

CPU U880D

Befehlsliste



**veb mikroelektronik › karl marx › erfurt**  
**stammbetrieb**



# CPU U880 D

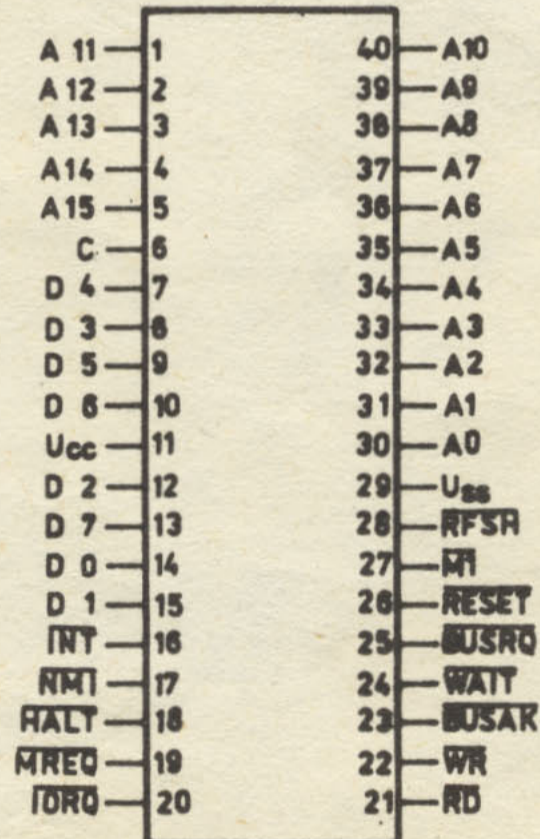
## Befehlskarte

Hauptregistersatz			Alternativsatz			
Akkumu- lator A	Flags F		Akkumu- lator A'	Flags F'		
B	C		B'	C'		} Register zur allgemeinen Verwendung
D	E		D'	E'		
H	L		H'	L'		

Interrupt Vektor I	Speicher Refresh R	
Index Register	IX	} Spezial- register
Index Register	IY	
Keller-Zeiger	SP	
Programmzähler	PC	

### U880 D - Registeraufbau

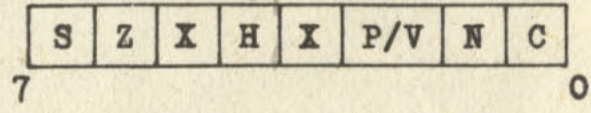


### U880 D - Anschlußbelegung



Befehl	Flags							Bemerkungen
	C	Z	P/V	S	N	H		
ADD s, ADC s	↑	↑	V	↑	0	↑	8-Bit-Addition oder Addition mit Übertrag	
SUB s, SBC s, CMP s, NEG	↑	↑	V	↑	1	↑	8-Bit-Subtraktion, Subtraktion mit Übertrag, Vergleich und Negation des Akkumulators	
AND s	0	↑	P	↑	0	1	Logische Operationen	
OR s, XOR s	0	↑	P	↑	0	0		
INC s	.	↑	V	↑	0	↑	8-Bit-Erhöhung	
DEC s	.	↑	V	↑	1	↑	8-Bit-Erniedrigung	
ADD HL,dd	↑	.	.	.	0	X	16-Bit-Addition	
ADC HL,dd	↑	↑	V	↑	0	X	16-Bit-Addition mit Übertrag	
SBC HL,dd	↑	↑	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	↑	.	↑	Dezimalen Einrichtung - 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	Block-Ein- 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	Blocktransfer-Befehle	
LDIR, LDDR	.	X	0	X	0	0	P/V = 1 wenn BC ≠ 0, sonst P/V = 0	
CPI, CPIR, CPD, CPDR	.	↑	↑	X	1	X	Block-Such-Befehle, 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-Akzeptanz-Flip-Flops 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	

723 64 32 16 8 4 2 1



X: Bit hat keine Bedeutung

Format des Flag-Registers



Symbol	Bedeutung
C	Übertragsflag. C = 1, wenn die Operation einen Übertrag vom MSB des Operanden oder des Ergebnisses erzeugt.
Z	Null-Flag. Z = 1, wenn das Ergebnis der Operation Null ist.
S	Vorzeichen-Flag. S = 1, wenn das MSB des Ergebnisses eins ist.
P/V	Paritäts- oder Überlauf-Flag. Parität (P) und Überlauf (V) benutzen das gleiche Flag. Logische Operationen beeinflussen das Flag entsprechend der Parität des Ergebnisses, arithmetische Operationen stellen dieses Flag entsprechend dem Überlauf des Ergebnisses. P/V = 1, wenn das Ergebnis paarig ist, P/V = 0, wenn das Ergebnis unpaarig ist. P/V = 1, wenn das Ergebnis einen Überlauf enthält.
H	Halbbyte-Übertragsflag. H = 1, wenn Addition oder Subtraktion einen Übertrag innerhalb von 4 Akkumulatorbits erzeugen.
N	Additions-/Subtraktionsflag. N = 1, wenn vorangegangene Operation eine Subtraktion war. H- und N-Flags werden für die Dezimalkorrektur (DAA) benutzt, um das Ergebnis einer Addition oder Subtraktion von gepackten BCD-Zahlen in das Format gepackter BCD-Zahlen zu wandeln.
‡	Flag wird entsprechend dem Ergebnis der Operation gestellt
.	Flag wird durch die Operation nicht beeinflusst
O	Flag wird durch die Operation gelöscht
1	Flag wird durch die Operation gesetzt
X	Flag unbestimmt
V	P/V-Flag entspricht dem Ergebnis-Überlauf der Operation
P	P/V-Flag entspricht der Parität des Ergebnisses der Operation
r	eines der U880D - Register A, B, C, D, E, H, L.
s	ein 8-Bit-Speicherplatz, der durch eine der für den jeweiligen Befehl zulässigen Adressierungsarten definiert ist.
dd	ein 16-Bit-Speicherplatz, der durch eine der für diesen Befehl zulässigen Adressierungsarten definiert ist.
ii	eines der zwei Indexregister IX oder IY
R	Auffrischzähler
n	8-Bit im Bereich 0 - 255
nn	16-Bit im Bereich 0 - 655 35
A	P/V-Flag ist 0, wenn das Ergebnis von BC-1 = 0, sonst P/V = 1
B	Z-Flag ist 1, wenn A = M, sonst Z = 0
IFF1	Interrupt-Annahme-Flip-Flop
IFF2	Interrupt-Zwischenspeicher-Flip-Flop
e	stellt die Abstandsangabe in der relativen Adressierungsart dar, bezogen auf das 1. Byte des Sprungbefehls, e ist ein Zweierkomplement mit Vorzeichen im Bereich -126 bis +129
e-2	ergibt im Operationscode die tatsächliche Adresse PC+e, da der Befehlszähler vor der Addition von e um 2 erhöht worden ist.
s <sub>b</sub>	bezeichnet das Bit b (0...7) des Speicherplatzes s
a	falls B-1 = 0, wird Z = 1 gesetzt, sonst Z = 0



Assembler Sprache	symbolische Operation	Flags					Operations-code	Bytes	M-Zyk-len	Takte	Bemerkung
		C	Z	P/V	S	H					
8 - Bit - Ladegruppe											
LD r <sub>1</sub> ,r <sub>2</sub>	r <sub>1</sub> ←r <sub>2</sub>	. . . . .	01	r <sub>1</sub>	r <sub>2</sub>		1	1	4	r <sub>1</sub> ,r <sub>2</sub>   Regi- ster	
LD r,n	r←n	. . . . .	00	r	110	- n -	2	2	7	000   B 001   C 010   D 011   E 100   H 101   L 111   A	
LD r,M	r←M	. . . . .	01	r	110		1	2	7	r <sub>1</sub> ,r <sub>2</sub> steht für eines der Register A,B,C,D,E,H,L.	
LD r,(IX+d)	r←(IX+d)	. . . . .	11	011	101	01 r 110 - d -	3	5	19		
LD r,(IY+d)	r←(IY+d)	. . . . .	11	111	101	01 r 110 - d -	3	5	19		
LD M,r	M←r	. . . . .	01	110	r		1	2	7		
LD (IX+d),r	(IX+d)←r	. . . . .	11	011	101	01 110 r - d -	3	5	19		
LD (IY+d),r	(IY+d)←r	. . . . .	11	111	101	01 110 r - d -	3	5	19		
LD M,n	M←n	. . . . .	00	110	110	- n -	2	3	10		
LD (IX+d),n	(IX+d)←n	. . . . .	11	011	101	00 110 110 - d - - n -	4	5	19		
LD (IY+d),n	(IY+d)←n	. . . . .	11	111	101	00 110 110 - d - - n -	4	5	19		
LD A,(BC)	A←(BC)	. . . . .	00	001	010		1	2	7		
LD A,(DE)	A←(DE)	. . . . .	00	011	010		1	2	7		
LD A,(nn)	A←(nn)	. . . . .	00	111	010	- n - - n -	3	4	13		
LD (BC),A	(BC)←A	. . . . .	00	000	010		1	2	7		
LD (DE),A	(DE)←A	. . . . .	00	010	010		1	2	7		
LD (nn),A	(nn)←A	. . . . .	00	110	010	- n - - n -	3	4	13		
LD A,I	A←I	. † IFF † 0 0	11	101	101	01 010 111	2	2	9		
LD A,R	A←R	. † IFF † 0 0	11	101	101	01 011 111	2	2	9		
LD I,A	I←A	. . . . .	11	101	101	01 000 111	2	2	9		
LD R,A	R←A	. . . . .	11	101	101	01 001 111	2	2	9		

