

68000 Instruction Set Summary

Mnemonic	Data Size	Operands	Instruction Description	Condition Codes	Dn	An	(An)	(An)+	-(An)	w(An)	b(An,Rx)	W	L	label(PC)	label(PC,Rx)	#immed
				X N Z V C	<ea>	-	Allowed Addressing Modes									
ABCD	B---	Dy, Dx -(Ay), -(Ax)	Add BCD with extend	* U * U *												
ADD	BWL	Dn, <ea> <ea>, Dn	ADD binary	* * * * *	X	X	X	X	X
ADDA	-WL	<ea>, An	ADD binary to An	- - - - -
ADDI	BWL	#, <ea>	ADD Immediate	* * * * *	.	X	X	X	X
ADDQ	BWL	#<1-8>, <ea>	ADD 3-bit Immediate	* * * * *	X	X	X
ADDX	BWL	Dy, Dx -(Ay), -(Ax)	ADD extended	* * * * *	X	X	X
AND	BWL	<ea>, Dn Dn, <ea>	Bit-wise AND	- * * 0 0	.	X	X	X	X
ANDI	BWL	#<data>, <ea> B--- #<data>, CCR -W- #<data>, SR	Bit-wise AND with Immediate	- * * 0 0 * * * * * * * * * *	.	X	X	X	X
ASL	BWL	#<1-8>, Dy Dx, Dy <ea>	Arithmetic Shift Left Dy shifted by Dx mod 64 One bit shift only	* * * * *	X	X	X	X	X
ASR	BWL	#<1-8>, Dy Dx, Dy <ea>	Arithmetic Shift Right Dy shifted by Dx mod 64 One bit shift only	* * * * *	X	X	X	X	X
Bcc *	Bcc, S Bcc	<label>	Conditional Branch * See notes at end for conditions	- - - - -										X	X	X
BCHG	B-L	Dn, <ea> #<data>, <ea>	Test a Bit and Change	- - * - -	2	X	3	3	3	3	3	3	3	X	X	X
BCLR	B-L	Dn, <ea> #<data>, <ea>	Test a Bit and Clear	- - * - -	2	X	3	3	3	3	3	3	3	X	X	X
BSET	B-L	Dn, <ea> #<data>, <ea>	Test a Bit and SET	- - * - -	2	X	3	3	3	3	3	3	3	X	X	X
BSR	BSR, S BSR	<label>	Branch to Subroutine	- - - - -										X	X	X
BTST	B-L	Dn, <ea> #<data>, <ea>	Bit Test	- - * - -	2	X	3	3	3	3	3	3	3	X	X	X
CHK	-W-	<ea>, Dn	Check Dn Against Bounds	- * U U U	.	X
CLR	BWL	<ea>	Clear	- 0 1 0 0	.	X	X	X	X
COMP	BWL	<ea>, Dn	Compare	- * * * *	.	1
CMPA	-WL	<ea>, An	Compare Address	- * * * *
CMPI	BWL	#<data>, <ea>	Compare Immediate	- * * * *	.	X	X	X	X
CMPI	BWL	(Ay)+, (Ax)+	Compare Memory	- * * * *	X	X	X
DBcc *	-W-	Dncc Dn, <label>	Looping Instruction * See notes at end for conditions	- - - - -												
DIVS	-W-	<ea>, Dn	DIVIDE Signed	- * * * 0	.	X
DIVU	-W-	<ea>, Dn	DIVIDE Unsigned	- * * * 0	.	X

