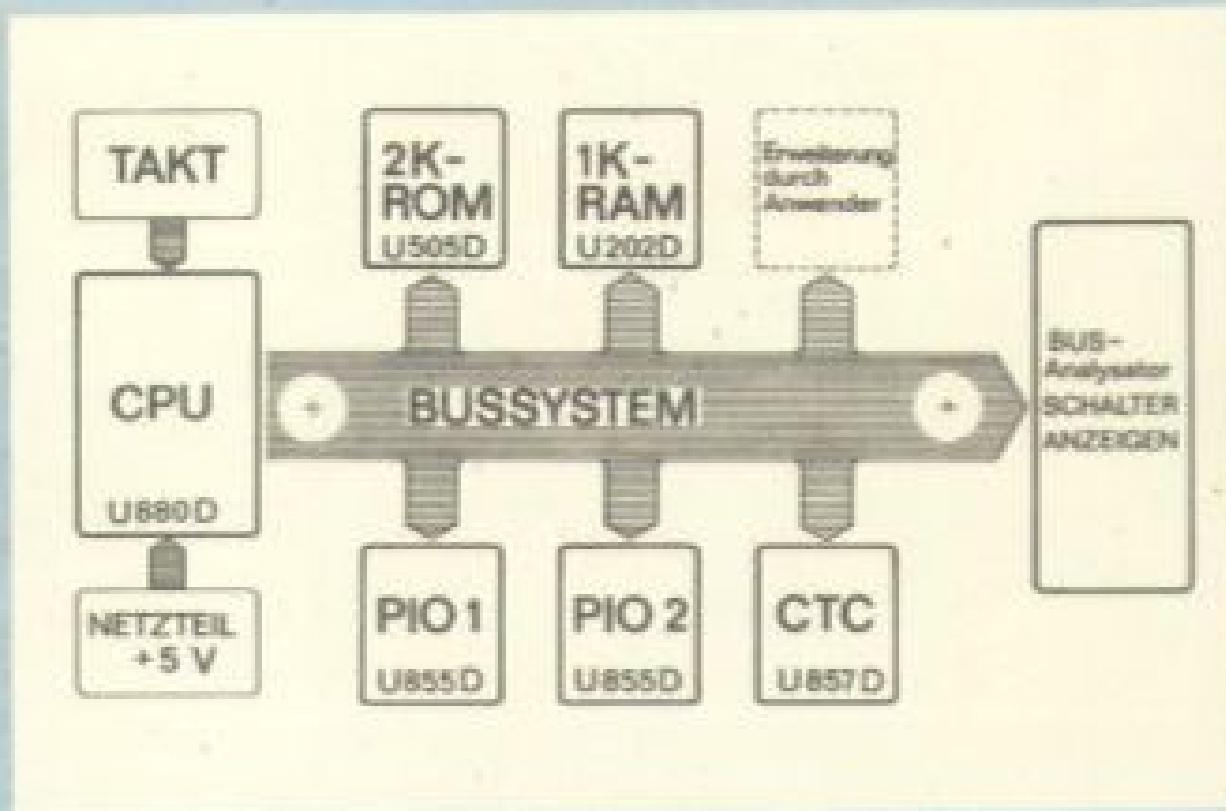


POLY-COMPUTER 880

Systemhandbuch
11 548 6.01



0 - 0883 - 11548

VEB KOMBINAT
POLYTECHNIK UND PRÄZISIONSGERÄTE
KARL-MARX-STADT



Systemhandbuch zum Mikrorechner-Lernsystem "Poly-Computer 880"

Dieses Handbuch enthält in konzentrierter Form alle Informationen, die zur Anwendung und Programmierung von Mikrorechnersystemen mit der Schaltkreisfamilie U880D, U855D, U856D, U857D notwendig sind. Es ist als Nachschlagwerk bei der Beschäftigung mit dem Arbeitsbuch sowie bei der Anwendung der erworbenen Kenntnisse und Fertigkeiten sinnvoll einsetzbar. Um die Bezeichnungsweise bei der Bearbeitung von Hard- und Softwareproblemen zu vereinheitlichen, wird in diesem Handbuch mit 0 der TTL-LOW-Pegel und mit 1 der TTL-HIGH-Pegel bezeichnet.

Inhaltsverzeichnis

CPU U880D

- Struktur
- Anschlußbeschreibung
- Funktion
- Unterbrechungsverarbeitung
- Befehlsformate
- Befehle, binär (mit Flagbeeinflussung)
- Assembler → Hex (Befehle nach Assemblerbezeichnung geordnet)
- Hex → Assembler (Befehle nach Operationskodes geordnet)
- Arbeitsblatt Befehlstabellen
- Elektrische Kennwerte

PIO U855D

- Struktur
- Anschlußbeschreibung
- Funktion
- Programmierung
- Elektrische Kennwerte

SIO U856D

- Struktur
- Anschlußbeschreibung
- Funktion
- Programmierung
- Elektrische Kennwerte

CTC U857D

- Struktur
- Anschlußbeschreibung
- Funktion
- Programmierung
- Elektrische Kennwerte

Speicherbausteine:

1KByte-ROM U505D
Anschlußbeschreibung
Elektrische Kennwerte

1KByte-EPROM U555D
Anschlußbeschreibung
Funktion
Elektrische Kennwerte

Autorenkollektiv:
Dipl.-Ing. Steffen Burkhardt
Dipl.-Ing. Uwe Hübner
Dip.-Ing. Andreas Troll

Hersteller:



**VEB Kombinat Polytechnik und
Präzisionsgeräte Karl-Marx-Stadt
Stammbetrieb VEB Polytechnik
Karl-Marx-Stadt**
**DDR - 9023 Karl-Marx-Stadt
Melanchthonstraße 4/8, PSF 93**
Deutsche Demokratische Republik

2KByte-E PROM 2716

Anschlußbeschreibung

Funktion

Elektrische Kennwerte

stat. 1KBit-RAM U202D

Anschlußbeschreibung

Elektrische Kennwerte

dyn. 16KBit-RAM U256C

Anschlußbeschreibung

Funktion

Elektrische Kennwerte

Hilfsbausteine in TTL-Technik:

8-Bit-E/A-Schaltkreis 8212

Anschlußbeschreibung

Funktion

Elektrische Kennwerte

4-Bit-bidirektionaler Treiber 8216

Anschlußbeschreibung

Funktion

Elektrische Kennwerte

1-aus-8-Dekoder 8205

Anschlußbeschreibung

Funktion

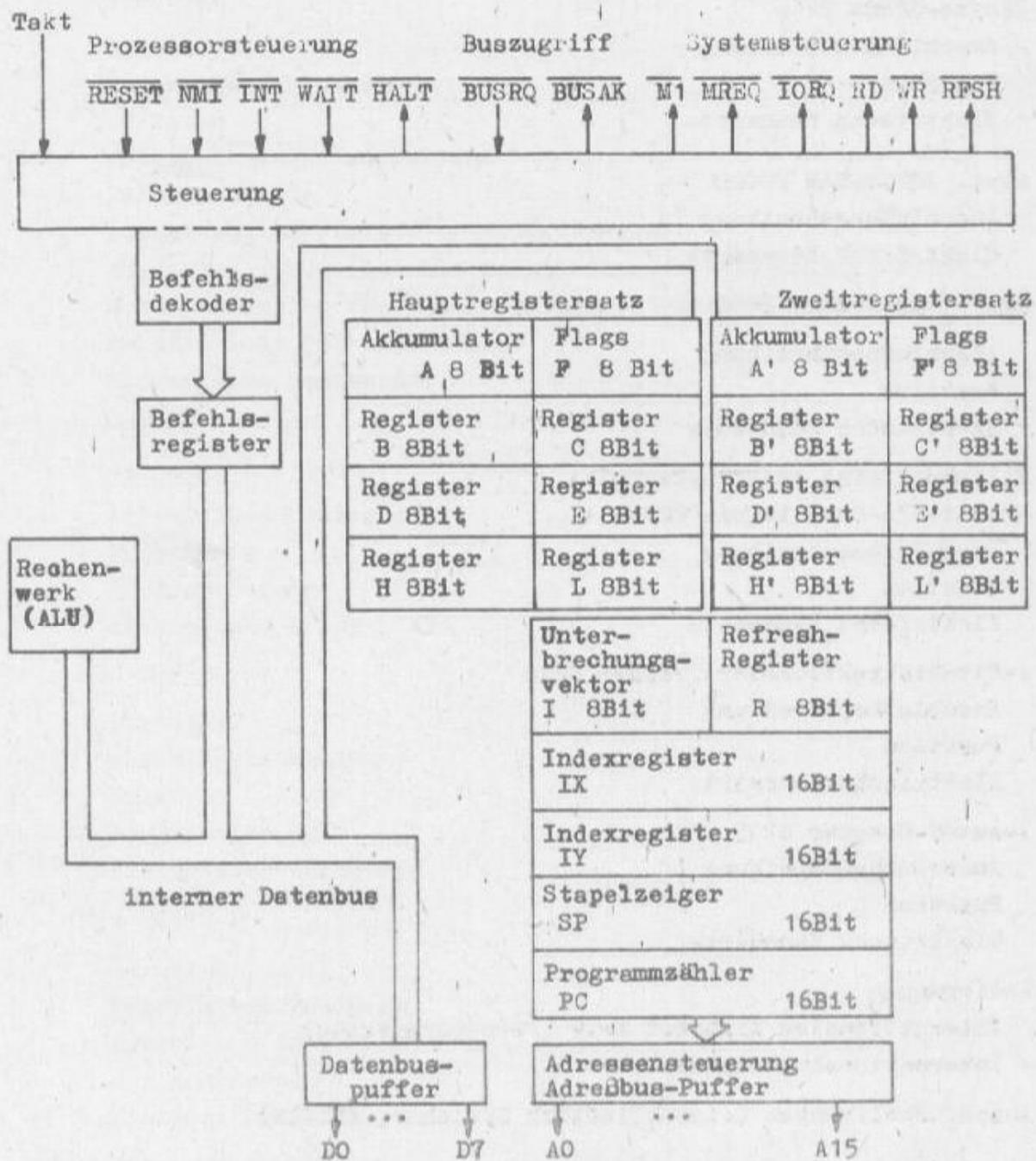
Elektrische Kennwerte

Kodierungen

Internationales Alphabet Nr.2 (Fernschreikode)

Internationales Alphabet Nr. 5

Anschlußbelegungen (einschließlich Speicher, TTL-IS)



- Merkmale:**
- Direkt adressierbare Speicherkapazität : 64 KByte
 - maskierbare (wahlweise sperrbare) Unterbrechung mit 3 möglichen Betriebsarten
 - spezielle nichtmaskierbare Unterbrechung (NMI)
 - Auffrischsteuerung für dynamische Speicher
 - Einphasentakt, Betriebsspannung +5V, TTL-kompatibel (Ausgänge treiben 1 Lasteinheit)
 - maximale Taktfrequenz des Grundtyps: 2,5 MHz, Abarbeitungszeit je Befehl: 1,6...9,2 µs
 - 158 Basisbefehle
 - 2 Registersätze, 2 Indexregister
 - Kellerspeicher (Stack) im externen RAM

CPU U880D Anschlußbeschreibung

14	DO	CPU U880	A0	30
15	D1		A1	31
12	D2		A2	32
8	D3		A3	33
7	D4		A4	34
9	D5		A5	35
10	D6		A6	36
13	D7		A7	37
24	WAIT		A8	38
16	INT		A9	39
17	NMI		A10	40
26	RESET		A11	1
25	BUSRQ		A12	2
6	C		A13	3
			A14	4
			A15	5
			M1	27
			MREQ	19
			IORQ	20
			RD	21
			WR	22
			RFSH	28
			HALT	18
			BUSAK	23

- A0 - A15 Ausgänge (tri-state) Systemadreßbus zur Adressierung von Speicher und E/A-Bausteinen
- D0 - D7 Ein-/Ausgänge (tri-state) Systemdatenbus
- M1 Ausgang O-Operationskode-Lesezyklus der CPU (Machine Cycle 1) bzw. Annahme einer Unterbrechungsanforderung (zusammen mit IORQ)
- MREQ Ausgang (tri-state) O-Speicheranforderung durch CPU (Memory Request)
- IORQ Ausgang (tri-state) O-Ein-/Ausgabeanforderung durch CPU (Input/Output Request)
- RD Ausgang (tri-state) Lesesignal (Read)
O-Lesen vom Speicher oder E/A-Bausteinen

CPU U880D Anschlußbeschreibung

- **WR** Ausgang (tri-state) Schreibsignal (Write)
0-Schreiben in Speicher oder E/A-Baustein
- **RFSH** Ausgang Speicherauffrischungssignal (Refresh)
0-Die niedrigewertigen 7 Bit des Adressbusses führen eine Adresse zum Auffrischen dynamischer Speicher.
- **HALT** Ausgang 0-CPU im Halt-Zustand, wird nach Ausführung des HALT-Befehls eingenommen und nur durch Unterbrechung oder Rücksetzen verlassen; es werden Leerbefehle (NOP) zur Speicherauffrischung ausgeführt.
- **WAIT** Eingang Warte-Anforderungen,
0-CPU wartet, bis Speicher oder E/A-Schaltung bereit ist.
- **INT** Eingang Unterbrechungsanforderung (Interrupt), maskierbar, 0-Ubergang in eine Unterbrechungsbehandlung (wenn Unterbrechungen erlaubt) nach Beendigung des gerade bearbeiteten Befehls entsprechend der vorgewählten Behandlungsart (IMO, 1,2)
- **NMI** Eingang nichtmaskierbare Unterbrechungsanforderung (Nonmaskable Interrupt).
0-Aufruf eines Behandlungsprogramms auf Adresse 66H.
- **RESET** Eingang Rücksetzen der CPU,
0-Unterbrechung sperren, PC:=0, I:=0, R:=0, IMO
Daten- und Adressbus hochohmig, übrige Ausgänge inaktiv.
- **BUSRQ** Eingang Busanforderung (Bus Request)

0-Adress-Daten- und Steuerbus (**MREQ**, **TORQ**, **RD**, **WR**) in hochohmigen Zustand bringen, so daß sie von anderen Einrichtungen benutzt werden können.
- **BUSA^K** Ausgang Busanforderungsbestätigung (Bus Acknowledge)
0-Adress-Daten- und Steuerbus sind in hochohmigen Zustand geschaltet.
- C Eingang Systemtakt (Clock)

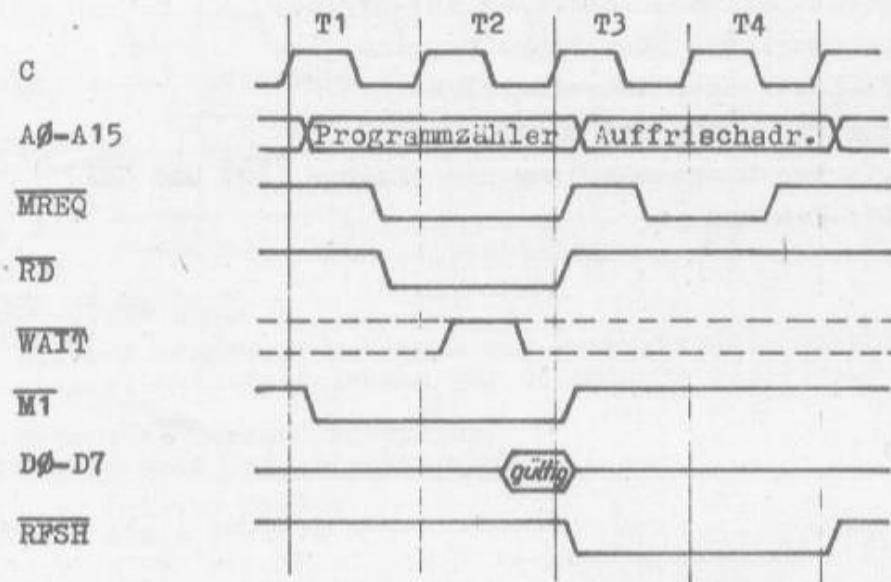
CPU U880D Funktion

Die Arbeit der CPU setzt sich aus folgenden Ablauftypen zusammen:

- Operationskode-Lesezyklus (M_1 -Zyklus)
- Speicherlese- oder Schreibzyklus
- E/A-Lese- oder Schreibzyklus
- Busanforderung und Busrückgabe
- Unterbrechungsanmeldung und Annahme (\overline{INT} und \overline{NMI})
- Halt-Zustand

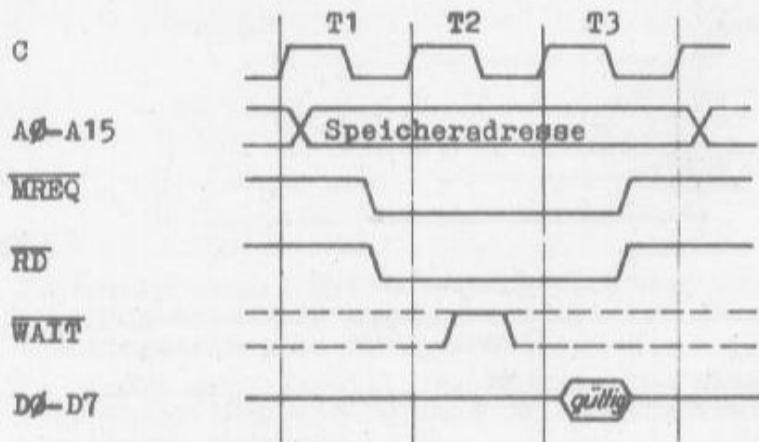
Achtung: Jeweils nicht dargestellte Steuersignale sind inaktiv! Die Darstellungen zeigen den qualitativen Ablauf, für Zeiten und Zeitverhältnisse sind nur die dynamischen Kennwerte relevant!

Operationskode-Lesezyklus (M1-Zyklus)

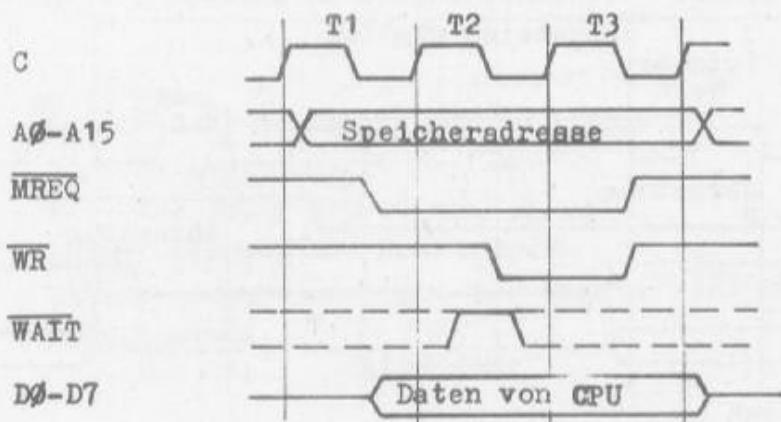


- MREQ wird aktiv, wenn die Adresse gültig ist; verwendbar zur Aktivierung von Speicher- bzw. E/A-Bausteinen
- Mit steigender Flanke von RD werden die Daten von der CPU übernommen
- Während T3, T4 wird ein Auffrischen dynamischer Speicher durchgeführt

Speicher-Lesezyklus

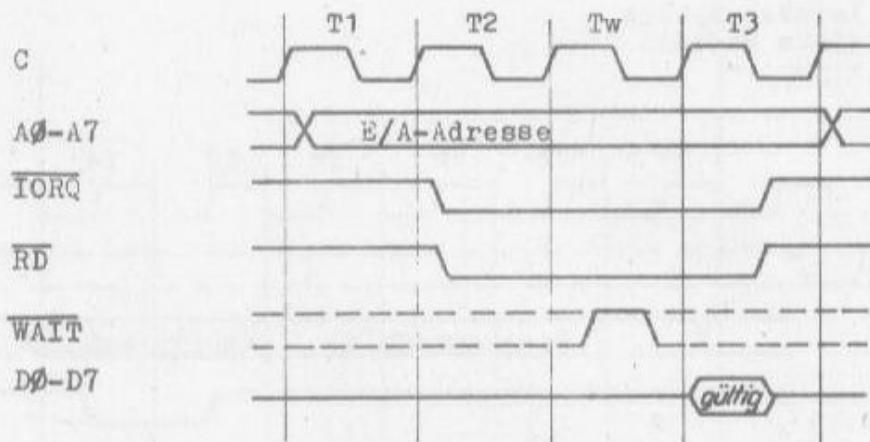


Speicher-Schreibzyklus



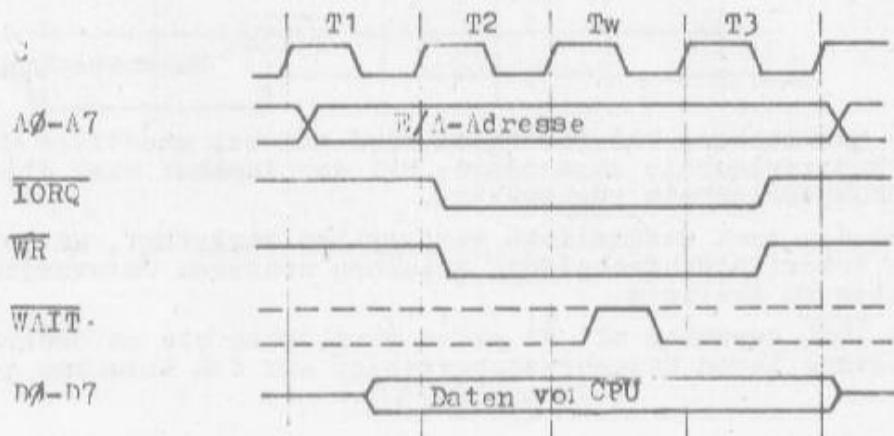
- Wenn \overline{WR} aktiv ist ($\overline{WR} = 0$), sind die Daten auf dem Datenbus gültig

Ein-/Ausgabe-Lesezyklus

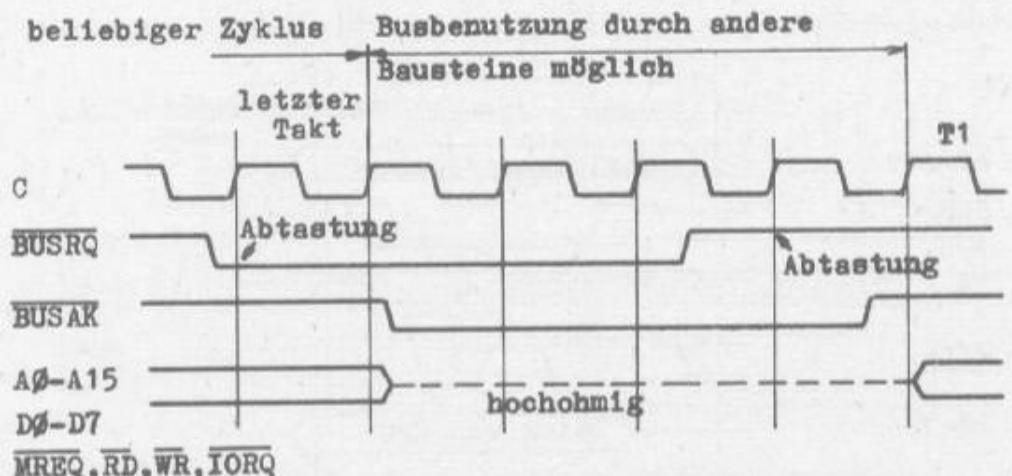


- Nach T2 wird automatisch ein Wartetakt Tw eingeführt, um für die Logik des E/A-Bausteins zusätzliche Verarbeitungszeit zu ermöglichen

Ein-/Ausgabe-Schreibzyklus

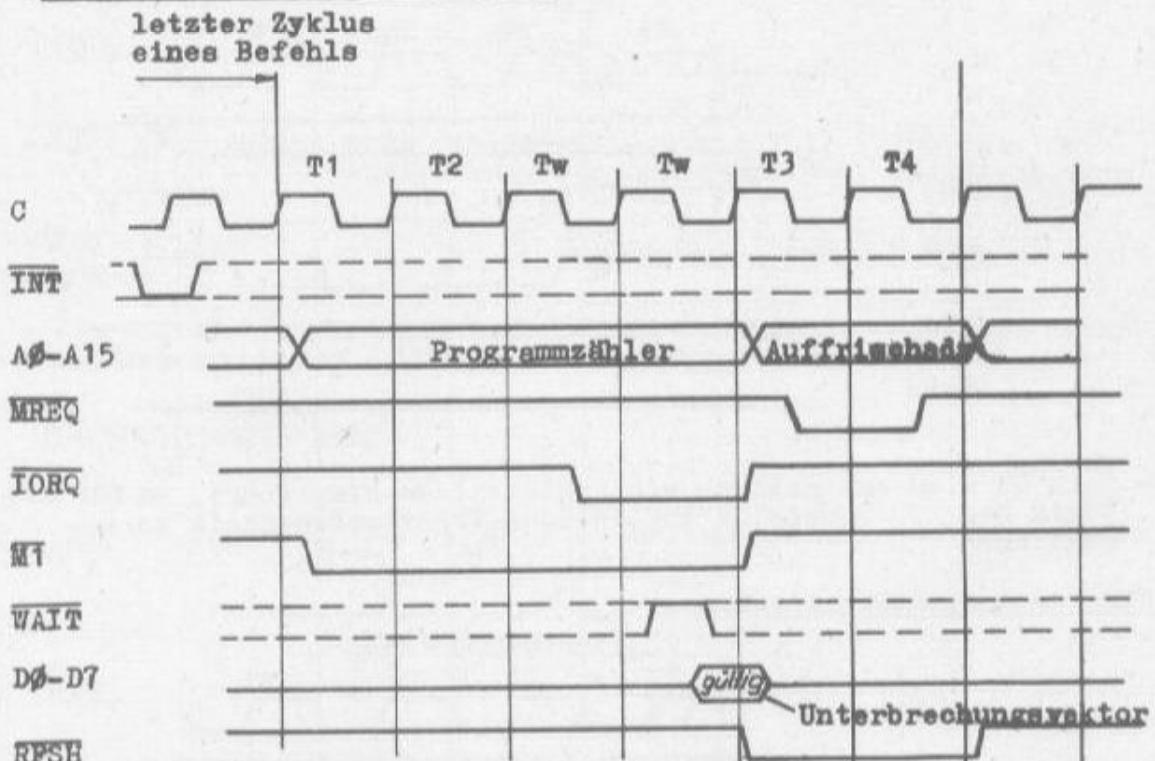


Busanforderung und Busrückgabe



- Während Adressbus, Datenbus und Steuersignale hochohmig sind, erfolgt kein Auffrischen für dynamische Speicher durch die CPU

Unterbrechungsempfahlyzyklus
(maskierbare Unterbrechung-INT)



- Eine maskierbare Unterbrechung wird nur bei gesetzter Unterbrechungserlaubnis angenommen, bei der Annahme wird die Unterbrechungserlaubnis rückgesetzt.
- Es werden zwei zusätzliche Wartezyklen eingefügt, um Zeit für eine Prioritätsentscheidung zwischen mehreren Unterbrechungsquellen zu gewinnen.
- Wenn IORQ zusammen mit M1 aktiv wird, kann die anfordarnde Schaltung ihren Unterbrechungsvektor auf den Datenbus geben.

Nichtmaskierbare Unterbrechung (NMI)

- Wird in jedem Fall angenommen (nicht durch Programm sperrbar)
- Nach Beendigung des Befehls, in dem NMI=0 erkannt wurde, wird ein Operationskode-Lesezyklus (M1-Zyklus) ausgeführt, der Datenbusinhalt wird aber ignoriert; anschließend wird der Programmszähler im Kellerspeicher abgelegt (wie bei CALL, RST) und die Adresse 66H angesprungen.
- Eine Reaktion auf weitere (maskierbare und nicht maskierbare) Unterbrechungen erfolgt erst nach Ausführung eines RETN- oder EI-Befehls.
- Dauer der NMI-Annahme: 10 Takte

Maskierbare (sperrbare) Unterbrechung (INT)

- Die Unterbrechungserlaubnis wird in dem Flip-Flop IFF1 gespeichert, d.h. wenn IFF1=0 ist die Unterbrechung gesperrt.
- Zur Sicherung des Zustands dieses Flip-Flops bei Behandlungen nichtmaskierbarer Unterbrechungen dient ein weiteres Flip-Flop (IFF2).
- Unterbrechungen können durch den Befehl EI erlaubt und durch den Befehl DI gesperrt werden.
- Nach dem Rücksetzen der CPU sowie nach Annahme einer Unterbrechung ist die Unterbrechungserlaubnis rückgesetzt (Unterbrechungen sind gesperrt)!

Die folgende Tabelle zeigt die Operationen mit dem Unterbrechungserlaubnis-Flip-Flops:

Aktion	IFF1	IFF2	
CPU RESET	0	0	
DI	0	0	
EI	1	1	
LD A,I	.	.	IFF2 → P/V-Flag
LD A,R	.	.	IFF2 → P/V-Flag
Annahme von NMI	0	.	
RETN	IFF2	.	IFF2 → IFF1
Annahme von INT	0	0	
RETI	.	.	

. bedeutet: keine Veränderung

- Der Befehl RETI dient dazu, daß unterbrechende Geräte die Beendigung ihrer Behandlung erkennen können.

- Dauer der INT-Annahme: 13 Takte bei IM1 und IM0 (RST-Ausführung)
19 Takte bei IM2

Der Prozessor kann für drei unterschiedliche Unterbrechungsverarbeitungsmoden programmiert werden:

IM0

Beim Unterbrechungsannahmzyklus legt das die Unterbrechung anfordernde Gerät ein Bitmuster auf den Datenbus, das vom Prozessor wie ein aus dem Speicher gelesener Befehl ausgeführt wird. Meist wird das ein Kurzruf (Restart-RST) sein, da so mit nur einem Befehlsbyte eines von 8 möglichen Unterbrechungsbehandlungsprogrammen aufgerufen werden kann.

Nach dem Rücksetzen des Prozessors ist dieser Modus gesetzt.

IM1

Beim Unterbrechungsannahmzyklus wird der Datenbusinhalt ignoriert, der Programmzähler wird wie bei CALL/RST im Kellerspeicher (Stack) abgespeichert, anschließend wird die Adresse 38H angesprungen (dort muß das Unterbrechungsbehandlungsprogramm beginnen).

IM2

Bei diesem leistungsfähigsten Modus führt der Prozessor mit Hilfe eines vom unterbrechenden Gerät gelieferten 8-Bit-Wertes (Unterbrechungsvektor) einen indirekten Unterprogrammaufruf zu einer beliebigen Speicheradresse aus.

Der Programmierer hat im Speicher eine Tabelle mit den Startadressen der Unterbrechungsbehandlungsprogramme bereitzustellen. Bei der Unterbrechungsannahme wird ein Zeiger gebildet, dessen höherwertige 8 Bits vom Inhalt des Registers I und niedrigerwertigen 8 Bits von dem vom unterbrechenden Gerät gelieferten Wert gebildet werden. Dieser Zeiger weist auf den niedrigerwertigen Teil der Startadresse in der Tabelle der Startadressen der Unterbrechungsbehandlungsprogramme. Der nächste (nächsthöhere) Speicherplatz der Tabelle enthält den höherwertigen Teil der Startadresse. Zu beachten ist, daß das niedrigerwertigste Bit des Zeigers und damit des vom unterbrechenden Gerät gelieferten 8-Bit-Wertes Null sein muß:



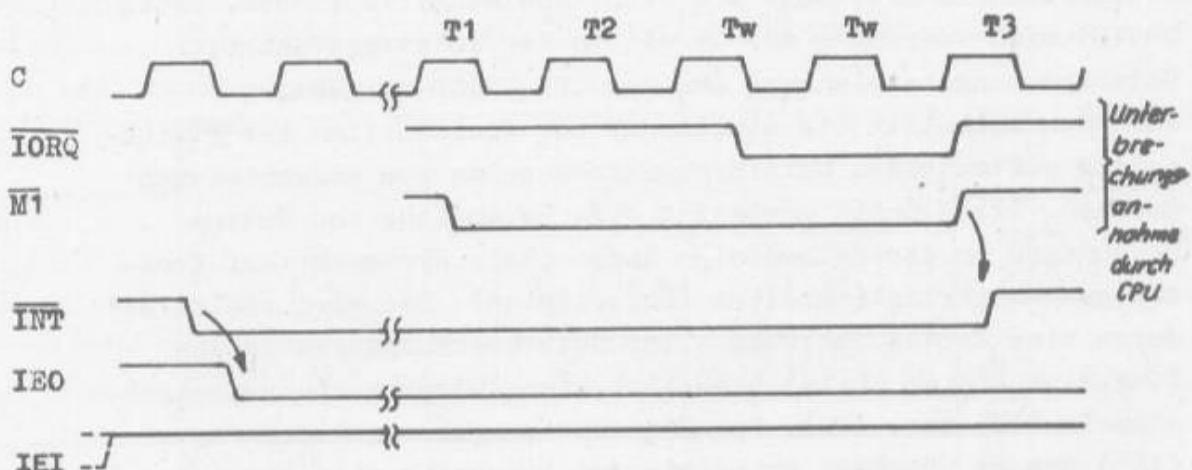
Außerdem wird vor Eintritt in das Behandlungsprogramm der alte Programmzähler im Kellerspeicher abgelegt.

CPU U880D Unterbrechungsverarbeitung

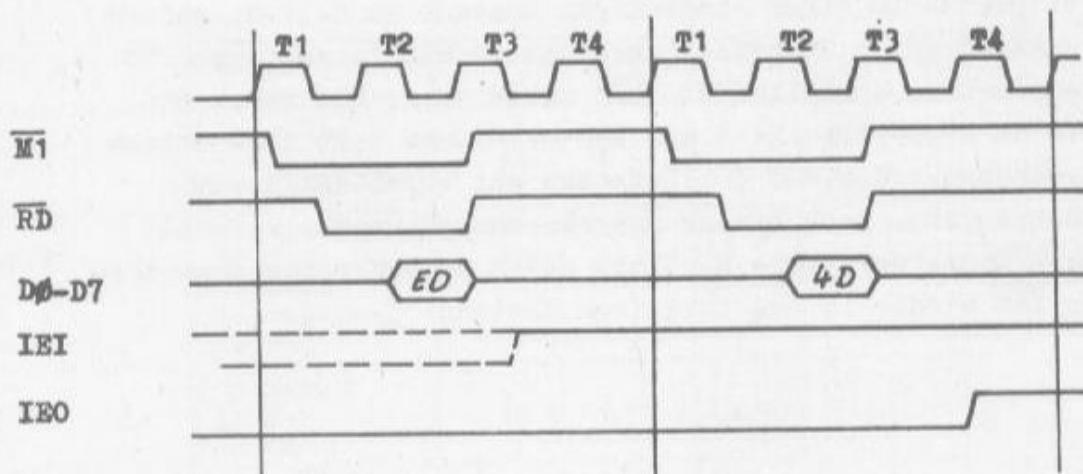
Die peripheren Bausteine der Schaltkreisfamilie (U855D, U856D, U857D) sind vorrangig zur Arbeit in der leistungsfähigsten Unterbrechungsbetriebsart IM2 der CPU U880D vorgesehen.

Außerdem enthalten sie eine Logik zur Koordination bei gleichzeitig auftretenden Unterbrechungswünschen aus verschiedenen Quellen. Diese Logik gestattet eine Behandlung von Unterbrechungen in der Reihenfolge ihrer (beim Systementwurf festzulegenden) Dringlichkeiten (Prioritäten). Sie wird realisiert durch eine Kettenschaltung aller unterbrechungserzeugenden Bausteine (Daisy chain) bezüglich eines Unterbrechungsfreigabe-signals (IEI bzw. IEO). Der Unterbrechungsfreigabeeingang (IEI) des am höchsten priorisierten Bausteins wird mit 1-Pegel (+5V) verbunden, d.h. der Baustein kann zu jeder Zeit Unterbrechungen anmelden. Wenn dieser Baustein eine Unterbrechung anmeldet, legt er seinem Unterbrechungsfreigabeausgang (IEO) auf 0-Pegel. Alle folgenden Bausteine in der Kette schalten diesen 0-Pegel von IEI zu IEO durch mit dem Ergebnis, daß an den Unterbrechungsfreigabeeingängen (IEI) aller restlichen Bausteine Ø-Pegel liegt und ihre Unterbrechungen dadurch gesperrt sind.

Um während eines Unterbrechungs-Annahmezyklus der CPU diese Prioritätstskette in einem eindeutigen Zustand zu halten, ändern die Bausteine ihren Unterbrechungszustand nicht mehr, wenn \bar{M}_1 aktiv wird. Wenn anschließend \overline{IORQ} aktiv wird, hat genau ein Baustein in der Kette $IEI=1$ und $IEO=0$. Dieser legt dann seinen Unterbrechungsvektor auf den Datenbus und veranlaßt damit seine Behandlung. Nach dieser Unterbrechungsannahme versetzt der betreffende periphere Baustein seine Unterbrechungsanmeldeleitung \overline{INT} wieder in den inaktiven Zustand.