16 - Bit - Ladegruppe

LD dd,nn	dd←nn	. . . . .	00	dd0	001	- n - - n -	3	3	10	dd   Paar 00   BC 01   DE 10   HL 11   SP
LD IX,nn	IX←nn	. . . . .	11	011	101	00 100 001 - n - - n -	4	4	14	
LD IY,nn	IY←nn	. . . . .	11	111	101	00 100 001 - n - - n -	4	4	14	
LD HL,(nn)	H←(nn+1) L←(nn)	. . . . .	00	101	010	- n - - n -	3	5	16	
LD dd,(nn)	ddH←(nn+1) ddL←(nn)	. . . . .	11	101	101	01 dd1 011 - n - - n -	4	6	20	



Assembler Sprache	symbolische Operation	Flags						Operationscode 76 543 210	By-tes	M-Zyk-len	Tak-te	Bemerkung
		C	Z	P/V	S	N	H					
LD IX, (nn)	IXH ← (nn+1) IXL ← (nn)	.	.	.	.	.	.	11 011 101 00 101 010 - n - - n -	4	6	20	
LD IY, (nn)	IYH ← (nn+1) IYL ← (nn)	.	.	.	.	.	.	11 111 101 00 101 010 - n - - n -	4	6	20	
LD (nn), HL	(nn+1) ← H (nn) ← L	.	.	.	.	.	.	00 100 010 - n - - n -	3	5	16	
LD (nn), dd	(nn+1) ← ddH (nn) ← ddL	.	.	.	.	.	.	11 101 101 01 dd0 011 - n - - n -	4	6	20	dd ist eines der Registerpaare BC, DE, HL, SP
LD (nn), IX	(nn+1) ← IXH (nn) ← IXL	.	.	.	.	.	.	11 011 101 00 100 010 - n - - n -	4	6	20	
LD (nn), IY	(nn+1) ← IYH (nn) ← IYL	.	.	.	.	.	.	11 111 101 00 100 010 - n - - n -	4	6	20	
LD SP, HL	SP ← HL	.	.	.	.	.	.	11 111 001	1	1	6	
LD SP, IX	SP ← IX	.	.	.	.	.	.	11 011 101 11 111 001	2	2	10	
LD SP, IY	SP ← IY	.	.	.	.	.	.	11 111 101 11 111 001	2	2	10	
PUSH qq	(SP-2) ← qqL (SP-1) ← qqH SP ← SP-2	.	.	.	.	.	.	11 qq0 101	1	3	11	qq   Paar 00   BC 01   DE 10   HL 11   AF
PUSH IX	(SP-2) ← IXL (SP-1) ← IXH SP ← SP-2	.	.	.	.	.	.	11 011 101 11 100 101	2	4	15	
PUSH IY	(SP-2) ← IYL (SP-1) ← IYH SP ← SP-2	.	.	.	.	.	.	11 111 101 11 100 101	2	4	15	qq ist eines der Registerpaare AF, BC, DE, HL.
POP qq	qqH ← (SP+1) qqL ← (SP) SP ← SP+2	.	.	.	.	.	.	11 qq0 001	1	3	10	(Paar)H bzw. (Paar)L bezieht sich auf die oberen bzw. unteren 8 Bits d. entspr. Registerpaares, d.h. BCL=C, AFH=A.
POP IX	IXH ← (SP+1) IXL ← (SP) SP ← SP+2	.	.	.	.	.	.	11 011 101 11 100 001	2	4	14	
POP IY	IYH ← (SP+1) IYL ← (SP) SP ← SP+2	.	.	.	.	.	.	11 111 101 11 100 001	2	4	14	

Austausch-, Blocktransfer- und Suchgruppe

EX DE, HL	DE ↔ HL	.	.	.	.	.	.	11 101 011	1	1	4	
EX AF	AF ↔ AF'	.	.	.	.	.	.	00 001 000	1	1	4	08
EXX	(BC) ↔ (BC') (DE) ↔ (DE') (HL) ↔ (HL')	.	.	.	.	.	.	11 011 001	1	1	4	Vertauschung Registersatz/ Alternativregistersatz
EX (SP), HL	H ↔ (SP+1) L ↔ (SP)	.	.	.	.	.	.	11 100 011	1	5	19	
EX (SP), IX	IXH ↔ (SP+1) IXL ↔ (SP)	.	.	.	.	.	.	11 011 101 11 100 011	2	6	23	
EX (SP), IY	IYH ↔ (SP+1) IYL ↔ (SP)	.	.	.	.	.	.	11 111 101 11 100 011	2	6	23	
LDI	(DE) ← M DE ← DE+1 HL ← HL+1 BC ← BC-1	.	X	.	.	X	0 0	11 101 101 10 100 000	2	4	16	
LDIR	(DE) ← M DE ← DE+1 HL ← HL+1 BC ← BC-1 Wiederholung bis BC=0	.	X	0	.	X	0 0	11 101 101 10 110 000	2 2	5 4	21 16	wenn BC ≠ 0 wenn BC = 0



Assembler Sprache	symbolische Operation	Flags C Z P/V S N H	Operations- code 76 543 210	By- tes	M- Zyk- len	Tak- te	Bemerkung
LDD	(DE) ← M DE ← DE-1 HL ← HL-1 BC ← BC-1	. X $\updownarrow$ X 0 0 A	11 101 101 10 101 000	2	4	16	
LDDR ED B8	(DE) ← M DE ← DE-1 HL ← HL-1 BC ← BC-1 Wiederholung bis BC=0	. X 0 X 0 0	11 101 101 10 111 000	2 2	5 4	21 16	wenn BC ≠ 0 wenn BC = 0
CPI	A ← M HL ← HL+1 BC ← BC-1	. $\updownarrow$ $\updownarrow$ X 1 X B A	11 101 101 10 100 001	2	4	16	
CPIR	A ← M HL ← HL+1 BC ← BC-1 Wiederholung bis BC=0 oder A=M	. $\updownarrow$ $\updownarrow$ X 1 X B A	11 101 101 10 110 001	2 2	5 4	21 16	wenn BC ≠ 0 u. A ≠ M wenn BC = 0 u. A = M
CPD	A ← M HL ← HL-1 BC ← BC-1	. $\updownarrow$ $\updownarrow$ X 1 X B A	11 101 101 10 101 001	2	4	16	
CPDR	A ← M HL ← HL-1 BC ← BC-1 Wiederholung bis BC=0 oder A=M	. $\updownarrow$ $\updownarrow$ X 1 X B A	11 101 101 10 111 001	2 2	5 4	21 16	wenn BC ≠ 0 u. A ≠ M wenn BC = 0 u. A = M

### 8-Bit-Arithmetik und logische Gruppe

Assembler Sprache	Operation	Flags C Z P/V S N H	Operations- code	By- tes	M- Zyk- len	Tak- te	Bemerkung																
ADD r	A ← A+r	$\updownarrow$ $\updownarrow$ V $\updownarrow$ 0	10 <span style="border: 1px solid black; padding: 2px;">000</span> r	1	1	4	<table border="1" style="display: inline-table; vertical-align: middle;"> <thead> <tr> <th>r</th> <th>Regi- ster</th> </tr> </thead> <tbody> <tr><td>000</td><td>B</td></tr> <tr><td>001</td><td>C</td></tr> <tr><td>010</td><td>D</td></tr> <tr><td>011</td><td>E</td></tr> <tr><td>100</td><td>H</td></tr> <tr><td>101</td><td>L</td></tr> <tr><td>111</td><td>A</td></tr> </tbody> </table>	r	Regi- ster	000	B	001	C	010	D	011	E	100	H	101	L	111	A
r	Regi- ster																						
000	B																						
001	C																						
010	D																						
011	E																						
100	H																						
101	L																						
111	A																						
ADD n	A ← A+n	$\updownarrow$ $\updownarrow$ V $\updownarrow$ 0	11 <span style="border: 1px solid black; padding: 2px;">000</span> 110 - n -	2	2	7																	
ADD M	A ← A+M	$\updownarrow$ $\updownarrow$ V $\updownarrow$ 0	10 <span style="border: 1px solid black; padding: 2px;">000</span> 110	1	2	7																	
ADD (IX+d)	A ← A+(IX+d)	$\updownarrow$ $\updownarrow$ V $\updownarrow$ 0	11 011 101 10 <span style="border: 1px solid black; padding: 2px;">000</span> 110 - d -	3	5	19																	
ADD (IY+d)	A ← A+(IY+d)	$\updownarrow$ $\updownarrow$ V $\updownarrow$ 0	11 111 101 10 <span style="border: 1px solid black; padding: 2px;">000</span> 110 - d -	3	5	19																	
ADC s	A ← A+s+CY	$\updownarrow$ $\updownarrow$ V $\updownarrow$ 0	<span style="border: 1px solid black; padding: 2px;">001</span>				s ist eines der r, n, M, (IX+d), (IY+d) wie beim ADD-Befehl  umrandete Bits ersetzen 000 in ADD-Befehl																
SUB s	A ← A-s	$\updownarrow$ $\updownarrow$ V $\updownarrow$ 1	<span style="border: 1px solid black; padding: 2px;">010</span>																				
SBC s	A ← A-s-CY	$\updownarrow$ $\updownarrow$ V $\updownarrow$ 1 $\updownarrow$	<span style="border: 1px solid black; padding: 2px;">011</span>																				
AND s	A ← A s	0 $\updownarrow$ P $\updownarrow$ 0 1	<span style="border: 1px solid black; padding: 2px;">100</span>																				
OR s	A ← A s	0 $\updownarrow$ P $\updownarrow$ 0 0	<span style="border: 1px solid black; padding: 2px;">110</span>																				
XOR s	A ← A s	0 $\updownarrow$ P $\updownarrow$ 0 0	<span style="border: 1px solid black; padding: 2px;">101</span>																				
CMP s	A - s	$\updownarrow$ $\updownarrow$ V $\updownarrow$ 1 $\updownarrow$	<span style="border: 1px solid black; padding: 2px;">111</span>																				
INC r	r ← r+1	. $\updownarrow$ V $\updownarrow$ 0 $\updownarrow$	00 r <span style="border: 1px solid black; padding: 2px;">100</span>	1	1	4																	
INC M	M ← M+1	. $\updownarrow$ V $\updownarrow$ 0 $\updownarrow$	00 110 <span style="border: 1px solid black; padding: 2px;">100</span>	1	3	11																	
INC (IX+d)	(IX+d) ← (IX+d)+1	. $\updownarrow$ V $\updownarrow$ 0 $\updownarrow$	11 011 101 00 110 <span style="border: 1px solid black; padding: 2px;">100</span> - d -	3	6	23																	
INC (IY+d)	(IY+d) ← (IY+d)+1	. $\updownarrow$ V $\updownarrow$ 0	11 111 101 00 110 <span style="border: 1px solid black; padding: 2px;">100</span> - d -	3	6	23																	



Assembler Sprache	symbolische Operation	Flags C Z P/V S N H	Operations- code 76 543 210	By- tes	M- Zyk- len	Tak- te	Bemerkung
DEC f	$f \leftarrow f-1$	. $\downarrow$ V $\downarrow$ 1 $\downarrow$	101				f ist eines der r, M, (IX+d), (IY+d) wie bei INC; gleiches Format u. Zustände wie INC. 100 durch 101 im Operationscode ersetzen.

