

Horst Völz

**Tabellen zum
Mikrorechnersystem
U 880 - Z 80**

UB Rostock

87 A

148

Akademie-Verlag Berlin

Horst Völz

Tabellen zum Mikrorechnersystem U 880 - Z 80

Die hier zusammengestellten Tabellen
und Bilder entstammen dem im selben Verlag
erschienenen Buch des Autors
„Elektronik. Grundlagen, Prinzipien,
Zusammenhänge“, 4., bearbeitete
und wesentlich erweiterte Auflage 1986



Akademie-Verlag Berlin
1986

Universitäts-
Bibliothek
Hofstad

87A 148

ISBN 3-05-500058-7

Erschienen im Akademie-Verlag Berlin, DDR-1086 Berlin, Leipziger Str. 3-4

(C) Akademie-Verlag Berlin 1986

Lizenznummer: 202 · 100/513/86

Printed in the German Democratic Republic

Gesamtherstellung: VEB Druckerei „Thomas Müntzer“, 5820 Bad Langensalza

Lektor: Dipl.-Phys. Ursula Heilmann

LSV 3537, 1087, 1187

Bestellnummer: 763 603 0 (6962)

00500

Inhaltsverzeichnis

	Seite
Kurzzeichen	5
Ladebefehle	6
Arithmetikbefehle	7
Bit- und Jumpbefehle	8
1-Byte-Zahlen mit und ohne Vorzeichen	9
Einige Befehlsabläufe	10
1-Byte und CB-n	11
ED-, DD- und FD-n	12
DD FD-CB-d-n und IM2-Interrupt	13
U 880 bzw. Z 80. Aufbau	14
U 880 bzw. Z 80. Zeitdiagramme	15
Interrupt-FF	16
U 880 bzw. Z 80. Beschreibung der Anschlüsse	17
Bedeutung der Flags	18
Flag-Operationen und CPU-Peripherie	19
PIO U 855	20
PIO U 855. Programmierung	21
PIO U 855, Bedeutung und Funktion der Anschlußleitungen	22
CTC U 857	23
SIO U 856	24
SIO U 856. Spezifische Anschlüsse	25
DMA U 858	25/26
Sockelschaltungen	27
Darstellung des internationalen 7-bit-Kodes für Texte	28



Beim Umgang mit der Mikrorechentechnik hat das System U 880 (Z 80), U 855, U 856, U 857, U 858 bzw. CPU, PIO, SIO, CTC, DMA eine besonders große Bedeutung. Immer wieder muß — vor allem, wenn kein Assembler oder Disassembler zur Verfügung steht — auf Tabellen zurückgegriffen werden. Solcher Rückgriff ist vorwiegend in Ausbildung und Lehre und bei nichtkommerziellen Anwendern, aber auch im Debugging erforderlich. Die vorgenommene Zusammenstellung soll in einfacher Weise diesem breiten Bedürfnis Rechnung tragen. Bei der Auswahl sind auch die wichtigsten Daten zur Initialisierung und zum Betrieb der Peripheriebausteine sowie einige Hardwareanliegen berücksichtigt worden. Quelle der hier vorgelegten Tabellen und Bilder ist das ebenfalls im Akademie-Verlag Berlin erschienene Buch des Autors: „Elektronik. Grundlagen, Prinzipien, Zusammenhänge“, 4., bearbeitete und wesentlich erweiterte Auflage. Lediglich die Seiten 9 und 27 sind darüber hinaus ergänzt. Dem Akademie-Verlag sei für die bereitwillige Unterstützung dieses Vorhabens gedankt.

September 1985

Horst Völz

Kurzzeichen

Bedeutung und Werte von Kurzzeichen bei Befehlen des U 880. Die eingeklammerten Bezeichnungen werden zuweilen in der Literatur benutzt. In der Spalte bit sind die Positionen angegeben, wo die zugehörigen bit in einem Byte stehen, in der Spalte Byte die Lage in einem (Mehrbyte-) Befehl. Die unteren Tabellen weisen die Bedeutung je nach Belegung der bit aus. Für d gilt 00 H bis 7F H $\hat{=}$ 0 D bis 127 D; 80 H bis 0F FH $\hat{=}$ - 128 D bis - 1 D.

Zeichen	Beschreibung	Wert	bit	Byte
b	Bit-Position im Bereich von 0-7	0 ÷ 7	3 ÷ 5	letztes
c	Bedingung für Sprünge (2 bit)		3 + 4	I
cc	Bedingung für Sprünge und Subroutinen		3 ÷ 5	I
d	Distanz im Speicherregister IX, IY	-128 ÷ 127	0 ÷ 7	III
dd	Registerpaare BC, DE, HL, SP		4 ÷ 5	I/II
e	relative Sprungweite (d, 2)	-126 ÷ 129	0 ÷ 7	II
(f)	siehe m			
ii	Indexregister IX oder IY		5	I
M	= (HL)			
m	r, (HL), (IX + d), (IY + d)			
n	Zahlenwert	0 ÷ 255	0 ÷ 7	letztes
nn	Zahlenwert	0 ÷ 65553	0 ÷ 7	beide letzten
p	wie t gebraucht			
pp	Registerpaare BC, DE, IX oder IY, SP		4 + 5	II
qq	Registerpaare BC, DE, HL, AF		4 + 5	I
r	Register A, B, C, D, E, H, L (n bzw. (HL))		3 ÷ 5	I/II
r'	wie r, jedoch speziell als Quelle		0 ÷ 2	I/II
(rr)	wie pp, jedoch nur IY, dann pp nur IX		4 + 5	II
s	r, n, (HL), (IX + d), (IY + d)			
(ss)	wie dd (Source, Drain)			
t	Sprungadressen		3 ÷ 5	I
tt	Registerpaare BC, DE		4	I

() bedeutet: Inhalt des Registers bestimmt die Adresse der Speicherzelle

bit	ii	tt
0	IX	BC
1	IY	DE

bit	c	dd	pp	qq
00	NZ	BC	BC	BC
01	Z	DE	DE	DE
10	NC	HL	IX/IY	HL
11	C	SP	SP	AF

bit	cc	b	r	t
000	NZ	0	B	00H = 0D
001	Z	1	C	08H = 8D
010	NC	2	D	10H = 16D
011	C	3	E	18H = 24D
100	PO	4	H	20H = 32D
101	PE	5	L	28H = 40D
110	P	6	n/(HL)	30H = 48D
111	M	7	A	38H = 56D

Ladebefehle

8-bit-LD		Quelle													indiziert (IX + d) (IY + d)	erweiterte Adresse (mm)	Direkt- wert n	
		Register										Register indirekt						
		1	R	A	B	C	D	E	H	L	M	(BC)	(DE)					
Ziel	Register	A	ED 47	ED 5F	7F	78	79	7A	7B	7C	7D	7E	0A	1A	DD 7E d	FD 7E d	3A n	3E n
		B			47	40	41	42	43	44	45	46			DD 4E d	FD 4E d		0E n
		C			4F	48	49	4A	4B	4C	4D	4E			DD 4E d	FD 4E d		0E n
		D			57	50	51	52	53	54	55	56			DD 5E d	FD 5E d		1E n
		E			5F	58	59	5A	5B	5C	5D	5E			DD 5E d	FD 5E d		1E n
		H			67	60	61	62	63	64	65	66			DD 6E d	FD 6E d		2E n
		L			6F	68	69	6A	6B	6C	6D	6E			DD 6E d	FD 6E d		2E n
	Register indirekt	M			77	70	71	72	73	74	75							3E n
	indiziert	(IX + d)				DD 77 d	DD 70 d	DD 71 d	DD 72 d	DD 73 d	DD 74 d	DD 75 d						DD 3E n
	erweiterte Adresse	(mm)																FD 3E n
implizit	R																	

Austauschbefehle	
EXX	D9
EXAF	08
EXDE	E1
EX(SP),HL	E3
EX(SP),IX	DD E3
EX(SP),IY	FD E3

Blocktransfer	
LDI	ED A0
LDIR	ED B0
LDD	ED A8
LDDR	ED B8

16 bit LD		Quelle								erweit. Direkt- wert mm	erweit. Adresse (mm)	indir. Reg. Adr. (SP)		
		Register												
		AF	BC	DE	HL	SP	IX	IY						
Ziel	Register	AF										F1		
		BC									01 n n	ED 4B n n	C1	
		DE									11 n n	ED 5B n n	D1	
		HL									21 n n	2A n n	E1	
		SP					F9			DD F9	FD F9	31 n n	ED 7B n n	
		IX										DD 21 n n	DD 2A n n	DD E1
		IY										FD 21 n n	FD 2A n n	FD E1
	erweit. Adr.	(mm)			ED 43 n n	ED 53 n n	ED 22 n n	ED 73 n n	DD 22 n n	FD 22 n n				
PUSH- Befehle	indir. Reg. Adr.	(SP)	F3	C5	D5	E5			DD E5	FD E5				

Achtung! Die PUSH- u. POP-Befehle verändern den SP-Inhalt nach jeder Ausführung

POP-Befehle

Arithmetikbefehle

8-bit-arithm. logisch	Quelle											
	Register-Adressierung								indir. Reg.-Adr.	indizierte Adressierung		Direktwert
	A	B	C	D	E	H	L	M		(IX - d)	(IX + d)	
Addition ADD	87	80	81	82	83	84	85	86	DD 86 d	FD 86 d	C6 n	
Addition mit Übertrag ADC	8F	88	89	8A	8B	8C	8D	8E	DD 8E d	FD 8E d	CE n	
Subtraktion SUB	97	90	91	92	93	94	95	96	DD 96 d	FD 96 d	D6 n	
Subtraktion mit Übertrag SBC	9F	98	99	9A	9B	9C	9D	9E	DD 9E d	FD 9E d	DE n	
„UND“ AND	A7	A0	A1	A2	A3	A4	A5	A6	DD A6 d	FD A6 d	E6 n	
exklusives oder XOR	AF	AS	A9	AA	AB	AC	AD	AE	DD AE d	FD AE d	EE n	
oder OR	B7	B0	B1	B2	B3	B4	B5	B6	DD B6 d	FD B6 d	F6 n	
Vergleich CP	BF	B8	B9	BA	BB	BC	BD	BE	DD BE d	FD BE d	FE n	
Erhöhen um 1 INC	3C	04	0C	14	1C	24	2C	34	DD 34 d	FD 34 d		
Erniedr. um 1 DEC	2D	05	0D	15	1D	25	2D	35	DD 35 d	FD 35 d		

