5, 6, 1, 3, 5678H, 8

Seite umschalten

Funktion: SWITCH
Aufruf: Seite in RC
Programmstartadresse in RH/RL
Rückgabe: -
Bemerkung: Register bleiben unverändert.
Rückschaltung muß vom gestarteten Programm
verwaltet werden.

Hardware - Erkennung

Funktion: HWERK
Aufruf: -
Rückgabe: Adresse eines Parameterblockes in H/L

Ausgabe Hexadezimalwert

Funktion: LBYTE
Aufruf: auszugebendes Byte RA
Rückgabe: -
Bemerkung: Register RH/RL und RD/RE sind unverändert.
Das Byte wird als zwei Zeichen über CO ausgegeben.

Ausgabe Hexadezimaladresse

Funktion: LADR
Aufruf: auszugebende Adresse RH/RL
Rückgabe: -
Bemerkung: Register RH/RL und RD/RE sind unverändert.
Die Adresse wird als vier Zeichen über CO ausgegeben.

Ausgabe CR, LF

Funktion: CRLF
Aufruf: -
Rückgabe: -
Bemerkung: Register RH/RL und RD/RE sind unverändert.
Es werden über CO die Zeichen CR (ODH) und LF (OAH) ausgegeben.

WandelroutinenASCII 7 Bit nach ASCII 8 Bit

Funktion: CODW78

Aufruf: zu wandelndes Zeichen RA

Rückgabe: gewandeltes Zeichen RA

Bemerkung: alle anderen Register bleiben unverändert.

ASCII 8 Bit nach ASCII 7 Bit

Funktion: CODW87

Aufruf: zu wandelndes Zeichen RA

Rückgabe: gewandeltes Zeichen RA

Bemerkung: alle anderen Register bleiben unverändert.

Hexadezimalzeichen nach ASCII - Zeichen

Funktion: CONV

Aufruf: zu wandelndes Zeichen RA

Rückgabe: ASCII - Zeichen RA und RC

Bemerkung: alle anderen Register bleiben unverändert.

Es werden nur die untersten vier Bit des Registers in ein ASCII - Zeichen gewandelt.

ASCII - Zeichen nach Hexadezimalwert

Funktion: NIBBLE

Aufruf: ASCII - Zeichen RA

Rückgabe: Hexawert in den untersten 4 Bit RA
CARRY = 0 gültiges Zeichen
CARRY = 1 kein Hexadezimalzeichen

Bemerkung: alle anderen Register bleiben unverändert.

HilfsroutinenRegister RH/RL + 1 und Vergleich mit Register RD/RE

Funktion: HILO
 Aufruf: Vergleichswerte RH/RL, RD/RE
 Rückgabe: -
 CARRY = 1 ZERO = 1 RH/RL = 0
 CARRY = 0 ZERO = 0 RH/RL < RD/RE
 CARRY = 0 ZERO = 1 RH/RL = RD/RE
 CARRY = 1 ZERO = 0 RH/RL > RD/RE
 Bemerkung: nur RA ist verändert.

Bereich mit Konstante füllen und Test ob richtig gefüllt

Funktion: FILLM
 Aufruf: Anfangsadresse RH/RL
 Endeadresse RD/RE
 Füllkonstante RC
 Rückgabe: Zelle nicht schreibbar CARRY
 Bemerkung: alle Register verändert.

Bereich verschieben

Funktion: MOVEM
 Aufruf: Quelladresse RH/RL
 Endeadresse RD/RE
 Zieladresse RB/RC
 Rückgabe: -
 Bemerkung: alle Register verändert.

Achtung: Bei Überlappung von Quell- und Zielbereich kann der Quellbereich zerstört werden. Es wird in aufsteigender Richtung umgeladen.

Stackpointer holen und merken

Funktion: SPGET

Aufruf: -

Rückgabe: Stack Pointer RH/RL

Bemerkung: außer RH/RL alle Register unverändert.

Speicherendeadresse laden

Funktion: MEMCK

Aufruf: -

Rückgabe: Endeadresse Anwender-Speicher RH/RL

Bemerkung: alle anderen Register unverändert.

Die Speicherendeadresse wird jeweils beim CP/M Kaltstart und Warmstart getestet.

Speicherendeadresse setzen

Funktion: MEMSET

Aufruf: Setzen Ende des Anwender-Speichers RH/RL

Rückgabe: -

Bemerkung: alle Register unverändert.

Random = Zahl = laden

Funktion: RNDZ

Aufruf: -

Rückgabe: Random Zahl RA

Bemerkung: alle anderen Register unverändert.

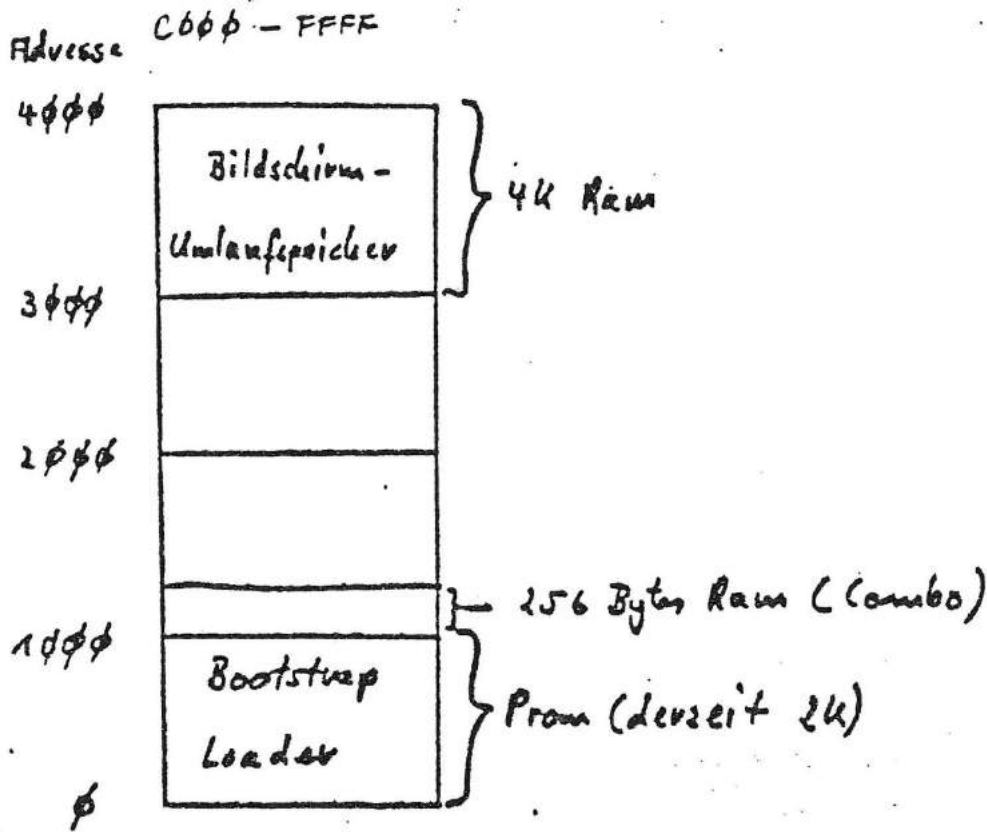
A N H A N G

1. Tabelle der Funktionsadressen:

ADR	NAME	FUNKTION
FE00	SWITCH	Speicherumschaltung
FE03	CI	Command Input zeichenweise
FE06	RI	Reader Input zeichenweise
FE09	CO	Command Output zeichenweise
FE0C		Reserviert
FE0F	LO	Lister Output zeichenweise
FE12	CSTS	Consolstatus abfragen
FE15	HWERK	Rückgabe Hardware
FE18	TI	Console CI mit CO
FE1B	HILO	HL + 1 Vergleich mit DE
FE1E	TXCO	Textausgabe über CO
FE21		Reserviert
FE24		Reserviert
FE27	FILLM	Bereich füllen
FE2A		Reserviert
FE2E		Reserviert
FE32		Reserviert
FE36		Reserviert
FE3A	SWGET	Switch - Byte holen
FE3E	SWPUT	Switch - Byte setzen
FE42		Reserviert
FE46		Reserviert
FE4A	MEMCK	Speicheradresse holen
FE4E	MEMSET	Speicheradresse setzen
FE52	SPGET	Stack-Pointer holen und merken
FE55	EXSTD	Eingabe Hexadezimalwerte

ADR	NAME	FUNKTION
FE58		Reserviert
FE5B		Reserviert
FE5E	V24T	V24 Treiber
FE61		Reserviert
FE64	CODW78	Codewandlung 7 nach 8 bit
FE67	CODW87	Codewandlung 8 nach 7 bit
FE6A		Reserviert
FE6D		Reserviert
FE70	DROPEN	Einstellung Drucker
FE73	TASTE	Tastatur Treiber
FE76	DISPE	Display Treiber (Zeichen direkt a. Bildschirm)
FE79	DRUCE	Drucker Treiber
FE7C		Reserviert
FE7F	MOVEM	Bereich verschieben
FE82	LBYTE	Hexadezimal-Byte ausgeben
FE85	LADR	Hexadezimal-Adresse ausgeben
FE88	CONV	Hexadezimalwert nach ASCII-Zeichen
FE8B	NIBBLE	ASCII-Zeichen nach Hexadezimalwert
FE8E	CRLF	CR, LF auf CO ausgeben
FE91		Reserviert
FE94		Reserviert
FE9A	FAOR	Steuerfunktion Display <i>Reliert mit Fun Chon...</i>
FE9D	RNDZ	Random - Zahl lesen
FEA1	DENTRY	Eingang Floppytreiber
FEA4	SENTRY	Select-Eingang Floppytreiber

Speicher und 10-Aufteilung Seite 8



10 - Adressen

Combo - Chip		E0	-	EF
Wait - Zyklen	OP-Code	F0		
"	Memory	F1		
"	IO	F2		
DTR/RTS	setzen	F3		
DSR/CTS	lesen	F0		
Page	umschalten	F6		
Kombi - Interface I		20	-	3F
Floppy - Interface		50	-	5F
Winchester - Platte		10	-	1F

- 0. Einleitung
- 1. Schnittstellen-Übersicht
 - 1.1 Klassifikation
 - 1.2 Aufbau des Treibereingangs
 - 1.3 Zwischenrückgabe
 - 1.4 Aufrufparameter
 - 1.5 Rückgabeparameter
- 2. Beschreibung der Treiberfunktionen
- 3. Datenorganisation auf einer Diskette
 - 3.1 Gliederung in Spuren und Sektoren
 - 3.2 Form eines Sektors

0. Einleitung

Der Floppy-Treiber ist ein Kanalprogramm zum Betrieb von maximal drei Doppelkopf-Laufwerken am SKS-MC80 System.

Die Steuerung des Treibers erfolgt über Aufruf- und Rückgabeparameter, welche in den Registern und Flags übergeben werden.

Der Floppy-Treiber ist nicht reentrant und normalerweise nicht unterbrechbar. Bei langdauernden Operationen unter Hardware-Kontrolle hat der Anwender die Möglichkeit der Zwischenrückgabe, um eigene Programme zwischenzeitlich abzuarbeiten.

1. Übersicht über Funktionen und Aufbau des Treibers

1.1 Treiberfunktionen

Folgende Funktionen lassen sich über den Treiber abwickeln

- Sektorweise Lesen und Schreiben im quasi - wahlfreien Zugriff zu Segmenten von 256 Bytes Länge (128 Bytes bei 8" Disketten).
- Schreib-/Lesekopfpositionierung spur- und sektorweise
- Initialisierung des Interfacecontrollers und der Laufwerke

1.2 Treibereingänge

Es existieren zwei Haupteingänge

1. Select Entry (SENTRY):

Laufwerksnummer und Diskettenseite werden in einem Parameter explizit angegeben.

2. Direct Entry (DENTRY):

Laufwerksnummer und Diskettenseite ändern sich gegenüber dem vorhergehenden Aufruf nicht.

1.3 Zwischenrückgabe Option

Einige Funktionen des Treibers dauern gegenüber den Programmlaufzeiten lang und laufen ausschliesslich unter Hardware- und Interface Kontrolle. (Überwiegend Positioniervorgänge).

In diesen Fällen wird der Treiber über eine Zwischenrückgabe verlassen.

Der Anwender hat dann die Möglichkeit, andere Programme abzuarbeiten (Hintergrundprogramme, Interruptserviceroutinen usw.).

Alle Register und Flags stehen zur Verfügung. Der Prozessor beginnt mit gesperrtem Interrupt.

Keine der Hardware-Funktionen ist zeitkritisch. Es liegt im Belieben des Anwenders, wie lange das Zwischenprogramm dauert.

Die Adresse des Zwischenprogramms muss mindestens einmal, bei Initialisierung des Treibers, übergeben werden, kann aber beliebig oft durch Neuinitialisierung geändert werden. Die Rückkehr in den Treiber erfolgt mit einem Return-Befehl (Rückkehradresse liegt im Stack!).

1.4 Allgemeine Aufrufparameter

Die Parameterübergabe bei Treiberaufruf geschieht funktions-spezifisch.

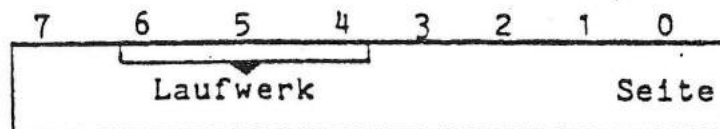
Die folgenden Parameter sind jedoch bei allen Funktionen gemeinsam und müssen in jedem Fall übergeben werden:

Eingang SENTRY: Funktionsnummer in RA
 Laufwerk, Seite in RL

Eingang DENTRY: Funktionsnummer in RA
 (Laufwerk, Seite wie bisher)

Die Funktionsnummern sind in Tabelle I zusammengefasst.

Der Parameter Laufwerk/Seite in RL
 hat die Form



Die funktions-spezifischen Parameter werden bei der Beschreibung der einzelnen Funktionen behandelt.

1.5 Allgemeine Rückgabeparameter

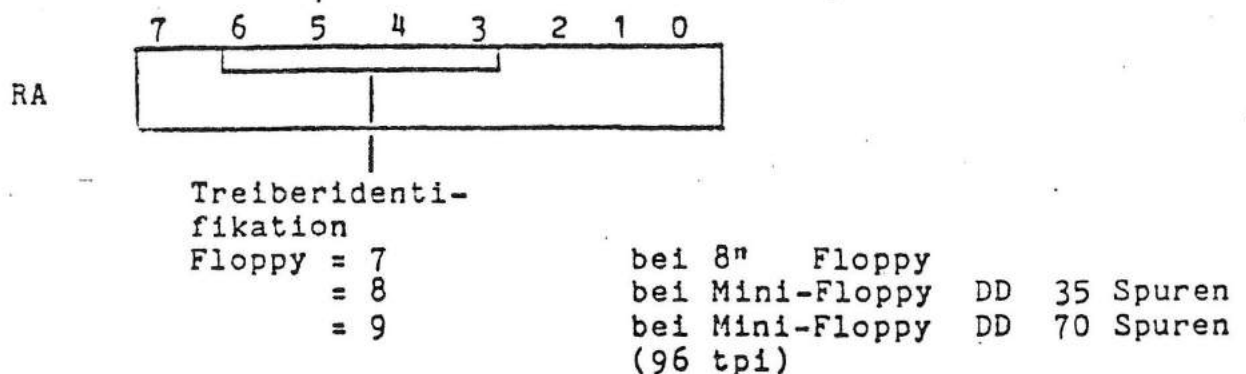
Folgende Parameter werden von allen Funktionen zurückgegeben:

Frühwarnung: ZERO = 1
 Fehler: CARRY = 1
 Status: in RB/RC (siehe Tabelle II)

Identifikation/Wiederholung in RA

Alle Register und Flags, ausser RH/RL werden verändert.

Der Parameter Identifikation hat die Form:



Anmerkung: Die Funktion "Treiberstatus lesen" (OF) bildet bezüglich Aufruf- und Rückgabeparameter Ausnahmen. Siehe Funktionsbeschreibungen.

Befehlsklasse IO		Befehlsklasse CTRL	
Nr.	Bit 2 ⁷ = 1	Nr.	Bit 2 ⁷ = 0
-	-	-	-
81	Sektorweise Schreiben bis zur angegebenen Pufferlänge (ohne read after write)	01	Initialisieren, Laufwerk/Spur selektieren
82	Sektorweise Lesen bis zur angegebenen Pufferlänge	-	-
83	Sektorweise Schreiben bis zur angegebenen Pufferlänge (mit read after write)	-	-
84	Positionieren auf Spur = TN; Sektor = SN (nur für formatierte Disk)	-	-
-	-	-	-
-	-	-	-
-	-	-	-
-	-	0A	Schreib-Lesekopf über Spur 0, Sektor 1 positionieren (BOD)
-	-	0C	Positionieren über Spur = TN, Sektor = 1 (für unformatierte Disketten)
-	-	0D	Spur formatieren
-	-	0E	Doorlock bedienen optional
-	-	0F	Treiberstatus lesen

Tabelle I: Funktionsnummern der Floppy Treiber

Reg. Bit	Bedeutung	Flag
	7 S/L-Kopf steht über letzter Spur, letzter Sektor	
	6 S/L-Kopf steht über erster Spur, erster Sektor	
RB	5 -	
	4* Laufwerk Interface unklar, Diskette fehlt	
	3 -	
	2* Schreibschutz	
	1 -	
	0 -	
	7 CRC - Fehler	SIGN
	6 Frühwarnung	ZERO
	5 -	
	4 Sektor nicht gefunden	
RC	3 -	
	2 -	
	1 Unzulässiger Vorgang oder Parameter	
	0 Fehlersummenmeldung (Anforderung nicht vollständig ausgeführt)	CARRY

Tabelle II Zusammenfassung der Statusrückmeldungen.

* siehe Funktion Treiberstatus lesen.

2. Beschreibung der Treiberfunktionen

2.1 Funktion Sektorweise Schreiben (81)

Es wird ab momentaner Position die im Aufruf angegebene Anzahl Datenbytes geschrieben. Der Schreib-Lesekopf steht nach dem Schreiben auf dem nächsten Sektoranfang. Es erfolgt kein Prüfllesen.

Aufruf:	Pufferadresse	RB/RC
	Anzahl Datenbytes	RD/RE
Rückgabe:	Position Sektor/Spur	RD/RE
	Schreibfehler	SIGN = 1, CARRY = 1

2.2 Funktion Sektorweise Lesen (82)

Es wird ab der momentanen Position die im Aufruf angegebene Anzahl Datenbytes gelesen. Der Schreib-Lesekopf steht nach dem Lesen auf dem nächsten Sektoranfang.

Aufruf:	Pufferadresse	RB/RC
	Anzahl Datenbytes	RD/RE
Rückgabe:	Position Sektor/Spur	RD/RE
	Lesefehler	SIGN = 1, CARRY = 1

2.3 Funktion Sektorweise Schreiben (83)

Wie Funktion 81, jedoch erfolgt anschliessend ein Prüfllesen.

2.4 Funktion absolutes Positionieren auf Spur und Sektor (84)

Der Schreib-Lesekopf wird auf die angegebene Position gebracht.

Aufruf:	Sektor	RD
	Spur	RE
Rückgabe:	Position Sektor/Spur	RD/RE

2.5 Funktion Initialisieren, Laufwerk/Spur selektieren (01)

Es wird die Zwischenrückgabeadresse eingestellt und an Laufwerk und Spur selektiert.

Aufruf:	Adresse eines Zwischenprogrammes	RB/RC
	Laufwerk/Seite (Belegung siehe 1.4)	RE

Rückgabe: siehe Allgemeine Rückgabeparameter (1.5).

2.6 Funktion Restore (0A)

Der Schreib-Lesekopf wird auf Spur 0, Sektor 1 positioniert.

Aufruf: siehe Allgemeine Aufrufparameter (1.4).

Rückgabe: siehe Allgemeine Rückgabeparameter (1.5).

2.7 Funktion Positionieren Spur (OC)

Es wird die angegebene Spur anpositioniert und das Sektorregister mit 1 geladen. Diese Funktion kann zum Positionieren unformatierter Disketten verwendet werden.

Aufruf: Spurnummer in RE

Rückgabe: siehe Allgemeine Rückgabeparameter (1.5).

2.8 Funktion Spur formatieren (OD)

Aufruf: RB/RC Beginn der Spurinformatio

Rückgabe: siehe Allgemeine Rückgabeparameter (1.5)

2.9 Doorlock bedienen - optional (OE)

Mit dieser Funktion kann in mit Doorlock ausgerüsteten Geräten dieser aktiviert bzw. deaktiviert werden.

Aufruf: RE = 00 Doorlock aus
RE = FF Doorlock ein

Rückgabe: siehe Allgemeine Rückgabeparameter

2.10 Funktion Treiberstatus lesen (OF)

Es wird der momentane Treiberstatus gelesen. Dabei werden nur die in Tabelle II mit * gekennzeichneten Bits zurückgemeldet.