Allgemeine Arithmetik und U880D - Steuergruppe

DAA	Wandelt AC-Inhalt in gepackt. BCD-Format nach Add o. Subtraktion v. gepackt. BCD-Zahlen	$\downarrow$ $\downarrow$ P $\downarrow$ . $\downarrow$	00 100 111	1	1	4	Dezimalkorrektur im Akkumulator
CPL	$A \leftarrow \bar{A}$	. . . . 1 1	00 101 111	1	1	4	Komplement d. Akkumulators; Binärkomplement
NEG	$A \leftarrow \bar{A}+1$	$\downarrow$ $\downarrow$ V $\downarrow$ 1 $\downarrow$	11 101 101 01 000 100	2	2	8	Negation AC; Zweierkomplement
CCF	$CY \leftarrow \bar{CY}$	$\downarrow$ . . . 0 X	00 111 111	1	1	4	Komplement d. Übertrags-Flag
SCF	$CY \leftarrow 1$	1 . . . 0 0	00 110 111	1	1	4	Setzen d. Übertrags-Flag
NOP	keine Operation	. . . . .	00 000 000	1	1	4	
HALT	U880D im HALT-Zustand	. . . . .	01 110 110	1	1	4	
DI	$IFF1 \leftarrow 0$ , $IFF2 \leftarrow 0$	. . . . .	11 110 011	1	1	4	
BI	$IFF1 \leftarrow 1$ , $IFF2 \leftarrow 1$	. . . . .	11 111 011	1	1	4	
IMO	Setzen d. Interrupt-Mode 0	. . . . .	11 101 101 01 000 110	2	2	8	
IM1	Setzen d. Interrupt-Mode 1	. . . . .	11 101 101 01 010 110	2	2	8	
IM2	Setzen d. Interrupt-Mode 2	. . . . .	11 101 101 01 011 110	2	2	8	

16-Bit-Arithmetik

ADD HL, dd	$HL \leftarrow HL+dd$	$\downarrow$ . . . 0 X	00 dd1 001	1	3	11	dd   Register
ADC HL, dd	$HL \leftarrow HL+dd+CY$	$\downarrow$ $\downarrow$ V $\downarrow$ 0 X	11 101 101 01 dd1 010	2	4	15	00   BC 01   DE 10   HL 11   SP
SBC HL, dd	$HL \leftarrow HL-dd-CY$	$\downarrow$ $\downarrow$ V $\downarrow$ 1 X	11 101 101 01 dd0 010	2	4	15	dd ist eines d. Registerpaare: BC, DE, HL, SP
ADD IX, pp	$IX \leftarrow IX+pp$	$\downarrow$ . . . 0 X	11 011 101 00 pp1 001	2	4	15	pp   Register 00   BC 01   DE 10   IX 11   SP pp ist eines d. Registerpaare: BC, DE, IX, SP

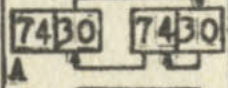
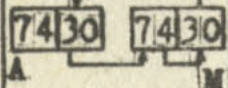


Assembler Sprache	symbolische Operation	Flags						Operationscode 76 543 210	By-tes	M-Zyk-len	Tak-te	Bemerkung
		C	Z	P/V	S	H	H					
ADD IY,pp	$IY \leftarrow IY + pp$	0	.	.	.	0	X	11 111 101 00 pp1 001	2	4	15	pp   Register 00   BC 01   DE 10   IY 11   SP
INC dd	$dd \leftarrow dd + 1$	.	.	.	.	.	.	00 dd0 011	1	1	6	
INC IX	$IX \leftarrow IX + 1$	.	.	.	.	.	.	11 011 101 00 100 011	2	2	10	
INC IY	$IY \leftarrow IY + 1$	.	.	.	.	.	.	11 111 101 00 100 011	2	2	10	pp ist eines d. Registerpaare; BC, DE, IY, SP
DEC dd	$dd \leftarrow dd - 1$	.	.	.	.	.	.	00 dd1 011	1	1	6	
DEC IX	$IX \leftarrow IX - 1$	.	.	.	.	.	.	11 011 101 00 101 011	2	2	10	
DEC IY	$IY \leftarrow IY - 1$	.	.	.	.	.	.	11 111 101 00 101 011	2	2	10	

Befehlsgruppe: Verschiebung und zyklische Verschiebung

RLCA		0	.	.	.	0	0	00 000 111	1	1	4	zyklische Verschiebung AC, linksherum
RLA		0	.	.	.	0	0	00 010 111	1	1	4	zyklische Verschiebung AC, nach links
RRCA		0	.	.	.	0	0	00 001 111	1	1	4	zyklische Verschiebung AC, rechtsherum
RRA		0	.	.	.	0	0	00 011 111	1	1	4	zyklische Verschiebung AC, nach rechts
RLC r		0	0	P	0	0	0	11 001 011 00 000 r	2	2	8	zyklische Verschiebung, Register, linksherum
RLC M		0	0	P	0	0	0	11 001 011 00 000 110	2	4	15	r   Register
RLC (IX+d)		0	0	P	0	0	0	11 011 101 11 001 011 - d - 00 000 110	4	6	23	000   B 001   C 010   D 011   E 100   H 101   L 111   A < 110 (HL)
RLC (IY+d)		0	0	P	0	0	0	11 111 101 11 001 011 - d - 00 000 110	4	6	23	
RL s		0	0	P	0	0	0	010				Befehlsformat u. Zustände wie bei RLC s; im Operationscode 000 durch d. umrandeten Bits ersetzen
RRC s		0	0	P	0	0	0	001				
RR s		0	0	P	0	0	0	011				
SLA s		0	0	P	0	0	0	100				
SRA s		0	0	P	0	0	0	101				
SRL s		0	0	P	0	0	0	111				



Assembler Sprache	symbolische Operation	Flags C Z P/V S N H	Operationscode 76 543 210	Bytes	M-Zyklen	Takte	Bemerkung
RLD		. $\downarrow$ P $\downarrow$ 0 0	11 101 101 01 101 111	2	5	18	zyklische Zahlenverschiebung links u. rechts zwischen AC u. M, Inhalt d. oberen Hälfte d. AC wird nicht beeinflusst
RRD		. $\downarrow$ P $\downarrow$ 0 0	11 101 101 01 100 111	2	5	18	

Gruppe Bit setzen, löschen und testen

BIT b,r	$Z \leftarrow \bar{r}_b$	. $\downarrow$ X X 0 1	11 001 011 01 b r	2	2	8	r   Register 000 B 001 C 010 D 011 E 100 H 101 L 111 A
BIT b,M	$Z \leftarrow \bar{M}_b$	. $\downarrow$ X X 0 1	11 001 011 01 b 110	2	3	12	b   getestetes Bit 000 0 001 1 010 2 011 3 100 4 101 5 110 6 111 7
BIT b, (IX+d)	$Z \leftarrow \overline{(IX+d)}_b$	. $\downarrow$ X X 0 1	11 011 101 11 001 011 - d - 01 b 110	4	5	20	
BIT b, (IY+d)	$Z \leftarrow \overline{(IY+d)}_b$	. $\downarrow$ X X 0 1	11 111 101 11 001 011 - d - 01 b 110	4	5	20	
SET b,r	$r_b \leftarrow 1$	. . . . .	11 001 011 11 b r	2	2	8	
SET b,M	$M_b \leftarrow 1$	. . . . .	11 001 011 11 b 110	2	4	15	
SET b, (IX+d)	$(IX+d)_b \leftarrow 1$	. . . . .	11 011 101 11 001 011 - d - 11 b 110	4	6	23	
SET b, (IY+d)	$(IY+d)_b \leftarrow 1$	. . . . .	11 111 101 11 001 011 - d - 11 b 110	4	6	23	
RES b,s	$s_b \leftarrow 0$ $s=r, M$ $(IX+d),$ $(IY+d)$		10				Zur Bildung d. neuen Operationscodes 11 in SET durch 10 ersetzen. Flags u. Zeiten wie bei SET

Befehlsgruppe: Sprünge

JMP nn	$PC \leftarrow nn$	. . . . .	11 000 011 - n - - n -	3	3	10	
JPcc nn	wenn Bedingung cc wahr ist, $PC \leftarrow nn$ , sonst weiter	. . . . .	11 cc 010 - n - - n -	3	3	10	cc   Bedingung 000 NZ nicht Null 001 Z Null 010 NC kein Übertrag 011 C Übertrag 100 PO unpaar. 101 PE paarig 110 P Vorzeich. positiv 111 M Vorzeich. negativ
JR e	$PC \leftarrow PC+e$	. . . . .	00 011 000 - e-2 -	2	3	12	
JRC e	wenn C=0 kein Sprung wenn C=1 $PC \leftarrow PC+e$	. . . . .	00 111 000 - e-2 -	2	2	7	Beding. nicht erfüllt
JRNC e	wenn C=1 kein Sprung wenn C=0 $PC \leftarrow PC+e$	. . . . .	00 110 000 - e-2 -	2	2	7	Beding. nicht erfüllt
				2	3	12	Beding. erfüllt.