DAA	27
CPL	8F
NEG	ED 44
CCF	3F
SCF	27

Blockvergleich	
CPI	ED A1
CPIC	ED B1
CPD	ED A9
CPDR	ED B9

10-bit-arithm.-logisch		Quelle					
		BC	DE	HL	SP	IX	IY
Ziel	ADD	HL	09	10	29	39	
		IX	DD 09	DD 19		DD 39	DD 29
		IY	FD 09	FD 19		FD 39	FD 29
Ziel	Addition mit Übertrag und Flagsetzer ADC	HL	ED 4A	ED 5A	ED 6A	ED 7A	
	Subtraktion mit Übertrag und Flagsetzer SBC	HL	ED 42	ED 52	ED 62	ED 72	
Erhöhen um 1 INC			03	13	23	33	DD 23 FD 23
Erniedrigen um 1 DEC			0B	1B	2B	3B	DD 2B FD 2B

Quelle und Ziel		A	B	C	D	E	H	L	M	(IX - d)	(IX + d)	A				
RLC	07	CB 00	CB 01	CB 02	CB 03	CB 04	CB 05	CB 06	CB 06	DD CB d D6	FD CB d D6	RLCA	07		zykl. Verschiebung links herum	RLC
RRC	0F	CB 08	CB 09	CB 0A	CB 0B	CB 0C	CB 0D	CB 0E	CB 0E	DD CB d DE	FD CB d DE	RRCA	0F		zykl. Verschiebung rechts herum	RRC
RL	17	CB 10	CB 11	CB 12	CB 13	CB 14	CB 15	CB 16	CB 16	DD CB d 16	FD CB d 16	RLA	17		zykl. Verschiebung links	RL
RR	1F	CB 18	CB 19	CB 1A	CB 1B	CB 1C	CB 1D	CB 1E	CB 1E	DD CB d 1E	FD CB d 1E	RRA	1F		zykl. Verschiebung rechts	RR
SLA	27	CB 20	CB 21	CB 22	CB 23	CB 24	CB 25	CB 26	CB 26	DD CB d 26	FD CB d 26				Verschiebung links, arithm.	SLA
SRA	2F	CB 28	CB 29	CB 2A	CB 2B	CB 2C	CB 2D	CB 2E	CB 2E	DD CB d 2E	FD CB d 2E				Verschiebung rechts, arithm.	SRA
SRL	3F	CB 38	CB 39	CB 3A	CB 3B	CB 3C	CB 3D	CB 3E	CB 3E	DD CB d 3E	FD CB d 3E				Verschiebung rechts logisch	SRL
RLD										ED 0F					zykl. Verschiebung	RLD
RRD										ED 07					zykl. Verschiebung	RRD

Bit- und Jumpbefehle

BIT	Registeradressierung							indir. Reg. (HL)	indiziert		
	A	B	C	D	E	H	L		(IX + d)	(IY + d)	
TEST „BIT“	0	CB 47	CB 40	CB 41	CB 42	CB 43	CB 44	CB 45	CB 46	DD CB d 40	FD CB d 46
	1	CB 4F	CB 48	CB 49	CB 4A	CB 4B	CB 4C	CB 4D	CB 4E	DD CB d 4E	FD CB d 4E
	2	CB 5F	CB 50	CB 51	CB 52	CB 53	CB 54	CB 55	CB 56	DD CB d 56	FD CB d 56
	3	CB 5F	CB 58	CB 59	CB 5A	CB 5B	CB 5C	CB 5D	CB 5E	DD CB d 5E	FD CB d 5E
	4	CB 6F	CB 60	CB 61	CB 62	CB 63	CB 64	CB 65	CB 66	DD CB d 66	FD CB d 66
	5	CB 6F	CB 68	CB 69	CB 6A	CB 6B	CB 6C	CB 6D	CB 6E	DD CB d 6E	FD CB d 6E
	6	CB 7F	CB 70	CB 71	CB 72	CB 73	CB 74	CB 75	CB 76	DD CB d 76	FD CB d 76
	7	CB 7F	CB 78	CB 79	CB 7A	CB 7B	CB 7C	CB 7D	CB 7E	DD CB d 7E	FD CB d 7E
LÖSCHEN „BUS“	0	CB 87	CB 80	CB 81	CB 82	CB 83	CB 84	CB 85	CB 86	DD CB d 86	FD CB d 86
	1	CB 8F	CB 88	CB 89	CB 8A	CB 8B	CB 8C	CB 8D	CB 8E	DD CB d 8E	FD CB d 8E
	2	CB 9F	CB 90	CB 91	CB 92	CB 93	CB 94	CB 95	CB 96	DD CB d 96	FD CB d 96
	3	CB 9F	CB 98	CB 99	CB 9A	CB 9B	CB 9C	CB 9D	CB 9E	DD CB d 9E	FD CB d 9E
	4	CB AF	CB A0	CB A1	CB A2	CB A3	CB A4	CB A5	CB A6	DD CB d A6	FD CB d A6
	5	CB AF	CB A8	CB A9	CB AA	CB AB	CB AC	CB AD	CB AE	DD CB d AE	FD CB d AE
	6	CB BF	CB B0	CB B1	CB B2	CB B3	CB B4	CB B5	CB B6	DD CB d B6	FD CB d B6
	7	CB BF	CB B8	CB B9	CB BA	CB BB	CB BC	CB BD	CB BE	DD CB d BE	FD CB d BE
BITSETZEN „SET“	0	CB CF	CB C0	CB C1	CB C2	CB C3	CB C4	CB C5	CB C6	DD CB d C6	FD CB d C6
	1	CB CF	CB C8	CB C9	CB CA	CB CB	CB CC	CB CD	CB CE	DD CB d CE	FD CB d CE
	2	CB DF	CB D0	CB D1	CB D2	CB D3	CB D4	CB D5	CB D6	DD CB d D6	FD CB d D6
	3	CB DF	CB D8	CB D9	CB DA	CB DB	CB DC	CB DD	CB DE	DD CB d DE	FD CB d DE
	4	CB EF	CB E0	CB E1	CB E2	CB E3	CB E4	CB E5	CB E6	DD CB d E6	FD CB d E6
	5	CB EF	CB E8	CB E9	CB EA	CB EB	CB EC	CB ED	CB EE	DD CB d EE	FD CB d EE
	6	CB FF	CB F0	CB F1	CB F2	CB F3	CB F4	CB F5	CB F6	DD CB d F6	FD CB d F6
	7	CB FF	CB F8	CB F9	CB FA	CB FB	CB FC	CB FD	CB FE	DD CB d FE	FD CB d FE

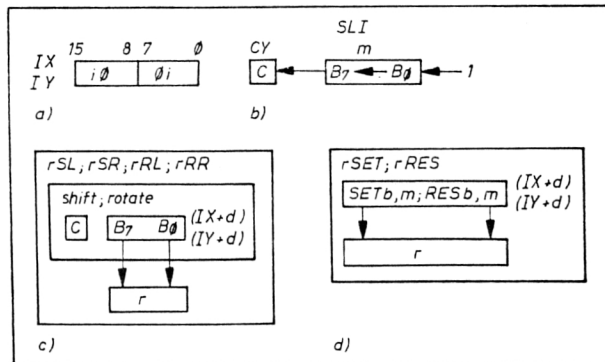
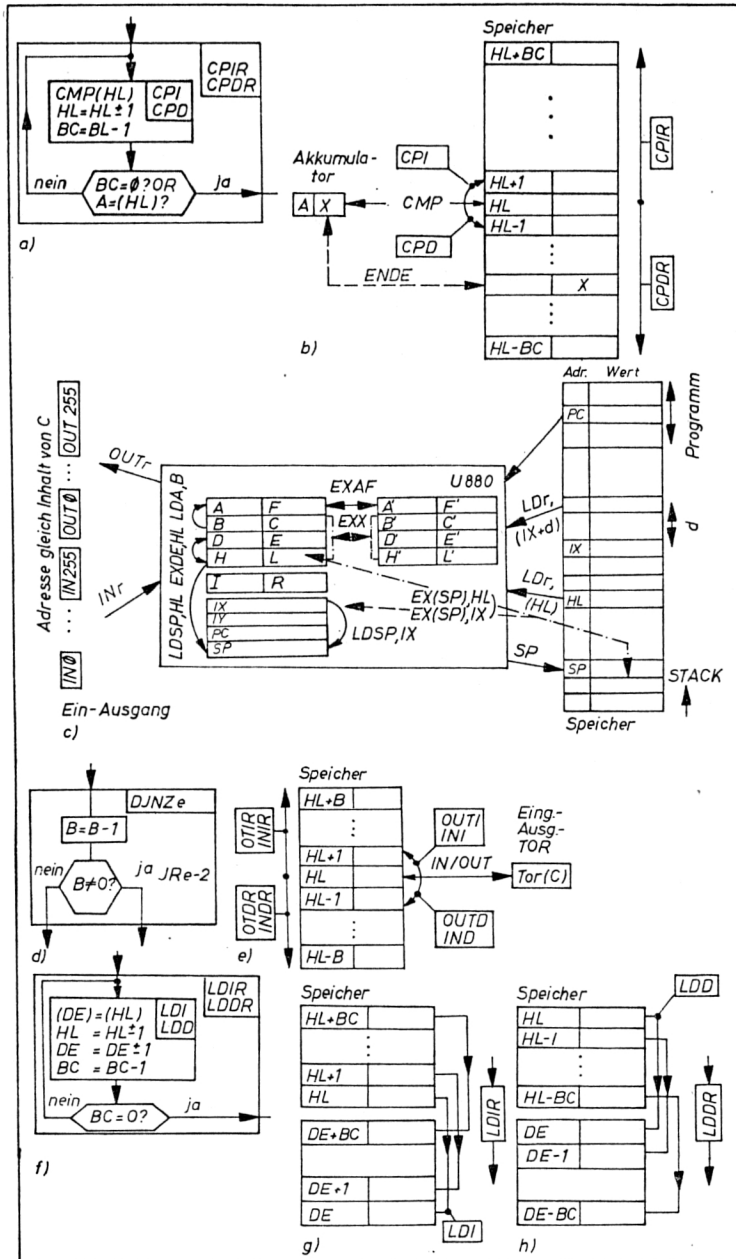
Ein-Ausgabe			Restart		
	IN	OUT	RST	HEX	CALL-Adr
n	DB n	D3 n	0	C7	0 0 0 0 H
A	ED 78	ED 79	8	CF	0 0 0 8 H
B	ED 40	ED 41	16	D7	0 0 1 0 H
C	ED 48	ED 49	24	DF	0 0 1 8 H
D	ED 50	ED 51	32	E7	0 0 2 0 H
E	ED 58	ED 59	40	EF	0 0 2 8 H
H	ED 60	ED 61	48	F7	0 0 3 0 H
L	ED 68	ED 69	56	FF	0 0 3 8 H
F	ED 70				

