Mnemon i c	Size	Address Mode	Dn	An	(An)	(An) +	-(An)	d ₁₆ (An)	d ₈ (An, Xn)	Abs.W (Abs).W	Abs.L (Abs).L	d ₁₆ (PC)	d _s (PC,Xn)	s = immed d = SR/CC	Opcode Bit Pattern	Boolean	Conditio Codes
ADD		D- d-	#	#	#	#	#	# 4	# 4	#	#	#	#	#	5432 1098 7654 3210	al Du Va	X N Z V
DU	L	s= Dn d= d= Dn s=	2	2	2	2	2	4	4	4	6	4	4	6	1101 DDD1 10EE EEEE 1101 DDD0 10ee eeee	$\frac{d + Dn \rightarrow d}{Dn + s \rightarrow d}$	****
DDA	L	d= An s=	2	2	2	2	2	4	4	4	6	4	4	6	1101 AAA1 11ee eeee	An + s →An	
ODI	_	s= Imm d=	6		_	_	_	_			-	-			0000 0110 10EE EEEE	Dn + # → Dn	****
DDQ		s= Imm3 d=	2		2	2	2	2	4	4	6	6			0101 QQQ0 10EE EEEE	d + # → d	****
DDX	L	s= Dn d=	2												1101 RRR1 1000 0rrr	$d + s + X \rightarrow d$	****
ND .	L	s= Dn d=			2	2	2	4	4	4	6				1100 DDD1 10EE EEEE	d <and>Dn → d</and>	- * * 0
		d= Dn s=	2		2	2	2	4	4	4	6	4	4	6	1100 DDD0 10ee eeee	Dn⟨and⟩s → Dn	
NDI ACD	L	s= Imm d=	6												0000 0010 10EE EEEE	d(and)# → d	-**0(
SL, ASR	L	count= Dn d=	2												1110 rrrf 1010 0DDD	× 🕶 🗀 🛣	***0>
		count= #1-8 d=	2												1110 QQQf 1000 0DDD	nghi X	
cc	В	d ₈ =											branch taken branch not taken	2	0110 CCCC PPPP PPPP	if an true them	
	W	d ₁₆ =											branch taken	4		if cc true, then PC + disp → PC	
		10											branch not taken	4			
CHG	В	bit# = Dn d=			2	2	2	4	4	4	6				0000 rrr1 01EE EEEE		*-
		bit# = Imm d=			4	4	4	6							0000 1000 01EE EEEE	$\overline{\text{(bit#)}}$ of $d \rightarrow Z$,	
	L	bit# = Dn d=	2												0000 rrr1 01EE EEEE 0000 1000 01EE EEEE	$\overline{\text{(bit#)}}$ of d \rightarrow (bit#) of d	
LR	В	bit# = Imm d= bit# = Dn d=	-		2	2	2	4	4	4	6				0000 1000 01EE EEEE		*-
		bit# = Imm d=			4	4	4	6							0000 1000 10EE EEEE	(bit#) of d→Z.	
	L	bit# = Dn d=	2												0000 rrr1 10EE EEEE	0→ (bit#)of d	
		bit# = Imm d=	4												0000 1000 10EE EEEE		
A	В	d ₈ =												2	0110 0000 PPPP PPPP	$PC + disp \rightarrow PC$	
ET	W	d ₁₆ = bit# = Dn d=			2	2	4	4	4	4	6			4	0000 rrr1 11EE EEEE		*-
		bit# = Imm d=			4	4	4	6			,				0000 1111 11EE EEEE	$\overline{\text{(bit#)}}$ of d \rightarrow Z,	
	L	bit# = Dn d=	2												0000 rrr1 11EE EEEE	1→ (bit#) of d	
		bit# = Imm d=	4												0000 1000 11EE EEEE		
R	B	d ₈ =							-					2	0110 0001 PPPP PPPP	$PC \rightarrow (SP)$, $PC + disp \rightarrow PC$	
ST	_	d ₁₆ = bit# = Dn d=			2	2	2	4	4	4	6	4	4	4	0000 rrr1 00EE EEEE	(bit#) of d → Z	*-
J.	0	bit# = Imm d=			4	4	4	6	*	-	3		*		0000 1771 00EE EEEE	(SILTY OF U 'Z	
	L	bit# = Dn d=	2												0000 rrr1 00EE EEEE		
		bit# = Imm d=	4												0000 1000 00EE EEEE		
R	B/W	d=	2		2	2	2	4	4	4	6				0100 0010 SSEE EEEE	0 → d	- 0 1 0
		d=	2		2	2	2	4	4	4	6						
P		d= Dn s=	2	2*	2	2	2	4	4	4	6	4	4	4	1011 DDD0 Ssee eeee	Dn-s	-***
PA	W	d= Dn s= d= An s=	2	2	2	2	2	4	4	4	6	4	4	6	1011 AAA0 11ee eeee	An-s	-***
	L	d= An s=	2	2	2	2	2	4	4	4	6	4	4	6	1011 AAA0 11cc cccc	All 3	
PI	B/W	s= Imm d=	4												0000 1100 SSEE EEEE	d-#	- * * *
	L	s= Imm d=	6														
/S	W	d= Dn s=	2		2	2	2	4	4	4	6	4	4	4	1000 DDD1 11ee eeee	$Dn_{32}/S_{16} \rightarrow Dn(R_{16}: Q_{16})$	-***
	L	d= Dn s=	4		4	4	4	6							0100 1100 01ee eeee 0DDD 1000 0000 0DDD	$Dn_{32}/S_{32} \rightarrow Dn(Q_{32})$	