Assembler Sprache	symbolische Operation	Flags						Operationscode 76 543 210	Bytes	M-Zyklen	Takte	Bemerkung
		C	Z	P/V	S	N	H					
JRZ e	wenn Z=0, kein Sprung wenn Z=1, PC←PC+e	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	00 101 000 - e-2 -	2	2	7	Beding. nicht erfüllt Beding. erfüllt	
								2	3	12		
JRNZ e	wenn Z=1, kein Sprung wenn Z=0, PC←PC+e	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	00 100 000 - e-2 -	2	2	7	Beding. nicht erfüllt Beding. erfüllt	
								2	3	12		
JMP M	PC←M	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	11 101 001	1	1	4		
JMP (IX)	PC←IX	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	11 011 101	2	2	8		
							11 101 001					
JMP (IY)	PC←IY	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	11 111 101	2	2	8		
							11 101 001					
DJNZ e	B←B-1 wenn B=0, kein Sprung wenn B≠0, PC←PC+e	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	00 010 000 - e-2 -	2	2	8	wenn B=0	
								2	3	13	wenn B≠0	

Befehlsgruppe: Unterprogrammaufruf und Rücksprung

CALL nn	(SP-1)←PCH (SP-2)←PCL PC←nn SP←SP-2	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	11 001 101 - n - - n -	3	5	17	
CA cc nn	wenn Bed. cc falsch ist, kein Sprung, sonst wie CALL nn	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	11 cc 100 - n -	3	3	10	wenn cc falsch ist
							- n -	3	5	17	wenn cc wahr ist
RET	PCL←(SP) PCH←(SP+1) SP←SP+2	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	11 001 001	1	3	10	
Rcc	wenn Bed. cc falsch ist, kein Sprung, sonst wie RET	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	11 cc 000	1	1	5	wenn cc falsch ist
								1	3	11	wenn cc wahr ist
RETI	Rücksprung v. Inter- rupt	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	11 101 101 01 001 101	2	4	14	cc   Bedingung
RETN	Rücksprung v. nicht maskierb. Interrupt	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	11 101 101 01 000 101	2	4	14	000 NZ nicht Null
											001 Z Null
RST p	(SP-1)←PCH (SP-2)←PCL PCH←0 PCL←p SP←SP-2	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	11 t 111	1	3	11	010 NC kein Übertrag
											011 C Übertrag
											100 PO unpaar.
											101 PE paarig
											110 P Vorz.pos.
											111 M Vorz.neg.
											t   p
											000 00H
											001 08H
											010 10H
				011 18H							
				100 20H							
				101 28H							
				110 30H							
				111 38H							

Befehlsgruppe: Ein- und Ausgabe

IN n	A←(n)	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	11 011 011 - n -	2	3	11	n zu A0 - A 7 AC zu A8 - A15
IN r INF	r←(C) wenn r=110, werden nur Flags ge- stellt	. † P † 0 0	. † P † 0 0	. † P † 0 0	. † P † 0 0	. † P † 0 0	11 101 101	2	3	12	C zu A0 - A 7 B zu A8 - A15
							01 r 000				
INI	M←(C) B←B-1 HL←HL+1	. † X X 1 X	. † X X 1 X	. † X X 1 X	. † X X 1 X	. † X X 1 X	11 101 101 10 100 010	2	4	16	C zu A0 - A 7 B zu A8 - A15



Assembler Sprache	symbolische Operation	Flags						Operationscode 76 543 210	Bytes	M-Zyklen	Tak- te	Bemerkung
		C	Z	P/V	S	N	H					
INIR	M ← (C) B ← B-1 HL ← HL+1 Wiederholg. bis B=0	. 1 X X 1 X					11 101 101 10 110 010	2	5	21	C zu A0 - A 7 B zu A8 - A15	
								2	B≠0 4 B=0	16		
IND	M ← (C) B ← B-1 HL ← HL-1	. ↓	X	X	1	X	11 101 101 10 101 010	2	4	16	C zu A0 - A 7 B zu A8 - A15	
INDR	M ← (C) B ← B-1 HL ← HL-1 Wiederholg. bis B=0	. 1 X X 1 X					11 101 101 10 111 010	2	5	21	C zu A0 - A 7 B zu A8 - A15	
								2	B≠0 4 B=0	16		
OUT n	(n) ← A	. . . . .					11 010 011 - n -	2	3	11	n zu A0 - A 7 AC zu A8 - A15	
OUT r	(C) ← r	. . . . .					11 101 101 01 r 001	2	3	12	C zu A0 - A 7 B zu A8 - A15	
OUTI	(C) ← M B ← B-1 HL ← HL+1	. ↓	X	X	1	X	11 101 101 10 100 011	2	4	16	C zu A0 - A 7 B zu A8 - A15	
OTIR	(C) ← M B ← B-1 HL ← HL+1 Wiederholg. bis B=0	. 1 X X 1 X					11 101 101 10 110 011	2	5	21	C zu A0 - A 7 B zu A8 - A15	
								2	B≠0 4 B=0	16		
OUTD	(C) ← M B ← B-1 HL ← HL-1	. ↓	X	X	1	X	11 101 101 10 101 011	2	4	16	C zu A0 - A 7 B zu A8 - A15	
OTDR	(C) ← M B ← B-1 HL ← HL-1 Wiederholg. bis B=0	. 1 X X 1 X					11 101 101 10 111 011	2	5	21	C zu A0 - A 7 B zu A8 - A15	
								2	B≠0 4 B=0	16		



Befehlsliste  
des USSO D -  
sortiert nach  
dem OP-Code

OP-Code	Mnemonic
00	NOP
01 nn	LD BC, nn
02	LD (BC), A
03	INC BC
04	INC B
05	DEC B
06 n	LD B, n
07	RLCA
08	EXAF
09	ADD HL, BC
0A	LD A, (BC)
0B	DEC BC
0C	INC C
0D	DEC C
0E n	LD C, n
0F	RRCA
10 e	DJNZ e
11 nn	LD DE, nn
12	LD (DE), A
13	INC DE
14	INC D
15	DEC D
16 n	LD D, n
17	RLA
18 e	JR e
19	ADD HL, DE
1A	LD A, (DE)
1B	DEC DE
1C	INC E
1D	DEC E
1E n	LD E, n
1F	RRA
20 e	JRNZ e
21 nn	LD HL, nn
22 nn	LD (nn), HL
23	INC HL
24	INC H
25	DEC H
26 n	LD H, n
27	DAA
28 e	JRZ e
29	ADD HL, HL
2A nn	LD HL, (nn)
2B	DEC HL
2C	INC L
2D	DEC L
2E n	LD L, n
2F	CPL
30 e	JRNC e
31 nn	LD SP, nn
32 nn	LD (nn), A
33	INC SP
34	INC M
35	DEC M
36 n	LD M, n
37	SCF
38 e	JRC e
39	ADD HL, SP
3A nn	LD A, (nn)
3B	DEC SP
3C	INC A
3D	DEC A
3E n	LD A, n
3F	CGF
40	LD B, B
41	LD B, C
42	LD B, D
43	LD B, E
44	LD B, H
45	LD B, L
46	LD B, M
47	LD B, A
48	LD C, B
49	LD C, C
4A	LD C, D
4B	LD C, E
4C	LD C, H
4D	LD C, L
4E	LD C, M

OP-Code	Mnemonic
4F	LD C, A
50	LD D, B
51	LD D, C
52	LD D, D
53	LD D, E
54	LD D, H
55	LD D, L
56	LD D, M
57	LD D, A
58	LD E, B
59	LD E, C
5A	LD E, D
5B	LD E, E
5C	LD E, H
5D	LD E, L
5E	LD E, M
5F	LD E, A
60	LD H, B
61	LD H, C
62	LD H, D
63	LD H, E
64	LD H, H
65	LD H, L
66	LD H, M
67	LD H, A
68	LD L, B
69	LD L, C
6A	LD L, D
6B	LD L, E
6C	LD L, H
6D	LD L, L
6E	LD L, M
6F	LD L, A
70	LD M, B
71	LD M, C
72	LD M, D
73	LD M, E
74	LD M, H
75	LD M, L
76	HALT
77	LD M, A
78	LD A, B
79	LD A, C
7A	LD A, D
7B	LD A, E
7C	LD A, H
7D	LD A, L
7E	LD A, M
7F	LD A, A
80	ADD B
81	ADD C
82	ADD D
83	ADD E
84	ADD H
85	ADD L
86	ADD M
87	ADD A
88	ADC B
89	ADC C
8A	ADC D
8B	ADC E
8C	ADC H
8D	ADC L
8E	ADC M
8F	ADC A
90	SUB B
91	SUB C
92	SUB D
93	SUB E
94	SUB H
95	SUB L
96	SUB M
97	SUB A
98	SBC B
99	SBC C
9A	SBC D
9B	SBC E
9C	SBC H
9D	SBC L
9E	SBC M
9F	SBC A
A0	AND B
A1	AND C
A2	AND D
A3	AND E

