

**MIKROPROZESSOREN UND MIKRORECHNER Teil 2**

J. Zaremba, KdF Berlin

C(arry) = 1<sup>1)</sup> genau dann, wenn A überläuft,  
d. h. wenn das Ergebnis > i. 111 111 ist

Z(ero) = 1 genau dann, wenn <A> = 00 000 000

S(ign) = 1 genau dann, wenn A<sub>7</sub> = 1<sup>2)</sup>

P(arity) = 1 genau dann, wenn die Anzahl der A<sub>i</sub> = 1<sup>2)</sup> gerade ist.

Konkret ist der Befehlsaufbau gemäß Tafel 5 festgelegt. Insgesamt können beim U 808 D 48 verschiedene Befehle gegeben werden. Sie sind in der Tafel 6 übersichtlich zusammengestellt. Der eigentliche Mikroprozessorbefehl, d. h. das speicherfähige Binärschema, ist in Spalte 4 angegeben. Da diese Dualzahlen schwer einprägsam sind, haben sich leicht zu merkende mnemonische Notationen eingebürgert. Die gebräuchlichen Versionen enthalten die Spalten 2 und 3. Es versteht sich, daß der Gebrauch der Mnemonics lediglich als Hilfe für den Programmierer aufgefaßt werden kann. Vor der Speicherung des Programms müssen diese Zeichen durch die entsprechenden Binärschemata substituiert werden. Auf der rechten Seite der Tafel 6 ist durch das Zeichen X eine Flag-Beeinflussung angezeigt. In allen Fällen bleibt der Stand der Flags C, Z, S, P unverändert oder er wird durch den konkret angegebenen Wert bestimmt. Im allgemeinen richten sich die Flags nach dem Akkumulatorinhalt, beim U 808 D also nach dem Register A.

In mehreren Fällen muß die Codierung eines Befehls konkretisiert werden. So ist z. B. beim Befehl Nr. 1 für DDD und SSS die duale Adresse der an der Ladeoperation beteiligten Register einzusetzen. Diese Festlegungen und weitere zum Verständnis von Tafel 6 wichtige Definitionen sind in Tafel 7 zusammengestellt.

<sup>1)</sup> Wenn eine Verwechslung des C-Flags mit dem Register C befürchtet werden muß, wird das Carry-Flag mit C<sub>y</sub> anstelle von C bezeichnet