VU	W	d= Dn s=	2		2	2	2	4	4	4	6	4	4	4	1000 DDD0 11ee eeee	$Dn_{32}/S_{16} \rightarrow (R_{16}:Q_{16})$	- * * *
	L	d= Dn s=	4		4	4	4	6							0100 1100 01ee eeee 0DDD 0000 0000 0DDD	$Dn_{32}/S_{32} \rightarrow Dn(Q_{32})$	
IR .	L	s= Dn d=	2		2	2	2	4	4	4	6				1011 rrr1 10EE EEEE	d ⊕ Dn → d	- * * 0
RI	L	s= Imm d=	6												0000 1010 1000 0DDD	d ⊕ #→ d	- * * 0
T	_	d=	2												0100 1000 1000 0DDD	bit 7 → bits 15:8	-**0
	L	d=	2												0100 1000 1100 0DDD	bit 15 → bits 31:16	
LEGAL															0100 1010 1111 1100	$PC \rightarrow -(SSP)$ $SR \rightarrow -(SSP)$	
																Vector offset → -(SSP)	
																(illegal vector) → PC	
Р		d=			2			4	4	4	6	4	4		0100 1110 11EE EEEE	$d \rightarrow PC$	
R		d=			2			4	4	4	6	4	4		0100 1110 10EE EEEE	$PC \rightarrow -(SP), d \rightarrow PC$	
A		d= An s=			2			4	4	4	6	4	4		0100 AAA1 11ee eeee	$s \rightarrow An$ $An \rightarrow -(SP), SP \rightarrow An,$	
NK	W	d ₁₆ = I mm s=		4											0100 1110 0101 0AAA	SP+disp→SP	
L,LSR	L	count=Dn d=	2												11110 rrrf 1010 1DDD	° ₹	***0
		count=#1-8 d=	2												1110 QQQf 1000 1DDD	○ → Let X	
VE	B/W	d=Dn s=	2	2	2	2	2	4	4	4	6	4	4	4	00XX RRRM MMee eeee	s → d	- * * 0
		d=An s=	MOVEA	MOVEA	MOVEA	MOVEA	MOVEA	MOVEA	MOVEA	MOVEA	MOVEA	MOVEA	MOVEA	MOVEA			
		d=(An) s=	2	2	2	2	2	4	4	4	6	4	4	4			
		d=(An)+ s=	2	2	2	2	2	4	4	4	6	4	4	4			
		d=-(An) s= d=d ₁₆ (An) s=	2	2	2	2	2	6	4	4	6	6	4	4			
		d=d ₁₆ (An) s= d=d _s (An, Xn*SF) s=		4	4	4	4	0				o .					
		_ ·	4	4	4	4	4										
		d=Abs.W s=		6	6	6	6										
		d=ADS.W S= d= Abs.L S=	6		2	2	2	4	4	4	6	4	4	6	0010 RRRM MMee eeee	s → d	- * * 0
	L	d= Abs.L s= d=Dn s=	2	2		MOVEA	MOVEA	MOVEA	MOVEA 4	MOVEA	MOVEA	MOVEA	MOVEA	MOVEA			
	L	d= Abs.L s= d=Dn s= d= An s=	2 MOVEA	MOVEA	MOVEA					4	6	4	4	6			
	L	d= Abs.L s= d=Dn s= d= An s= d=(An) s=	2 MOVEA 2	MOVEA 2	2	2	2	4		-		Λ	4	6			
	L	d= Abs.L s= d=Dn s= d= An s= d=(An) s= d= (An)+ s=	2 MOVEA 2 2	MOVEA		2 2 2	2 2	4 4	4 4	4	6	4	4	6			
	L	d= Abs.L s= d=Dn s= d= An s= d=(An) s=	2 MOVEA 2	MOVEA 2 2	2 2	2	2	4	4	4	6						
	L	d= Abs.L s= d=Dn s= d= An s= d=(An) s= d=(An) + s= d=-(An) s= d=d ₁₆ (An) s= d=d ₈ (An, Xn*SF) s=	2 MOVEA 2 2 2 4	2 2 2 2 4 4	2 2 2 4 4	2 2 4 4	2 2 4 4	4	4	4	6	4					
	L	d= Abs.L s= d=Dn s= d= An s= d=(An) s= d= (An) + s= d=-(An) s= d=-(An) s= d=d_{16}(An) s= d=d_{16}(An, Xn*SF) s= d=Abs.W s=	2 MOVEA 2 2 2 2 4 4	2 2 2 2 4 4	2 2 2 4 4 4	2 2 4 4 4	2 2 4 4 4	4	4	4	6	4					
	L	d= Abs.L s= d=Dn s= d= An s= d=(An) s= d= (An) + s= d=-(An) s= d=d ₁ ₀ (An) s= d=d ₃ ₀ (An, Xn*SF) s= d=Abs.L s=	2 MOVEA 2 2 2 4 4 4 4 6	2 2 2 2 4 4	2 2 2 4 4	2 2 4 4	2 2 4 4	4	4	4	6	4		6	0100 0100		
/E CCR	L W W	d= Abs.L s= d=On s= d= An s= d=(An) s= d=(An) s= d=-(An) s= d=-(An) s= d=d _{1,5} (An) s= d=d _{1,5} (An) s= d=d ₃ (An, Xn×SF) s= d=Abs.L s= d=CCR s=	2 MOVEA 2 2 2 4 4 4 4 6	2 2 2 2 4 4	2 2 2 4 4 4	2 2 4 4 4	2 2 4 4 4	4	4	4	6	4			0100 0100 11ee eeee	lower 8 bits of s → CCR	****
	W	d= Abs.L s= d=Dn s= d= An s= d=(An) s= d= (An) + s= d=-(An) s= d=d ₁ ₀ (An) s= d=d ₃ ₀ (An, Xn*SF) s= d=Abs.L s=	2 MOVEA 2 2 2 4 4 4 4 6	2 2 2 2 4 4	2 2 2 4 4 4	2 2 4 4 4	2 2 4 4 4	4	4	4	6	4		6	0100 0100 11ee eeee 0100 0010 1100 0DDD 0011 AAAO 01ee eeee	lower 8 bits of s → CCR $CCR \rightarrow Dn$ s → An	