OP-Code	Mnemonic
A4	AND H
A5	AND L
A6	AND M
A7	AND A
A8	XOR B
A9	XOR C
AA	XOR D
AB	XOR E
AC	XOR H
AD	XOR L
AE	XOR M
AF	XOR A
B0	OR B
B1	OR C
B2	OR D
B3	OR E
B4	OR H
B5	OR L
B6	OR M
B7	OR A
B8	CP B
B9	CP C
BA	CP D
BB	CP E
BC	CP H
BD	CP L
BE	CP M
BF	CP A
C0	RNZ
C1	POP BC
C2 nn	JPNZ nn
C3 nn	JMP nn
C4 nn	CANZ nn
C5	PUSH BC
C6 n	ADD A, n
C7	RST 0
C8	RZ
C9	RET
CA nn	JPZ nn
CB nn	CAZ nn
CC nn	CALL nn
CE n	ADC n
CF	RST 8
D0	RNC
D1	POP DE
D2 nn	JPNC nn
D3 n	OUT n
D4 nn	CANC nn
D5	PUSH DE
D6 n	SUB n
D7	RST 10H
D8	RC
D9	EXX
DA nn	JPC nn
DB nn	IN A, n
DC nn	CAC nn
DE n	SBC n
DF	RST 18H
E0	RPO
E1	POP HL
E2 nn	JPPO nn
E3	EX (SP), HL
E4 nn	CAPO nn
E5	PUSH HL
E6 n	AND n
E7	RST 20H
E8	RPE
E9	JP M
EA nn	JPPE nn
EB	EX DE, HL
EC nn	CAPE nn
EE	XOR n
EF	RST 28H
F0	RP
F1	POP AF
F2 nn	JPP nn
F3	DI
F4 nn	CAP nn
F5	PUSH AF
F6	OR n
F7	RST 30H
F8	RM
F9	LD SP, HL
FA nn	JPM nn
FB	RI



OP-Code	Mnemonic
FC nn	CAM nn
FE n	GP n <i>CM</i>
FF	RST 38H
CB00	RLC B
CB01	RLC C
CB02	RLC D
CB03	RLC E
CB04	RLC H
CB05	RLC L
CB06	RLC M
CB07	RLC A
CB08	RRC B
CB09	RRC C
CB0A	RRC D
CB0B	RRC E
CB0C	RRC H
CB0D	RRC L
CB0E	RRC M
CB0F	RRC A
CB10	RL B
CB11	RL C
CB12	RL D
CB13	RL E
CB14	RL H
CB15	RL L
CB16	RL M
CB17	RL A
CB18	RR B
CB19	RR C
CB1A	RR D
CB1B	RR E
CB1C	RR H
CB1D	RR L
CB1E	RR M
CB1F	RR A
CB20	SLA B
CB21	SLA C
CB22	SLA D
CB23	SLA E
CB24	SLA H
CB25	SLA L
CB26	SLA M
CB27	SLA A
CB28	SRA B
CB29	SRA C
CB2A	SRA D
CB2B	SRA E
CB2C	SRA H
CB2D	SRA L
CB2E	SRA M
CB2F	SRA A
CB38	SRL B
CB39	SRL C
CB3A	SRL D
CB3B	SRL E
CB3C	SRL H
CB3D	SRL L
CB3E	SRL M
CB3F	SRL A
CB40	BIT 0,B
CB41	BIT 0,C
CB42	BIT 0,D
CB43	BIT 0,E
CB44	BIT 0,H
CB45	BIT 0,L
CB46	BIT 0,M
CB47	BIT 0,A
CB48	BIT 1,B
CB49	BIT 1,C
CB4A	BIT 1,D
CB4B	BIT 1,E
CB4C	BIT 1,H
CB4D	BIT 1,L
CB4E	BIT 1,M
CB4F	BIT 1,A
CB50	BIT 2,B
CB51	BIT 2,C
CB52	BIT 2,D
CB53	BIT 2,E
CB54	BIT 2,H
CB55	BIT 2,L
CB56	BIT 2,M
CB57	BIT 2,A
CB58	BIT 3,B
CB59	BIT 3,C

OP-Code	Mnemonic
CB5A	BIT 3,D
CB5B	BIT 3,E
CB5C	BIT 3,H
CB5D	BIT 3,L
CB5E	BIT 3,M
CB5F	BIT 3,A
CB60	BIT 4,B
CB61	BIT 4,C
CB62	BIT 4,D
CB63	BIT 4,E
CB64	BIT 4,H
CB65	BIT 4,L
CB66	BIT 4,M
CB67	BIT 4,A
CB68	BIT 5,B
CB69	BIT 5,C
CB6A	BIT 5,D
CB6B	BIT 5,E
CB6C	BIT 5,H
CB6D	BIT 5,L
CB6E	BIT 5,M
CB6F	BIT 5,A
CB70	BIT 6,B
CB71	BIT 6,C
CB72	BIT 6,D
CB73	BIT 6,E
CB74	BIT 6,H
CB75	BIT 6,L
CB76	BIT 6,M
CB77	BIT 6,A
CB78	BIT 7,B
CB79	BIT 7,C
CB7A	BIT 7,D
CB7B	BIT 7,E
CB7C	BIT 7,H
CB7D	BIT 7,L
CB7E	BIT 7,M
CB7F	BIT 7,A
CB80	RES 0,B
CB81	RES 0,C
CB82	RES 0,D
CB83	RES 0,E
CB84	RES 0,H
CB85	RES 0,L
CB86	RES 0,M
CB87	RES 0,A
CB88	RES 1,B
CB89	RES 1,C
CB8A	RES 1,D
CB8B	RES 1,E
CB8C	RES 1,H
CB8D	RES 1,L
CB8E	RES 1,M
CB8F	RES 1,A
CB90	RES 2,B
CB91	RES 2,C
CB92	RES 2,D
CB93	RES 2,E
CB94	RES 2,H
CB95	RES 2,L
CB96	RES 2,M
CB97	RES 2,A
CB98	RES 3,B
CB99	RES 3,C
CB9A	RES 3,D
CB9B	RES 3,E
CB9C	RES 3,H
CB9D	RES 3,L
CB9E	RES 3,M
CB9F	RES 3,A
CBA0	RES 4,B
CBA1	RES 4,C
CBA2	RES 4,D
CBA3	RES 4,E
CBA4	RES 4,H
CBA5	RES 4,L
CBA6	RES 4,M
CBA7	RES 4,A
CBA8	RES 5,B
CBA9	RES 5,C
CBAA	RES 5,D
CBAB	RES 5,E
CBAC	RES 5,H
CBAD	RES 5,L

OP-Code	Mnemonic
CBAE	RES 5,M
CBAF	RES 5,A
CBBO	RES 6,B
CBB1	RES 6,C
CBB2	RES 6,D
CBB3	RES 6,E
CBB4	RES 6,H
CBB5	RES 6,L
CBB6	RES 6,M
CBB7	RES 6,A
CBB8	RES 7,B
CBB9	RES 7,C
CBBA	RES 7,D
CBBB	RES 7,E
CBBC	RES 7,H
CBBD	RES 7,L
CBBE	RES 7,M
CBBF	RES 7,A
CBC0	SET 0,B
CBC1	SET 0,C
CBC2	SET 0,D
CBC3	SET 0,E
CBC4	SET 0,H
CBC5	SET 0,L
CBC6	SET 0,M
CBC7	SET 0,A
CBC8	SET 1,B
CBC9	SET 1,C
CBCA	SET 1,D
CBCB	SET 1,E
CBCC	SET 1,H
CBCD	SET 1,L
CBCE	SET 1,M
CBCF	SET 1,A
CBDO	SET 2,B
CBD1	SET 2,C
CBD2	SET 2,D
CBD3	SET 2,E
CBD4	SET 2,H
CBD5	SET 2,L
CBD6	SET 2,M
CBD7	SET 2,A
CBD8	SET 3,B
CBD9	SET 3,C
CBDA	SET 3,D
CBDB	SET 3,E
CBDC	SET 3,H
CBDD	SET 3,L
CBDE	SET 3,M
CBDF	SET 3,A
CBE0	SET 4,B
CBE1	SET 4,C
CBE2	SET 4,D
CBE3	SET 4,E
CBE4	SET 4,H
CBE5	SET 4,L
CBE6	SET 4,M
CBE7	SET 4,A
CBE8	SET 5,B
CBE9	SET 5,C
CBEA	SET 5,D
CBEB	SET 5,E
CBEC	SET 5,H
CBED	SET 5,L
CBEE	SET 5,M
CBEF	SET 5,A
CBFO	SET 6,B
CBF1	SET 6,C
CBF2	SET 6,D
CBF3	SET 6,E
CBF4	SET 6,H
CBF5	SET 6,L
CBF6	SET 6,M
CBF7	SET 6,A
CBF8	SET 7,B
CBF9	SET 7,C
CBFA	SET 7,D
CBFB	SET 7,E
CBFC	SET 7,H
CBFD	SET 7,L
CBFE	SET 7,M
CBFF	SET 7,A
DD09	ADD IX, BC
DD19	ADD IX, DE