Aufruf: siehe Allgemeine Aufrufparameter (1.4)

Rückgabe: RB (siehe Tabelle II)

3. Datenorganisation auf einer Diskette

3.1 Gliederung in Spuren und Sektoren

Eine Diskette hat 35 (77) Spuren zu je 16 (26) Sektoren.

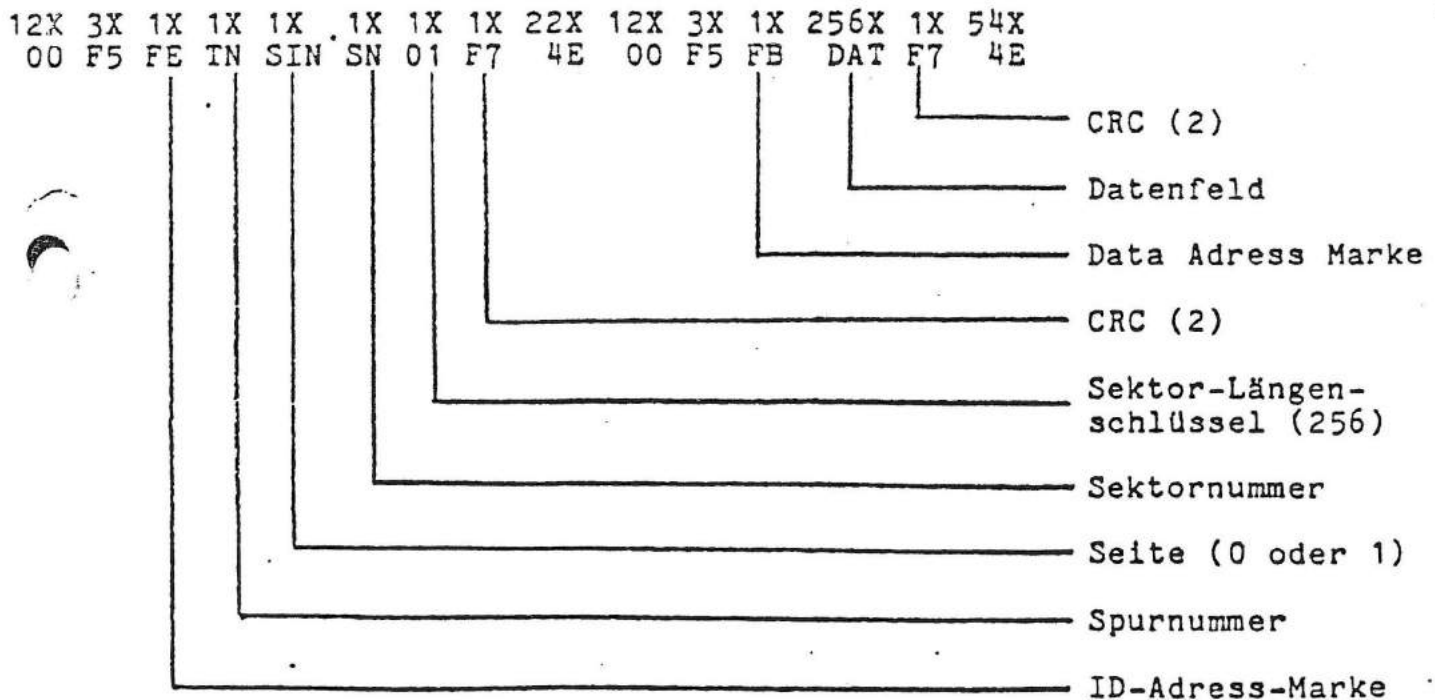
1 Sektor hat eine Länge von 256 (128) Bytes.

Die Nummerierung der Spuren beginnt bei 0, die der Sektoren bei 1.

Die Zahlen in Klammern beziehen sich auf 8" Disketten.

Alle Ein/Ausgabeoperationen beginnen an der Sektorkante, die durch einen vorangegangenen IO- oder Steuerbefehl erreicht wurde.

3.2 Die Form eines Sektors



TYPE 3080

SOFTWARE

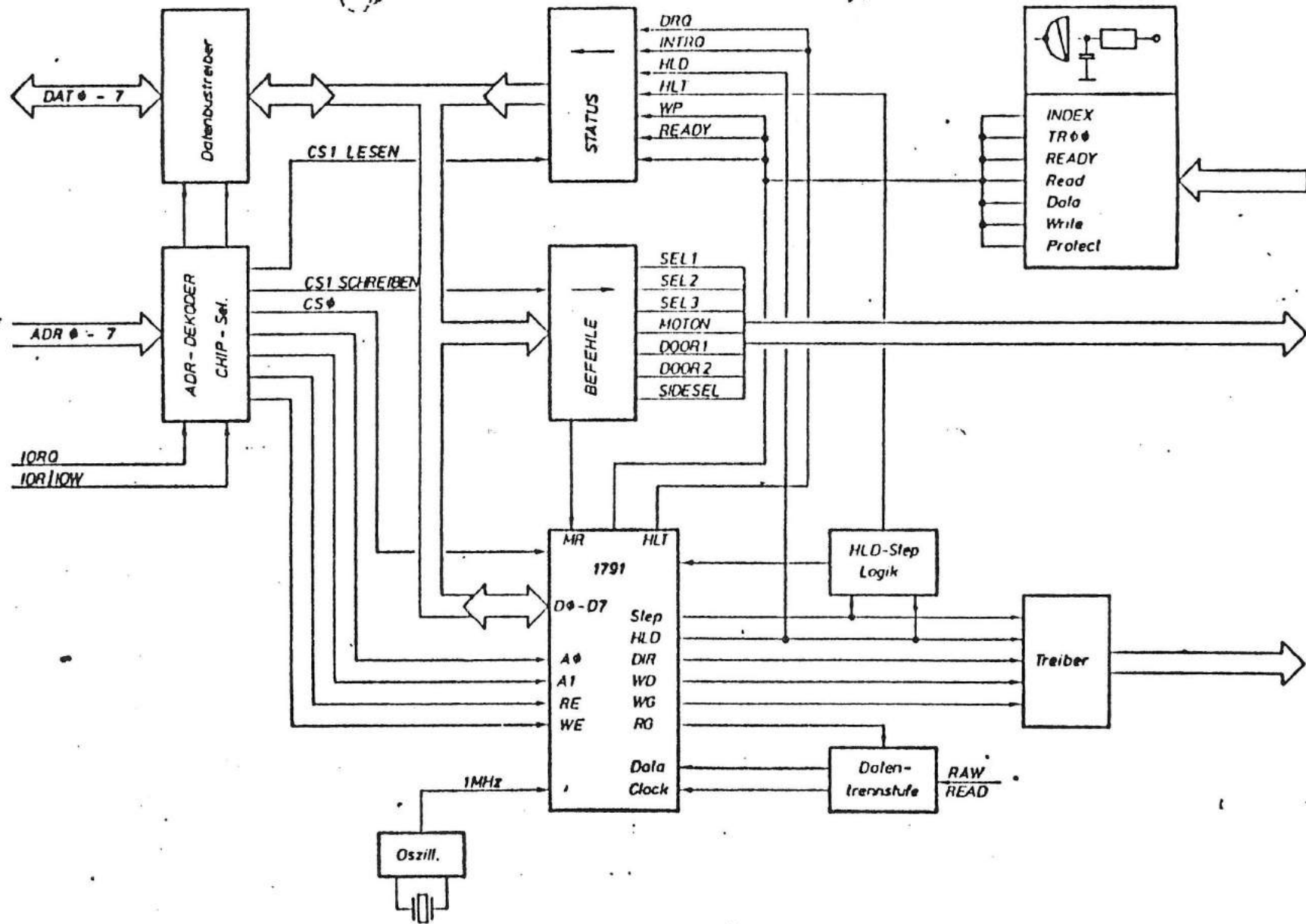
1

BESCHREIBUNG

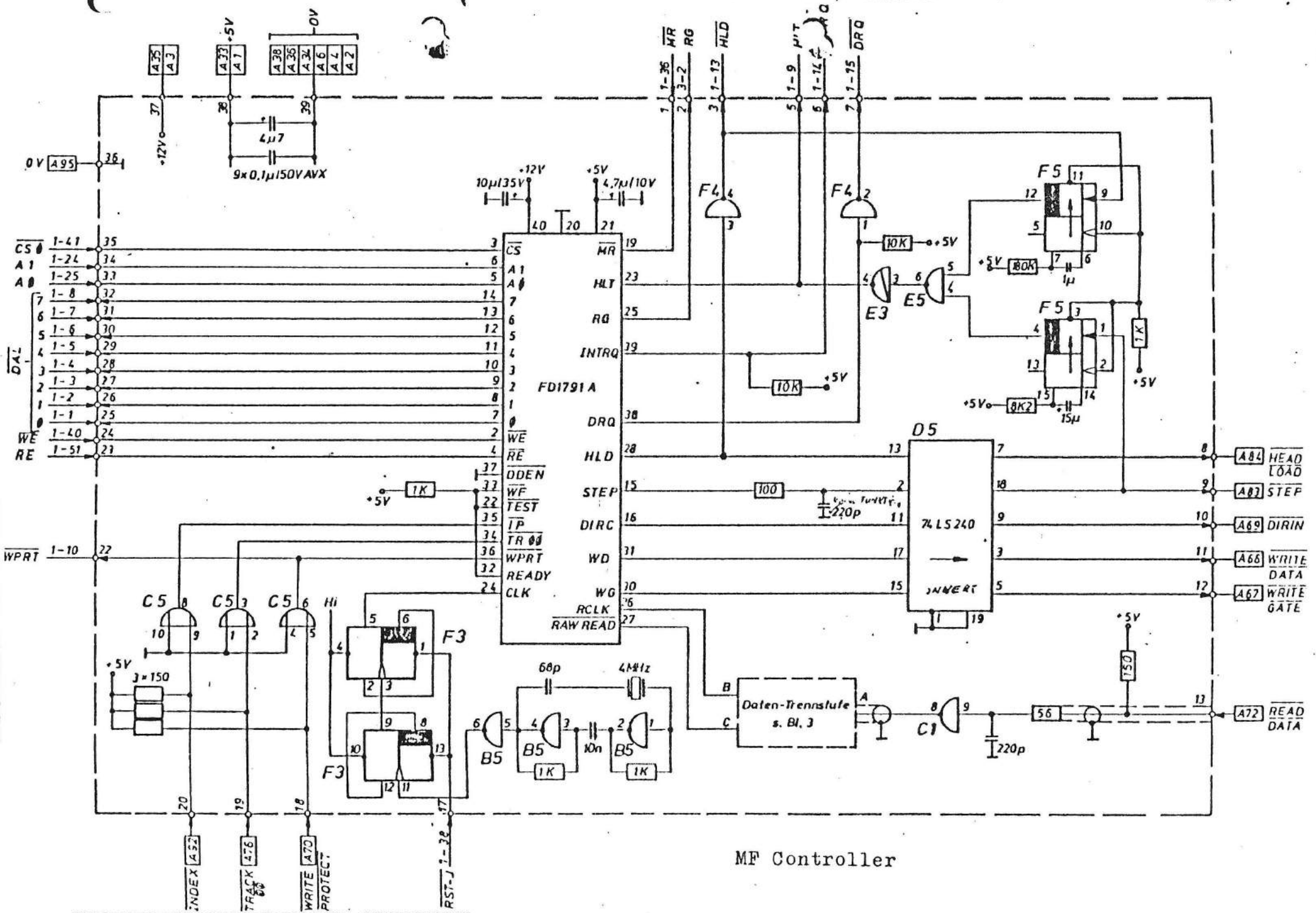
Schnittstelle

Floppy - Treiber

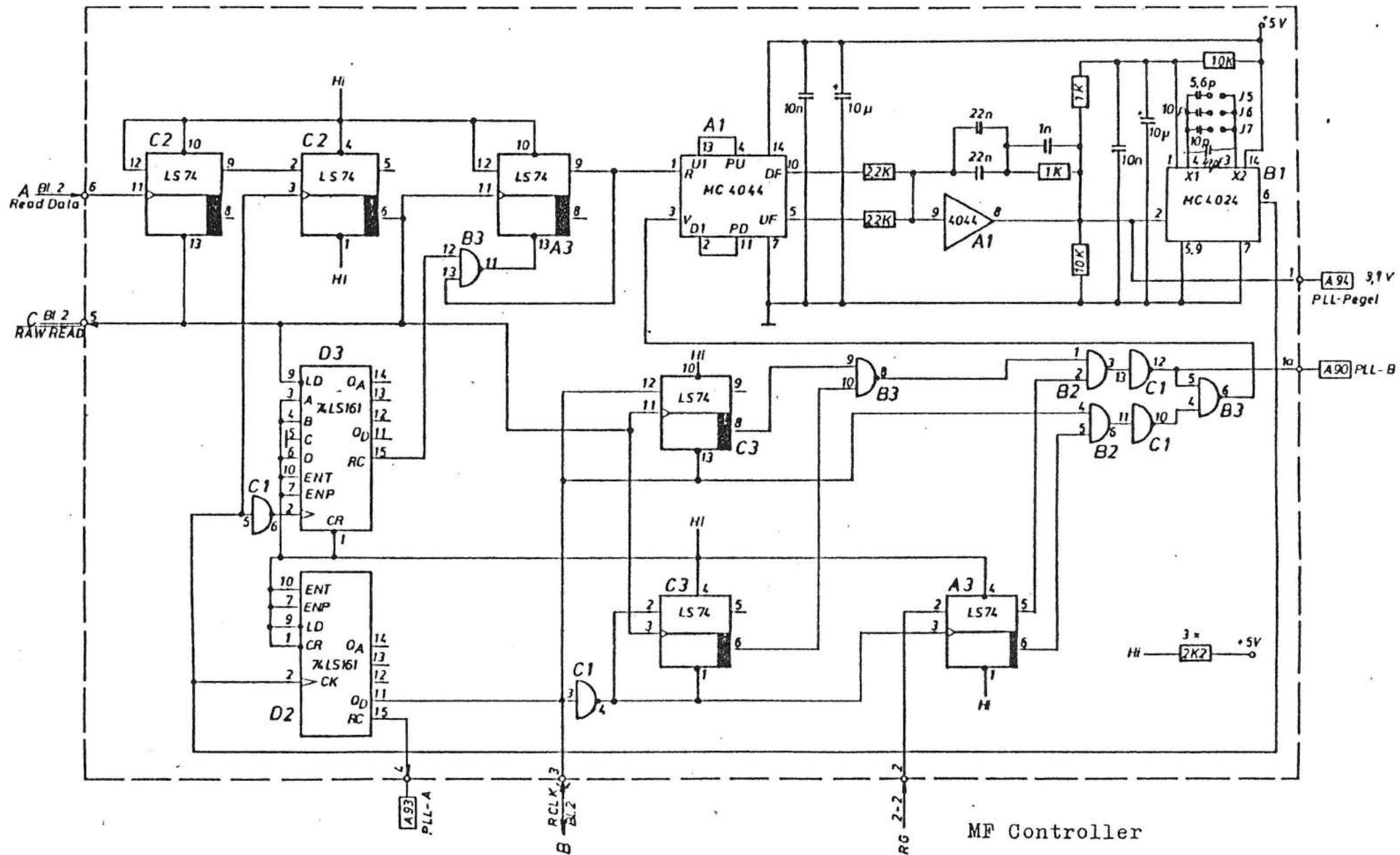
System - Schnittstelle



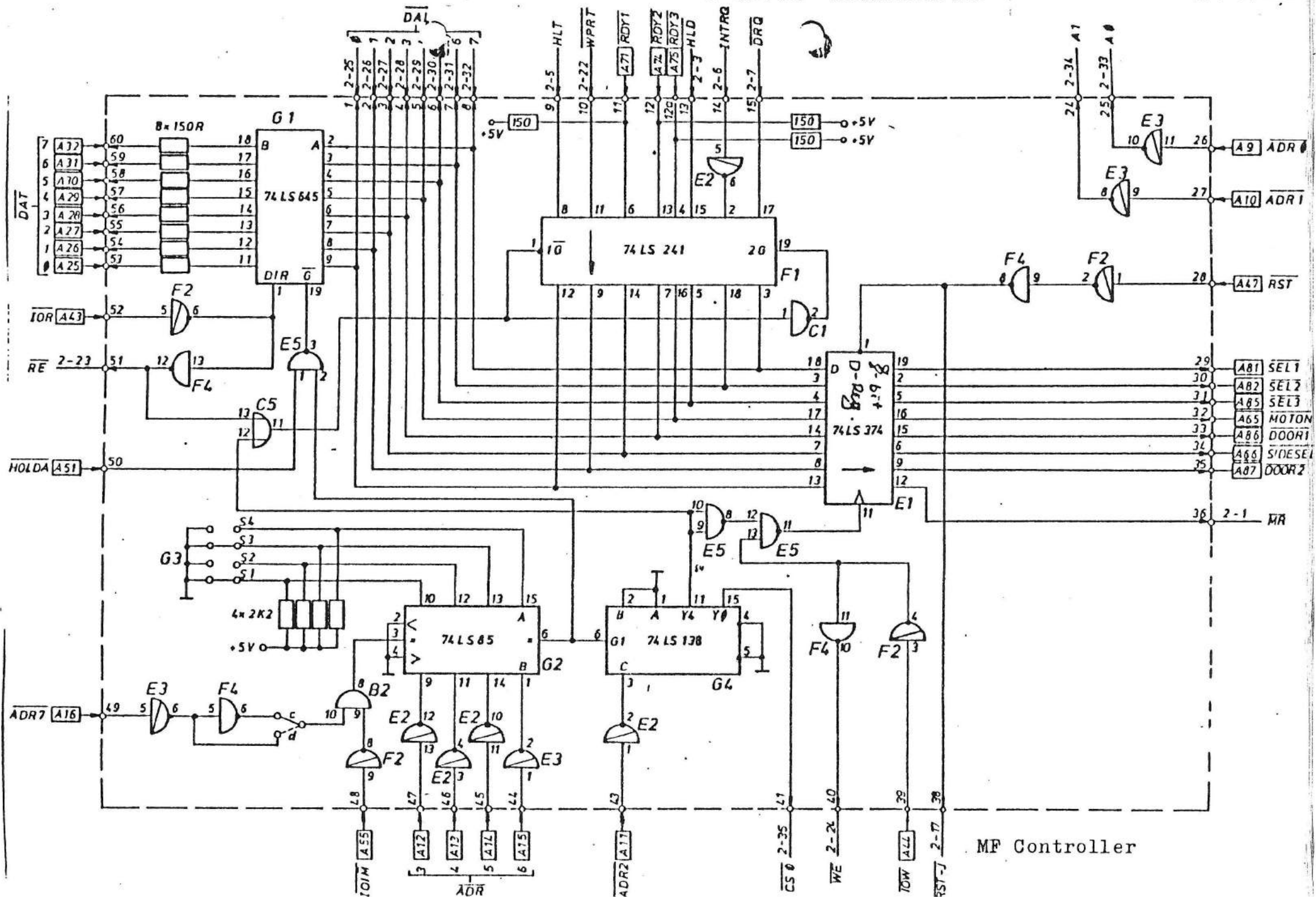
Floppy - Schnittstelle



MF Controller



MF Controller



MF Controller

ITT 3030

AUF WACHSTUM PROGRAMMIERT !

Technische Funktionsbeschreibung

=====

1. Allgemeines

Die Baugruppe "Mini-Floppy-Controller" (Doppelte Dichte) ist eine Grundelektronik, die zur Datenübertragung zwischen Rechner und Floppy-Speicher dient.

Floppy-Laufwerke (maximal drei) von Typ Philips X 3112 oder ähnliche Geräte können über den Controller am ITT 3030-BUS oder anderen, BUS-orientierten Rechnersystemen betrieben werden.

Beim Datentransfer zwischen System-BUS und Floppy hat der Controller im wesentlichen die folgenden Aufgaben:

- BUS-Entkoppelung
- Zwischenpufferung der Geräteadresse und der Steuersignale
- zeichenweise Datenpufferung
- Datendekodierung
- Datenformatierung
- Schreibtakterzeugung

Der Controller ist unter Verwendung des Floppy-Disk-Formatter/Controller-Bausteins FD 1791 B-02 von Western Digital realisiert.

2. Funktionsgruppenübersicht

Die zwischen CPU und Basis-Controller ausgetauschte Information unterteilt sich in:

- a) Befehle an die Laufwerke
- b) Befehle an den Formatter/Controller-Baustein
- c) Statusinformation des Basis-Controllers und der Laufwerke
- d) Statusinformation des Formatter/Controller-Bausteins
- e) Schreibdaten
- f) Lesedaten

ITT 3030

AUF WACHSTUM PROGRAMMIERT !