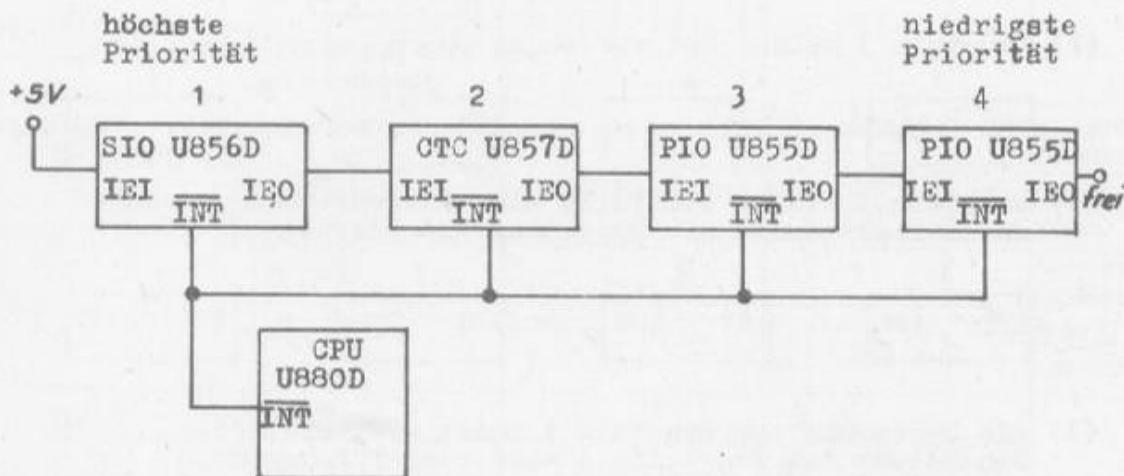


Am Ende einer Unterbrechungsbehandlung muß die Unterbrechung für niedriger priorisierte Bausteine wieder freigegeben werden, das geschieht dadurch, daß der in Behandlung befindliche Baustein (erkennbar an $IEI=1$ und $IEO=0$) den Befehl RETI (ED4D) dekodiert und daraufhin seinen Unterbrechungsfreigabeausgang wieder auf 1 setzt. Damit werden Unterbrechungen durch niedriger priorisierte Bausteine wieder gestattet.



CPU U880D Unterbrechungsverarbeitung

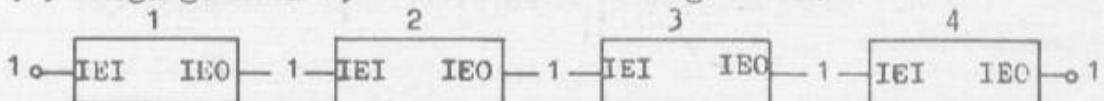
Das folgende Bild zeigt ein Beispiel für die Anordnung der Bausteine:



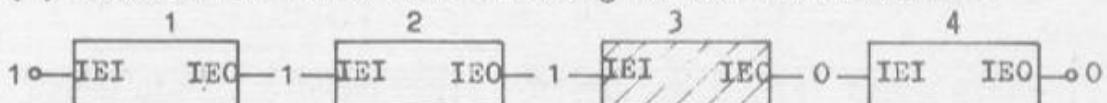
Ein möglicher Ablauf ist der, daß zuerst der Baustein 3 eine Unterbrechung auslöst und die CPU mit deren Behandlung begonnen und erneute Unterbrechungen wieder erlaubt hat.

Zu diesem Zeitpunkt meldet der höher priorisierte Baustein 1 eine Unterbrechung an. Die Behandlung des Bausteins 3 wird dadurch unterbrochen, der Baustein 1 wird behandelt, bis ein RETI-Befehl das Ende von dessen Unterbrechungsbehandlungsprogramm anzeigt. Daraufhin kann das den Baustein 3 behandelnde Programm weitergeführt werden, bis dessen abschließendes RETI das Behandlungsende anzeigt. Das folgende Bild zeigt die Arbeit der Prioritätskette bei dem beschriebenen Ablauf:

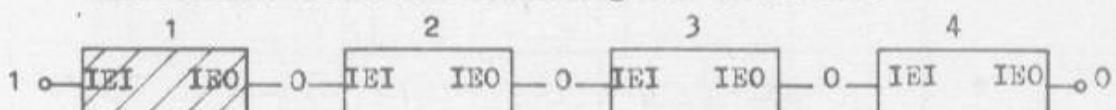
(1) Ausgangszustand, keine Unterbrechung wirksam



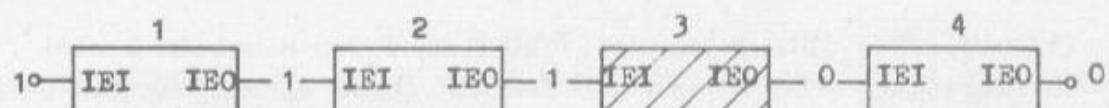
(2) Baustein 3 meldet Unterbrechung an und wird behandelt



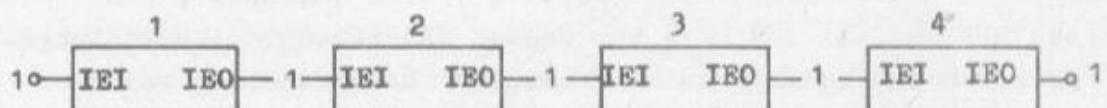
(3) Baustein 1 meldet ebenfalls eine Unterbrechung an und unterbricht damit die Bedienung von Baustein 3



(3) Die Bedienung von Baustein 1 endet mit RETI, die Behandlung des Bausteins 3 wird wieder aufgenommen



(4) Die Behandlung des Bausteins 3 endet mit RETI, es steht keine Unterbrechung mehr an



Da sowohl bei der Unterbrechungsannahme als auch beim RETI-Befehl nur eine begrenzte Zeit zur Verfügung steht, ist die Anzahl der Bausteine in der Kette begrenzt. Für mehr als 4 zu verkettende Bausteine ist eine externe Logik zur Beschleunigung des Kettendurchlaufs erforderlich, diese muß sowohl für die Unterbrechungsannahme als auch bei der Ausführung des RETI-Befehls wirksam sein.

CPU U880D Befehlsformate

• 1-Byte-Befehle

Operationskode

Beispiele: ADD (HL)
LD D,H

• 2-Byte-Befehle

Operationskode 1

DD: Indexregister IX behandeln
FD: Indexregister IY behandeln
CB: Bit- und Rotationsbefehle
ED: Ein-/Ausgaben, Blockbefehle,
16-Bit-Verarbeitung u.a.

Operationskode 2

Beispiele: INC IX
BIT 3,C
IM2

oder

Operationskode

n

Beispiele: LD C,0FFH
CP 20H
JR 05

• 3-Byte-Befehle

Operationskode 1

DD: Indizierte Adressierung
mit IX

Operationskode 2

FD: Indizierte Adressierung
mit IY

d

Beispiel: ADD (IY+5),
Versatz LD (IX+10),B

oder

Operationskode

n

n

niederwertiger Teil eines
16-Bit-Wertes

höherwertiger Teil eines
16-Bit-Wertes

Beispiel: LD A,(4000H)
JP 800H

• 4-Byte-Befehle

Operationskode 1

DD: Indizierte Adressierung
mit IX

Operationskode 2

FD: Indizierte Adressierung
mit IY

d

CB: Bit-u.Rotationsbefehle

Operationskode 3

Versatz
Beispiele: RR (IY+2),
SET 1,(IX-2)

oder

Operationskode 1

DD: Indizierte Adressierung
mit IX

Operationskode 2

FD: Indizierte Adressierung
mit IY

d

Beispiel: LD (IY+3),10

n

Versatz

oder

Operationskode 1

DD: Indexregister IX behandeln
FD: Indexregister IY behandeln

Operationskode 2

ED: 16-Bit-Ladebefehle

n

niederwertiger Teil eines
16-Bit-Wertes

n

höherwertiger Teil eines
16-Bit-Wertes

Beispiele: LD IX,500
LD (1000H),DE

Es folgt die
nach Befehlsgruppen geordnete Befehlsliste mit binärer Kodierung,
Flagbeeinflussung, Taktzyklenangabe, Funktionssymbolik

Erläuterung der verwendeten Symbolik:

r, r ₁ , r ₂	8-Bit-Register: A, B, C, D, E, H, L
dd	16-Bit-Register: BC, DE, HL, SP
qq	16-Bit-Register: BC, DE, HL, AF
rr	16-Bit-Register: BC, DE, IX, SP
ss	16-Bit-Register: BC, DE, IY, SP
cc	Bedingungskode: NZ ungleich Null Z gleich Null NC kein Übertrag C Übertrag PO ungerade Parität bzw. kein Überlauf PE gerade Parität bzw. Überlauf P positiv {Bit 7=0} N negativ {Bit 7=1}

H,L als Index höherwertiger bzw. niederwertiger Teil eines
16-Bit-Registers

n	8-Bit-Wert: 0...255
nn	16-Bit-Wert: 0...65535
d	8-Bit-Abstandswert bei indizierter Adressierung (IX,IY) in Zweierkomplementdarstellung: -128...+127
e	8-Bit-Abstandswert bei relativen Sprüngen in Zweier- komplementdarstellung: -126...+129
Achtung:	Im Operationskode ist e-2 einzusetzen, da PC vor der Addition mit e bereits intern um 2 erhöht wurde.
p	Kurzrufadresse (00H, 08H, 10H, 18H, 20H, 28H, 30H, 38H)
s _b	Bit eines Registers oder einer Speicherstelle: b = 0, 1, 2, 3, 4, 5, 6, 7
()	Ausdruck zwischen den Klammern stellt einen Zeiger (Adresse) auf eine Speicherstelle oder E/A-Port dar Anm.: Statt (HL) kann auch M stehen

Erläuterungen zum Flagsatz

Flag Bedeutung

- CY Übertrag: CY=1, wenn Operation einen Übertrag erzeugt bzw.
bei Verschiebeop. eine 1 herausgeschoben
wurde
- Z Null: Z=1, wenn Ergebnis einer Operation gleich Null,
bei Blocksuchbefehlen: Übereinstimmung
gefunden
- S Vorzeichen: S=1, wenn beim Ergebnis einer Operation Bit7=1
- P/V Parität: P=1, wenn bei logischen Operationen gerade
Parität (Anzahl der 1-Bits)
- Überlauf: V=1, wenn bei arithmetischen Operationen in der
Zweierkomplementdarstellung der zulässige
Bereich -128...+127 verlassen wurde
bei Blocktransfer- und Blocksuchbefehlen:
P/V=1, wenn BC noch nicht Null ist
- bei LD A,I/LD A,R: P/V:=-IFF2 (Unterbrechungserlaubnis-Flip-
flop)
- H Halbbyte-Übertrag: H=1, wenn Übertrag vom niedrigerwertigen
in das höherwertige Halbbyte auftritt
- N Addition/Subtraktion: N=1, wenn vorangegangene Operation
eine Subtraktion war

Symbolik der Flagbeeinflussung

- † Flag wird entsprechend dem Ergebnis der Operation gestellt
- Flag wird durch die Operation nicht beeinflußt
- Flag wird durch die Operation gelöscht
- 1 Flag wird durch die Operation gesetzt
- x Flag unbestimmt

Besonderheiten:

- a P/V-Flag: =0, wenn BC Null erreicht hat, sonst P/V:=1
- b Z-Flag: =1, wenn A= (HL), sonst P/V:=1
- c Z-Flag: =1, wenn B Null erreicht hat, sonst Z:=0

Aufbau des Flag-Bytes:

D7	D6	D5	D4	D3	D2	D1	D0
S	Z	X	H	X	P/V	N	CY

X - D3 und D5 enthalten keine Flagbedeutung

- 18 -
CPU U880D Befehle, binär

Assembler- sprache	symbolische Operation	Flags CY Z P/V S N H	Op.-Kode 76 543 210	Tak- te	Bemerkungen
8 - Bit - Ladebefehle					
LD r ₁ , r ₂	r ₁ := r ₂	•	01 r ₁ r ₂	4	r ₁ , r ₂ , Register r
LD r,n	r := n	•	00 r - n -	7	
LD r, (HL)	r := (HL)	•	01 r 110	7	
LD r,(IX+d)	r := (IX+d)	•	11 011 101 01 r 110 - d -	19	
LD r,(IY+d)	r := (IY+d)	•	11 111 101 01 r 110 - d -	19	
LD (HL),r	(HL) := r	•	01 110 r	7	
LD (IX+d),r	(IX+d) := r	•	11 011 101 01 110 r - d -	19	
LD (IY+d),r	(IY+d) := r	•	11 111 101 01 110 r - d -	19	
LD (HL),n	(HL) := n	•	00 110 110 - n -	10	
LD (IX+d),n	(IX+d) := n	•	11 011 101 00 110 110 - d - - n -	19	
LD (IY+d),n	(IY+d) := n	•	11 111 101 00 110 110 - d - - n -	19	
LD A,(BC)	A := (BC)	•	00 001 010	7	
LD A,(DE)	A := (DE)	•	00 011 010	7	
LD A, (nn)	A := (nn)	•	00 111 010 - n - - n -	13	
LD (BC),A	(BC) := A	•	00 000 010	7	
LD (DE),A	(DE) := A	•	00 010 010	7	
LD (nn),A	(nn) := A	•	00 110 010 - n - - n -	13	
LD A,I	A := I	. # IFF # 0 0	11 101 101 01 010 111	9	
LD A,R	A := R	. # IFF # 0 0	11 101 101 01 011 111	9	
LD I,A	I := A	•	11 101 101 01 000 111	9	
LD R,A	R := A	•	11 101 101 01 001 111	9	

Assembler- sprache	symbolische Operation	Flags CY Z P/V S N H	Op.-Kode 76 543 210	Tak- te	Bemerkungen
-----------------------	--------------------------	-------------------------	------------------------	------------	-------------

16-Bit-Ladebefehle

LD dd,nn	dd := nn	00 dd0 001 - n - - n -	10	dd	Register
LD IX,nn	IX := nn	11 011 101 00 100 001 - n - - n -	14	00 01 10 11	BC DE HL SP
LD IY,nn	IY := nn	11 111 101 00 100 001 - n - - n -	14		
LD HL,(nn)	H := (nn+1) L := (nn)	00 101 010 - n - - n -	16		
LD dd,(nn)	dd _H := (nn+1) dd _L := (nn)	11 101 101 01 dd1 011 - n - - n -	20		
LD IX,(nn)	IX _H := (nn+1) IX _L := (nn)	11 011 101 00 101 010 - n - - n -	20		
LD IY,(nn)	IY _H := (nn+1) IY _L := (nn)	11 111 101 00 101 010 - n - - n -	20		
LD (nn),HL	(nn+1) := H (nn) := L	00 100 010 - n - - n -	16		
LD (nn),dd	(nn+1) := dd _H (nn) := dd _L	11 101 101 01 dd0 011 - n - - n -	20		
LD (nn),IX	(nn+1) := IX _H (nn) := IX _D	11 011 101 00 100 010 - n - - n -	20		
LD (nn),IY	(nn+1) := IY _H (nn) := IY _L	11 111 101 00 100 010 - n - - n -	20		
LD SP,HL	SP := HL	11 111 001	6		
LD SP,IX	SP := IX	11 011 101 11 111 001	10		
LD SP,IY	SP := IY	11 111 101 11 111 001	10		

Assembler-sprache	symbolische Operation	Flags CY Z P/V S N H	Op.-Kode 76 543 210	Tak-te	Bemerkungen
PUSH qq	{SP-2} := qq_L {SP-1} := qq_H SP := SP-2	11 qq0 101	11	qq Registe
PUSH IX	{SP-2} := IX_L {SP-1} := IX_H SP := SP-2	11 011 101 11 100 101	15	00 BC 01 DE
PUSH IY	{SP-2} := IY_L {SP-1} := IY_H SP := SP-2	11 111 101 11 100 101	15	10 HL 11 AP
POP qq	qq_L := (SP) qq_H := (SP+1) SP := SP+2	11 qq0 001	10	
POP IX	IX_L := (SP) IX_H := (SP+1) SP := SP+2	11 011 101 11 100 001	14	
POP IY	IY_L := (SP) IY_H := (SP+1) SP := SP+2	11 111 101 11 100 001	14	

Bit testen, setzen, rücksetzen

BIT b,r	Z := \overline{r}_b	. # x x 0 1 01 b r	11 001 011 01 b 110	8	r Register
BIT b, (HL)	Z := $(\overline{HL})_b$. # x x 0 1 01 b 110	11 001 011 01 b 110	12	000 B 001 C 010 D
BIT b, (IX+d)	Z := $(\overline{IX+d})_b$. # x x 0 1 01 b 110	11 011 101 11 001 011 - d - 01 b 110	20	011 E 100 H 101 L 111 A
BIT b, (IY+d)	Z := $(\overline{IY+d})_b$. # x x 0 1 01 b 110	11 111 101 11 001 011 - d - 01 b 110	20	b Bit 000 0
SET b, r	r_b := 1	11 001 011 11 b r	8	001 1 010 2 011 3
SET b, (HL)	(HL)_b := 1	11 001 011 11 b 110	15	100 4 101 5 110 6 111 7
SET b, (IX+d)	(IX+d)_b := 1	11 011 101 11 001 011 - d - 11 b 110	23	
SET b, (IY+d)	(IY+d)_b := 1	11 111 101 11 001 011 - d - 11 b 110	23	
RES b,s	S_b := 0 S = r, (HL), (IX+d), (IY+d)		10		Zur Bildung des Op.-Kodes 11 in SET durch 10 ersetzen, Flags und Zeiten wie bei SET

Assembler- sprache	symbolische Operation	Flags CY Z P/V S N H	Op.-Kode 76 543 210	Tak- te	Bemerkungen
Austausch-, Blocktransfer- und Blocksuchbefehle					
EX DE,HL	DE \leftrightarrow HL	11 101 011	4	
EXAF	AF \leftrightarrow AF'	00 001 000	4	
EXX	BC \leftrightarrow BC' DE \leftrightarrow DE' HL \leftrightarrow HL'	11 011 001	4	Vertauschung der Register- sätze
EX (SP),HL	H \leftrightarrow (SP+1) L \leftrightarrow (SP)	11 100 011	19	
EX (SP),IX	IX _H \leftrightarrow (SP+1) IX _L \leftrightarrow (SP)	11 011 101 11 100 011	23	
EX (SP),IY	IY _H \leftrightarrow (SP+1) IY _L \leftrightarrow (SP)	11 111 101 11 100 011	23	
LDI	(DE) := (HL) DE := DE + 1 HL := HL + 1 BC := BC - 1	. x { a x 0 0	11 101 101 10 100 000	16	
LDIR	(DE) := (HL) DE := DE + 1 HL := HL + 1 BC := BC - 1 Wiederholung bis BC=0	. x 0 x 0 0	11 101 101 10 110 000	21 16	wenn BC \neq 0 wenn BC = 0
LDI	(DE) := (HL) DE := DE - 1 HL := HL - 1 BC := BC - 1	. x { a x 0 0	11 101 101 10 101 000	16	
LDDR	(DE) := (HL) DE := DE - 1 HL := HL - 1 BC := BC - 1 Wiederholung bis BC=0	. x 0 x 0 0	11 101 101 10 111 000	21 16	wenn BC \neq 0 wenn BC = 0
CPI	A - (HL) HL := HL + 1 BC := BC - 1	. { b { a { 1 {	11 101 101 10 100 001	16	
CPIR	A - (HL) HL := HL + 1 BC := BC - 1 Wiederholung bis BC=0 oder A = (HL)	. { b { a { 1 {	11 101 101 10 110 001	21 16	wenn BC \neq 0 wenn BC = 0
CPD	A - (HL) HL := HL - 1 BC := BC - 1	. { b { a { 1 {	11 101 101 10 101 001	16	
CPDR	A - (HL) HL := HL - 1 BC := BC - 1 Wiederholung bis BC=0 oder A = (HL)	. { b { a { 1 {	11 101 101 10 111 001	21 16	wenn BC \neq 0 wenn BC = 0

CPU US80D Befehle, binär

Assembler Sprache	symbolische Operation	Flags CY Z P/V S N H	Op.-Kode 76 543 210te	Tak- te	Bemerkungen
<u>8-Bit-Arithmetik und Logik</u>					
ADD r	A := A+r	V 0	10 000	r	4
ADD n	A := A+n	V 0	11 000	110 - n -	7
ADD (HL)	A := A+(HL)	V 0	10 000	110	7
ADD (IX+d)	A := A+(IX+d)	V 0	11 011 101		
			10 000	110 - d -	19
ADD (IY+d)	A := A+(IY+d)	V 0	11 111 101		
			10 000	110 - d -	19
ADC s	A := A+s+CY	V 0	001		
SUB s	A := A-s	V 1	010		s ist r,n, (HL) {IX+d} oder {IY+d} wie bei ADD.
SBC s	A := A-s-CY	V 1	011		
AND s	A := A&s	0 P 0 1	100		umrandete Bits ersetzen 000 in ADD
OR s	A := A s	0 P 0 0	110		
XOR s	A := A^s	0 P 0 0	101		
CMP s	A - s	V 1	111		
INC r	r := r+1	. V 0	00 r 100	4	
INC (HL)	(HL) := (HL)+1	. V 0	00 110 100	11	
INC (IX+d)	(IX+d) := (IX+d)+1	. V 0	11 011 101		
			00 110 100	- d -	23
INC (IY+d)	(IY+d) := (IY+d)+1	. V 0	11 111 101		
			00 110 100	- d -	23
DEC t	t := t-1	. V 1		101	t ist r,(HL), (IX+d) oder (IY+d) wie bei INC. 100 durch 101 im Op.- kode ersetzen

Assembler- sprache	symbolische Operation	Flags CY Z P/V C N H	Op.-Kode 76 543 210	Tak- te	Bemerkungen
-----------------------	--------------------------	-------------------------	------------------------	------------	-------------

Allgemeine Arithmetik und ZVE-Steuerbefehle

DAS	Korrektur von A nach Add./Sub. gepackter BCD-Zahlen	↑ ↓ P ↓ . ↓	00 100 111	4	
CPL	A := \bar{A} 1 1	00 101 111	4	bitweise Negation Einerkomplement
NEG	A := $\bar{A} + 1$	↓ ↓ V ↓ 1 ↓	11 101 101 01 000 100	8	Zweierkomple- ment
CCF	CY := \bar{CY}	↓ 0 X	00 111 111	4	Komplement des Übertrags-Flags
SCF	CY := 1	1 0 0	00 110 111	4	Setzen des Übertrags-Flags
NOP	keine Op.	00 000 000	4	
HALT	ZVE in HALT	01 110 110	4	
DI	IFFF := 0	11 110 011	4	Int sperren
EI	IFFF := 1	11 111 011	4	Int erlauben
IMO	Setzen Int.-Mode 0	11 101 101 01 000 110	8	
IM1	Setzen Int.-Mode 1	11 101 101 01 010 110	8	
IM2	Setzen Int.-Mode 2	11 101 101 01 011 110	8	

16-Bit-Arithmetik

ADD HL,dd	HL := HL+dd	↓ . . . 0 x	00 dd1 001	11	dd Register
ADC HL,dd	HL := HL+dd+CY	↓ ↓ V ↓ 0 x	11 101 101 01 dd1 010	15	
SBC HL,dd	HL := HL-dd-CY	↓ ↓ V ↓ 1 x	11 101 101 01 dd0 010	15	
ADD IX,rr	IX := IX+rr	↓ . . . 0 x	11 011 101 00 rr1 001	15	rr Register
ADD IY,ss	IY := IY+ss	↓ . . . 0 x	11 111 101 00 ss1 001	15	
INC dd	dd := dd+1	00 dd0 011	6	
INC IX	IX := IX+1	11 011 101 00 100 011	10	
INC IY	IY := IY+1	11 111 101 00 100 011	10	
DEC dd	dd := dd-1	00 dd1 011	6	
DEC IX	IX := IX-1	11 011 101 00 101 011	10	
DEC IY	IY := IY-1	11 111 101 00 101 011	10	

Assembler- sprache	Symbolische Operation	Flags CY Z P/V S NH	Op.-Kode 76 543 210	Tak- te	Bemerkungen
Verschiebe- und Rotationsbefehle					
RLCA		↓ . . . 0	00 000 111	4	
RLA		↓ . . . 0	00 010 111	4	
RRCA		↓ 0	00 001 111	4	
RRA		↓ 0	00 011 111	4	
RLC r		↓ ↓ P ↓ 0 0	11 001 011 00 000 r	8	r Register
RLC (HL)		↓ ↓ P ↓ 0 0	11 001 011 00 000 110	15	
RLC (IX+d)		↓ ↓ P ↓ 0 0	11 011 101 11 001 011 - d - 00 000 110	23	
RLC (IY+d)		↓ ↓ P ↓ 0 0	11 111 101 11 001 011 - d - 00 000 110	23	
RL t		↓ ↓ P ↓ 0 0	010		t ist r, (HL), (IX+d), (IY+d) wie bei RLC um- randete Bits ersetzen 000 in RLC
RRC t		↓ ↓ P ↓ 0 0	001		
RR t		↓ ↓ P ↓ 0 0	011		
SLA t		↓ ↓ P ↓ 0 0	100		
SRA t		↓ ↓ P ↓ 0 0	101		
SRL t		↓ ↓ P ↓ 0 0	111		
RLD		. ↓ P ↓ 0 0	11 101 101 01 101 111	18	Verschiebung um eine Dezi- malstelle zwischen A und (HL)
RRD		. ↓ P ↓ 0 0	11 101 101 01 100 111	18	obere Hälfte von unbe- einflußt

Assembler- sprache	Symbolische Operation	Flags CY Z P/V S N H	Op.-Kode 76 543 210	Tak- te	Bemerkungen
Sprünge, Unterprogrammaufrufe, Rücksprünge					
JMP nn	PC := nn	11 000 011 - n - - n -	10	cc Beding. 000 NZ nicht Null 001 Z Null 010 NC kein Über. 011 C Übertr. 100 PO ungerade 101 PE gerade 110 P positiv 111 " negativ
JPcc	wenn cc erfüllt dann PC := nn	11 cc 010 - n - - n -	10	Beding. nicht erf. Beding. erfüllt
JR e	PC := PC + e	00 011 000 - e-2 -	12	Beding. nicht erf. Beding. erfüllt
JRC e	wenn CY = 1, dann PC := PC + e	00 111 000 - e-2 -	7 12	Beding. nicht erf. Beding. erfüllt
JRNC e	wenn CY = 0, dann PC := PC + e	00 110 000 - e-2 -	7	Beding. nicht erf.
JRZ e	wenn Z = 1, dann PC := PC + e	00 101 000 - e-2 -	7 12	Beding. nicht erf. Beding. erfüllt
JRNZ e	wenn Z = 0 dann PC := PC + e	00 100 000 - e-2 -	7 12	Beding. nicht erf. Beding. erfüllt
JMP (HL)	PC := HL	11 101 001	4	
JMP (IX)	PC := IX	11 011 101 11 101 001	8	
JMP (IY)	PC := IY	11 111 101 11 101 001	8	
DjNZ e	B := B - 1 wenn B = 0, dann PC := PC + e	00 010 000 - e-2 -	8 13	wenn B = 0 wenn B = 1
CALL nn	(SP-1) := PC _H (SP-2) := PC _L PC := nn SP := SP-2	11 001 101 - n - - n -	17	
CAcc nn	wenn cc erfüllt, dann wie CALL nn	11 cc 100 - n - - n -	10 17	wenn cc nicht erfüllt wenn cc er- füllt
RET	PC _L := (SP) PC _H := (SP+1) SP := SP+2	11 001 001	10	
Rcc	wenn cc erfüllt, dann wie RET	11 cc 000	5 11	wenn cc nicht erfüllt wenn cc er- füllt
RETI	Rücksprung von Interrupt	11 101 101 01 001 101	14	
RETN	Rücksprung vom nicht maskier- baren Interrupt	11 101 101 01 000 101	14	
RST p	(SP-1) := PC _H (SP-2) := PC _L PC := p SP := p-2	11 u 111	11	u P 000 00H 001 08H 010 10H 011 18H 100 20H 101 28H 110 30H 111 38H

Assembler- sprache	Symbolische Operation	Flags CY Z P/V S N H	Op.-Kode 76 543 210	Tak- te	Bemerkungen
Ein- und Ausgabebefehle					
IN n	A := (n)	11 011 011 - n -	11	bei IN n: n an A0-A7 A an A8-A15
IN r	r := (C)	. { P } O }	11 101 101 01 r 000	12	sonst: C an A0-A7 B an A8-A15
INF	Flags stellen entsprechend(C)	. { P } O }	11 101 101 01 110 000	12	
INI	(HL) := (C) B := B-1 HL := HL+1	. {c x x 1 x}	11 101 101 10 100 010	16	
INIR	(HL) := (C) B := B-1 HL := HL+1 Wiederholung bis B=0	. 1 x x 1 x	11 101 101 10 110 010	21 16	wenn B ≠ 0 wenn B = 0
IND	(HL) := (C) B := B-1 HL := HL-1	. {c x x 1 x}	11 101 101 10 101 010	16	
INDR	(HL) := (C) B := B-1 HL := HL-1 Wiederholung bis B=0	. {c x x 1 x}	11 101 101 10 111 010	21 16	wenn B ≠ 0 wenn B = 0
OUT n	(n) := A	11 010 011 - n -	11	bei OUT n: n an A0-A7 A an A8-A15
OUT r	(C) := r	11 101 101 01 r 001	12	sonst: C an A0-A7 B an A8-A15
OUTI	(C) := (HL) B := B-1 HL := HL+1	. {c x x 1 x}	11 101 101 10 100 011	16	
OTIR	(C) := (HL) B := B-1 HL := HL+1 Wiederholung bis B=0	. 1 x x 1 x	11 101 101 10 110 011	21 16	wenn B ≠ 0 wenn B = 0
OUTD	(C) := (HL) B := B-1 HL := HL-1	. {c x x 1 x}	11 101 101 10 101 011	16	
OTDR	(C) := (HL) B := B-1 HL := HL-1 Wiederholung bis B=0	. 1 x x 1 x	11 101 101 10 111 011	21 16	wenn B ≠ 0 wenn B = 0

Erläuterungen zu den Befehlslisten Hex.-Assembler, Assembler-Hex.:

nn Bytekonstante

vv Verschiebekonstante

hh höherwertiger Teil einer 16-Bit-Konstante

ll niedrigerwertiger Teil einer 16-Bit-Konstante

Anordnung nach alphabetischer Reihenfolge der Assemblerbefehle

8E	ADC	(HL)	A5	AND	L	CB61	BIT	4,C
DD8Evv	ADC	(IX+d)	E6nn	AND	n	CB62	BIT	4,D
FD8Evv	ADC	(IY+d)	CB46	BIT	0,(HL)	CB63	BIT	4,E
8F	ADC	A	DDCBvv46	BIT	0,(IX+d)	CB64	BIT	4,H
88	ADC	B	FDCBvv46	BIT	0,(IY+d)	CB65	BIT	4,L
89	ADC	C	CB47	BIT	0,A	CB6E	BIT	5,(HL)
8A	ADC	D	CB48	BIT	0,B	DDCBvv6E	BIT	5,(IX+d)
8B	ADC	E	CB49	BIT	0,C	FDCBvv6E	BIT	5,(IY+d)
8C	ADC	H	CB42	BIT	0,D	CB6F	BIT	5,A
8D	ADC	L	CB43	BIT	0,E	CB68	BIT	5,B
CE nn	ADC	n	CB44	BIT	0,H	CB69	BIT	5,C
ED4A	ADC	HL,BC	CB45	BIT	0,L	CB6A	BIT	5,D
ED5A	ADC	HL,DE	CB4E	BIT	1,(HL)	CB6B	BIT	5,E
ED6A	ADC	HL,HL	DDCBvv4E	BIT	1,(IX+d)	CB6C	BIT	5,H
ED7A	ADC	HL,SP	FDCBvv4E	BIT	1,(IY+d)	CB6D	BIT	5,L
86	ADD	(HL)	CB4F	BIT	1,A	CB76	BIT	6,(HL)
DD86vv	ADD	(IX+d)	CB48	BIT	1,B	DDCBvv76	BIT	6,(IX+d)
FD86vv	ADD	(IY+d)	CB49	BIT	1,C	FDCBvv76	BIT	6,(IY+d)
87	ADD	A	CB4A	BIT	1,D	CB77	BIT	6,A
88	ADD	B	CB4B	BIT	1,E	CB78	BIT	6,B
81	ADD	C	CB4C	BIT	1,H	CB71	BIT	6,C
82	ADD	D	CB4D	BIT	1,L	CB72	BIT	6,D
83	ADD	E	CB56	BIT	2,(HL)	CB73	BIT	6,E
84	ADD	H	DDCBvv56	BIT	2,(IX+d)	CB74	BIT	6,H
85	ADD	L	FDCBvv56	BIT	2,(IY+d)	CB75	BIT	6,L
CB nn	ADD	n	CB57	BIT	2,A	CB7E	BIT	7,(HL)
99	ADD	HL,BC	CB50	BIT	2,B	DDCBvv7E	BIT	7,(IX+d)
19	ADD	HL,DE	CB51	BIT	2,C	FDCBvv7E	BIT	7,(IY+d)
29	ADD	HL,HL	CB52	BIT	2,D	CB7F	BIT	7,A
39	ADD	HL,SP	CB53	BIT	2,E	CB78	BIT	7,B
DD99	ADD	IX,BC	CB54	BIT	2,H	CB79	BIT	7,C
DD19	ADD	IX,DE	CB55	BIT	2,L	CB7A	BIT	7,D
DD29	ADD	IX,IX	CB5E	BIT	3,(HL)	CB7B	BIT	7,E
DD39	ADD	IX,SP	DDCBvv5E	BIT	3,(IX+d)	CB7C	BIT	7,H
FD99	ADD	IY,BC	FDCBvv5E	BIT	3,(IY+d)	CB7D	BIT	7,L
FD19	ADD	IY,DE	CB5F	BIT	3,A	DC11hh	CAC	nn
FD29	ADD	IY,IY	CB58	BIT	3,B	CD11hh	CALL	nn
FD39	ADD	IY,SP	CB59	BIT	3,C	FC11hh	CAM	nn
A6	AND	(HL)	CB5A	BIT	3,D	D411hh	CANC	nn
DDA6vv	AND	(IX+d)	CB58	BIT	3,E	C411hh	CANZ	nn
FDA6vv	AND	(IY+d)	CB5C	BIT	3,H	F411hh	CAP	nn
A7	AND	A	CB5D	BIT	3,L	EC11hh	CAPE	nn
A8	AND	B	CB66	BIT	4,(HL)	E411hh	CAPO	nn
A1	AND	C	DDCBvv66	BIT	4,(IX+d)	CC11hh	CAZ	nn
A2	AND	D	FDCBvv66	BIT	4,(IY+d)	3F	CCF	
A3	AND	E	CB67	BIT	4,A	BE	CMP (HL)	
A4	AND	H	CB60	BIT	4,B	DDBEvv	CMP (IX+d)	

FDBEvv	CMP (IY+d)	34	INC (HL)	DD74vv	LD (IX+d),H
BF	CMP A	DD34vv	INC (IX+d)	DD75vv	LD (IX+d),L
B8	CMP B	FD34vv	INC (IY+d)	DD36vvnn	LD (IX+d),n
B9	CMP C	3C	INC A	FD77vv	LD (IY+d),A
BA	CMP D	04	INC B	FD78vv	LD (IY+d),B
BB	CMP E	03	INC BC	FD71vv	LD (IY+d),C
BC	CMP H	0C	INC C	FD72vv	LD (IY+d),D
BD	CMP L	14	INC D	FD73vv	LD (IY+d),E
FE nn	CMP n	13	INC DE	FD74vv	LD (IY+d),H
EDA9	CPD	1C	INC E	FD75vv	LD (IY+d),L
EDB9	CPDR	24	INC H	FD36vwnn	LD (IY+d),n
EDA1	CPI	23	INC HL	32llhh	LD (nn),A
ED81	CPIR	DD23	INC IX	ED43llhh	LD (nn),BC
2F	CPL	FD23	INC IY	ED53llhh	LD (nn),DE
27	DAS	2C	INC L	22llhh	LD (nn),HL
35	DEC (HL)	33	INC SP	DD22llhh	LD (nn),IX
DD35vv	DEC (IX+d)	EDAA	IND	FD22llhh	LD (nn),IY
FD35vv	DEC (IY+d)	EDBA	INDR	ED73llhh	LD (nn),SP
3D	DEC A	EDA2	INI	0A	LD A,(BC)
05	DEC B	EDB2	INIR	1A	LD A,(DE)
0B	DEC BC	E9	JMP (HL)	7E	LD A,(HL)
0D	DEC C	DDE9	JMP (IX)	DD7Evv	LD A,(IX+d)
15	DEC D	FDE9	JMP (IY)	FD7Evv	LD A,(IY+d)
18	DEC DE	C3llhh	JMP nn	3A11hh	LD A,(nn)
1D	DEC E	DA11hh	JPC nn	7F	LD A,A
25	DEC H	FA11hh	JPM nn	78	LD A,B
2B	DEC HL	D211hh	JPNc nn	79	LD A,C
DD2B	DEC IX	C211hh	JPNZ nn	7A	LD A,D
FD2B	DEC IY	F211hh	JPP nn	7B	LD A,E
2D	DEC L	EA11hh	JPPE nn	7C	LD A,H
3B	DEC SP	E211hh	JPP0 nn	ED57	LD A,I
F3	DI	CA11hh	JPZ nn	7D	LD A,L
1Φ vv	DJNZ e	38 vv	JRC e	3Enn	LD A,n
FB	EI	18 vv	JR e	ED5F	LD A,R
E3	EX (SP),HL	3Φ vv	JRNC e	46	LD B,(HL)
DDE3	EX (SP),IX	2Φ vv	JRNZ e	DD46vv	LD B,(IX+d)
FDE3	EX (SP),IY	28 vv	JRZ e	FD46vv	LD B,(IY+d)
Φ8	EXAF	02	LD (BC),A	47	LD B,A
EB	EX DE,HL	12	LD (DE),A	4Φ	LD B,B
D9	EXX	77	LD (HL),A	41	LD B,C
76	HALT	7Φ	LD (HL),B	42	LD B,D
ED46	IMΦ	71	LD (HL),C	43	LD B,E
ED56	IM1	72	LD (HL),D	44	LD B,H
ED5E	IM2	73	LD (HL),E	45	LD B,L
ED78	IN A	74	LD (HL),H	Φ6nn	LD B,n
ED4Φ	IN B	75	LD (HL),L	ED48llhh	LD BC,(nn)
ED48	IN C	36 nn	LD (HL),n	Φ111hh	LD BC,nn
ED5Φ	IN D	DD77vv	LD (IX+d),A	4E	LD C,(HL)
ED58	IN E	DD7Φvv	LD (IX+d),B	DD4EVV	LD C,(IX+d)
ED7Φ	INF	DD71vv	LD (IX+d),C	FD4EVV	LD C,(IY+d)
ED6Φ	IN H	DD72vv	LD (IX+d),D	4F	LD C,A
ED68	IN L	DD73vv	LD (IX+d),E	48	LD C,B
DB nn	IN n				