Op-Code	Mnemonic
DD21 nn	LD IX, nn
DD22 nn	LD (nn), IX
DD23	INC IX
DD29	ADD IX, IX
DD2A nn	LD IX, (nn)
DD2B	DEC IX
DD34 d	INC (IX+d)
DD35 d	DEC (IX+d)
DD36 dn	LD (IX+d), n
DD39	ADD IX, SP
DD46 d	LD B, (IX+d)
DD4E d	LD C, (IX+d)
DD56 d	LD D, (IX+d)
DD5E d	LD E, (IX+d)
DD66 d	LD H, (IX+d)
DD6E d	LD L, (IX+d)
DD70 d	LD (IX+d), B
DD71 d	LD (IX+d), C
DD72 d	LD (IX+d), D
DD73 d	LD (IX+d), E
DD74 d	LD (IX+d), H
DD75 d	LD (IX+d), L
DD77 d	LD (IX+d), A
DD7E d	LD A, (IX+d)
DD86 d	ADD (IX+d)
DD8E d	ADC (IX+d)
DD96 d	SUB (IX+d)
DD9E d	SBC (IX+d)
DDA6 d	AND (IX+d)
DDAE d	XOR (IX+d)
DDB6 d	OR (IX+d)
DDBE d	CP (IX+d)
DDE1	POP IX
DDE3	EX (SP), IX
DDE5	PUSH IX
DDE9	JMP (IX)
DDF9	LD SP, IX
DDCB d06	RLC (IX+d)
DDCB d0E	RRC (IX+d)
DDCB d16	RL (IX+d)
DDCB d1E	RR (IX+d)
DDCB d26	SLA (IX+d)
DDCB d2E	SRA (IX+d)
DDCB d3E	SRL (IX+d)
DDCB d46	BIT 0, (IX+d)
DDCB d4E	BIT 1, (IX+d)
DDCB d56	BIT 2, (IX+d)
DDCB d5E	BIT 3, (IX+d)
DDCB d66	BIT 4, (IX+d)
DDCB d6E	BIT 5, (IX+d)
DDCB d76	BIT 6, (IX+d)
DDCB d7E	BIT 7, (IX+d)
DDCB d86	RES 0, (IX+d)
DDCB d8E	RES 1, (IX+d)
DDCB d96	RES 2, (IX+d)
DDCB d9E	RES 3, (IX+d)
DDCB dA6	RES 4, (IX+d)
DDCB dAE	RES 5, (IX+d)
DDCB dB6	RES 6, (IX+d)
DDCB dBE	RES 7, (IX+d)
DDCB dC6	SET 0, (IX+d)
DDCB dCE	SET 1, (IX+d)
DDCB dD6	SET 2, (IX+d)
DDCB dDE	SET 3, (IX+d)
DDCB dE6	SET 4, (IX+d)
DDCB dEE	SET 5, (IX+d)
DDCB dF6	SET 6, (IX+d)
DDCB dFE	SET 7, (IX+d)
ED40	IN B
ED41	OUT B
ED42	SBC HL, BC
ED43 nn	LD (nn), BC
ED44	NEG
ED45	RETN
ED46	IMO
ED47	LD I, A
ED48	IN C
ED49	OUT C
ED4A	ADC HL, BC
ED4B nn	LD BC, (nn)
ED4D	RETI
ED4F	LD R, A

Op-Code	Mnemonic
ED50	IN D
ED51	OUT D
ED52	SBC HL, DE
ED53 nn	LD (nn), DE
ED56	IM1
ED57	LD A, I
ED58	IN E
ED59	OUT E
ED5A	ADC HL, DE
ED5B nn	LD DE, (nn)
ED5E	IM2
ED5F	LD A, R
ED60	IN H
ED61	OUT H
ED62	SBC HL, HL
ED67	RRD
ED68	IN L
ED69	OUT L
ED6A	ADC HL, HL
ED6F	RLD
ED70	INF
ED72	SBC HL, SP
ED73 nn	LD (nn), SP
ED78	IN A
ED79	OUT A
ED7A	ADC HL, SP
ED7B nn	LD SP, (nn)
EDA0	LDI
EDA1	CPI
EDA2	INI
EDA3	OUTI
EDA8	LDD
EDA9	CPD
EDAA	IND
EDAB	OUTD
EDB0	LDIR
EDB1	CPIR
EDB2	INIR
EDB3	OTIR
EDB8	LDDR
EDB9	CPDR
EDBA	INDR
EDBB	OTDR
FDO9	ADO IY, BC
FD19	ADD IY, DE
FD21 nn	LD IY, nn
FD22 nn	LD (nn), IY
FD23	INC IY
FD29	ADD IY, IY
FD2A nn	LD IY, (nn)
FD2B	DEC IY
FD34 d	INC (IY+d)
FD35 d	DEC (IY+d)
FD36 dn	LD (IY+d), n
FD39	ADD IY, SP
FD46 d	LD B, (IY+d)
FD4E d	LD C, (IY+d)
FD56 d	LD D, (IY+d)
FD5E d	LD E, (IY+d)
FD66 d	LD H, (IY+d)
FD6E d	LD L, (IY+d)
FD70 d	LD (IY+d), B
FD71 d	LD (IY+d), C
FD72 d	LD (IY+d), D
FD73 d	LD (IY+d), E
FD74 d	LD (IY+d), H
FD75 d	LD (IY+d), L
FD77 d	LD (IY+d), A
FD7E d	LD (IY+d)
FD86 d	ADD (IY+d)
FD8E d	ADC (IY+d)
FD96 d	SUB (IY+d)
FD9E d	SBC (IY+d)
FDA6 d	AND (IY+d)
FDAE d	XOR (IY+d)
FDB6 d	OR (IY+d)
FDBE d	CP (IY+d)
FDE1	POP IY
FDE3	EX (SP), IY
FDE5	PUSH IY
FDE9	JMP (IY)
FDF9	LD SP, IY

Op-Code	Mnemonic
FDCB d06	RLC (IY+d)
FDCB d0E	RRC (IY+d)
FDCB d16	RL (IY+d)
FDCB d1E	RR (IY+d)
FDCB d26	SLA (IY+d)
FDCB d2E	SRA (IY+d)
FDCB d3E	SRL (IY+d)
FDCB d46	BIT 0, (IY+d)
FDCB d4E	BIT 1, (IY+d)
FDCB d56	BIT 2, (IY+d)
FDCB d5E	BIT 3, (IY+d)
FDCB d66	BIT 4, (IY+d)
FDCB d6E	BIT 5, (IY+d)
FDCB d76	BIT 6, (IY+d)
FDCB d7E	BIT 7, (IY+d)
FDCB d86	RES 0, (IY+d)
FDCB d8E	RES 1, (IY+d)
FDCB d96	RES 2, (IY+d)
FDCB d9E	RES 3, (IY+d)
FDCB dA6	RES 4, (IY+d)
FDCB dAE	RES 5, (IY+d)
FDCB dB6	RES 6, (IY+d)
FDCB dBE	RES 7, (IY+d)
FDCB dC6	SET 0, (IY+d)
FDCB dCE	SET 1, (IY+d)
FDCB dD6	SET 2, (IY+d)
FDCB dDE	SET 3, (IY+d)
FDCB dE6	SET 4, (IY+d)
FDCB dEE	SET 5, (IY+d)
FDCB dF6	SET 6, (IY+d)
FDCB dFE	SET 7, (IY+d)

Befehlsliste - sortiert nach der Mnemonik

8E	ADC A, (HL)
DD8E d	ADC (IX+d)
FD8E d	ADC (IY+d)
8F	ADC A
88	ADC B
89	ADC C
8A	ADC D
8B	ADC E
8C	ADC H
8D	ADC L
CE n	ADC n
ED4A	ADC HL, BC
ED5A	ADC HL, DE
ED6A	ADC HL, HL
ED7A	ADC HL, SP
86	ADD M
DD86 d	ADD (IX+d)
FD86 d	ADD (IY+d)
87	ADD A
80	ADD B
81	ADD C
82	ADD D
83	ADD E
84	ADD H
85	ADD L
C6 n	ADD n
09	ADD HL, BC
19	ADD HL, DE
29	ADD HL, HL
39	ADD HL, SP
DD09	ADD IX, BC
DD19	ADD IX, DE
DD29	ADD IX, IX
DD39	ADD IX, SP
FD09	ADD IY, BC
FD19	ADD IY, DE
FD29	ADD IY, IY
FD39	ADD IY, SP
A6	AND M
DDA6 d	AND (IX+d)
FDA6 d	AND (IY+d)
A7	AND A
A0	AND B
A1	AND C



OP-Code	Mnemonic
A2	AND D
A3	AND E
A4	AND H
A5	AND L
E6 n	AND n
CB46	BIT 0, M
DDCB d46	BIT 0, (IX+d)
FDCB d46	BIT 0, (IY+d)
CB47	BIT 0, A
CB40	BIT 0, B
CB41	BIT 0, C
CB42	BIT 0, D
CB43	BIT 0, E
CB44	BIT 0, H
CB45	BIT 0, L
CB4E	BIT 1, M
DDCB d4E	BIT 1, (IX+d)
FDCB d4E	BIT 1, (IY+d)
CB4F	BIT 1, A
CB48	BIT 1, B
CB49	BIT 1, C
CB4A	BIT 1, D
CB4B	BIT 1, E
CB4C	BIT 1, H
CB4D	BIT 1, L
CB56	BIT 2, M
DDCB d56	BIT 2, (IX+d)
FDCB d56	BIT 2, (IY+d)
CB57	BIT 2, A
CB50	BIT 2, B
CB51	BIT 2, C
CB52	BIT 2, D
CB53	BIT 2, E
CB54	BIT 2, H
CB55	BIT 2, L
CB5E	BIT 3, M
DDCB d5E	BIT 3, (IX+d)
FDCB d5E	BIT 3, (IY+d)
CB5F	BIT 3, A
CB58	BIT 3, B
CB59	BIT 3, C
CB5A	BIT 3, D
CB5B	BIT 3, E
CB5C	BIT 3, H
CB5D	BIT 3, L
CB66	BIT 4, M
DDCB d66	BIT 4, (IX+d)
FDCB d66	BIT 4, (IY+d)
CB67	BIT 4, A
CB60	BIT 4, B
CB61	BIT 4, C
CB62	BIT 4, D
CB63	BIT 4, E
CB64	BIT 4, H
CB65	BIT 4, L
CB6E	BIT 5, M
DDCB d66	BIT 5, (IX+d)
FDCB d6E	BIT 5, (IY+d)
CB6F	BIT 5, A
CB68	BIT 5, B
CB69	BIT 5, C
CB6A	BIT 5, D
CB6B	BIT 5, E
CB6C	BIT 5, H
CB6D	BIT 5, L
CB76	BIT 6, M
DDCB d76	BIT 6, (IX+d)
FDCB d76	BIT 6, (IY+d)
CB77	BIT 6, A
CB70	BIT 6, B
CB71	BIT 6, C
CB72	BIT 6, D
CB73	BIT 6, E
CB74	BIT 6, H
CB75	BIT 6, L
CB7E	BIT 7, M
DDCB d7E	BIT 7, (IX+d)
FDCB d7E	BIT 7, (IY+d)
CB7F	BIT 7, A
CB78	BIT 7, B
CB79	BIT 7, C
CB7A	BIT 7, D
CB7B	BIT 7, E
CB7C	BIT 7, H
CB7D	BIT 7, L
DC nn	CAC nn
FC nn	CAM nn