Elektrische Spezifikation

=====

Stromaufnahme:

+ 5 V : \leq 350 mA

+ 12 V : \leq 12 mA

Monozeiten:

F 5/12 : 40 bis 100 ms

F 5/4 : 20 bis 60 ms

Oszillator:

4 MHz \pm 10^{-4}

Brückenbelegungsliste

1. S1 - S4 Adreßbereich

vorzugsweise kaschiert: S1, S3 geschlossen
S2, S4 offen

entspricht Adresse $5\emptyset_H$

2. J5 - J7 dient zum Abgleich der PLL-Schaltung

J5 geschlossen: schaltet zum festkaschierten 47pF
einen 5,6pF

J6 geschlossen: schaltet zum festkaschierten 47pF
einen 10pF

J7 geschlossen: schaltet zum festkaschierten 47pF
einen 10pF

Brückenbelegungsliste

1. S1 - S4 Adreßbereich

vorzugsweise kaschiert: S1, S3 geschlossen
S2, S4 offen

entspricht Adresse $5\sigma_H$

2. J5 - J7 dient zum Abgleich der PLL-Schaltung

J5 geschlossen: schaltet zum festkaschierten 47pF
einen 5,6pF

J6 geschlossen: schaltet zum festkaschierten 47pF
einen 10pF

J7 geschlossen: schaltet zum festkaschierten 47pF
einen 10pF

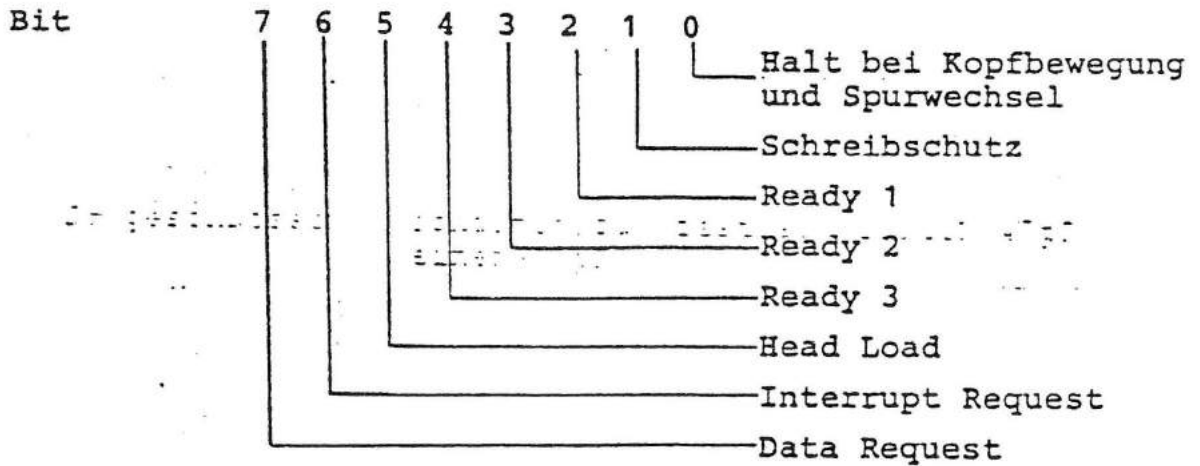
ITT 3030

AUF WACHSTUM PROGRAMMIERT I

3. Adreßbelegung

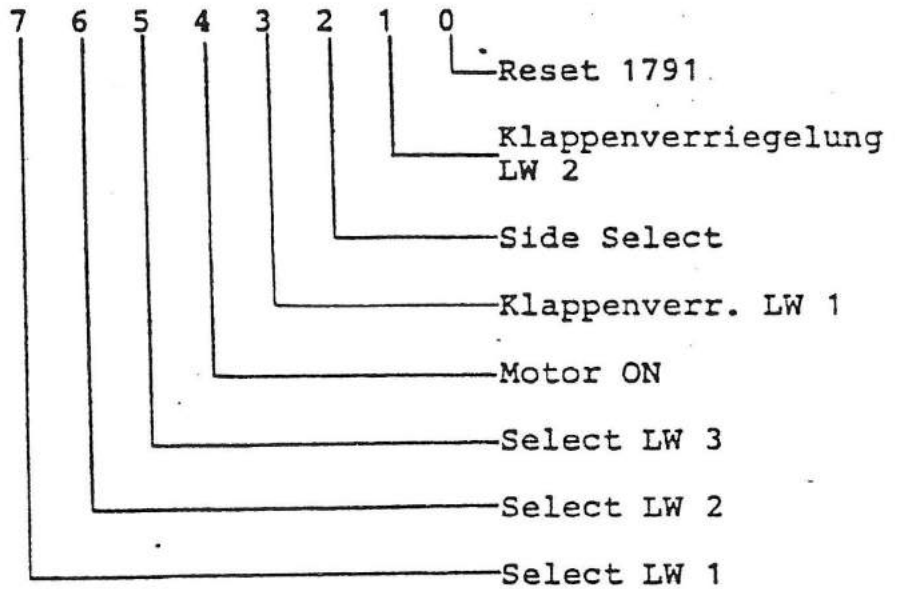
Adreßbit	7	6	5	4	3	2	1	0	
	Kanaladr.5					0	0	0	} CS - 1791
						0	0	1	
						0	1	0	
						0	1	1	
						1	0	0	CS-Statusregister CS-Befehlsregister

IN 54_H



OUT 54_H

Bit



4. Daten-Trenn-Stufe

Diese besteht aus zwei Funktionsgruppen:

- dem Phasenregelkreis, mit dem die Referenzfrequenz für Öffnungszeitpunkt und -dauer der Daten- und Taktfenster der Bitfolgefrequenz des Lesedatenstromes nachgeführt wird.
- der Missing-Clock-Detect-Logic, die bei 4 aufeinanderfolgenden Fehltakten die Zuordnung zwischen Datenfenster und Taktfenster umdreht.

5. Head-Load/Step-Logic

Wenn der Formatter/Controller FD 1791 den Befehl zum Neupositionieren und Aufsetzen des Kopfes ausgibt, müssen die mechanischen Reaktionszeiten berücksichtigt werden.

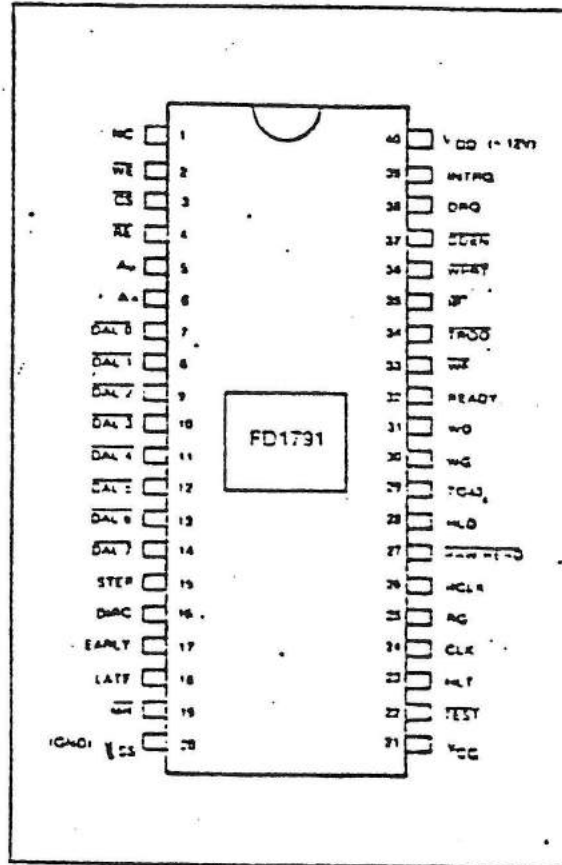
Mit Hilfe monostabiler Kippstufen werden für einen Spurwechsel 40 msec und für das Kopfaufsetzen 60 msec-Intervalle erzeugt.

Der FD 1791 kann den Ablauf dieser Zeiten über den HLT-Eingang abfragen.

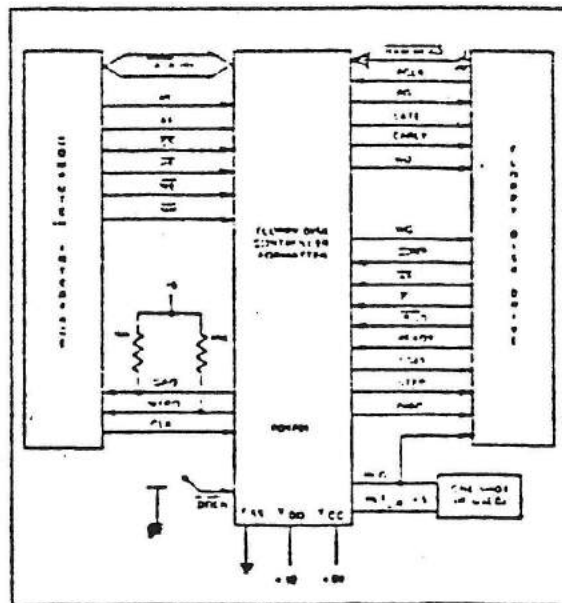
6. Taktgenerator

Die Grundfrequenz eines 4 MHz-Oszillators wird durch 4 geteilt und als 1 MHz-Takt dem FD 1791 zugeführt.

(Mini-Floppy = 1 MHz, Floppy-Disk = 2 MHz)



PIN CONNECTIONS

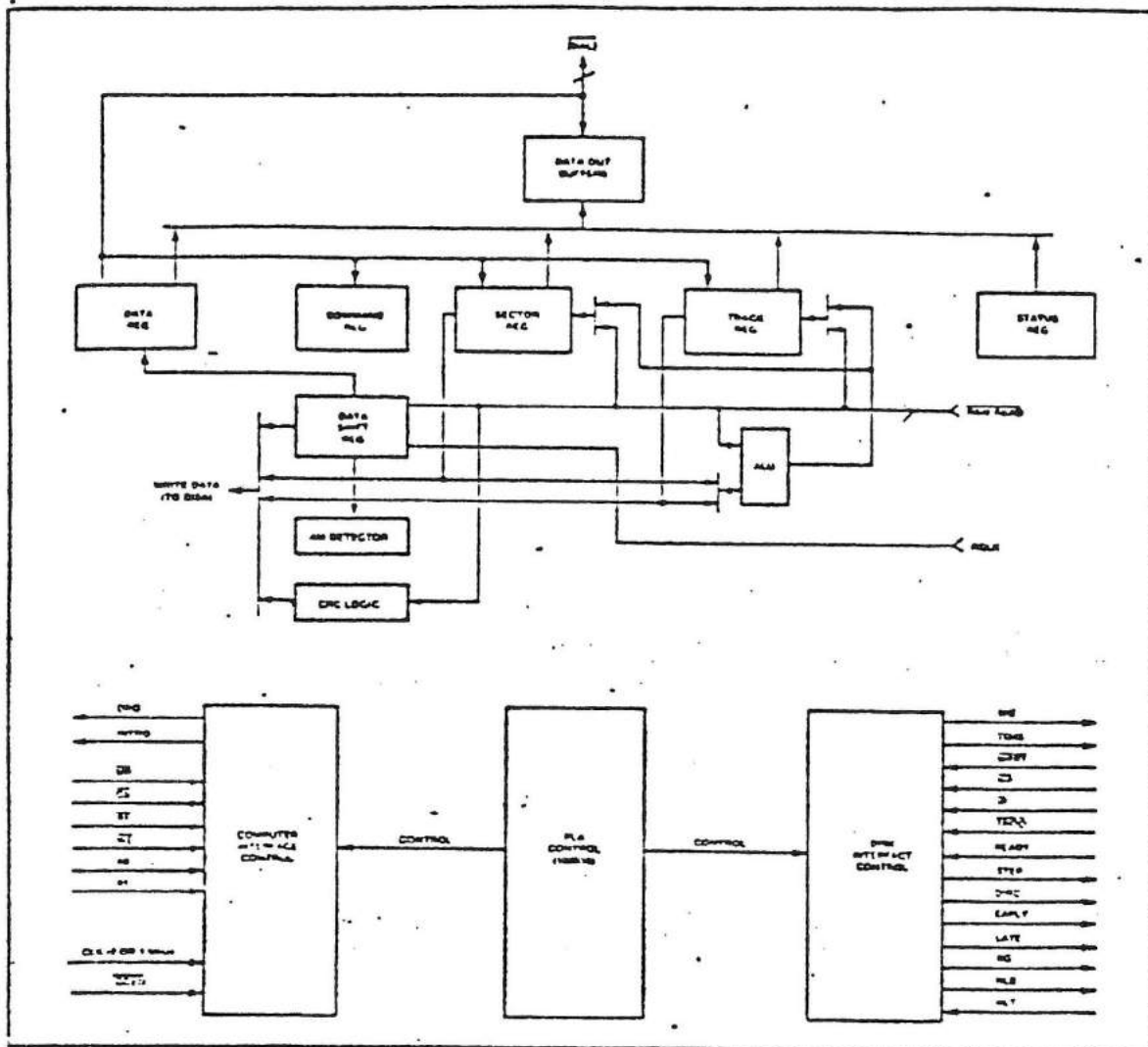


1791 SYSTEM BLOCK DIAGRAM

PIN OUTS

PIN NUMBER	PIN NAME	SYMBOL	FUNCTION																				
20	POWER SUPPLIES	V _{ss}	Ground																				
21		V _{cc}	+5V																				
40		V _{bb}	+12V																				
19		MR	A logic low on this input resets the device and loads nex CS into the command register. The Not Ready (Status Bit 7) is reset during MR ACTIVE. When MR is brought to a logic high a Restore Command is executed, regardless of the state of the Ready signal from the drive. Also, hex 01 is loaded into sector register.																				
COMPUTER INTERFACE:																							
7-14	DATA ACCESS LINES	DAL0-DAL7	Eight bit inverted Bidirectional bus used for transfer of data, control, and status. This bus is a receiver enabled by WE or a transmitter enabled by RE.																				
3	CHIP SELECT	CS	A logic low on this input selects the chip and enables computer communication with the device.																				
5,6	REGISTER SELECT LINES	A0, A1	These inputs select the register to receive/transfer data on the DAL lines under RE and WE control: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>A1</th> <th>A0</th> <th>RE</th> <th>WE</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Status Reg</td> <td>Command Reg</td> </tr> <tr> <td>0</td> <td>1</td> <td>Track Reg</td> <td>Track Reg</td> </tr> <tr> <td>1</td> <td>0</td> <td>Sector Reg</td> <td>Sector Reg</td> </tr> <tr> <td>1</td> <td>1</td> <td>Data Reg</td> <td>Data Reg</td> </tr> </tbody> </table>	A1	A0	RE	WE	0	0	Status Reg	Command Reg	0	1	Track Reg	Track Reg	1	0	Sector Reg	Sector Reg	1	1	Data Reg	Data Reg
A1	A0	RE	WE																				
0	0	Status Reg	Command Reg																				
0	1	Track Reg	Track Reg																				
1	0	Sector Reg	Sector Reg																				
1	1	Data Reg	Data Reg																				
4	READ ENABLE	RE	A logic low on this input controls the placement of data from a selected register on the DAL when CS is low.																				
2	WRITE ENABLE	WE	A logic low on this input gates data on the DAL into the selected register when CS is low.																				
38	DATA REQUEST	DRQ	This open drain output indicates that the DR contains assembled data in Read operations, or the DR is empty in Write operations. This signal is reset when serviced by the computer through reading or loading the DR in Read or Write operation, respectively. Use 10K pull-up resistor to +5.																				
39	INTERRUPT REQUEST	INTRQ	This open drain output is set at the completion or termination of any operation and is reset when a new command is loaded into the command register or the status register is read. Use 10K pull-up resistor to +5.																				
24	CLOCK	CLK	This input requires a free-running square wave clock for internal timing reference, 2 MHz for regular drives, 1 MHz for mini-drives.																				
FLOPPY DISK INTERFACE:																							
25	READ GATE	RG	A high level on this output indicates to the data separator circuitry that a field of zeros (or ones) has been encountered, and is used for synchronization.																				
31	WRITE DATA	WD	A 250 ns (MFM) or 500 ns (FM) pulse per flux transition. WD contains the unique Address marks as well as data in both FM and MFM formats.																				
26	READ CLOCK	RCLK	A nominal square-wave clock signal derived from the data stream must be provided to this input. Phasing (i.e. RCLK transitions) relative to RAW READ is important but polarity (RCLK high or low) is not.																				
27	RAW READ	RAW READ	The data input signal directly from the drive. This input shall be a negative pulse for each recorded flux transition.																				
25	HEAD LOAD	HLD	The HLD output controls the loading of the Read-Write head against the media. When a logic high is found on the HLT input the head is assumed to be engaged.																				
23	HEAD LOAD TIMING	HLT																					
15	STEP	STEP	Step and direction motor control. The step output contains a pulse for each step and the direction output is active high when stepping in, active low when stepping out.																				
16	DIRECTION	DIRC																					

17	EARLY	EARLY	Indicates that the write data pulse occurring while Early is active (high) should be shifted early for write precompensation.
18	LATE	LATE	Indicates that the write data pulse occurring while Late is active (high) should be shifted late for write precompensation.
29	TRACK GREATER THAN 43	TG43	This output informs the drive that the Read-Write head is positioned between the 44-75. This output is valid only during Read and Write Commands.
30	WRITE GATE	WG	This output is made valid when writing is to be performed on the diskette.
32	READY	READY	This input indicates disk readiness and is sampled for a logic high before Read or Write commands are performed. If Ready is low the Read or Write operation is not performed and an interrupt is generated. A Seek operation is performed regardless of the state of Ready. The Ready input appears in inverted format as Status Register bit 7.
33	$\overline{\text{WRITE FAULT}}$	$\overline{\text{WF}}$	This input detects writing faults indications from the drive. When WG = 1 and $\overline{\text{WF}}$ goes low the current Write command is terminated and the Write Fault status bit is set. The $\overline{\text{WF}}$ input should be made inactive (high) when WG becomes inactive.
34	$\overline{\text{TRACK 00}}$	$\overline{\text{TR00}}$	This input informs the FD1791 that the Read-Write head is positioned over Track 00 when a logic low.
35	$\overline{\text{INDEX PULSE}}$	$\overline{\text{IP}}$	Input, when low for a minimum of 10 μsac , informs the FD1791 when an index mark is encountered on the diskette.
36	$\overline{\text{WRITE PROTECT}}$	$\overline{\text{WPRT}}$	This input is sampled whenever a Write Command is received. A logic low terminated the command and sets the Write Protect Status bit.
37	$\overline{\text{DOUBLE DENSITY}}$	$\overline{\text{DDEN}}$	This pin selects either single or double density operation. When $\overline{\text{DDEN}} = 0$, double density is selected. When $\overline{\text{DDEN}} = 1$, single density is selected.
22	$\overline{\text{TEST}}$	$\overline{\text{TEST}}$	This input is used for testing purposes only and should be tied to +5V or left open by the user unless interfacing to voice actuated motors.



FD1791 BLOCK DIAGRAM

PROCESSOR INTERFACE

The interface to the processor is accomplished through the eight Data Access Lines (DAL) and associated control signals. The DAL are used to transfer Data, Status, and Control words out of, or into the FD1791. The DAL are three state buffers that are enabled as output drivers when Chip Select (CS) and Read Enable (RE) are active (low logic state) or act as input receivers when CS and Write Enable (WE) are active.

When transfer of data with the Floppy Disk Controller is required by the host processor, the device address is decoded and CS is made low. The least-significant address bits A1 and A0, combined with the signals RE during a Read operation or WE during a Write operation are interpreted as selecting the following registers:

A1-A0	READ (RE)	WRITE (WE)
0 0	Status Register	Command Register
0 1	Track Register	Track Register
1 0	Sector Register	Sector Register
1 1	Data Register	Data Register

During Direct Memory Access (DMA) types of data transfers between the Data Register of the FD1791 and the processor, the Data Request (DRQ) output is used in Data Transfer control. This signal also appears as status bit 1 during Read and Write operations.

On Disk Read operations the Data Request is activated (set high) when an assembled serial input byte is transferred in parallel to the Data Register. This bit is cleared when the Data Register is read by

the processor. If the Data Register is read after one or more characters are lost, by having new data transferred into the register prior to processor readout, the Lost Data bit is set in the Status Register. The Read operation continues until the end of sector is reached.

On Disk Write operations the Data Request is activated when the Data Register transfers its contents to the Data Shift Register, and requires a new data byte. It is reset when the Data Register is loaded with new data by the processor. If new data is not loaded at the time the next serial byte is required by the Floppy Disk, a byte of zeroes is written on the diskette and the Lost Data bit is set in the Status Register.

At the completion of every command an INTRQ is generated. INTRQ is reset by either reading the status register or by loading the command register with a new command. In addition, INTRQ is generated if a Force Interrupt command condition is met.