	W	d= Abs.L s= d=Dn s= d= An s= d=(An) s= d=(An) s= d=(An) s= d=(An) s= d=d ₁₆ (An) s= d=d ₆ (An, Xn×SF) s= d=Abs.L s= d=CCR s= s=CCR d=	2 MOVEA 2 2 2 4 4 4 6 2 2	MOVEA 2 2 2 4 4 6	2 2 2 4 4 4 6	2 2 4 4 4 6	2 2 4 4 4 6	4 4 6	4	4 4	6	6	4	4	0100 0010 1100 0DDD	CCR → Dn	
/EA	W	d= Abs.L s= d=Dn s= d= An s= d=(An) s= d=(An) + s= d=(An) s= d=Abs.W s= d=Abs.L s= d=COR s= s=COR d= d=An s=	2 MOVEA 2 2 2 4 4 4 6 2 2	MOVEA 2 2 2 4 4 4 6 6	2 2 2 4 4 4 6	2 2 4 4 4 6	2 2 4 4 4 6	4 4 6	4 4	4 4	6	4 6	4	4	0100 0010 1100 0DDD 0011 AAA0 01ee eeee	CCR → Dn	
/EA	W	d= Abs.L s= d=On s= d= An s= d=(An) s=(An) s= d=(An) s=(An) s=(A	2 MOVEA 2 2 2 4 4 4 6 2 2	MOVEA 2 2 2 4 4 4 6 6	2 2 2 4 4 4 6	2 2 4 4 4 6	2 2 4 4 4 6	4 4 4 4	4 4	4 4	6	4 6	4	4	0100 0010 1100 0DDD 0011 AAA0 01ee eeee 0010 AAA0 01ee eeee	$\begin{array}{c} CCR \rightarrow Dn \\ s \rightarrow An \\ \\ \hline \\ Xn \dots \rightarrow d \\ s \rightarrow Xn \dots \end{array}$	
/EA /EM	W W L	d= Abs.L s= d=Dn s= d= An s= d=(An) s= d=(An) + s= d=(An) + s= d=(An) s= d=(An) s= d=(An) s= d=(An) s= d=(An) s= d=(An) s= d=Abs.W s= d=Abs.L s= d=CCR s= s=CCR d= d= An s= d= An s= d= An s= s= Xnd= d= Xnd= s= Imm8 d=	2 MOVEA 2 2 2 4 4 6 2 2 2 2 2	MOVEA 2 2 2 4 4 4 6 6	2 2 4 4 4 6	2 2 4 4 4 6	2 2 4 4 4 6	4 4 4 6 6 6	4 4 4	4 4 4	6 6 6	4 4 4	4 4 4	4 4 6	0100 0010 1100 0DDD 0011 AAA0 01ee eeee 0010 AAA0 01ee eeee 0100 1000 11EE EEEE 0100 1100 11ee eeee 0111 DDDO QQQQ QQQQ	$CCR \rightarrow Dn$ $s \rightarrow An$ $Xn \rightarrow d$ $s \rightarrow Xn$ # $\rightarrow Dn$ (sign ext)	
/EA /EM	W W L	d= Abs.L s= d=Dn s= d= An s= d=(An) s= d=(An) s= d=(-(An) s= d=(-(An) s= d=d_{16}(An, Xn*SF) s= d=Abs.W s= d=Abs.L s= d=CCR s= s=CCR d= d= An s= d= An s= s= Xnd= d= Xnd=	2 MOVEA 2 2 2 4 4 6 2 2 2 2 2	MOVEA 2 2 2 4 4 4 6 6	2 2 2 4 4 4 6	2 2 4 4 4 6	2 2 4 4 4 6	4 4 6 4 4 6	4 4	4 4	6	4 6	4	4	0100 0010 1100 0DDD 0011 AAA0 01ee eeee 0010 AAA0 01ee eeee 0100 1000 11EE EEEE 0100 1100 11ee eeee 0111 DDDO QQQQ QQQQ 1100 DDD1 11ee eeee	$\begin{array}{c} CCR \rightarrow Dn \\ s \rightarrow An \\ \\ \hline \\ Xn \dots \rightarrow d \\ s \rightarrow Xn \dots \end{array}$	
/EA /EM	W W L	d= Abs.L s= d=Dn s= d= An s= d=(An) s= d=(An) + s= d=(An) + s= d=(An) s= d=(An) s= d=(An) s= d=(An) s= d=(An) s= d=(An) s= d=Abs.W s= d=Abs.L s= d=CCR s= s=CCR d= d= An s= d= An s= d= An s= s= Xnd= d= Xnd= s= Imm8 d=	2 MOVEA 2 2 2 4 4 6 2 2 2 2 2	MOVEA 2 2 2 4 4 4 6 6	2 2 4 4 4 6	2 2 4 4 4 6	2 2 4 4 4 6	4 4 4 6 6 6	4 4 4	4 4 4	6 6 6	4 4 4	4 4 4	4 4 6	0100 0010 1100 0DDD 0011 AAA0 01ee eeee 0010 AAA0 01ee eeee 1010 1000 11EE EEEE 1010 1100 1100 eeee 0111 DDD0 0QQQ QQQQQ 1100 DDD1 11ee eeee 0100 1100 00ee eeee	$CCR \rightarrow Dn$ $s \rightarrow An$ $Xn \rightarrow d$ $s \rightarrow Xn$ # $\rightarrow Dn$ (sign ext)	
/EA	W W L	d= Abs.L s= d=Dn s= d=An s= d=(An) s= d=An s= d=An s= d=An s= d=An s= s= Imm8 d= d=Dn s=	2 MOVEA 2 2 2 4 4 4 6 2 2 2 2 2	MOVEA 2 2 2 4 4 4 6 6	2 2 2 4 4 4 6	2 2 4 4 4 6 2 2 2	2 2 4 4 4 4 6 6 2 2	4 4 6 4 4 4 6 6	4 4 4	4 4 4	6 6 6	4 4 4	4 4 4	4 4 6	0100 0010 1100 0DDD 0011 AAA0 01ee eeee 0010 AAA0 01ee eeee 0100 1000 11EE EEEE 0100 1100 11ee eeee 0111 DDDO QQQQ QQQQ 1100 DDD1 11ee eeee	$\begin{array}{c} CCR \rightarrow Dn \\ s \rightarrow An \\ \\ \hline \\ Xn \rightarrow d \\ s \rightarrow Xn \\ \# \rightarrow Dn (sign ext) \\ Dn_{16}*s_{16} \rightarrow Dn_{32} \\ \end{array}$	 -**(-**(