68000 Instruction Set Summary

Mnemonic	Data Size	Operands	Instruction Description	Condition Codes	Dn	An	(An)	(An)+	-(An)	w(An)	b(An,Rx)	W	L	label(PC)	label(PC,Rx)	#immed
				X N Z V C	<ea>	-	Allowed Addressing Modes									
BOB	BWL	Dn, <ea>	Exclusive OR	- * * 0 0	.	X	X	X	X
EOB1	BWL	#<data>, <ea> B--- #<data>, CCR -W- #<data>, SR	Exclusive OR Immediate	- * * 0 0 * * * * *	.	X	X	X	X
EXG	-L	Rx, Ry	Exchange any two registers	- - - - -												
EXT	-WL	Dn	Sign Extend	- * * 0 0												
ILLBGL			ILLBGL-Instruction Exception	- - - - -												
JMP		<ea>	Jump to Affective Address	- - - - -										X	X	X
JSR	-L	<ea>	Jump to Subroutine	- - - - -										X	X	X
LEA		<ea>, An	Load Effective Address	- - - - -										X	X	X
LINK		An, #<displacement>	Allocate Stack Frame	- - - - -												
LSL	BWL	#<1-8>, Dy Dx, Dy <ea>	Logical Shift Left Dy shifted by Dx mod 64 One bit shift only	* * * 0 *	X	X	X	X	X
LSR	BWL	#<1-8>, Dy Dx, Dy <ea>	Logical Shift Right Dy shifted by Dx mod 64 One bit shift only	* * * 0 *	X	X	X	X	X
MOVE	BWL	<ea>, <ea>	Between Effective Addresses	- * * 0 0	1	X	X	X
MOVEA	-WL	<ea>, An	MOVE Address	- - - - -												
MOVEM	-WL	<reg_list>, <ea> <ea>, <reg_list>	MOVE Multiple	- - - - -										X	X	X
MOVEP	-WL	Dn, x(An) x(An), Dn	MOVE Peripheral	- - - - -										X	X	X
MOVEQ	-L	#<-128, +127>, Dn	MOVE 8-bit Immediate	- * * 0 0										X	X	X
MULS	-W-	<ea>, Dn	Multiply Signed	- * * 0 0	.	X
MULU	-W-	<ea>, Dn	Multiply Unsigned	- * * 0 0	.	X
NEGCD	B--	<ea>	Negate BCD	* U * U *	.	X	X	X	X
NEG	BWL	<ea>	Negate	* * * * *	.	X	X	X	X
NEGX	BWL	<ea>	Negate with extend	* * * * *	.	X	X	X	X
NOP			No Operation	- - - - -												
NOT	BWL	<ea>	Form one's complement	- * * 0 0	.	X	X	X	X

68000 Instruction Set Summary

Mnemonic	Data Size	Operands	Instruction Description	Condition Codes	Dn	An	(An)	(An)+	-(An)	w(An)	b(An,Rx)	W	L	label(PC)	label(PC,Rx)	#immed
OR	BWL	<ea>, Dn, <ea>	Bit-wise OR	X N Z V C	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X
ORI	BWL	#<data>, <ea>	Bit-wise OR with Immediate	- * * 0 0	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X
	-W	#<data>, SR		* * * * *	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X
PSA	-L	<ea>	Push Effective Address	- - - - -	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X
RESET			RESET all external devices	- - - - -												
ROL	BWL	#<1-8>, Dn, Dn, <ea>	Rotate Left	- * * 0 *	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X
ROR	BWL	#<1-8>, Dn, Dn, <ea>	Rotate Right	- * * 0 *	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X
ROXL	BWL	#<1-8>, Dn, Dn, <ea>	Rotate Left with extend	* * * 0 *	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X
ROXR	BWL	#<1-8>, Dn, Dn, <ea>	Rotate Right with extend	* * * 0 *	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X
RTE			Return from Exception	I I I I I												
RTR			Return and Restore	I I I I I												
RTS			Return from Subroutine	- - - - -												
SECD	B--	Dx, Dy, -(Ax), -(Ay)	Subtract BCD with extend	* U * U *												
Sec *	B--	<ea>	Set to -1 if True, 0 if False	- - - - -	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X
			* See notes at end for conditions													
STOP		#<data>	Enable & wait for interrupts	I I I I I												
SUB	BWL	Dn, <ea>	Subtract binary	* * * * *	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X
		<ea>, Dn			. 1	. 1	. 1	. 1	. 1	. 1	. 1	. 1	. 1	. 1	. 1	. 1
SUBA	-WL	<ea>, An	Subtract binary from An	- - - - -	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X
SUBI	BWL	#x, <ea>	Subtract Immediate	* * * * *	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X
SUBQ	BWL	#<data>, <ea>	Subtract 3-bit immediate	* * * * *	. 1	. 1	. 1	. 1	. 1	. 1	. 1	. 1	. 1	. 1	. 1	. 1
SUBX	BWL	Dy, Dx, -(Ay), -(Ax)	Subtract extended	* * * * *	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X
SWAP	-W	Dn	SWAP words of Dn	- * * 0 0												
TAS	B--	<ea>	Test & Set MSB & Set N/Z-bits	- * * 0 0	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X
TRAP		#<vector>	Execute TRAP Exception	- - - - -												
TRAPV			TRAPV Exception if V-bit Set	- - - - -												
TST	BWL	<ea>	Test for negative or zero	- * * 0 0	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X	. X
UNLK		An	Deallocate Stack Frame	- - - - -												