FLOPPY DISK INTERFACE

The 1791 has two modes of operation according to the state of DDEN (Pin 37). When DDEN = 1, single density is selected. In either case, the CLK input (Pin 24) is at 2 MHz. However, when interfacing with the mini-floppy, the CLK input is set at 1 MHz for both single density and double density. When the clock is at 2 MHz, the stepping rates of 3, 6, 10, and 15 ms are obtainable. When CLK equals 1 MHz these times are doubled.

HEAD POSITIONING

Four commands cause positioning of the Read-Write head (see Command Section). The period of each positioning step is specified by the *r* field in bits 1 and 0 of the command word. After the last directional step an additional 15 milliseconds of head settling time takes place if the Verify flag is set in Type I commands. Note that this time doubles to 30 ms for a 1 MHz clock. If $\overline{\text{TEST}} = 0$, there is zero settling time. There is also a 15 ms head settling time if the *E* flag is set in any Type 2 or 3 command.

The rates (shown in Table 1) can be applied to a Step-Direction Motor through the device interface.

Step—A 2 μs (MFM) or 4 μs (FM) pulse is provided as an output to the drive. For every step pulse issued, the drive moves one track location in a direction determined by the direction output.

Direction (DIRC)—The Direction signal is active high when stepping in and low when stepping out. The Direction signal is valid 12 μs before the first stepping pulse is generated.

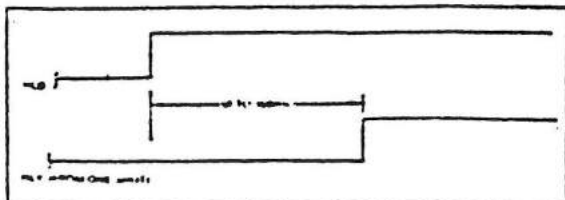
When a Seek, Step or Restore command is executed an optional verification of Read-Write head position can be performed by setting bit 2 ($V = 1$) in the command word to a logic 1. The verification operation begins at the end of the 15 millisecond settling time after the head is loaded against the media. The track number from the first encountered ID Field is compared against the contents of the Track Register. If the track numbers compare and the ID Field Cyclic Redundancy Check (CRC) is correct, the verify operation is complete and an INTRQ is generated with no errors. The FD1791 must find an ID field with correct track number and correct CRC within 5 revolutions of the media; otherwise the seek error is set and an INTRQ is generated.

Table 1. STEPPING RATES

CLK	2 MHz	2 MHz	1 MHz	1 MHz	2 MHz	1 MHz
DDEN	0	1	0	1		
R1 R0	TEST=1	TEST=1	TEST=1	TEST=1	TEST=1	TEST=0
0 0	3 ms	3 ms	6 ms	6 ms	Approx.	Approx.
0 1	6 ms	6 ms	12 ms	12 ms	200 μs	400 μs
1 0	10 ms	10 ms	20 ms	20 ms		
1 1	15 ms	15 ms	30 ms	30 ms		

The Head Load (HLD) output controls the movement of the read/write head against the media. HLD is activated at the beginning of a Type I command if the *h* flag is set ($h = 1$), at the end of the Type I command if the verify flag ($V = 1$), or upon receipt of any Type II or III command. Once HLD is active it remains active until either a Type I command is received with ($h = 0$ and $V = 0$); or if the FD1791 is in an idle state (non-busy) and 15 index pulses have occurred, it is reset.

Head Load Timing (HLT) is an input to the FD1791 which is used for the head engage time. When $\text{HLT} = 1$, the FD1791 assumes the head is completely engaged. The head engage time is typically 30 to 100 ms depending on drive. The low to high transition on HLD is typically used to fire a one shot. The output of the one shot is then used for HLT and supplied as an input to the FD1791.



HEAD LOAD TIMING

When both HLD and HLT are true, the FD1791 will then read from or write to the media. The "and" of HLD and HLT appears as a status bit in Type I status.

In summary for the Type I commands: if $h = 0$ and $V = 0$, HLD is reset. If $h = 1$ and $V = 0$, HLD is set at the beginning of the command and HLT is not sampled nor is there an internal 15 ms delay. If $h = 0$ and $V = 1$, HLD is set near the end of the command, an internal 15 ms occurs, and the FD1791 waits for HLT to be

true. If $h = 1$ and $V = 1$, HLD is set at the beginning of the command. Near the end of the command, after all the steps have been issued, an internal 15 ms delay occurs and the FD1791 then waits for HLT to occur.

For Type II and III commands with *E* flag off, HLD is made active and HLT is sampled until true, with *E* flag on HLD is made active, an internal 15 ms delay occurs and then HLT is sampled until true.

DISK READ OPERATIONS

Sector lengths of 128, 256, 512 or 1024 are obtainable in either FM or MFM formats. For FM, DDEN should be placed to logical "1." For MFM formats, DDEN should be placed to a logical "0." Sector lengths are determined at format time by a special byte in the "ID" field. If this Sector Length byte in the ID field is zero, then the sector length is 128 bytes. If 01 then 256 bytes. If 02, then 512 bytes. If 03, then the sector length is 1024 bytes. The number of sectors per track as far as the FD1791 is concerned can be from 1 to 255 sectors. The number of tracks as far as the FD1791 is concerned is from 0 to 255 tracks. For IBM 3740 compatibility, sector lengths are 128 bytes with 26 sectors per track. For System 34 compatibility (MFM), sector lengths are 256 bytes/sector with 26 sectors/track; or lengths of 1024 bytes/sector with 6 sectors/track.

For read operations, the FD1791 requires RAW READ Data (Pin 27) signal which is a 250 ns pulse per flux transition and a Read clock (RCLK) signal to indicate flux transition spacings. The RCLK (Pin 26) signal is provided by some drives but if not it may be derived externally by Phase lock loop, one shots, or counter techniques. In addition, a Read Gate Signal is provided as an output (Pin 25) which informs some phase lock loops when to acquire synchronization. When reading from the media in FM, RG is made true when 2 bytes of zeroes are detected. The FD1791 must find an address mark within the next 10 bytes; otherwise RG is reset and the search for 2 bytes of zeroes begins all over again. If an address mark is found within 10 bytes, RG remains true as long as the FD1791 is deriving any useful information from the data stream. Similarly for MFM, RG is made active true when 4 bytes of "00" or "FF" are detected. The FD1791 must find an address mark within the next 16 bytes, otherwise RG is reset and search resumes.

DISK WRITE OPERATION

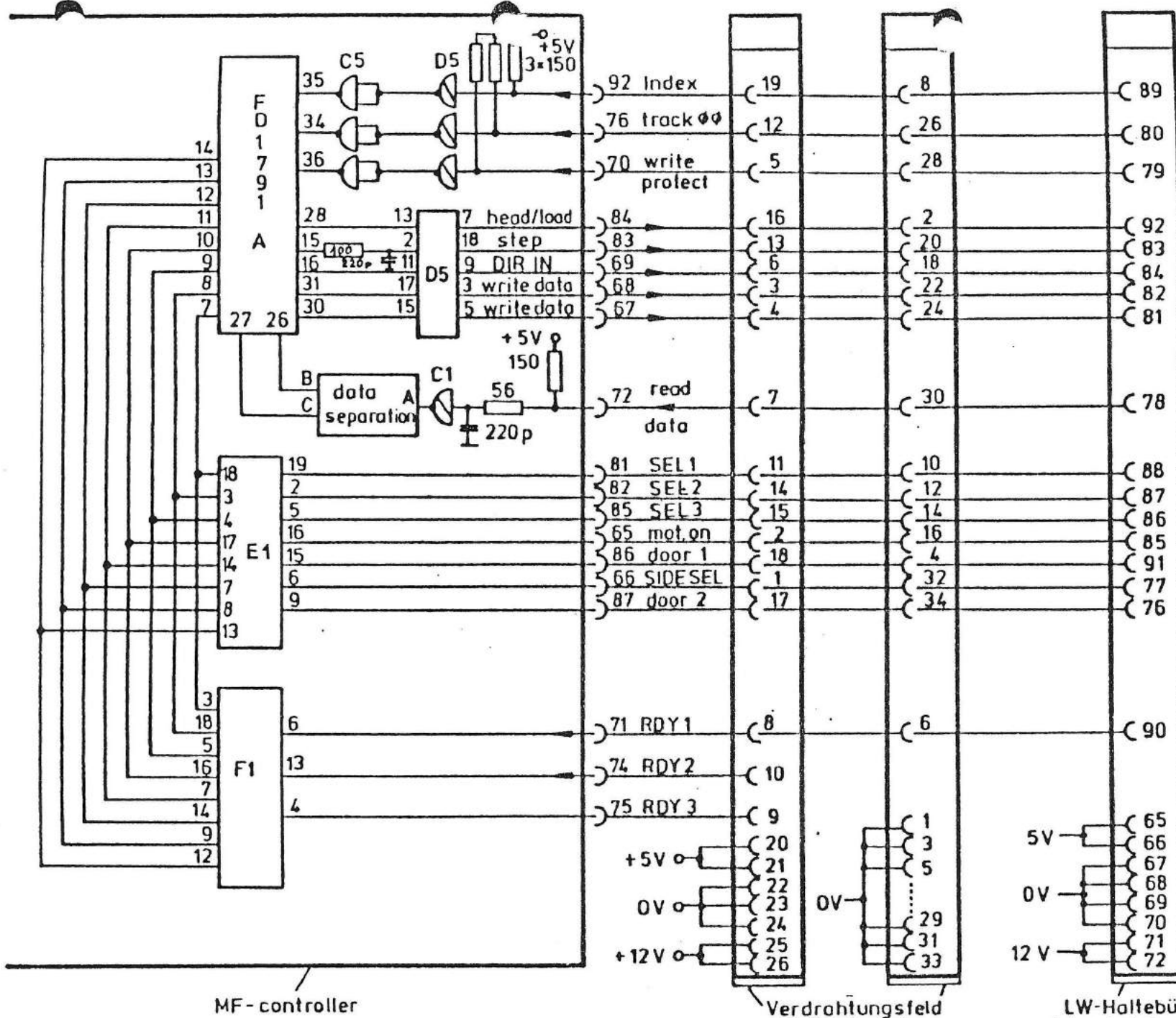
When writing is to take place on the diskette the Write Gate (WG) output is activated, allowing current to flow into the Read/Write head. As a precaution to erroneous writing the first data byte must be loaded into the Data Register in response to a Data Request from the FD1791; before the Write Gate signal can be activated.

Writing is inhibited when the Write Protect input is a logic low, in which case any Write command is immediately terminated, an interrupt is generated and the Write Protect status bit is set. The Write Fault input, when activated, signifies a writing fault condition detected in disk drive electronics such as failure to detect write current flow when the Write Gate is activated. On detection of this fault the FD1791 terminates the current command, and sets the Write Fault bit (bit 5) in the Status Word. The Write Fault input should be made inactive when the Write Gate output becomes inactive.

For write operation, the FD1791 provides Write Gate (Pin 30) and Write Data (Pin 31) outputs. Write data consists of a series of 500 ns pulses in FM ($\overline{DDEN} = 1$) and 250 ns pulses in MFM ($\overline{DDEN} = 0$). Write Data provides the unique address marks in both formats.

Also during write, two additional signals are provided for write precompensation. These are EARLY (Pin 17) and LATE (Pin 18). EARLY is active true when the WD pulse appearing on (Pin 30) is to be written early. EARLY is valid for the duration of the pulse. LATE is active true when the WD pulse is to be written late. If both are low when a WD pulse is present, the WD pulse is to be written at nominal. Since write precompensation values vary from disk manufacturer to disk manufacturer, the actual value is determined by several one shots or delay lines which are located external to the FD1791. The write precompensation signals EARLY and LATE are valid in both FM and MFM formats.

Whenever a Read or Write command (Type II or III) is received the FD1791 samples the Ready input. If this input is logic low the command is not executed and an interrupt is generated. The Seek or Step Type I commands are performed regardless of the state of the Ready input. Also, whenever a Type II or III command is received, the TG43 signal output is updated.



← Anschluß des 3. Laufwerks connection of the 3. drive

- C1 = 74 LS 14
- C5 = 74 LS 00
- D5 = 74 LS 240
- E1 = 74 LS 374
- F1 = 74 LS 241

MF-controller

Verdrahtungsfeld motherboard

LW-Haltebügel drive-holding device

Buchse X plug connection X

Buchse I/II plug connection I/II

Buchse III plug connection III

```

*****
*
*  BEDIENUNGSANLEITUNG  *
*
*  I E E E - 4 8 8    *
*
*      INTERFACE      *
*
*****

```

1. Lieferumfang

Dem IEEE - 488 Interface liegen die folgenden Komponenten bei:

- IEEE - 488 Interface
- + Steckverbindung

2. Installation des IEEE - 488 Interfaces

Neuinstallation: Die Installation des ITT 3030 Computer-Systems entnehmen Sie bitte Ihrer

" BEDIENUNGSANLEITUNG ITT 3030 "

Anmerkung: Die Hinweise auf die ABBILDUNGEN beziehen sich auf die

" BEDIENUNGSANLEITUNG ITT 3030 "

Beachten Sie bitte zur Ihrer Erleichterung folgenden Punkt:

- Nach dem Einsatz des Monitor - Adapters auf dem Verdrahtungsfeld Abb. 2 (8) setzen Sie Ihr IEEE - 488 Interface nach der nachfolgenden Beschreibung ab PUNKT 2.6 auf.

Systemerweiterung: Bei einer Systemerweiterung bzw. Neuinstallation gehen Sie bitte wie folgt vor.

2.1 NETZ - SCHALTER am ITT 3030 ausschalten. Abb.9(1)

2.2 GEHAUSE des ITT 3030 öffnen. Abb.1

2.3 FLOPPY - DISK LAUFWERKE herausnehmen. Abb.10

2.4 VERRIEGELUNGS - BÜGEL des Floppy - Haltebügels öffnen. Abb.3(6)

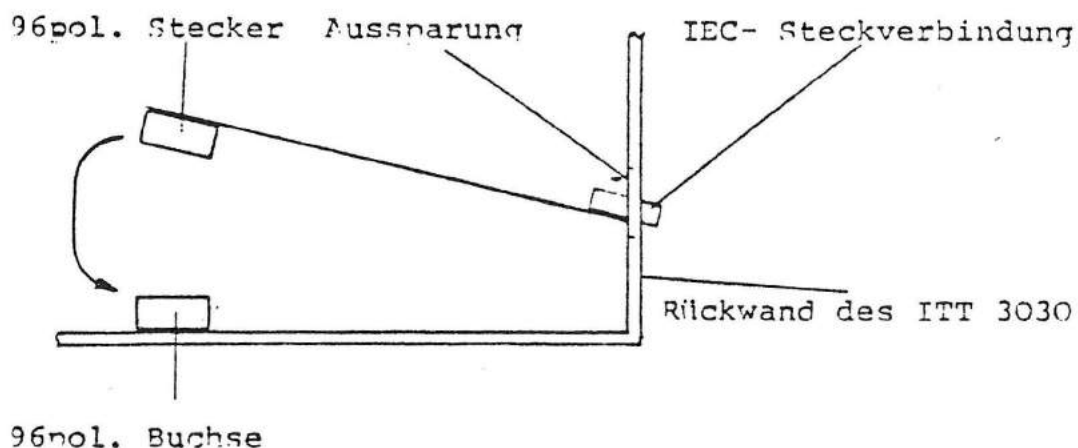
2.5 FLOPPY - HALTEBÜGEL abnehmen.

>>>>

2.6 ABDECK - FLÄTTCHEN in der Rückwand ,das diesem Steckplatz zugeordnet ist ,entfernen.

2.7 IEEE - 488 INTERFACE auf einen der 96pol. Steckverbindungen an den Erweiterungs - Slots aufstecken.

2.8



2.9 FLOPPY - HALTEBÜGEL einsetzen und die beiden Floppy - Disk Laufwerke einstecken. Abb.10

3. Inbetriebnahme: Bevor Sie einen Dialog mit dem IEEE - 488 Interface durchführen können, legen Sie bitte eine ADRESSE für den IEC- Bus mit den Brücken ..1... bis ..5... fest.

Mit der Brücke ..6... ist Ihr ITT 3030 ein Controller oder ein Slave.

Nach dem NETZ einschalten und dem Laden eines Betriebssystems für den ITT 3030 ist Ihr Interface betriebsbereit.

Kontroll-Bus
=====

- PIN
- Jeder der fünf Leitungen des Kontroll-Busses hat eine besondere Funktion:
- | | | |
|----|-----|---|
| 19 | ATN | Attention; wird vom Kontroller aktiviert. Mit dieser Leitung werden der Daten- und Befehlsmode unterschieden. |
| 21 | IFC | Interface clear; wird vom Kontroller aktiviert. Diese Leitung wird benutzt (IFC = logisch 1), um das Interface der Geräte in einen spezifischen Anfangszustand zu setzen. |
| 23 | REN | Remote enable; wird vom Kontroller aktiviert. Mit dieser Leitung können die angeschlossenen Geräte vom Lokalbetrieb in den Fernbetrieb (REN = logisch 1) umgeschaltet werden. |
| 17 | SRQ | Service request; wird von irgendeinem Gerät aktiviert. Wenn diese Leitung logisch 1 ist, weiß der Kontroller, daß ein Gerät ihm etwas mitteilen möchte (Seriell-Polling). Der Kontroller kann dann das laufende Programm unterbrechen und die Nachricht des Gerätes abholen. |
| 25 | EOI | End or identify; diese Leitung hat zwei Funktionen. Im Datenmode (ATN = logisch 0) kann ein Talker EOI als Datenende-Kriterium verwenden (EOI = logisch 0). Im Befehlsmode (ATN = logisch 1) benutzt der Kontroller die EOI-Leitung für die Ausführung des Parallel-Pollings. |

Handshake-Bus =====

Die Übertragung von Informationen auf dem Datenbus wird durch Aktivitäten auf dem Handshake-Bus begleitet. Es gibt drei Handshake-Leitungen:

- 27 DAV Data valid; wird vom Talker aktiviert. DAV = logisch 1, bedeutet, daß die Nachricht auf dem Bus korrekt und zur Übernahme geeignet ist.

- 19 NRFD Not ready for data; wird von den Listnern aktiviert. NRFD = logisch 0, bedeutet, daß alle Listener bereit sind, die Nachricht zu übernehmen.

- 31 NDAC Not data accepted; wird von den Listnern aktiviert. NDAC = logisch 0, bedeutet daß alle Listener die Nachricht akzeptiert haben.

Kleines IEC-Bus-Lexikon =====

Adressierung

Alle Geräte werden durch den Kontroller mittels der Adressierung in die betreffende Funktion geschaltet (Talker oder Listener). Jedes der möglichen 15 Geräte hat eine hardwaremäßig einstellbare Adresse. Für die Adressierung wird der ISO-7-Bit-Code verwendet. Die bei der Adressierung vom Kontroller gesendeten ASCII-Zeichen werden über die 8 Datenleitungen übertragen. Damit Daten von Adressen unterschieden werden können, wird bei der Adressierung zusätzlich die Steuerleitung ATN aktiviert. Ein Gerät, das als Talker und Listener arbeiten kann, hat für jede Funktion eine eigene Adresse, die sich jedoch nur in Bit 6 und 7 unterscheiden. Ist das Bit 7 gesetzt, arbeitet das Gerät als Listener. Bit 1 bis 5 werden am Gerät eingestellt.

ATN (attention)

Das Aktivieren der Steuerleitung ATN bedeutet, daß auf der Datenleitung Adressen oder Befehle übertragen werden.

Bus-Kommandos

Nicht alle Geräte reagieren auf alle Bus-Kommandos. Meistens sind in den Datenblättern die Kommandos angegeben, die das Gerät ausführen kann. Man unterscheidet Universal-Kommandos, adressierte Kommandos und Entadressierte Kommandos. Kommandos werden auf den Datenleitungen übertragen, nachdem die ATN-Leitung aktiviert wurde.

Bus-Länge

Das Impedanzverhalten der Verbindungskabel läßt unter Berücksichtigung der Signalzeiten eine maximale Buslänge von 20 m zu.

- Pro Gerät darf jedoch eine Kabellänge von 2 m nicht überschritten werden.
- Datenübertragung Die Übermittlung von Daten ist nicht genormt. Das jeweilige Ende einer Zeichenkette wird durch die Steuerleitung EOI angezeigt. Ausserdem hat es sich in der allgemeinen Datenübertragung eingebürgert, daß ein CR- (carriage return) und ein LF- (line feed) -ASCII-Zeichen eine Übertragung abschließen.
- DAV (data valid) Diese Handshake-Leitung wird vom Talker aktiviert, wenn die zu übertragenden Meßdaten richtig sind. Dieses Kommando ist eine Art Freigabe.
- DCL (device clear) Dieses Kommando bringt alle Geräte in den Einschaltzustand.
- EOI (end or identify) Diese Kontrolleitung wird benutzt, um das Ende einer Übertragung anzuzeigen oder bei der Identifizierung eines Gerätes.
- Fan-Out Die Ausgangstreiber der Busleitungen haben ein Fan-Out von 30. Damit werden die Threshold-Pegel (Schwellwerte) auch bei voller Gerätezahl erreicht.
- GET (group execute trigger) Dieses Kommando löst eine Messung bei allen vorprogrammierten Geräten aus.
- GPIB (General Purpose Interface Bus) Synonym für IEC-Bus und IEEE-488-Bus.
- GTL (go to local) Dieses Kommando bringt adressierte Geräte in die Handbedienung zurück.