49	LD C,C	6F	LD L,A	D5	PUSH DE
4A	LD C,D	68	LD L,B	E5	PUSH HL
4B	LD C,E	69	LD L,C	DDE5	PUSH IX
4C	LD C,H	6A	LD L,D	FDE5	PUSH IY
4D	LD C,L	6B	LD L,E	D8	RC
ΦE nn	LD C,n	6C	LD L,H	CB86	RES Φ,(HL)
56	LD D,(HL)	6D	LD L,L	DDCBvv86	RES Φ,(IX+d)
DD56vv	LD D,(IX+d)	2E nn	LD L,n	FDCBvv86	RES Φ,(IY+d)
FD56vv	LD D,(IY+d)	ED4F	LD R,A	CB87	RES Φ,A
57	LD D,A	ED78llhh	LD SP,(nn)	CB80	RES Φ,B
58	LD D,B	F9	LD SP, HL	CB81	RES Φ,C
51	LD D,C	DDF9	LD SP, IX	CB82	RES Φ,D
52	LD D,D	FDF9	LD SP, IY	CB83	RES Φ,E
53	LD D,E	31 llhh	LD SP, nn	CB84	RES Φ,H
54	LD D,H	EDA8	LD0	CB85	RES Φ,L
55	LD D,L	EDB8	LD0R	CB8E	RES 1,(HL)
16 nn	LD D,n	EDAΦ	LDI	DDCBvv8E	RES 1,(IX+d)
ED58llhh	LD DE,(nn)	EDBΦ	LDIR	FDCBvv8E	RES 1,(IY+d)
11 llhh	LD DE, nn	ED44	NEG	CB8F	RES 1,A
5E	LD E,(HL)	ΦΦ	NOP	CB88	RES 1,B
DD5EVV	LD E,(IX+d)	B6	OR (HL)	CB89	RES 1,C
FD5EVV	LD E,(IY+d)	DD86vv	OR (IX+d)	CB8A	RES 1,D
5F	LD E,A	FD86vv	OR (IY+d)	CB8B	RES 1,E
58	LD E,B	B7	OR A	CB8C	RES 1,H
59	LD E,C	BΦ	OR B	CB8D	RES 1,L
5A	LD E,D	B1	OR C	CB96	RES 2,(HL)
5B	LD E,E	B2	OR D	DDCBvv96	RES 2,(IX+d)
5C	LD E,H	B3	OR E	FDCBvv96	RES 2,(IY+d)
5D	LD E,L	B4	OR H	CB97	RES 2,A
1E nn	LD E,n	B5	OR L	CB9Φ	RES 2,B
66	LD H,(HL)	F6 nn	OR n	CB91	RES 2,C
DD66vv	LD H,(IX+d)	ED88	OTDR	CB92	RES 2,D
FD66vv	LD H,(IY+d)	ED83	OTIR	CB93	RES 2,E
67	LD H,A	ED79	OUT A	CB94	RES 2,H
60	LD H,B	ED41	OUT B	CB95	RES 2,L
61	LD H,C	ED49	OUT C	CB9E	RES 3,(HL)
62	LD H,D	EDAB	OUTD	DDCBvv9E	RES 3,(IX+d)
63	LD H,E	ED51	OUT D	FDCBvv9E	RES 3,(IY+d)
64	LD H,H	ED59	OUT E	CB9F	RES 3,A
65	LD H,L	ED61	OUT H	CB98	RES 3,B
26 nn	LD H,n	EDA3	OUTI	CB99	RES 3,C
2A llhh	LD HL,(nn)	ED69	OUT L	CB9A	RES 3,D
21 llhh	LD HL, nn	D3 nn	OUT n	CB9B	RES 3,E
ED47	LD I,A	F1	POP AF	CB9C	RES 3,H
DD2A llhh	LD IX,(nn)	C1	POP BC	CB9D	RES 3,L
DD21 llhh	LD IX, nn	D1	POP DE	CBA6	RES 4,(HL)
FD2A llhh	LD IY,(nn)	E1	POP HL	DDCBvvA6	RES 4,(IX+d)
FD21 llhh	LD IY, nn	DDE1	POP IX	FDCBvvA6	RES 4,(IY+d)
6E	LD L,(HL)	FDE1	POP IY	CBA7	RES 4,A
DD6EVV	LD L,(IX+d)	F5	PUSH AF	CBAΦ	RES 4,B
FD6EVV	LD L,(IY+d)	C5	PUSH BC	CBA1	RES 4,C

CB A2	RES	4,D	CBΦ7	RLC	A	9B	SBC	E
CB A3	RES	4,E	CBΦ1	RLC	C	9C	SBC	H
CBA4	RES	4,H	CBΦ2	RLC	D	ED42	SBC	HL, BC
CBA5	RES	4,L	CBΦ3	RLC	E	ED52	SBC	HL, DE
CBAE	RES	5,(HL)	CBΦ4	RLC	H	ED62	SBC	HL, HL
DDCBvvAE	RES	5,(IX+d)	CBΦ5	RLC	L	ED72	SBC	HL, SP
FDCBvvAE	RES	5,(IY+d)	ED6F	RLD		9D	SBC	L
CBAF	RES	5,A	F8	RM		DE nn	SBC	n
CBA8	RES	5,B	DΦ	RNC		37	SCF	
CBA9	RES	5,C	CΦ	RNZ		CBC6	SET	0(HL)
CBAA	RES	5,D	FΦ	RP		DDCBvvC6	SET	0,(IX+d)
CBAB	RES	5,E	E8	RPE		FDCBvvC6	SET	0,(IY+d)
CBAC	RES	5,H	EΦ	RPO		CBC7	SET	0,A
CBAD	RES	5,L	CB1E	RR	(HL)	CBCΦ	SET	0,B
CBB6	RES	6,(HL)	DDCBvv1E	RR	(IX+d)	CBC1	SET	0,C
DDCBvvB6	RES	6,(IX+d)	FDCBvv1E	RR	(IY+d)	CBC2	SET	0,D
FDCBvvB6	RES	6,(IY+d)	1F	RRA		CBC3	SET	0,E
CBB7	RES	6,A	CB1F	RR	A	CBC4	SET	0,H
CBBΦ	RES	6,B	CB18	RR	B	CBC5	SET	0,L
CBB1	RES	6,C	CB19	RR	C	CBCE	SET	1,(HL)
CBB2	RES	6,D	CB1A	RR	D	DDCBvvCE	SET	1,(IX+d)
CBB3	RES	6,E	CB1B	RR	E	FDCBvvCE	SET	1,(IY+d)
CBB4	RES	6,H	CB1C	RR	H	CBCF	SET	1,A
CBB5	RES	6,L	CB1D	RR	L	CBC8	SET	1,B
CBBE	RES	7,(HL)	CBΦE	RRC	(HL)	CBC9	SET	1,C
DDCBvvBE	RES	7,(IX+d)	DDCBvvΦE	RRC	(IX+d)	CBCA	SET	1,D
FDCBvvBE	RES	7,(IY+d)	FDCBvvΦE	RRC	(IY+d)	CBCB	SET	1,E
CBBF	RES	7,A	ΦF	RRCA		CBCC	SET	1,H
CBB8	RES	7,B	CBΦF	RRC	A	CBCD	SET	1,L
CBB9	RES	7,C	CBΦ8	RRC	B	CB06	SET	2,(HL)
CBB4	RES	7,D	CBΦ9	RRC	C	DDCBvvD6	SET	2,(IX+d)
CBB8	RES	7,E	CBΦA	RRC	D	FDCBvvD6	SET	2,(IY+d)
CBBC	RES	7,H	CBΦB	RRC	E	CB07	SET	2,A
CBBD	RES	7,L	CBΦC	RRC	H	CBDΦ	SET	2,B
C9	RET		CBΦD	RRC	L	CBD1	SET	2,C
ED4D	RETI		ED67	RRD		CBD2	SET	2,D
ED45	RETN		C7	RST	ΦΦH	CBD3	SET	2,E
CB16	RL	(HL)	CF	RST	ΦΦH	CBD4	SET	2,H
DDCBvv16	RL	(IX+d)	D7	RST	1ΦH	CBD5	SET	2,L
FDCBvv16	RL	(IY+d)	DF	RST	18H	CBD6	SET	3,(HL)
17	RLA		E7	RST	2ΦH	DDCBvvDE	SET	3,(IX+d)
CB17	RL	A	EF	RST	28H	FDCBvvDE	SET	3,(IY+d)
CB1Φ	RL	B	F7	RST	3ΦH	CBDF	SET	3,A
CB11	RL	C	FF	RST	38H	CBDB	SET	3,B
CB12	RL	D	C8	RZ		CBD9	SET	3,C
CB13	RL	E	9E	SBC	(HL)	CBDA	SET	3,D
CB14	RL	H	DD9EvV	SBC	(IX+d)	CBDB	SET	3,E
CB15	RL	L	FD9EvV	SBC	(IY+d)	CBDC	SET	3,H
CBΦ6	RLC	(HL)	9F	SBC	A	CBDD	SET	3,L
DDCBvvΦ6	RLC	(IX+d)	98	SBC	B	CBE6	SET	4,(HL)
FDCBvvΦ6	RLC	(IY+d)	99	SBC	C	DDCBvvE6	SET	4,(IX+d)
Φ7	RLCA		9A	SBC	D	FDCBvvE6	SET	4,(IY+d)

CBE7	SET	4,A	CB29	SRA	C
CBE8	SET	4,B	CB2A	SRA	D
CBE1	SET	4,C	CB2B	SRA	E
CBE2	SET	4,D	CB2C	SRA	H
CBE3	SET	4,E	CB2D	SRA	L
CBE4	SET	4,H	CB3E	SRL	(HL)
CBE5	SET	4,L	DDCBvv3E	SRL	(IX+d)
CBE6	SET	5,(HL)	FDCBvv3E	SRL	(IY+d)
DDCBvvEE	SET	5,(IX+d)	CB3F	SRL	A
FDCBvvEE	SET	5,(IY+d)	CB38	SRL	B
CBEF	SET	5,A	CB39	SRL	C
CBE8	SET	5,B	CB3A	SRL	D
CBE9	SET	5,C	CB3B	SRL	E
CBEA	SET	5,D	CB3C	SRL	H
CBEB	SET	5,E	CB3D	SRL	L
CBEC	SET	5,H	96	SUB	(HL)
CBED	SET	5,L	DD96vv	SUB	(IX+d)
CBF6	SET	6,(HL)	FD96vv	SUB	(IY+d)
DDCBvvF6	SET	6,(IX+d)	97	SUB	A
FDCBvvF6	SET	6,(IY+d)	98	SUB	B
CBF7	SET	6,A	91	SUB	C
CBF8	SET	6,B	92	SUB	D
CBF1	SET	6,C	93	SUB	E
CBF2	SET	6,D	94	SUB	H
CBF3	SET	6,E	95	SUB	L
CBF4	SET	6,H	D6nn	SUB	n
CBF5	SET	6,L	AE	XOR	(HL)
CBFE	SET	7,(HL)	DDAEvv	XOR	(IX+d)
DDCBvvFE	SET	7,(IX+d)	FDAEvv	XOR	(IY+d)
FDCBvvFE	SET	7,(IY+d)	AF	XOR	A
CBFF	SET	7,A	A8	XOR	B
CBF8	SET	7,B	A9	XOR	C
CBF9	SET	7,C	AA	XOR	D
CBFA	SET	7,D	AB	XOR	E
CBFB	SET	7,E	AC	XOR	H
CBFC	SET	7,H	AD	XOR	L
CBFD	SET	7,L	EEnn	XOR	n
CB26	SLA	(HL)			
DDCBvv26	SLA	(IX+d)			
FDCBvv26	SLA	(IY+d)			
CB27	SLA	A			
CB28	SLA	B			
CB21	SLA	C			
CB22	SLA	D			
CB23	SLA	E			
CB24	SLA	H			
CB25	SLA	L			
CB2E	SRA	(HL)			
DDCBvv2E	SRA	(IX+d)			
FDCBvv2E	SRA	(IY+d)			
CB2F	SRA	A			
CB28	SRA	B			

Anordnung nach aufsteigenden Operationskodes

00	NOP	33	INC	SP	66	LD	H, (HL)
01 llhh	LD BC, nn	34	INC	(HL)	67	LD	H, A
02	LD (BC), A	35	DEC	(HL)	68	LD	L, B
03	INC BC	36 nn	LD	(HL), n	69	LD	L, C
04	INC B	37	SCF		6A	LD	L, D
05	DEC B	38 vv	JRC	e	6B	LD	L, E
06 nn	LD B, n	39	ADD	HL, SP	6C	LD	L, H
07	RLCA	3A llhh	LD	A, (nn)	6D	LD	L, L
08	EXAF	3B	DEC	SP	6E	LD	L, (HL)
09	ADD HL, BC	3C	INC	A	6F	LD	L, A
0A	LD A, (BC)	3D	DEC	A	70	LD	(HL), B
0B	DEC BC	3E nn	LD	A, n	71	LD	(HL), C
0C	INC C	3F	CCF		72	LD	(HL), D
0D	DEC C	40	LD	B, B	73	LD	(HL), E
0E nn	LD C, n	41	LD	B, C	74	LD	(HL), H
0F	RRCA	42	LD	B, D	75	LD	(HL), L
10 vv	DJNZ e	43	LD	B, E	76	HALT	
11 llhh	LD DE, nn	44	LD	B, H	77	LD	(HL), A
12	LD (DE), A	45	LD	B, L	78	LD	A, B
13	INC DE	46	LD	B, (HL)	79	LD	A, C
14	INC D	47	LD	B, A	7A	LD	A, D
15	DEC D	48	LD	C, B	7B	LD	A, E
16 nn	LD D, n	49	LD	C, C	7C	LD	A, H
17	RLA	4A	LD	C, D	7D	LD	A, L
18 vv	JR e	4B	LD	C, E	7E	LD	A, (HL)
19	ADD HL, DE	4C	LD	C, H	7F	LD	A, A
1A	LD A, (DE)	4D	LD	C, L	80	ADD	B
1B	DEC DE	4E	LD	C, (HL)	81	ADD	C
1C	INC E	4F	LD	C, A	82	ADD	D
1D	DEC E	50	LD	D, B	83	ADD	E
1E nn	LD E, n	51	LD	D, C	84	ADD	H
1F	RRA	52	LD	D, D	85	ADD	L
20 vv	JRNZ e	53	LD	D, E	86	ADD	(HL)
21 llhh	LD HL, nn	54	LD	D, H	87	ADD	A
22 llhh	LD (nn), HL	55	LD	D, L	88	ADC	B
23	INC HL	56	LD	D, (HL)	89	ADC	C
24	INC H	57	LD	D, A	8A	ADC	D
25	DEC H	58	LD	E, B	8B	ADC	E
26 nn	LD H, n	59	LD	E, C	8C	ADC	H
27	DAS	5A	LD	E, D	8D	ADC	L
28 vv	JRZ e	5B	LD	E, E	8E	ADC	(HL)
29	ADD HL, HL	5C	LD	E, H	8F	ADC	A
2A llhh	LD HL, (nn)	5D	LD	E, L	90	SUB	B
2B	DEC HL	5E	LD	E, (HL)	91	SUB	C
2C	INC L	5F	LD	E, A	92	SUB	D
2D	DEC L	60	LD	H, B	93	SUB	E
2E nn	LD L, n	61	LD	H, C	94	SUB	H
2F	CPL	62	LD	H, D	95	SUB	L
30 vv	JRNC e	63	LD	H, E	96	SUB	(HL)
31 llhh	LD SP, nn	64	LD	H, H	97	SUB	A
32 llhh	LD (nn), A	65	LD	H, L	98	SBC	B

CPU U880D Hex — Assembler

99	SBC	C	CB00	RLC	B	CB3B	SRL	E
9A	SBC	D	CB01	RLC	C	CB3C	SRL	H
9B	SBC	E	CB02	RLC	D	CB3D	SRL	L
9C	SBC	H	CB03	RLC	E	CB3E	SRL	(HL)
9D	SBC	L	CB04	RLC	H	CB3F	SRL	A
9E	SBC	(HL)	CB05	RLC	L	CB40	BIT	0,B
9F	SBC	A	CB06	RLC	(HL)	CB41	BIT	0,C
A0	AND	B	CB07	RLC	A	CB42	BIT	0,D
A1	AND	C	CB08	RRC	B	CB43	BIT	0,E
A2	AND	D	CB09	RRC	C	CB44	BIT	0,H
A3	AND	E	CB0A	RRC	D	CB45	BIT	0,L
A4	AND	H	CB0B	RRC	E	CB46	BIT	0,(HL)
A5	AND	L	CB0C	RRC	H	CB47	BIT	0,A
A6	AND	(HL)	CB0D	RRC	L	CB48	BIT	1,B
A7	AND	A	CB0E	RRC	(HL)	CB49	BIT	1,C
A8	XOR	B	CB0F	RRC	A	CB4A	BIT	1,D
A9	XOR	C	CB10	RL	B	CB4B	BIT	1,E
AA	XOR	D	CB11	RL	C	CB4C	BIT	1,H
AB	XOR	E	CB12	RL	D	CB4D	BIT	1,L
AC	XOR	H	CB13	RL	E	CB4E	BIT	1,(HL)
AD	XOR	L	CB14	RL	H	CB4F	BIT	1,A
AE	XOR	(HL)	CB15	RL	L	CB50	BIT	2,B
AF	XOR	A	CB16	RL	(HL)	CB51	BIT	2,C
B0	OR	B	CB17	RL	A	CB52	BIT	2,D
B1	OR	C	CB18	RR	B	CB53	BIT	2,E
B2	OR	D	CB19	RR	C	CB54	BIT	2,H
B3	OR	E	CB1A	RR	D	CB55	BIT	2,L
B4	OR	H	CB1B	RR	E	CB56	BIT	2,(HL)
B5	OR	L	CB1C	RR	H	CB57	BIT	2,A
B6	OR	(HL)	CB1D	RR	L	CB58	BIT	3,B
B7	OR	A	CB1E	RR	(HL)	CB59	BIT	3,C
B8	CMP	B	CB1F	RR	A	CB5A	BIT	3,D
B9	CMP	C	CB20	SLA	B	CB5B	BIT	3,E
BA	CMP	D	CB21	SLA	C	CB5C	BIT	3,H
BB	CMP	E	CB22	SLA	D	CB5D	BIT	3,L
BC	CMP	H	CB23	SLA	E	CB5E	BIT	3,(HL)
BD	CMP	L	CB24	SLA	H	CB5F	BIT	3,A
BE	CMP	(HL)	CB25	SLA	L	CB60	BIT	4,B
BF	CMP	A	CB26	SLA	(HL)	CB61	BIT	4,C
C0	RNZ		CB27	SLA	A	CB62	BIT	4,D
C1	POP	BC	CB28	SRA	B	CB63	BIT	4,E
C2 llhh	JPNZ	nn	CB29	SRA	C	CB64	BIT	4,H
C3 llhh	JMP	nn	CB2A	SRA	D	CB65	BIT	4,L
C4 llhh	CANZ	nn	CB2B	SRA	E	CB66	BIT	4,(HL)
C5	PUSH	BC	CB2C	SRA	H	CB67	BIT	4,A
C6 nn	ADD	n	CB2D	SRA	L	CB68	BIT	5,B
C7	RST	00H	CB2E	SRA	(HL)	CB69	BIT	5,C
C8	RZ		CB2F	SRA	A	CB6A	BIT	5,D
C9	RET		CB38	SRL	B	CB6B	BIT	5,E
CA llhh	JPZ	nn	CB39	SRL	C	CB6C	BIT	5,H
			CB3A	SRL	D	CB6D	BIT	5,L

<i>CB6E</i>	<i>BIT</i>	<i>5,(HL)</i>	<i>CB A1</i>	<i>RES</i>	<i>4,C</i>	<i>CB D4</i>	<i>SET</i>	<i>2,H</i>
<i>CB6F</i>	<i>BIT</i>	<i>5,A</i>	<i>CB A2</i>	<i>RES</i>	<i>4,D</i>	<i>CB D5</i>	<i>SET</i>	<i>2,L</i>
<i>CB70</i>	<i>BIT</i>	<i>6,B</i>	<i>CB A3</i>	<i>RES</i>	<i>4,E</i>	<i>CB D6</i>	<i>SET</i>	<i>2,(HL)</i>
<i>CB71</i>	<i>BIT</i>	<i>6,C</i>	<i>CB A4</i>	<i>RES</i>	<i>4,H</i>	<i>CB D7</i>	<i>SET</i>	<i>2,A</i>
<i>CB72</i>	<i>BIT</i>	<i>6,D</i>	<i>CB A5</i>	<i>RES</i>	<i>4,L</i>	<i>CB D8</i>	<i>SET</i>	<i>3,B</i>
<i>CB73</i>	<i>BIT</i>	<i>6,E</i>	<i>CB A6</i>	<i>RES</i>	<i>4,(HL)</i>	<i>CB D9</i>	<i>SET</i>	<i>3,C</i>
<i>CB74</i>	<i>BIT</i>	<i>6,H</i>	<i>CB A7</i>	<i>RES</i>	<i>4,A</i>	<i>CB DA</i>	<i>SET</i>	<i>3,0</i>
<i>CB75</i>	<i>BIT</i>	<i>6,L</i>	<i>CB A8</i>	<i>RES</i>	<i>5,B</i>	<i>CB DB</i>	<i>SET</i>	<i>3,E</i>
<i>CB76</i>	<i>BIT</i>	<i>6,(HL)</i>	<i>CB A9</i>	<i>RES</i>	<i>5,C</i>	<i>CB DC</i>	<i>SET</i>	<i>3,H</i>
<i>CB77</i>	<i>BIT</i>	<i>6,A</i>	<i>CB AA</i>	<i>RES</i>	<i>5,D</i>	<i>CB DD</i>	<i>SET</i>	<i>3,L</i>
<i>CB78</i>	<i>BIT</i>	<i>7,B</i>	<i>CB AB</i>	<i>RES</i>	<i>5,E</i>	<i>CB DE</i>	<i>SET</i>	<i>3,(HL)</i>
<i>CB79</i>	<i>BIT</i>	<i>7,C</i>	<i>CB AC</i>	<i>RES</i>	<i>5,H</i>	<i>CB DF</i>	<i>SET</i>	<i>3,A</i>
<i>CB7A</i>	<i>BIT</i>	<i>7,D</i>	<i>CB AD</i>	<i>RES</i>	<i>5,L</i>	<i>CB EΦ</i>	<i>SET</i>	<i>4,B</i>
<i>CB7B</i>	<i>BIT</i>	<i>7,E</i>	<i>CB AE</i>	<i>RES</i>	<i>5,(HL)</i>	<i>CB E1</i>	<i>SET</i>	<i>4,C</i>
<i>CB7C</i>	<i>BIT</i>	<i>7,H</i>	<i>CB AF</i>	<i>RES</i>	<i>5,A</i>	<i>CB E2</i>	<i>SET</i>	<i>4,D</i>
<i>CB7D</i>	<i>BIT</i>	<i>7,L</i>	<i>CB BΦ</i>	<i>RES</i>	<i>6,B</i>	<i>CB E3</i>	<i>SET</i>	<i>4,E</i>
<i>CB7E</i>	<i>BIT</i>	<i>7,(HL)</i>	<i>CB B1</i>	<i>RES</i>	<i>6,C</i>	<i>CB E4</i>	<i>SET</i>	<i>4,H</i>
<i>CB7F</i>	<i>BIT</i>	<i>7,A</i>	<i>CB B2</i>	<i>RES</i>	<i>6,D</i>	<i>CB E5</i>	<i>SET</i>	<i>4,L</i>
<i>CB8Φ</i>	<i>RES</i>	<i>Φ,B</i>	<i>CB B3</i>	<i>RES</i>	<i>6,E</i>	<i>CB E6</i>	<i>SET</i>	<i>4,(HL)</i>
<i>CB81</i>	<i>RES</i>	<i>Φ,C</i>	<i>CB B4</i>	<i>RES</i>	<i>6,H</i>	<i>CB E7</i>	<i>SET</i>	<i>4,A</i>
<i>CB82</i>	<i>RES</i>	<i>Φ,D</i>	<i>CB B5</i>	<i>RES</i>	<i>6,L</i>	<i>CB E8</i>	<i>SET</i>	<i>5,B</i>
<i>CB83</i>	<i>RES</i>	<i>Φ,E</i>	<i>CB B6</i>	<i>RES</i>	<i>6,(HL)</i>	<i>CB E9</i>	<i>SET</i>	<i>5,C</i>
<i>CB84</i>	<i>RES</i>	<i>Φ,H</i>	<i>CB B7</i>	<i>RES</i>	<i>6,A</i>	<i>CB EA</i>	<i>SET</i>	<i>5,D</i>
<i>CB85</i>	<i>RES</i>	<i>Φ,L</i>	<i>CB B8</i>	<i>RES</i>	<i>7,B</i>	<i>CB EB</i>	<i>SET</i>	<i>5,E</i>
<i>CB86</i>	<i>RES</i>	<i>Φ,(HL)</i>	<i>CB B9</i>	<i>RES</i>	<i>7,C</i>	<i>CB EC</i>	<i>SET</i>	<i>5,H</i>
<i>CB87</i>	<i>RES</i>	<i>Φ,A</i>	<i>CB BA</i>	<i>RES</i>	<i>7,D</i>	<i>CB ED</i>	<i>SET</i>	<i>5,L</i>
<i>CB88</i>	<i>RES</i>	<i>1,B</i>	<i>CB BB</i>	<i>RES</i>	<i>7,E</i>	<i>CB EE</i>	<i>SET</i>	<i>5,(HL)</i>
<i>CB89</i>	<i>RES</i>	<i>1,C</i>	<i>CB BC</i>	<i>RES</i>	<i>7,H</i>	<i>CB EF</i>	<i>SET</i>	<i>5,A</i>
<i>CB8A</i>	<i>RES</i>	<i>1,D</i>	<i>CB BD</i>	<i>RES</i>	<i>7,L</i>	<i>CB FΦ</i>	<i>SET</i>	<i>6,B</i>
<i>CB8B</i>	<i>RES</i>	<i>1,E</i>	<i>CB BE</i>	<i>RES</i>	<i>7,(HL)</i>	<i>CB F1</i>	<i>SET</i>	<i>6,C</i>
<i>CB8C</i>	<i>RES</i>	<i>1,H</i>	<i>CB BF</i>	<i>RES</i>	<i>7,A</i>	<i>CB F2</i>	<i>SET</i>	<i>6,D</i>
<i>CB8D</i>	<i>RES</i>	<i>1,L</i>	<i>CB CΦ</i>	<i>SET</i>	<i>Φ,B</i>	<i>CB F3</i>	<i>SET</i>	<i>6,E</i>
<i>CB8E</i>	<i>RES</i>	<i>1,(HL)</i>	<i>CB C1</i>	<i>SET</i>	<i>Φ,C</i>	<i>CB F4</i>	<i>SET</i>	<i>6,H</i>
<i>CB8F</i>	<i>RES</i>	<i>1,A</i>	<i>CB C2</i>	<i>SET</i>	<i>Φ,D</i>	<i>CB F5</i>	<i>SET</i>	<i>6,L</i>
<i>CB9Φ</i>	<i>RES</i>	<i>2,B</i>	<i>CB C3</i>	<i>SET</i>	<i>Φ,E</i>	<i>CB F6</i>	<i>SET</i>	<i>6,(HL)</i>
<i>CB91</i>	<i>RES</i>	<i>2,C</i>	<i>CB C4</i>	<i>SET</i>	<i>Φ,H</i>	<i>CB F7</i>	<i>SET</i>	<i>6,A</i>
<i>CB92</i>	<i>RES</i>	<i>2,D</i>	<i>CB C5</i>	<i>SET</i>	<i>Φ,L</i>	<i>CB F8</i>	<i>SET</i>	<i>7,B</i>
<i>CB93</i>	<i>RES</i>	<i>2,E</i>	<i>CB C6</i>	<i>SET</i>	<i>Φ,(HL)</i>	<i>CB F9</i>	<i>SET</i>	<i>7,C</i>
<i>CB94</i>	<i>RES</i>	<i>2,H</i>	<i>CB C7</i>	<i>SET</i>	<i>Φ,A</i>	<i>CB FA</i>	<i>SET</i>	<i>7,D</i>
<i>CB95</i>	<i>RES</i>	<i>2,L</i>	<i>CB C8</i>	<i>SET</i>	<i>1,B</i>	<i>CB FB</i>	<i>SET</i>	<i>7,E</i>
<i>CB96</i>	<i>RES</i>	<i>2,(HL)</i>	<i>CB C9</i>	<i>SET</i>	<i>1,C</i>	<i>CB FC</i>	<i>SET</i>	<i>7,H</i>
<i>CB97</i>	<i>RES</i>	<i>2,A</i>	<i>CB CA</i>	<i>SET</i>	<i>1,D</i>	<i>CB FD</i>	<i>SET</i>	<i>7,L</i>
<i>CB98</i>	<i>RES</i>	<i>3,B</i>	<i>CB CB</i>	<i>SET</i>	<i>1,E</i>	<i>CB FE</i>	<i>SET</i>	<i>7,(HL)</i>
<i>CB99</i>	<i>RES</i>	<i>3,C</i>	<i>CB CC</i>	<i>SET</i>	<i>1,H</i>	<i>CB FF</i>	<i>SET</i>	<i>7,A</i>
<i>CB9A</i>	<i>RES</i>	<i>3,D</i>	<i>CB CD</i>	<i>SET</i>	<i>1,L</i>	<i>CC 11hh</i>	<i>CAZ</i>	<i>nn</i>
<i>CB9B</i>	<i>RES</i>	<i>3,E</i>	<i>CB CE</i>	<i>SET</i>	<i>1,(HL)</i>	<i>CD 11hh</i>	<i>CALL</i>	<i>nn</i>
<i>CB9C</i>	<i>RES</i>	<i>3,H</i>	<i>CB CF</i>	<i>SET</i>	<i>1,A</i>	<i>CE nn</i>	<i>ADC</i>	<i>n</i>
<i>CB9D</i>	<i>RES</i>	<i>3,L</i>	<i>CB DΦ</i>	<i>SET</i>	<i>2,B</i>	<i>CF</i>	<i>RST</i>	<i>Φ8H</i>
<i>CB9E</i>	<i>RES</i>	<i>3,(HL)</i>	<i>CB D1</i>	<i>SET</i>	<i>2,C</i>	<i>DΦ</i>	<i>RNC</i>	
<i>CB9F</i>	<i>RES</i>	<i>3,A</i>	<i>CB D2</i>	<i>SET</i>	<i>2,D</i>	<i>D1</i>	<i>PDP</i>	<i>DE</i>
<i>CB AΦ</i>	<i>RES</i>	<i>4,B</i>	<i>CB D3</i>	<i>SET</i>	<i>2,E</i>	<i>D2 11hh</i>	<i>JPNC</i>	<i>nn</i>

D3 nn	OUT	n	DDCBvv46	BIT	0,(IX+d)	ED47	LD I,A
D4 llhh	CANC	nn	DDCBvv4E	BIT	1,(IX+d)	ED48	IN C
D5	PUSH	DE	DDCBvv56	BIT	2,(IX+d)	ED49	OUT C
D6 nn	SUB	n	DDCBvv5E	BIT	3,(IX+d)	ED4A	ADC HL,BC
D7	RST	1FH	DDCBvv66	BIT	4,(IX+d)	ED4Bllhh	LD BC,(nn)
D8	RC		DDCBvv6E	BIT	5,(IX+d)	ED4D	RETI
D9	EXX		DDCBvv76	BIT	6,(IX+d)	ED4F	LD R,A
DA llhh	JPC	nn	DDCBvv7E	BIT	7,(IX+d)	ED50	IN D
DB nn	IN	n	DDCBvv86	RES	0,(IX+d)	ED51	OUT D
DC llhh	CAC	nn	DDCBvv8E	RES	1,(IX+d)	ED52	SBC HL,DE
DD #9	ADD	IX,BC	DDCBvv96	RES	2,(IX+d)	ED53llhh	LD (nn),DE
DD19	ADD	IX,DE	DDCBvv9E	RES	3,(IX+d)	ED56	IM1
DD21 llhh	LD	IX,nn	DDCBvvA6	RES	4,(IX+d)	ED57	LD A,I
DD22 llhh	LD	(nn),IX	DDCBvvAE	RES	5,(IX+d)	ED58	IN E
DD23	INC	IX	DDCBvvB6	RES	6,(IX+d)	ED59	OUT E
DD29	ADD	IX,IX	DDCBvvBE	RES	7,(IX+d)	ED5A	ADC HL,DE
DD2A llhh	LD	IX,(nn)	DDCBvvC6	SET	0,(IX+d)	ED5Bllhh	LD DE,(nn)
DD2B	DEC	IX	DDCBvvCE	SET	1,(IX+d)	ED5E	IM2
DD84 vv	INC	(IX+d)	DDCBvvD6	SET	2,(IX+d)	ED5F	LD A,R
DD35 vv	DEC	(IX+d)	DDCBvvDE	SET	3,(IX+d)	ED60	IN H
DD36 vvnn	LD	(IX+d),n	DDCBvvE6	SET	4,(IX+d)	ED61	OUT H
DD39	ADD	IX,SP	DDCBvvEE	SET	5,(IX+d)	ED62	SBC HL,HL
DD46 vv	LD	B,(IX+d)	DDCBvvF6	SET	6,(IX+d)	ED67	RRD
DD4E vv	LD	C,(IX+d)	DDCBvvFE	SET	7,(IX+d)	ED68	IN L
DD56 vv	LD	D,(IX+d)	DDE1	POP	IX	ED69	OUT L
DD5E vv	LD	E,(IX+d)	DDE3	EX	(SP),IX	ED6A	ADC HL,HL
DD66 vv	LD	H,(IX+d)	DDE5	PUSH	IX	ED6F	RLD
DD6E vv	LD	L,(IX+d)	DDE9	JMP	(IX)	ED70	INF
DD70 vv	LD	(IX+d),B	DDF9	LD	SP,IX	ED72	SBC HL,SP
DD71 vv	LD	(IX+d),C	DEnn	SBC	n	ED73 llhh	LD (nn),SP
DD72 vv	LD	(IX+d),D	DF	RST	18H	ED78	IN A
DD73 vv	LD	(IX+d),E	EΦ	RPO		ED79	OUT A
DD74 vv	LD	(IX+d),H	E1	POP	HL	ED7A	ADC HL,SP
DD75 vv	LD	(IX+d),L	E2 llhh	JPP0nn		ED7B llhh	LD SP,(nn)
DD77 vv	LD	(IX+d),A	E3	EX	(SP),HL	EDAΦ	LD I
DD7EVv	LD	A,(IX+d)	E4 llhh	CAPE	nn	EDA1	CPI
DD86 vv	ADD	(IX+d)	E5	PUSH	HL	EDA2	IN I
DD8Evv	ADC	(IX+d)	E6 nn	AND	n	EDA3	OUT I
DD96 vv	SUB	(IX+d)	E7	RST	2FH	EDA8	LDD
DD9Evv	SBC	(IX+d)	E8	RPE		EDA9	CPD
DDA6 vv	AND	(IX+d)	E9	JMP	(HL)	EDA0	IND
DDAEvv	XOR	(IX+d)	EA llhh	JPPE	nn	EDAB	OUTD
DD86vv	OR	(IX+d)	EB	EX	DE,HL	ED8Φ	LDI
DDBEvv	CMP	(IX+d)	EC llhh	CAPE	nn	EDB1	CPIR
DDCBvvΦ6	RLC	(IX+d)	ED4Φ	IN	B	EDB2	INIR
DDCBvvΦE	RRC	(IX+d)	ED41	OUT	B	EDB3	OTIR
DDCBvv16	RL	(IX+d)	ED42	SBC	HL,BC	EDB8	LODR
DDCBvv1E	RR	(IX+d)	ED43llhh	LD	(nn),BC	EDB9	CPDR
DDCBvv26	SLA	(IX+d)	ED44	NEG		EDBA	INDR
DDCBvv2E	SRA	(IX+d)	ED45	RETN		EDB8	OTDR
DDCBvv3E	SRL	(IX+d)	ED46	IMΦ			