/EA	W W L L W L W	d= Abs.L s= d=Dn s= d=An s= d=(An) s= d=(An) + s= d=(An) + s= d=(An) s= d=(An) s= d=(An) s= d=(An) s= d=Abs.W s= d=Abs.L s= d=CCR s= s=CCR d= d= An s= d= An s= d= Xn d= d= Xn d= s= Imm8 d= d= Dn s= d= Dn s=	2 MOVEA 2 2 4 4 4 6 2 2 2 2 2 2	MOVEA 2 2 2 4 4 4 6 6	2 2 2 4 4 4 6 6	2 2 4 4 4 6 2 2 2	2 2 4 4 4 6 5 2 2 4 4	4 4 4 4 4 6 6 6	4 4 4	4 4 4	6 6 6	4 4 4	4 4 4	4 4 6	0100 0010 1100 0DDD 0011 AAA0 01ee eeee 0010 AAA0 01ee eeee 1010 1000 11EE EEEE 0100 1100 11ee eeee 0111 DDDD 0200 0200 0200 1100 DDD 11ee eeee 0100 1100 00ee eeee 0100 1100 0000 00	$\begin{array}{c} \text{CCR} \rightarrow \text{Dn} \\ \text{s} \rightarrow \text{An} \\ \\ \\ \text{Xn} \rightarrow \text{d} \\ \text{s} \rightarrow \text{Xn} \\ \# \rightarrow \text{Dn} \left(\text{sign ext} \right) \\ \text{Dn}_{16} * \text{s}_{16} \rightarrow \text{Dn}_{32} \\ \\ \\ \text{s}_{22} * \text{s}_{32} \rightarrow \text{s}_{32} \\ \\ \text{Dn}_{16} * \text{s}_{16} \rightarrow \text{Dn}_{32} \\ \end{array}$	 -**C
/EM /EQ .S	W L L W L	d= Abs.L s= d=Dn s= d= An s= d=(An) s=(An) s= d=(An) s=(An)	2 MOVEA 2 2 2 4 4 4 4 6 6 2 2 2 2 2 2 2 2	MOVEA 2 2 2 4 4 4 6 6	2 2 2 4 4 4 6	2 2 4 4 4 6 2 2 2 4 2	2 2 4 4 4 6 2 2 4 2	4 4 6 4 4 4 6 6 6 4	4 4 4	4 4 4	6 6 6	4 4 4	4 4 4	4 4 6	0100 0010 1100 0DDD 0011 AAA0 01ee eeee 0010 AAA0 01ee eeee 0110 1000 11EE EEEE 0100 1100 11ee eeee 0111 DDD0 0000 0000 1100 DDD1 11ee eeee 0100 1100 0000 0000 0100 0000 00	$\begin{array}{c} CCR \rightarrow Dn \\ s \rightarrow An \\ \\ \hline \\ Xn \rightarrow d \\ s \rightarrow Xn \\ \# \rightarrow Dn \ (sign \ ext) \\ Dn_{16}*s_{16} \rightarrow Dn_{22} \\ \\ s_{22}*s_{32} \rightarrow s_{32} \end{array}$	 -**(-**(
/EA //EM //EQ .S .LU	W L L W L	d= Abs.L s= d=Dn s= d=An s= d=(An) s= d=An s= d=An s= d=An s= d=An s= d= An s= d= Imm8 d= d=Dn s= d=Dn s= d=Dn s= d=Dn s= d=An s=	2 MOVEA 2 2 2 4 4 4 6 2 2 2 2 2 2 2 2 2 4 4 4 4	MOVEA 2 2 2 4 4 4 6 6	2 2 2 4 4 4 6	2 2 4 4 4 6 2 2 2 4 2	2 2 4 4 4 6 2 2 4 2	4 4 6 4 4 4 6 6 6 4	4 4 4	4 4 4	6 6 6	4 4 4	4 4 4	4 4 6	0100 0010 1100 0DDD 0011 AAA0 01ee eeee 0010 AAA0 01ee eeee 1010 1000 11EE EEEE 1000 1100 1100 eeee 1110 DDD QQQQ QQQQ 11100 DDD1 11ee eeee 1010 1100 00ee eeee 0DDD 1000 1100 DDD 11ee eeee	$\begin{array}{c} \text{CCR} \rightarrow \text{Dn} \\ \text{s} \rightarrow \text{An} \\ \\ \hline \\ Xn \rightarrow \text{d} \\ \text{s} \rightarrow \text{Xn} \\ \# \rightarrow \text{Dn} \left(\text{sign ext} \right) \\ \text{Dn}_{16} * \text{s}_{16} \rightarrow \text{Dn}_{32} \\ \\ \text{s}_{22} * \text{s}_{32} \rightarrow \text{s}_{32} \\ \\ \text{Dn}_{18} * \text{s}_{16} \rightarrow \text{Dn}_{32} \\ \\ \\ \text{Dn}_{32} * \text{s}_{32} \rightarrow \text{Dn}_{32} \\ \end{array}$	 -**C -**C