Handshake-Bus	Drei Leitungen (DAV,NRFD, NDAC), mit denen die Kommunikation unterschiedlich schneller Geräte koordiniert wird.
Handshake-Timing	Jede Datenübertragung von einem Gerät zu anderen Geräten wird nicht eher begonnen, bis alle zuhörenden Geräte ihr Data Ready (NRFD) gesendet haben und nicht eher abgeschlossen, bis alle zuhörenden Geräte ihr Data Acceptes (NDAC) gesendet haben.
IEC-Bus	Bus-System mit 16 parallelen Leitungen, das einen Befehls- und Datenaustausch zwischen den beteiligten Geräten gestattet. Es basiert auf einem schon 1965 ausgearbeiteten Interface-System für programmierbare Meßgeräte von Hewlett-Packard. Bei dem Entwurf der IEC (Internationale Elektrotechnische Commission) wurde ein anderer Stecker verwendet als bei dem amerikanischen Entwurf (IEEE-488).
IEEE-488-Bus	Amerikanischer Entwurf eines Interface-Systems für programmierbare Meßgeräte. Er unterscheidet sich vom IEC-Bus nur durch den Stecker.
IFC (interface clear)	Mit dieser Kontrolleitung wird die Adressierung der Geräte gelöscht und der Bus in einen Beginnzustand gebracht.
Kontroll-Bus	Die fünf Leitungen (IFC,ATN,SRQ, REN,EOI) ermöglichen verschiedene Kommandos an die angeschlossenen Geräte.

ITT 3030

Blatt 4

AUF WACHSTUM PROGRAMMIERT !

- Kontroller** Auf dem IEC-Bus kann nur ein Kontroller (ITT 3030) tätig sein. Er hat Zugriff zum Kontroll-Bus. Die Leitungen ATN, IFC und REN können nur vom Kontroller aktiviert werden. Außerdem kann er Informationen von den angeschlossenen Geräten empfangen.
- Listener** Der Listener (Empfänger) nimmt Informationen entgegen. Es können mehrere Listener gleichzeitig auf dem IEC-Bus tätig sein.
- LLO (local lock out)** Dieses Kommando setzt alle Handbedienelemente des Gerätes außer Betrieb, auch "Reset".
- NDAC (not data accepted)** Die Listener aktivieren diese Handshake-Leitung, wenn sie die angebotene Information akzeptiert haben (High-Signal).
- Negativ-True-Logik** Der IEC-Bus arbeitet mit TTL-Pegeln in der Negativ-True-Logik (0... 1,4 V = logisch 1; 2,5 V ... 5 V = logisch 0).
- NRFD (not ready for data)** Die Listener geben über diese Handshake-Leitung ein Bereitschaftssignal, daß sie zur Übernahme von Informationen bereit sind (High-Signal). Erst wenn alle adressierten Listener dieses Signal gegeben haben, beginnt die Übertragung von Daten.
- Parallel-Polling** Die parallele Abfrage des Status erfordert bei bis zu acht Geräten nur eine einzige Statusabfrage. Dann ist jedem Gerät ein Bit des Statusbytes zugeordnet (vorher per Programm zugewiesen).
- Polling** Verfahren, bei dem der Kontroller fragt, welches Gerät die Steuerleitung SRQ aktiviert hat. Man unterscheidet serielles und paralleles "polling".

ITT 3030

Blatt 5

AUF WACHSTUM PROGRAMMIERT!

PPC (parallel poll configure)	Dieses Kommando bestimmt, welches Bit ein Gerät beim parallel "polling" aktivieren soll.
PPU (parallel poll unconfigure)	Dieses Kommando veranlaßt alle Geräte, das vorher bestimmte Bitmuster zu löschen.
REN (remote enable)	Diese Kontrolleitung ermöglicht die Fernbedienung.
Seriell-Polling	Alle Geräte werden nacheinander vom Kontroller nach dem Statusbyte abgefragt. Ein bestimmtes oder auch mehrer Bits (vom Hersteller vorgegeben) geben dann über den Gerätestatus Auskunft, z.B. ob die SRQ-Leitung aktiviert wurde.
SPC (selective device clear)	Dieses Kommando bringt das adressierte Gerät in den Einschaltzustand.
SPD (serial poll disable)	Dieses Kommando löscht die Bedingungen für Statusabfragen.
SPE (serial poll enable)	Dieses Kommando setzt alle Bedingungen für Statusabfragen.
SRQ (service request)	Diese Kontrolleitung ermöglicht eine Bedienungsanforderung eines Gerätes, z.B. an den Kontroller während eines Programmablaufes. Der Kontroller hat dann durch eine Statusabfrage (polling) herauszufinden, welches Gerät die Leitung aktiviert hat.
Talker	Der Talker (Sender) kann Informationen auf den IEC-Bus geben. Es kann immer nur ein Talker auf dem Bus tätig sein.

ITT 3030

Blatt 6

AUF WACHSTUM PROGRAMMIERT!

TCT (take control)

Dieses Kommando übergibt die Kontrolle vom zur Zeit aktiven Controller an das adressierte Gerät.

Übertragungsgeschwindigkeit

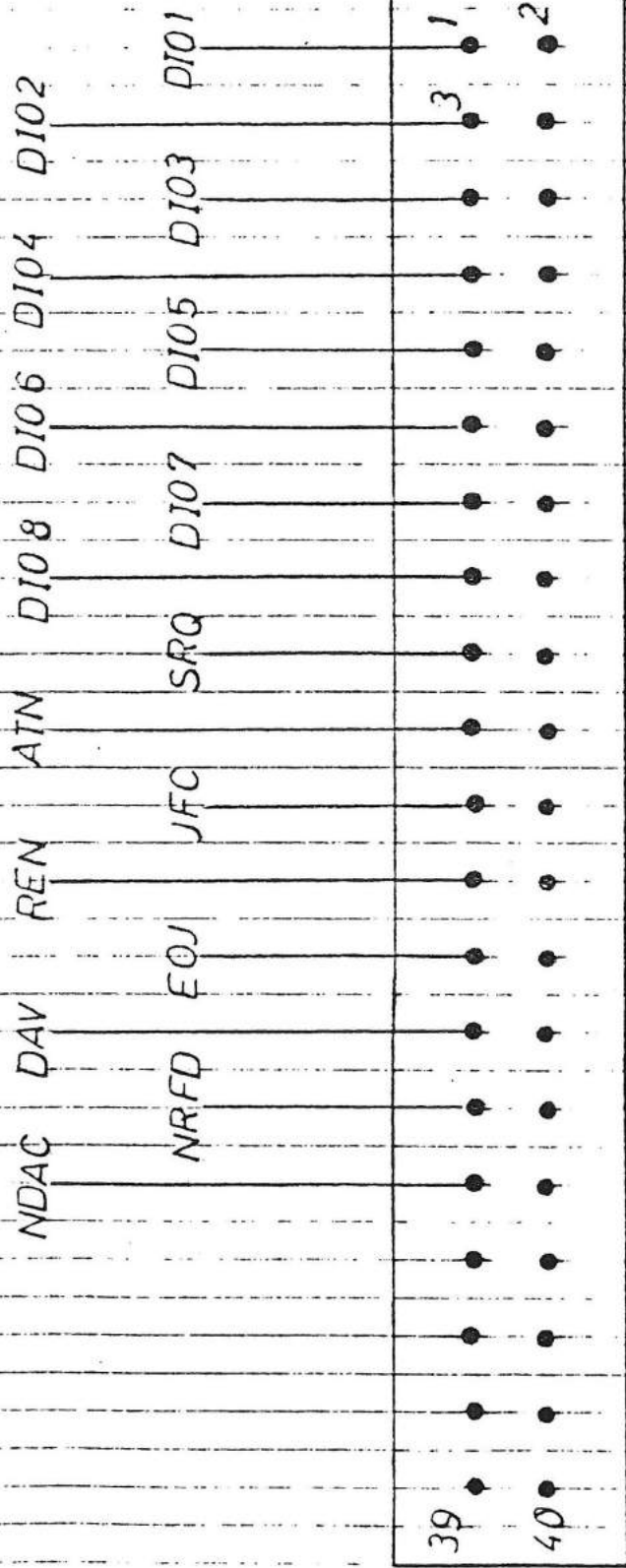
Die Übertragungsgeschwindigkeit beträgt auf dem IEC-Bus 250 KByte/s, wenn alle 2 m Einheitslasten angebracht sind und 48-mA-Treiber mit offenem Kollektor verwendet werden. Die maximale Übertragungsgeschwindigkeit von 1 MByte/s wird bei einer Kabellänge von 0.5 m pro Gerät erreicht und wenn 48-mA-Tri-State-Treiber benutzt werden.

UNL (unlisten)

Kommando, mit dem alle Listener gelöscht werden.

UNT (untalk)

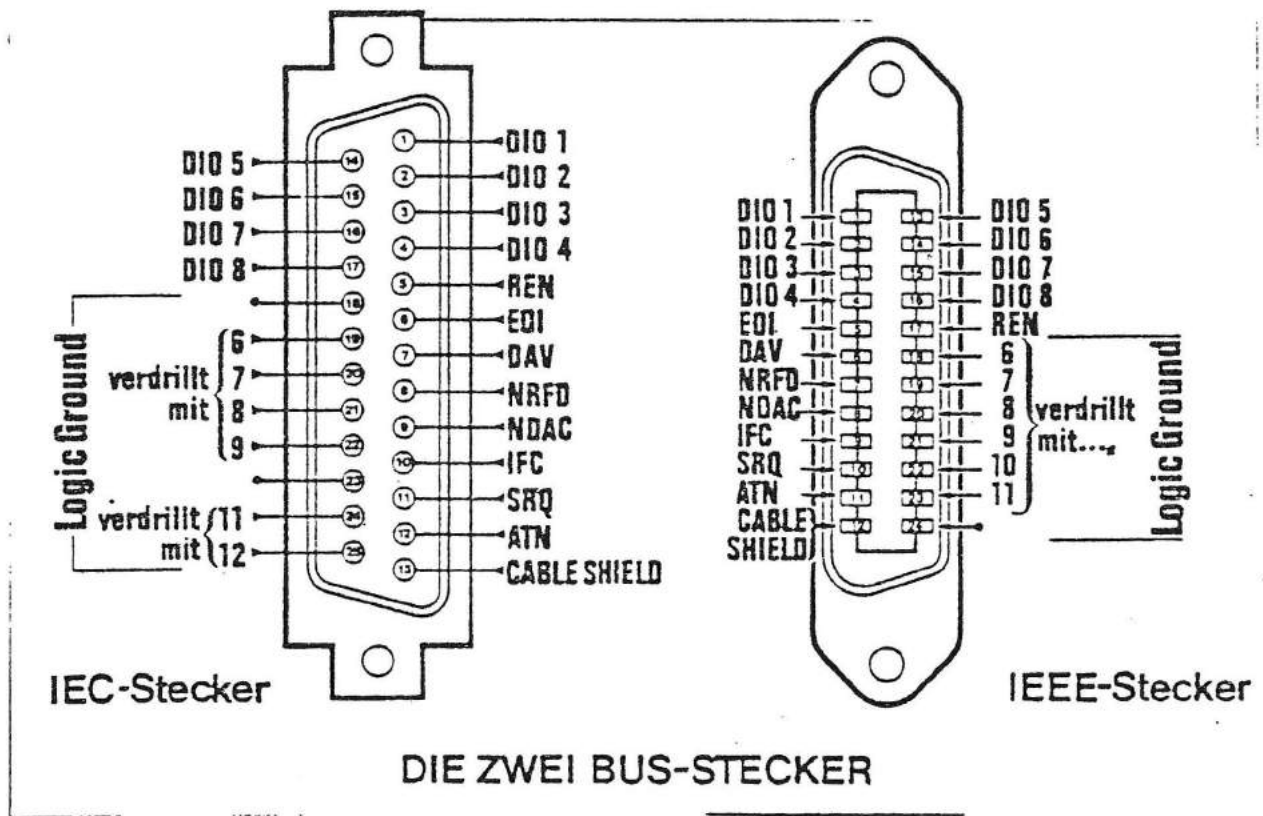
Kommando, mit dem alle Talker gelöscht werden. Talker können auch durch eine nicht verwendete Talkeradresse gelöscht werden.



JEC-Bus-Adapter-Slecker

ITT 3030

AUF WACHSTUM PROGRAMMIERT!



=====
 Buchsenbelegung am ITT 3030 Parallelport
 =====

Beide Stecker haben die gleiche Belegung

40 poliger Stecker direkt am ITT

PA7	— o 1	A	Centronics
PA6	— o 3	l	Data 8
PA5	— o 5	l	⋮
PA4	— o 7	e	⋮
PA3	— o 9		⋮
PA2	— o 11	g	⋮
PA1	— o 13	e	⋮
PA0	— o 15	r	Data 1
		a	
PB7	— o 17	d	Select
PB6	— o 19	e	Select IN/
PB5	— o 21	n	INIT/
PB4	— o 23		STROBE/
PB3	— o 25	P	PE
PB2	— o 27	i	FAULT/
PB1	— o 29	n	Busy
PB0	— o 31	s	Ackn/
<u>ASTB</u>	— o 33		
		G	
<u>BSTB</u>	— o 35	R	
		O	
<u>ARDY</u>	— o 37	U	
		N	
<u>BRDY</u>	— o 39	D	

25 polige Buchsen

PA7	— o 1		
		14 o —	PB7
PA6	— o 2		
		15 o —	PB6
PA5	— o 3		
		16 o —	PB5
PA4	— o 4		
		17 o —	PB4
PA3	— o 5		
		18 o —	PB3
PA2	— o 6		
		19 o —	PB2
PA1	— o 7		
		20 o —	PB1
PA0	— o 8		
		21 o —	PB0
frei	— o 9		
		22 o —	frei
<u>ASTB</u>	— o 10		
		23 o —	<u>BSTB</u>
<u>ARDY</u>	— o 11		
		24 o —	<u>BRDY</u>
GROUND	— o 12		
		25 o —	GROUND
GROUND	— o 13		

Beide Stecker haben die gleiche Belegung

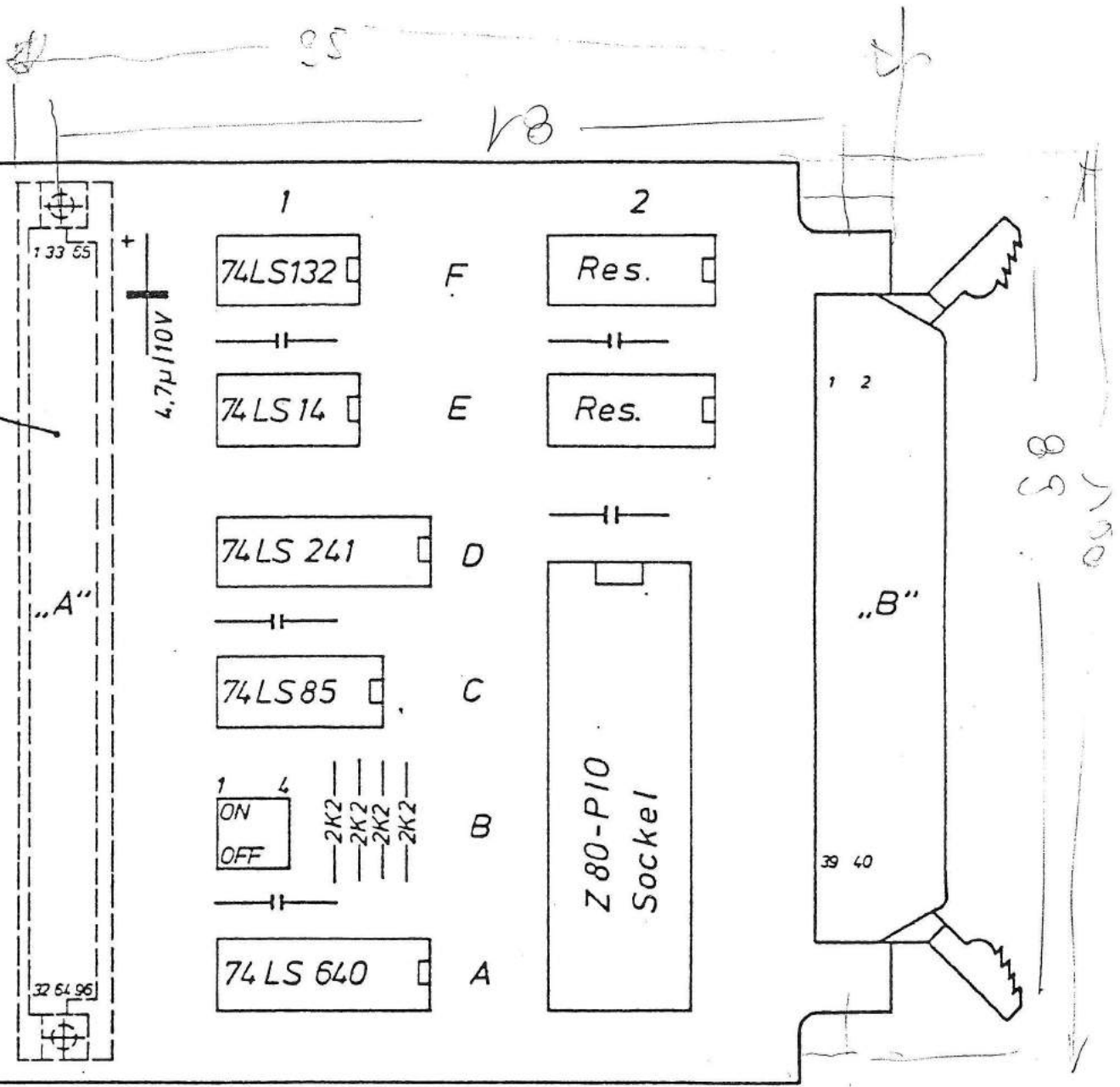
25 polige Buchsen

				Data 8
				⋮
				⋮
				⋮
				⋮
				⋮
				⋮
				⋮
				Data 1
PA7	— o 1			
		14 o —	PB7	Select
PA6	— o 2			
		15 o —	PB6	Select IN/
PA5	— o 3			
		16 o —	PB5	INIT/
PA4	— o 4			
		17 o —	PB4	STROBE/
PA3	— o 5			
		18 o —	PB3	PE
PA2	— o 6			
		19 o —	PB2	FAULT/
PA1	— o 7			
		20 o —	PB1	Busy
PA0	— o 8			
		21 o —	PB0	Ackn/
frei	— o 9			
		22 o —		frei
<u>ASTB</u>	— o 10			
		23 o —		<u>BSTB</u>
<u>ARDY</u>	— o 11			
		24 o —		<u>BRDY</u>
GROUND	— o 12			
		25 o —		GROUND
GROUND	— o 13			

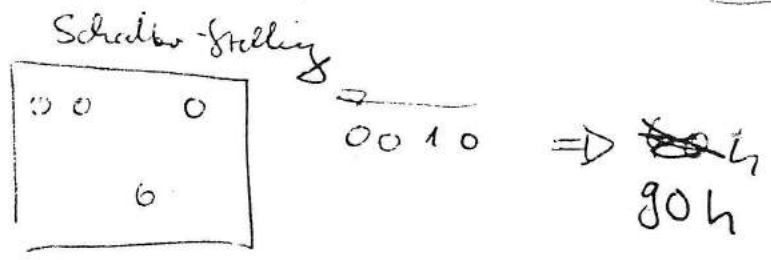
Parallel I/O

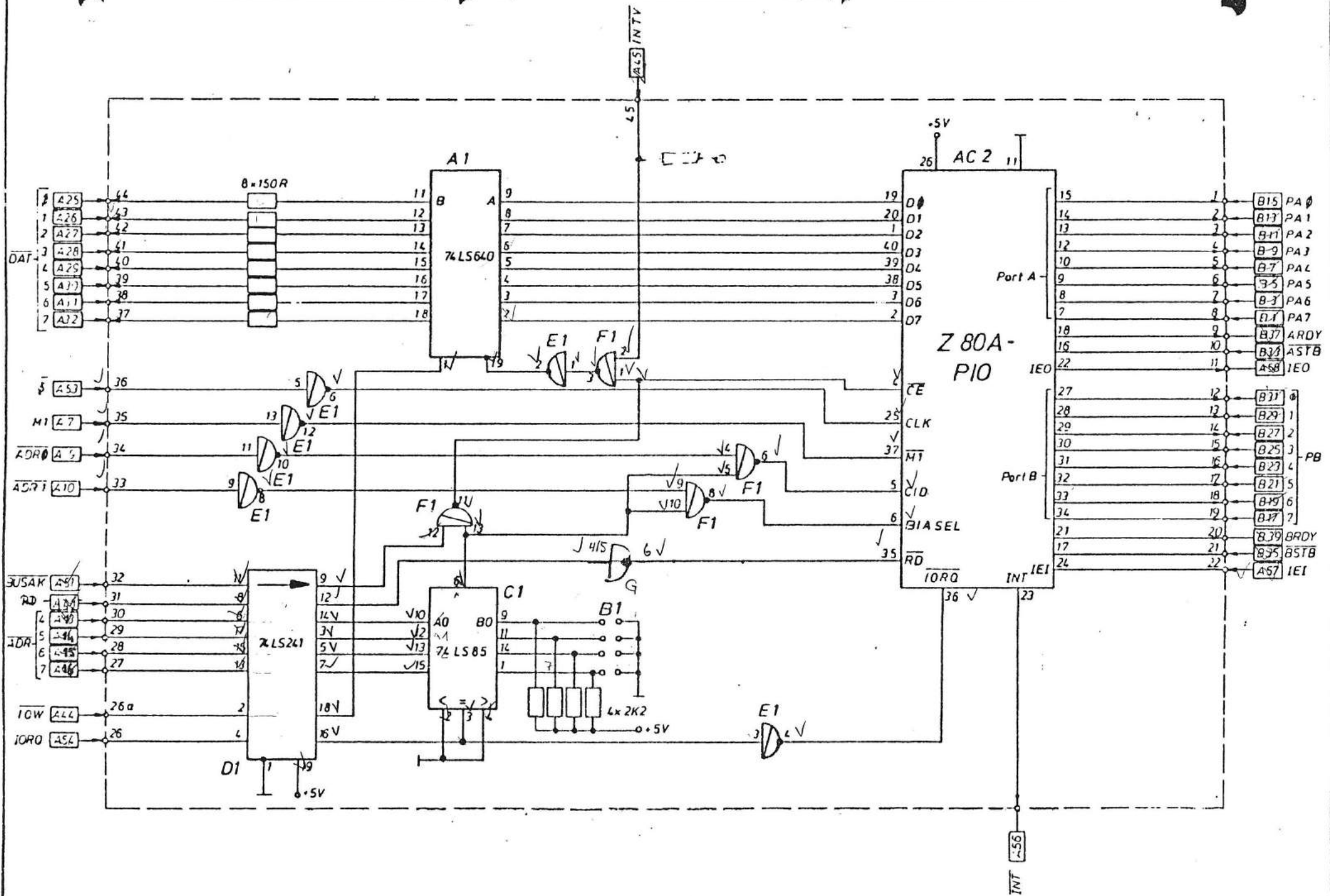
ITT 3030

M 1:14 AUF WACHSTUM PROGRAMMIERT!