EE nn	XOR	n	FDCBvv16	RL	(IY+d)
EF	RST	28H	FDCBvv1E	RR	(IY+d)
F0	RP		FDCBvv26	SLA	(IY+d)
F1	POP	AF	FDCBvv2E	SRA	(IY+d)
F2 l(hh	JPP	nn	FDCBvv3E	SRL	(IY+d)
F3	DI		FDCBvv46	BIT	0,(IY+d)
F4 l(hh	CAP	nn	FDCBvv4E	BIT	1,(IY+d)
F5	PUSH	AF	FDCBvv56	BIT	2,(IY+d)
F6 nn	OR	n	FDCBvv5E	BIT	3,(IY+d)
F7	RST	3FH	FDCBvv66	BIT	4,(IY+d)
F8	RM		FDCBvv6E	BIT	5,(IY+d)
F9	LD	SP,HL	FDCBvv76	BIT	6,(IY+d)
FA l(hh	JPM	nn	FDCBvv7E	BIT	7,(IY+d)
FB	EI		FDCBvv86	RES	0,(IY+d)
FC l(hh	CAM	nn	FDCBvv8E	RES	1,(IY+d)
FD 09	ADD	IY,BC	FDCBvv96	RES	2,(IY+d)
FD 19	ADD	IY,DE	FDCBvv9E	RES	3,(IY+d)
FD 21 l(hh	LD	IY,nn	FDCBvvA6	RES	4,(IY+d)
FD 22 l(hh	LD	(nn),IY	FDCBvvAE	RES	5,(IY+d)
FD 23	INC	IY	FDCBvvB6	RES	6,(IY+d)
FD 29	ADD	IY,IY	FDCBvvBE	RES	7,(IY+d)
FD 2A l(hh	LD	IY,(nn)	FDCBvvC6	SET	0,(IY+d)
FD 2B	DEC	IY	FDCBvvCE	SET	1,(IY+d)
FD 34 vv	INC	(IY+d)	FDCBvvD6	SET	2,(IY+d)
FD 35 vv	DEC	(IY+d)	FDCBvvDE	SET	3,(IY+d)
FD 36 vvnn	LD	(IY+d),n	FDCBvvE6	SET	4,(IY+d)
FD 39	ADD	IY,SP	FDCBvvEE	SET	5,(IY+d)
FD 46 vv	LD	B,(IY+d)	FDCBvvF6	SET	6,(IY+d)
FD 4E vv	LD	C,(IY+d)	FDCBvvFE	SET	7,(IY+d)
FD 56 vv	LD	D,(IY+d)	FDE1	POP	IY
FD 5E vv	LD	E,(IY+d)	FDE3	EX	(SP),IY
FD 66 vv	LD	H,(IY+d)	FDE5	PUSH	IY
FD 6E vv	LD	L,(IY+d)	FDE9	JMP	(IY)
FD 70 vv	LD	(IY+d),B	FDF9	LD	SP,IY
FD 71 vv	LD	(IY+d),C	FEnn	CMP	n
FD 72 vv	LD	(IY+d),D	FF	RST	38H
FD 73 vv	LD	(IY+d),E			
FD 74 vv	LD	(IY+d),H			
FD 75 vv	LD	(IY+d),L			
FD 77 vv	LD	(IY+d),A			
FD 7E vv	LD	A,(IY+d)			
FD 86 vv	ADD	(IY+d)			
FD 8E vv	ADC	(IY+d)			
FD 96 vv	SUB	(IY+d)			
FD 9E vv	SBC	(IY+d)			
FD A6 vv	AND	(IY+d)			
FDAE vv	XOR	(IY+d)			
FD B6 vv	OR	(IY+d)			
FD BE vv	CMP	(IY+d)			
FDCBvv06	RLC	(IY+d)			
FDCBvv0E	RRC	(IY+d)			

8-Bit-Ladebefehle, Arithmetik, Logik									M= (ML)
	B	C	D	E	H	L	M	A	n
LD B	40	41	42	43	44	45	46	47	06
LD C	48	49	4A	4B	4C	4D	4E	4F	0E
LD D	50	51	52	53	54	55	56	57	16
LD E	58	59	5A	5B	5C	5D	5E	5F	1E
LD H	60	61	62	63	64	65	66	67	26
LD L	68	69	6A	6B	6C	6D	6E	6F	2E
LD M	70	71	72	73	74	75	76	77	36
LD A	78	79	7A	7B	7C	7D	7E	7F	3E
ADD	80	81	82	83	84	85	86	87	C6
ADC	88	89	8A	8B	8C	8D	8E	8F	CE
SUB	90	91	92	93	94	95	96	97	D6
SBC	98	99	9A	9B	9C	9D	9E	9F	DE
AND	A0	A1	A2	A3	A4	A5	A6	A7	E6
XOR	A8	A9	AA	AB	AC	AD	AE	AF	EE
OR	B0	B1	B2	B3	B4	B5	B6	B7	F6
CP	B8	B9	BA	BB	BC	BD	BE	BF	FE
INC	04	0C	14	1C	24	2C	34	3C	
DEC	05	0D	15	1D	25	2D	35	3D	
x x	(BC)	(DE)	(nn)		I	R			
LD A, x	0A	1A	3A		ED57	ED5F			
LD xx, A	02	12	32		ED47	ED4F			

16-Bit-Ladebefehle, Arithmetik

x	AF	BC	DE	HL	SP
PUSH x	F5	C5	D5	E5	
POP x	F1	C1	D1	E1	
LD x, nn	01	11	21	31	
LD x, (nn)	ED4B	ED5B	2A	ED7B	
LD (nn), x	ED43	ED53	22	EDA3	
ADD HL,x	09	19	29	39	
ADD HL,x	ED4A	ED5A	ED6A	ED7A	
SBC HL,x	ED42	ED52	ED62	ED72	
INC x	03	19	23	33	
DEC x	0B	1B	2B	3B	

Sprungbefehle, UP-Aufruf und Rückkehr

ub.	C	NC	Z	NZ	PE	PO	H	P
JP	C3	DA	D2	CA	C2	EA	E2	FA
CALL	CD	DC	D4	CC	C4	EC	E4	FC
RET	C9	DB	D0	CB	C0	E8	E0	F8
JR	18	38	30	28	20			

Hex: 0 08 10 18 20 28 30 38

RST	C7	CF	D7	DF	E7	EF	F7	FF
-----	----	----	----	----	----	----	----	----

RETI	ED4D
RETN	ED45
JMP (HL)	E9
DJNZ	10

1.00-Byte:	
für IX	DD
für IY	FD

ZVE - Steuerung			
NOP	00	1M0	ED46
HALT	76	1M1	ED56
DI	F3	1M2	ED5E
EI	FB		

1.00-Byte CB		Bit- und Rotationsbefehle							
		B	C	D	E	H	L	M	A
BIT 0	40	41	42	43	44	45	46	47	
1	48	49	4A	4B	4C	4D	4E	4F	
2	50	51	52	53	54	55	56	57	
3	58	59	5A	5B	5C	5D	5E	5F	
4	60	61	62	63	64	65	66	67	
5	68	69	6A	6B	6C	6D	6E	6F	
6	70	71	72	73	74	75	76	77	
7	78	79	7A	7B	7C	7D	7E	7F	
SET 0	C0	C1	C2	C3	C4	C5	C6	C7	
1	C8	C9	CA	CB	CD	CE	CF		
2	D0	D1	D2	D3	D4	D5	D6	D7	
3	D8	D9	DA	DB	DC	DD	DE	DF	
4	E0	E1	E2	E3	E4	E5	E6	E7	
5	E8	E9	EA	EB	ED	EE	EF		
6	F0	F1	F2	F3	F4	F5	F6	F7	
7	F8	F9	FA	FB	FD	FE	FF		
RES 0	80	81	82	83	84	85	86	87	
1	88	89	8A	8B	8C	8D	8E	8F	
2	90	91	92	93	94	95	96	97	
3	98	99	9A	9B	9C	9D	9E	9F	
4	A0	A1	A2	A3	A4	A5	A6	A7	
5	A8	A9	AA	AB	AC	AD	AE	AF	
6	B0	B1	B2	B3	B4	B5	B6	B7	
7	B8	B9	BA	BB	BC	BD	BF	BF	
RLC	00	01	02	03	04	05	06	07	
RRC	08	09	0A	0B	0C	0D	0E	0F	
RL	10	11	12	13	14	15	16	17	
RR	18	19	1A	1B	1C	1D	1E	1F	
SLA	20	21	22	23	24	25	26	27	
SRA	28	29	2A	2B	2C	2D	2E	2F	
SRL	30	31	3A	3B	3C	3D	3E	3F	

spezielle Rot.-Befehle

RLCA	07
RRCA	0F
RLA	17
RRA	1F
RRD	ED67
RLD	ED6F

Austauschbefehle u.a.	
EX DE,HL	EB
EXAF	#8
EXX	D9
EX (SP)HL	E3
LD SP,HL	F9

1.00-Byte ED		Einf. Ausgabebefehle							
		B	C	D	E	H	L	F	A
IN	40	48	50	58	60	68	70	78	
OUT	41	49	51	59	61	69			79

IN n	DB
OUT n	D3

1.00-Byte ED		Blockbefehle							
		LDI	AD	INI	A2				
LDI	80	80	80	80	80				
LDIR	80	80	80	80	80				
LDD	AB	AB	AB	AB	AB				
LDDR	B8	B8	B8	B8	B8				
SCF	3F	3F	3F	3F	3F				
CPI	A1	A1	A1	A1	A1				
CPIR	B1	B1	B1	B1	B1				
NEG	ED64	ED64	ED64	ED64	ED64				

Grenzwerte

Betriebsspannung, Eingangsspannungen auf U_{SS} (0-Potential) bezogen	-0,5 + 7V
Betriebstemperaturbereich	0 + 70°C
Lagertemperaturbereich	- 55 + 125 °C
Verlustleistung bei $\vartheta_A = 25^\circ C$	$\leq 1,1 W$

gemeinsame Kenngrößen der Schaltkreisfamilie:

Kenngröße	Zeichen	Min.	Max.	Einheit	Bemerkung
Betriebsspannung	U_{CC}	4,75	5,25	V	
Eingangsspannung	U_{IL}	-0,5	0,8		
	U_{IH}	2	U_{CC}	V	
Takteingangsspannung	U_{ILC}	-0,5	0,45		
	U_{IHC}	$U_{CC} - 0,2$	U_{CC}	μA	
Ausgangsspannung	U_{OL}		0,4		bei $I_{OL} = 1,8 mA$
	U_{OH}	2,4			bei $I_{OH} = -0,25 mA$
Eingangsreststrom Reststrom der tri-state Anschlüsse im hohohhm.Zustand	I_L		10	μA	Betrag! bei $U_I = 0 \dots U_{CC}$

statische Kennwerte und Kapazitäten - CPU

Eingangskapazität	C_x	5	pF	bei $\vartheta_A = 25^\circ C$ $f = 0,5 \dots 2 MHz$
Takteingangskapazität	C_c	50		
Ausgangskapazität	C_o	10		
Stromaufnahme	I_{CC}	200	mA	bei $U_{CC} = 5,25 V$ $\vartheta_A = 25^\circ C$

CPU USR00D elektrische Kennwerte

Dynamische Betriebsbedingungen :

Kenngröße		Zeichen	Min.	Max.	Einheit
Takt C	Periode Dauer 1-Zustand Dauer 0-Zustand Anstiegs-, Abfallzeit	t_c $t_w(CH)$ $t_w(CL)$ t_r, t_f	400 180 180 50	1) 2000 2000 50	
<u>WAIT</u>	Bereitstellzeit vor \ - Takt	$t_s(WT)$	70		
<u>RESET</u>	Bereitstellzeit vor \ - Takt	$t_s(RS)$	90		
<u>INT</u>	Bereitstellzeit vor \ - Takt	$t_s(IT)$	80		
<u>NMI</u>	Impulsbreite (0-Zustand)	$t_s(NMI)$	80		
<u>BUSREQ</u>	Bereitstellzeit vor \ - Takt	$t_s(BQ)$	80		
Daten D ₀ -D ₇	Bereitstellzeit bis \ - Takt (M1-Zyklus)	$t_{sc}(D)$	50		
Daten D ₀ -D ₇	Bereitstellzeit bis \ - Takt (außer M1)	$t_{sc}(D)$	60		
alle	Nachwirkzeiten	t_H	0		

$$1) \quad t_c = t_w(CH) + t_w(CL) + t_r + t_f$$

Taktsignalabhängige Zeitbeziehungen (Min) :

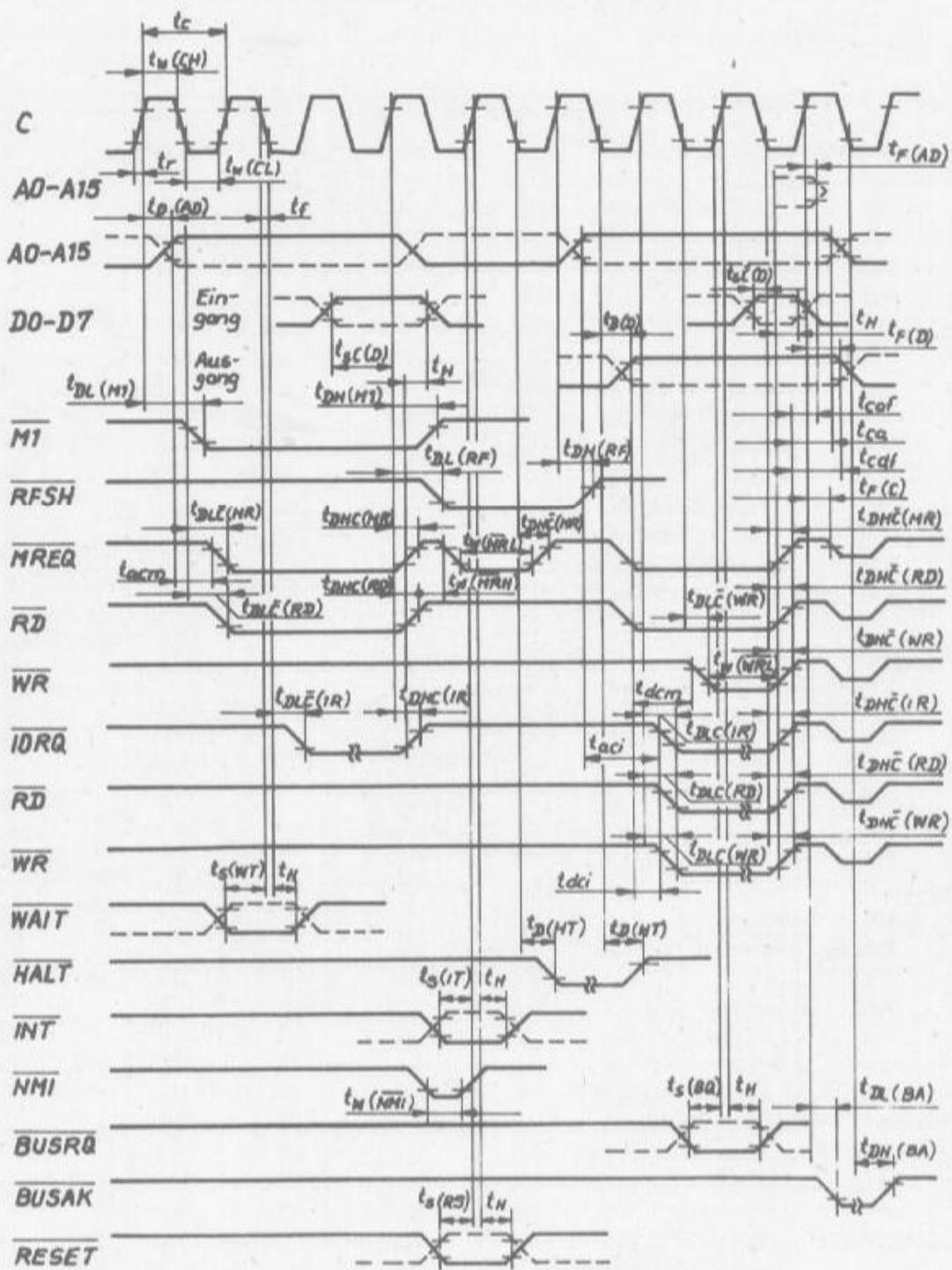
vor MREQ stabil (Speicher-Zyklus)		$t_{acm} = t_w(CH) + t_f - 75 \text{ ns}$
vor IORQ, RD oder WR stabil (E/A-Zyklus)		$t_{aci} = t_c - 80 \text{ ns}$
Adressen A ₀ -A ₁₅	nach RD oder WR stabil	$t_{ca} = t_w(CL) + t_r - 40 \text{ ns}$
	nach RD oder WR stabil (bei Übergang in hochohm. Zustand)	$t_{caf} = t_w(CL) + t_r - 60 \text{ ns}$
Daten D ₀ -D ₇	vor WR stabil (Speicher-Zyklus)	$t_{dcm} = t_c - 180 \text{ ns}$
Daten D ₀ -D ₇	vor WR stabil (E/A-Zyklus)	$t_{dci} = t_w(CL) + t_r - 180 \text{ ns}$
	nach WR stabil	$t_{dcf} = t_w(CL) + t_r - 50 \text{ ns}$
MREQ	Impulsbreite 0-Zustand	$t_w(\overline{MRL}) = t_c - 40 \text{ ns}$
	Impulsbreite 1-Zustand	$t_w(\overline{MRH}) = t_w(CH) + t_f - 30 \text{ ns}$
WR	Impulsbreite 0-Zustand	$t_w(\overline{WR}) = t_c - 40 \text{ ns}$
M1	vor IORQ stabil (Unterbrechungsannahme)	$t_{mr} = 2t_c + t_w(CH) + t_f - 80 \text{ ns}$

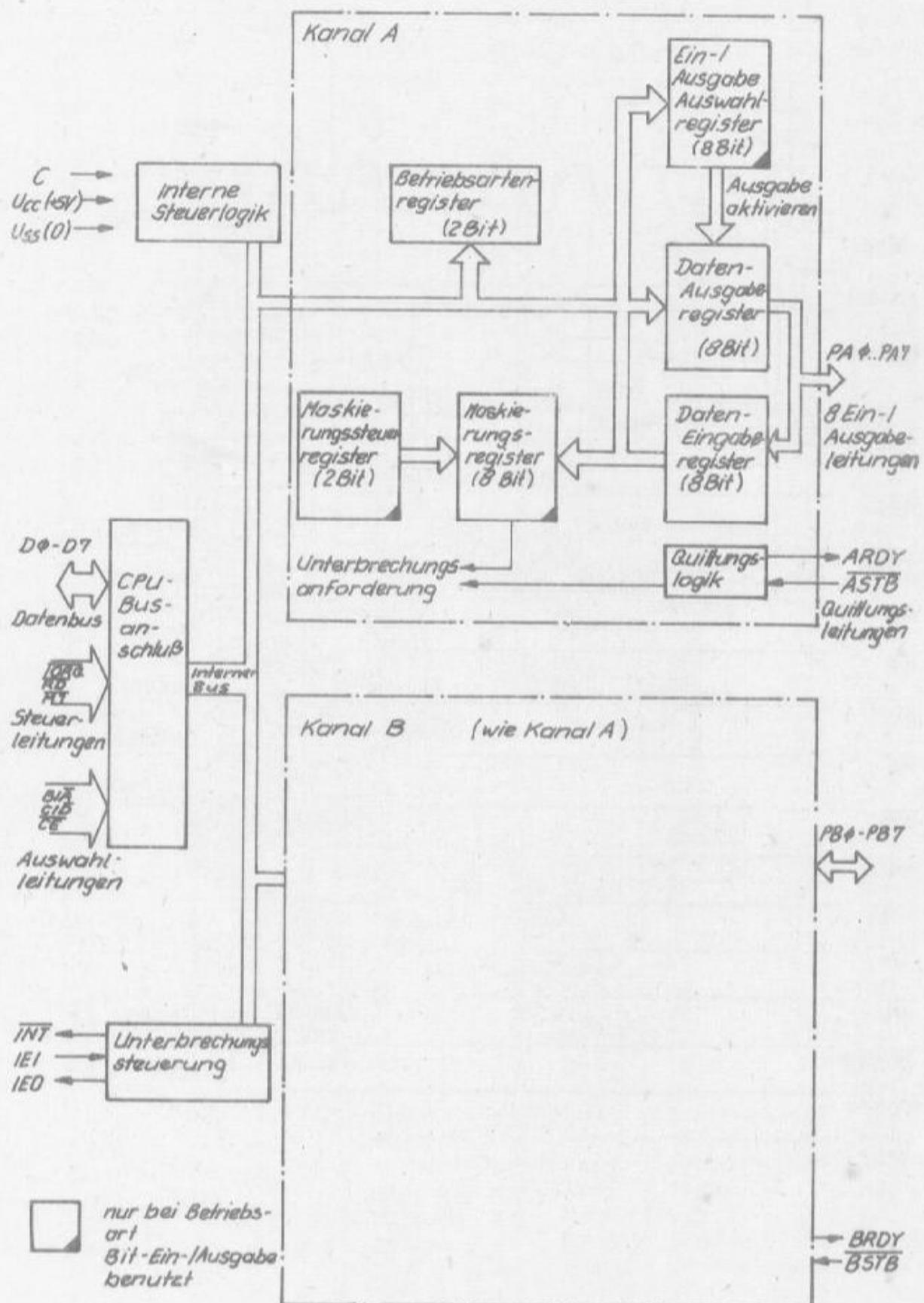
CPU U880D elektrische Kennwerte

Verzögerungszeiten:

bei : $U_{CC} = 4,75V$; $U_{IL} = 0,8V$; $U_{IH} = 2V$
 $U_{IIL} = 0,45V$; $U_{IHC} = 4,55V$
 $C_L = 100\text{pF}$; $\vartheta_a = 70^\circ\text{C}$

Kenngröße	Zeichen	Max.	Einheit
Ausgangsverzögerungszeit Adressen $A_0 - A_{15}$ Verzögerung bis hochohm. Zustand	t_D (AD)	160	
Ausgangsverzögerungszeit Daten $D_0 - D_7$ Verzögerung bis hochohm. Zustand	t_F (D)	110	
Verzögerung von \bar{I} -Takt bis $\overline{MREQ}=0$ Verzögerung von \bar{I} -Takt bis $\overline{MREQ}=1$ Verzögerung von \bar{I} -Takt bis $\overline{MREQ}=1$	$t_{DL\bar{C}}$ (MR) $t_{DH\bar{C}}$ (MR) $t_{DH\bar{C}}$ (MR)	110 110 110	
Verzögerung von \bar{I} -Takt bis $\overline{IORQ}=0$ Verzögerung von \bar{I} -Takt bis $\overline{IORQ}=0$ Verzögerung von \bar{I} -Takt bis $\overline{IORQ}=1$ Verzögerung von \bar{I} -Takt bis $\overline{IORQ}=1$	$t_{DL\bar{C}}$ (IR) $t_{DL\bar{C}}$ (IR) $t_{DH\bar{C}}$ (IR) $t_{DH\bar{C}}$ (IR)	120 110 120 110	
Verzögerung von \bar{I} -Takt bis $\overline{RD}=0$ Verzögerung von \bar{I} -Takt bis $\overline{RD}=0$ Verzögerung von \bar{I} -Takt bis $\overline{RD}=1$ Verzögerung von \bar{I} -Takt bis $\overline{RD}=1$	$t_{DL\bar{C}}$ (RD) $t_{DL\bar{C}}$ (RD) $t_{DH\bar{C}}$ (RD) $t_{DH\bar{C}}$ (RD)	110 140 110 120	ns
Verzögerung von \bar{I} -Takt bis $\overline{WR}=0$ Verzögerung von \bar{I} -Takt bis $\overline{WR}=0$ Verzögerung von \bar{I} -Takt bis $\overline{WR}=1$	$t_{DL\bar{C}}$ (WR) $t_{DL\bar{C}}$ (WR) $t_{DH\bar{C}}$ (WR)	90 100 110	
Verzögerung von \bar{I} -Takt bis $\overline{M1}=0$ Verzögerung von \bar{I} -Takt bis $\overline{M1}=1$	t_{DL} (M1) t_{DH} (M1)	145 145	
Verzögerung von \bar{I} -Takt bis $\overline{RFSH}=0$ Verzögerung von \bar{I} -Takt bis $\overline{RFSH}=1$	t_{DL} (RF) t_{DH} (RF)	195 160	
Verzögerung von \bar{I} -Takt bis $\overline{HALT}=0$	t_{DL} (HT)	310	
Verzögerung von \bar{I} -Takt bis $\overline{BUSAK}=0$ Verzögerung von \bar{I} -Takt bis $\overline{BUSAK}=1$	t_{DL} (BA) t_{DH} (BA)	130 120	
$MREQ, IORQ, RD, WR$ Verzögerung bis hochohm. Zustand	t_F (C)	110	





PIO U855D Anschlußbeschreibung

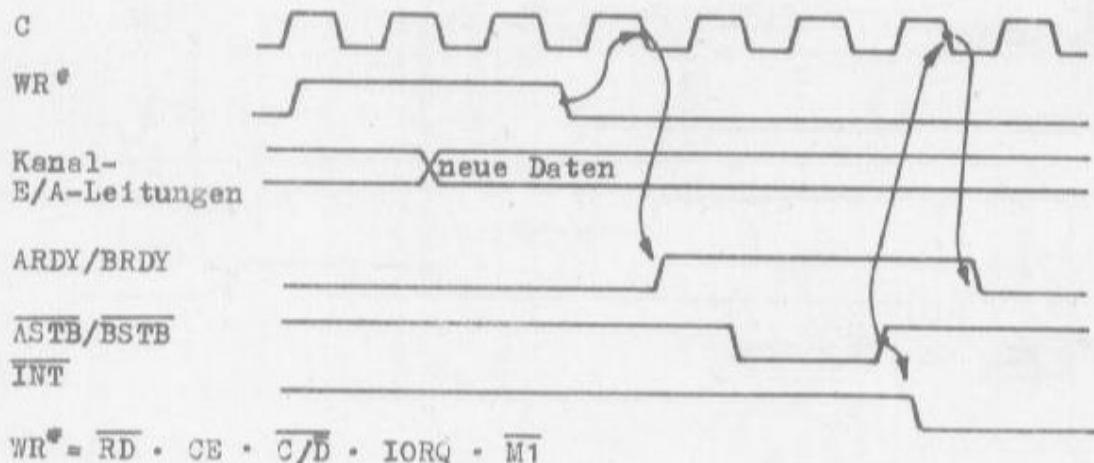
PIO			
19	D \emptyset	U855D	PA \emptyset
20	D1		PA1
1	D2		PA2
40	D3		PA3
39	D4		PA4
38	D5		PA5
3	D6		PA6
2	D7		PA7
			ARDY
16	ASTB		PB \emptyset
17	BSTB		PB1
6	B/ \bar{A}		PB2
5	C/D		PB3
4	CE		PB4
37	M1		PB5
36	IORQ		PB6
35	RD		PB7
25	C		BRDY
24	IEI		
			INT
			IEO
			23
			22

- D \emptyset -D7 Ein-/Ausgänge (tri-state) Systemdatenbus
- B/ \bar{A} Eingang Kanalauswahl, oft mit A1 verbunden
0 - Kanal A
1 - Kanal B
- C/D Eingang Auswahl Steuerinformationen/Daten (Control/ Data), oft mit A \emptyset verbunden
0-Daten
1-Steuerwort
- \overline{CE} Eingang Bausteinsauswahl, mit E/A-Adreßdekodierung zu verbinden (Chip Enable)
0-Baustein wird gelesen oder geschrieben
- $\overline{M1}$ Eingang Operationskode-Lesezyklus der CPU (Machine Cycle 1)
0-zusammen mit IORQ: Unterbrechungsannahme durch CPU, PIO kann Unterbrechungsvektor auf Datenbus geben (wenn INT von Seiten der PIO angemeldet und Priorität ausreichend)
0- ohne gleichzeitig IORQ oder RD aktiv: Rücksetzen des PIO-Bausteins!
- IORQ Eingang Ein-/Ausgabeanforderung der CPU (Input/ Output Request)
0 - E/A-Baustein lesen, schreiben oder Unterbrechungsannahme

<u>RD</u>	Eingang	Lesesignal (Read) 0 - Lesen von Speicher oder E/A-Bausteinen
<u>INT</u>	Ausgang mit offenem Drain	Unterbrechungsanforderung (Interrupt) 0 - PIO meldet eine Unterbrechung an
C	Eingang	Systemtakt (Clock)
<u>IEI</u>	Eingang	Unterbrechungsfreigabe (Interrupt Enable In) 1 - kein höher priorisierte Baustein wird gerade bedient
<u>IEO</u>	Ausgang	Unterbrechungsfreigabe (Interrupt Enable Out) 1 - wenn IEI=1 und keine Unterbrechung der PIO gerade behandelt wird
PA0-PAT PB0-PB7	Ein-/Ausgänge (tri-state)	Peripherieanschlüsse von Kanal (Port) A bzw. B
<u>ASTB</u> , <u>BSTB</u>	Eingänge	Datenübernahmesignal (Strobe) für Kanal A bzw. B bei Ausgabe: <input checked="" type="checkbox"/> - Peripherie hat Daten abgeholt bei Eingabe: <input checked="" type="checkbox"/> - Peripherie veranlaßt Datenübernahme in PIO, im Bit-Modus unwirksam Details siehe Funktionsbeschreibung/Ablaufdiagramme
<u>ARDY</u> , <u>BRDY</u>	Ausgänge	Quittungssignal (Ready) für Kanal A bzw. B bei Ausgabe: 1 - Daten stehen zur Abholung bereit bei Eingabe: 1 - Eingaberegister aufnahmebereit für neue Daten, im Bit-Modus: nicht verwendet (ständig 0) Details siehe Funktionsbeschreibung/Ablaufdiagramme

Betriebsarten:Byte-Ausgabe (Mode Ø)

Beim Ausführen eines Ausgabebefehls durch die CPU werden die auszugebenden Daten in das Ausgaberegister des ausgewählten Kanals eingeschrieben. Die Daten stehen danach an den Ein-/Ausgabeleitungen an, bis neue Daten auf dem Kanal ausgegeben werden oder eine Betriebsartänderung bzw. Rücksetzen erfolgt. Wenn das durch die UND-Verknüpfung $\overline{WR^*} \cdot \overline{RD} \cdot CE \cdot \overline{CD} \cdot IORQ \cdot \overline{M1}$ in der PIO gewonnene Signal inaktiv wird, wird das Quittungssignal (READY) ARDY, BRDY auf 1 gesetzt, um der angeschlossenen Peripherie anzuzeigen, daß neue Daten anliegen. Die angeschlossene Peripherie zeigt ihrerseits die Übernahme der neuen Daten durch eine steigende Flanke des Datenübernahmesignals (STROBE) $\overline{ASTB}/\overline{BSTB}$ an. Daraufhin geht das Quittungssignal ARDY/BRDY wieder auf Ø. Außerdem wird, falls erlaubt, jetzt eine Unterbrechung ausgelöst, die der CPU signalisiert, daß die Daten abgeholt worden sind und neue Daten auf den Kanal ausgegeben werden können.

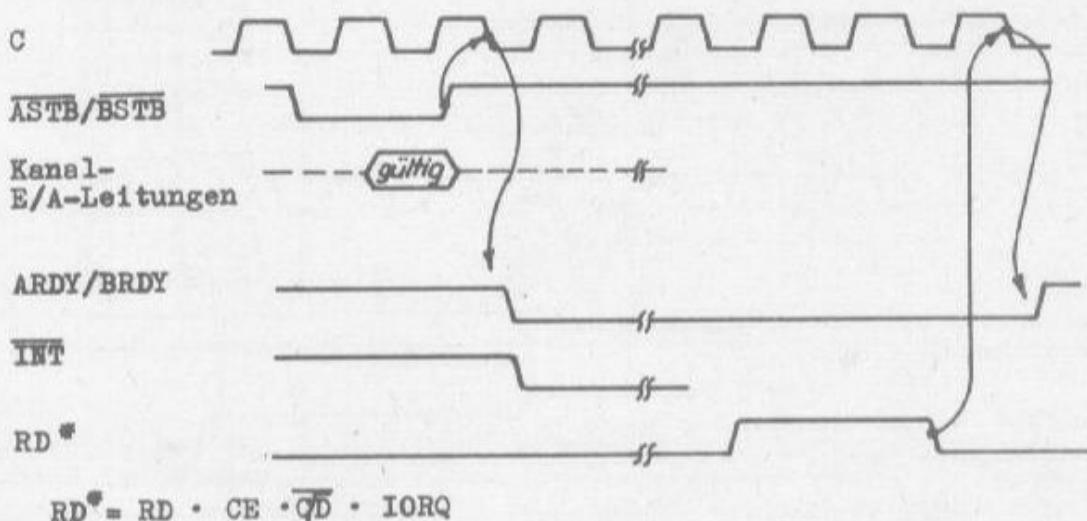


Byte-Eingabe (Mode 1)

Die an den Kanal-Ein-/Ausgabeeleitungen anstehenden Daten von der Peripherie werden mit der ansteigenden Flanke des Datenübernahmesignals $\overline{ASTB/BSTB}$ in das Dateneingaberegister übernommen.

Daraufhin wird das Quittungssignal ARDY, BRDY auf 0 gesetzt, damit wird signalisiert, daß sich im Eingaberegister Daten befinden, die noch nicht von der CPU abgeholt wurden.

Bei der Datenübernahme wird, falls erlaubt, eine Unterbrechung der CPU ausgelöst. Wenn die CPU die Daten von der PIO abgeholt hat, geht das Quittungssignal ARDY/BRDY wieder auf 1 um anzuseigen, daß der Kanal zur Übernahme neuer Daten bereit ist.



Byte-Ein-/Ausgabe, Zweirichtungsbetrieb (Mode 2)

- Nur Kanal A; Kanal B kann dann nur für Bit-Ein-/Ausgabe (Mode 3) programmiert werden!

Diese Betriebsart kombiniert "Byte-Eingabe" und Byte-Ausgabe" an den Ein-/Ausgabeleitungen des Kanals A. Es werden alle 4 Quittungsleitungen benutzt:

RDY, \overline{ASTB} für Byte-Ausgabe (wie Mode 0)
BRDY, \overline{ESTB} für Byte-Eingabe (wie Mode 1)

Die Datenausgabe erfolgt nur während $\overline{ASTB}=0$, ansonsten sind die Ein-/Ausgabeleitungen als Eingänge zu betrachten.

Bit-Ein-/Ausgabe (Mode 3)

Es wird nicht mit Quittungssignalen gearbeitet, ausgegebene Daten erscheinen sofort an den als Ausgang definierten E/A-Leitungen des Kanals.

Beim Lesen eines Kanals enthalten die als Eingänge definierten Bits den Zustand der jeweiligen E/A-Leitung (zum Zeitpunkt der fallenden Flanke von \overline{IORQ} bzw. \overline{RD}); die als Ausgänge definierten Bits liefern beim Lesen den Inhalt des Ausgaberegisters für die jeweiligen Leitungen.

Einzelne als Eingabe definierten Leitungen oder auch Verknüpfungen mehrerer Eingabeleitungen können Unterbrechungen auslösen (siehe Programmierung).

Die Betriebsart Bit-Ein-/Ausgabe ist auch dann zweckmäßig, wenn eine Byte-Ein- oder Ausgabe ohne Quittungsbetrieb (RDY/ \overline{STB}) realisiert werden soll.

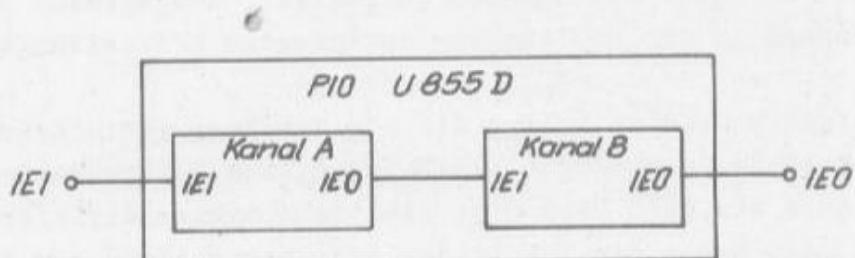
PIO U855D Funktion

Unterbrechungsbetrieb:

In den Betriebsarten Byte-Eingabe, Byte-Ausgabe und Byte-Ein-/Ausgabe können Unterbrechungen von der steigenden Flanke des Datenübernahmesignals ASTB/BSTB erzeugt werden.

In der Betriebsart Bit-Ein-/Ausgabe kann programmiert werden, daß eine bestimmte Eingangsleitung bzw. die Verknüpfung mehrerer Eingangsleitungen (mit UND oder ODER) eine Unterbrechung auslöst, wenn sich ein bestimmter logischer Zustand (0 oder 1) einstellt.

Die Unterbrechungsannahme ist ebenso wie die Schreib- und Lesezyklen des Bausteins (E/A-Lese- bzw. Schreibzyklus) im Abschnitt CPU beschrieben. Die beiden Kanäle bilden auch innerhalb der PIO eine Prioritätskette, der Kanal A hat die höhere Priorität:



Die gewünschte Funktion des PIO-Bausteins muß durch Ausgabe von Steuerworten auf die Steuerwortadresse des jeweiligen Kanals festgelegt werden.

Die verschiedenen Arten von Steuerworten werden durch bestimmte feste Bitbelegungen unterschieden. Eine Ausnahme sind die Bitmasken für die Ein-/Ausgabe- und Unterbrechungsfestlegung in der Betriebsart Bit-Ein-/Ausgabe, die auf bestimmte Ankündigungssteuerworte folgen müssen.