OP-Code	Mnemonic
D4 nn	CANC nn
CB nn	CALL nn
C4 nn	CANZ nn
F4 nn	CAP nn
EC nn	CAPE nn
E4 nn	CAPO nn
CC nn	CAZ nn
3F	CCF
BE	CMP M
DDBE d	CMP (IX+d)
FDBE d	CMP (IY+d)
BF	CMP A
B8	CMP B
B9	CMP C
BA	CMP D
BB	CMP E
BC	CMP H
BD	CMP L
FE n	CMP n
EDA9	CPD
EDB9	CPDR
EDA1	CPI
EDB1	CPIR
2F	CPL
27	DAA
35	DEC M
DD35 d	DEC (IX+d)
FD35 d	DEC (IY+d)
3D	DEC A
05	DEC B
0B	DEC BC
0D	DEC C
15	DEC D
1B	DEC DE
1D	DEC E
25	DEC H
2B	DEC HL
DD2B	DEC IX
FD2B	DEC IY
2D	DEC L
3B	DEC SP
F3	DI
10 e	DJNZ e
FB	EI
E3	EX (SP), HL
DDE3	EX (SP), IX
FDE3	EX (SP), IY
08	EXAF
EB	EX DE, HL
D9	EXX
76	HALT
ED46	IMO
ED56	IM1
ED5E	IM2
ED78	IN A
DB n	IN n
ED40	IN B
ED48	IN C
ED50	IN D
ED58	IN E
ED60	IN H
ED68	IN L
34	INC M
DD34 d	INC (IX+d)
FD34 d	INC (IY+d)
3C	INC A
04	INC B
03	INC BC
0C	INC C
14	INC D
13	INC DE
1C	INC E
24	INC H
23	INC HL
DD23	INC IX
FD23	INC IY
2C	INC L
33	INC SP
EDAA	IND
EDBA	INDR
ED70	INF
EDA2	INI
EDB2	INIR
E9	JMP M
DDE9	JMP (IX)
FDE9	JMP (IY)
DA nn	JPC nn

OP-Code	Mnemonic
FA nn	JPM nn
D2 nn	JPNC nn
C3 nn	JMP nn
C2 nn	JPNZ nn
F2 nn	JPP nn
EA nn	JPPE nn
E2 nn	JPP0 nn
CA nn	JPZ nn
38 e	JRC e
18 e	JR e
30 e	JRNC e
20 e	JRNZ e
28 e	JRZ e
02	LD (BC), A
12	LD (DE), A
77	LD M, A
70	LD M, B
71	LD M, C
72	LD M, D
73	LD M, E
74	LD M, H
75	LD M, L
36 n	LD M, n
DD77 d	LD (IX+d), A
DD70 d	LD (IX+d), B
DD71 d	LD (IX+d), C
DD72 d	LD (IX+d), D
DD73 d	LD (IX+d), E
DD74 d	LD (IX+d), H
DD75 d	LD (IX+d), L
DD36 dn	LD (IX+d), n
FD77 d	LD (IY+d), A
FD70 d	LD (IY+d), B
FD71 d	LD (IY+d), C
FD72 d	LD (IY+d), D
FD73 d	LD (IY+d), E
FD74 d	LD (IY+d), H
FD75 d	LD (IY+d), L
FD36 dn	LD (IY+d), n
32 nn	LD (nn), A
ED43 nn	LD (nn), BC
ED53 nn	LD (nn), DE
22 nn	LD (nn), HL
DD22 nn	LD (nn), IX
FD22 nn	LD (nn), IY
ED73 nn	LD (nn), SP
0A	LD A, (BC)
1A	LD A, (DE)
7E	LD A, (HL)
DD7E d	LD A, (IX+d)
FD7E d	LD A, (IY+d)
3A nn	LD A, (nn)
7F	LD A, A
78	LD A, B
79	LD A, C
7A	LD A, D
7B	LD A, E
7C	LD A, H
ED57	LD A, I
7D	LD A, L
3E n	LD A, n
ED5F	LD A, R
46	LD B, M
DD46 d	LD B, (IX+d)
FD46 d	LD B, (IY+d)
47	LD B, A
40	LD B, B
41	LD B, C
42	LD B, D
43	LD B, E
44	LD B, H
45	LD B, L
06 n	LD B, n
ED4E nn	LD BC, (nn) → ED 4B
01 nn	LD BC, nn
4E	LD C, M
DD4E d	LD C, (IX+d)
FD4E d	LD C, (IY+d)
4F	LD C, A
48	LD C, B
49	LD C, C
4A	LD C, D
4B	LD C, E
4C	LD C, H
4D	LD C, L
0E n	LD C, n
56	LD D, M
DD56 d	LD D, (IX+d)



OP-Code	Mnemonic
FD56 d	LD D, (IX+d)
57	LD D, A
50	LD D, B
51	LD D, C
52	LD D, D
53	LD D, E
54	LD D, H
55	LD D, L
16 n	LD D, n
ED5B nn	LD DE, (nn)
11 nn	LD DE, nn
5E	LD E, M
DD5E d	LD E, (IX+d)
FD5E d	LD E, (IY+d)
5F	LD E, A
58	LD E, B
59	LD E, C
5A	LD E, D
5B	LD E, E
5C	LD E, H
5D	LD E, L
1E n	LD E, n
66	LD H, M
DD66 d	LD H, (IX+d)
FD66 d	LD H, (IY+d)
67	LD H, A
60	LD H, B
61	LD H, C
62	LD H, D
63	LD H, E
64	LD H, H
65	LD H, L
26 n	LD H, n
2A nn	LD HL, (nn)
21 nn	LD HL, nn
ED47	LD I, A
DD2A nn	LD IX, (nn)
DD21 nn	LD IX, nn
FD2A nn	LD IY, (nn)
FD21 nn	LD IY, nn
6E	LD L, M
DD6E d	LD L, (IX+d)
FD6E d	LD L, (IY+d)
6F	LD L, A
68	LD L, B
69	LD L, C
6A	LD L, D
6B	LD L, E
6C	LD L, H
6D	LD L, L
2E n	LD L, n
ED4F	LD R, A
ED7B nn	LD SP, (nn)
F9	LD SP, HL
DDF9	LD SP, IX
FDF9	LD SP, IY
31 nn	LD SP, nn
EDAF 8	LDD
EDBS	LDDR
EDAO	LDI
EDBO	LDIR
ED44	NEG
00	NOP
B6	OR M
DDB6 d	OR (IX+d)
FDB6 d	OR (IY+d)
B7	OR A
B0	OR B
B1	OR C
B2	OR D
B3	OR E
B4	OR H
B5	OR L
F6 n	OR n
EDBB	OTDR
EDB3	OTIR
ED79	OUT A
ED41	OUT B
ED49	OUT C
ED51	OUT D
ED59	OUT E
ED61	OUT H
ED69	OUT L
B3 n	OUT n
EDAB	OUTD
EDA3	OUTI
F1	POP AF
G1	POP BC

OP-Code	Mnemonic
D1	POP DE
E1	POP HL
DDE1	POP IX
FDE1	POP IY
F5	PUSH AF
C5	PUSH BC
D5	PUSH DE
E5	PUSH HL
DDE5	PUSH IX
FDE5	PUSH IY
CB86	RES 0, M
DDCB d86	RES 0, (IX+d)
FDCB d86	RES 0, (IY+d)
CB87	RES 0, A
CB80	RES 0, B
CB81	RES 0, C
CB82	RES 0, D
CB83	RES 0, E
CB84	RES 0, H
CB85	RES 0, L
CB8E	RES 1, M
DDCB d8E	RES 1, (IX+d)
FDCB d8E	RES 1, (IY+d)
CB8F	RES 1, A
CB88	RES 1, B
CB89	RES 1, C
CB8A	RES 1, D
CB8B	RES 1, E
CB8C	RES 1, H
CB8D	RES 1, L
CB96	RES 2, M
DDCB d96	RES 2, (IX+d)
FDCB d96	RES 2, (IY+d)
CB97	RES 2, A
CB90	RES 2, B
CB91	RES 2, C
CB92	RES 2, D
CB93	RES 2, E
CB94	RES 2, H
CB95	RES 2, L
CB9E	RES 3, M
DDCB d9E	RES 3, (IX+d)
FDCB d9E	RES 3, (IY+d)
CB9F	RES 3, A
CB98	RES 3, B
CB99	RES 3, C
CB9A	RES 3, D
CB9B	RES 3, E
CB9C	RES 3, H
CB9D	RES 3, L
CBA6	RES 4, M
DDCB dA6	RES 4, (IX+d)
FDCB dA6	RES 4, (IY+d)
CBA7	RES 4, A
CBA0	RES 4, B
CBA1	RES 4, C
CBA2	RES 4, D
CBA3	RES 4, E
CBA4	RES 4, H
CBA5	RES 4, L
CBAE	RES 5, M
DDCB dAE	RES 5, (IX+d)
FDCB dAE	RES 5, (IY+d)
CBAF	RES 5, A
CBAB	RES 5, B
CBAB	RES 5, C
CBAA	RES 5, D
CBAB	RES 5, E
CBAC	RES 5, H
CBAD	RES 5, L
CB86	RES 6, M
DDCB dB6	RES 6, (IX+d)
FDCB dB6	RES 6, (IY+d)
CB87	RES 6, A
CB80	RES 6, B
CB81	RES 6, C
CB82	RES 6, D
CB83	RES 6, E
CB84	RES 6, H
CB85	RES 6, L
CB8E	RES 7, M
DDCB dBE	RES 7, (IX+d)
FDCB dBE	RES 7, (IY+d)
CB8F	RES 7, A
CB88	RES 7, B
CB89	RES 7, C
CB8A	RES 7, D