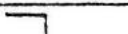

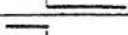


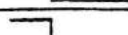
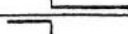
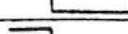
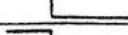
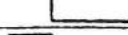
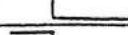
5 x 0.1µ / 50V






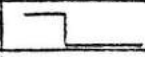
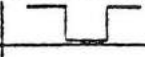
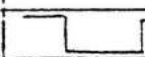

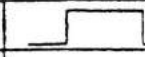
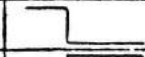
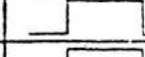
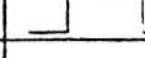
ITT 3030

AUF WACHSTUM PROGRAMMIERT !

PIN	Bezeichnung	Diagramm	Erläuterungen	Blatt Nr.
01	+ 5V			
02	0V			
03	+ 12V			
04	0V			
05	- 12V			
06	0V			
07	M1		Machine Cycle one	
08				
09	$\overline{\text{ADR}}_0$		Adreß-Bit 0	
10	$\overline{\text{ADR}}_1$		Adreß-Bit 1	
11				
12				
13	$\overline{\text{ADR}}_4$		Adreß-Bit 4	
14	$\overline{\text{ADR}}_5$		Adreß-Bit 5	
15	$\overline{\text{ADR}}_6$		Adreß-Bit 6	
16	$\overline{\text{ADR}}_7$		Adreß-Bit 7	
17				
18				
19				
20				
21				
22				
23				
24				
25	$\overline{\text{DAT}}_0$		Daten-Bit 0	
26	$\overline{\text{DAT}}_1$		Daten-Bit 1	
27	$\overline{\text{DAT}}_2$		Daten-Bit 2	
28	$\overline{\text{DAT}}_3$		Daten-Bit 3	
29	$\overline{\text{DAT}}_4$		Daten-Bit 4	
30	$\overline{\text{DAT}}_5$		Daten-Bit 5	
31	$\overline{\text{DAT}}_6$		Daten-Bit 6	
32	$\overline{\text{DAT}}_7$		Daten-Bit 7	

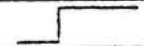




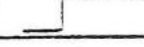
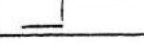
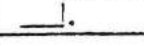
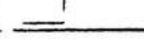
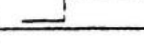
ITT 3030

AUF WACHSTUM PROGRAMMIERT !

PIN	Bezeichnung	Diagramm	Erläuterungen	Blatt Nr.
33	+ 5V			
34	0V			
35	+ 12V			
36	0V			
37	- 12V			
38	0V			
39				
40				
41				
42				
43	$\overline{\text{IOR}}$		IN/OUT Read	
44	$\overline{\text{IOW}}$		IN/OUT Write	
45	$\overline{\text{INTV}}$		Interrupt Vector	
46				
47				
48				
49				
50				
51	$\overline{\text{BUSAK}}$		BUS Acknowledge	
52				
53	\emptyset		Systemtakt	
54	$\overline{\text{IORQ}}$		Input/Output Read	
55				
56	$\overline{\text{INT}}$		Interrupt	
57	$\overline{\text{IEI}}$		Interrupt Enable IN	
58	$\overline{\text{IEO}}$		Interrupt Enable OUT	
59				
60				
61				
62				
63				
64				

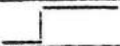

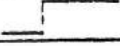

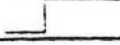
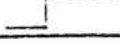
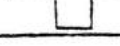
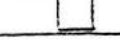
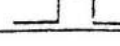
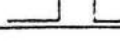
ITT 3030

AUF WACHSTUM PROGRAMMIERT !

PIN	Bezeichnung	Diagramm	Erläuterungen	Blatt Nr.
01	PA7		Port A BUS	
02	OV			
03	PA6		Port A BUS	
04	OV			
05	PA5		Port A BUS	
06	OV			
07	PA4		Port A BUS	
08	OV			
09	PA3		Port A BUS	
10	OV			
11	PA2		Port A BUS	
12	OV			
13	PA1		Port A BUS	
14	OV			
15	PA0		Port A BUS	
16	OV			
17	PB7		Port B BUS	
18	OV			
19	PB6		Port B BUS	
20	OV			

ITT 3030

AUF WACHSTUM PROGRAMMIERT !

PIN	Bezeichnung	Diagramm	Erläuterungen	Blatt Nr.
21	PB5		Port B BUS	
22	OV			
23	PB4		Port B BUS	
24	OV			
25	PB3		Port B BUS	
26	OV			
27	PB2		Port B BUS	
28	OV			
29	PB1		Port B BUS	
30	OV			
31	PB0		Port B BUS	
32	OV			
33	$\overline{\text{ASTB}}$		Port A Strobe	
34	OV			
35	$\overline{\text{BSTB}}$		Port B Strobe	
36	OV			
37	ARDY		Register A Ready	
38	OV			
39	BRDY		Register B Ready	
40	OV			

ITT 3030

AUF WACHSTUM PROGRAMMIERT !

Stromaufnahme:

+ 5V \pm 5% \leq 150 mA

Technik der Welt **ITT**

ITT 3030

AUF WACHSTUM PROGRAMMIERT !

Allgemeine Eigenschaften:

- Z80A - PIO direkt kompatibel mit Z80 CPU
- Zwei 8-Bit Ports
- Vier programmierbare Betriebsarten
- Byte Input, Byte Output, Byte Input/Output (Port A) und Bit in/out
- Port B Ausgänge sind geeignet zum Ansteuern von Darlington-Transistoren.

Technische Beschreibung

Über einen bidirektionalen Bustreiber (LS 640) ist der PIO mit dem Systembus verbunden.

Die Adreßbits 4 - 7 steuern die Adreßerkennung der Karte und erzeugen einen Chipselect für den PIO. Die Adresse 0 und 1 steuern die Umschaltung zwischen Befehl und Daten und die Umschaltung zwischen Port A und Port B. Die Ausgänge des PIO sind auf einen 40-pol. Stecker geführt.

ITT 3030

AUF WACHSTUM PROGRAMMIERT!

Basis-Adresse geändert auf

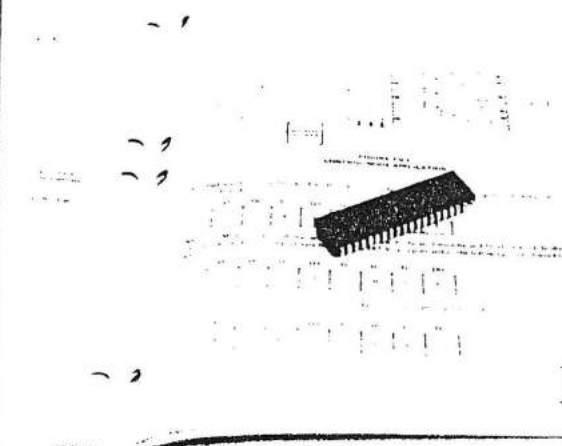
90_h

Portbelegung:

ADR	7	6	5	4	3	2	1	0	
	Basisadr.				X	X	0	0	Befehle PORT B (40) 90
	0 1 0 0				X	X	0	1	Daten PORT B (41) 91
	4				X	X	1	0	Befehle PORT A (42) 92
					X	X	1	1	Daten PORT A (43) 93

Befehle:	0F _H	Daten ausgeben
	4F _H	Daten eingeben
	8F _H	Bidirektional
	CF _H	Control Mode

Über die Signale IEI und IEO kann der Baugruppe eine Interrupt Priorität zugewiesen werden.



Z 80-PPIO Einsatz und Programmierung

Detlef Grell

Schnittstellen sind für ein Computersystem eine Art 'Tor zur Außenwelt'. Die Programmierung komfortabler Peripherie-ICs erfordert aber ein (vielfach ungeliebtes) 'Denken auf Maschinenebene'. Dieser Beitrag gibt Hilfestellung bei der Anwendung des Z.80-PPIO, eines der meistbenutzten Peripherie-Chips. In diesem Zusammenhang wird auch auf einige Tücken der Interrupt-Technik eingegangen, mit der oft auch fortgeschrittene Programmierer auf Kriegsfuß stehen.

Das Businterface stellt über den internen Bus die Verbindung zwischen dem Systembus und entweder einem Ein-Ausgabe-Kanal, der Steuerlogik oder der Interruptlogik her. Jedes PPIO-IC besitzt zwei Adreßleitungen (s. a. Bild 2). Damit bestimmt die CPU, ob Kanal A oder B angesprochen wird und ob Steueranweisungen (Control) erteilt oder Daten (Data) übermittelt werden. Durch die Steueranweisungen werden die Hardware-Funktionen festgelegt.

Für Kanal B sind nur 3 möglich.

Die Betriebsarten (Modes)

- Mode 0 Output (Ausgabe)
- Mode 1 Input (Eingabe)
- Mode 2 Bidirektionaler Betrieb (Zwei-Richtungs-Verkehr, nur bei Kanal A möglich)
- Mode 3 Control-Mode. Jede Datenleitung kann einzeln zur Ein- oder Ausgabe verwendet werden. (Die Steuerleitungen sind wirkungslos.)

Kanal A kann in 4 verschiedenen Modes betrieben werden.

Anders als bei Logik-ICs (Gatter, Flip-Flops) werden verschiedene Funktionen bei modernen Peripherie-ICs durch Programmierung festgelegt. Ein mit derartigen Bausteinen aufgebautes Mikroprozessorsystem kann man daher ohne LötKolben verschiedenen Hardware-Gegebenheiten anpassen.

Der Ein-Ausgabe von 8 Bit gleichzeitig (8 parallele Datenleitungen) verdankt das IC seinen Namen PPIO (Parallel-Input-Output).

Das Blockschaltbild

Bild 1 zeigt die Architektur des Z.80-PPIO. Das IC vereinfacht vor allem den Aufbau von Parallel-Schnittstellen. Zwei unabhängige Kanäle (Ports) mit je 8 Datenleitungen ermöglichen einen Datenverkehr mit externen Geräten, zum Beispiel mit Druckern oder anderen Computersystemen. Außerdem verfügt jeder Kanal über 2 Leitungen zur Ablaufsteuerung des Datenaustauschs (Ready, Strobe).

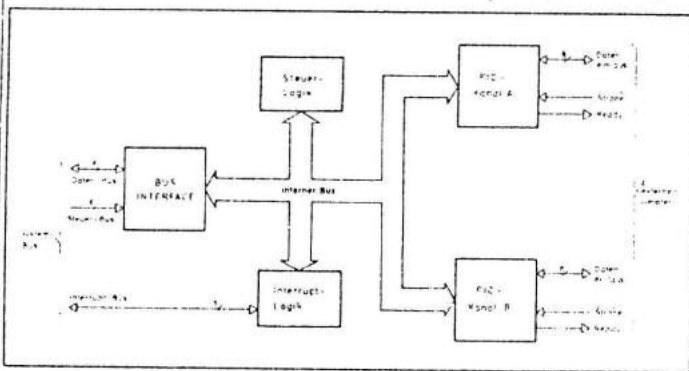


Bild 1. Blockschaltbild des Z.80-PPIO-Bausteins.

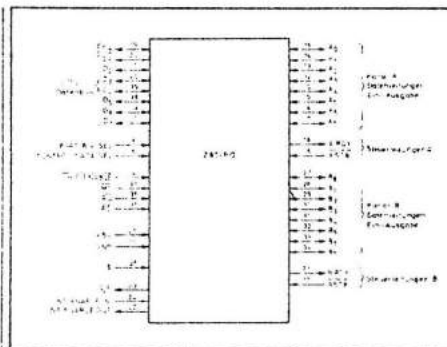


Bild 2. Anschlußbelegung des Z.80-PPIO-Bausteins.

Die gewünschte Betriebsart wird eingestellt, indem der Steuerlogik (Kanal A oder B) ein Steuerwort nach folgendem Muster übergeben wird (per OUT-Befehl).

Betriebsarten-Steuerwort:

D7	D6	D5	D4	D3	D2	D1	D0
M1	M0	X	X	1	1	1	1
Mode	ohne Funktion (beliebig)	Kennzeichnung: Mode-Auswahl					

Dabei besteht folgende Zuordnung:

Mode	M1	M0
0	0	0
1	0	1
2	1	0
3	1	1

Betriebsart 0

Die CPU kann ein Datenbyte in das Kanal-Ausgaberegister einschreiben. Damit steht dieses Byte an den Port-Datenleitungen an und kann von externen Geräten gelesen werden.

Wichtig:

Unabhängig davon, ob dieses Datenbyte von einem externen Gerät auch wirklich übernommen wurde oder nicht, kann die CPU es durch Ausgabe eines neuen Bytes überschreiben. Darüber hinaus kann das im Ausgaberegister stehende Wort (zu Kontrollzwecken) auch von der CPU gelesen werden.

Die Steuerleitungen sind für einen sogenannten Handshake-Betrieb (PIO und externes Gerät 'schütteln sich die Hände') gedacht. Sie sind dann folgendermaßen zu verwenden (Bild 3):

Die Ready-Leitung wird aktiv (1), wenn die CPU per OUT-Befehl ein Byte in das Ausgaberegister schreibt. Das externe Gerät erkennt die Aktivierung der Leitung, übernimmt daraufhin das Datenwort und quittiert die Übernahme durch Aktivierung der Strobe-Leitung (Strobe aktiv = 0). Auf die positive Flanke des Strobe-Signals (wenn Strobe also wieder deaktiviert wird), geht das Ready-Signal in den nichtaktiven Zustand (0) zurück. Gleichzeitig kann, sofern programmiert, dabei ein Interrupt ausgelöst werden.

Der Status der Ready-Leitung kann von der CPU nicht ermittelt werden (kein-Statusregister). Dadurch stellt das Auslösen eines Interrupts die einzige Möglichkeit dar, der CPU anzuzeigen, daß das Datenwort extern gelesen wurde.

Betriebsart 1

Mit einem IN-Befehl kann die CPU ein Datenwort aus dem Eingangsregister eines Kanals lesen. Dabei wird die Ready-Leitung aktiviert. Das zeigt dem externen Gerät an, daß das Eingangsregister neu beschrieben werden darf. Dazu wird die Strobe-Leitung aktiviert und damit das nächste Wort ins Eingangsregister geschrieben. Wiederum wird mit der posi-

ven Flanke des Strobe-Signals die Ready-Leitung deaktiviert, und es kann dabei ein Interrupt ausgelöst werden.

Betriebsart 2 (nur für Kanal A möglich)

Hierbei wird Kanal A zur Ein- und Ausgabe eingesetzt, ohne zwischen Betriebsart 0 und 1 umprogrammiert werden zu müssen. Dabei benutzt Kanal A seine eigenen Handshake-Leitungen zur Steuerung der Ausgabe und die von Kanal B für die Eingabe. Außerdem wird die Interruptlogik von Kanal B mitverwendet, so daß Kanal B besser ganz 'stillgelegt' werden sollte. Dazu an anderer Stelle noch Genaueres.

Anders als in Betriebsart 0 liegt das Datenwort im Ausgangsregister nicht ständig an den Datenleitungen an. Es käme sonst ständig zu (unter Umständen Hardware-gefährdenden) Kollisionen zwischen ein- und ausgehenden Daten. Deshalb wird das Ausgaberegister nur für die Dauer des (extern gegebenen) Strobe-Signals an die Datenleitungen geschaltet.

Die Eingabe-Prozedur verläuft wie in Betriebsart 1, nur werden die Steuerleitungen von Kanal B verwendet.

Betriebsart 3

Die interessantesten und vielfältigsten Möglichkeiten, beispielsweise für Steuerungsaufgaben, bietet die Betriebsart 3. Dementsprechend ist die Programmierung auch etwas komplizierter.

Die 8 Datenleitungen jedes Kanals können beliebig als Ein-

oder Ausgänge definiert werden. Diese Festlegung muß unmittelbar nach der Einstellung von Betriebsart 3 erfolgen. Denn jedes möglicherweise 'ganz anders gemeinte' jezt folgende Steuerwort legt die Ein-Ausgangs-Konfiguration fest.

Steuerwort für Ein-Ausgangs-Festlegung:

D7	D6	D5	D4	D3	D2	D1	D0
I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O

I steht für Input (Eingang). Das 'zuständige' Bit wird zu 1 gesetzt.

O steht für Output (Ausgang). Das entsprechende Bit wird zu 0 gesetzt.

Wenn beispielsweise folgendes Steuerwort gesendet wird:

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	1	1

ergibt sich die korrespondierende Ein-Ausgangs-Konfiguration

D7	D6	D5	D4	D3	D2	D1	D0
IN	OUT	OUT	IN	OUT	OUT	IN	IN

Wird jetzt ein Datenwort in das Ausgaberegister geschrieben, so werden nur die Pegel der programmierten Ausgangsleitungen beeinflußt. Im umgekehrten Fall, wenn die CPU ein Datenwort liest, enthält dieses sowohl den Zustand der Ausgangs- als auch den der Eingangsleitungen. Die weiteren Programmierungsmöglichkeiten in dieser Betriebsart betreffen die Arten der Interrupt-Auslösung. Zum besseren Verständnis wird daher zunächst diese Technik beschrieben.

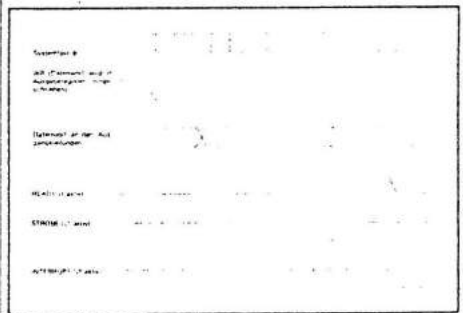


Bild 3. Handshake-Ablauf.

Interrupts

In erster Linie handelt es sich bei Interrupts um Unterprogrammaufrufe, vergleichbar mit den CALL-Befehlen der Z80-CPU. Diese Aufrufe erfolgen jedoch nicht aus einem bestimmten Programm heraus, sondern sie werden hardwareseitig durch Steuersignale an den Prozessor ausgelöst. Ein Peripheriebaustein, der beispielsweise Daten empfangen hat, kann einen Interrupt auslösen. Dann wird das laufende Programm unterbrochen, die CPU springt, wie bei einem CALL-Befehl, zu einer Service-Routine, in der die eingegangenen Daten zum Beispiel gespeichert werden. Anschließend erfolgt die Rückkehr in das unterbrochene Programm.