Betriebsartensteuerwort

D7	D6	D5	D4	D3	D2	D1 ^c	DO
		x	x	1	1	1	1

Kennzeichnung des Steuerwortes als Betriebsartensteuerwort

0	0	Byte-Ausgabe
0	1	Byte-Eingabe
1	0	Byte-Ein-/Ausgabe (nur Kanal A)
1	1	Bit-Ein-/Ausgabe

} mit Quittungsbetrieb

Eingangs-/Ausgangsfestlegung bei Bit-Ein-/Ausgabe (muß bei dieser Betriebsart dem Betriebsartensteuerwort folgen)

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

Leitung ist Ausgang

Leitung ist Eingang

Unterbrechungsvektor (existiert für jeden Kanal separat)

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

- Kennzeichnung des Steuerwortes als Unterbrechungsvektor

Unterbrechungs-Steuerwort

D7	D6	D5	D4	D3	D2	D1	DO
x	x	x	0	0	1	1	

0 Unterbrechung durch PIO-Kanal gesperrt

1 Unterbrechung durch PIO-Kanal erlaubt

x - beliebig

Unterbrechungs-Steuerwort für Bit-Ein-/Ausgabe

D7	D6	D5	D4	D3	D2	D1	D \emptyset
				Ø	1	1	1

1 - Es folgt die Bitmaske zur Festlegung der für eine Unterbrechungsauslösung zu berücksichtigenden Leitungen

1 - 1 - Zustand der Kanalleitungen löst Unterbrechung aus
0 - 0 - Zustand der Kanalleitungen löst Unterbrechung aus

1 - UND - Verknüpfung der ausgewählten Kanalleitungen
0 - ODER - Verknüpfung der ausgewählten Kanalleitungen

1 - Unterbrechung durch PIO-Kanal erlaubt
0 - Unterbrechung durch PIO-Kanal gesperrt

Bitmaske zur Festlegung der für eine Unterbrechungsauslösung zu berücksichtigenden Leitungen (folgt unmittelbar nach Unterbrechungs-Steuerwert für Bit-Ein-/Ausgabe mit D4=1)

D7	D6	D5	D4	D3	D2	D1	D \emptyset
0	0	0	0	0	0	0	0

0 - Leitung in Verknüpfung einbezogen
1 - Leitung nicht berücksichtigt

Grenzwerte und allgemeine statische Kennwerte wie CPU
statische Kennwerte und Kapazitäten - PIO:

Kenngröße	Zeichen	Min	Max	Einheit	Bemerkungen
Eingangskapazität	C_I		7	pF	bei $\vartheta_a = 25^\circ C$, $f = 0,5 \dots 2 \text{ MHz}$
Takteingangskapazität	C_J		14		
Ausgangskapazität	C_O		10		
Stromaufnahme	I_{CC}		100	mA	bei $U_{cc} = 5,25V$ $\vartheta_a = 25^\circ C$
Laststrom der Darlington-Treiber-Ausgänge (nur Kanal B)	I_{OHD}	1,5	3,8	mA	bei $U_{OH} = 1,5V$ $R_{ext} = 390 \Omega$

Dynamische Betriebsbedingungen:

Kenngröße	Zeichen	Min	Max	Einheit
Takt C	wie CPU			
$\overline{CE}, \overline{C/D}, \overline{B/A}$ Setzzeit zu \overline{J} -Takt	$t_{SC(CE)}$	280		
$\overline{D}_0\text{-}D_7$ Setzzeit zu \overline{J} -Takt	$t_{SC(D)}$	50		
\overline{IEI} Setzzeit zu \overline{L} -IORQ bei Unterbrechungsannahme	$t_S(IEI)$	140		
\overline{RD} Setzzeit zu \overline{J} -Takt	$t_{SC(RD)}$	240		
\overline{IORQ} Setzzeit zu \overline{J} -Takt	$t_{SC(IORQ)}$	250		ns
$\overline{M1}$ Setzzeit zu \overline{J} -Takt	$t_{SC(M1)}$	210		
$\overline{PA}_0\text{-}PA_7$ $\overline{PB}_0\text{-}PB_7$ Setzzeit zu \overline{J} -ASTB bzw. \overline{J} -BSTB (Modus 1)	$t_S(PD)$	200		
$\overline{ASTB}, \overline{BSTB}$ Impulsbreite Impulsbreite in Modus 2 (nur \overline{BSTB})	$t_W(ST)$ $2t_W(ST)$	150 $>t_S(PD)$		
$\overline{M1}$ als Dauer für RESET Rücksetzen	$t_{M1(RESET)}$	$2 t_C$		
alle Haltezeiten für die angegebenen Setzzeiten	t_H	0		

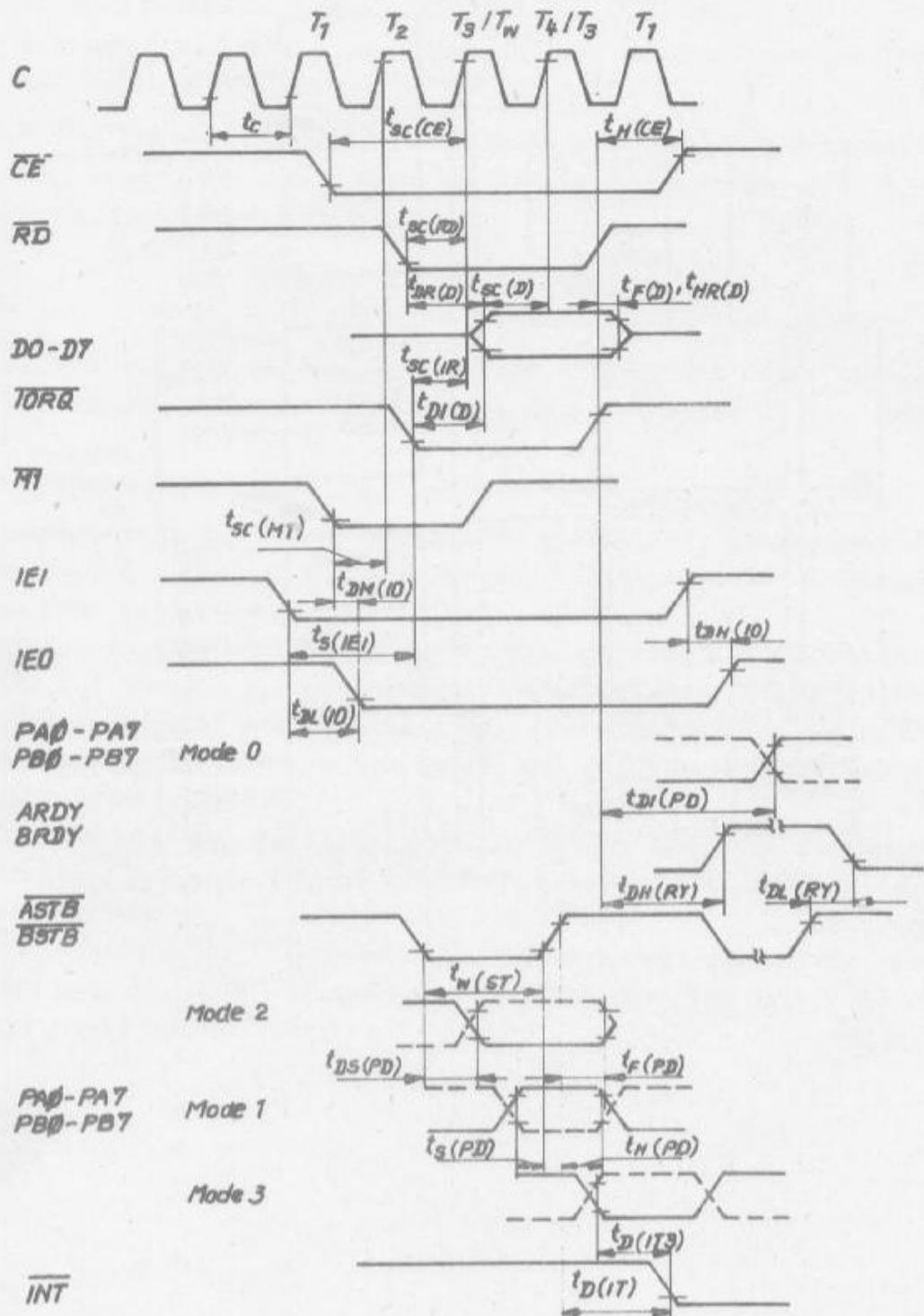
Verzögerungszeiten

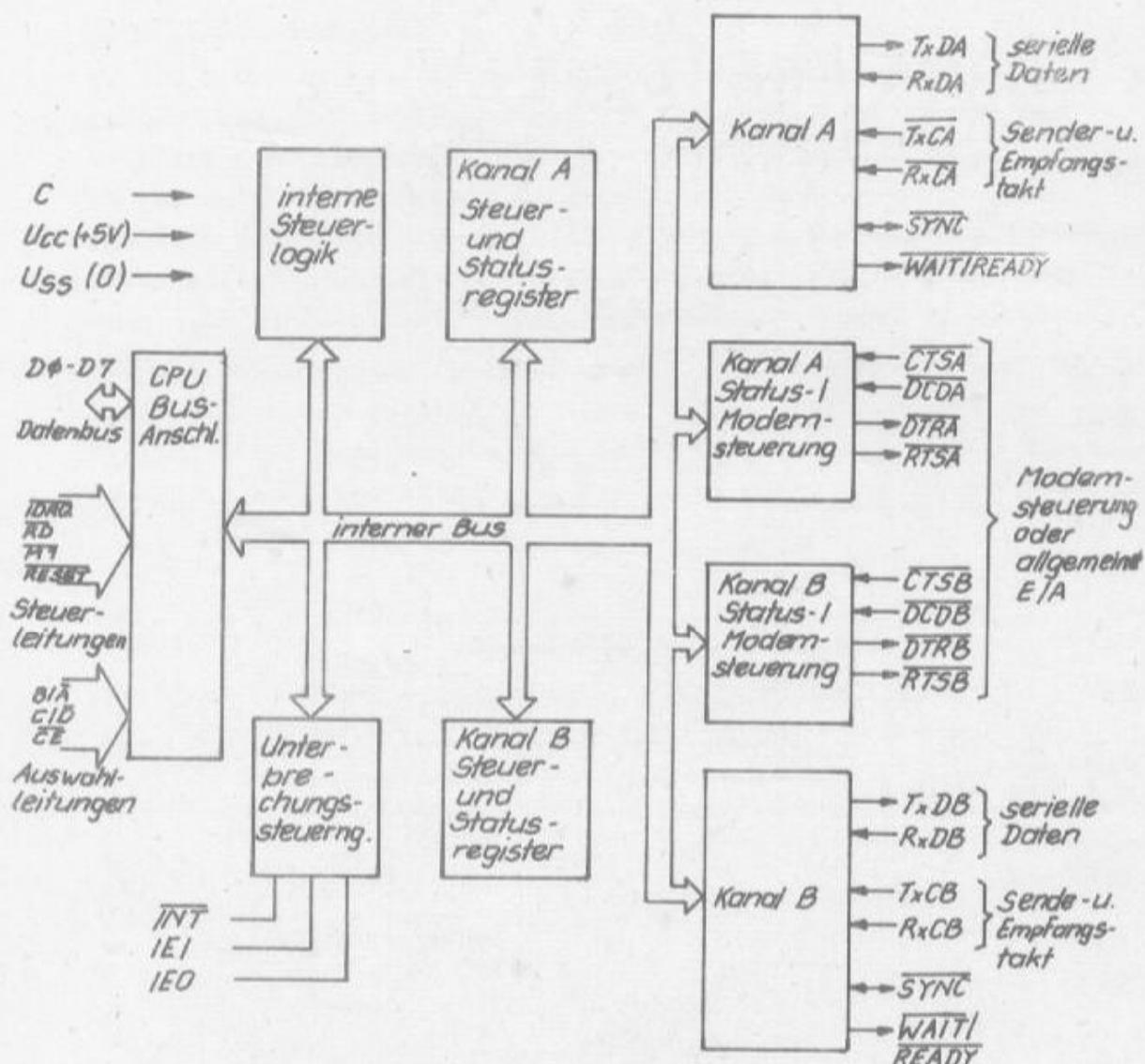
bei $U_{CC} = 4,75V$; $U_{IL} = 0,8V$; $U_{IH} = 2V$

$U_{ILC} = 0,45V$; $U_{IHC} = 4,55V$

$C_L = 100pF$; $T_a = 70^\circ C$

Kenngröße	Zeichen	Min	Max	Einheit
Daten DØ-D7	Ausgabe ab \neg -RD Ausgabe ab \neg -IORQ bei Unterbrechungsan- nahme Übergang in hochohm. Zustand	$t_{DR(D)}$ $t_{DI(D)}$ $t_F(D)$	440 350 170	
I/O	Verzögerung zu \neg -IEI Verzögerung zu \neg -IEI Verzögerung zu \neg -M1 (Unterbrechung noch vor M1)	$t_{DH(I0)}$ $t_{DL(I0)}$ $t_{DM(I0)}$	220 200 310	
PAØ-PA7 PBØ-PB7	Ausgabe ab \neg -ASTB (Modus 2) Übergang in hochohm. Zustand ab \neg -ASTB (Modus 2) Datengültig ab \neg -IORQ (Schreibzyklus, Modus Ø)	$t_{DS(PD)}$ $t_F(PD)$ $t_D(PD)$	240 210 210	ns
INT	Verzögerung zu \neg -ASTB/BSTB Verzögerung zu Daten- bedingung erfüllt (in Modus 3)	$t_{D(IT)}$ $t_{D(IT3)}$	500 660	
ARDY BRDY	Verzögerung zu \neg -ASTB/BSTB Verzögerung zu \neg -IORQ	$t_{DL(RY)}$ $t_{DH(RY)}$	$t_C + 410$ $t_C + 470$	



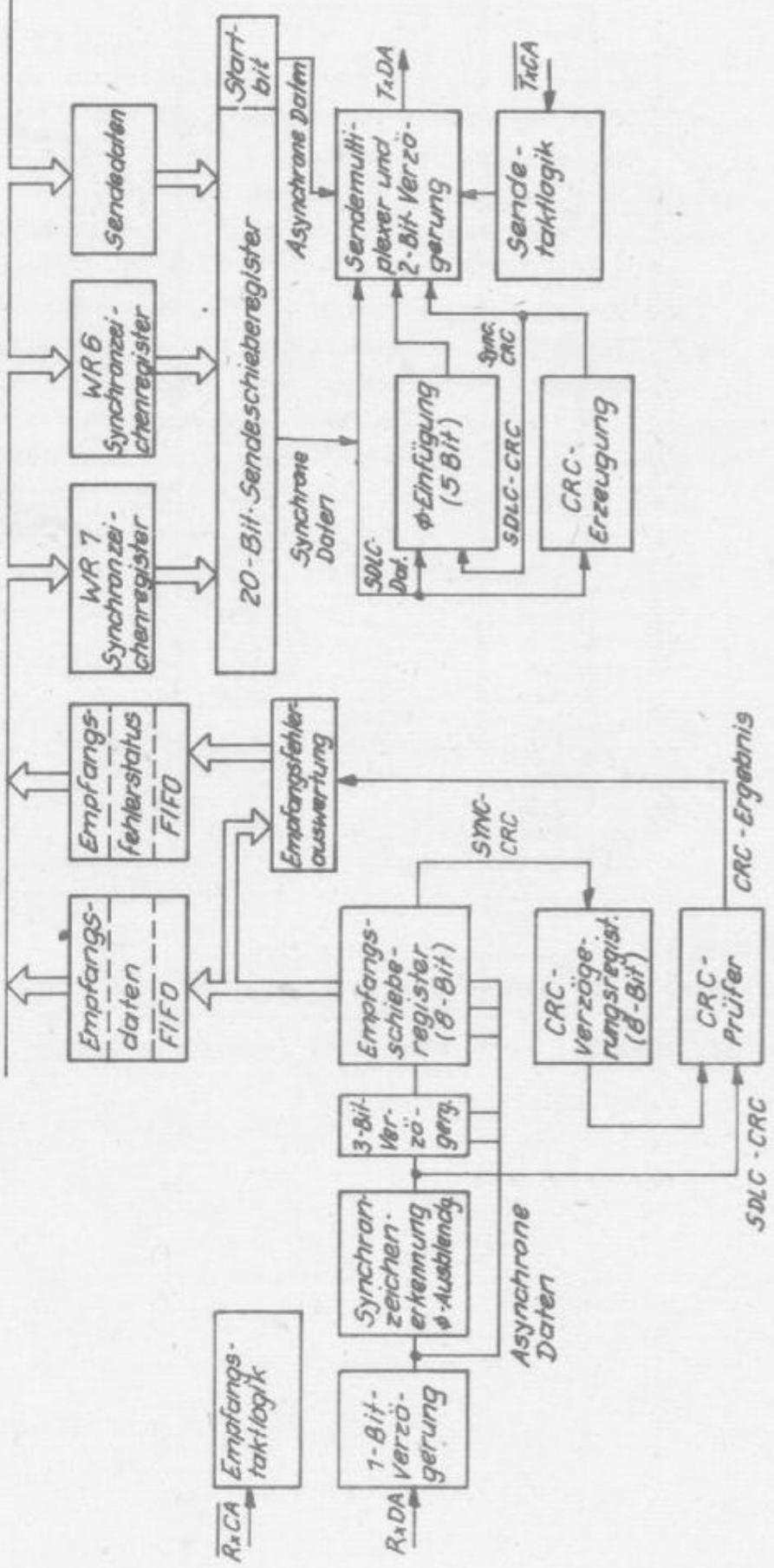


Interne Struktur eines SIO-Kanals
(Kanal A):

Systemdatenbus D₀-D₇



interner Datenbus



SIO U856D	
40	D \emptyset
1	D1
39	D2
2	D3
38	D4
3	D5
37	D6
4	D7
34	B/A
33	C/D
35	CE
8	M1
36	IORQ
32	RD
20	C
21	RESET
6	IEI
12	RxD A
13	RxC A
14	TxC A
18	CTS A
19	DCDA
28	RxD B
27	RxC B
23	TxC B
22	CTS B
	DCDB
	TxD A
	W/RDY A
	RTSA
	DTRA
	SYNCA
	26
	W/RDY B
	RTSB
	DTRB
	SYNCB
	INT
	IEO
	15
	10
	17
	16
	11
	24
	25
	29
	5
	7

Achtung !

Dargestellt ist die Bondvariante mit TxCB, RxCB auf gemeinsamen Anschluß (angefügte Ziffer \emptyset)!

Unterschiede bei 2. Bondvariante (angefügte Ziffer 1): TxCB - 26

RxCB - 27

TxD B - 25

DTRB - nicht herausgeführt

D ₀ -D ₇	Ein-/Ausgänge (tri-state)	Systemdatenbus
B/A	Eingang	Kanalauswahl, oft mit A ₁ verbunden 0 - Kanal A 1 - Kanal B
C/D	Eingang	Auswahl Steuerinformationen/Daten (Central/ Data), oft mit A ₀ verbunden 0 - Daten 1 - Steuerwert
CE	Eingang	Bausteineauswahl (Chip Enable) mit E/A-Adressekodierung zu verbinden 0-Baustein wird gelesen oder geschrieben
MI	Eingang	Operationskode-Lesezyklus der CPU (Machine Cycle 1) 0 - zusammen mit IORQ: Unterbrechungsan- nahme durch CPU, SIO kann Unterbrechungs- vektor auf Datenbus geben (wenn INT von SIO angemeldet und Priorität vorhanden) außerdem: Erkennen des RETI-Befehls durch SIO
IORQ	Eingang	Ein-/Ausgabeanforderung der CPU (Input/ Output Request) 0 - E/A-Baustein lesen, schreiben oder Unterbrechungsannahme
RD	Eingang	Lesesignal (Read) 0 - Lesen von Speicher oder E/A-Bausteinen
INT	Ausgang mit offenem Drain	Unterbrechungsanforderung (Interrupt) 0 - SIO meldet Unterbrechung an
RESET	Eingang	Rücksetzen 0,- Sperrung von Sender und Empfänger, TxDA, TxDB nehmen 1-Pegel an, Modussteuerleitungen in 1-Zustand, Unterbrechungen gesperrt Die Steuerregister müssen neu programmiert werden, bevor der SIO-Baustein wieder arbeitsfähig ist.
IEI	Eingang	Unterbrechungsfreigabe (Interrupt Enable In) 1 - kein höher priorisierte Baustein wird gerade bedient
IEO	Ausgang	Unterbrechungsfreigabe (Interrupt Enable Out) 1 - wenn IEI=1 und keine Unterbrechung der SIO gerade behandelt wird

C	Eingang	Systemtakt (Clock)
<u>CTS</u> A, <u>CTS</u> B	Eingänge mit Trigger-verhalten	Sendebereitschaft (Clear to Send) In der Betriebsart "Bereitschaft durch externes Signal" wird bei Ø der Sendeteil des jeweiligen Kanals in den Bereit-Zustand gesetzt, sonst als allgemeine Eingänge verwendbar. Unterbrechung erfolgt bei 0-1 und 1-0 Übergängen, wenn erlaubt.
<u>DCDA</u> , <u>DCDB</u>	Eingänge m. Trigger-verhalten	Datenträgermeldung (Data Carrier Detect) In der Betriebsart "Bereitschaft durch externes Signal" wird bei Ø der Empfangsteil des jeweiligen Kanals in den Bereit-Zustand gesetzt, sonst als allgemeine Eingänge verwendbar. Unterbrechung erfolgt bei 0-1 und 1-0 Übergängen, wenn erlaubt.
<u>DTR</u> A, <u>DTR</u> B	Ausgänge	Datenendeinrichtung bereit (Data Terminal Ready) Folgt der Programmierung des DTR-Bits in SR5, DTR=1 bewirkt Ø am Ausgang des entsprechenden Kanals, als allgemeiner Ausgang verwendbar.
<u>RXCA</u> , <u>RXC</u> B	Eingänge m. Trigger-verhalten	Empfangstakt (Receive Clock) Die Empfangsdatenabtastung erfolgt bei einer aufsteigenden Flanke des Empfangstaktes, im Asynchronbetrieb auf das 1, 16, 32, 64-fache der Datenrate programmierbar.
<u>RxD</u> A, <u>RxD</u> B	Eingänge	Empfangsdaten (Receive Data)
<u>RTS</u> A, <u>RTS</u> B	Ausgänge	Sendeanforderung (Ready to Send) Folgt der Programmierung des RTS-Bits in SR5, RTS=1 bewirkt Ø am Ausgang des entsprechenden Kanals, wenn im Asynchronbetrieb das Bit RTS=0 gesetzt wird, geht der Ausgang auf sobald der Sendepuffer leer ist, ansonsten als allgemeiner Ausgang verwendbar.
<u>SYN</u> A, <u>SYN</u> B	Eingänge/ Ausgänge (programmierbar)	Synchronisation Asynchronbetrieb: Eingänge auf SYNC-Bit in LRØ, Ø am Eingang bewirkt SYNC-Bit=1 Synchronbetrieb, externe Synchronisation: Eingang muß auf Ø gesetzt sein bei der zweiten aufsteigenden Flanke von Rx C nach derjenigen aufsteigenden Flanke von Rx C, mit der das letzte Bit des Synchronzeichens empfangen wurde. Synchronbetrieb, interne Synchronisation Ausgang, liefert Ø während jeder Periode von Rx C, in der ein Synchronzeichen erkannt wurde, unabhängig von den Zeichengrenzen.

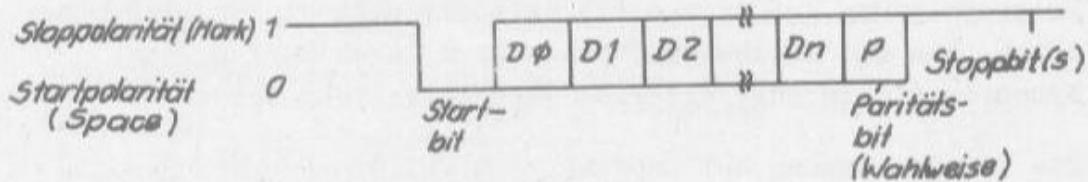
TxCA, TxCB	Eingänge m. Trigger- verhalten	Sendetakt (Transmitter Clock) Die neuen Daten stehen mit der fallenden Flanke von TxC an. Im Asynchronbetrieb auf das 1, 16, 32, 64-fache der Datenrate programmierbar (zusammen mit Empfangstakt).
TxDA, TxDB	Ausgänge	Sendedaten (Transmit Data)
W/RDY _A , W/RDY _B	Ausgang normal/mit offenem Drain (pro- grammierbar)	Warteanforderung/Bereitmeldung als WAIT programmiert (und nach dem Rück- setzen): Ausgang mit offenem Drain, WAIT- Signal zur Synchronisierung mit CPU U880D als READY programmiert: Bereit-Signal für direkten Speicherzugriff (DMA)

Der hochintegrierte Schaltkreis U856D ist zur Unterstützung serieller Datenübertragungsaufgaben, vorrangig in Mikrorechensystemen mit der CPU U880D vorgesehen und wird durch folgende Merkmale charakterisiert:

- Zwei unabhängige Kanäle
- In jedem Kanal gleichzeitiges Senden und Empfangen möglich (Duplex)
- Betriebsweisen:
 - Asynchronbetrieb mit 5-8 Datenbits und 1, 1,5 oder 2 Stopbits je Zeichen
 - Synchronbetrieb mit 8-Bit-Synchronzeichen (Monosync),
16-Bit-Synchronzeichen (Bisync),
externer Synchronisation, oder
SDLC/HDLC-Modus
- Wahlweise automatische Paritätserzeugung und -Auswertung
- Bei Synchronbetrieb automatische CRC-Erzeugung und- Auswertung
- Je Kanal 2 Eingänge und 2 Ausgänge zur Modemsteuerung oder als allgemeine E/A-Anschlüsse.
- Taktvorteilerfaktor programmierbar $x1$, $x16$, $x32$, $x64$
- Takt braucht nicht symmetrisch sein und kann daher z.B. von einem CTC U856D geliefert werden
- Datenübertragungsrate 0-550 KBit/s
- Direkt anschließbar an U880D Bussystem, TTL-kompatibel, eine Versorgungsspannung
- Leistungsfähiges Unterbrechungssystem mit unterbrechungsursachenabhängigem Vektor
- Fähigkeit zur Synchronisierung mit CPU (WAIT) oder DMA-Schaltung (READY)

Asynchronbetrieb

- Senden und Empfangen unabhängig voneinander mit 5...8 Bit je Zeichen sowie wahlweise angefügtem Paritätsbit.
- 1, 1 1/2 oder 2 Stopbits je Zeichen beim Senden
- Beim Empfang muß mindestens 1 Stopbit vorhanden sein, sonst erfolgt die Statusmeldung "Zeichenrahmenfehler"
- Zur Unterdrückung kurzzeitiger Störungen wird ein Startbit bei Empfang nur dann als gültig gewertet, wenn vom 1-0-Ubergang an mindestens für eine halbe Bitzeit Startpolarität (0) anliegt, anschließend werden jeweils in der Mitte der Informationsbits die Empfangsdaten abgetastet.
Die Bitsynchronisation kann nur dann durch den Schaltkreis vorgenommen werden, wenn ein Taktteilerfaktor # 16 benutzt wird.
- Empfänger und Sender beginnen ihre Arbeit mit dem Setzen der entsprechenden Freigabebits in den Schreibregistern. In der Betriebsart "Externe Freigabe" müssen zusätzlich noch die Eingangssignale CTS (für Sender) bzw. DCD (für Empfänger) aktiv sein.
- Das BREAK-Kommando bewirkt, daß die Sendedatenleitung sofort Stoppolarität (1) annimmt, ein gerade gesendetes Zeichen geht verloren.
- Der Empfang des BREAK-Zustands wird durch ein Statusbit angezeigt.
Wenn externe Statusunterbrechungen erlaubt sind, führt sowohl der Übergang in den BREAK-Zustand wie auch dessen Verlassen zu je einer Unterbrechung.



Synchronbetrieb (außer SDLC/HDLC)

- Taktteilerfaktor x_1 ist bei jeder synchronen Übertragung zu programmieren, die Bitsynchronisation muß daher extern erfolgen.
- Empfangsdaten werden mit der steigenden Flanke von Rx.C abgetastet, Sendedatenänderungen erfolgen mit der fallenden Flanke von Tx.C.
- Die Betriebsweisen mit 8-Bit-Synchronzeichen (Monosync), 16-Bit-Synchronzeichen (Bisync) oder externer Synchronisation unterscheiden sich nur in der Art und Weise der Erstsynchronisation.
- Wenn keine Daten zum Senden zur Verfügung stehen, wird, falls aktiviert, eine Prüffolge (CRC), und anschließend Synchronzeichen gesendet.
- Die Prüffolge kann nach der Vorschrift CRC 16 ($x^{16} + x^{15} + x^2 + 1$) oder CRC-CCITT ($x^{16} + x^{12} + x^5 + 1$) erfolgen, vor Beginn des Sendens der damit zu sichernden Daten muß der CRC-Generator zurückgesetzt werden, nach dem Laden mindestens der ersten 2 Sendebits muß das Flip-Flop Sendeleerlauf/Blockende rückgesetzt werden (Schreibregister \emptyset)
- Nach der Freigabe des Empfängers (die erst am Ende der Programmierung erfolgen darf) befindet sich dieser im Funkbetrieb, bis die Zeichensynchronisation durch Empfang von Synchronzeichen oder externes Signal erreicht wurde.
- Die CRC-Berechnung bei Empfang kann für jedes empfangene Zeichen erlaubt oder gesperrt werden unter Nutzung der Tatsache, daß die CRC-Berechnung mit einer Verzögerung von einem Zeichen erfolgt.

SIO U856D Funktion

Synchrone Datenformate:

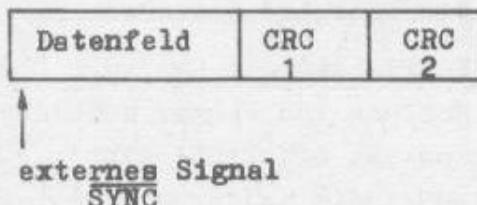
Monosync:

Synchron- Byte	Datenfeld	CRC 1	CRC 2
-------------------	-----------	----------	----------

Bisync:

Synchron- Byte 1	Synchron- Byte 2	Datenfeld	CRC 1	CRC 2
---------------------	---------------------	-----------	----------	----------

Externe Synchronisation



SDLC/HDL.C

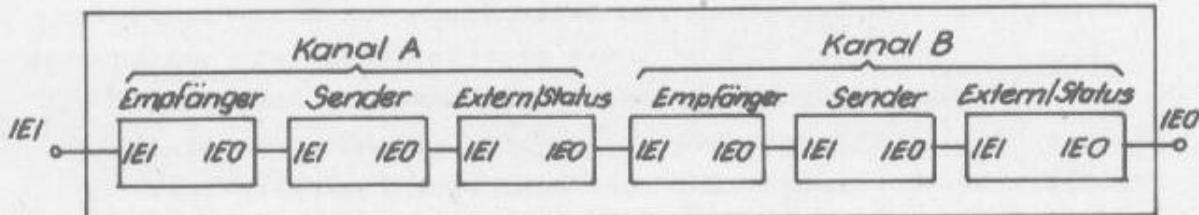
- Im Ruhezustand werden bei freigegebenem Sender Fl. ge sendet, bei gesperrtem Sender liegt TxD auf 1-Pegel.
- Die CRC-Erzeugung beim Senden ist wie folgt zu steuern:
 - Rücksetzen Sende-CRC-Generator (SR0)
 - Ausgabe der ersten 2 Datenbytes
 - Rücksetzen des Sendeleerlauf/Blockende-Flip-Flops (SR0)
 - Ausgabe der restlichen Datenbytes.
- Wenn keine Datenbytes mehr ausgegeben werden, wird automatisch die Prüfbitfolge gesendet.
- Das Rücksetzen des Empfangs CRC-Prüfers erfolgt automatisch durch den Empfang von Flags, nach dem vollständigen Empfang eines Rahmens ist das Prüfergebnis gültig, außerdem ist zu diesem Zeitpunkt die Kodierung des I-Feld-Rests gültig.
(Verarbeitung von Rahmen mit nicht ganzzahliger Byteanzahl)
- Das Senden von Rahmen mit nicht ganzzahliger Bytezahl ist möglich; die Anzahl der von einem ausgegebenem Byte zu sendenden Bits kann während des Sendens geändert werden, wobei zu beachten ist, daß die gewünschte Bitanzahl bei der Übernahme des jeweiligen Zeichens in das Sendeschreibregister gültig sein muß.
- Auf ein Kommando hin wird eine Abbruchsequenz gesendet 8-13 1-Bits, anschließend Flagsenden), Daten im Puffer und Schreibregister gehen verloren.
- Der Empfang einer Abbruchsequenz beeinflußt das entsprechende Statusbit, Beginn und Ende der Abbruchsequenz lösen eine externe Statusunterbrechung aus, falls erlaubt.
- Es kann programmiert werden, daß nur Rahmen mit einer bestimmten vorgewählten Adresse und der Globaladresse 11111111 empfangen werden (d.h. Unterbrechungen auslösen).
- Wenn ein Rahmen nicht bis zu Ende empfangen werden soll, können die Daten durch Setzen des Zustandes "Suche nach Synchronzeichen" (SR3) übergangen werden.
- Eine Arbeit mit Paritätsbit kann nur bei nicht gleichzeitigem Senden und Empfangen erfolgen, da Parität nicht getrennt für Sender und Empfänger aktivierbar (z.B. Adresse muß ohne Parität gesendet werden).
- Nach 5 1-Bits wird beim Senden eine Ø eingefügt, damit Daten von Flags unterschieden werden können. Diese Ø-Bits werden beim Empfang wieder ausgebündet.

SDLC/HDLC-Datenformat:

Flag 01111110	Adresse 8-Bit	Datenfeld n Bit	CRC 1 8Bit	CRC 2 8Bit	Flag 01111110
------------------	------------------	--------------------	------------------	------------------	------------------

E/A Betriebsweisen:

- Abfragebetrieb Alle Unterbrechungen sind gesperrt, Leseregister Ø enthält die Meldung über eingegangene Daten bzw. Leerung des Sendepuffer. Leseregister 1 enthält Fehlermitteilungen.
- Unterbrechungsbetrieb: Jeder SIO Kanal verfügt über 3 Quellen von Unterbrechungen:
1) Empfänger
2) Sender
3) Externe/Statusänderung
Diese Unterbrechungsquellen bilden intern bereits eine Prioritätskette, wobei der Empfänger von Kanal A die höchste Priorität hat:



Empfängerunterbrechungen: können ausgelöst werden entweder beim ersten Zeichen einer Datenfolge (sinnvoll z.B., wenn die folgenden Daten durch DMA eingelesen werden) oder bei jedem Zeichen.

Wenn eine Empfängerunterbrechung erlaubt ist, tritt noch eine weitere Unterbrechungsursache auf:

spezielle Empfangsbedingung

Das kann sein:-Paritätsfehler

- Empfängerüberlauf
- Zeichenrahmenfehler
- Rahmenende (SDLC/HDLC)

Sendunterbrechung: tritt auf, wenn Sendepuffer geleert wurde und neue Daten aufnehmen kann.

Unterbrechung durch externe/Statusänderung: ausgelöst durch

Zustandsänderungen (0→1, 1→0) an DCD
CTS
SYNC

- außerdem durch:
- Senderleerung (Underflow), nachdem keine Daten zum Senden nachgeliefert werden
 - Beginn und Ende eines BREAK-Signals bzw. einer Abbruchsequenz

Unterbrechungsvektor: ändert sich je nach Ursache, wenn programmiert wurde "Status modifiziert Unterbrechungsvektor" (SR1).

Damit kann direkt das behandelnde Programm für eine bestimmte Unterbrechungsursache aktiviert werden.

Es werden unterschiedliche Vektoren erzeugt für

- Empfangenes Zeichen verfügbar
(erstes Zeichen oder jedes Zeichen)
- Spezielle Empfangsbedingung
- Sendepuffer leer
- Externe/Statusänderung

(unterschiedlich für beide Kanäle)

. Zugriffssynchronisierung durch WAIT: Die CPU wird beim Lesen bzw. Schreiben von Daten durch das WAIT-Signal von der SIO so lange blockiert, bis ein empfangenes Zeichen gelesen wurde bzw. der Sendepuffer ein zu sendendes Zeichen aufnehmen konnte. (Vorrangig für sehr hohe Übertragungsraten).

. Ein-/Ausgabe durch DMA: Der Datentransfer zwischen dem Speicher und der SIO erfolgt nicht durch die CPU, sondern durch eine Schaltung zum direkten Speicherzugriff (DMA). Die SIO stellt dabei das Signal READY zur Verfügung, das deren Bereitschaft zur Datenübernahme bzw.-Übergabe signalisiert.

In den beiden letztgenannten Betriebsweisen kann ein Kanal nicht gleichzeitig senden und empfangen.

SIO U856D Programmierung

- Die gewünschte Funktion des SIO-Bausteins wird durch Ausgaben von Steuerbytes zu den Schreibregistern des Bausteins ausgewählt. Die Programmierung der Steuerregister kann effektiv mit Block-E/A-Befehlen erfolgen.
- Die Reihenfolge der Programmierung der Schreibregister ist nicht willkürlich, sie muß so erfolgen, daß die grundlegende Betriebsartenauswahl zuerst erfolgt und keine vorzeitigen Funktionsfreigaben auftreten.

Eine mögliche Reihenfolge ist SR 2 (Unterbrechungsvektor-nur im Kanal B

SR 4
SR 6 } nur bei Synchronen Betriebs-
SR 7 } weisen erforderlich
SR 5
SR 3
SR 1

- Der Zugriff zu den Lese- und Schreibregistern außer LRØ und SRØ erfordert eine vorherige Adressierung durch eine Ausgabe in SRØ
- Nach erfolgtem Zugriff auf ein Register + SRØ oder nach einem Rücksetzen (Kommando oder RESET-Signal) wird beim nächsten Lesen oder Schreiben immer auf LRØ bzw. SRØ zugegriffen.
- Gleichzeitig mit dem Kommando "Kanal rücksetzen" kann kein Register + 0 adressiert werden, nach diesem Kommando darf erst ≥ 4 Takte später wieder eine Schreiboperation in den betreffenden Kanal erfolgen.
- Beim Auftreten einer Änderung des Status (z.B. Modemsteuerleitung, Abbruchbedingung) werden die Statusinformationen in LRØ gespeichert, dadurch werden auch sehr kurze Änderungen registriert. Um die Übernahme neuer Statusänderungen zu erlauben, muß (u.a. nach jeder Behandlung der Unterbrechung "externe oder Statusänderung") das Kommando "Rücksetzen der externen und Statusunterbrechungen" (SRØ) gegeben werden.
- Wenn die Betriebsart "Unterbrechung nur beim ersten empfangenen Zeichen" programmiert ist, muß diese Unterbrechung nach jedem vollständig empfangenen Informationsblock erneut erlaubt werden mit dem Kommando "Unterbrechung erlaubt beim ersten empfangenen Zeichen".