Block-Ein-Ausgabe		
	IN	OUT
I	ED A2	ED A3
IRL	ED B2	ED B3
D	ED AA	ED AB
DIRL	ED DA	ED DB

NOP	00
HALT	76
DI	F3
EI	FD
IM0	ED 46
IM1	ED 56
IM2	ED 5F

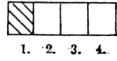
Spang	Bedingung	Übertrag U		kein Übertr. Z		NZ		Z		kein Übertr. Z		NZ		Z		kein Übertr. Z		NZ		Z		kein Übertr. Z		NZ		Z	
		unbedingt	unbedingt	DA	DB	DC	DD	DE	DF	DA	DB	DC	DD	DE	DF	DA	DB	DC	DD	DE	DF	DA	DB	DC	DD	DE	DF
erweiterter Direktwert	nn	C3	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
relativ	FC + 0	18	0-2	38	0-2	38	0-2	38	0-2	38	0-2	38	0-2	38	0-2	38	0-2	38	0-2	38	0-2	38	0-2	38	0-2	38	0-2
indirekt Register	M	ED		ED		ED		ED		ED		ED		ED		ED		ED		ED		ED		ED		ED	
erweiterter Direktwert	nn	CD	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
relativ	FC + 0	ED		ED		ED		ED		ED		ED		ED		ED		ED		ED		ED		ED		ED	
indirekt Register	(SP)	ED		ED		ED		ED		ED		ED		ED		ED		ED		ED		ED		ED		ED	
indirekt Register	(SP + 1)	ED		ED		ED		ED		ED		ED		ED		ED		ED		ED		ED		ED		ED	
indirekt Register	(SP + 1)	ED		ED		ED		ED		ED		ED		ED		ED		ED		ED		ED		ED		ED	
indirekt Register	(SP + 1)	ED		ED		ED		ED		ED		ED		ED		ED		ED		ED		ED		ED		ED	
indirekt Register	(SP + 1)	ED		ED		ED		ED		ED		ED		ED		ED		ED		ED		ED		ED		ED	

Einige Befehlsabläufe



1-Byte und CB-n

		2. Tetrade															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1. Tetrade	0	NOI _e	LD BC, _{nn}	LD(BC) _{,A}	INC BC	INC B	DEC B	LD B, _n	RLCA	EX AF,AF'	ADD HL,BC	LDA, (BC)	DEC BC	INC C	DEC C	LD C	RRCA
	1	DINZ _e	LD DE, _{nn}	LD(DE) _{,A}	INC DE	INC D	DEC D	LD D, _n	RLA	JR, _e	ADD HL,DE	LDA, (DE)	DEC DE	INC E	DEC E	LD E, _n	RRA
	2	JR NZ, _e	LD HL, _{nn}	LD(nn) _{,HL}	INC HL	INC H	DEC H	LD H, _n	DAA	JR Z, _e	ADD HL,HL	LD HL, _(nn)	DEC HL	INC L	DEC L	LD L, _n	CPL
	3	JR NC, _e	LD SP, _{nn}	LD(nn) _{,A}	INC SP	INC (HL)	DEC (HL)	LD(HL) _{,n}	SCF	JR C, _e	ADD HL,SP	LDA (n,n)	DEC SP	INC A	DEC A	LD A, _n	CF
	4	LD B,B	LD B,C	LD B,D	LD B,E	LD B,H	LD B,L	LD B,(HL)	LD B,A	LD C,B	LD C,C	LD C,D	LD C,E	LD C,H	LD C,L	LD C,A	LD C,A
	5	LD D,B	LD D,C	LD D,D	LD D,E	LD D,H	LD D,L	LD D,(HL)	LD D,A	LD E,B	LD E,C	LD E,D	LD E,E	LD E,H	LD E,L	LD E,A	LD E,A
	6	LD H,B	LD H,C	LD H,D	LD H,E	LD H,H	LD H,L	LD H,(HL)	LD H,A	LD L,B	LD L,C	LD L,D	LD L,E	LD L,H	LD L,L	LD L,A	LD L,A
	7	LD (HL),B	LD (HL),C	LD (HL),D	LD (HL),E	LD (HL),H	LD (HL),L	HALT	LD (HL),A	LD A,B	LD A,C	LD A,D	LD A,E	LD A,H	LD A,L	LD A,A	LD A,A
	8	ADD B	ADD C	ADD D	ADD E	ADD H	ADD (HL)	ADD (HL)	ADD A	ADC B	ADC C	ADC D	ADC E	ADC H	ADC L	ADC (HL)	ADC A
	9	SUB B	SUB C	SUB D	SUB E	SUB H	SUB (HL)	SUB (HL)	SUB A	SBC B	SBC C	SBC D	SBC E	SBC H	SBC L	SBC (HL)	SBC A
	A	AND B	AND C	AND D	AND E	AND H	AND (HL)	AND (HL)	AND A	XOR B	XOR C	XOR D	XOR E	XOR H	XOR L	XOR (HL)	XOR A
	B	OR B	OR C	OR D	OR E	OR H	OR (HL)	OR (HL)	OR A	CP B	CP C	CP D	CP E	CP H	CP L	CP (HL)	CP A
	C	RET NZ	POP BC	JP NZ, _{nn}	JP _{nn}	CALL NZ, _{nn}	PUSH BC	ADD A, _n	RST 0	RET Z	RET	JP Z, _{nn}	Bid 8.6.22a	CALL Z, _{nn}	CALL _{nn}	ADC A, _n	RST 8
	D	RET NC	POP DE	JP NC, _{nn}	OUT _n	CALL NC, _{nn}	PUSH DE	SUB _n	RST 10H	RET C	EXX	JP C, _{nn}	IN _n	CALL C, _{nn}	Bid 8.6.23	SBC A, _n	RST 10H
	E	RET PO	POP HL	JP PO, _{nn}	EX (SP),HL	CALL PO, _{nn}	PUSH HL	AND _n	RST 20H	RET PE	JP (HL)	JP PE, _{nn}	EX DE,HL	CALL PE, _{nn}	Bid 8.6.23	XOR A, _n	RST 20H
	F	RET P	POP AF	JP P, _{nn}	DI	CALL P, _{nn}	PUSH AF	OR _n	RST 30H	RET M	LD (SP),HL	JP M, _{nn}	EI	CALL M, _{nn}	Bid 8.6.23	CP	RST 30H



		2. Tetrade															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1. Tetrade	0	RLC B	RLC C	RLC D	RLC E	RLC H	RLC L	RLC (HL)	RLC A	RRC B	RRC C	RRC D	RRC E	RRC H	RRC L	RRC (HL)	RRC A
	1	RL B	RL C	RL D	RL E	RL H	RL L	RL (HL)	RL A	RR B	RR C	RR D	RR E	RR H	RR L	RR (HL)	RR A
	2	SLA B	SLA C	SLA D	SLA E	SLA H	SLA L	SLA (HL)	SLA A	SRA B	SRA C	SRA D	SRA E	SRA H	SRA L	SRA (HL)	SRA A
	3	SLI B	SLI C	SLI D	SLI E	SLI H	SLI L	SLI (HL)	SLI A	SRL B	SRL C	SRL D	SRL E	SRL H	SRL L	SRL (HL)	SRL A
	4	BIT 0,B	BIT 0,C	BIT 0,D	BIT 0,E	BIT 0,H	BIT 0,L	BIT 0,(HL)	BIT 0,A	BIT 1,B	BIT 1,C	BIT 1,D	BIT 1,E	BIT 1,H	BIT 1,L	BIT 1,(HL)	BIT 1,A
	5	BIT 2,B	BIT 2,C	BIT 2,D	BIT 2,E	BIT 2,H	BIT 2,L	BIT 2,(HL)	BIT 2,A	BIT 3,B	BIT 3,C	BIT 3,D	BIT 3,E	BIT 3,H	BIT 3,L	BIT 3,(HL)	BIT 3,A
	6	BIT 4,B	BIT 4,C	BIT 4,D	BIT 4,E	BIT 4,H	BIT 4,L	BIT 4,(HL)	BIT 4,A	BIT 5,B	BIT 5,C	BIT 5,D	BIT 5,E	BIT 5,H	BIT 5,L	BIT 5,(HL)	BIT 5,A
	7	BIT 6,B	BIT 6,C	BIT 6,D	BIT 6,E	BIT 6,H	BIT 6,L	BIT 6,(HL)	BIT 6,A	BIT 7,B	BIT 7,C	BIT 7,D	BIT 7,E	BIT 7,H	BIT 7,L	BIT 7,(HL)	BIT 7,A
	8	RES 0,B	RES 0,C	RES 0,D	RES 0,E	RES 0,H	RES 0,L	RES 0,(HL)	RES 0,A	RES 1,B	RES 1,C	RES 1,D	RES 1,E	RES 1,H	RES 1,L	RES 1,(HL)	RES 1,A
	9	RES 2,B	RES 2,C	RES 2,D	RES 2,E	RES 2,H	RES 2,L	RES 2,(HL)	RES 2,A	RES 3,B	RES 3,C	RES 3,D	RES 3,E	RES 3,H	RES 3,L	RES 3,(HL)	RES 3,A
	A	RES 4,B	RES 4,C	RES 4,D	RES 4,E	RES 4,H	RES 4,L	RES 4,(HL)	RES 4,A	RES 5,B	RES 5,C	RES 5,D	RES 5,E	RES 5,H	RES 5,L	RES 5,(HL)	RES 5,A
	B	RES 6,B	RES 6,C	RES 6,D	RES 6,E	RES 6,H	RES 6,L	RES 6,(HL)	RES 6,A	RES 7,B	RES 7,C	RES 7,D	RES 7,E	RES 7,H	RES 7,L	RES 7,(HL)	RES 7,A
	C	SET 0,B	SET 0,C	SET 0,D	SET 0,E	SET 0,H	SET 0,L	SET 0,(HL)	SET 0,A	SET 1,B	SET 1,C	SET 1,D	SET 1,E	SET 1,H	SET 1,L	SET 1,(HL)	SET 1,A
	D	SET 2,B	SET 2,C	SET 2,D	SET 2,E	SET 2,H	SET 2,L	SET 2,(HL)	SET 2,A	SET 3,B	SET 3,C	SET 3,D	SET 3,E	SET 3,H	SET 3,L	SET 3,(HL)	SET 3,A
	E	SET 4,B	SET 4,C	SET 4,D	SET 4,E	SET 4,H	SET 4,L	SET 4,(HL)	SET 4,A	SET 5,B	SET 5,C	SET 5,D	SET 5,E	SET 5,H	SET 5,L	SET 5,(HL)	SET 5,A
	F	SET 6,B	SET 6,C	SET 6,D	SET 6,E	SET 6,H	SET 6,L	SET 6,(HL)	SET 6,A	SET 7,B	SET 7,C	SET 7,D	SET 7,E	SET 7,H	SET 7,L	SET 7,(HL)	SET 7,A



