MC68000 Instruction Set

Instruction	Instruction Description	Assembler Syntax	Data Size	X	N	Z	V	C
ADD	ADD binary	-(Ax),-(Ay)/ Dn,<ea>/ <ea>,Dn	BWL	*	*	*	*	*
ADDA	ADD binary to An	<ea>,An		-	-	-	-	-
ADDI	ADD Immediate	#x,<ea>	BWL	*	*	*	*	*
ADDQ	ADD 3-bit immediate	#<1-8>,<ea>	BWL	*	*	*	*	*
ADDX	ADD eXtended	Dy,Dx	BWL	*	*	*	*	*
AND	Bit-wise AND	-(Ay),-(Ax)/ <ea>,Dn/ Dn,<ea>	BWL	-	*	*	0	0
ANDI	Bit-wise AND with Immediate	#<data>,<ea>	BWL	-	*	*	0	0
ASL	Arithmetic Shift Left	#<1-8>,Dy/ Dx,Dy/ <ea>	BWL	*	*	*	*	*
ASR	Arithmetic Shift Right	...	BWL	*	*	*	*	*
Bcc	Conditional Branch	Bcc.S <label>\ Bcc.W <label>	BW-	-	-	-	-	-
BCHG	Test a Bit and CHanGe	Dn,<ea>/ #<data>,<ea>	B-L	-	-	*	-	-
BCLR	Test a Bit and CLeaR	...	B-L	-	-	*	-	-
BSET	Test a Bit and SET	...	B-L	-	-	*	-	-
BSR	Branch to SubRoutine	BSR.S <label>/ BSR.W <label>	BW-	-	-	-	-	-
BTST	Bit TeST	Dn,<ea>/ #<data>,<ea>	B-L	-	-	*	-	-
CHK	CHecK Dn Against Bounds	<ea>,Dn	-W-	-	*	U	U	U
CLR	CLeaR	<ea>	BWL	-	0	1	0	0
CMP	CoMPare	<ea>,Dn	BWL	-	*	*	*	*
CMPA	CoMPare Address	<ea>,An		-	*	*	*	*
CMPI	CoMPare Immediate	#<data>,<ea>	BWL	-	*	*	*	*
CMPM	CoMPare Memory	(Ay)+,(Ax)+	BWL	-	*	*	*	*
DBcc	Looping Instruction	DBcc Dn,<label>	-W-	-	-	-	-	-
DIVS	DIVide Signed	<ea>,Dn	-W-	-	*	*	*	0
DIVU	DIVide Unsigned	<ea>,Dn	-W-	-	*	*	*	0
EOR	Exclusive OR	Dn,<ea>	BWL	-	*	*	0	0
EORI	Exclusive OR Immediate	#<data>,<ea>	BWL	-	*	*	0	0
EXG	Exchange any two registers	Rx,Ry		-	-	-	-	-
EXT	Sign EXTend	Dn		-	*	*	0	0
JMP	JuMP to Affective Address	<ea>		-	-	-	-	-
JSR	Jump to SubRoutine	<ea>		-	-	-	-	-
LSL	Logical Shift Left	Dx,Dy/ #<1-8>,Dy/ <ea>	BWL	*	*	*	0	*
LSR	Logical Shift Right	...	BWL	*	*	*	0	*
MOVE	Between Effective Addresses	<ea>,<ea>	BWL	-	*	*	0	0
MOVE	To CCR	<ea>,CCR	-W-	I	I	I	I	I
MOVE	To SR	<ea>,SR	-W-	I	I	I	I	I
MOVE	From SR	SR,<ea>	-W-	-	-	-	-	-
MOVE	USP to/from Address Register	USP,An/n,USP		-	-	-	-	-
MOVEA	MOVE Address	<register list>,<ea>		-	-	-	-	-
MOVEM	MOVE Multiple	<ea>,<register list>		-	-	-	-	-
MOVEQ	MOVE 8-bit immediate	#<-128.+127>,Dn		-	*	*	0	0
MULS	MULtiply Signed	<ea>,Dn	-W-	-	*	*	0	0
MULU	MULtiply Unsigned	<ea>,Dn	-W-	-	*	*	0	0
NEG	NEGate	<ea>	BWL	*	*	*	*	*
NEGX	NEGate with eXtend	<ea>	BWL	*	*	*	*	*
NOT	Form one's complement	<ea>	BWL	-	*	*	0	0
OR	Bit-wise OR	<ea>,Dn/Dn,<ea>	BWL	-	*	*	0	0
ORI	Bit-wise OR with Immediate	#<data>,<ea>	BWL	-	*	*	0	0
ROL	ROtate Left	#<1-8>,Dy\ Dx,Dy/ <ea>	BWL	-	*	*	0	*
ROR	ROtate Right	...	BWL	-	*	*	0	*
ROXL	ROtate Left with eXtend	...	BWL	*	*	*	0	*

ROXR	ROtate Right with eXtend	...	BWL	*	*	*	0	*
RTE	ReTurn from Exception	RTE		I	I	I	I	I
RTR	ReTurn and Restore	RTR		I	I	I	I	I
RTS	ReTurn from Subroutine	RTS		-	-	-	-	-
Scc	Set to -1 if True, 0 if	False <ea>	B--	-	-	-	-	-
SUB	SUBtract binary	Dn,<ea>/ <ea>,Dn	BWL	*	*	*	*	*
SUBA	SUBtract binary from An	<ea>,An		-	-	-	-	-
SUBI	SUBtract Immediate	#x,<ea>	BWL	*	*	*	*	*
SUBQ	SUBtract 3-bit immediate	#<data>,<ea>	BWL	*	*	*	*	*
SUBX	SUBtract eXtended	Dy,Dx-\ (Ay),-(Ax)	BWL	*	*	*	*	*
SWAP	SWAP words of Dn	Dn	-W-	-	*	*	0	0
TRAP	Execute TRAP Exception	#<vector>		-	-	-	-	-
TRAPV	TRAPV Exception if V-bit	Set TRAPV		-	-	-	-	-
TST	TeST for negative or zer	o <ea>	BWL	-	*	*	0	0

Symbol	**Meaning**
*	Set according to result of operation
-	Not affected
0	Cleared
1	Set
U	Outcome (state after operation) undefined
I	Set by immediate data
<ea>	Effective Address Operand
<data>	Immediate data
<label>	Assembler label
<vector>	TRAP instruction Exception vector (0-15)
<rg.lst>	MOVEM instruction register specification list
<displ.>	LINK instruction negative displacement
...	Same as previous instruction

Condition Codes for Bcc, DBcc and Scc Instructions

Relationship	**Signed**	**Unsigned**
D1 < D0	CS - Carry Bit Set	LT - Less Than
D1 <= D0	LS - Lower or Same	LE - Less than or Equal
D1 = D0	EQ - Equal (Z-bit Set)	EQ - Equal (Z-bit Set)
D1 != D0	NE - Not Equal (Z-bit Clear)	NE - Not Equal (Z-bit Clear)
D1 > D0	HI - HIgher than	GT - Greater Than
D1 >= D0	CC - Carry Bit Clear PL - PLus (N-bit Clear) VC - V-bit Clear (No Overflow)	GE - Greater than or Equal MI - Minus (N-bit Set) VS - V-bit Set (Overflow)
	RA - BRanch Always	
DBcc Only	F - Never Terminate (DBRA is an alternate to DBF) T - Always Terminate	
Scc Only	SF - Never Set ST - Always Set	