Schreibregister:

Schreibregister Ø: Registeradressierung, Kommandos, CRC-Behandlung

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

0	0	0	0	0	0	0	0	Register Ø
0	0	0	0	0	1	0	1	Register 1
0	0	0	0	1	0	0	1	Register 2
0	0	0	0	1	1	1	1	Register 3
0	0	0	0	1	0	0	0	Register 4
0	0	0	0	1	0	1	0	Register 5
0	0	0	0	1	1	0	1	Register 6
0	0	0	0	1	1	1	1	Register 7

Adressierung des nächsten zu lesenden oder zu schreibenden Registers

Kommandos:

0	0	0	0	0	0	0	0	keine Aktion (z.B. nur Adressierung)
0	0	0	0	0	1	0	1	Sende SDLC-Abbruchsequenz (8...13x1)
0	1	0	0	0	0	0	0	Rücksetzen der externen und Statusunterbrechungen
0	1	1	0	0	0	0	1	Kanal rücksetzen
1	0	0	0	0	0	0	0	Unterbrechung erlaubt beim ersten empfangenen Zeichen
1	0	1	0	0	0	0	1	Sendeunterbrechung unterdrücken, bis wieder ein Zeichen in den SIO-Baustein geladen wurde
1	1	0	0	0	0	0	0	Fehlerstatus (in Leseregister 1) rücksetzen, sollte nach jeder Auswertung der Fehler erfolgen
1	1	1	0	0	0	0	1	Nur im Kanal A: Gleiche Wirkung wie RETI-Befehl von CPU

0	0	0	0	0	0	0	0	keine Aktion
0	1	0	0	0	0	0	0	Empfangs-CRC-Prüfer rücksetzen
1	0	0	0	0	0	0	0	Sende-CRC-Generator rücksetzen
1	1	0	0	0	0	0	0	Sendeleerlauf/Blockende-Flip-Flop rücksetzen

Schreibregister 1: Unterbrechungs- und WAIT/READY-Steuerung

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

1	0	0	0	0	0	0	0	1 - Externe Unterbrechungen erlauben (Pegeländerungen an DCD, CTS, SYNC, Abbruchbedingung, Beginn des CRC oder Synchronzeichensenders)
1	0	0	0	0	1	0	0	1 - Sendeunterbrechungen erlauben (Sendepufferleerung)
0	0	0	0	0	1	0	0	1 Status modifiziert Unterbrechungsvektor
0	0	0	0	1	0	0	0	Empfangsunterbrechungen gesperrt
0	0	0	1	0	0	0	0	Empfangsunterbrechungen nur beim ersten Zeichen
1	0	0	1	0	0	0	0	Empfangsunterbrechungen bei jedem Zeichen, Parität ändert Vektor
1	0	1	0	0	0	0	0	Empfangsunterbrechungen bei jedem Zeichen, Parität ändert Vektor nicht
1	1	0	0	0	0	0	0	1 - WAIT/READY, wenn Empfangspuffer leer
0	1	0	0	0	0	0	0	0 - WAIT/READY, wenn Sendepuffer voll
1	1	1	0	0	0	0	0	1 - WAIT/READY ist READY für direkten Speicherzugriff
0	1	1	1	0	0	0	0	0 - WAIT/READY ist WAIT für CPU
1	1	1	1	0	0	0	0	1 WAIT/READY wird benutzt
0	1	1	1	1	0	0	0	0 WAIT/READY = 1, wenn D6 = 1 bzw. hochohmig, wenn D6 = 0

Schreibregister 2: Unterbrechungsvektor

D7	D6	D5	D4	D3	D2	D1	D0	
V7	V6	V5	V4	V3	V2	V1	V0	
Wenn "Status beeinflußt Vektor" programmiert ist, werden bei einer Unterbrechungsannahme V1-V3 entsprechend folgender Tabelle modifiziert:								
0 0 0 Sendepuffer leer								
0 0 1 Externe bzw. Statusunterbrechung								
0 1 0 Empfangenes Zeichen steht bereit								
0 1 1 Besondere Empfangsbedingung								
1 0 0] Kanal B								
1 0 1 } analog für Kanal A								
1 1 0]								
1 1 1]								

Besondere Empfangsbedingung: Paritätsfehler
Überlauf
CRC /Rahmenfehler
Rahmenende (SDLC)

Schreibregister 3: Empfangslogik

D7	D6	D5	D4	D3	D2	D1	D0	
1 Empfang erlauben								
1 Synchronzeichen nicht in Empfangs- puffer laden, CRC-Berechnung wird durch Synchronzeichen nicht gestoppt								
1 Adressensuche (SDLC), alle Blöcke, die nicht die programmierte oder die Global- adresse 1111 1111 haben, werden übergangen (erzeugen keine Unterbrechungen)								
1 CRC-Berechnung einschalten, beginnt mit Über- nahme des nächsten Zeichens, vom Empfangs- schieberegister in den Puffer								
1 Suche nach Synchronzeichen, bei Verlust der Zeichensynchronisation oder zum Übergehen nicht benötigter Blöcke in SDLC								
1 Bereitschaft durch externes Signal, DCD und CTS er- lauben Empfang und Senden								
0	0	5	Bit je Zeichen] beim Empfang				
0	1	7	Bit je Zeichen] beim Empfang				
1	0	6	Bit je Zeichen] beim Empfang				
1	1	8	Bit je Zeichen] beim Empfang				

Schreibregister 4: Betriebsarten

D7	D6	D5	D4	D3	D2	D1	D0	
							1	- Paritätsbit beim Senden angefügt, beim Empfang erwartet
						1	= gerade Parität	
					0	0	0 = ungerade Parität	
				0	0	0	= Synchronbetrieb	
			0	1	0	1	= 1 Stopbit je Zeichen	
			1	0	1	0	= 1 1/2 Stopbits je Zeichen	
			1	1	1	1	= 2 Stopbits je Zeichen	
		0	0	0	0	0	8-Bit-Synchronzeichen	
		0	1	0	0	1	16-Bit-Synchronzeichen	
	1	0	0	0	0	0	SDLC-Betrieb (0111110-Flag)	
	1	1	0	0	0	0	Externe Synchronisation	
0	0	0	0	0	0	0	= x1 - Takt	
0	1	0	0	0	0	0	= x 16 - Takt	
1	0	0	0	0	0	0	= x 32 - Takt	
1	1	0	0	0	0	0	= x 64 - Takt	

Für Synchronbetrieb ist x1 - Takt notwendig, bei Asynchronbetrieb und x1 -Takt muß die Bitsynchronisation extern erfolgen.

Schreibregister 5: Sendelogik

D7	D6	D5	D4	D3	D2	D1	D0	
								1 = Sende-CRC-Berechnung erlauben RTS-Bit, Steuerung für RTS-Anschluß CRC 16 Ø CRC-CCITT (SDLC/HDLC) } Berechnungsvor- schrift für Prüf- bitfolge
								1 = Senden erlauben 1 = Sende Abbruchzustand (TxD-Anschluß auf 1)
0	0	5 oder weniger Bit je Zeichen						
0	1	7 Bit je Zeichen						
1	0	6 Bit je Zeichen						
1	1	8 Bit je Zeichen						
								DTR-Bit, Steuerung für DTR-Anschluß

Datenformat zum Senden von 5 und weniger Bits je Zeichen:

D7	D6	D5	D4	D3	D2	D1	D0	
1	1	1	1	0	0	0	D	1 Bit
1	1	1	0	0	0	D	D	2 Bit
1	1	0	0	0	D	D	D	3 Bit
1	0	0	0	D	D	D	D	4 Bit
0	0	0	D	D	D	D	D	5 Bit

Schreibregister 6: Synchronzeichen Bit Ø-7 bzw. SDLC-Adresfeld

D7	D6	D5	D4	D3	D2	D1	DØ
S7	S6	S5	S4	S3	S2	S1	SØ

Schreibregister 7: Synchronzeichen Bit 8-15 bzw. bei
SDLC: 01111110 (Flag)

D7	D6	D5	D4	D3	D2	D1	DØ
S15	S14	S13	S12	S11	S10	S9	S8

Statusregister (Leseregister)

Leseregister #

D7	D6	D5	D4	D3	D2	D1	D0
							1 - mindestens ein Zeichen im Empfangspuffer verfügbar
						1 - Unterbrechung in SIO aktiv (beliebige nur in Kanal A, in Kanal B ständig #)	
					1 - Sendepuffer leer (außer bei CRC-Senden)		
			32D, $\overline{D3D}$ invertiert, wird gespeichert bei einer Externen/Statusänderung (DCT, CTS, SYNC/HUNT, BRE K, CRC/SYNC-Senden)				
			1 = Suche nach Synchronzeichen				SYNC/HUNT-Statusbit
			0 = Zeichensynchronisation erfolgt				
			Asynchronbetrieb: SYNC - Eingangsanschluß CTS, CTS invertiert				
			1 - wenn außerdem D2=0: CRC wird gerade gesendet				
			D2=1: Synchronzeichen werden gesendet				
1 -			BRE K in synchronbetrieb erkannt bzw. bruchsequenz ($\geq 7 \times 1$) bei SDLC erkannt				

Leseregister 1

D7	D6	D5	D4	D3	D2	D1	D0	
								1 = Sendepuffer und -Schieberegister vollständig geleert (Asynchronbetrieb); bei Synchronbetrieb immer 1
								Länge des I-Feldes bei SDLC (nur am Rahmenende gültig)
								<i>I-Bits im letzten Byte</i> <i>I-Bits im vorletzten Byte</i>
								1 0 0 0 3 0 1 0 0 4 1 1 0 0 5 0 0 1 0 6 1 0 1 0 7 0 1 1 0 8 → ganzzahlige Byteanzahl 1 1 1 1 8 0 0 0 2 8
								1 = Paritätsfehler } wird erst durch Kommando "Fehlerstatus rücksetzen" (SR0) gelöscht 1 = Empfängerüberlauf }
								= Zeichenformatfehler bei Asynchronbetrieb CRC-Fehler bei Synchronbetrieb
								1 = Rahmenende, bei SDLC sind CRC-Prüfergebnis und I-Feld-Längenkodierung gültig

Leseregister 2 - nur im Kanal B

D7	D6	D5	D4	D3	D2	D1	D \emptyset
V7	V6	V5	V4	V3	V2	V1	V \emptyset

- Augenblicklich gültiger Unterbrechungsvektor,
wenn "Status beeinflußt Vektor"
programmiert und keine Unterbrechung anliegt, gilt
V3, V2, V1 = 011

Grenzwerte und allgemeine statische Kennwerte wie CPU

statische Kennwerte und Kapazitäten - SIO

Kenngröße	Zeichen	Min.	Max.	Einheit	Bemerkung
Eingangskapazität	C_I		7		bei $T_a=25^\circ C$ $f=0,5 \dots 2 MHz$
Takteingangskapazität	C_C		50	pF	
Ausgangskapazität	C_O		14		
Stromaufnahme	I_{CC}		140	mA	bei $U_{CC}=5,25 V$ $a=25^\circ C$

Dynamische Betriebsbedingungen:

Kenngröße	Zeichen	Min.	Max.	Einheit
Takt	wie CPU			
$CE, C/D$ Setzzeit zu \sqcup -Takt, $\overline{IORQ}, B/A$ (Lese- oder Schreib- zyklus)	$t_{SC}(CE)$	160		
$D0-D7$ Setzzeit zu \sqcup -Takt	$t_{SC}(D)$	50		
IEI Setzzeit zu \sqcup - \overline{IORQ} bei Unterbrechungs- annahme	$t_{S(IEI)}$	200		
RD Setzzeit zu \sqcup -Takt	$t_{SC(RD)}$	240		
$M1$ Setzzeit zu \sqcup -Takt	$t_{SC(M1)}$	210		ns
CTS, DCD Dauer 1 - Pegel	$t_W(PH)$	200		
$SYNC$ Dauer 0 - Pegel	$t_W(PL)$	200		
$SYNC$ Vorbereitungszeit bis 1) \sqcup - \overline{RxC} (externe Sync)	$t_{SL(SY)}$	100		
RxC Periode	$t_C(TRC)$	400	∞	
2) Dauer 1 - Pegel	$t_W(TRCH)$	180	∞	
2) Dauer 0 - Pegel	$t_W(TRCL)$	180	∞	
RxD Setzzeit zu \sqcup - \overline{RxC} (x1-Betrieb)	$t_S(RxD)$	0		
RxD Haltezeit nach \sqcup - \overline{RxC} (x1-Betrieb)	$t_H(RxD)$	140		
Alle Haltezeichen für die an- gegebenen Setzzeichen	t_H	0		

1) SYNC - Impulsbreite muß \geq RxC -Periode sein

2) In allen Betriebsarten muß der Systemtakt C mindestens um den Faktor 4,5 höher sein als die maximale Datenrate.

SIO U856P elektrische Kennwerte

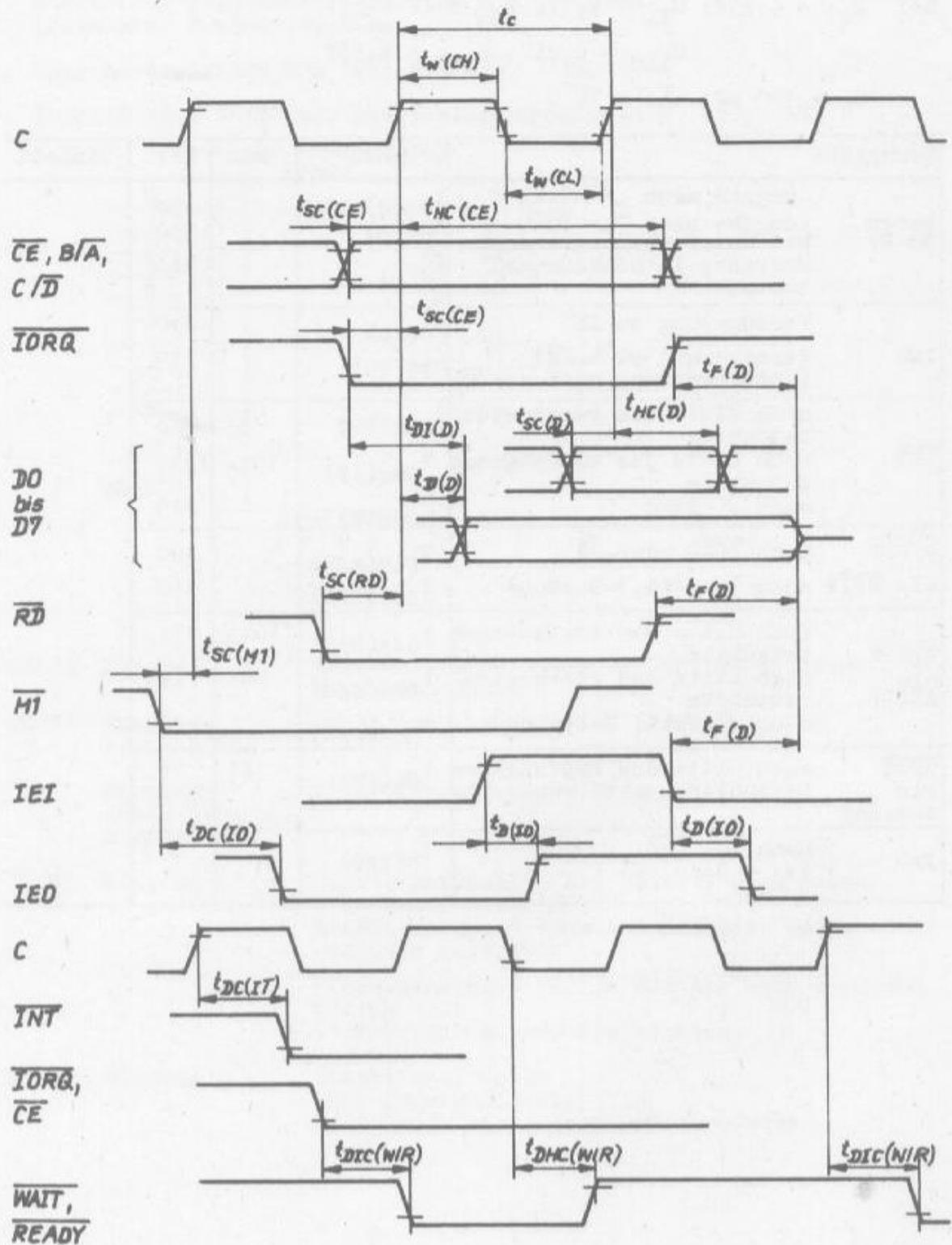
Verzögerungszeiten

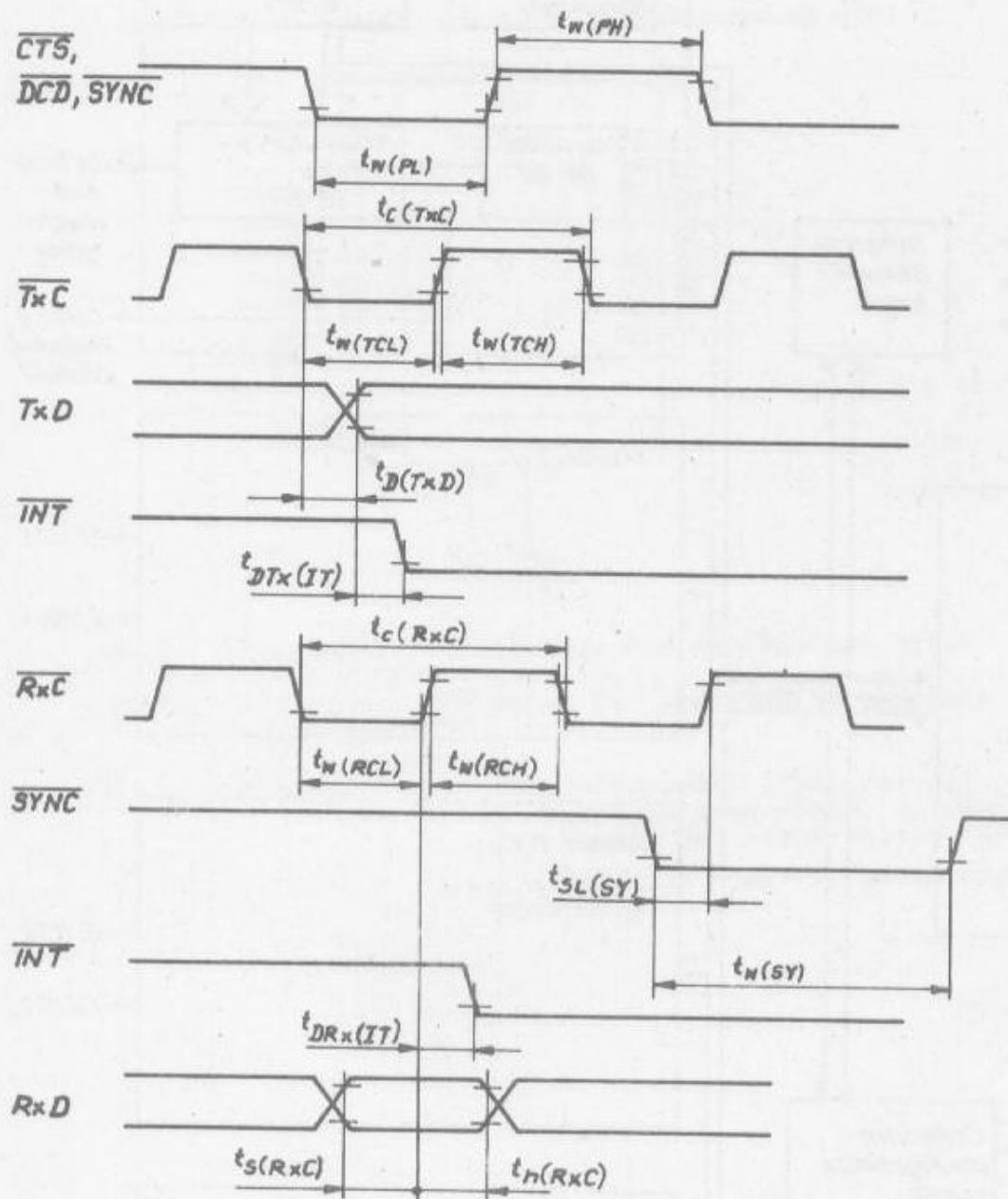
bei $U_{CC} = 4,75V$; $U_{IL} = 0,8V$; $U_{IH} = 2V$

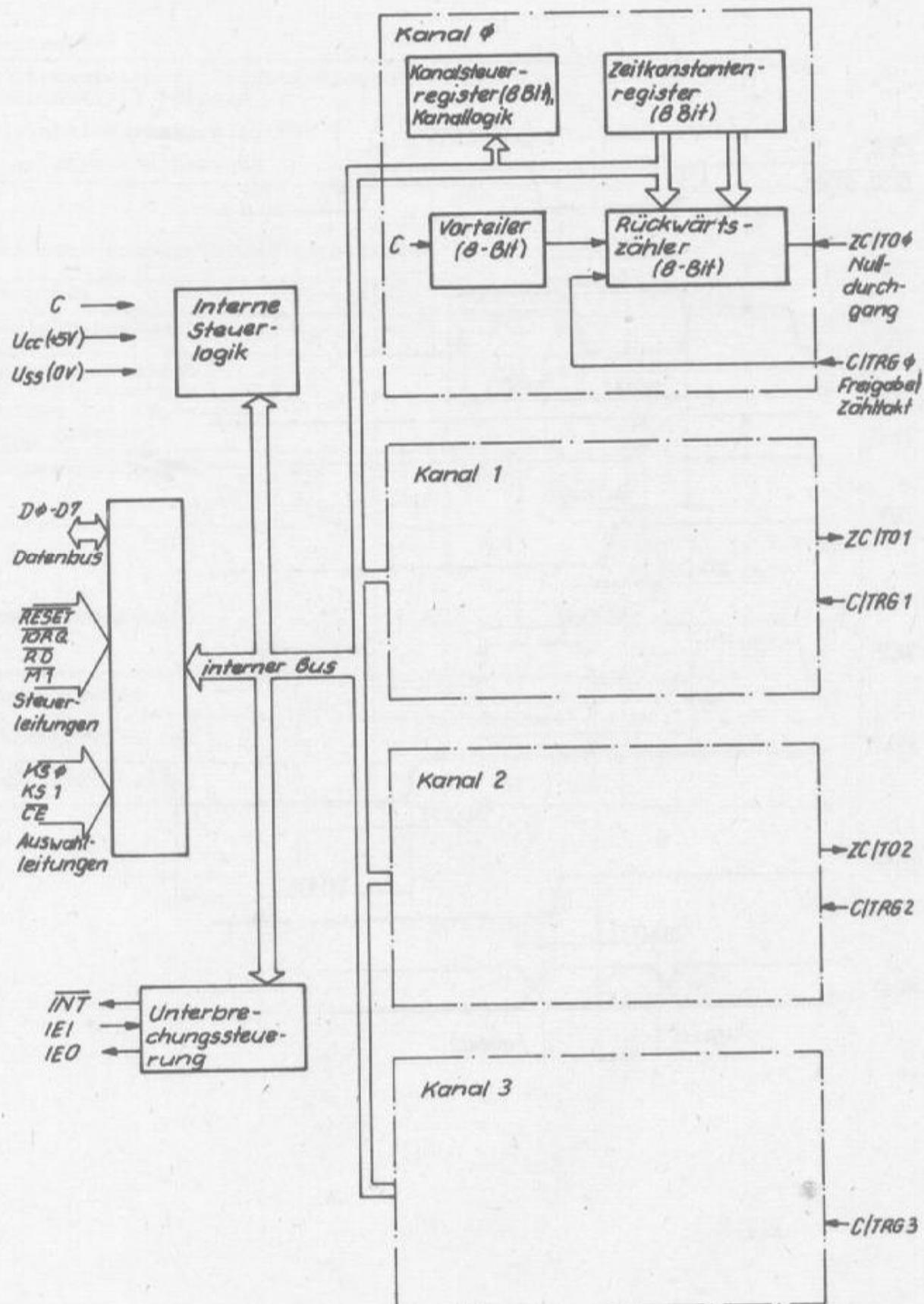
$U_{ILC} = 0,45V$; $U_{IHC} = 4,55V$

$C_L = 100 \text{ pF}$; $\mathcal{T}_a = 70^\circ\text{C}$

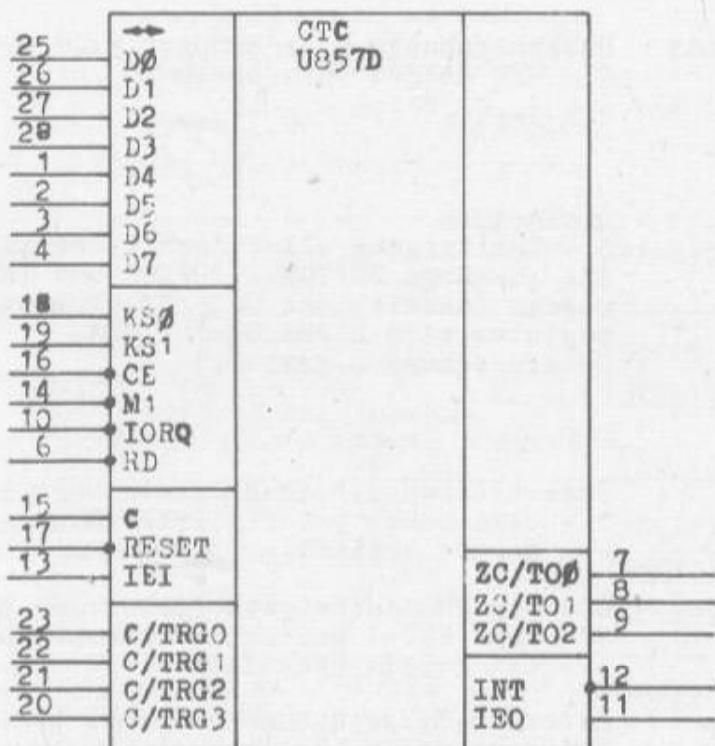
Kenngröße	Zeichen	Min	Max	Einheit
Daten D0-D7	Ausgabe nach \sqcup -Takt Ausgabe nach \sqcup -IORQ bei Unterbrechungsannahme Übergang in hochohmigen Zustand	$t_D(D)$ $t_{DI}(D)$ $t_F(D)$	490 350 240	
IEO	Verzögerung zu IEI Verzögerung zu \sqcup -M1 (Unterbrechung noch vor M1)	$t_{D(IO)}$ $t_{DC}(IO)$	210 310	
INT	nach Mitte des gesendeten Datenbits nach Mitte des empfangenen Datenbits nach \sqcup -Takt	$T_{DTx(IT)}$ $T_{DRx(IT)}$ $T_{DC(IT)}$	$5t_C$ $10t_C$ 210	$9t_C$ $13t_C$ ns
W/RDY	nach IORQ oder CS als WAIT nach \sqcup -Takt, 1-Zustand	$T_{DIC(W/R)}$ $T_{DHC(W/R)}$	190 160	
W/RDY als READY	nach Mitte des empfangenen Datenbits nach Mitte des gesendeten Datenbits nach \sqcup -Takt, 0-Zustand	$t_{DRx(W/R)}$ $t_{DTx(W/R)}$ $t_{DLC(W/R)}$	$10t_C$ $5t_C$ 130	$13t_C$ $9t_C$
SYNC als Ausgang	nach Mitte des empfangenen Datenbits	$t_{DL(SY)}$	$4t_C$	$7t_C$
TxD	nach \sqcup -TxC (x1 - Betrieb)	$t_D(TxD)$		410







Logisches Schaltbild



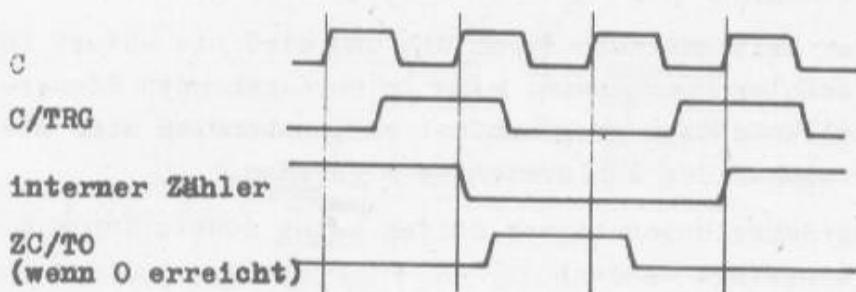
D0 - D7	Ein-/Ausgänge (tri-state)	Systemdatenbus
KS0, KS1	Eingänge	Kanalauswahl, oft mit A0, A1 verbunden 00 - Kanal 0 01 - Kanal 1 10 - Kanal 2 11 - Kanal 3
CE	Eingang	Baustein auswahl (Chip Enable) mit E/A-Adressdekodierung zu verbinden 0-Baustein wird gelesen oder geschrieben
M1	Eingang	Operationskode-Lesezyklus der CPU (Machine Cycle 1) 0 - zusammen mit IORQ: Unterbrechungsannahme durch CPU, CTC kan Unterbrechungsvektor auf Datenbus geben (wenn INT von CTC angemeldet und Priorität vorhanden) außerdem: Erkennen des RETI-Befehls durch CTC
IORQ	Eingang	Ein-/Ausgabeanforderung der CPU (Input/ Output/Request) (- E/A-Baustein lesen, schreiben oder Unterbrechungsannahme
RD	Eingang	Lesesignal (Read) 0 - Lesen von Speicher oder E/A-Baustein

<u>INT</u>	Ausgang mit offenem Drain	Unterbrechungsanforderung (Interrupt) 0 - CTC meldet Unterbrechung an
<u>RESET</u>	Eingang	Rücksetzen 0 - Zählvorgang aller Kanäle abbrechen, die Ausgänge ZC/T0 \emptyset ...ZC/T02 und <u>INT</u> werden inaktiv, das Unterbrechungsvektor- register wird nicht beeinflußt, Unterbrechungen gesperrt.
<u>IEI</u>	Eingang	Unterbrechungsfreigabe (Interrupt Enable In) 1 - kein höher priorisierte Baustein wird gerade bedient.
<u>IEO</u>	Ausgang	Unterbrechungsfreigabe (Interrupt Enable Out) 1 - wenn IEI=1 und keine Unterbrechung der CTC gerade behandelt wird.
C/TRG \emptyset ... C/TRG3	Eingänge	Externer Takteingänge (Zähler) bzw. Triggereingang (Zeitgeber) der Kanäle \emptyset ...3
ZC/T0 \emptyset ... ZC/T02	Ausgänge	Nulldurchgang des Zählers bzw. Zeitgeberstand "0" nur für die Kanäle \emptyset , 1, 2

Betriebsarten

Zähler: Die programmierte Flanke am C/TRG-Eingang bewirkt einen Zählschritt des Rückwärtszählers. Die maximale verarbeitbare Taktfrequenz beträgt $f_c/2$ (f_c - Systemtaktfrequenz), außerdem sind bestimmte Mindestlängen des 0- und 1-Zustands einzuhalten.

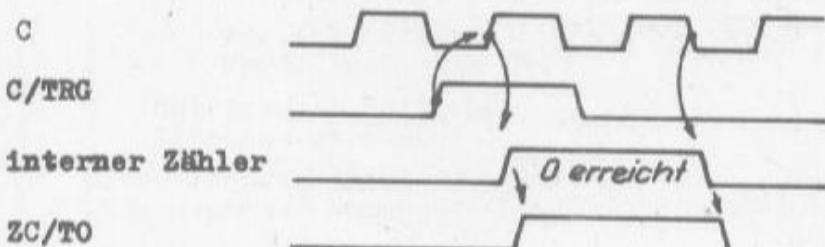
Der interne Zähler arbeitet synchron mit dem Systemtakt.



Zeitgeber: Wenn "externe Freigabe" programmiert wurde, startet der Zeitgebevorgang mit der programmierten Flanke an C/TRG. (Danach hat C/TRG keinen Einfluß mehr). Eine bestimmte Mindestdauer des Impulses ist einzuhalten.

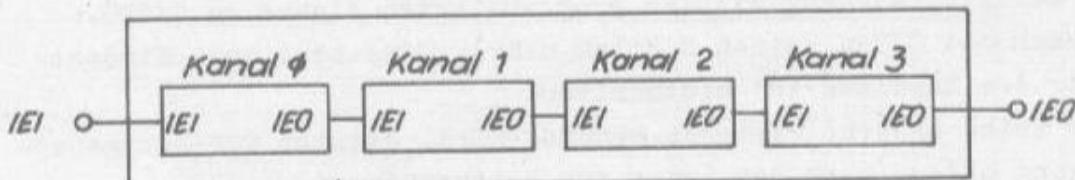
Wenn keine externe Freigabe benutzt wird, startet der Zeitgebervorgang sofort nach dem Laden der Zeitkonstante.

Der Rückwärtszähler zählt den durch 16 oder 256 geteilten Systemtakt.



Nutzung des Zählers/Zeitgebers

- Der Rückwärtszähler kann von der CPU gelesen werden.
- Unmittelbar nachdem er den Zählerstand \emptyset erreicht hat, wird an ZC/TO ein Impuls abgegeben. Falls erlaubt wird eine Unterbrechung ausgelöst.
- Bei jedem Erreichen des Zählerstandes \emptyset wird automatisch die programmierte Zeitkonstante in den Rückwärtszähler geladen, der Zähler-/Zeitgebervorgang läuft erneut an.
- Beim Laden der Zeitkonstante durch die CPU wird sie sofort in den Rückwärtszähler übernommen, wenn im vorangehenden Steuerwort ein Kanalrücksetzen programmiert war, ansonsten wird sie erst nach Erreichen des Zählerstandes \emptyset geladen.
- Bei Lese- und Schreiboperationen dürfen keine zusätzlichen Wartezyklen eingefügt werden!
- Bei der Auslösung von Unterbrechungen hat Kanal \emptyset die höchste Priorität:



- Es ist nur ein Unterbrechungsvektor (über Kanal \emptyset) zu programmieren. Die unterschiedlichen Vektoren für die einzelnen Kanäle werden von CTC gebildet.