<sup>2)</sup> s. Tafel 7

Tafel 6. Befehlsliste des Mikroprozessors U 808 D

N	Befehl			Zyklen	Anzahl Zustände	Erklärung	Flags			
	Mnemonic Z	Mnemonic B	Codierung				C	Z	S	P
<b>Lade-Befehle</b>										
1	MOV r <sub>1</sub> , r <sub>2</sub>	Lr <sub>1</sub> r <sub>2</sub>	11 DDD SSS	PCI	5	<r <sub>1</sub> > := <r <sub>2</sub> >				
2	MOV r, M	LrM	11 DDD 111	PCI, PCR	8	<r> := <M>				
3	MVI r, I	LrI	00 DDD 110	PCI, PCR	8	<r> := I				
4	MOV M, r	LMr	11 111 SSS	PCI, PCW	7	<M> := <r>				
5	MVI M, I	LMI	00 111 110	PCI, PCR, PCW	9	<M> := I				
<b>Inkrement-/Dekrement-Befehle</b>										
6	INCr	INr	00 DDD 000	PCI	5	<r> := <r> + 1; r ≠ A		X	X	X
7	DCRr	DCr	00 DDD 001	PCI	5	<r> := <r> - 1; r ≠ A		X	X	X
<b>Verschiebe-Befehle</b>										
8	RLO	RLO	00 000 010	PCI	5	A <sub>j+1</sub> := A <sub>j</sub> ; A <sub>0</sub> := A <sub>7</sub> ; <C <sub>y</sub> > := A <sub>1</sub>				X
9	RRC	RRC	00 001 010	PCI	5	A <sub>j</sub> := A <sub>j+1</sub> ; A <sub>7</sub> := A <sub>0</sub> ; <C <sub>y</sub> > := A <sub>0</sub>				X
10	RAL	RAL	00 010 010	PCI	5	A <sub>j+1</sub> := A <sub>j</sub> ; A <sub>0</sub> := <C <sub>y</sub> > <C <sub>y</sub> > := A <sub>7</sub>				X
11	RAR	RAR	00 011 010	PCI	5	A <sub>j</sub> := A <sub>j+1</sub> ; A <sub>7</sub> := <C <sub>y</sub> > <C <sub>y</sub> > := A <sub>0</sub>				X
<b>Additions-Befehle</b>										
12	ADD r	ADr	10 000 SSS	PCI	5	<A> := <A> + <r>		X	X	X
13	ADD M	ADM	10 000 111	PCI, PCR	8	<A> := <A> + <M>		X	X	X
14	ADI I	ADI	00 000 100	PCI, PCR	8	<A> := <A> + I		X	X	X
15	ADC r	ACr	10 001 SSS	PCI	5	<A> := <A> + <r> + <C <sub>y</sub> >		X	X	X
16	ADC M	ACM	10 001 111	PCI, PCR	8	<A> := <A> + <M> + <C <sub>y</sub> >		X	X	X
17	ACI I	ACI	00 001 100	PCI, PCR, PCW	8	<A> := <A> + I + <C <sub>y</sub> >		X	X	X
<b>Subtraktions-Befehle</b>										
18	SUB r	SUr	10 010 SSS	PCI	5	<A> := <A> - <r>		X	X	X
19	SUB M	SUM	10 010 111	PCI, PCR	8	<A> := <A> - <M>		X	X	X
20	SUI I	SUI	00 010 100	PCI, PCR	8	<A> := <A> - I		X	X	X
21	SBB r	SBr	10 011 SSS	PCI	5	<A> := <A> - <r> - <C <sub>y</sub> >		X	X	X
22	SBB M	SBM	10 011 111	PCI, PCR	8	<A> := <A> - <M> - <C <sub>y</sub> >		X	X	X
23	SBI I	SBI	00 011 100	PCI, PCR	8	<A> := <A> - I - <C <sub>y</sub> >		X	X	X
<b>Logische Operations-Befehle</b>										
24	ANA r	NDr	10 100 SSS	PCI	5	<A> := <A> ∧ <r>		0	X	X
25	ANA M	NDM	10 100 111	PCI, PCR	8	<A> := <A> ∧ <M>		0	X	X
26	ANI I	NDI	00 100 100	PCI, PCR	8	<A> := <A> ∧ I		0	X	X
27	ORA r	ORr	10 110 SSS	PCI	5	<A> := <A> ∨ <r>		0	X	X
28	ORA M	ORM	10 110 111	PCI, PCR	8	<A> := <A> ∨ <M>		0	X	X
29	ORI I	ORI	00 110 100	PCI, PCR	8	<A> := <A> ∨ I		0	X	X
30	XRA r	XRr	10 101 SSS	PCI	5	<A> := <A> ∨ <r>		0	X	X
31	XRA M	XRM	10 101 111	PCI, PCR	8	<A> := <A> ∨ <M>		0	X	X
32	XRI I	XRI	00 101 100	PCI, PCR	8	<A> := <A> ∨ I		0	X	X
<b>Vergleiche-Befehle</b>										
33	CMP r	CTr	10 111 SSS	PCI	5	<A> > <r>; <A> unverändert	<A> > ..	0	0	
34	CMP M	CTM	10 111 111	PCI, PCR	8	<A> > <M>; <A> unverändert	<A> = ...	0	1	

Tafel 6. (Fortsetzung)