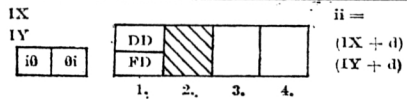
ED-, DD- und FD-n

		2. Tetrade																			
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F				
1. Tetrade	0																				
	1																				
	2																				
	3																				
	4	IN B	OUT B	SBC BC	LD (nn),BC	NEG	RETn	IM 0	LD 1,A	IN C	OUT C	ADC BC	LD BC,(nn)				RET1	LD R,A			
	5	IN D	OUT D	SBC DE	LD (nn),DE			IM 1	LD A,I	IN E	OUT E	ADC DE	LD DE,(nn)				IM 2	LD A,R			
	6	IN H	OUT H	SBC HL					RRD	IN L	OUT L	ADC HL						RLD			
	7			SBC SP	LD (nn),SP					IN A	OUT A	ADC SP	LD SP,(nn)								
	8																				
	9																				
	A	LDI	CPI	INI	OUTI													LDD	CPD	IND	OUTD
	B	LDIR	CPDR	INIR	OTIR													LDDR	CPDE	INDR	OTDR
	C																				
	D																				
	E																				
	F																				



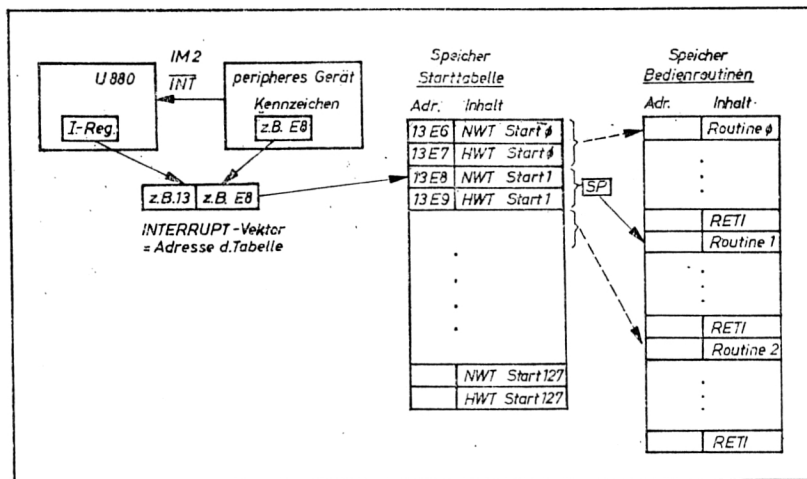
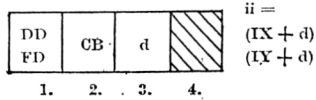
1. 2. 3. 4.

		2. Tetrade																					
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F						
1. Tetrade	0																						
	1																						
	2	LD ii, mn	LD (nn),ii	INC ii	INC i0	DEC i0	LD i0,n											ADD ii,BC					
	3							INC (ii)	DEC (ii)	LD (ii),n											ADD ii,DE		
	4							LD B,i0	LD B,0i	LD B,(ii)											ADD ii,SP		
	5							LD D,i0	LD D,0i	LD D,(ii)											LD C,i0	LD C,0i	LD C,(ii)
	6	LD i0,B	LD i0,C	LD i0,D	LD i0,E	LD i0,ii	LD i0,0i	LD H,(ii)	LD i0,A	LD 0i,B	LD 0i,C	LD 0i,D	LD 0i,E	LD 0i,0i	LD 0i,0i	LD L,(ii)	LD 0i,A						
	7	LD (ii),B	LD (ii),C	LD (ii),D	LD (ii),E	LD (ii),H	LD (ii),L		LD (ii),A											LD A,i0	LD A,0i	LD A,(ii)	
	8							ADD A,i0	ADD A,0i	ADD (ii)											ADC A,i0	ADC A,0i	ADC (ii)
	9							SUB i0	SUB 0i	SUB (ii)											SBC i0	SBC 0i	SBC (ii)
	A							AND i0	AND 0i	AND (ii)											XOR i0	XOR 0i	XOR (ii)
	B							OR i0	OR 0i	OR (ii)											CP i0	CP 0i	CP (ii)
	C																	Bild 8.6.24					
	D																						
	E	POP ii					EX (SP),ii					PUSH ii							JP (ii)				
	F																	LD SP,ii					