VEM VEQ LS LU G G G G G G X	W L L W L	d= Abs.L s= d=Dn s= d=An s= d=(An) s= d=(An) + s= d=(An) + s= d=(An) s= d=(An) s= d=(An) s= d=(An) s= d=(An) s= d=Abs.W s= d=Abs.L s= d=CCR s= s=CCR d= d=An s= d=An s= d=An s= d= An s= d= Dn s= d=Dn s=	2 MOVEA 2 2 4 4 4 6 6 2 2 2 2 2 2 2 2 2 2 2 2 2	MOVEA 2 2 2 4 4 4 6 6	2 2 2 4 4 4 6	2 2 4 4 4 6 2 2 2 4 2	2 2 4 4 4 6 2 2 4 2	4 4 6 4 4 4 6 6 6 4	4 4 4	4 4 4	6 6 6	4 4 4	4 4 4	4 4 6	0100 0010 1100 0DDD 0011 AAA0 01ee eeee 0010 AAA0 01ee eeee 1010 1000 11EE EEEE 1010 1100 1100 0000 00	$\begin{array}{c} \text{CCR} \rightarrow \text{Dn} \\ \text{s} \rightarrow \text{An} \\ \\ \\ \text{Xn} \rightarrow \text{d} \\ \text{s} \rightarrow \text{Xn} \\ \\ \# \rightarrow \text{Dn (sign ext)} \\ \\ \text{Dn}_{16} * \text{s}_{16} \rightarrow \text{Dn}_{32} \\ \\ \\ \text{s}_{32} * \text{s}_{32} \rightarrow \text{s}_{32} \\ \\ \text{Dn}_{18} * \text{s}_{16} \rightarrow \text{Dn}_{32} \\ \\ \\ \text{Dn}_{28} * \text{s}_{32} \rightarrow \text{Dn}_{32} \\ \\ \\ \text{O-Dn} \rightarrow \text{Dn} \end{array}$	 -**0 -**0 ****
VEA VEA VEA G G G G G X P P Remonlo	W W L L W L L L L L L L L L L L L L L L	d= Abs.L s= d=Dn s= d= An s= d=(An) s=(An) s= d=(An) s=(An) s=(An	2 MOVEA 2 2 2 4 4 4 4 4 6 6 2 2 2 2 2 2 2 4 4 2 2 2 4 4 2 2 2 2	MOVEA 2 2 2 4 4 4 4 6 6	2 2 2 4 4 4 6 6 2 2 2 2 4 4 4 4 4 4 6 2 2 4 4 4 4	2 2 4 4 4 6 2 2 2 2 4 4	2 2 4 4 4 6 2 2 4 2	4 4 4 4 4 6 6 6 4 6	4 4 4	4 4 4 4 4 Abs.W	6 6 6 6	4 4 4	4 4 4	6 4 4 6 4 4 s = immed	0100 0010 1100 0DDD 0011 AAA0 01ee eeee 0010 AAA0 01ee eeee 1100 1000 11EE EEEE 0100 1100 1100 11ee eeee 01110 DDD0 0000 0000 11100 DDD1 11ee eeee 0100 1100 0000 0000 1100 0000 00	$\begin{array}{c} \text{CCR} \rightarrow \text{Dn} \\ \text{s} \rightarrow \text{An} \\ \\ \\ \text{Xn} \rightarrow \text{d} \\ \text{s} \rightarrow \text{Xn} \\ \\ \# \rightarrow \text{Dn} \text{ (sign ext)} \\ \\ \text{Dn}_{16} * \text{s}_{16} \rightarrow \text{Dn}_{32} \\ \\ \\ \text{s}_{32} * \text{s}_{32} \rightarrow \text{s}_{32} \\ \\ \text{Dn}_{18} * \text{s}_{16} \rightarrow \text{Dn}_{32} \\ \\ \\ \text{Dn}_{28} * \text{s}_{32} \rightarrow \text{Dn}_{32} \\ \\ \\ \text{O-Dn} \rightarrow \text{Dn} \\ \\ \text{O-Dn-X} \rightarrow \text{Dn} \\ \\ \end{array}$	******0 -**0 **** ****
VEA VEM VEQ LS LU GG	W W L L W L L L L L L L L L L L L L L L	d= Abs.L s= d=Dn s= d= An s= d=(An) s=(An) s= d=(An) s=(An) s= d=(An) s=(An)	2 MOVEA 2 2 4 4 4 6 6 2 2 2 2 2 2 2 2 2 2 2 2 2	MOVEA 2 2 2 4 4 4 6 6	2 2 2 4 4 4 6	2 2 4 4 4 6 2 2 2 2 4 4	2 2 4 4 4 6 2 2 4 2	4 4 4 4 4 6 6 6 4 6	4 4 4	4 4 4 4 4 Abs.W	6 6 6	4 4 4	4 4 4	4 4 6	0100 0010 1100 0DDD 0011 AAA0 01ee eeee 0010 AAA0 01ee eeee 1010 1000 11EE EEEE 1000 1100 110e eeee 1110 DDD 0QQQ QQQQ 1100 DDD1 11ee eeee 1010 1000 0000 0000 1100 DDD0 11ee eeee 1010 1100 000e eeee 1010 1100 000 0000 1100 DDD0 11ee eeee 1010 1100 000e eeee 1010 1100 000e 0000 1100 0DD0 1100 0000 1100 0DD0 0000 000	$\begin{array}{c} \text{CCR} \rightarrow \text{Dn} \\ \text{s} \rightarrow \text{An} \\ \\ \\ \text{Xn} \rightarrow \text{d} \\ \text{s} \rightarrow \text{Xn} \\ \# \rightarrow \text{Dn} \left(\text{sign ext} \right) \\ \text{Dn}_{16} * \text{s}_{16} \rightarrow \text{Dn}_{32} \\ \\ \text{S}_{32} * \text{S}_{32} \rightarrow \text{Dn}_{32} \\ \\ \text{Dn}_{16} * \text{s}_{16} \rightarrow \text{Dn}_{32} \\ \\ \text{Dn}_{26} * \text{s}_{32} \rightarrow \text{Dn}_{32} \\ \\ \text{O-Dn} \rightarrow \text{Dn} \\ \\ \text{O-Dn} \rightarrow \text{Dn} \\ \\ \text{none} \\ \end{array}$	 -**0 -**0 ****