Nr.	Befehl			Zyklen	Anzahl Zustände	Erklärung	Flags			
	Mnemonic A	Mnemonic B	Codierung				C	Z	S	P
Sprung-Befehle										
36	JMP I <sub>1</sub> I <sub>2</sub>	JMP I <sub>1</sub> I <sub>2</sub>	01 XXX 100 BB BBB BBB XX BBB BBB	PCI, PCR, PCR	11	<BZ> := I <sub>2</sub> , I <sub>1</sub>				
37	JFC I <sub>1</sub> I <sub>2</sub>	JFC I <sub>1</sub> I <sub>2</sub>	01 000 000 BB BBB BBB XX BBB BBB	PCI, PCR, PCR	11 9	<c> = 0: <BZ> := I <sub>2</sub> , I <sub>1</sub> <c> = 1: <BZ> := <BZ> + 3				
38	JNC JNZ JF JFO	JFC JFZ JFS JFP	J1c I <sub>1</sub> I <sub>2</sub>	01 100 000 BB BBB BBB XX BBB BBB	PCI, PCR, PCR	11 9	<c> = 1: <BZ> := I <sub>2</sub> , I <sub>1</sub> <c> = 0: <BZ> := <BZ> + 3			
	JC JZ JM JFE	JTC JTZ JTB JTP								
Sprung-Befehle mit Rückkehrabsicht										
39	GALL I <sub>1</sub> I <sub>2</sub>	GAL I <sub>1</sub> I <sub>2</sub>	01 XXX 110 BB BBB BBB XX BBB BBB	PCI, PCR, PCR	11	<STACK> := <BZ> + 3; <STP> := <STP> + 1; <BZ> := I <sub>2</sub> , I <sub>1</sub>				
40	CFc I <sub>1</sub> I <sub>2</sub>	CFc I <sub>1</sub> I <sub>2</sub>	01 000 010 BB BBB BBB XX BBB BBB	PCI, PCR, PCR	11 9	<c> = 0: <STACK> := <BZ> + 3; <STP> := <STP> + 1; <BZ> := I <sub>2</sub> , I <sub>1</sub> <c> = -1: <BZ> := <BZ> + 3				
41	CFc I <sub>1</sub> I <sub>2</sub>	CFc I <sub>1</sub> I <sub>2</sub>	01 100 010 BB BBB BBB XX BBB BBB	PCI, PCR, PCR	11 9	<c> = 1: <STACK> := <BZ> + 3; <STP> := <STP> + 1; <BZ> := I <sub>2</sub> , I <sub>1</sub> <c> = 0: <BZ> := <BZ> + 3				
	CG CZ CM CFE	CTC CTZ CTS CTP								
Rückkehrsprung-Befehle										
42	RET	RET	00 XXX 111	PCI	5	<STP> := <STP> - 1; <BZ> := <STACK>				
43	RNC RNZ RP RPO	RFC RFZ RFS RFP	00 000 011	PCI	5 3	<c> = 0: <STP> := <STP> - 1; <BZ> := <STACK> <c> = 1: <BZ> := <BZ> + 1				
44	RC RZ RM RPE	RTC RTZ RTS RTP	00 100 011	PCI	5 3	<c> = 1: <STP> := <STP> - 1; <BZ> := <STACK> <c> = 0: <BZ> := <BZ> + 1				
Restart-Befehl										
45	RST	RST	00 AAA 101	PCI	5	<STACK> := <BZ> + 1; <BZ> := 000 000 00 AAA 000				
E/A-Befehle										
46	IN	INP	01 00M MM1	PCI, PCC	8	<A> := <Eingabebitor MMM>				
47	OUT	OUT	01 RRM MM1	PCI, PCC	6	<Ausgabebitor RRRMM> := <A>; RR ≠ 00				
Halt-Befehl										
48	HLT	HLT	00 000 00X	PCI	4					
	HLT	HLT	11 111 111	PCI	4					
Leerbefehl										
	NOP		11 000 000	PCI	5	Eigentlich: <A> := <A>				

Einige Beispiele sollen den Gebrauch der Tafeln 6 und 7 verdeutlichen:

Operation: <B> := <D>

Befehl: LBD  $\Delta$  11 001 011

Operation: <L> := <L> + 1

Befehl: INL  $\Delta$  00 110 000

Operation: <A> := <A> + <E>

Befehl: ADE  $\Delta$  10 000 100

Operation: <A> := <A>  $\wedge$  11 110 090

Befehl: NDI  $\Delta$  00 100 100

I 11 110 000

Operation: <A>  $\geq$  <C>?

Befehl: CPC  $\Delta$  10 111 010

Operation:  $\rightarrow$  102, wenn <Z> = 1

Befehl: JTZ  $\Delta$  01 101 000

I<sub>1</sub> 01 100 110

I<sub>2</sub> 00 000 000

Für die Befehle, die sich auf einen adressierten Speicherplatz beziehen, müssen zuvor die Anweisungen

<H> := Adresse<sub>H</sub> und <L> := Adresse<sub>L</sub>

ausgeführt werden. Die Register H und L sind daher fast immer für die Aufstellung der Speicheradressen zu reservieren und entfallen in der Regel als Zwischenspeicher.

Tafel 7. Bezeichnungen und Festlegungen, Teil 2

A <sub>j</sub>	<A> = A <sub>7</sub> A <sub>6</sub> A <sub>5</sub> A <sub>4</sub> A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>		
AAA	3stellige Binärzeichenfolge		
BBBBBBBB	8stellige Binärzeichenfolge		
c	eines der Flags C, Z, S, P		
CC	Flagadresse	Adresse	Name
		00	C
		01	Z
		10	S
		11	P
DDD	Zielregisteradresse	000	A
		001	B
		010	C
		011	D
		100	E
		101	H
		110	L
I, I <sub>1</sub> , I <sub>2</sub>	Direktoperanden (1 Byte)		
M	adressierter Speicherplatz		
MMM	8stellige Binärzeichenfolge für E/A-Toradresse		
F, F <sub>1</sub> , F <sub>2</sub>	eines der Register A, B, ..., L		
RR	2stellige Binärzeichenfolge für E/A-Toradresse		
SSS	Quellregisteradresse, wie DDD		
X	Beliebiges Bit		
$\vee, \wedge, \nabla$	bitweise logische Verknüpfungen		
	Wertetabelle: a b	a $\wedge$ b a $\vee$ b a $\nabla$ b	
	0 0	0 0 0	
	0 1	0 1 1	