OP-Code	Mnemonic
CB8B	RES 7, E
CB8C	RES 7, H
CB8D	RES 7, L
C9	RET
D8	RC
F8	RM
DO	RNC
CO	RNZ
FO	RP
E8	RPE
EO	RPO
C8	RZ
ED4D	RETI
ED45	RETN
CB16	RL M
DDCB d16	RL (IX+d)
FDCB d16	RL (IY+d)
CB17	RL A
CB10	RL B
CB11	RL C
CB12	RL D
CB13	RL E
CB14	RL H
CB15	RL L
17	RLA
CBO6	RLC M
DDCB d06	RLC (IX+d)
FDCB d06	RLC (IY+d)
CBO7	RLC A
CBO0	RLC B
CBO1	RLC C
CBO2	RLC D
CBO3	RLC E
CBO4	RLC H
CBO5	RLC L
07	RLCA
ED6F	RLD
CB1E	RR M
DDCB d1E	RR (IX+d)
FDCB d1E	RR (IY+d)
CB1F	RR A
CB18	RR B
CB19	RR C
CB1A	RR D
CB1B	RR E
CB1C	RR H
CB1D	RR L
1F	RRA
CBOE	RRC M
DDCB d0E	RRC (IX+d)
FDCB d0E	RRC (IY+d)
CBOF	RRC A
CB08	RRC B
CB09	RRC C
CBOA	RRC D
CBOB	RRC E
CBOC	RRC H
CB0D	RRC L
OF	RRCA
ED67	RRD
C7	RST 0
D7	RST 10H
DF	RST 18H
E7	RST 20H
EF	RST 28H
F7	RST 30H
FF	RST 38H
CF	RST 8
9E	SBC M
DD9E d	SBC (IX+d)
FD9E d	SBC (IY+d)
9F	SBC A
98	SBC B
99	SBC C
9A	SBC D
9B	SBC E
9C	SBC H
9D	SBC L
DE n	SBC n
ED42	SBC HL, BC
ED52	SBC HL, DE
ED62	SBC HL, HL
ED72	SBC HL, SP
37	SCF
CBC6	SET 0, M
DDCB dC6	SET 0, (IX+d)
FDCB dC6	SET 0, (IY+d)



OP-Code	Mnemonic
CBC7	SET 0,A
CBC0	SET 0,B
CBC1	SET 0,C
CBC2	SET 0,D
CBC3	SET 0,E
CBC4	SET 0,H
CBC5	SET 0,L
CBCE	SET 1,M
DDCB dCE	SET 1,(IX+d)
FDCB dCE	SET 1,(IY+d)
CBCF	SET 1,A
CBC8	SET 1,B
CBC9	SET 1,C
CBCA	SET 1,D
CBCB	SET 1,E
CBCC	SET 1,H
CBCD	SET 1,L
CBD6	SET 2,M
DDCB dD6	SET 2,(IX+d)
FDCB dD6	SET 2,(IY+d)
CBD7	SET 2,A
CBDO	SET 2,B
CBD1	SET 2,C
CBD2	SET 2,D
CBD3	SET 2,E
CBD4	SET 2,H
CBD5	SET 2,L
CBD8	SET 3,B
CBDE	SET 3,M
DDCB dDE	SET 3,(IX+d)
FDCB dDE	SET 3,(IY+d)
CBDF	SET 3,A
CBD9	SET 3,C
CBDA	SET 3,D
CBDB	SET 3,E
CBDC	SET 3,H
CBDD	SET 3,L
CBE6	SET 4,M
DDCB dE6	SET 4,(IX+d)
FDCB dE6	SET 4,(IY+d)
CBE7	SET 4,A
CBE0	SET 4,B
CBE1	SET 4,C

OP-Code	Mnemonic
CBE2	SET 4,D
CBE3	SET 4,E
CBE4	SET 4,H
CBE5	SET 4,L
CBEE	SET 5,M
DDCB dEE	SET 5,(IX+d)
FDCB dEE	SET 5,(IY+d)
CBEF	SET 5,A
CBE8	SET 5,B
CBE9	SET 5,C
CBEA	SET 5,D
CBEB	SET 5,E
CBEC	SET 5,H
CBED	SET 5,L
CBF6	SET 6,M
DDCB dF6	SET 6,(IX+d)
FDCB dF6	SET 6,(IY+d)
CBF7	SET 6,A
CBF0	SET 6,B
CBF1	SET 6,C
CBF2	SET 6,D
CBF3	SET 6,E
CBF4	SET 6,H
CBF5	SET 6,L
CBFE	SET 7,M
DDCB dFE	SET 7,(IX+d)
FDCB dFE	SET 7,(IY+d)
CBFF	SET 7,A
CBF8	SET 7,B
CBF9	SET 7,C
CBFA	SET 7,D
CBFB	SET 7,E
CBFC	SET 7,H
CBFD	SET 7,L
CB26	SLA M
DDCB d26	SLA (IX+d)
FDCB d26	SLA (IY+d)
CB27	SLA A
CB20	SLA B
CB21	SLA C
CB22	SLA D
CB23	SLA E
CB24	SLA H

OP-Code	Mnemonic
CB25	SLA L
CB2E	SRA M
DDCB d2E	SRA (IX+d)
FDCB d2E	SRA (IY+d)
CB2F	SRA A
CB28	SRA B
CB29	SRA C
CB2A	SRA D
CB2B	SRA E
CB2C	SRA H
CB2D	SRA L
CB3E	SRL M
DDCB d3E	SRL (IX+d)
FDCB d3E	SRL (IY+d)
CB3F	SRL A
CB38	SRL B
CB39	SRL C
CB3A	SRL D
CB3B	SRL E
CB3C	SRL H
CB3D	SRL L
96	SUB M
DD 96d	SUB (IX+d)
FD 96 d	SUB (IY+d)
97	SUB A
90	SUB B
91	SUB C
92	SUB D
93	SUB E
94	SUB H
95	SUB L
D6 n	SUB n
AE	XOR M
DDA Ed	XOR (IX+d)
FDA Ed	XOR (IY+d)
AF	XOR A
A8	XOR B
A9	XOR C
AA	XOR D
AB	XOR E
AC	XOR H
AD	XOR L
EE n	XOR n



## Programmierung der Peripheriebausteine

### CTC

DI EI	Zeitgeber Zähler	16 256 <sup>*</sup>	neg. Pos.	*	- folgt	nein ja	1	0 1
INT	MODE	VOR- TEILER	TRIGGER- FLANKE	TRIGGER- ZEITPKT.	ZEIT- KONSTANTE	RESET		

\* nur bei Zeitgeber

### PIO

MODE -  
AUSWAHL

M1	M0	X	X	1	1	1	1
----	----	---	---	---	---	---	---

MODE 0	0	0	Byte - Ausgabe
1	0	1	Byte - Eingabe
2	1	0	Byte - Ein - /Ausgabe
3	1	1	Byte - Ein - /Ausgabe

E/A -  
Festlegung  
(Nur im MODE 3)

E/ A	E/ A	E/ A	E/ A	E/ A	E/ A	E/ A	E/ A
---------	---------	---------	---------	---------	---------	---------	---------

0 ≙ Ausgabe

1 ≙ Eingabe

INT -  
Steuerung  
IM MODE 3

DI EI	OR AND	LOW HIGH	- MASKE	0	1	1	1	0 1
----------	-----------	-------------	------------	---	---	---	---	--------

INT-MASKE  
IM MODE 3

G/ NG	G/ NG	G/ NG	G/ NG	G/ NG	G/ NG	G/ NG	G/ NG
----------	----------	----------	----------	----------	----------	----------	----------

0 ≙ generierbar

1 ≙ nicht generierbar

INT -  
Steuerung  
MODE 0-2

DI EI	X	X	X	S	S	1	1	0 1
----------	---	---	---	---	---	---	---	--------



Tabelle der Hexadecimalwerte

6	5	4	3	2	1
HEX = DEC	HEX = DEC	HEX = DEC	HEX = DEC	HEX = DEC	HEX = DEC
0 0	0 0	0 0	0 0	0 0	0 0
1 1,048,576	1 65,536	1 4,096	1 256	1 16	1 1
2 2,097,152	2 131,072	2 8,192	2 512	2 32	2 2
3 3,145,728	3 196,608	3 12,288	3 768	3 48	3 3
4 4,194,304	4 262,144	4 16,384	4 1,024	4 64	4 4
5 5,242,880	5 327,680	5 20,480	5 1,280	5 80	5 5
6 6,291,456	6 393,216	6 24,576	6 1,536	6 96	6 6
7 7,340,032	7 458,752	7 28,672	7 1,792	7 112	7 7
8 8,388,608	8 524,288	8 32,768	8 2,048	8 128	8 8
9 9,437,184	9 589,824	9 36,864	9 2,304	9 144	9 9
A 10,485,760	A 655,360	A 40,960	A 2,560	A 160	A 10
B 11,534,336	B 720,896	B 45,056	B 2,816	B 176	B 11
C 12,582,912	C 786,432	C 49,152	C 3,072	C 192	C 12
D 13,631,488	D 851,968	D 53,248	D 3,328	D 208	D 13
E 14,680,064	E 917,504	E 57,344	E 3,584	E 224	E 14
F 15,728,640	F 983,040	F 61,440	F 3,840	F 240	F 15
0123	4567	0123	4567	0123	4567
BYTE		BYTE		BYTE	

Potenzen von 2

$2^n$	n
256	8
512	9
1 024	10
2 048	11
4 096	12
8 192	13
16 384	14
32 768	15
65 536	16
131 072	17
262 144	18
524 288	19
1 048 576	20
2 097 152	21
4 194 304	22
8 388 608	23
16 777 216	24

Potenzen von 16

$16^n$	n
1	0
16	1
256	2
4 096	3
65 536	4
1 048 576	5
16 777 216	6
268 435 456	7
4 294 967 296	8
68 719 476 736	9
1 099 511 627 776	10
17 592 186 044 416	11
281 474 976 710 656	12
4 503 599 627 370 496	13
72 057 594 037 927 936	14
1 152 921 504 606 846 976	15