A Address Register Number f Direction: R Destination Register V Vector Number * Set according to result of operation

C Test Condition 0 = Right 1 = Left r Source Register XX Move size: - Not affected by operation

D Data Register Number M Destination EA Mode S Size: 01 = Byte 0 Cleared

E Destination Effective Address P Displacement 00 = Byte 01 = Word 11 = Word 1 Set

Source Effective Address Q Quick Immediate Data 10 = Long U Udefined after operation

Mnemonic	Size	Address Mode	Dn	An	(An)	(An) +	-(An)	d ₁₆ (An)	d ₈ (An, Xn)	Abs.W (Abs).W	Abs.L (Abs).L	d ₁₆ (PC)	d _s (PC,Xn)	s = immed d = SR/CC	Opcode Bit Pattern	Boolean	Condition Codes
			#	#	#	#	#	#	#	#	#	#	#	#	5432 1098 7654 3210		XNZVC
NOT	L	d=	2												0100 0110 1000 0DDD	Dn → Dn	-**00
OR	L	s= Dn d=			2	2	2	4	4	4	6				1000 DDD1 10EE EEEE	d <or>Dn → d</or>	-**00
		d= Dn s=	2		2	2	2	4	4	4	6	4	4	6	1000 DDD0 10ee eeee	Dn <or>s → Dn</or>	-**00
ORI	L	s= Imm d=	6												0000 0000 1000 0DDD	Dn <or># → Dn</or>	-**00
PEA	L	d=			2			4	4	4	6	4	4		0100 1000 01ee eeee	address of d →(SP)	
RTE															0100 1110 0111 0011	(SP+2)→ SP (SP+4)→SR SP + 8→ PC Adjust stack according to format	****
RTS			2												0100 1110 0111 0101	(SP)+→ PC	
ST0P		d=												4	0100 1110 0111 0010	#→SR, wait for interrupt	****
SUB	L	s= Dn d=		SUBA	2	2	2	4	4	4	6				1001 DDD1 10EE EEEE	$d - Dn \rightarrow d$	****
		d= Dn s=	2	2	2	2	2	4	4	4	6	4	4	6	1001 DDD0 10ee eeee	Dn-s→Dn	
SUBA	L	d= An s=	2	2	2	2	2	4	4	4	6	4	4	6	1001 AAA1 11ee eeee	An-s→ An	
SUBI	L	s= Imm d=	6	SUBA											0000 0100 1000 0DDD	d-#→ d	****
SUBQ	L	s= Imm3 d=	2	2	2	2	2	4	4	4	6				0101 QQQ1 10EE EEEE	d -# → d	****
SUBX	L	s= Dn d=	2												1001 RRR1 1000 0rrr	$d-s-X \rightarrow d$	****
SWAP	W	d=	2												0100 1000 0100 0DDD	Dn(31:16) ↔ Dn(15:0)	-**00
TAS	В	d=			2	2	2	4	4	4	6				0100 1010 11EE EEEE	test d→cc, 1 → bit 7 of d	-**00
TRAP		vector=#0-15	2												0100 1110 0100 VVVV	$\begin{array}{c} PC \rightarrow -(SSP) \\ SR \rightarrow -(SSP) \\ Format/Offset \rightarrow -(SSP) \\ (VBR+0x80+4*n) \rightarrow PC \end{array}$	
TST	В	d=	2		2	2	2	4	4	4	6	4	4	4	0100 1010 SSEE EEEE	test d→CC	-**00
	W	d=	2	2	2	2	2	4	4	4	4	4	4	4			
	٦	d=	2	2	2	2	2	4	4	4	4	4	4	6			
UNLK				2											0100 1110 0101 1AAA	$An \rightarrow SP$, $(SP)+\rightarrow An$	
Mnemon i c	Size	Address Mode	Dn	An	(An)	(An) +	(An) -	d ₁₆ (An)	d ₈ (An, Xn)	Abs.W (Abs).W	Abs.L (Abs).L	d ₁₆ (PC)	d _s (PC,Xn)	s = immed d = SR/CC	Opcode Bit Pattern	Boo I ean	Condition Codes
			#	#	#	#	#	#	#	#	#	#	#	#	5432 1098 7654 3210		XNZVC