Für den Fall, daß mehrere Interrupt-Anforderungen gleichzeitig auftreten, muß für eine bestimmte Reihenfolge der Abarbeitung gesorgt werden. Durch spezielle Hardware-Schaltungen (Daisy-Chain) wird dazu eine Rangfolge (Prioritäts-Erkennung realisiert. Z80-Peripherie-ICs verfügen intern bereits über eine derartige Schaltung. Alle Bausteine werden wie bei einer Kette miteinander verbunden, und wer in dieser Kette am weitesten 'vorn' liegt, wird bevorzugt behandelt.

Wird die Interrupt-Anforderung eines niedrig-priorisierten Bausteines gerade bearbeitet, kann ein hoher-priorisierter Baustein die laufende Service-Routine unterbrechen. Erst wenn dessen Routine ausgeführt wurde, kann die alte fortgesetzt werden.

Interrupt-Behandlung (IM2)

Peripherie-Bausteine der Z80-Familie zeichnen sich durch eine weitere Besonderheit gegenüber ähnlichen Produkten an anderer Hersteller aus. Sie sind nämlich speziell für die Unterstützung der Z80-Interrupt-Mode 2 (IM2) ausgelegt. Dieser Beitrag beschränkt sich auf die Betrachtung dieser Betriebsart, da sie für den Z80-Anwender am wichtigsten ist.

Die CPU arbeitet grundsätzlich angefangene Befehle komplett

ab. Dem auslösenden Baustein (höchster Priorität) wird signalisiert, daß seine Anforderung akzeptiert wird. Daraufhin legt dieser die untere Hälfte (Lower-Byte) eines Adressenzeichers auf den Datenbus. Diese untere Hälfte wird im folgenden auch als Interruptvektor bezeichnet. Das CPU-interne I-Register enthält die obere Hälfte (Upper-Byte) dieses Zeigers. Aus Lower- und Upper-Byte wird eine vollständige 16-Bit-Adresse erstellt (Bild 4). Die zwei aufeinanderfolgenden Bytes, die unter dieser Adresse abgelegt sind, werden dann als neue Programmadresse wie für einen CALL-Befehl interpretiert. Nachdem der alte Programmzählerstand auf dem Stack abgelegt wurde, wird zu der neuen Programmadresse verzweigt. Die Rückkehr aus der Interruptroutine erfolgt durch den RETI-Befehl (Return from Interrupt).

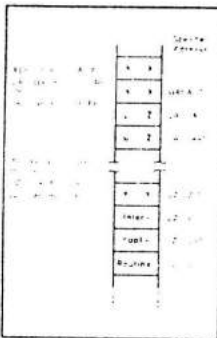


Bild 4. Aufsuchen einer Interrupt-Routine in IM2.

Das Bit 0 des Lower-Byte ist stets 0 und kann nicht verändert werden. Damit wird erreicht, daß die 2 Byte umfassenden Sprungadressen nur mit geradzahigen Adressenzeichern tabelliert werden. So sind bis zu 128 Peripherie-Kanäle ohne zusätzliche Hard- oder Software-Maßnahmen unterscheidbar.

Interrupt-programmierung des Z80-PIO

Beide PIO-Kanäle können eigenständig Interrupts auslösen. Intern ist festgelegt, daß Kanal

A die höhere Priorität besitzt. Der Interruptvektor (Lower-Byte des Adressenzeichers) wird für Kanal A und B gesondert geladen. Er ist beliebig wählbar, aber man sollte sich, um Mehrdeutigkeiten zu vermeiden, davon überzeugen, daß nicht andere Bausteine die gewählte Adresse bereits benutzen.

Interruptvektor:

D7	D6	D5	D4	D3	D2	D1	D0
IV7	IV6	IV5	IV4	IV3	IV2	IV1	0

Der Interruptvektor wird wie ein Steuerwort an den Kanal übergeben. Darüber hinaus muß aber noch durch ein Interrupt-Steuerwort bestimmt werden, ob überhaupt ein Interrupt ausgelöst werden soll und unter welchen Bedingungen.

Interrupt-Steuerwort:

D7	D6	D5	D4	D3	D2	D1	D0
IS7	IS6	IS5	IS4	0	1	1	1

Inter-upt ja/nein

IS7 = 1
Kanal löst Interrupt aus
IS7 = 0
Kanal löst keinen Interrupt aus

Das Interrupt-Steuerwort kann zu beliebigen Zeitpunkten von einem laufenden Programm erneut an den Kanal gegeben werden. (Allerdings darf das nie durch eine Interrupt-Routine erfolgen!) So können Interruptanforderungen von bestimmten Bausteinen unterbunden werden.

Die Funktion der Steuerbits IS4 bis IS6 betrifft nur Betriebsart 3. IS4 = 1 signalisiert der Steuerlogik, daß das nächste Steuerwort eine sogenannte Monitor-Maske ist. Sie legt fest, welche Datenleitungen zum Auslösen eines Interrupts überwacht werden sollen.

Monitor-Maske:

D7	D6	D5	D4	D3	D2	D1	D0
MS7	MS6	MS5	MS4	MS3	MS2	MS1	MS0

Nur die Datenleitungen, für die ein Bit in der Monitor-Maske zu 0 gesetzt wird, werden für die Interrupt-Auslösung ausgewertet. Dabei ist es unerheblich, ob es sich um Ein- oder Ausgänge handelt!

Wenn die Hardware-Funktionen des PIO festgelegt sind, kann ein beliebiges Programm den Interrupt des Bausteins

nachträglich zulassen oder unterbinden. Wenn dann keine neue Monitor-Maske erforderlich ist, muß IS4 = 0 gesetzt werden. Sonst wird einerseits das nächste Steuerwort als Monitor-Maske interpretiert, andererseits ist bis zur Ausgabe dieser Maske die Interrupt-Auslösung gesperrt.

Die Bits IS5 und IS6 bestimmen die Art, in der die durch die Maske ausgewählten Leitungen ausgewertet werden.

IS6 = 0

Die ausgewählten Leitungen werden nach einer ODER-Funktion überwacht. Das heißt, jede Leitung, die in den aktiven Zustand wechselt, kann allein einen Interrupt auslösen. Weitere Leitungsaktivierungen lösen zunächst keinen neuen Interrupt aus. Erst müssen alle Eingangsleitungen deaktiviert worden sein.

IS6 = 1

Die ausgewählten Leitungen werden nach einer UND-Funktion verknüpft. Sie müssen daher zur Auslösung eines Interrupts alle aktiv sein. Hier genügt die Desaktivierung einer Leitung, um durch deren erneute Aktivierung den nächsten Interrupt auszulösen.

IS5 = 0

Der Zustand 0 der Leitungen wird als aktiv angesehen.

IS5 = 1

Als aktiver Zustand gilt 1. Damit sind alle möglichen Programmierungen des Bausteines beschrieben.

Hinweise zum Umgang mit Interrupt-Routinen

Sehr oft werden Interrupt-Routinen bei Echtzeitanwendungen (Steuerungsaufgaben) benötigt. Dann ist eine Erprobung dieser Routinen mit einem Monitor-Programm, zum Beispiel mit Breakpoints oder im Single-Step-Betrieb nahezu unmöglich. Die Routinen müssen also im wesentlichen durch Tüfteln zum Laufen gebracht werden.

Dazu ein paar häufig überlesene Hinweise aus dem CPU-Manual.

Wird ein Interrupt IM2 ausgelöst und von der CPU akzeptiert, bewirkt dies zunächst dasselbe wie der Befehl DI (Dis-

able Interrupt). Weitere Interrupts werden erst nach Erteilen des Befehles EI (Enable Interrupt) bearbeitet. Sollen also höher priorisierte Interrupts laufende Routinen unterbrechen können, dann muß der erste Befehl in einer untergeordneten Routine EI sein. Wenn nicht, so muß aber spätestens vor Verlassen der Interrupt-Routine EI gegeben werden, da sonst weiterhin kein Interrupt möglich ist.

Interrupt-Routinen können ein laufendes Zeitprogramm zu jedem beliebigen Zeitpunkt unterbrechen. Daher ist nie vorhersehbar, welche Register gerade verändert werden dürfen und welche nicht. Es sollten deshalb alle in der Routine veränderten Register vorher auf dem Stack abgelegt und vor dem Verlassen der Routine wiederhergestellt werden.

Aus demselben Grund sollte auch kein Datenaustausch mit dem laufenden Programm über Register erfolgen, sondern über vereinbarte Speicherzellen.

Hinweise zur Hardware:

Kleinere Z80-Systeme (vorwiegend einfache Einkartensysteme) lassen mitunter durch ihre Hardware-Auslegung den Betrieb in IM2 nicht zu. Wenn zum Beispiel nur ein Peripheriebaustein bedient werden soll, so ist dies auch über andere Interrupt-Betriebsarten möglich. Wichtig ist nur, daß man weiß, ob das benutzte System IM2-fähig ist. Wenn das nicht gesondert in den technischen Unterlagen aufgeführt ist, sollte man zunächst davon ausgehen, daß IM2 nicht möglich ist.

Bei fertigen Z80-Systemen sind oft Bustreiber hinter den Datenleitungen der PIO-Kanäle angeordnet. Meistens können mit Steckbrücken nur alle 8 Datentreiber ein- oder Ausgabebereich fest eingestellt werden. Bei eigenen Versuchen mit den Betriebsarten 0 und 1 ist diese Richtung vor der PIO-Programmierung entsprechend zu wählen.

Betriebsart 2 ist mit Treiber-ICs nur möglich, wenn die Hardware selbstständig eine Umschaltung in die entsprechende Richtung veranlaßt. Dieser Fall ist aber sehr unwahrscheinlich. Deshalb sollten in dieser, ganz sicher aber in Betriebsart 3, Treiber-ICs in der Probierphase zunächst entfernt werden.

Wenn Bus-Treiber-Ausgänge auf PIO-Ausgänge treffen, wird mit großer Sicherheit der PIO-Baustein zerstört. Soll eine richtige Schnittstelle zu externen Geräten bleibend installiert werden, muß man aber jede Leitung mit einem Treiber (in die richtige Richtung) versehen.

Zusammenfassung der Programmierung

Betriebsart 0 und 1
Es werden nur 3 Steuerworte an den ausgewählten Kanal übergeben.

1. Interruptvektor (Lower-Byte) (Der Speicherbelegung anpassen, eventuelle Gleichbelegung durch andere Bausteine vermeiden)
2. Betriebsartensteuerwort (Mode 0 oder 1 einstellen)
3. Interrupt-Steuerwort (Interrupt ja/nein durch IS7, IS4 unbedingt zu 0 setzen)

Alle Steuerworte werden an die Steuerungsadresse (Control) des gewünschten Kanals übergeben. Der anschließende Datentransfer erfolgt über die Datenadresse (Data). Diese Adressen liegen systemspezifisch fest und müssen den technischen Unterlagen entnommen werden (siehe auch Bild 2).

- Soll in diesen Betriebsarten grundsätzlich keine Interruptauslösung (IS7 = 0) erfolgen, entfällt die Programmierung des Interruptvektors ersatzlos.
- Betriebsart 2**
Kanal A:
1. Interruptvektor (Lower-Byte)
 2. Betriebsartensteuerwort (Mode 2)
 3. Interrupt-Steuerwort (wie Mode 1 oder 2)

Kanal B muß in diesem Fall seinerseits auf Betriebsart 3 eingestellt werden. Kanal A benutzt die Handshake-Leitungen und den Interruptvektor von Kanal B für die Eingabe. In Mode 2 ist ein Betrieb ohne Interrupts sinnlos, so daß die Interruptauslösung von Kanal B nicht unterbunden werden kann. Aktiver Betrieb von Kanal B in Mode 3 würde aber zu unplanmäßigen Aufrufen der Eingaberoutine führen. Kanal B wird daher stillgelegt.

Kanal B:

1. Interrupt-Vektor (für Eingabe-Routine Kanal A)
2. Betriebsartensteuerwort (Mode 3)
3. Steuerwort zur Festlegung der Ein-Ausgangs-Konfiguration (beliebig, muß aber übergeben werden)
4. Interrupt-Steuerwort (IS4 = 1)
5. Monitor-Maske (alle Bits zu 1 setzen, dadurch kann keine Datenleitung Interrupt auslösen.)

Betriebsart 3

Die Reihenfolge der Steuerwort-Ausgabe ist wie unter Betriebsart 2 für Kanal B ausgeführt. Die Steuerworte sind entsprechend der eingangs gegebenen Beschreibung zu wählen.

Betriebsart 3 eignet sich am ehesten dazu, ohne Interrupts zu arbeiten. So kann man Eingangsleitungen dazu verwenden,

den, der CPU mitzuteilen, daß ein bestimmtes Ereignis eingetreten ist. Die CPU muß dazu den Kanal (durch das laufende Programm) gelegentlich lesen. Vom Zustand der Eingangsleitungen kann man dann den weiteren Programmablauf abhängig machen. Dann vereinfacht sich die Kanal-Programmierung zu:

1. Betriebsartensteuerwort
2. Ein-Ausgabe-Steuerwort
3. Interruptsteuerwort (IS7 = 0, IS4 = 0)

Wer sich im Umgang mit dem PIO üben möchte ohne externe Geräte oder den PIO selbst zu gefährden, kann sich folgende Schaltung selbst aufbauen (Bild 5). Sie dient dazu, die Zustände an den Datenleitungen eines PIO-Kanals durch Leuchtdioden (LEDs) anzuzeigen. Es werden zwar Inverter verwendet (IC1), aber die LEDs leuchten, wenn die Ausgänge der Inverter auf logisch 0 wechseln. Eine leuchtende LED entspricht daher einer logischen 1 auf der Datenleitung.

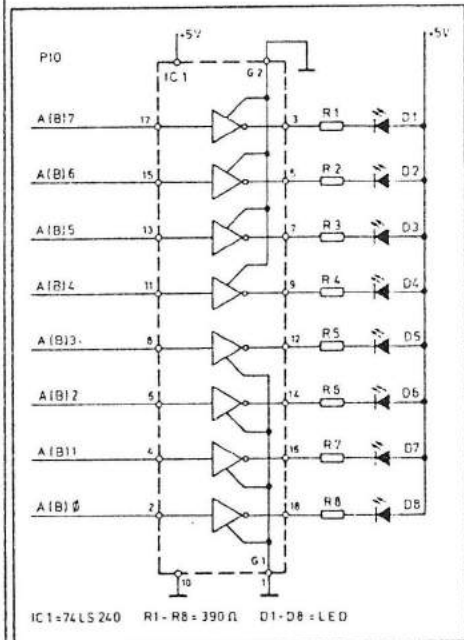


Bild 5. Testschaltung

PARALLEL INPUT/OUTPUT INTERFACE

LIST OF CONTENTS

1.	Introduction.....	3
2.	General Description.....	4
3.	Installation Instructions.....	5
	- Definition of Priorities.....	5
	- Hardware Addressing of the PCBs (Microswitch).....	7
4.	Software Control and Programming.....	9
4.1	Selecting an Operating Mode.....	9
4.2	Description of the Operating Modes.....	10
4.3	Interrupt Control.....	12
4.4	Programming Example.....	15
5.	Circuit Diagram.....	17
6.	Technical Data.....	18
7.	Connector Description.....	19
8.	Ordering Instructions.....	20

© 1983 by Standard Elektrik Lorenz AG, Pforzheim, Germany; All rights reserved
 Production: INDUSTRIEHANSA Systemelektronik GmbH, München
 Printer: Pfeiler Druck GmbH, Münchener Str. 39, 8200 Rosenheim

This book must not, in whole or in part, be copied,
 printed or reproduced in any material form without the
 permission of Standard Elektrik Lorenz AG, Pforzheim.

The publication of information in this document does not imply
 freedom from patent and other protective rights of
 Standard Elektrik Lorenz AG, Pforzheim, or others.

List of Figures

1. ITT 3010 with the Parallel Input/Output Interface.....3

2. Block Diagram of the Parallel Input/Output Interface.....4

3. Installation of the PIO PCB.....6

4. PIO Layout with Addressing Microswitch.....7

5. Detailed Block Diagram of an Input/Output Port.....11

6. Practical Example of the Use of the PIO.....15

7. PIO Circuit Diagram.....17

8. View of the Input/Output Connector.....19

1. INTRODUCTION

The Parallel Input/Output Interface is a programmable digital device designed to enable the ITT 3030 microcomputer to operate with external equipment. It can be used, for example to set switches, to determine switch positions, to operate peripheral equipment or to monitor and control processes.

Parallel Input/Output means that as signals PA0 - PA7 (Port A) and PB0 - PB7 (Port B) are presented simultaneously to the input/output connections. Each channel can operate in either input or output mode. The signals presented to each channel can have one of two possible values (digital TTL values):

- Logic 1 = High (5V)
- Logic 0 = Low (0V)

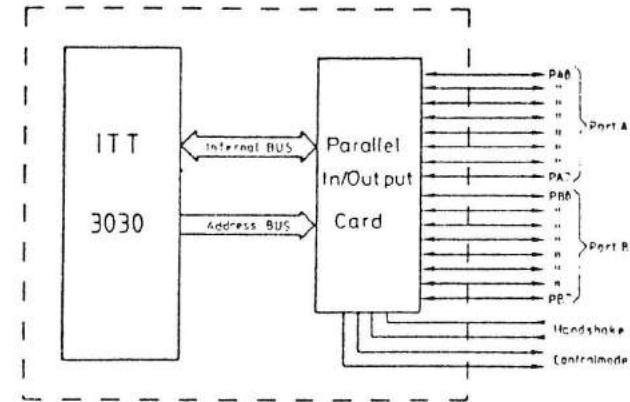


Fig.1: ITT 3030 with the Parallel Input/Output Interface

As the PIO is a programmable device using the ITT 3030, it is possible to change the signal direction i.e. input or output and the value of the output signals. It is also possible for the CPU to read the state of the channels. The major design characteristics of the PIO can be summarised as follows:

1. Two independent peripheral interface ports with "handshake" data transfer control.
2. Interrupt driven "handshake" for fast response.
3. Four distinct modes of operation.
4. Daisy chain priority interrupt logic included for automatic interrupt vectoring without external logic.
5. All inputs and outputs are fully TTL compatible.

These features are described below.

2. GENERAL DESCRIPTION

The Parallel Input/Output Interface consists of 16 "channels" that can be either transmitted or received by the ITT 3030. The 16 channels (bits) are combined into two 8-bit groups called Port A and Port B. Each port can be individually interrogated and programmed. As a result, four distinct modes of operation are available:

Mode 0 - Transmission of data
 Mode 1 - Reception of data
 Mode 2 - Bidirectional data bus operation
 (Port A only, Port B cannot be used)
 Mode 3 - Single bit input/output

These modes can be programmed using special control words or instructions either in BASIC or ASSEMBLER. Operation is also possible under full interrupt control (see section 3.3).

BASIC Instructions : INP, OUT, WAIT
 ASSEMBLER Instructions : IN, OUT

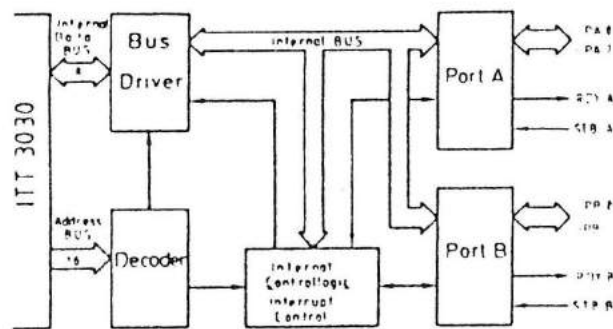


Fig. 2: Block Diagram of the Parallel Input/Output Interface

All input/output gates are fully TTL-compatible and the 8 outputs forming Port B can also be used to drive Darlington transistors.

3. INSTALLATION INSTRUCTIONS

Instructions for the assembly and dismantling of the ITT 3030 are contained in the User's handbook, but the following instructions for the insertion of PCB's are given for the sake of completeness:

1. Switch the equipment OFF and remove the power cables from the mains sockets.
2. Remove the top cover (see the ITT 3030 User's handbook).
3. As required, remove the Floppy Disc Drives (this assumes that this operation is not the first installation of the system).
4. Release the Floppy Disc Drives retaining plate and remove. To achieve this, remove the black AMP connector from the right hand side of the retaining plate. Release the locking device on the underside of the left hand side of the retaining plate by turning it to the left. The retaining plate is now free and can be removed.
5. Select the required connector position on the rear of the unit and remove the protective cover from the slot. If the computer is to be used for interrupt operation, the selection of the appropriate connector must follow the interrupt priority that is fixed by the hardware. The priority is fixed via a daisy chain and in this instance is as follows:

Viewed from the front:

1. Left hand connector (highest priority)
2. Middle connector
3. Right hand connector (lowest priority)

The daisy chain must not be interrupted and it may be necessary to re-arrange already installed PCBs in order to achieve the required priority sequence. The addresses for these PCBs will not be changed by this process.