Das Betriebsverhalten des CTC-Bausteines wird durch die Ausgabe von Steuerwerten festgelegt:

Unterbrechungsvektor (Nur über Kanal Ø ladbar)

D7	D6	D5	D4	D3	D2	D1	DØ
V7	V6	V5	V4	V3	X	X	O

V7 V6 V5 V4 V3 X X O - Kennzeichnung des Steuerwertes als Unterbrechungsvektor

Bei Auslösung einer Unterbrechung wird von CTC für D1, D2 der auslösende Kanal (binär) eingesetzt:

D2	D1	
0	0	Kanal Ø
0	1	Kanal 1
1	0	Kanal 2
1	1	Kanal 3

Kanalsteuerwort

D7	D6	D5	D4	D3	D2	D1	DØ
							1 - Kennzeichnung des Steuerwertes als Kanalsteuerwort
							1 - Kanal rücksetzen, Fortsetzung der Arbeit nach dem Laden einer Zeitkonstante bzw. eines neuen Kanalsteuerwertes
							0 - Kanal zählt weiter
							1 - Nächstes Steuerwert ist die Zeitkonstante,
							0 - es folgt keine Zeitkonstante
							0 - Zeitgeber läuft sofort nach Laden der Zeitkonstante an
							1 - Zeitgeber läuft nach externer Freigabe an (C/TRG)
							0 - fallende Flanke an C/TRG wirksam
							1 - steigende Flanke an C/TRG wirksam
							0 - Vorteilerfaktor 16 } bei Betriebsart Zeitgeber 1 - Vorteilerfaktor 256 }
							0 Betriebsart Zeitgeber 1 Betriebsart Zähler
							0 Unterbrechung durch CTC-Kanal gesperrt 1 Unterbrechung durch CTC-Kanal freigegeben

Zeitkonstante:

D7	D6	D5	D4	D3	D2	D1	DØ	00000000256
Z7	Z6	Z5	Z4	Z3	Z2	Z1	ZØ	

CTC U857D elektrische Kennwerte

Grenzwerte und allgemeine statische Kennwerte wie CPU

Statische Kennwerte und Kapazitäten - CTC

Kenngröße	Zeichen	Min	Max	Einheit	Bemerkung
Eingangskapazität	C_I		7	pF	$T_a=25^\circ C$ $f=0,5..2MHz$
Takteingangskapazität	C_C		25		
Ausgangskapazität	C_O		14		
Stromaufnahme	I_{CC}		80		
Laststrom der Darlington-Treiber-Ausgänge ZG/TO \emptyset , 1,2	I_{OHD}	1,5	3,8	mA	bei $U_{OH}=1,5V$ $T_a=25^\circ C$ $R_{ext}=390\Omega$

Dynamische Betriebsbedingungen

Kenngröße	Zeichen	Min	Max	Einheit
Takt C			wie CPU	
D \emptyset -D7	Setzzeit zu \sqcap -Takt	$t_{SC(D)}$	60	
KS \emptyset KS1, CE	Setzzeit zu \sqcap -Takt	$t_{SC(KS)}$	160	
IOR \bar{Q}	Setzzeit zu \sqcap -Takt	$t_{SC(IR)}$	250	
M1	Setzzeit zu \sqcap -Takt	$t_{SC(M1)}$	210	
RD	Setzzeit zu \sqcap -Takt (Schreib-oder Unterbrechungsannahmezyklus)	$t_{DCW(RD)}$	115	ns
	Setzzeit zu \sqcap -Takt (Lese- oder M1-Zyklus)	$t_{DCR(RD)}$	240	
C/TRG \emptyset bis C/TRG3	Periode (Zählerbetrieb)	$t_C(CK)$	$2t_C$	
	Setzzeit zu \sqcap -Takt (Zählerbetrieb)	$t_S(SK)$	210	
	Trigger-Setzzeit zu \sqcap -Takt (Zeitgeberbetrieb, Freigabe auf folgende \sqcap -Taktflanke)	$t_S(TR)$	210	
	Anstiegs- und Abfallzeiten	t_r, t_f		
	Dauer 0-Zustand	$t_w(CTH)$	200	
	Dauer 1-Zustand	$t_w(CTL)$	200	
Haltezeiten für die angegebenen Setzzeiten		t_H	0	

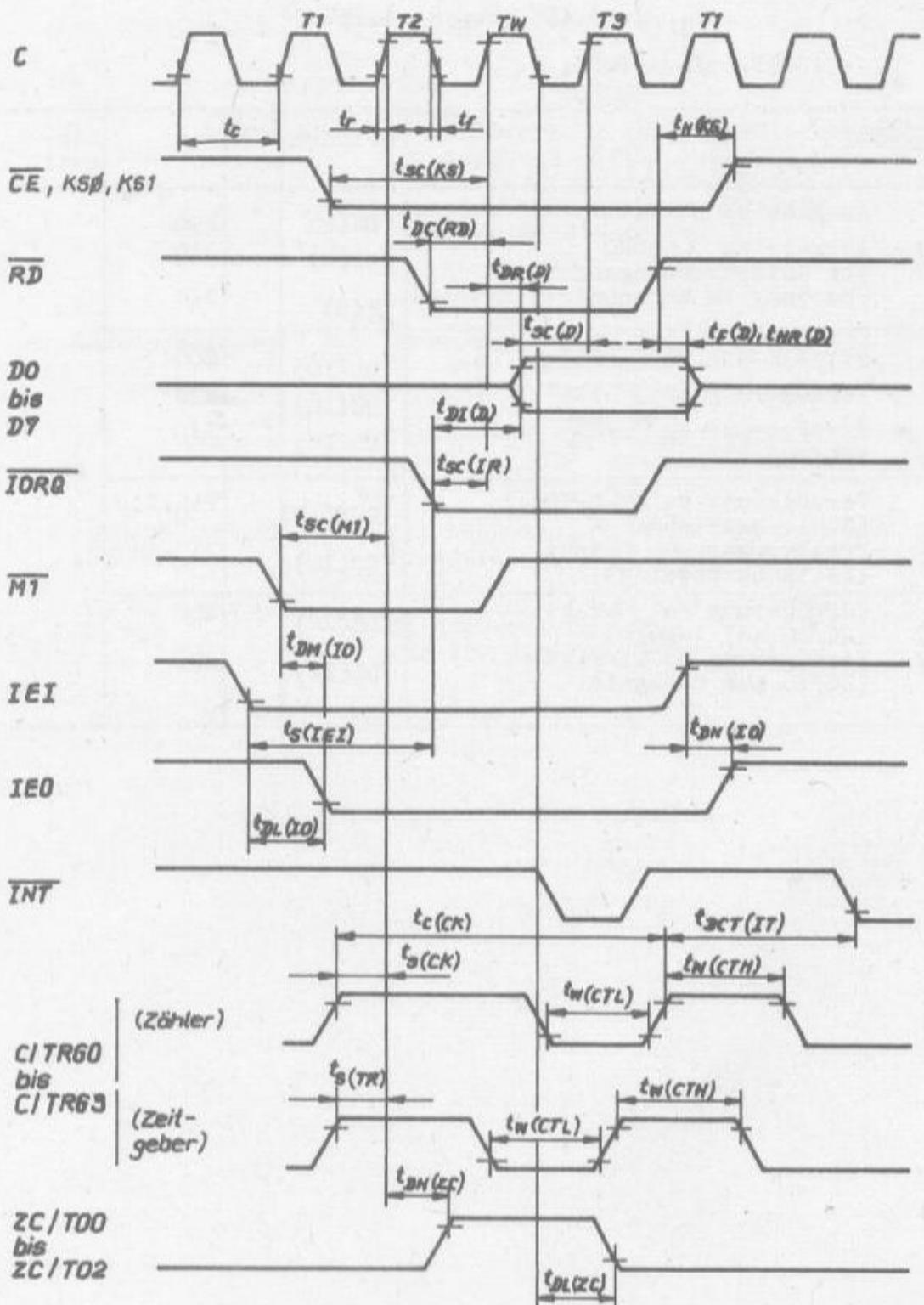
Verzögerungszeiten

bei $U_{CC} = 4,75V$, $U_{IL} = 0,8V$, $U_{IH} = 2V$

$U_{ILC} = 0,45V$, $U_{IHC} = 4,55V$

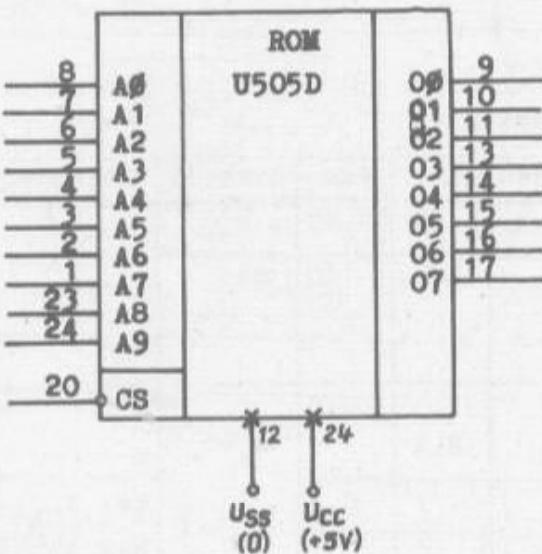
$C_L = 100pF$; $\vartheta_a = 70^\circ C$;

Kenngröße	Zeichen	Min	Max	Einheit
Daten D \emptyset -D7	Ausgabe ab $\sqrt{-}$ -Takt	$t_{DR(D)}$	490	ns
	Ausgabe ab $\sqrt{-}$ -IORQ bei Unterbrechungsannahme	$t_{DI(D)}$	350	
	Übergang in hochohmigen Zustand	$t_F(D)$	240	
IEO	Verzögerung zu $\sqrt{-}$ -IEI	$t_{DL(IO)}$	200	ns
	Verzögerung zu $\sqrt{-}$ -IEI	$t_{DH(IO)}$	230	
	Verzögerung zu $\sqrt{-}$ -M1 während RETI	$t_{DM(IO)}$	310	
INT	Verzögerung zu $\sqrt{-}$ -C/TRG (Zählerbetrieb)	$t_{DCT(IT)}$	$2t_C + 210$	ns
	Verzögerung zu $\sqrt{-}$ -Takt (Zeitgeberbetrieb)	$t_{DC(IT)}$	$2t_C + 210$	
ZC/TO 0,1,2	Verzögerung zu $\sqrt{-}$ -Takt (ZC/TO auf 1-Pegel)	$t_{DH(ZC)}$	200	ns
	Verzögerung zu $\sqrt{-}$ -Takt (ZC/TO auf 0-Pegel)	$t_{DL(ZC)}$	200	



1 KByte-ROM U505D Anschlußbeschreibung

- Maskenprogrammierter Festwertspeicher
- Speicherkapazität 8 x 1024 Bit bzw. 1 KByte
- Zugriffzeit \leq 450ns, eine Betriebsspannung +5V



Anm.: Die Anschlüsse 18, 19, 21 können mit Potentialen $-5V \leq U \leq +12V$ belegt werden.

A0-A9	Eingänge	Adresse des zu lesenden Bytes
Q0-Q7	Ausgänge (tri-state)	Daten der ausgewählten Adresse
CS	Eingang	Bausteinauswahl (Chip Select) 0 - Baustein ausgewählt, an Q0-Q7 liegt der Inhalt der ausgewählten Adresse an. 1 - Baustein inaktiv, Datenausgänge hochohmig.

1 KByte-ROM U505D Elektr.Kennwerte

Grenzwerte:

Betriebsspannung, Eingangsspannungen auf U_{SS} (0-Potential) bezogen	-0,5...+7V
Betriebstemperaturbereich	0...+70°C
Lagertemperaturbereich	-55...+125 °C

Statische Kennwerte und Kapazitäten

Kenngröße	Zeichen	Min	Max	Einheit	Bemerkung
Betriebsspannung	U_{CC}	4,75	5,25	V	
Eingangsreststrom und Reststrom der hoch-ohmigen tri-state-Ausgänge (Betrag)	I_R		10	μA	
Eingangsspannung	U_{IL} U_{IH}	2,4	0,8	V	
	U_{OL}		0,4		bei $I_{OL} = 1,8 \text{ mA}$, $U_{CC} = 5,25 \text{ V}$
Ausgangsspannung	U_{OH}	2,4			bei $I_{OL} = -0,4 \text{ mA}$, $U_{CC} = 4,75 \text{ V}$
Stromaufnahme	I_{CC}		120	mA	bei $U_{CC} = 5,25 \text{ V}$
Eingangskapazität	C_I		10		
Ausgangskapazität	C_O		15	pF	

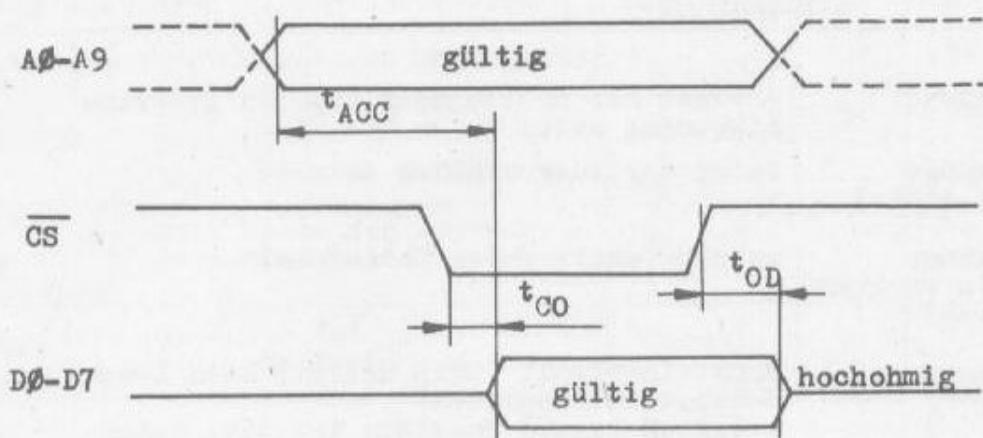
1 KByte-ROM U505D Elektrische Kennwerte

Dynamische Kennwerte

bei $U_{CC} = 4,75 \text{ V}$; $U_{IL} = 0,8 \text{ V}$; $U_{IH} = 2,4 \text{ V}$

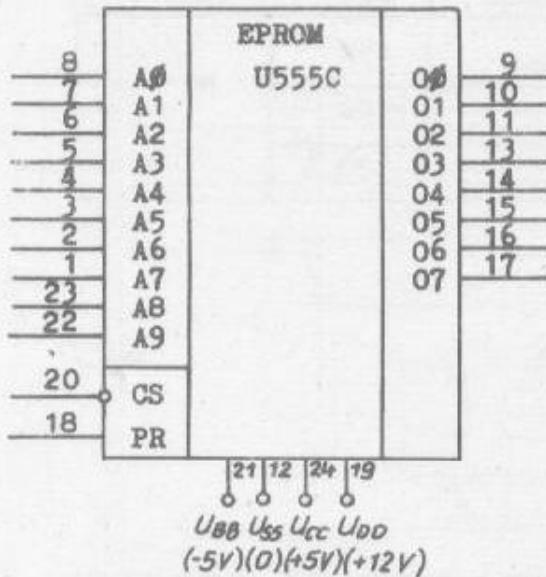
$C_L = 120 \text{ pF}$; $T_a = 70^\circ\text{C}$

Kenngröße	Zeichen	Max	Einheit
Zugriffszeit	t_{ACC}	450	ns
Deselektionszeit	t_{OD}	100	
Selektionszeit	t_{CO}	120	



1 KByte-EPROM U555C Anschlußbeschreibung

- Elektrisch programmierbarer und mit UV-Licht löscherbarer Festwertspeicher
- Speicherkapazität 8×1024 Bit bzw. 1 KByte
- Zugriffszeit ≤ 450 ns, Betriebsspannungen +12V, +5V, -5V



A0-A9	Eingänge	Adresse des zu lesenden bzw. zu programmierenden Bytes
00-07	Ausgänge (tri-state)	Daten der ausgewählten Adresse
	Eingänge (beim Programmieren)	zu programmierender Dateninhalt
CS/WE	Eingang	Bausteinsauswahl (Chip Select) beim Lesen 0-Baustein ausgewählt 1-(TTL-H-Pegel) Baustein inaktiv, Datenausgänge hochohmig Schreibfreigabe (Write Enable) beim Programmieren +12V-Programmieren freigegeben.
PR	Eingang	Programmierimpuls +26V-Programmierung beim Lesen mit 0 oder +5V verbinden

1 KByte-EPROM U555C Funktion

Lesebetrieb:

- . Zu beachten ist die eingeschränkte TTL-Kompatibilität der Eingänge, die Eingangsspannung $U_{IH} \geq 3V$ ist gegebenenfalls durch Widerstände nach +5V zu sichern.
- . Die Bereitstellung der Betriebsspannungen hat so zu erfolgen, daß die Spannung U_{BB} (-5V) gleichzeitig oder vor U_{CC} (+5V) und U_{DD} (+12V) angelegt wird und erst gleichzeitig oder nach dem Abschalten von U_{CC} und U_{DD} abgeschaltet wird. Der Zustand, daß U_{DD} und U_{CC} anliegen, U_{BB} aber nicht, darf nur maximal 10 ms andauern, sonst ist mit einer Zerstörung des Schaltkreises zu rechnen.
- . Der Schaltkreis ist äquivalent zum Typ 2708 sowie anschlußkompatibel zum ROM U505D.

Programmierbetrieb:

- . Im Neuzustand oder nach dem Löschen sind alle Bits mit 1 belegt, beim Programmervorgang werden ausgewählte Bits mit 0 belegt.
- . Zum Programmieren ist der \overline{CS}/WE -Eingang an +12V zu legen, es ist die Adresse anzulegen sowie die zu programmierenden Daten (an 08-07). Anschließend wird auf den Eingang PR ein Programmierimpuls gegeben.
- . Es sind fortlaufend alle 1024 Speicheradressen mit je einem Programmierimpuls zu programmieren.
Dieser Zyklus ist zu wiederholen, bis für jede Adresse eine aufsummierte Programmierzeit von 50 ms erreicht ist.
- . Im Programmierbetrieb ist der eingeschränkte Temperaturbereich $20^{\circ}\text{C} \leq \vartheta_a \leq 30^{\circ}\text{C}$ zu beachten.
- . Die Reihenfolge der Bereitstellung der Betriebsspannungen entspricht der beim Lesebetrieb, insbesondere ist dies beim Stecken bzw. Entfernen aus der Programmierfassung zu beachten.

Lösung:

- . Löschdauer 17...30 min. bei einem Strahlungsfluß von $10...15\text{mWcm}^{-2}$
- . mindestens mögliche Anzahl der Programmier-Löszyklen : 30

1 KByte-EPROM U555C Elektrische Kennwerte

Grenzwerte (Spannungen bezogen auf U_{BB} !):

Betriebsspannung U_{DD}	-0,3...20V
Eingangsspannungen A ₉ -A ₀ -07, CS/WE;	-0,3...15V
Betriebsspannungen U_{SS} , U_{CC}	
Programmierspannung U_{PR}	-0,3...32V
Betriebstemperatur ϑ_a	0...70 °C
Gesamtverlustleistung bei $\vartheta_a = 45$ °C P_V	≤ 1,5W

Statische Kenngrößen und Kapazitäten:

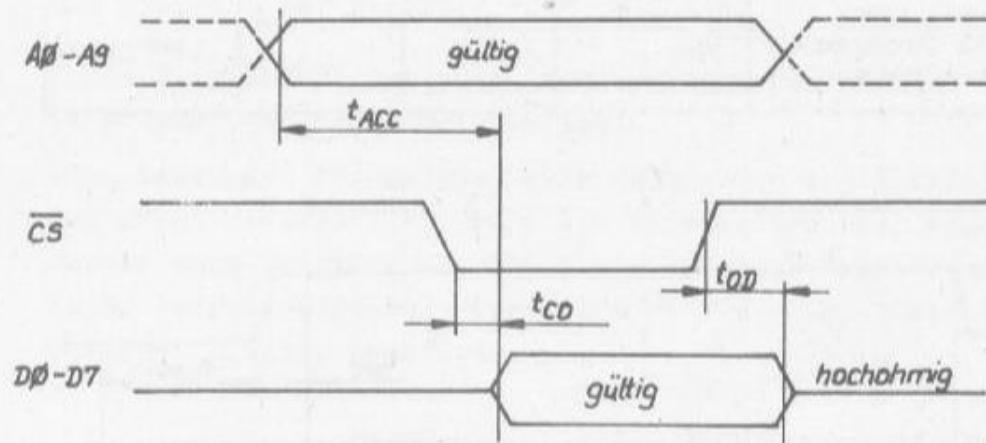
Kenngröße	Zeichen	Min	Max	Einheit	Bemerkung
Betriebsspannungen bezogen auf U_{SS} (0-Potential)	U_{BB} U_{CC} U_{DD}	-5,25 4,75 11,4	-4,75 5,25 12,6	V	
Eingangsspannung	U_{IH} U_{IL}	3 -0,3	$U_{CC} + 0,5$ 0,8	V	
Eingangsspannung an CS/WE für Programmierung U_{IWE}		11,4	12,6		
Ausgangsspannung	U_{OL} U_{OH}	2,4	0,4		bei $I_{OL} = 1,6mA$ bei $I_{OH} = -1mA$
Programmiereingang PR beim Programmieren	U_{PRL} U_{PRH}	0 25	1 27		
Eingangsströme, Ausgangsrestströme (Betrag)	I_R		10	μA	
Stromaufnahme	I_{BB} I_{CC} I_{DD}	-45	10 65	mA	
Programmierstrom (bei $U_{IWE} = 11,4V$)	I_{PR1} I_{PR2}	-3	20		bei $U_{PR} = 26 \pm 1V$ bei $U_{PR} = 1V$
Eingangskapazität (A ₉ -A ₀ , CS/WE)	C_I		6		
Ausgangskapazität (0 ₉ -07) C_O			12	pF	
Eingangskapazität PR	C_{PR}		35		

1 KByte-EPROM U555C Elektrische Kennwerte

Dynamische Kennwerte - Lesebetrieb:

bei $U_{BB} = -4,75V$, $U_{CC} = 4,75V$, $U_{DD} = 11,6V$

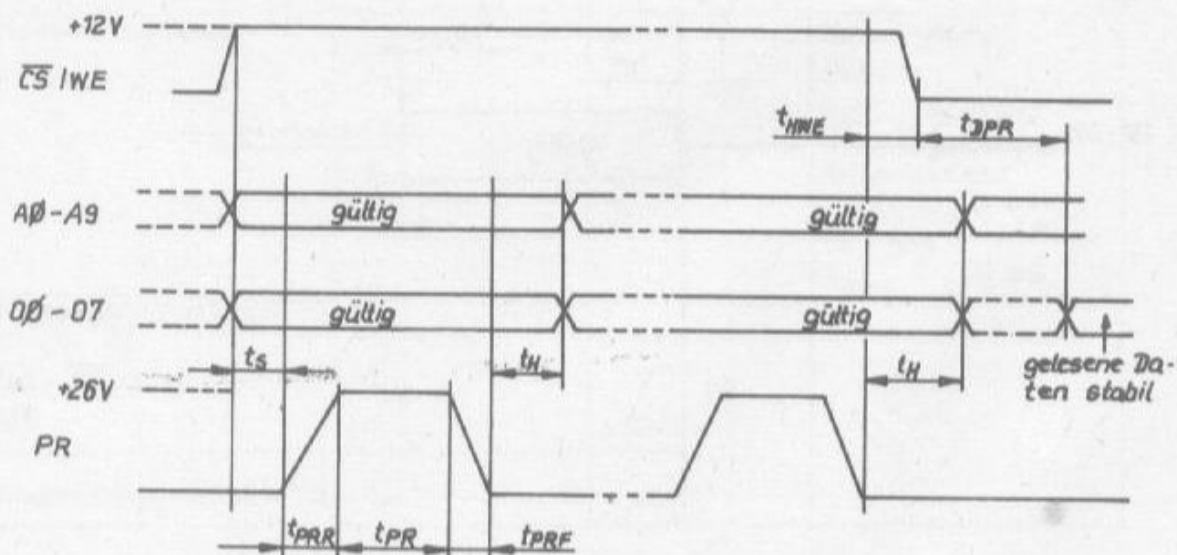
Kenngröße	Zeichen	Max	Einheit
Zugriffszeit	t_{ACC}	450	ns
Deselektionszeit	t_{OD}	120	
Selektionszeit	t_{CO}	120	



1 KByte-EPROM U555C Elektrische Kennwerte

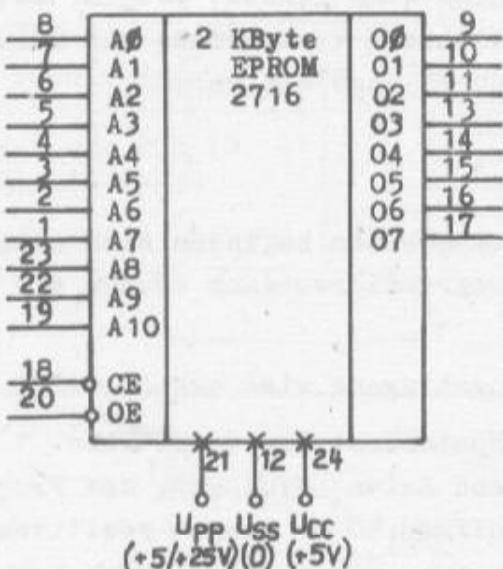
Dynamische Kennwerte - Programmierung:

Kenngröße		Min	Max	Einheit
Setzzeiten für A ₀ -A ₉ , O ₀ -O ₇ , CS/WE vor dem Programmierimpuls	t _S	10		μs
Haltezeit für A ₀ -A ₉ , O ₀ -O ₇ nach dem Programmierimpuls	t _H	1		
Haltezeit für CS/WE nach dem Programmierimpuls	t _{HWE}	0,5		
Verzögerung Übergang von Programmieren zu Lesen für O ₀ -O ₇	t _{DPR}		10	
Programmierimpulslänge	t _{PR}	100	1000	
Programmierimpuls- anstiegs- u. abfallzeit	t _{PRR} , t _{PRF}	0,5	2	
aufsummierte Program- mierzeit je Adresse	t _{PS}	50		ms



2 KByte-EPROM 2716 Anschlußbeschreibung

- mit UV-Licht löscherbarer Festwertspeicher 2048 x 8 Bit
- Zugriffszeit ≤ 450 ns, nur eine Versorgungsspannung +5V



A0-A10	Eingänge	zu lesende oder zu programmierende Adresse
O0-O7	Ausgänge (tri-state)	ausgelesene Daten (Lesebetrieb)
	Eingänge	zu programmierende Daten (Programmierbetrieb)
\overline{CE}/PR	Eingang	Bausteinauswahl (Chip Enable) 0-Baustein ausgewählt Programmierimpulseingang (Programmierbetrieb)
\overline{OE}	Eingang	Ausgangaktivierung 0- am Ausgang liegen Daten an (\overline{CE} muß ebenfalls 0 sein) 1- Ausgänge hochohmig
U_{PP}	Eingang	Programmierspannung +5V - Lesen +25V - Programmieren

2 KByte-EPROM 2716 Funktion

Lesebetrieb:

- Programmierspannungseingang U_{PP} mit +5V verbinden!
- Der Ausgangsfreigabeeingang kann genutzt werden, um bei der \overline{CS} -Dekodierung eine zusätzliche Verknüpfung mit \overline{MREQ} der CPU einzusparen, \overline{MREQ} kann an \overline{OE} angelegt werden.

Programmierbetrieb:

- Im Neuzustand und nach dem Löschen befinden sich alle Datenbits im 1-Zustand, durch den Programmervorgang können sie auf 0 geändert werden.
- An den Programmierspannungseingang wird ständig +25V angelegt.
- Zum Programmieren einer Speicherstelle (8 Bit bzw. 1 Byte) sind die entsprechenden Adressen und Daten anzulegen, das Programmieren erfolgt durch einen einmaligen, 50 ms langen positiven Impuls mit TTL-Pegel an $\overline{CE/PR}$.
Die Speicherstellen können einzeln bzw. in beliebiger Reihenfolge programmiert werden.
- Das Lesen zur Kontrolle kann mit angelegter Programmierspannung $U_{PP} = +25V$ und $\overline{CE/PROG} = 0$, $\overline{OE} = 0$ erfolgen.
- Die Spannung $U_{PP} = +25V$ darf nur anliegen, wenn auch $U_{CC} = +5V$ anliegt, der Schaltkreis sollte deshalb nicht bei anliegender Programmierspannung $U_{PP} = 25V$ in die Fassung gesteckt bzw. aus ihr entfernt werden.

Grenzwerte:

Programmierspannung U_{PP} bezogen auf $U_{SS}(0)$	-0,3...+26,5V
Spannung an den übrigen Anschlüssen bezogen auf $U_{SS}(0)$	-0,3...+6V
Betriebstemperatur	0...70 °C

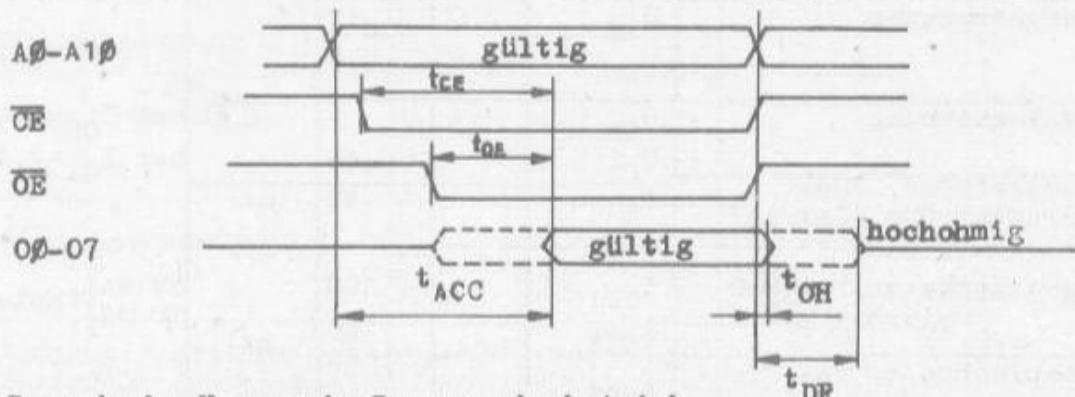
Statische Kennwerte

Kenngröße	Zeichen	Min	Max	Einheit	Bemerkung
Betriebsspannung	U_{CC}	4,75	5,25	V	
Programmierspannung beim Programmieren	U_{PP}	24	26		
Eingangsspannung	U_{IH} U_{IL}	2,0 -0,1	$U_{CC} + 1$ 0,8		
Ausgangsspannung	U_{OH} U_{OL}	2,4	0,45	μA	bei $I_{OH} = 400 \mu A$ bei $I_{OL} = 2,1 mA$
Eingangsströme, Aus- gangsrestströme (Betrag)	I_R		10		
Stromaufnahme aktiv- $\overline{CE}=0$ standby - $\overline{CE}=1$	I_{CC1} I_{CC2}		100 25	mA	57 mA } typisch 10 mA }
Stromaufnahme an U_{PP} während Programmierung	I_{PP1} I_{PP2}		5 30		
Eingangskapazität	C_I		6	pF	
Ausgangskapazität	C_O		12		

2 KByte-EPROM 2716 Elektrische Kennwerte

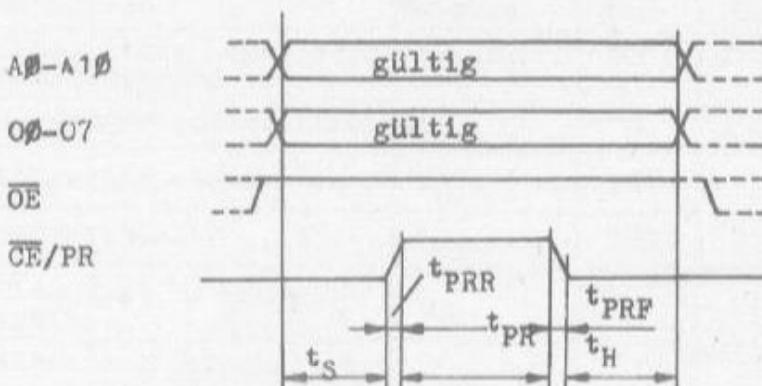
Dynamische Kennwerte - Lesebetrieb:

Kenngröße	Zeichen	Min	Max	Einheit
Zugriffszeit (ab Anliegen der Adresse)	t_{acc}		450	ns
Verzögerung von \overline{CE} bis zur Datengültigkeit	t_{CE}		450	
Verzögerung von \overline{OE} bis zur Datengültigkeit	t_{OE}		120	
Datenhaltezeit nach Adresswechsel, $CE \rightarrow 1$ oder $OE \rightarrow 1$	t_H	0		
Verzögerung nach $OE \rightarrow 1$ bis hochohmiger Zustand an $O\bar{O}-07$	t_{DF}	0	100	



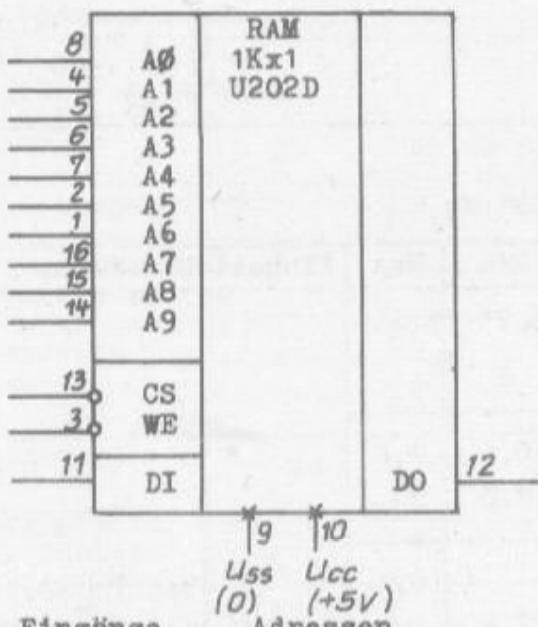
Dynamische Kennwerte-Programmierbetrieb

Kenngröße	Zeichen	Min	Max	Einheit	Bemerkung
Setz- u. Halbzeiten für $A\bar{O}-A10$, $O\bar{O}-07$, \overline{OE}	t_S , t_H	2		μs	Sollwert: 50ms
Programmierimpulslänge an CE/PR	t_{PR}	45	55	ms	
Anstiegs- und Abfallzeiten des Programmierimpulses	t_{PRR} t_{PRF}	5		ns	



Stat. 1 KBit-RAM U202D Anschlußbeschreibung

- Statischer Schreib-Lesespeicher
- Speicherkapazität 1024 x 1 Bit
- Zugriffszeit ≤ 400 ns, eine Betriebsspannung +5V



A0-A9	Eingänge	U_{ss} (0)	U_{cc} (+5V)	Adressen
CS	Eingang	Bausteinauswahl (Chip Select) 0-Baustein kann geschrieben oder gelesen werden, DO führt Daten 1-Baustein inaktiv, DO hochohmig		
DI	Eingang	zu schreibende Daten (Data In)		
DO	Ausgang (tri-state)	gelesene Daten (Data Out)		
WE	Eingang	Schreibfreigabe (Write Enable) 0-Schreiben 1-Lesen		

Funktionstabelle:

	CS	WE	DI	DO
Ruhezustand	1	X	X	hochohmig
Schreiben 0	0	0	0	0
Schreiben 1	0	0	1	1
Lesen	0	1	X	Inhalt der ausgewählten Speicherzelle

- Der Datenausgang DO ist bei Schreiboperationen nicht hochohmig, er kann also nicht direkt mit DI bzw. einem bidirektionalen Datenbus verbunden werden.
- Durch Absenken der Betriebsspannung unter 4,75V auf minimal 2V wird ein "Schlafzustand" erreicht, in dem nur noch ein Daten-Erhält erfolgt, Lese-/Schreiboperationen aber blockiert sind.

Stat. 1 KBit-RAM U202D Elektrische Kennwerte

Grenzwerte:

Betriebsspannung, Eingangsspannungen auf U_{SS} (0-Potential) bezogen	-0,5...+7V
Verlustleistung P_V	$\leq 1W$
Betriebstemperaturbereich	0...+70 °C
Lagertemperaturbereich	-55...+125 °C

Statische Kennwerte und Kapazitäten:

Kenngröße	Zeichen	Min	Max	Einheit	Bemerkung
Betriebsspannung	U_{CC}	4,75	5,25	V	bei $I_{OL} = 2,1 \text{ mA}$ bei $I_{OH} = 100 \mu\text{A}$
Schlafspannung	U_{CCS}	2			
Eingangsspannung	U_{IL}	-0,5	0,8		
	U_{IH}	2,0	U_{CC}		
Ausgangsspannung	U_{OL}		0,4		
	U_{OH}	2,4			
Eingangsreststrom und Reststrom des hoch-ohmigen tri-state Ausgangs	I_R		10	μA	
Stromaufnahme	I_{CC}		45	mA	
	I_{CCS}		30		
Eingangskapazität	C_I		5	pF	ns
Ausgangskapazität	C_O		10		

Dynamische Betriebsbedingungen:

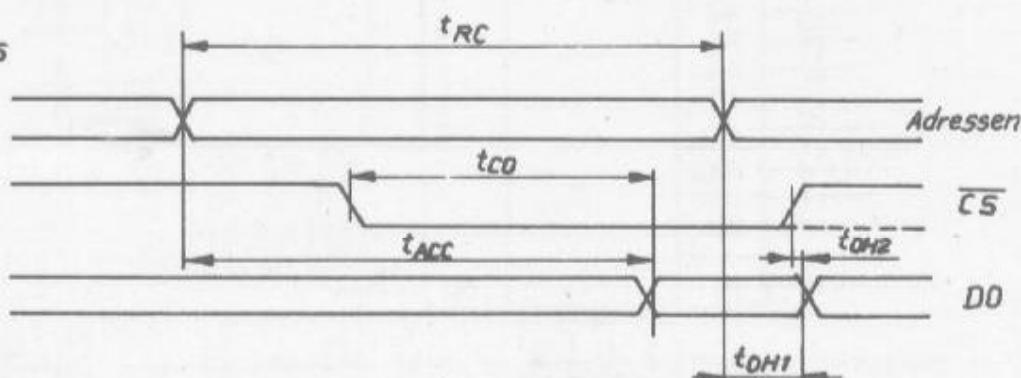
Kenngröße	Zeichen	Min	Max	Einheit
Dauer eines Lese- oder Schreibzyklus	t_{RC}, t_{WC}	400		
Adressenbereitstellzeit	t_{AW}	20		
Schreibimpulsbreite	t_{WP}	300		
Datensetzzeit	t_{DW}	300		
CS=0-Zeit im Schreibzyklus	t_{CW}	300		
Adressen-/Datenhaltezeit bezüglich \overline{WE}	t_{AH}, t_{DH}	0		
Einschlaf-Verzögerung	t_{SE}	0		
Aufwach-Verzögerung	t_{SA}	400		

Stat. 1 KBit-RAM U202D Elektrische Kennwerte

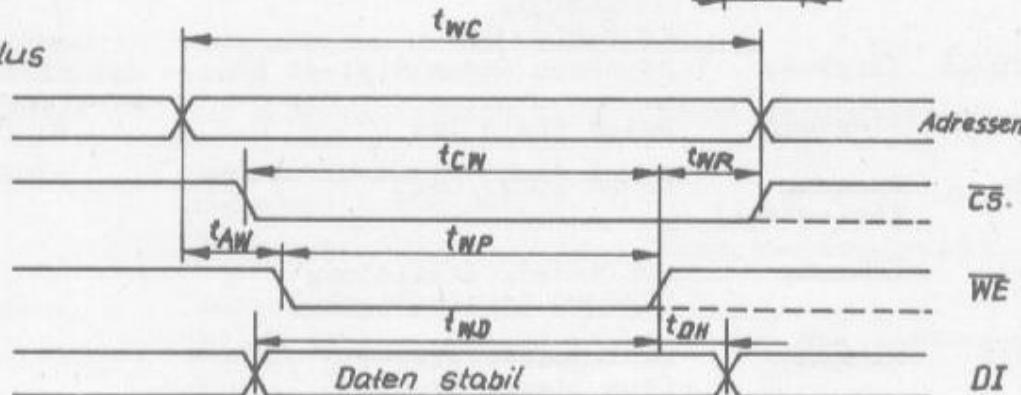
Dynamische Kennwerte:

Kenngröße	Zeichen	Min	Max	Einheit
Zugriffszeit	t_{ACC}		400	ns
Ausgangsverzögerung	t_{CO}		200	
Gültigkeitsdauer von DO nach Adressenänderung	t_{OH1}	40		
nach \overline{CS} -Änderung	t_{OH2}	0		

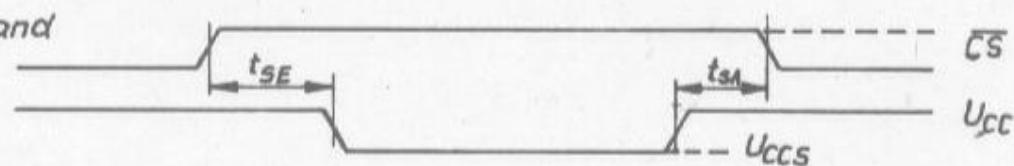
Lesezyklus



Schreibzyklus



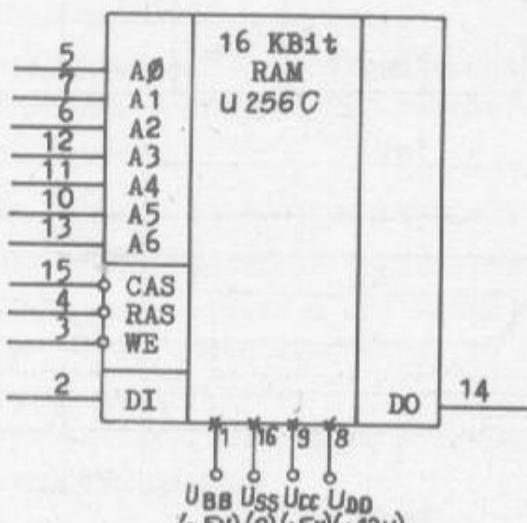
Schlafzustand



Dyn. 16 KBit-RAM U256C Anschlußbeschreibung

Merkmale:

- Dynamischer Schreib-Lesespicher, 16.384 Bit x 1 organisiert
- Alle Ein- und Ausgänge sind TTL-Kompatibel
- Übergabe der 14 Adreßbits an 7 Adressanschlüsse im Zeitmultiplex
- Betriebsspannungen -5V, +5V, +12V ($\pm 10\%$)



A ₀ -A ₆	Eingänge	Adressen (gemultipliert höher- und niedrigwertiger Teil)
DI	Eingang	Daten (Data In)
DO	Ausgang (tri-state)	Daten (Data Out)
CAS	Eingang	Spaltenadreßtastung (Column Address Strobe)
RAS	Eingang	Zeilensadreßtastung (Row Address Strobe)
WE	Eingang	Schreibfreigabe (Write Enable)

Dyn. 16 KBit-RAM U256C Funktion

- Der Tri-state-Ausgang wird von $\overline{\text{CAS}}=0$ in den niederohmigen Zustand gesteuert; die gelesenen Daten liegen am Ausgang an, bis $\overline{\text{CAS}}$ wieder inaktiv wird. Die am Ausgang einmal anliegenden Daten werden nicht mehr verändert, so daß während $\overline{\text{CAS}}=0$ ein Schreiben in die angewählte Speicherstelle erfolgen kann (kombinierter Lese-Schreib-Zyklus) oder Auffrischen mit $\overline{\text{RAS}}$ ausgeführt werden kann.
- Mit jeder $\overline{\text{RAS}}$ -Aktivierung wird die gesamte zugegriffene Zeile aufgefrischt, wobei $\overline{\text{RAS}}$ den Datenausgang nicht beeinflußt. Das Auffrischen muß für jede der 128 Zeilen mindestens alle 2 ms erfolgen. Es kann bewirkt werden durch normale Lese- bzw. Schreibzyklen, durch Auffrischzyklen, bei denen nur $\overline{\text{RAS}}$ aktiviert wird oder durch "verstecktes Auffrischen" in Zusammenhang mit einem Lesezyklus wie oben beschrieben.
- Eine besonders niedrige Zugriffszeit ergibt sich, wenn nur die Speicherstellen einer Zeile bearbeitet werden (Seiten-Modus). Die Übergabe der Zeilenadresse (mit $\overline{\text{RAS}}$) ist dann nur einmalig notwendig. Diese Betriebsart ist aber im Mikrorechnersystem mit U880D im allgemeinen nicht erforderlich.
- Eine bestimmte Reihenfolge beim Zuschalten der Betriebsspannungen ist nicht erforderlich. Nach dem Einschalten oder nach längeren Pausen ohne Auffrischen müssen mindestens 8 Zugriffszyklen (z.B. Auffrischzyklen) ausgeführt werden sein, bevor der Speicher richtig arbeitet.