General Notes

Word Only

Number of Bytes in Instruction

s Source (s10 = base 10 operand)

d Destination (d10 = base 10 operand)

< Value is Maximum Number

Complement (invert) 8-Bit Displacement 16-Bit Displacement Immediate Data Immediate Data, 3 Bits Immediate Data, 8 Bits

 A Address Register Number
 f
 Direction:
 R
 R
 Destination Register

 C Test Condition
 0 = Right
 1 = Left
 r
 Sucree Register

 D Data Register Number
 M
 Destination EA Mode
 S
 Size:

 E Destination Effective Address
 P
 Displacement
 0
 0 = Byte
 01 = Word

 e Source Effective Address
 Q
 Quick Immediate Data
 10 = Long
 10 = Long

Mnemonic

Н

CC(HS)

CS(LO)

NE EQ

MI

GE

Condition Code Notation Set according to result of operation Not affected by operation V Vector Number *
XX Move size: -

01 = Byte 11 = Word

0 Cleared 1 Set U Undefined after operation

Conditional Tests

Encoding

0001

0010

0011

0100

0101

0110

0111

1001 1010

1011

1100

1101 1110

1111

Test

<u>C</u>⋅Z

Z

N

 $\frac{\mathsf{N} \cdot \mathsf{V} + \overline{\mathsf{N}} \cdot \overline{\mathsf{V}}}{\mathsf{N} \cdot \overline{\mathsf{V}} + \overline{\mathsf{N}} \cdot \mathsf{V}}$

 $N \cdot V \cdot \overline{Z} + \overline{N} \cdot \overline{V} \cdot \overline{Z}$ $Z + N \cdot \overline{V} + \overline{N} \cdot V$

Condition

true false

high low or same

carry clear

carry set

not equal

equal

overflow clear overflow set

plus

minus

greater or equal

less than

		Powers of 1	6,	Powers	of 2	
16 ⁿ	2 ⁿ	Va ue		16 ⁿ	2 ⁿ	Va ue
m=	n=	varue		m=	n=	varue
0	0	1		4	16	65 536
	1	2			17	131 072
	2	4			18	262 144
	3	8			19	524 288
1	4	16		5	20	1 048 576
	5	32			21	2 097 152
	6	64			22	4 194 304
	7	128			23	8 388 608
2	8	256		6	24	16 777 216
	9	512			25	33 554 432
	10	1 024			26	67 108 864
	11	2 048			27	134 217 728
3	12	4 096		7	28	268 435 456
	13	8 192			29	536 870 912
	14	16 384			30	1 07 374 1824
	15	32 768			31	2 147 483 648
		-		8	32	4 294 967 296

	Exceptio	n Vector	Assignment
Vector Number(s)	Vector Offset (Hex)	Stacked Program Counter	Assignment
0	0x000	-	Initial Stack Pointer
1	0x004		Initial Program Counter
2	0x008	Fault	Access Error
3	0x00C	Fault	Address Error
4	0x010	Fault	Illegal Instruction
5	0x014	Fault	Divide by Zero
6-7	0x018-0x01C	-	Reserved
8	0x020	Fault	Privilege Violation
9	0x024	Next	Trace
10	0x028	Fault	Unimplemented line-a opcode
11	0x02C	Fault	Unimplemented line-f opcode
12	0x030	Next	Debug Interrupt
13	0x034	-	Reserved
14	0x038	Fault	Format Error
15-23	0x03C-0x05C	-	Reserved
24	0x060	Next	Spurious Interrupt
25-31	0x064-0x07C	-	Reserved
32-47	0x080-0x0BC	Next	Trap #0-15 Instructions
48-63	0x0C0-0x0FC	-	Reserved
64-255	0x100-0x3FC	Next	User-defined Interrupts
	the PC of the r		n that caused the exception: ction that follows the instruction

1				w .		
			Operati			
	(One	Word,	Specifies	Operation	and	Modes)

uı	gi catei tilali	
LE	less or equal	
	Operation Word	٦
(One Word, Spi	ecifies Operation and Modes)	╛
Exter	nsion Word (If Any)	
Exter	nsion Word (If Any)	٦

						0a -	- D		٥	ہ ر								»
F	Ε	D	С	В	Α	9	8	7	6	5	4	3	2	1	0			ASCII
<u>s</u>	SO	CR	FF	VT	LF	HT	BS	BEL	ACK	ENQ	E0T	ETX	STX	SOH	NUL	0		
S	RS	SĐ	FS	ESC	SUB	Ð	CAN	ETB	NAS	NAK	DC4	DC3	DC2	DC1	DLE	1	×	Character
/		1		+	*))	•	βo	ેશ્	\$	#	"		SP	2	MS. [l '
٠->	~	Ш	^	٠.		9	8	7	6	5	4	3	2	-	0	ω	Dig	ş
0	z	Z	٦	~	J	_	Η	G	F	Е	D	С	В	Α	0	4		(7-Bit
1	>]	١	-	Z	Υ	Х	₩	٧	u	Т	S	R	Q	Ρ	5		₽
0	n	3	_	~	j		h	œ	f	е	d	С	Ь	а	,	6		S
DEL	ı	}		~	z	У	×	*	٧	_	t	s	r	q	Р	7		Code)