6. The required PCB can now be installed into the slot and the connector for the internal data bus pressed into place on the adjacent circuit board.

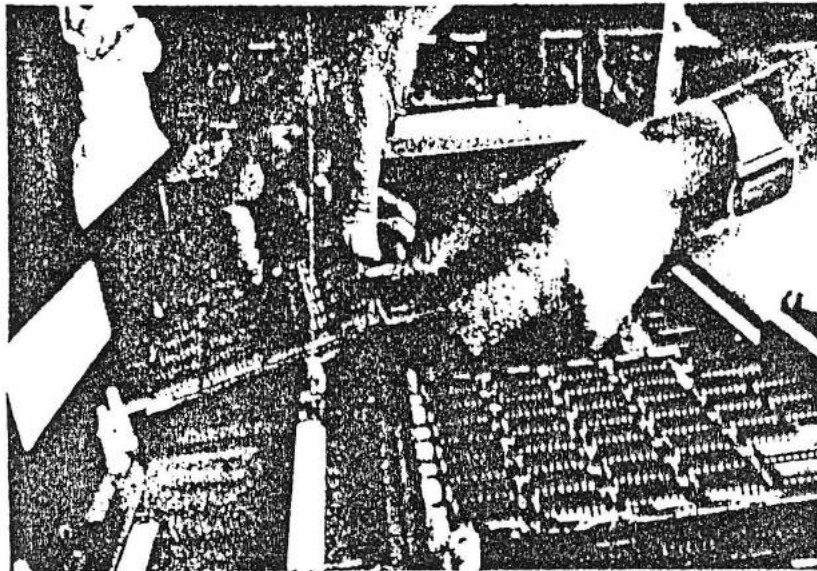
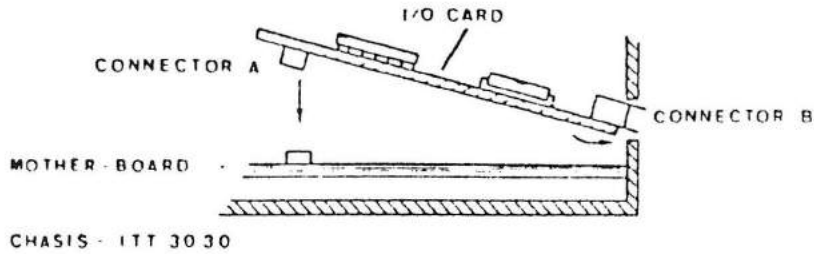
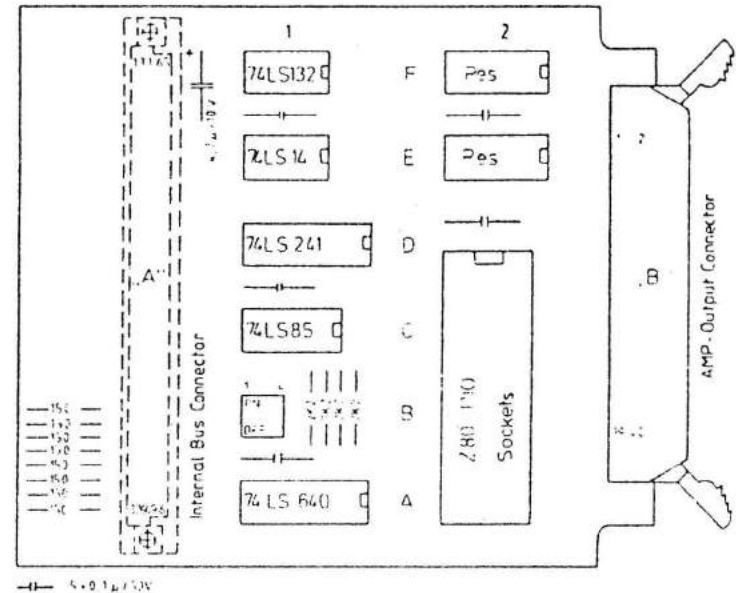


Fig.3: Installation of the PIO PCB

7. Addressing

Every PCB must, at installation, be given an individual address that can be used by the program to uniquely identify the card. No two interfaces (including other types such as RS 232 and IEEE Bus Interfaces) may have identical addresses.



Address Switches (4Bit-Base address)

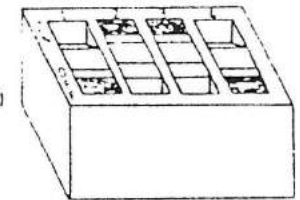


Fig.4: PIO Layout with Addressing Microswitch

The address is selected using the microswitch on the PCB in accordance with the following table:

Address		Switch			
Dec	Hex	S4	S3	S2	S1
0	00	0	0	0	0
16	10	0	0	0	1
32	20	0	0	1	0
48	30	0	0	1	1
64	40	0	1	0	0
80	50	0	1	0	1
96	60	0	1	1	0
112	70	0	1	1	1
128	80	1	0	0	0
144	90	1	0	0	1
160	A0	1	0	1	0
176	B0	1	0	1	1
192	C0	1	1	0	0
208	D0	1	1	0	1
224	E0	1	1	1	0
240	F0	1	1	1	1

0 = Switch position OFF
1 = Switch position ON

A few addresses have already been allocated and must not be used. These are:

00, 10, 20, 30, 50, 60, E0 and F0 (hexadecimal)

The PIO Interface will be delivered with address 90H preset by the manufacturer.

8. Reassembly

Reassembly is in the reverse order in accordance with the User handbook. (For initial installation of the system, follow the assembly instructions in the User documentation.)

4. SOFTWARE CONTROL AND PROGRAMMING

4.1 SELECTING AN OPERATING MODE

It is necessary to define the operating mode for the Parallel Input/Output Interface using control words before undertaking any data transfers. The PIO will decode the lower 8 bits of the address word to determine whether the following information is to be interpreted as an instruction or as data. They also define which port is to be used.

These 8 bits have the following format:

Address Bits								Decode Information
A7	A6	A5	A4	A3	A2	A1	A0	
X	0	0	0	0	0	0	0	Instructions Port B
X	0	0	0	1	0	0	0	Data Port B
X	0	1	0	0	0	0	0	Instructions Port A
X	0	1	1	0	0	0	0	Data Port A
X	1	0	0	0	0	0	0	Reset after Interrupt
PCB Address (set by microswitch)								

X = not used

The PIO PCB must be identified (addressed) within the micro-computer system itself. The address is set when the PIO PCB is first installed into the system using the in-built microswitches. It is usually specified in hexadecimal based on the calculated values of bits A4-A7 and assuming that bits A0-A3=0. When, for example bits A4 and A7 are set using microswitch positions S1 and S4, this PCB address is 90H, where the H stands for hexadecimal.

The Control Word for mode selection has the following format:

Data Bits								Mode
D7	D6	D5	D4	D3	D2	D1	D0	
0	0	X	X	1	1	1	1	0=Byte Output
0	1	X	X	1	1	1	1	1=Byte Input
1	0	X	X	1	1	1	1	2=Bidirectional Bus
1	1	X	X	1	1	1	1	3=Bit Input/Output

X = not used

Byte= 8 bits

Bits D3-D0 must all be set to "1" to indicate that a new mode is to be set.

4.2 DESCRIPTION OF THE OPERATING MODES

MODE 0 - BYTE OUTPUT

When Mode 0 is selected, the 8 data bits will be sent via the internal data bus connecting the PIO with the CPU to the output data bus for the required Port. The contents of the output register can be changed at any time simply by sending a new data word from the CPU to the PIO. Also, the contents of the output register can be read back to the CPU using an input instruction.

When data for the CPU has been written to the selected port, the READY line (1-high) for the required port will be activated, notifying the peripheral that data is available. This signal will remain active until a "received" signal is sent from the receiving peripheral. This "received" signal is in the form of a STROBE pulse whose rising edge generates an interrupt (if enabled) by the appropriate control word, see also section 4.1) and resets the READY line to low. This very simple handshake system is common to many peripheral devices.

MODE 1 - BYTE INPUT

When Mode 1 is selected, the required port will be set to input data from the peripheral. To start the handshake routine, the CPU performs an input read operation from the port. This activates the READY line to the peripheral to indicate that data should be loaded into the empty input register. The peripheral then loads the data into the input register using the STROBE signal. Again, the rising edge of the strobe causes an interrupt request (if enabled) and resets the READY line.

MODE 2 - BIDIRECTIONAL DATA TRANSFER

This operating mode allows a two directional exchange of information between the ITT 1030 and peripheral devices such as printers or other computers that must operate using a parallel interface. As this mode requires the use of all four handshake lines, only Port A can be used. Port B must be set to Mode 2 and its signals (PB0-PB7) suppressed using an appropriate mask (see section 4.3). The handshake signals for Port A are used for output control and those for Port B are used for input control. The only difference between Mode 2 output operation and Mode 0 is that data from the Port A output register is allowed onto the port data bus only when STROBE A is active.

MODE 3 - BIT OPERATION

Mode 3 operation is intended for status and control applications and does not utilise the handshake signals (see the example in section 4.4). When Mode 3 is selected, the next word sent to the PIO must be a control word and must define which of the port data bus lines are to be inputs and which are to be outputs.

The control word has the following format:

D7	D6	D5	D4	D3	D2	D1	D0
I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O

If any bit is set to "1" then the corresponding data bus line will be used as an input. Conversely, if the bit is reset i.e. "0", the line will be used as an output. During Mode 3 operation the strobe signal is ignored and the READY line remains inactive. Data may be written to a port or read from a port by the CPU at any time during Mode 3 operation. When reading a port, the data returned to the CPU will be composed of input data from the port data bus lines assigned as inputs plus port output register data from those lines assigned as outputs. Additionally, this mode offers the possibility to generate interrupts on reaching specified conditions at the input lines. This option frees the processor from status checking (polling) and is explained further in the following chapter.

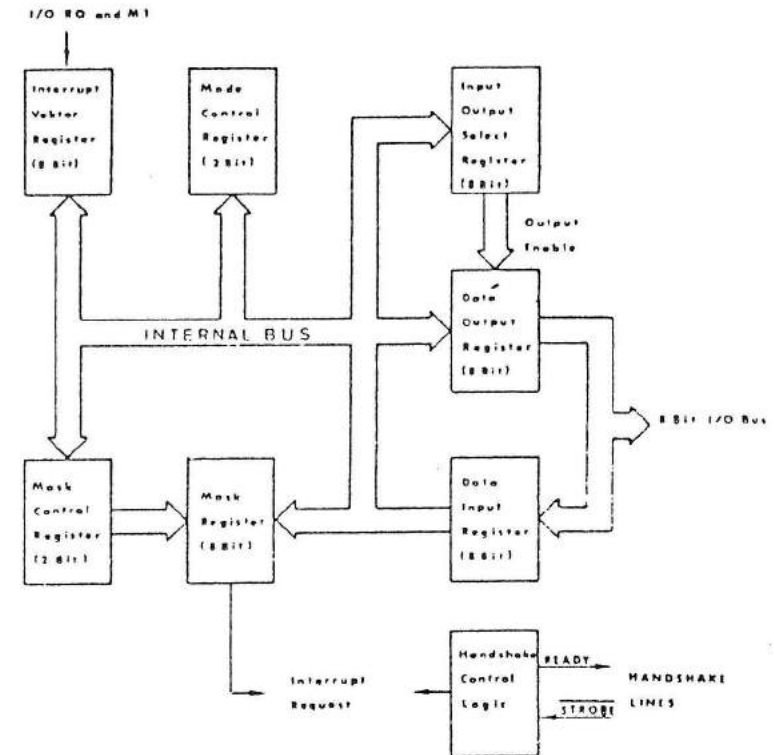


Fig.5: Detailed Block Diagram of an Input/Output Port

1.3 INTERRUPT CONTROL

As stated in the previous section, the Parallel Input/Output interface can operate under full interrupt control. A description of this follows the description of the mode of operation of the Z80 CPU interrupt.

The PIO has been developed to operate with the Z80 CPU in Interrupt Mode 2. This operating mode requires the CPU to receive an interrupt address (vector) from the peripheral requesting the interrupt. This interrupt will cause the CPU to suspend the sequencing of the program in progress and, in this case, force the CPU to service the PIO port requesting the interrupt (e.g. a real time input of new data)

The interrupt vector will be placed on the internal data bus by the interrupting port with the highest priority (Port A before Port B) and will point to a table in the memory. This table forms a series of pointers to the locations where the service routines for the appropriate interrupts are to be found.

The interrupt vector is loaded into the PIO by writing a control word to the desired port of the PIO using the following format:

D7	D6	D5	D4	D3	D2	D1	D0
V7	V6	V5	V4	V3	V2	V1	0

The data bit D0 is used as a flag bit and signifies that this control word is an interrupt vector and must be loaded into the vector register. At interrupt acknowledge time, the vector of the interrupting port will appear on the Z80 data bus exactly as shown in the format above.

It is necessary to have an interrupt control word for each port in addition to the interrupt vector. This control word will allow the interrupt to be processed or suppressed as required, and in Mode 3 operation informs the PIO that the next control word to be received is a mask.

The interrupt control word has the following format:

D7	D6	D5	D4	D3	D2	D1	D0
IE	AO	HL	MA	0	1	1	1

signifies interrupt control word

- IE = Interrupt Enable (permits interrupts)
- AO = AND/OR (AND=1, OR=0)
- HL = HIGH/LOW (HIGH=1, LOW=0)
- MA = Mask (Mask follows = 1)

Bits D4, D5 and D6 are only used in Mode 3 operation and will be ignored in all other modes. These three bits are used to allow for interrupt operation in Mode 3 when any group of the input/output lines adopts certain defined states. However, setting bit D4 of the interrupt control word during any mode of operation will cause any pending interrupt to be reset.

Bit D6 (AND/OR) defines the logical operation to be performed in port monitoring. If bit D6=1, an AND function is selected and all specified bits must adopt a defined state before an interrupt is generated, while the OR function will generate an interrupt if any specified bit goes to the active state.

D5 defines the active polarity of the port data bus. If bit D5=1, the port data lines are monitored for a high state while, if D5=0 they will be monitored for a low state.

When bit D4=1 in Mode 3 the next control word sent to the PIO must define a mask as follows:

D7	D6	D5	D4	D3	D2	D1	D0
MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0

Only those port lines whose mask bit is zero will be monitored for generating an interrupt.

This particular interrupt operation (i.e. one that reacts automatically to set external conditions) results in a considerable saving in computer time that would otherwise be used in determining the status of the peripheral devices.

An example using interrupt routines is given in section 4.4.

Interrupt Priority (Daisy Chain)

The interrupt control logic section handles all CPU interrupt protocols for nested priority structures. The priority of any device is determined by its physical location in a "daisy chain" configuration. Two lines are provided in each PIO to form this daisy chain. The device closest to the CPU has the highest priority. In this case, the system has been hardware configured so that the left connector (viewed from the front) has the highest priority after the combo chip on the CPU PCB. Within the PIO itself, Port A interrupts have a higher priority than those of Port B. The PIO provides for complete control of nested interrupts. That is, lower priority devices may not interrupt higher priority devices that have not had their interrupt service routine completed by the CPU. Higher priority devices may interrupt the servicing of lower priority devices.

Interrupt Service Routines

When writing the interrupt service routine, it is imperative to take note of a peculiarity in its operation. Normally, the interrupting device will be reset using the instruction RETI (Return from Interrupt). The PIO interface requires an additional instruction in order to reset the interrupting port. Therefore, the final two instructions of the interrupt servicing routine must be in the form:

```
OUT PCB address + 4 (i.e. Address bit A2=1)
RETI
```

Note:

This instruction resets both PIO ports (hardware reset). Therefore both ports must be re-initialized following each interrupt.

4.4 PROGRAMMING EXAMPLE

Central Heating Control

A typical example for the use of single-bit Mode 3 operation is the control of a central heating system. In this example the individual bits for the control and status words have the following definition:

Bit	Signal	Input/Output
D0	Equipment "ON" switch	PA0
D1	Heating "ON"	PA1
D2	not used	PA2
D3	Temperature Alarm	PA3
D4	Excess Pressure Alarm	PA4
D5	Emergency "OFF" switch	PA5
D6	Alarm signal "ON"	PA6
D7	not used	PA7

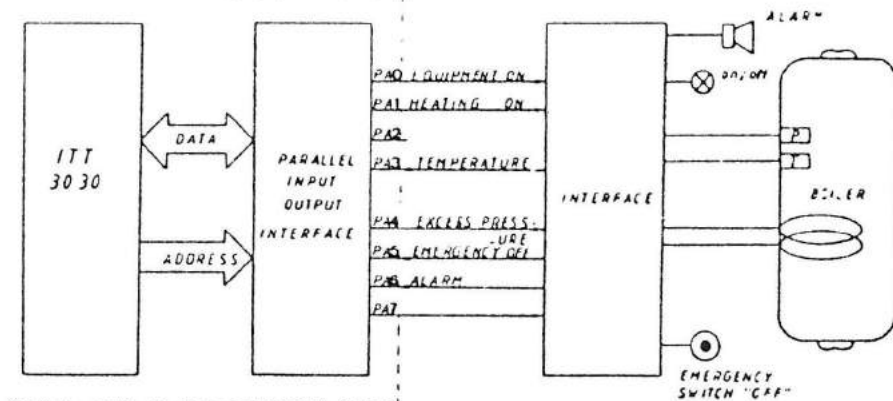


Fig.6: Practical Example of the Use of the PIO

6. TECHNICAL DATA

Internal voltage requirements for the computer :

$$V_{cc} = 5V \pm 5\%, I_{cc} = 150mA$$

Absolute Maximum Ratings :

Storage Temperature	-25 C to +65 C rel. humidity of the air 10-80%
Working Temperature	+10 C to +35 C rel. humidity of the air 20-80%
Voltage on any pin with respect to ground	-0.3V to +7 V
Power dissipation	0.6W

Input/Output Channel Characteristics :

16 parallel input/output channels configured as two 8-bit ports Ports A and B.

All channels fully TTL-compatible

The outputs from Port B (PB0-PB7) can drive Darlington transistors.

Priorities : defined by card position (leftmost position - highest priority)
Port A has priority over Port B.

D.C. Characteristics

Specification	Parameter	Min	Typ	Dim
Input Voltage Low (0)	V	-0.1	0.8	V
Input Voltage High(1)	V	2	V_{cc}	V
Output Voltage Low (0)	V	$I = 2mA$	0.4	V
Output Voltage High(1)	V	$I = -250\mu A$	2.4	V
Input Leakage Current	I	V = 0-5V	10	μA
Tristate Output Leakage Current (High)	I	V = 2.4-5V	10	μA
Tristate Output Leakage Current (Low)	I	V = 0.4V	-10	μA
Darlington Drive Current	I	V = 1.5V R = 390 only Port B	-1.5	3.8 mA

7. CONNECTOR DESCRIPTION

The Output connector (B) for the PIO Interface has the following identifier:

AMP Latch 826/40 Pin

The connector has the following pin allocations:

Pin	Identifier	Diagram	Remarks
1	PA7		Port A output 7
2	Ground		
3	PA6		Port A output 6
4	Ground		
5	PA5		Port A output 5
6	Ground		
7	PA4		Port A output 4
8	Ground		
9	PA3		Port A output 3
10	Ground		
11	PA2		Port A output 2
12	Ground		
13	PA1		Port A output 1
14	Ground		
15	PA0		Port A output 0
16	Ground		
17	PB7		Port B output 7
18	Ground		
19	PB6		Port B output 6
20	Ground		
21	PB5		Port B output 5
22	Ground		
23	PB4		Port B output 4
24	Ground		
25	PB3		Port B output 3
26	Ground		
27	PB2		Port B output 2
28	Ground		
29	PB1		Port B output 1
30	Ground		
31	PB0		Port B output 0
32	Ground		
33	ASTB		Port A Strobe
34	Ground		
35	BSTB		Port B Strobe
36	Ground		
37	ARDY		Port A READY
38	Ground		
39	BRDY		Port B READY
40	Ground		

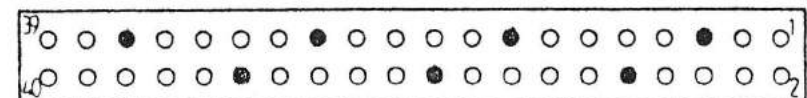


Fig.8: View of the Input/Output Connector (B)

8. ORDERING INSTRUCTIONS

Description	Part Number	Remarks
ITT 1070 Parallel Input/Output PCB	59712142	16 inputs/outputs 2 Ports
Interface Connector		Male for I/O Connector (B)