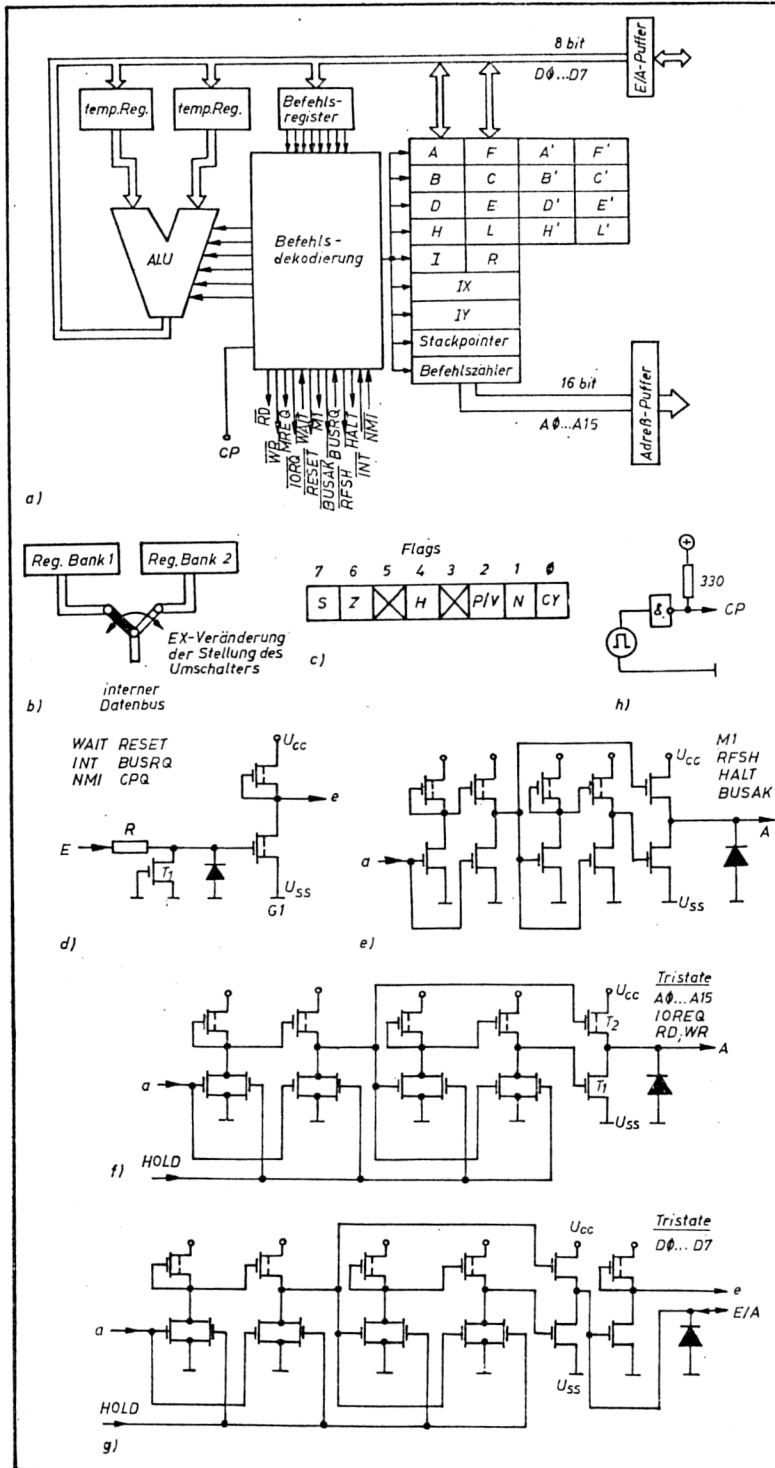


DD -CB-d-n und IM2-Interrupt

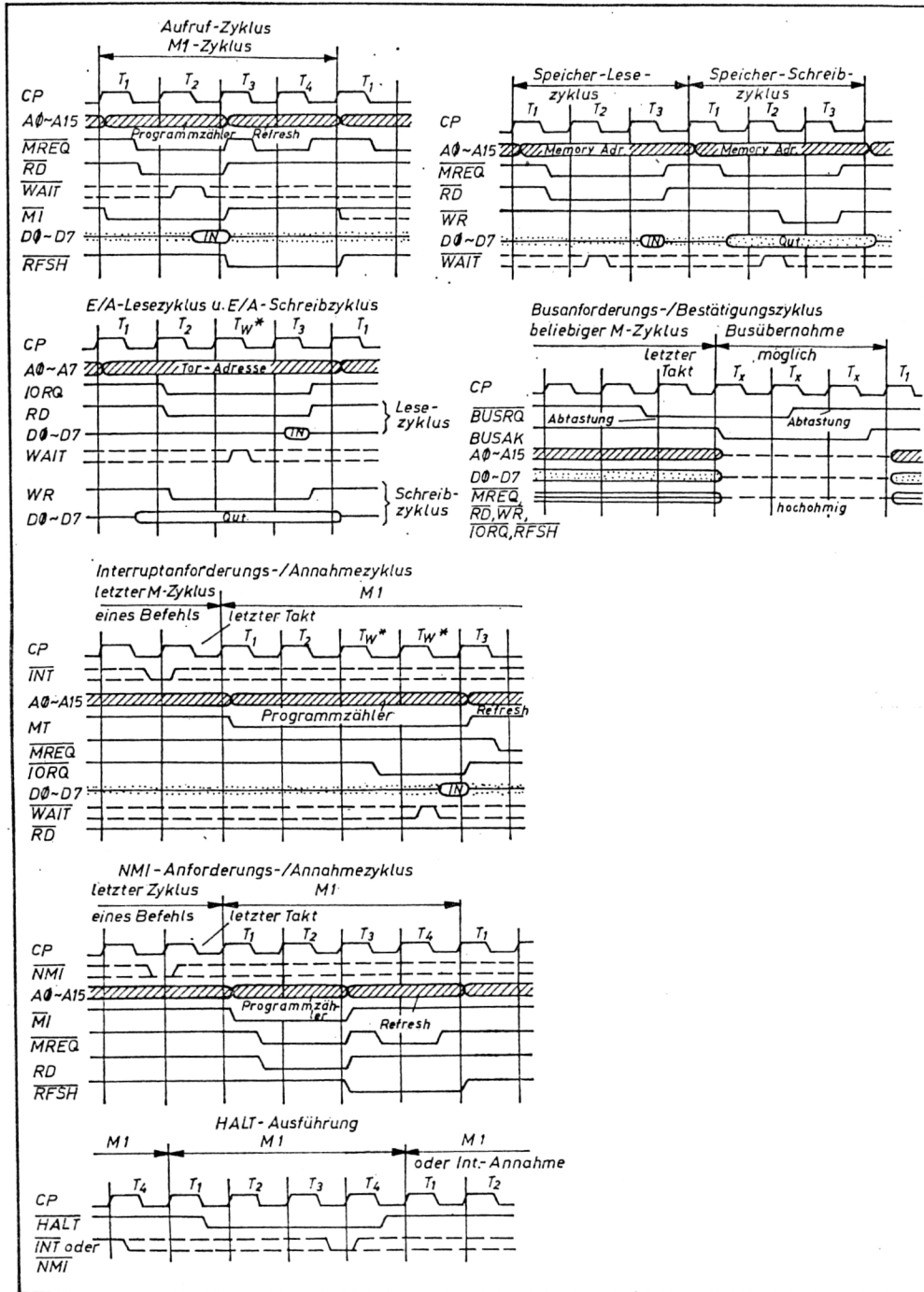
		2. Tetrade															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1. Tetrade	0	rRLC B	rRLC C	rRLC D	rRLC E	rRLC H	rRLC L	RLC ii	rRLC A	rRRC B	rRRC C	rRRC D	rRRC E	rRRC H	rRRC L	RRC ii	rRRC A
	1	rRL B	rRL C	rRL D	rRL E	rRL H	rRL L	RL ii	rRL A	rRR B	rRR C	rRR D	rRR E	rRR H	rRR L	RR ii	rRR A
	2	rSLA B	rSLA C	rSLA D	rSLA E	rSLA H	rSLA L	SLA ii	rSLA A	rSRA B	rSRA C	rSRA D	rSRA E	rSRA H	rSRA L	SRA ii	rSRA A
	3	rSLI B	rSLI C	rSLI D	rSLI E	rSLI H	rSLI L	SLI ii	rSLI A	rSRL B	rSRL C	rSRL D	rSRL E	rSRL H	rSRL L	SRL ii	rSRL A
	4							BIT 0,ii									BIT 1,ii
	5							BIT 2,ii									BIT 3,ii
	6							BIT 4,ii									BIT 5,ii
	7							BIT 6,ii									BIT 7,ii
	8	rRES B,0	rRES C,0	rRES D,0	rRES E,0	rRES H,0	rRES L,0	RES 0,ii	rRES A,0	rRES B,1	rRES C,1	rRES D,1	rRES E,1	rRES H,1	rRES L,1	RES 1,ii	rRES A,1
	9	rRES B,2	rRES C,2	rRES D,2	rRES E,2	rRES H,2	rRES L,2	RES 2,ii	rRES A,2	rRES B,3	rRES C,3	rRES D,3	rRES E,3	rRES H,3	rRES L,3	RES 3,ii	rRES A,3
	A	rRES B,4	rRES C,4	rRES D,4	rRES E,4	rRES H,4	rRES L,4	RES 4,ii	rRES A,4	rRES B,5	rRES C,5	rRES D,5	rRES E,5	rRES H,5	rRES L,5	RES 5,ii	rRES A,5
	B	rRES B,6	rRES C,6	rRES D,6	rRES E,6	rRES H,6	rRES L,6	RES 6,ii	rRES A,6	rRES B,7	rRES C,7	rRES D,7	rRES E,7	rRES H,7	rRES L,7	RES 7,ii	rRES A,7
	C	rSET B,0	rSET C,0	rSET D,0	rSET E,0	rSET H,0	rSET L,0	SET 0,ii	rSET A,0	rSET B,1	rSET C,1	rSET D,1	rSET E,1	rSET H,1	rSET L,1	SET 1,ii	rSET A,1
	D	rSET B,2	rSET C,2	rSET D,2	rSET E,2	rSET H,2	rSET L,2	SET 2,ii	rSET A,2	rSET B,3	rSET C,3	rSET D,3	rSET E,3	rSET H,3	rSET L,3	SET 3,ii	rSET A,3
	E	rSET B,4	rSET C,4	rSET D,4	rSET E,4	rSET H,4	rSET L,4	SET 4,ii	rSET A,4	rSET B,5	rSET C,5	rSET D,5	rSET E,5	rSET H,5	rSET L,5	SET 5,ii	rSET A,5
	F	rSET B,6	rSET C,6	rSET D,6	rSET E,6	rSET H,6	rSET L,6	SET 6,ii	rSET A,6	rSET B,7	rSET C,7	rSET D,7	rSET E,7	rSET H,7	rSET L,7	SET 7,ii	rSET A,7



**U 880 bzw. Z 80
Aufbau**



U 880 bzw. Z 80
Zeitdiagramme



Interrupt-FF

CPU-Aktionen	Flip-Flop	Bemerkungen
RESET	IFF1 = IFF2 = 0	IM0; I-REG = 0
DI	IFF1 = IFF2 = 0	
EI	IFF1 = IFF2 = 1	
LDA, I und LDA, R	unverändert	P/V = IFF2
INT	IFF1 = IFF2 = 0	
NMI	IFF2 = IFF1; IFF1 = 0	
RETN und RETI	IFF2 bleibt; IFF1 = IFF2	

U 880 bzw. Z 80 Beschreibung der Anschlüsse

A0 ... A15	16-Bit-Adreßbus	<ul style="list-style-type: none"> - Tri-State-Ausgänge - liefern Adressen für externe Speicher (max. 64 kByte) und E/A-Bausteine
D0 ... D7	8-Bit-bidirektionaler Datenbus	<ul style="list-style-type: none"> - Tri-State-Ein-/Ausgänge - Datenaustausch zwischen CPU und Speicher bzw. CPU und E/A-Bausteinen
$\overline{M1}$	Maschinenzyklus 1	<ul style="list-style-type: none"> - Tri-State-Ausgang, low aktiv - bei $\overline{M1}$ aktiv ist der laufende Maschinenzyklus der Operationscode-Lesezyklus der momentanen auszuführenden Anweisung - Bestätigung der Interruptanforderung (nur zusammen mit \overline{IORQ} aktiv)
\overline{MREQ}	Speicheranforderung ("Memory Request")	<ul style="list-style-type: none"> - Tri-State-Ausgang, low aktiv - Speicherzugriff (Lesen oder Schreiben)
\overline{IORQ}	E/A-Anforderung, ("Input/Output-Request")	<ul style="list-style-type: none"> - Tri-State-Ausgang, low aktiv - niederwertige 8 bit des Adreßbusses liefert Adresse zur E/A-Torauswahl (Ein- oder Ausgabe) - Bestätigung der Interruptanforderung (nur zusammen mit $\overline{M1}$ aktiv)
\overline{RD}	Lesen ("Read")	<ul style="list-style-type: none"> - Tri-State-Ausgang, low aktiv - Lesen der Daten vom Speicher oder von einem E/A-Baustein
\overline{WR}	Schreiben ("Write")	<ul style="list-style-type: none"> - Tri-State-Ausgang, low aktiv - CPU hält Daten für Speicher oder E/A-Baustein auf Datenbus bereit
\overline{RFSH}	Auffrischen der Information ("Refresh")	<ul style="list-style-type: none"> - Tri-State-Ausgang, low aktiv - niederwertige 7 bit des Adreßbusses führen Refresh-adresse für dynamische Speicher - Verwendung der laufenden \overline{MREQ}-Signale zur Einleitung eines Refreshzyklus für alle angeschlossenen Speicher
\overline{HALT}	Halt-Zustand („Halt“)	<ul style="list-style-type: none"> - Ausgang, low aktiv - CPU im (Software-) Halt-Zustand - zur weiteren Programmabarbeitung ist Interrupt-Signal nötig - CPU führt zur Sicherstellung des Refresh-Vorganges Leerbefehle (NOPs) aus
\overline{WAIT}	Warte-Anforderung („Wait“)	<ul style="list-style-type: none"> - Eingang, low aktiv - Warten der CPU bis Speicher oder E/A-Schaltung bereit
\overline{INT}	Maskierte Interrupt-Anforderung (Interrupt-Request")	<ul style="list-style-type: none"> - Eingang, low aktiv - Unterbrechungsanforderung an die CPU (wird erst nach Abarbeitung des in Ausführung befindlichen Befehls berücksichtigt)
\overline{NMI}	Nichtmaskierbare Interrupt-Anforderung ("nonmaskable Interrupt")	<ul style="list-style-type: none"> - Eingang, low aktiv - Interrupt höherer Priorität - bei Anlagen eines Low-Signals an \overline{NMI} wird entsprechend einer RESTART-Instruktion die Programmbehandlung bei Speicheradresse 0066 H fortgesetzt
\overline{RESET}	Rücksetzen ("Reset")	<ul style="list-style-type: none"> - Eingang, low aktiv - Rücksetzen von Interrupt-Freigabe-Flip-Flop, Befehlszähler, Register I und R - CPU in Interruptsbetriebsart 0 während \overline{RESET} aktiv - Daten- und Adreßbus hochohmig, die übrigen Ausgänge inaktiv
\overline{BUSRQ}	Bus-Anforderung ("Bus-Request")	<ul style="list-style-type: none"> - Eingang, low aktiv - Adreß-, Daten- und Steuerbus hochohmig, so daß diese Leitungen von externen Schaltungen benutzt werden können
\overline{BUSAk}	Bus-Anforderungsbestätigung ("Bus-Acknowledgement")	<ul style="list-style-type: none"> - Ausgang, low aktiv - Bestätigung, daß Adreß-, Daten- und Steuerbus in hochohmigem Zustand sind
\overline{CP}	Systemtakt (clock pulse)	<ul style="list-style-type: none"> - Takteingang für einphasigen TTL-Takt

Bedeutung der Flags

CY	F0	Carry-Flag: Übertrag: wird gesetzt bei Überlauf von Additionen und beim Borgen bei Subtraktionen. Es wird z. T. beim Schieben und Rotieren verwendet.
N	F1	Additions-Subtraktions-Flag: Es dient zur Steuerung des DAA-Befehls für BCD-Zahlen.
P/V	F2	Parity-overflow-Flag: Parität—Überlauf. Dieses Flag hat zwei Aufgaben: bei logischen Operationen Anzeige der Parität, bei arithmetischen Operationen Anzeige eines Überlaufs bei vorzeichenbehafteten Zweierkomplementen. Es ist also gesetzt bei Zahlen kleiner -128 oder größer $+127$.
—	F3	nicht vorhanden
H	F4	Halbbyte-Übertrags-Flag: gilt bezüglich BCD-Zahlen, wenn ein Überlauf des niederwertigen Byte auftritt, wichtig für den Befehl DAA.
—	F5	nicht vorhanden
Z	F6	Zero-Flag: Null: wird gesetzt, wenn Ergebnis im Akkumulator Null ist.
S	F7	Signum-Flag: Vorzeichen: dient der Verarbeitung von vorzeichenbehafteten Zahlen, speichert daher den letzten Wert von Bit 7 im Akkumulator.

