Motorola 68000 Instruction Set.

Instruction	Description	Assembler Syntax	Data Size	Condition Codes X	N	Z	V	C
ABCD	Add BCD with extend	Dx,Dy	B--	*	U	*	U	*
		-(Ax),-(Ay)						
ADD	ADD binary	Dn,<ea>	BWL	*	*	*	*	*
		<ea>,Dn						
ADDA	ADD binary to An	<ea>,An	-WL	-	-	-	-	-
ADDI	ADD Immediate	#x,<ea>	BWL	*	*	*	*	*
ADDQ	ADD 3-bit immediate	#<1-8>,<ea>	BWL	*	*	*	*	*
ADDX	ADD eXtended	Dy,Dx	BWL	*	*	*	*	*
		-(Ay),-(Ax)						
AND	Bit-wise AND	<ea>,Dn	BWL	-	*	*	0	0
		Dn,<ea>						
ANDI	Bit-wise AND with Immediate	#<data>,<ea>	BWL	-	*	*	0	0
ASL	Arithmetic Shift Left	#<1-8>,Dy	BWL	*	*	*	*	*
		Dx,Dy						
		<ea>						
ASR	Arithmetic Shift Right	...	BWL	*	*	*	*	*
Bcc	Conditional Branch	Bcc.S <label>	BW-	-	-	-	-	-
		Bcc.W <label>						
BCHG	Test a Bit and CHanGe	Dn,<ea>	B-L	-	-	*	-	-
		#<data>,<ea>						
BCLR	Test a Bit and CLeaR	...	B-L	-	-	*	-	-
BSET	Test a Bit and SET	...	B-L	-	-	*	-	-
BSR	Branch to SubRoutine	BSR.S <label>	BW-	-	-	-	-	-
		BSR.W <label>						
BTST	Bit TeST	Dn,<ea>	B-L	-	-	*	-	-
		#<data>,<ea>						
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	-WL	-	*	*	*	*
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	--L	-	-	-	-	-
EXT	Sign EXTend	Dn	-WL	-	*	*	0	0
ILLEGAL	ILLEGAL-Instruction Exception	ILLEGAL		-	-	-	-	-
JMP	JuMP to Affective Address	<ea>		-	-	-	-	-
JSR	Jump to SubRoutine	<ea>		-	-	-	-	-
LEA	Load Effective Address	<ea>,An	--L	-	-	-	-	-
LINK	Allocate Stack Frame	An,#<displacement>		-	-	-	-	-
LSL	Logical Shift Left	Dx,Dy	BWL	*	*	*	0	*
		#<1-8>,Dy						
		<ea>						
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	--L	-	-	-	-	-
		An,USP						

MOVEA	MOVE Address	<ea>,An	-WL	- - - - -
MOVEM	MOVE Multiple	<register list>,<ea> <ea>,<register list>	-WL	- - - - -
MOVEP	MOVE Peripheral	Dn,x(An) x(An),Dn	-WL	- - - - -
MOVEQ	MOVE 8-bit immediate	#<-128.+127>,Dn	--L	- * * 0 0
MULS	MULtiply Signed	<ea>,Dn	-W-	- * * 0 0
MULU	MULtiply Unsigned	<ea>,Dn	-W-	- * * 0 0
NBCD	Negate BCD	<ea>	B--	* U * U *
NEG	NEGate	<ea>	BWL	* * * * *
NEGX	NEGate with eXtend	<ea>	BWL	* * * * *
NOP	No OPeration	NOP		- - - - -
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
PEA	Push Effective Address	<ea>	--L	- - - - -
RESET	RESET all external devices	RESET		- - - - -
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		- - - - -
SBCD	Subtract BCD with eXtend	Dx,Dy	B--	* U * U *