

Mnemonic	Funktion	Flags	effektive Adresse/Zahl der Taktzyklen															
			XNZVC	Dn	An	(An)	(An)+	-(An)	d(An)	d(An,Rx)	xx.W	xxxx.L	d(PC)	d(PC,Rx)	#xxxx	CCR/SR		
R																		
ROR																		
	Quelle I (Ziel) -> Ziel	-**00	B	04	-	12	12	14	16	18	16	20	-	-	-	-	-	-
			W	04	-	12	12	14	16	18	16	20	-	-	-	-	-	-
			L	08	-	20	20	22	24	26	24	28	-	-	-	-	-	-
RORI #xxxx,ea		-**00	B	08	-	16	16	18	20	22	20	24	-	-	-	-	-	20*
			W	08	-	16	16	18	20	22	20	24	-	-	-	-	-	20
			L	16	-	28	28	30	32	34	32	36	-	-	-	-	-	-
RXG ea,Dy	Quelle <-> Ziel	-----	B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			W	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			L	06	06	-	-	-	-	-	-	-	-	-	-	-	-	-
RXG Ax,Ay		-----	B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			W	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			L	-	06	-	-	-	-	-	-	-	-	-	-	-	-	-
RXT Dx		-**00	B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			W	04	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			L	04	-	-	-	-	-	-	-	-	-	-	-	-	-	-
I																		
ILLEGAL	PC -> -(SSP);SR -> -(SSP);Trap #4	-----		34														
J																		
JMP ea	Ziel->PC	-----	L	-	-	08	-	-	10	14	10	12	10	14	-	-	-	-
JSR ea	PC -> -(SP);Ziel->PC	-----	L	-	-	16	-	-	18	22	18	20	18	22	-	-	-	-
L																		
LEA ea,Ay	Ziel->Ay	-----	B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			W	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			L	-	-	04	-	-	08	12	08	12	08	12	-	-	-	-
LINK Ax,#xx	Ax -> -(SP);SP -> Ax;SP + D -> SP	-----	L															
LSL ea,Dy	(Ziel) leftshift um (Quelle) Bits C/X=MSB; LSB=0 bei #: 1..8	***0*	B	6:2x	-	-	-	-	-	-	-	-	-	-	-	-	6:2x	-
			W	6:2x	-	-	-	-	-	-	-	-	-	-	-	-	6:2x	-
			L	8:2x	-	-	-	-	-	-	-	-	-	-	-	-	8:2x	-
LSL ea	Verschiebung um ein Bit	***0*	B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			W	-	-	12	12	14	16	18	16	20	-	-	-	-	-	-
			L	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
LSR ea,Dy	(Ziel) rightshift um (Quelle) Bits C/X=LSB; MSB=0 bei #: 1..8	***0*	B	6:2x	-	-	-	-	-	-	-	-	-	-	-	-	6:2x	-
			W	6:2x	-	-	-	-	-	-	-	-	-	-	-	-	6:2x	-
			L	8:2x	-	-	-	-	-	-	-	-	-	-	-	-	8:2x	-
LSR ea	Verschiebung um ein Bit	***0*	B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			W	-	-	12	12	14	16	18	16	20	-	-	-	-	-	-
			L	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
M																		
MOVE ea,Dy	Quelle->Ziel	-**00	B	04	-	08	08	10	12	14	12	16	12	14	08	-	-	-
			W	04	04	08	08	10	12	14	12	16	12	14	08	-	-	-
			L	04	04	12	12	14	16	18	16	20	16	18	12	-	-	-
MOVE ea,(Ay)		-**00	B	08	-	12	12	14	16	18	16	20	16	18	12	-	-	-
			W	08	08	12	12	14	16	18	16	20	16	18	12	-	-	-
			L	12	12	20	20	22	24	26	24	28	24	26	20	-	-	-
MOVE ea,(Ay)+		-**00	B	08	-	12	12	14	16	18	16	20	16	18	12	-	-	-
			W	08	08	12	12	14	16	18	16	20	16	18	12	-	-	-
			L	12	12	20	20	22	24	26	24	28	24	26	20	-	-	-
MOVE ea,-(Ay)		-**00	B	08	-	12	12	14	16	18	16	20	16	18	12	-	-	-
			W	08	08	12	12	14	16	18	16	20	16	18	12	-	-	-
			L	12	12	20	20	22	24	26	24	28	24	26	20	-	-	-
MOVE ea,d(Ay)		-**00	B	12	-	16	16	18	20	22	20	24	20	22	16	-	-	-
			W	12	12	16	16	18	20	22	20	24	20	22	16	-	-	-
			L	16	16	24	24	26	28	30	28	32	28	30	24	-	-	-