Bei *Programmverzweigungen* werden folgende Entscheidungen bezüglich der Flags verwendet:

Wert = 1		Wert = 0		Flag	
C	Übertrag	NC	kein Übertrag	F1	CY
PE	gerade (even) bzw. Überlauf	PO	ungerade (odd) bzw. kein Überlauf	F2	P/V
Z	Null	NZ	nicht Null	F6	Z
M	minus (negativ)	P	positiv	F7	S

Flag-Operationen und CPU-Peripherie

‡ : Flag wird entsprechend dem Ergebnis verändert

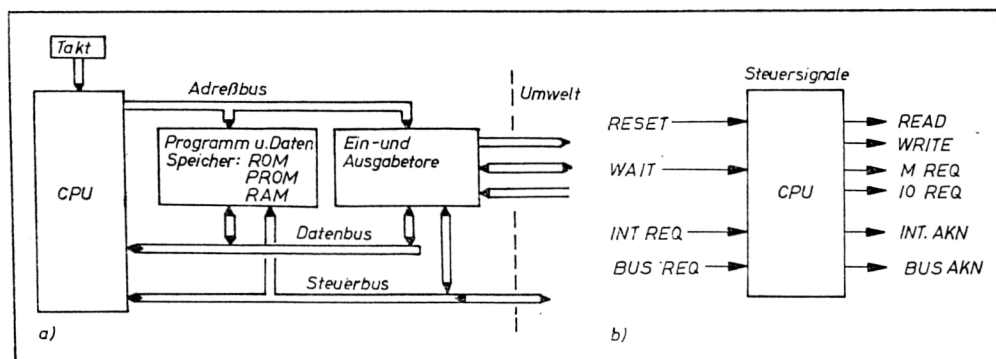
. : Flag wird nicht verändert

0: Flag wird gelöscht

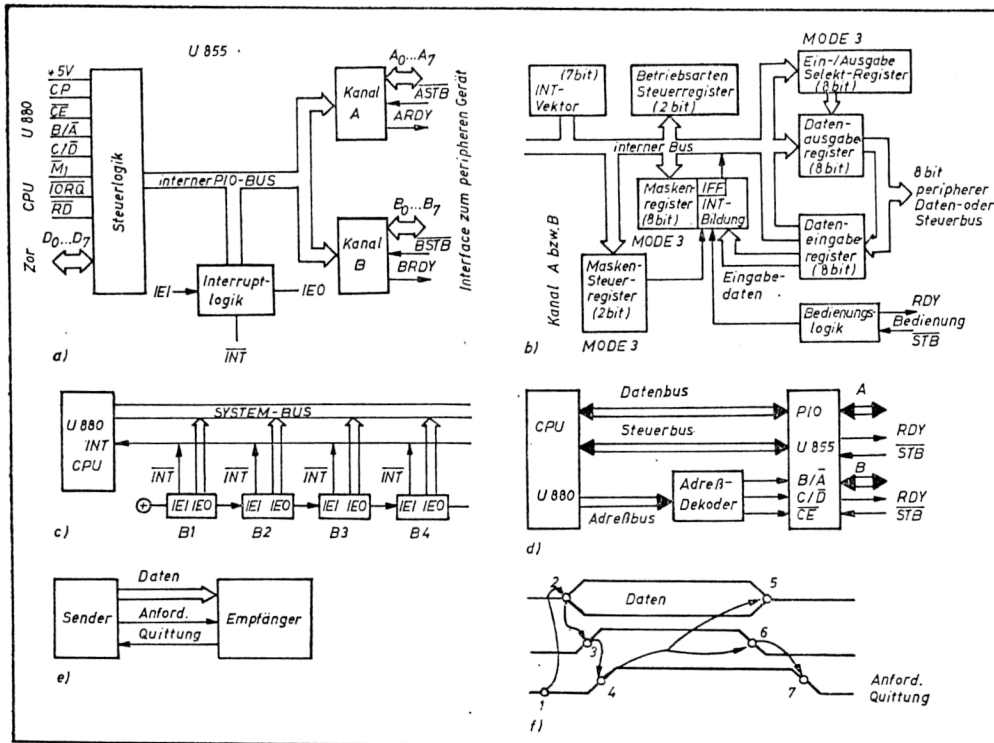
1: Flag wird gesetzt

X: Flag ist unbestimmt

Befehl	Flags						Bemerkungen
	C	Z	P/V	S	N	H	
ADD s, ADC s	‡	‡	V	‡	0	‡	8-bit-Addition oder Addition mit Übertrag 8-bit-Subtraktion, Subtraktion mit Übertrag, Vergleich und Negation des Akkumulators
SUB s, SBC s, CMP s, NEG	‡	‡	V	‡	1	‡	
AND s	0	‡	P	‡	0	1	logische Operationen
OR s, XOR s	0	‡	P	‡	0	0	
INC d	.	‡	V	‡	0	‡	8-bit-Erhöhung
DEC d	.	‡	V	‡	1	‡	8-bit-Erniedrigung
ADD HL, ss	‡	.	.	.	0	X	16-bit-Addition
ADC HL, ss	‡	‡	V	‡	0	X	16-bit-Addition mit Übertrag
SBC HL, ss	‡	‡	V	‡	1	X	16-bit-Subtraktion mit Übertrag
RLA, RLCA, RRA, RRCA	‡	.	.	.	0	0	zyklische Verschiebung – Akkumulator
RL s, RLC s, RR s, RRC s	‡	‡	P	‡	0	0	zyklische Verschiebung – Speicherplatz s
SLA s, SRA s, SRL s	‡	‡	P	‡	0	0	Verschiebung – Speicherplatz s
RLD, RRD	.	‡	P	‡	0	0	zyklische Verschiebung – Zahl links und rechts
DAA	‡	‡	P	‡	.	‡	Dezimalkorrektur – Akkumulator
CPL	1	1	Komplement des Akkumulators
SCF	1	.	.	.	0	0	Setzen des Übertrags
CCF	‡	.	.	.	0	X	Komplement des Übertrags
IN r, INF	.	‡	P	‡	0	0	Eingabe, indirekte Registeradresse
INI, IND, OUTI, OUTD	.	‡	X	X	1	X	Blockein- und -ausgabe, Z = 0, wenn B = 0, sonst Z = 1
INIR, INDR, OTIR, OTDR	.	1	X	X	1	X	Z = 0, wenn B = 0, sonst Z = 1
LDI, LDD	.	X	‡	X	0	0	Blocktransferbefehle
LDIR, LDDR	.	X	0	X	0	0	P/V = 1, wenn BC = 0, sonst P/V = 0
CPI, CPIR, CPD, CPDR	.	‡	‡	X	1	X	Blocksuchbefehle, Z = 1, wenn A = (HL), sonst Z = 0; P/V = 1, wenn BC ≠ 0, sonst P/V ≠ 0
LD A,I; LD A,R	.	‡	IFF2	‡	0	0	Inhalt des Interrupt-Annahmeflipflops 2 (IFF2) ins P/V-Flag überführt
BIT b,s	.	‡	X	X	0	1	Zustand des Bits b im Speicherplatz s ins Z-Flag überführt



PIO U 855



PIO U 855 Programmierung

Es gilt die positive Logik $1 \triangleq H$, $0 \triangleq L$.

Voraussetzungen: $CE = 0$; $C/\bar{D} = 1$

Kanalwahl: $B/\bar{A} = 0$: Kanal A; $B/\bar{A} = 1$: Kanal B

1. Eingabe des Interrupt-Vektors

$D0 = 0$ bestimmt Adresse: Interrupt-Register (INT-Vektor)
D7 bis D1 bestimmen den Low-Teil der Sprungadresse

2. Wahl der Betriebsart

D3 ... D0 = 1111 = 0FH bestimmt die Adresse: Betriebsartenregister
D7, D6 bestimmen Betriebsart = MODE; D5, D4 ohne Bedeutung

MODE	D7	D6	Betriebsart
0	0	0	Byteausgabe
1	0	1	Byteeingabe
2	1	0	bidirektional nur für A möglich (dann B nur noch Bitbetrieb)
3	1	1	Bitbetrieb. Es muß unmittelbar 2a folgen

2a. *Ein-Ausgabe-Wahlwort*: Adresse: Ein-Ausgabe-Selektregister
D7 ... D0: Bit-Wert = 1 Eingabe; Bit-Wert = 0 Ausgabe

3. Interrupt-Kontrollwort

D3 ... D0 = 0111 = 7H bestimmt Adressen: Interrupt-Flip-Flop und Maskensteuerregister
D7 = 1 Interrupt möglich

nur bei Mode 3	D6, D5	Interruptart:	D6		D5	
					0	1
			0	OR; Low	AND; Low	
			1	OR; High	AND; High	
		D4 bestimmt, ob folgt: ↓				

3a. *Maskierungswort* (bei zuvor D4 = 1): Adresse: Maskenregister

D7 ... D0: Bitwert = 1 Wert wird für Interrupt genutzt
Bitwert = 0 Wert wird nicht für Interrupt genutzt

4. Interrupt-Kontrollwort

D3 ... D0 = 0011 = 3H bestimmt Adresse: Interrupt-Flipflop
D7 = 1 Interrupt möglich; D4 bis D6 ohne Bedeutung

PIO U 855

Bedeutung und Funktion der Anschlußleitungen

Bedeutung und Funktion der Anschlußleitungen der PIO U 855

Folgende Anschlüsse werden mit den gleichnamigen der CPU U 880 verbunden:

\overline{INT} ; \overline{IORQ} ; \overline{MI} ; \overline{CP} ; \overline{RD} ; D0 bis D7

Zusätzliche Bedeutung besitzen dabei:

- \overline{RD} hieraus wird auch ein Schreibsignal intern generiert $\overline{WR} = \overline{RD} \wedge \overline{CE} \wedge \overline{IORQ}$
- $\overline{D0}$ bis $\overline{D7}$ übertragen Daten- und Steuerwörter
- \overline{IEI} und \overline{IEO} (interrupt enable in bzw. out) für Prioritätskette (daisy chain) siehe unten
- \overline{CE} (chip enable) aktiviert bei Signal Low die PIO; wird meist aus den Adressen A2 bis A7 logisch abgeleitet
- $\overline{B/A}$ Auswahl von Kanal (Port) A (Low) bzw. B (High); wird meist von A0 abgeleitet (siehe unten)
- $\overline{C/D}$ (control-/data-select) legt fest, ob auf Datenleitungen ein Datenwort (Low) oder ein Steuerwort (High) liegt; wird meist von A1 abgeleitet (siehe unten)
- A0 bis A7
B0 bis B7 bidirektionale Tristate-Busleitungen der Kanäle A bzw. B

Für den Handshake- bzw. Interrupt-Betrieb existieren je Kanal zwei Leitungen, deren Funktion von der Betriebsart (Mode 0 bis 4) abhängt.