Dyn. 16 KBit-RAM U256C Elektrische Kennwerte

Grenzwerte:

Betriebsspannung U_{DD}	-1...+15V
U_{CC}	-1...+
Eingangsspannung $U_{BB} (-5V)$	-0,5...+20V
Betriebstemperaturbereich	0...70 °C

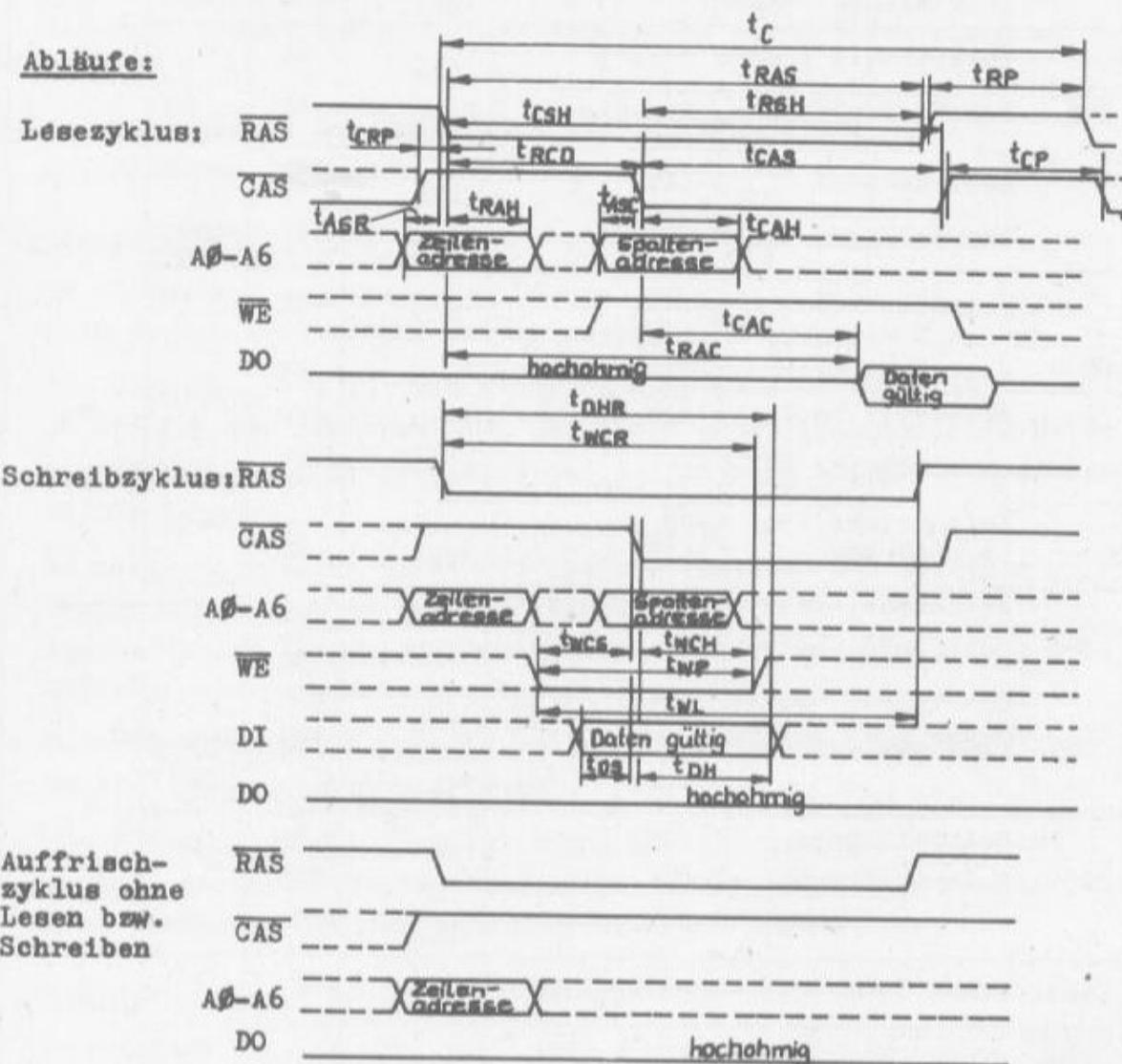
Statische Kennwerte und Kapazitäten:

Kenngröße	Zeichen	Min	Max	Einheit	Bemerkung
	U_{BB}	-5,5	-4,5	V	
Betriebsspannungen bezogen auf U_{SS} (0-Potential)	U_{CC}	4,5	5,5		
	U_{DD}	10,8	13,2		
Eingangsspannung RAS, CAS, WE	U_{IH}	2,7	7		
	U_{IL}	-1	0,8		
Eingangsspannung A0-A6, DI	U_{IH}	2,4	7		
	U_{IL}	-1	0,8		
Ausgangsspannung	U_{OH}	2,4			bei $I_{OH} = -5mA$ bei $I_{OL} = 4,2mA$
	U_{OL}		0,4		
Eingangsströme und Reststrom des hoch-ohmigen Ausgangs	I_R		10	μA	
Stromaufnahmen	I_{BB1}	0,05		mA	Ruheszustand
	I_{BB2}	0,3			$\overline{RAS}=0$
	I_{CC}	0,01			$\overline{CAS}=1$ (sonst: Ausgangsstrom)
	I_{DD1}	1,5			Ruheszustand
	I_{DD2}	35			Zykluszeit $t_{RC} = 375$ ns, $t_{RAS} = 200$ ns
Eingangskapazität A0-A6, DI RAS, WE CAS	C_{I1}		5	pF	
	C_{I2}		7		
	C_{I3}		10		
Ausgangskapazität DO	C_O		7		

Dyn. 16 KBit-RAM U256C Elektrische Kennwerte

Dynamische Kennwerte:

Kenngröße	Zeichen	Min	Max	Einheit
Zugriffszeit nach $\overline{1}$ -RAS	t_{RAC}		200	ns
Zugriffszeit nach $\overline{1}$ -CAS	t_{CAC}		135	
Verzögerung von $\overline{1}$ -CAS bis DO hochohmig	t_D	0	60	



Dyn. 16 KBit-RAM U256C Elektrische Kennwerte

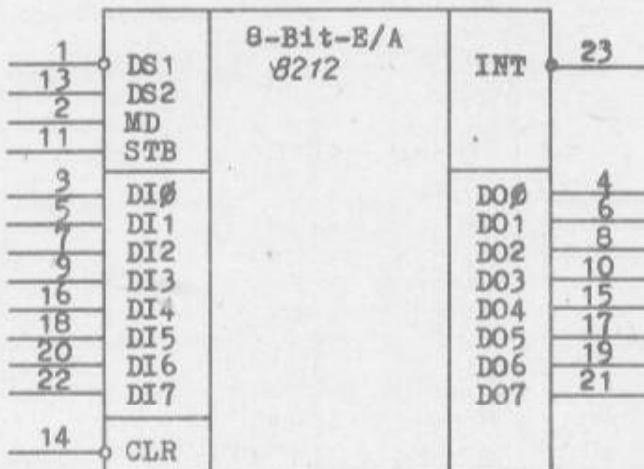
Dynamische Betriebsbedingungen (für Schreib. Lese- und Auffrischzyklen)

Kenngroße	Zeichen	Min	Max	Einheit
Auffrischzeitsabstand	t_{REF}		2	ms
Vorladezeit (Dauer $\overline{RAS}=1$)	t_{RP}	120		
\overline{RAS} - Verzögerung von $\overline{\text{CAS}}$ bis $\overline{\text{RAS}}$	t_{CRP}	-20		
Haltezeit ab $\overline{\text{CAS}}$	t_{RSH}	135		
- Impulslänge $\overline{RAS}=0$	t_{RAS}	200	10000	
Vorladezeit (Dauer $\overline{CAS}=1$)	t_{CPN}	25		
\overline{CAS} - Verzögerung von $\overline{\text{RAS}}$ bis $\overline{\text{CAS}}$	t_{RCD}	25	65	
Haltezeit ab $\overline{\text{RAS}}$	t_{CSH}	200		
Impulslänge $\overline{CAS}=0$	t_{CAS}	135	10000	
Setzzeit vor $\overline{\text{CAS}}$	t_{WCS}	-20		
\overline{WE} Haltezeit nach $\overline{\text{CAS}}$	t_{WCH}	55		
Haltezeit nach $\overline{\text{RAS}}$	t_{WCR}	120		ns
Impulslänge $\overline{WE}=0$	t_{WP}	55		
Zeitabstand von $\overline{\text{WE}}$ zu $\overline{\text{RAS}}$ bzw. $\overline{\text{CAS}}$	t_{WL}	80		
Zeilenadr.-setzzeit vor $\overline{\text{RAS}}$	t_{ASR}	0		
Zeilenadr.-haltezeit nach $\overline{\text{RAS}}$	t_{RAH}	25		
Spaltenadr.-setzzeit vor $\overline{\text{CAS}}$	t_{ASC}	-10		
Spaltenadr.-haltez.nach $\overline{\text{CAS}}$	t_{CAH}	55		
nach $\overline{\text{RAS}}$	t_{AR}	120		
Setzzeit vor $\overline{\text{CAS}}$	t_{DS}	0		
DI Haltezeit nach $\overline{\text{CAS}}$	t_{DH}	55		
nach $\overline{\text{RAS}}$	t_{DHR}	120		
Dauer eines Lese- oder Schreibzyklus	t_C	375		
Dauer eines Lese- oder Schreibzyklus im Seiten-Modus	t_{CP}	225		
Anstiegs- und Abfallzeiten der Steuersignale	t_r, t_f	3	50	

8-Bit-E/A-Schaltkreis 8212 Anschlußbeschreibung

Merkmale:

- 8-Bit-Pufferregister mit tri-state-Ausgängen
- Verwendbar zur Eingabe, Ausgabe, als Bustreiber (in einer Richtung) u.a.
- Low-power Schottky-TTL, damit Eingangslast nur 0,25 mA,
- Ausgangsleistungsfaktor 10, Ausgangsspannung $U_{OH} = 3,65V$ (EPROM-Ansteuerung u.a.)



DI0-DI7 Eingänge Daten (Data In)

DO0-DO7 Ausgänge Daten (Data Out)
(tri-state)

DS1 Eingänge Bausteinauswahl, (Device Select)
DS2 DS1=0 und DS2=1: Baustein ausgewählt

CLR Eingang Rücksetzen (Clear)
0-Rücksetzen der Flip-Flops

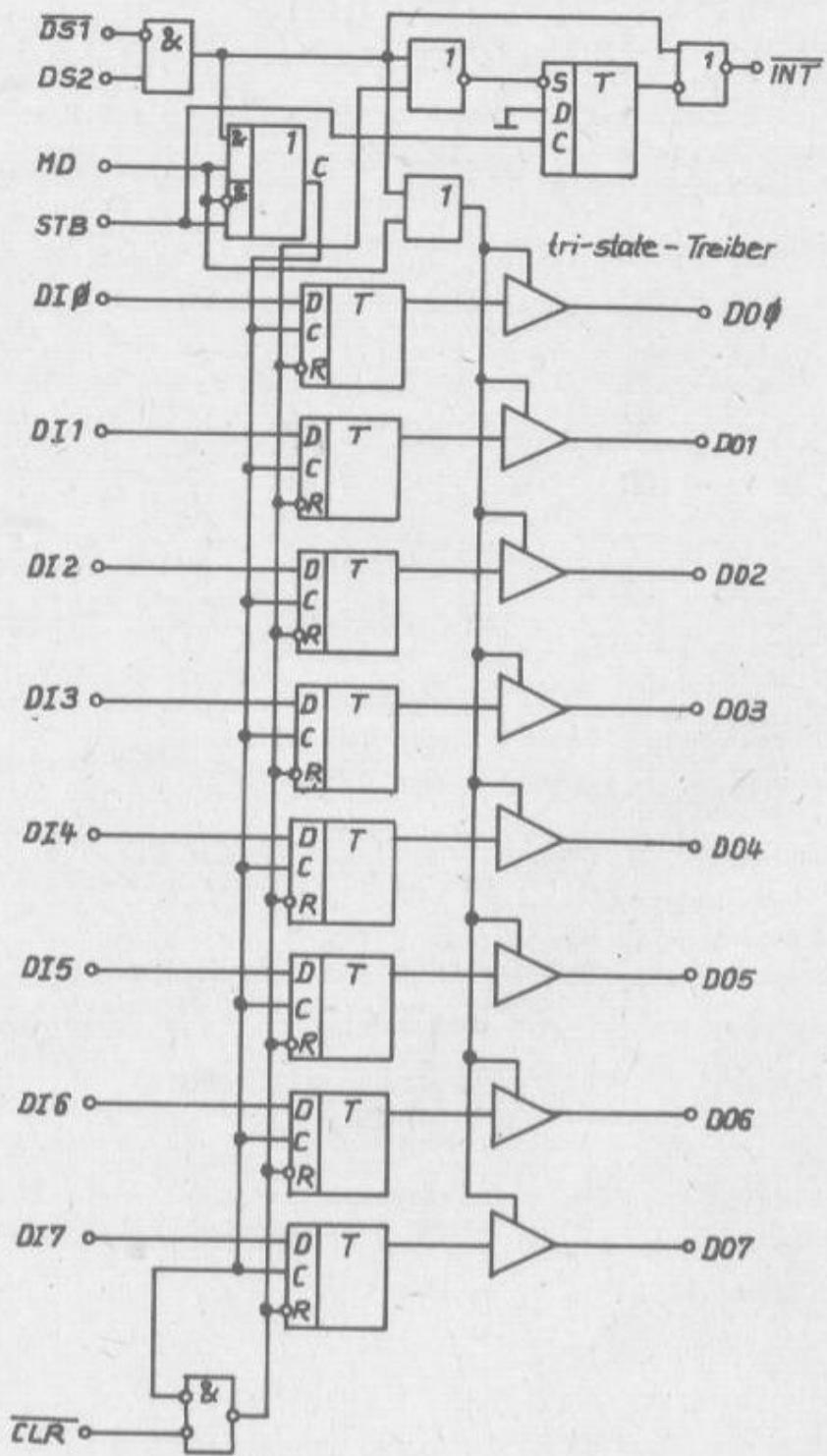
INT Ausgang 0-Unterbrechungsanmeldung (Interrupt)

/MD Eingang Betriebsartenauswahl (Mode)
0-Eingabebaustein
1-Ausgabebaustein

STB Eingang Übernahmehilfespuls (Strobe)
1-Übernahme der Eingangsbelegung (statisch)

8-Bit-E/A-Schaltkreis 8212 Funktion

Innenschaltung



8-Bit-E/A-Schaltkreis 8212 Funktion

- Die Ausgänge der statischen D-Flip-Flops in den Datenleitungen folgen unmittelbar den Eingängen, wenn der (interne) Takt C=1 ist, wenn C=0 wird, bleibt die letzte Belegung gespeichert.
Die Flip-Flops werden durch $\overline{CLR}=0$ rückgesetzt, wenn der interne Takt C=0 (passiv) ist.
- Betrieb als Eingabebaustein (MD=0):
Die tri-state-Treiber am Ausgang schalten durch, wenn der Baustein angewählt ist ($DS1=0$ und $DS2=1$), sonst sind die Ausgänge hoch-ohmig.
Die Datenübernahme der Daten-Flip-Flops wird durch STB gesteuert ($C=STB$)
- Betrieb als Ausgabebaustein (MD=1):
Die tri-state-Treiber sind ständig durchgeschaltet (kein hoch-ohmiger Zustand)
Die Datenübernahme der Daten-Flip-Flops wird durch die Auswahlsignale DS1 und DS2 gesteuert ($C=DS1 \cdot DS2$)
- Das Unterbrechungsanmeldungs-Flip-Flop wird von der fallenden Flanke von CLR sowie bei Vorliegen der Auswahlbedingung ($DS1=0$, $DS2=1$) gesetzt (Unterbrechung dann inaktiv!).
Das Rücksetzen (Unterbrechungsanmeldung) erfolgt mit der fallenden Flanke von STB.

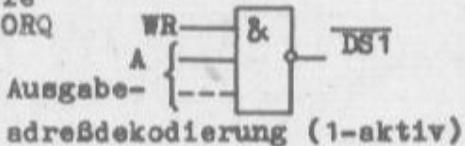
Anwendungen:

- Lastfaktorerweiterung für unidirektionale Busse (Adressbus...) mit Abschaltungsmöglichkeit: STB, $\overline{CLR} \rightarrow 1$ (+5V)
 $MD, DS1 \rightarrow 0$

DS2 - Ausgangaktivierung (1-Ausgabe aktiv)

Die Eingangsbelegungen werden sofort durchgeschaltet

- Ausgabebaustein am U880D-System (ohne Hardware-Quittungsbetrieb)
D16-DI7 an Systemdatenbus (bzw. Ausgabedatenbus)
D08-D07 zur Peripherie
 $MD \rightarrow 1$ (+5V), $DS2 \rightarrow IORQ$



8-Bit-E/A-Schaltkreis 8212 Elektrische Kennwerte

Grenzwerte, gültig für 8212, 8216, 8205

Betriebsspannung	-0,5...+7V
Eingangsspannungen	-1...+5,5V
Ausgangsströme	≤ 125mA

Statische Kennwerte

bei $T_a = 0 \dots 70^\circ C$, $U_{CC} = 4,75 \dots 5,25V$

Kennwert	Zeichen	Min	Max	Einheit	Bemerkung
Eingangsströme DS2, CLR, DI ₆ -DI ₇ STB	I _{IL}	-250		μA	alle I _{IL} bei U _{IL} = 0,45V I _{IH} bei U _{IH} = 5,25V
	I _{IH}		10		
Eingangsströme MD	I _{ILMD}	-750		μA	
	I _{IHMD}		30		
Eingangsstrom DS1	I _{IL}	-1000			
	I _{IHDS1}		40		
Eingangsspannungen	U _{IL}			V	bei I _{OL} = 15mA
	U _{IH}	2,0	0,85		
Ausgangsspannungen	U _{OL}		0,45		bei I _{OL} = -1mA
	U _{OH}	3,65			
Stromaufnahme	I _{CC}		130	mA	

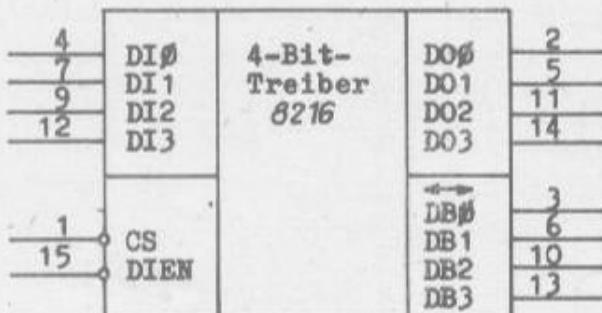
Dynamische Kennwerte:

Kennwert	Symbol	Min	Max	Einheit
Impulslänge an STB bzw. DS1 · DS2	t _{PW}	30		ns
Durchlaufverzögerung DI → DO im durchgeschalt. Zustand	t _{PD}		30	
Verzögerungszeit von STB bzw. DS1 · DS2 bis DO	t _D		40	
Datensetzezeit bis zur STB bzw. DS1 · DS2 Aktivierung	t _S	15		

4-Bit-bidirektonaler Treiber 8216 Anschlußbeschreib.

Merkmale:

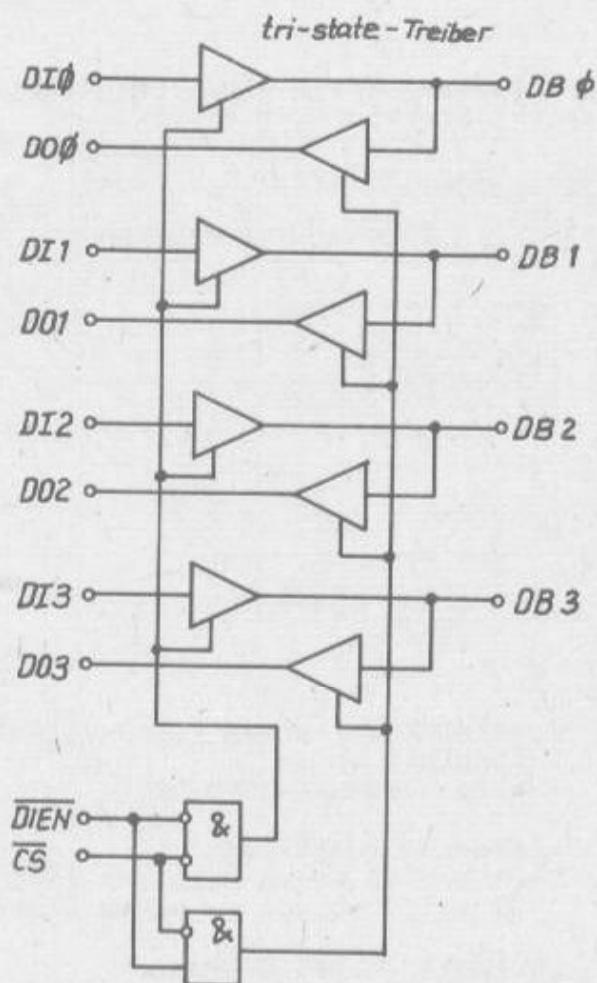
- Zweirichtungs-Treiberschaltung als Bustreiber für bidirektionale Busse oder Aufspaltung bidirektonaler Busse.
- Low-Power-Schottky-TTL, Eingangslast nur 0,25 mA.
- Ausgangsbelastbarkeit $I_{OL} \geq 55$ mA; Ausgangsspannung $U_{OH} (DO) = 3,65V$.



<u>CS</u>	Eingang	Bausteinaktivierung (Chip Select) 0-Baustein aktiv 1-alle Ausgänge hochohmig
<u>DIEN</u>	Eingang	Richtungsumschaltung 0-DI auf DB durch schalten (Eingabe) 1-DB auf DO durch schalten (Ausgabe)
DB0-DB3	Ein/Ausgänge (tri-state)	bidirektonaler Datenbus
DI0-DI3	Eingänge	Eingabedatenbus
DO0-DO3	Ausgänge (tri-state)	Ausgabedatenbus

4-Bit-bidirektonaler Treiber 8216 Funktion

Innenschaltung:



Funktionsabelle:

CS	DIEN	Funktion
0	0	DI \rightarrow DB, DO hochohmig
0	1	DB \rightarrow DO
1	0	
1	1	Ausgänge DB, DO hochohmig

Anwendungen:

- Treiberschaltung für bidirektionalen CPU-Datenbus
 $\overline{CS} \rightarrow$ BUSRQ-Signal (Freigabe auch des gepufferten Datenbusses)
 $\overline{DIEN} \rightarrow M1 \vee RD$
alle DI an entsprechende DO \rightarrow CPU-Datenbus
DB \rightarrow Datenbus mit erweitertem Lastfaktor
 - Ankopplung von Speicherbausteinen mit getrennten Datenein- und Ausgängen an einem bidirektionalen Datenbus:
 $\overline{CS} \rightarrow CS$ des Speichers DO \rightarrow DI des Speichers
 $\overline{DIEN} \rightarrow RD$ DB \rightarrow Datenbus zur CPU
DI \rightarrow DO des Speichers

4-Bit-bidirektonaler Treiber 8216 Elektrische Kennwerte

Grenzwerte: siehe 8212

Statische Kennwerte:

Kennwert	Zeichen	Min	Max	Einheit	Bemerkung
Eingangsstrom für DIEN, CS	I _{ILS}	-500		μA	alle I _{IL} bei U _{IL} = 0,45V I _{IH} bei U _{IH} = 5,25V
	I _{IHS}		80		
Eingangsstrom DI0-DI3	I _{ILDI}	-250			
	I _{IHD}		40		
Eingangsstrom und Reststrom der tri-state Ausgänge DB0-DB3	I _{ILDB}	-250			
	I _{IHD}		100		
Reststrom der tri-state Ausgänge DO0-DO3	I _{RDO}		20		
Eingangsspannung	U _{IL}		0,95	V	
	U _{IH}	2,0			
Ausgangsspannung DO0-DO3	U _{OL}		0,45		bei I _{OL} = 15 mA
	U _{OH}	3,65			bei I _{OH} = -1 mA
Ausgangsspannung DB0-DB3	U _{OL}		0,45		bei I _{OL} = 25 mA
	U _{OH}	2,4	0,6		bei I _{OL} = 55 mA
Stromaufnahme	I _{CC}		130	mA	bei I _{OH} = -10 mA

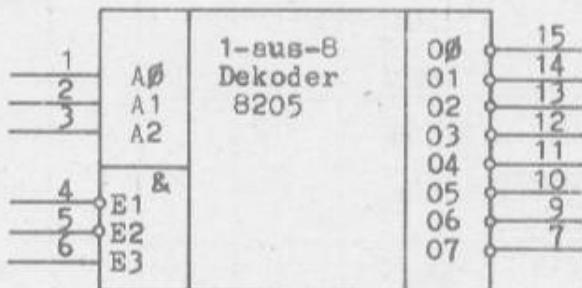
Dynamische Kennwerte:

Kennwert	Max	Einheit
Verzögerung von DB nach DO	25	ns
Verzögerung von DI nach DB	30	
Verzögerung von DIEN/CS-Aktivierung bis DO,DB aktiv	65	
Verzögerung von DIEN/CS-Deaktivierung bis DO, DB hochohmig	35	

1-aus-8-Dekoder 8205 Anschlußbeschreibung

Merkmale:

- Speicher- oder E/A-Adreßdekoder
- Low-Power-Schottky-TTL; Eingangslast nur 0,25 mA;
Durchschaltzeit \leq 18 ns
- Drei zusätzliche Freigabesignale zur Kaskadierung mehrerer Bausteine bzw. zur Verknüpfung mit bestimmten Steuersignalen.



A₀-A₂

Eingänge

Dekodereingänge

$\overline{E_1}, \overline{E_2}, E_3$

Eingänge

Freigabe, es wird nur dann ein Ausgang aktiviert, wenn $\overline{E_1}=0$; $\overline{E_2}=0$; $E_3=1$.

00-07

Ausgänge

Dekoderausgänge, nur der ausgewählte Ausgang führt 0-Pegel.

1-aus-8-Dekoder 8205 Funktion

Funktionstabelle:

A2	A1	A0	E1	E2	E3	O0	O1	O2	O3	O4	O5	O6	O7
0	0	0	0	0	1	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	0	1	1	1	1	1	1
0	1	0	0	0	1	1	1	0	1	1	1	1	1
0	1	1	0	0	1	1	1	1	0	1	1	1	1
1	0	0	0	0	1	1	1	1	1	0	1	1	1
1	0	1	0	0	1	1	1	1	1	1	0	1	1
1	1	0	0	0	1	1	1	1	1	1	1	0	1
1	1	1	0	0	1	1	1	1	1	1	1	1	0
beliebig	jede Belegung außer												
	0 0 1			1	1	1	1	1	1	1	1	1	1

Anwendung:

- Dekoder zur Speicheransteuerung
z.B. für 1 KByte-Blöcke:
A0-A9 von CPU direkt an Speicher
A0,1,2 des 8205 → A10,11,12 von CPU
 $\overline{E2} \rightarrow MREQ$ (oder E3 → MREQ)
mit den restlichen Freigabeeingängen können
z.B. A13,14,15 in die Dekodierung einbezogen werden.
Die Ausgänge O0-O7 können direkt mit den
 \overline{CS} -Eingängen der Speicherblöcke verbunden werden.
- Dekoder zur E/A-Ansteuerung
z.B. \overline{CS} -Ansteuerung für 8 PIO's U855D
A0-A1 von CPU → Kanalauswahl und Daten/Steuerauswahl für PIO
A0,1,2 des 8205 → A2,3,4 von CPU
mit den Freigabeeingängen kann die Lage im
E/A-Adressraum bestimmt werden,
 $\overline{E1} \rightarrow A5$
 $\overline{E2} \rightarrow A6$
 $\overline{E3} \rightarrow A7$ } von CPU
ergibt Basisadresse 80H für 1.PIO
84H für 2.PIO
9CH für 8.PIO

1-aus-8-Dekoder 8205 Elektrische Kennwerte

Grenzwerte: siehe 8212

Statische Kennwerte:

bei $T_a = 0 \dots 70^\circ\text{C}$, $U_{CC} = 4,75 \dots 5,25\text{V}$

Kennwert	Zeichen	Min	Max	Einheit	Bemerkung
Eingangsstrom	I_{IL}	-250		μA	bei $U_{IL} = 0,45\text{V}$ bei $U_{IH} = 5,25\text{V}$
	I_{IH}		10		
Eingangsspannung	U_{IL}		0,85	V	bei $I_{OL} = 10 \text{ mA}$ bei $I_{OH} = -1,5 \text{ mA}$
	U_{IH}	2,0			
Ausgangsspannung	U_{OL}		0,45		bei $I_{OL} = 10 \text{ mA}$ bei $I_{OH} = -1,5 \text{ mA}$
	U_{OH}	2,4			
Stromaufnahme	I_{CC}		70	mA	

Dynamische Kennwerte:

Kennwert	Zeichen	Min	Max	Einheit
Verzögerungszeit von Adresse bzw. Freigabe bis zum Ausgang	t_V		18	ns

Internationales Alphabet Nr. 2 (Fernschreibkode)

Nr.	Buchstabe	Ziffer/ Zeichen	b_4	b_3	b_2	b_1	b_0	Hex
1	A	-	0	0	0	1	1	03
2	B	?	1	1	0	0	1	19
3	C	:	0	1	1	1	0	0E
4	D	+	0	1	0	0	1	09
5	E	3	0	0	0	0	1	01
6	F		0	1	1	0	1	0D
7	G		1	1	0	1	0	1A
8	H		1	0	1	0	0	14
9	I	8	0	0	1	1	0	06
10	J	凶	0	1	0	1	1	0B
11	K	(0	1	1	1	1	0F
12	L)	1	0	0	1	0	12
13	M	.	1	1	1	0	0	1C
14	N	,	0	1	1	0	0	0C
15	O	9	1	1	0	0	0	18
16	P	0	1	0	1	1	0	16
17	Q	1	1	0	1	1	1	17
18	R	4	0	1	0	1	0	0A
19	S	'	0	0	1	0	1	05
20	T	5	1	0	0	0	0	10
21	U	7	0	0	1	1	1	07
22	V	=	1	1	1	1	0	1E
23	W	2	1	0	0	1	1	13
24	X	/	1	1	1	0	1	1D
25	Y	6	1	0	1	0	1	15
26	Z	+	1	0	0	0	1	11
27	<		0	1	0	0	0	08
28	≡		0	0	0	1	0	02
29	A...		1	1	1	1	1	1F
30	1...		1	1	0	1	1	1B
31	Leerschritt		0	0	1	0	0	04
32	Sonderverwendung		0	0	0	0	0	00

Achtung ! Das niederwertige Bit b_0 ist das zuerst gesendete!

Internationales Alphabet Nr.5 - Sonderzeichenbelegung nach
TGL 23 207

höherwertig		b ₆	0	0	0	0	1	1	1	1		
		b ₅	0	0	1	1	0	0	1	1		
nieder-	wertig	b ₄	0	1	0	1	0	1	0	1		
b ₃	b ₂	b ₁	b ₀	Hex	0	1	2	3	4	5	6	7
0	0	0	0	NUL	DLE	SP	0	@(€◊)	P	~	p	
0	0	0	1	SOH	DC1	!	1	A	Q	¤	%	
0	0	1	0	STX	DC2	"	2	B	R	¤	x	
0	0	1	1	ETX	DC3	#(£)	3	C	S	c	a	
0	1	0	0	EOT	DC4	¤(§)	4	D	T	d	t	
0	1	0	1	ENQ	NAK	%	5	E	U	e	u	
0	1	1	0	ACK	SYN	&	6	F	V	f	v	
0	1	1	1	BEL	ETB	'	7	G	W	g	w	
1	0	0	0	BS	CAN	(8	H	X	h	x	
1	0	0	1	HT	EM)	9	I	Y	i	y	
1	0	1	0	LF	SUB	*	:	J	Z	j	z	
1	0	1	1	VT	ESC	+	:	K	(A*)	k	{(a)}	
1	1	0	0	FF	FS	,	<	L	¤(Ö◊)	l	(ö)	
1	1	0	1	CR	GS	-	=	M	(Ü)	m	(ü)	
1	1	1	0	SO	RS	.	>	N	†(^-)	n	†(ß~)	
1	1	1	1	SI	US	/	?	O	-	o	DEL	

Bedeutung der wichtigsten Steuerzeichen:

- BEL - Klingel
- BS - Rückwärtsschritt (Back Space)
- HT - Horizontaltabulator
- LF - Zeilenschaltung (Line Feed)
- VT - Vertikaltabulator
- FF - Formularvorschub (Form Feed)
- CR - Wagenrücklauf (Carriage Return)
- ESC - Abbruch (Escape)
- SP - Leerzeichen (Space)

Anschlußbelegungen

A11	1	40	A10	D2	1	40	D3
A12	2	39	A9	D7	2	39	D4
A13	3	38	A8	D6	3	38	D5
A14	4	37	A7	CE	4	37	M1
A15	5	36	A6	C1D	5	36	TORQ
C	6	35	A5	B1A	6	35	RD
D4	7	34	A4	PA7	7	34	PB7
D3	8	33	A3	PA6	8	33	PB6
D5	9	CPU	32	A2	PA5	9	PIO
D6	10	U880D	31	A1	PA4	10	U855D
(+5V)UCC	11		30	AΦ	(D) USS	11	30
D2	12		29	USS (0)	PA3	12	29
D7	13		28	RFSH	PA2	13	28
DΦ	14		27	M1	PA1	14	27
D1	15		26	RESET	PAΦ	15	26
INT	16		25	BUSRQ	ASTB	16	25
NMI	17		24	WAIT	BSTB	13	24
HALT	18		23	BUSAK	ARDY	18	23
MREQ	19		22	WR	DΦ	19	22
TORQ	20		21	RD	D1	20	21

D1	1	40	DΦ	D4	1	28	D3
D3	2	39	D2	D5	2	27	D2
D5	3	38	D4	D6	3	26	D1
D7	4	37	D6	D7	4	25	DΦ
INT	5	36	TORQ	(D) USS	5	24	UCC (+5V)
IEI	6	35	CE	RD	6	CTC	CITRG Φ
IED	7	34	B1A	ZC1TOΦ	7	U857D	CITRG 1
M1	8	33	C1D	ZC1TO1	8	21	CITRG 2
(+5V)UCC	9	32	RD	ZC1TO2	9	20	CITRG 3
WIRDYA	10	S10	31	USS (0)	TORQ	10	K5 1
SYNCA	11	U856D	30	WIRDYB	IEO	11	K5 Φ
RxD A	12	*	29	SYNCB	INT	12	RESET
RxC A	13		28	RxD B	IEI	13	CE
TxA	14		27	RxTxCB	M1	14	C
TxD A	15		26	TxD B			
DTRA	16		25	DTRB			
RTSA	17		24	RTSB			
CTS A	18		23	CTS B			
DCDA	19		22	DCDB			
C	20		21	RESET			

* Bondversion Φ

Anschlußbelegungen

A7	1	24	U_{CC} (+5V)	A7	1	24	U_{CC} (+5V) (-5V)	U_{BB}	1	16	U_{SS} (0)		
A6	2	EPROM	23	A8	2	EPROM	23	A8	DI	2	RAM	15	\overline{CAS}
A5	3	U555C	22	A9	3	2716	22	A9	\overline{WE}	3	DD	14	\overline{DO}
A4	4		21	U_{BB} (-5V)	4		21	U_{PP}	RAS	4	U256C	13	A6
A3	5		20	\overline{CS}	5		20	\overline{OE}	A0	5		12	A3
A2	6		19	U_{DD} (+12V)	6		19	A10	A2	6		11	A4
A1	7		18	PR	7		18	\overline{CE}	A1	7		10	A5
A0	8		17	07	8		17	07	(+12V)	UDD	8	9	U_{CC} (+5V)
00	9		16	06	00	9	16	06					
01	10		15	05	01	10	15	05					
02	11		14	04	02	11	14	04					
(0)	U _{SS}		13	03	(0)	U _{SS}	12	03					

Anschlußbelegungen