Symbol	Meaning	Symbol	Meaning
*	Set according to result of operation	<ea>	Effective Address Operand
-	Not affected	<data>	Immediate data
0	Cleared	<label>	Assembler label
1	Set	<vector>	TRAP instruction Exception vector (0-15)
U	Outcome (state after operation) undefined	<reg, lab>	MOVEM instruction register list
I	Set by immediate data	<diapl>	LINK instruction negative displacement

Addressing Modes	Syntax	Legend
Data Register Direct	Dn	Dn Data Register (n is 0-7)
Address Register Direct	An	An Address Register (n is 0-7)
Address Register Indirect	(An)	b 08-bit constant
Address Register Indirect with Post-Increment	(An)+	w 16-bit constant
Address Register Indirect with Pre-Decrement	-(An)	1 32-bit constant
Address Register Indirect with Displacement	w(An)	x 8-, 16-, 32-bit constant
Address Register Indirect with Index	b(An, Rx)	Rx Index Register specification, one of: Dn, W Low 16 bits of Data Register Dn, L All 32 bits of Data Register An, W Low 16 bits of Address Register An, L All 32 bits of Address Register
Absolute Short	w	X This mode is not available.
Absolute Long	l	1 This mode is available.
Program Counter with Displacement	label(PC)	2 Word & long size only.
Program Counter with Index	label(PC, Rx)	3 Byte size only.
Immediate	#x	
Status Register	SR	
Condition Code Register	CCR	

Condition Codes for Bcc, Dbec and Sec Instructions.

Condition Codes set after CMP D0, D1 instruction.

Relationship	Unsigned	Signed
D1 < D0	LO, CS - Carry Bit Set	LT - Less Than
D1 <= D0	LS - Lower or Same	LE - Less than or Equal
D1 = D0	EQ - Equal (2-bit Set)	EQ - Equal (2-bit Set)
D1 != D0	NE - Not Equal (2-bit Clear)	NE - Not Equal (2-bit Clear)
D1 > D0	HI - Higher than	GT - Greater Than
D1 >= D0	HS, CC - Carry Bit Clear	GE - Greater than or Equal
	PL - Plus (N-bit Clear)	MI - Minus (N-bit Set)
	VC - V-bit Clear (No Overflow)	VS - V-bit Set (Overflow)
	RA - Branch Always	

Dbec Only - F - Never Terminate (DBRA is an alternate to DBP)
T - Always Terminate

Sec Only - SF - Never Set
ST - Always Set

Parts from "Programming the 68000" by Steve Williams. (c) 1985 Sybex Inc.
Parts from BYTE Magazine article.
Compiled by Diego Barros. e-mail: allen@zikzak.apana.org.au

Severely hacked around by Phil Rice 26/2/97
rice@ronmark.dendigo.lacrobe.edu.au