A STB Strobe pulse = Quittierungssignal des peripheren Geräts der Kanäle A bzw. B

B STB

A RDY Ready = Bereitschaftssignal der Kanäle A bzw. B

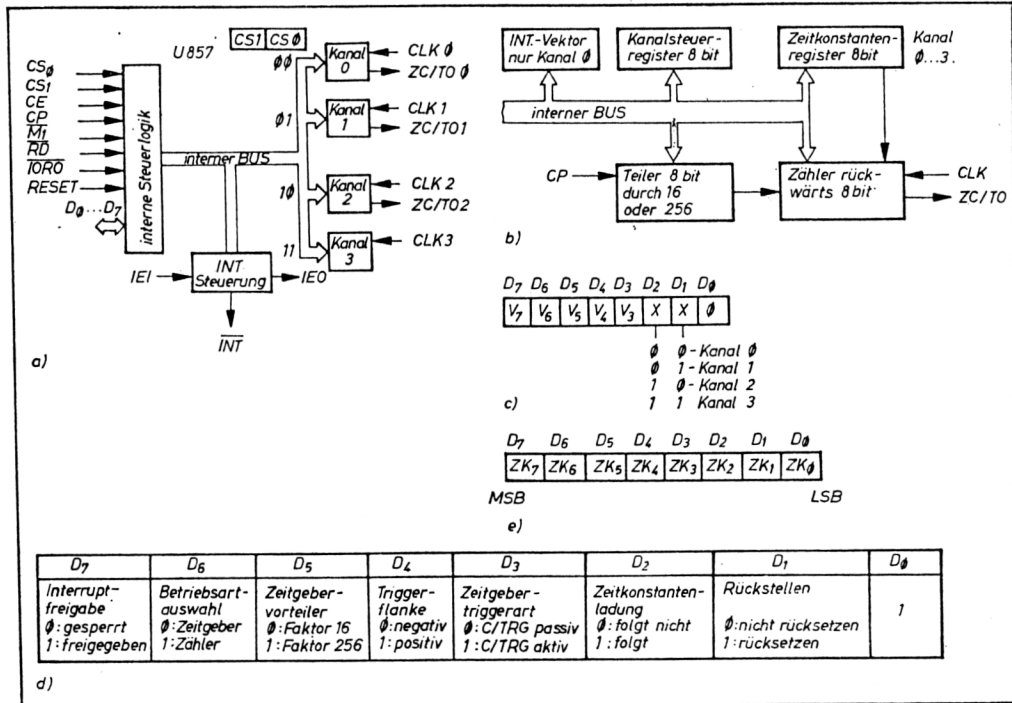
B RDY

Rücksetzen erfolgt beim Einschalten der PIO oder durch $\overline{MI} \wedge \overline{RD} \wedge \overline{IORQ}$ (dabei bleibt INT-Vektor erhalten)

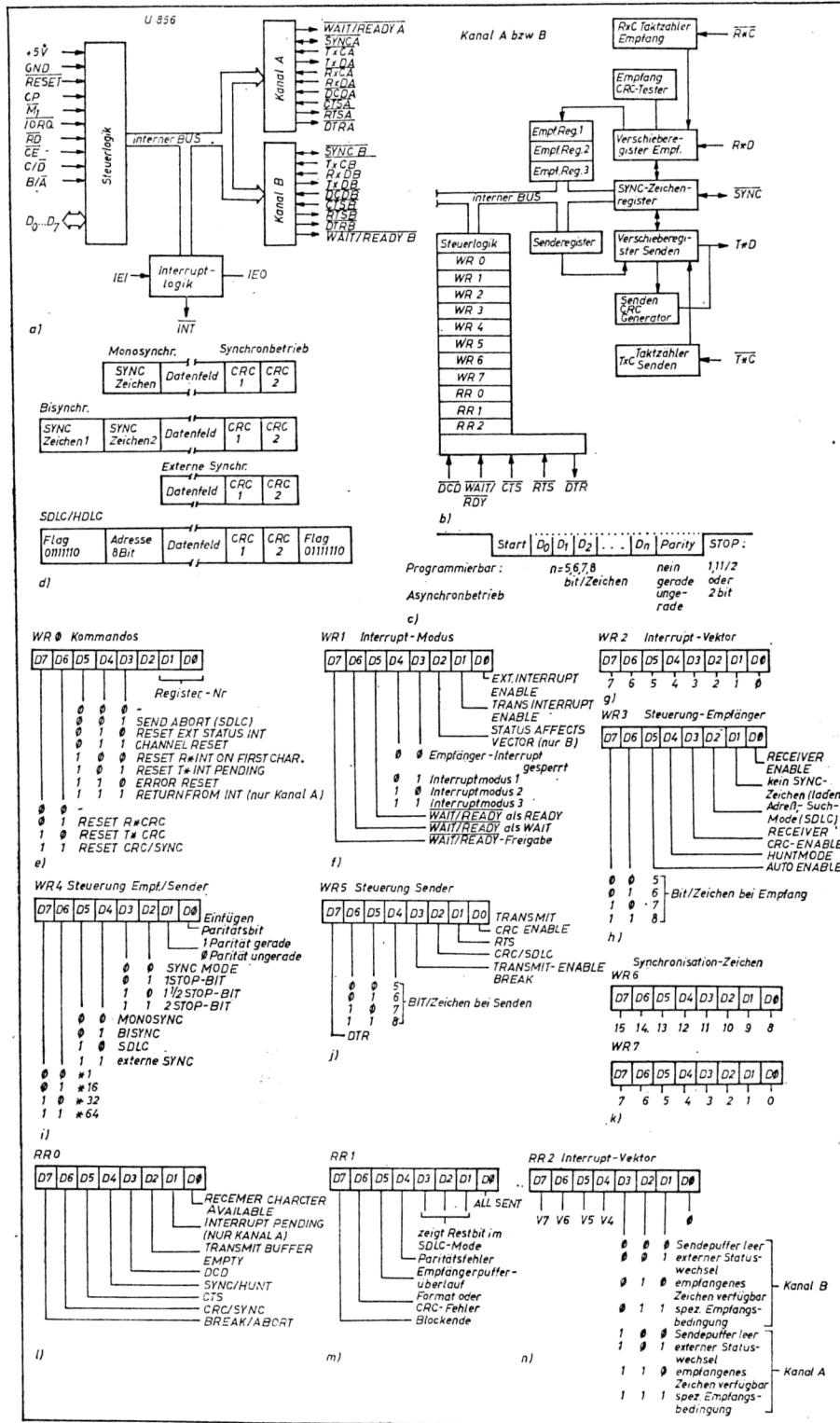
IEI	H	Interrupt-Forderung	möglich
	L		nicht möglich
IEO	H	Interrupt-Programm	läuft nicht
	L		läuft

Wert	A1	A0	Bedeutung
	C/D	B/A	
0	0	0	Datenwort für Kanal A
1	0	1	Datenwort für Kanal B
2	1	0	Steuerwort für Kanal A
3	1	1	Steuerwort für Kanal B

CTC U 857



SIO U 856



SIO U 856

Spezifische Anschlüsse

Die ergänzenden Bezeichnungen für den Kanal A bzw. B sind hinzuzufügen.

Auf Empfänger bezogen

R × D	Receive Data
R × C	Receiver Clock
DCD	Data Carrier Detect
DTR	Data terminal Ready

Auf Sender bezogen

T × D	Transmit Data
T × C	Transmitter Clock
CTS	Clear to Send
RTS	Request to Send

Allgemeine Anschlüsse

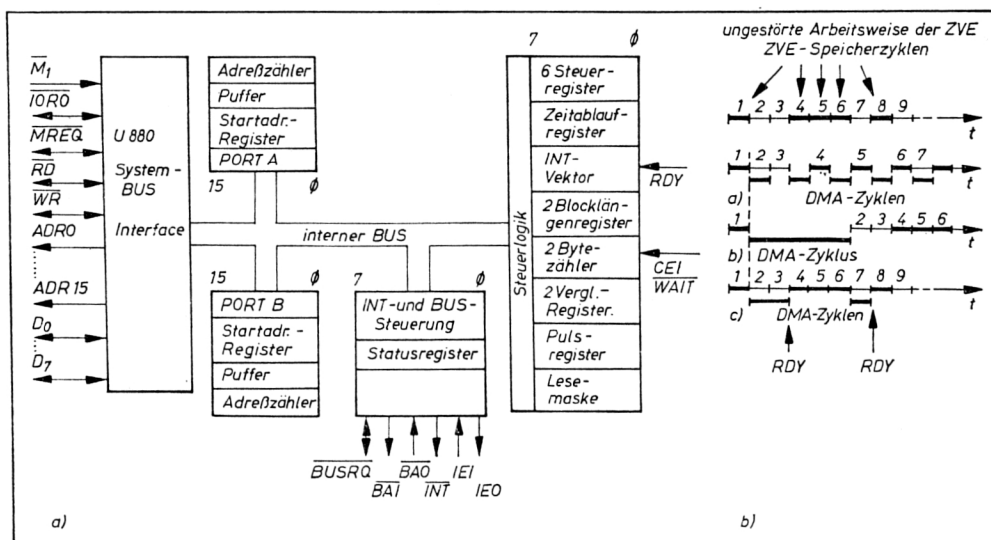
WAIT/RDY	Wait/Ready
SYNC	Synchronization