		Exception Grouping and Priority
Group	Except ion	Processing
0	Reset Address Error Bus Error	Exception processing begins within two clock cycles. (Highest Priority)
1	Trace Interrupt Illegal Privilege	Exception processing begins before the next instruction.
2	TRAP Zero Divide	Exception processing is started by normal instruction execution. (Lowest Priority)

		Status	Re	gist	er						
Sys	tem Byte				(Co	nd i t i	User on Co		egist	er)	
	人		\	_			J				_
15 * 13	12 *	10 9	8	*	*	*	4	3	2	1	0
T S	м ///	I ₂ I ₁	I _o				Х	N	Z	٧	С
Trace Mode	•					Exte	end Wegat	ive			
 Supervisor		Interrupt	Mask					Z	ero.		
	 aster/Inte tate	rrupt							0ver		arry

ant Bit + Boolean OR	f	ga ga	\$	총	Result Operand -Most Significant Bit	₽
Boolean AND	č	ě	tion 4	E 8	See Special Definition	Î
+ B;+	2 5	ign gn	e st	* *	Source Operand Most Significant Bit	2 8
7						NOTES:
C = D,-1	٠->	0	*	*	*	ASR, LSR
$C = D_{mr+1}$	0	0	*	*	*	LSL
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	-2	>	*	*	*	ASL
$Z = \overline{\Omega n}$	1	1	->		FR .	BTST, BCHG, BSET, BCLR
$\begin{array}{c} V=Dm+Rm\\ C=Dm+Rm\\ Z=Z+\overline{Rm}+\dots+\overline{R0} \end{array}$	>	>	٠-১	*	*	NEGX
$V = Dm \cdot Rm$ C = Dm + Rm	0	0	*	*	*	
	0	0	*	*	רוו –	
V = Division Overflow	0	.2	*	*	VI -	DIVS, DIVU
$V = \overline{Sm} \cdot Dm \cdot \overline{Rm} + Sm \cdot \overline{Dm} \cdot Rm$ $C = Sm \cdot \overline{Dm} + Rm \cdot \overline{Dm} + Sm \cdot Rm$	٠->	-~	*	*		OMP. CMPI
$\begin{array}{llllllllllllllllllllllllllllllllllll$.2	>	-2	*	*	SUBX
$V = \overline{Sm} \bullet Dm \bullet \overline{Rm} + Sm \bullet \overline{Dm} \bullet Rm$ $C = Sm \bullet \overline{Dm} + Rm \bullet \overline{Dm} + Sm \bullet Rm$	>	>	*	*	ii. SUBQ *	SUB, SUBI,
	0	0	*	*	EQ, MOVE, CLR, TAS, TST	AND, ANDI, EOR EORI MOVEQ, MC OR, ORI, CLR, EXT, NOT, TAS,
$\begin{array}{llllllllllllllllllllllllllllllllllll$	٠,	>	٠,	*	*	ADDX
$V = Sm \cdot Dm \cdot Rm + Sm \cdot Dm \cdot Rm$ $C = Sm \cdot Dm + Rm \cdot Dm + Sm \cdot Rm$	0	3	*	*	ADDQ *	ADD, ADDI, ADDO
V C Special Definition	c	<	Z	z	Operations X	0pe
tion Gode Computation	뢰	ဂ္ဂ				

Effec	tive Ac	ldressing M	ode Categories	
Туре	Mode	Register	Generation	Assembler Syntax
Data Register Direct	000	reg. no.	EA = Dn	Dn
Address Register Direct	001	reg. no.	EA = An	An
Register Indirect	010	reg. no.	EA = (An)	(An)
Postincrement Register Indirect	011	reg. no.	EA = (An), An ← An + N	(An)+
Predecrement Register Indirect	100	reg. no.	An ← An –N, EA = (An)	-(An)
Register Indirect with Offset	101	reg. no.	$EA = (An) + d_{16}$	d ₁₆ (An)
Indexed Register Indirect with Offset	110	reg. no.	$EA = (An) + (Xn*SF) + d_8$	d _g (An,Xn*SF)
Absolute Short	111	000	EA = (Next Word)	xxxx.W (xxxx).W
Absolute Long	111	001	EA = (Next Two Words)	xxxxxxxx.L (xxxxxxxx).L
PC Relative with Offset	111	010	$EA = (PC) + d_{16}$	d ₁₆ (PC)
PC Relative with Index and Offset	111	011	$EA = (PC) + (Xn*SF) + d_8$	d _g (PC, Xn*SF)
Immediate	111	100	Data = Next Word(s)	#xxx
Quick Immediate	-	-	Inherent Data	#xxx(1-8)
Implied Register	-	-	EA = SR, USP, SP, PC	-

Notes: EA = Effective Address An = Address Register Xn = Address or Data Register used as Index Register

SR = Status Register d₈ = Eight bit Offset (displacement)
PC = Program Counter d₁₆ = Sixteen bit Offset (displacement)
Dn = Data Registers N = 1 for byte, 2 for words and 4 for long words
() = Contents of
← = Replaces