Mnemonic	Funktion	Flags	effektive Adresse/Zahl der Taktzyklen															
			XNZVC	Dn	An	(An)	(An)+	-(An)	d(An)	d(An,Rx)	xx.W	xxxx.L	d(PC)	d(PC,Rx)	#xxxx	CCR/SR		
R																		
MOVE ea,d(Ay,Ri)		-**00	B	14	-	18	18	20	22	24	22	26	22	24	18	-	-	-
			W	14	14	18	18	20	22	24	22	26	22	24	18	-	-	-
			L	18	18	26	26	28	30	32	30	34	30	32	26	-	-	-
MOVE ea,xx.W		-**00	B	12	-	16	16	18	20	22	20	24	20	22	16	-	-	-
			W	12	12	16	16	18	20	22	20	24	20	22	16	-	-	-
			L	16	16	24	24	26	28	30	28	32	28	30	24	-	-	-
MOVE ea,xxxx.L		-**00	B	16	-	24	24	26	28	30	28	32	28	30	24	-	-	-
			W	16	16	20	20	22	24	26	24	28	24	26	20	-	-	-
			L	20	20	28	28	30	32	34	32	36	32	34	28	-	-	-
MOVE ea,CCR		*****	B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			W	12	-	16	16	18	20	22	20	24	20	22	16	-	-	-
			L	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MOVE ea,SR		*****	B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			W	12*	-	16*	16*	18*	20*	22*	20*	24*	20*	22*	16*	-	-	-
			L	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MOVE SR,ea		-----	B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			W	06	-	12	12	14	16	18	16	20	-	-	-	-	-	-
			L	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MOVE USP,Ay		-----	B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			W	-	-	04*	-	-	-	-	-	-	-	-	-	-	-	-
			L	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MOVE Ax,USP		-----	B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			W	-	-	04*	-	-	-	-	-	-	-	-	-	-	-	-
			L	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MOVEA ea,Ay	16 Bit: Vorzeichenweiterung	-----	B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			W	04	04	08	08	10	12	14	12	16	12	14	08	-	-	-
			L	04	04	12	12	14	16	18	16	20	16	18	12	-	-	-
MOVEM ea,RL	Quelle -> mehrere Register	-----	B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			W	-	-	12+4n	12+4n	-	-	-	-	-	-	-	-	-	-	-
			L	-	-	12+8n	12+8n	-	-	-	-	-	-	-	-	-	-	-
MOVEM RL,ea	mehrere Register -> Quelle	-----	B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			W	-	-	8+5n	8+5n	12+5n	14+5n	12+5n	16+5n	-	-	-	-	-	-	-
			L	-	-	8+10n	8+10n	12+10n	14+10n	12+10n	16+10n	-	-	-	-	-	-	-
MOVEP Dx,d(Ay)	Daten <-> 8bit-Peripherie Übertragung zw. folgenden gerade/ungerade Adr. u. Dtarget.	-----	B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			W	-	-	-	-	-	-	16	-	-	-	-	-	-	-	-
			L	-	-	-	-	-	-	24	-	-	-	-	-	-	-	-
MOVEP d(Ax),Dy	Es werden erst höherwertige Bytes übertragen Ax/y = Ax/y + 2	-----	B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			W	-	-	-	-	-</										

CC	Bedeutung	Test
TR	true	1
F	false	0
HI	high	$!C \& !Z$
LS	low same	$C Z$
CC	carry clear	$!C$
CS	carry set	C
NE	not equal	$!Z$
EQ	equal	Z
VC	overflow clear	$!V$
VS	overflow set	V
PL	plus	$!N$
MI	minus	N
GE	greater equal	$(N \& V) (!N \& !V)$
LT	less	$(N \& !V) (!N \& V)$
GT	greater	$(N \& V \& !Z) (!N \& !V \& !Z)$
LE	less equal	$Z (N \& !V) (!N \& V)$

Status-Register															
System-Byte							User-Byte								
T		S			12	11	10				X	N	Z	V	C
Trace		Supervisor			Interrupt- maske						Extended	Negativ	Zero	Overflow	Carry