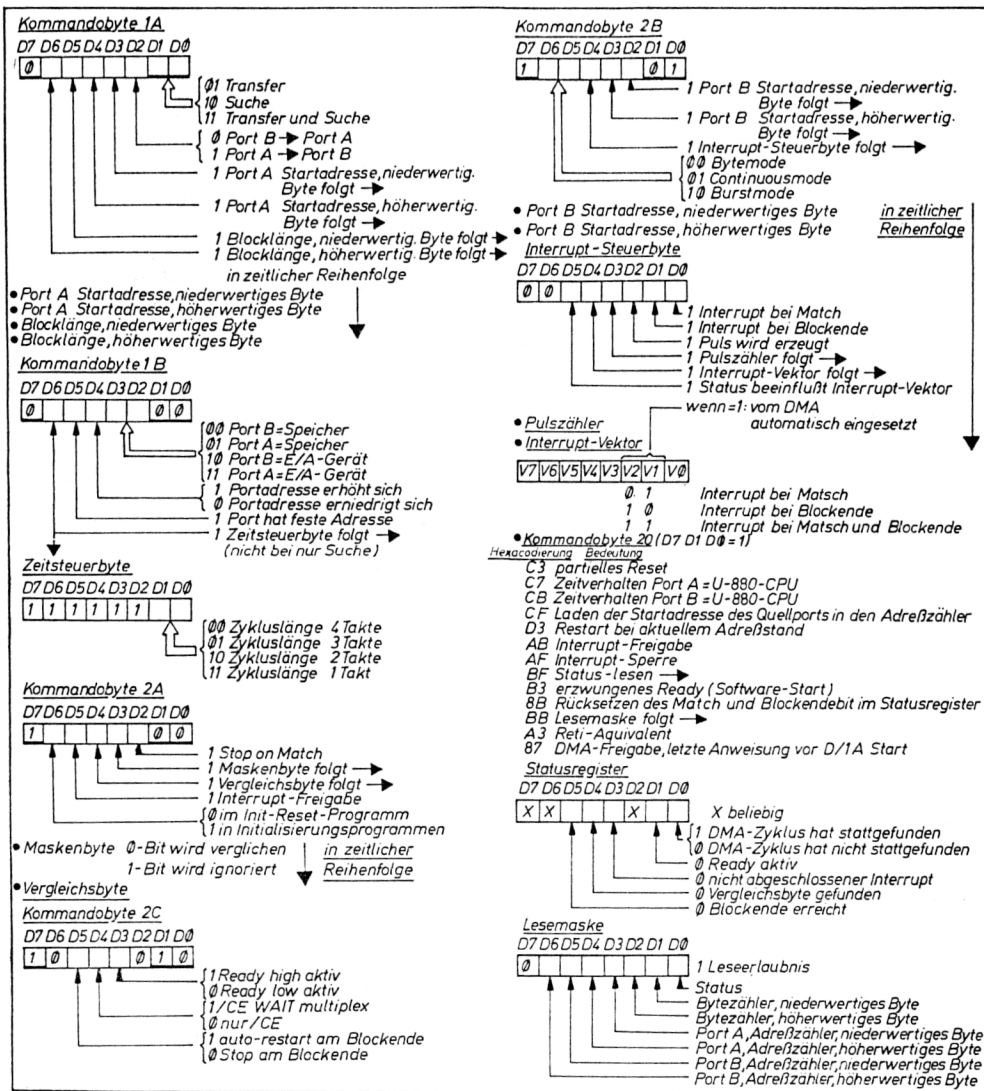
Dateneingang
Takteingang
Programmierbar ist, ob die Taktfrequenz 1-, 16-, 32- oder 64mal so hoch wie die Baudrate sein soll. Dies gilt insbesondere bei synchronem Betrieb. Die Datenrate darf höchstens $\frac{1}{4,5}$ des Systemtaktes CP sein. Bei einer Taktfrequenz von 2,5 MHz sind folglich Daten bis zu 550 kbit/s verarbeitbar.
Empfänger sperren (handshake)
Empfänger bereit (handshake)

Datenausgang
analog zu R × C, mit gleichem Faktor bei gleichem Kanal
Sender sperren (handshake)
Sendegesuch (handshake)

Generierung von Wartezuständen bzw. DMA-Steuerung
Ein- und Ausgang für spezielle externe Synchronisationen

DMA U 858





Sockelschaltungen

U880		Z80		U855		PIO		U856		SIO	
A11	1	40	A10	D2	1	40	D3	D1	1	40	D0
A12	2	39	A9	D7	2	39	D4	D3	2	39	D2
A13	3	38	A8	D6	3	38	D5	D5	3	38	D4
A14	4	37	A7	CE	4	37	MT	D7	4	37	D6
A15	5	36	A6	CID	5	36	TORQ	INT	5	36	TORQ
C	6	35	A5	BI \bar{A}	6	35	RD	IEI	6	35	CE
D4	7	34	A4	A7	7	34	B7	IEO	7	34	BI \bar{A}
D3	8	33	A3	A6	8	33	B6	MT	8	33	CID
D5	9	32	A2	A5	9	32	B5	U _{cc}	9	32	RD
D6	10	31	A1	A4	10	31	B4	WIRDYA	10	31	U _{ss}
U _{cc}	11	30	A0	U _{ss}	11	30	B3	SYNCA	11	30	WIRDYB
D2	12	29	U _{ss}	A3	12	29	B2	RxDA	12	29	SYNCB
D7	13	28	RFSH	A2	13	28	B1	RxCA	13	28	RxDB
D0	14	27	MT	A1	14	27	B0	TxCA	14	27	RxTxCB
D1	15	26	RESET	A0	15	26	U _{cc}	TxDA	15	26	TxDB
INT	16	25	BUSRQ	ASTB	16	25	C	DTRA	16	25	DTRB
NMT	17	24	WAIT	BSTB	17	24	IEI	RTSA	17	24	RTSB
HALT	18	23	BUSAK	ARDY	18	23	INT	CTSA	18	23	CTSB
MREQ	19	22	WR	D0	19	22	IEO	DCDA	19	22	DCDB
TORQ	20	21	RD	D1	20	21	BRDY	C	20	21	RESET

U857		CTC		U858		DMA	
D4	1	28	D3	A5	1	40	A6
D5	2	27	D2	A4	2	39	A7
D6	3	26	D1	A3	3	38	IEI
D7	4	25	D0	A2	4	37	INT
U _{ss}	5	24	U _{cc}	A1	5	36	IEO
RD	6	23	CITRG0	A0	6	35	D0
ZCITO 0	7	22	CITRG1	C	7	34	D1
ZCITO 1	8	21	CITRG2	WR	8	33	D2
ZCITO 2	9	20	CITRG3	RD	9	32	D3
TORQ	10	19	KS1	TORQ	10	31	D4
IEO	11	18	KS0	U _{cc}	11	30	U _{ss}
INT	12	17	RESET	MREQ	12	29	D5
IEI	13	16	CE	BA0	13	28	D6
MT	14	15	C	BA1	14	27	D7
				BUSRQ	15	26	MT
				CSIWAIT	16	25	RDY
				A15	17	24	A8
				A14	18	23	A9
				A13	19	22	A10
				A12	20	21	A11

Darstellung des internationalen 7-bit-Kodes für Texte

Tabelle 8.1.5. Darstellung des internationalen 7-bit-Kodes für Texte. Die Bedeutung der Steuerzeichen NULL, SOH usw. wird hier nicht weiter erklärt [B 24]. Die Zeichen (◀) mit den Dezimal-Nr. 91 bis 94 und 123 bis 126 können international unterschiedlich belegt werden. Im Deutschen liegen hier zuweilen die Umlaute und ß.

Bit	b6	b5	b4	b3	b2	b1	b0	Spalte								
								0	1	2	3	4	5	6	7	
								HEX								
								Zeile								
	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0
	0	0	1	0	0	1	0	0	1	0	1	0	0	0	0	0
	0	0	1	1	1	1	0	0	1	0	1	0	0	0	0	0
	0	1	0	0	0	0	0	4	0	0	0	0	0	0	0	0
	0	1	0	1	1	0	0	5	0	0	0	0	0	0	0	0
	0	1	1	0	0	1	0	6	0	0	0	0	0	0	0	0
	0	1	1	1	1	1	0	7	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0
	1	0	0	1	1	0	0	9	0	0	0	0	0	0	0	0
	1	0	1	0	1	0	0	10; A	0	0	0	0	0	0	0	0
	1	0	1	1	1	1	0	11; B	0	0	0	0	0	0	0	0
	1	1	0	0	0	0	0	12; C	0	0	0	0	0	0	0	0
	1	1	0	1	1	0	0	13; D	0	0	0	0	0	0	0	0
	1	1	1	0	0	1	0	14; E	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	0	15; F	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0
	0	0	1	0	0	1	0	0	1	0	1	0	0	0	0	0
	0	0	1	1	1	1	0	0	1	0	1	0	0	0	0	0
	1	0	0	0	0	0	0	8	(FE0) BS	CAN	()	8	H	X	h	x
	1	0	0	1	1	0	0	9	(FE1) HT	EM)	9	I	Y	i	y
	1	0	1	0	1	0	0	10; A	(FE2) LF	SUB	*	:	J	Z	j	z
	1	0	1	1	1	1	0	11; B	(FE3) VT	ESC	~	;	K	[(◌)◀	k	{(ü)◀
	1	1	0	0	0	0	0	12; C	(FE4) EF	(IS4) FS	,	<	L	[(◌)◀	l	{(ö)◀
	1	1	0	1	1	0	0	13; D	(FE5) CR	(IS3) GS	-	=	M	[(◌)◀	m	{(ü)◀
	1	1	1	0	0	1	0	14; E	SO	(IS2) RS	.	>	N	^	n	-(ß)
	1	1	1	1	1	1	0	15; F	SI	(IS1) US	/	?	O	~	o	DEL

3.
stock

Der Autor, Jahrgang 1930, studierte in Greifswald Physik, behandelte in seiner Diplomarbeit RC-Generatoren, in seiner Dissertation die Theorie der Frequenzmodulation und habilitierte sich mit einer Verallgemeinerung der Informationstheorie. 1969 verlieh ihm die Akademie der Wissenschaften der DDR die Professur.

Seine wissenschaftlichen Arbeiten, die u. a. in weit über hundert Publikationen sowie in Büchern zur Spannungstabilisation, 6 Bänden zur Information und zur Elektronik niedergelegt sind, betreffen vor allem Informationsspeicherung, Informationstheorie, Kybernetik und Elektronik. Großes Interesse bringt er ethischen, ästhetischen und philosophischen Problemen entgegen. Organisatorisch ist er in mehreren wissenschaftlichen und wissenschaftlich-strategischen Gremien tätig. Er war der erste Direktor und damit Gründer des Zentralinstituts für Kybernetik und Informationsprozesse der AdW der DDR.