68000 Instruction Set Summary

Mnemonic	Data Size	Operands	Instruction Description	Condition Codes X N Z V C	Dn	An	(An)	(An)+	-(An)	w(An)	b(An,Rx)	W	L	label(PC)	label(PC,Rx)	#immed
					<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 1	. .	. .	. .	. .	. .	. .	. .	X .	X .	X .
ADDA	-WL	<ea>,An	ADD binary to An	- - - - -	.	.	.	.	.	.	.	.	.	.	.	.
ADDI	BWL	#x,<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	* * * * *												
AND	BWL	<ea>,Dn Dn,<ea>	Bit-wise AND	- * * 0 0	. X	X X	. .	. .	. .	. .	. .	. .	. .	. X	. X	. X
ANDI	BWL B-- -W-	#<data>,<ea> #<data>,CCR #<data>,SR	Bit-wise AND with Immediate	- * * 0 0 * * * * * * * * * *	.	X	.	.	.	.	.	.	.	X	X	X
ASL	BWL -W-	#<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 -W-	#<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	- - - - -												
BCHG	B-L	Dn,<ea> #<data>,<ea>	Test a Bit and CHanGe	- - * - -	2 2	X X	3 3	3 3	3 3	3 3	3 3	3 3	3 3	X X	X X	X X
BCLR	B-L	Dn,<ea> #<data>,<ea>	Test a Bit and CLeaR	- - * - -	2 2	X X	3 3	3 3	3 3	3 3	3 3	3 3	3 3	X X	X X	X X
BSET	B-L	Dn,<ea> #<data>,<ea>	Test a Bit and SET	- - * - -	2 2	X X	3 3	3 3	3 3	3 3	3 3	3 3	3 3	X X	X X	X X
BSR	BSR.S BSR	<label>	Branch to SubRoutine	- - - - -												
BTST	B-L	Dn,<ea> #<data>,<ea>	Bit TeST	- - * - -	2 2	X X	3 3	3 3	3 3	3 3	3 3	3 3	3 3	3 3	3 3	3 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
CMP	BWL	<ea>,Dn	CoMPare	- * * * *	.	1	.	.	.	.	.	.	.	.	.	.
CMPA	-WL	<ea>,An	CoMPare Address	- * * * *	.	.	.	.	.	.	.	.	.	.	.	.
CMPI	BWL	#<data>,<ea>	CoMPare Immediate	- * * * *	.	X	.	.	.	.	.	.	.	X	X	X
CMPM	BWL	(Ay)+,(Ax)+	CoMPare Memory	- * * * *												
DBcc*	-W-	DBcc 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 X N Z V C	Dn	An	(An)	(An)+	-(An)	w(An)	b(An,Rx)	W	L	label(PC)	label(PC,Rx)	#immed
					<ea> - Allowed Addressing Modes											
EOR	BWL	Dn,<ea>	Exclusive OR	- * * 0 0	.	X	.	.	.	.	.	.	.	X	X	X
EORI	BWL	#<data>,<ea>	Exclusive OR Immediate	- * * 0 0	.	X	.	.	.	.	.	.	.	X	X	X
	B--	#<data>,CCR		* * * * *												
	-W-	#<data>,SR		* * * * *												
EXG	--L	Rx,Ry	Exchange any two registers	- - - - -												
EXT	-WL	Dn	Sign EXTend	- * * 0 0												
ILLEGAL			ILLEGAL-Instruction Exception	- - - - -												
JMP		<ea>	JuMP to Affective Address	- - - - -	X	X	.	X	X	.	.	.	.	.	.	X
JSR	--L	<ea>	Jump to SubRoutine	- - - - -	X	X	.	X	X	.	.	.	.	.	.	X
LEA		<ea>,An	Load Effective Address	- - - - -	X	X	.	X	X	.	.	.	.	.	.	X
LINK		An,#<displacement>	Allocate Stack Frame	- - - - -												
LSL	BWL	#<1-8>,Dy	Logical Shift Left	* * * 0 *												
		Dx,Dy	Dy shifted by Dx mod 64													
	-W-	<ea>	One bit shift only		X	X	.	.	.	.	.	.	.	X	X	X
LSR	BWL	#<1-8>,Dy	Logical Shift Right	* * * 0 *												
		Dx,Dy	Dy shifted by Dx mod 64													
	-W-	<ea>	One bit shift only		X	X	.	.	.	.	.	.	.	X	X	X
MOVE	BWL	<ea>,<ea>	Between Effective Addresses	- * * 0 0												
				Source <ea>	.	1	.	.	.	.	.	.	.	.	.	.
				Destination <ea>	.	X	.	.	.	.	.	.	.	X	X	X
MOVE	-W-	<ea>,CCR	To CCR	I I I I I	.	X	.	.	.	.	.	.	.	.	.	.
MOVE	-W-	CCR,<ea>	From CCR	- - - - -	.	X	.	.	.	.	.	.	.	X	X	X
MOVE	-W-	<ea>,SR	To SR	I I I I I	.	X	.	.	.	.	.	.	.	.	.	.
MOVE	-W-	SR,<ea>	From SR	- - - - -	.	X	.	.	.	.	.	.	.	X	X	X
MOVE	--L	SP,An	SP to/from Address Register	- - - - -												
		An,SP														
MOVEA	-WL	<ea>,An	MOVE Address	- - - - -	.	.	.	.	.	.	.	.	.	.	.	.
MOVEM	-WL	<rg.lst>,<ea>	MOVE Multiple	- - - - -	X	X	.	X	.	.	.	.	.	X	X	X
		<ea>,<rg.lst>			X	X	.	.	X	.	.	.	.	.	.	X
MOVEP	-WL	Dn,x(An)	MOVE Peripheral	- - - - -												
		x(An),Dn														
MOVEQ	--L	#<-128.+127>,Dn	MOVE 8-bit immediate	- * * 0 0												
MULS	-W-	<ea>,Dn	MULtiply Signed	- * * 0 0	.	X	.	.	.	.	.	.	.	.	.	.
MULU	-W-	<ea>,Dn	MULtiply Unsigned	- * * 0 0	.	X	.	.	.	.	.	.	.	.	.	.
NBCD	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 X N Z V C	Dn	An	(An)	(An)+	-(An)	w(An)	b(An,Rx)	W	L	label(PC)	label(PC,Rx)	#immed
					<ea> - Allowed Addressing Modes											
OR	BWL	<ea>,Dn	Bit-wise OR	- * * 0 0	.	X	.	.	.	.	.	.	.	.	.	.
		Dn,<ea>			X	X	.	.	.	.	.	.	.	X	X	X
ORI	BWL	#<data>,<ea>	Bit-wise OR with Immediate	- * * 0 0	.	X	.	.	.	.	.	.	.	X	X	X
	-W-	#<data>,CCR		* * * * *												
	-W-	#<data>,SR		* * * * *												
PEA	--L	<ea>	Push Effective Address	- - - - -	X	X	.	X	X	.	.	.	.	.	.	X
RESET			RESET all external devices	- - - - -												
ROL	BWL	#<1-8>,Dy	ROtate Left	- * * 0 *												
		Dx,Dy														
		<ea>			X	X	.	.	.	.	.	.	.	X	X	X
ROR	BWL	#<1-8>,Dy	ROtate Right	- * * 0 *												
		Dx,Dy														
		<ea>			X	X	.	.	.	.	.	.	.	X	X	X
ROXL	BWL	#<1-8>,Dy	ROtate Left with eXtend	* * * 0 *												
		Dx,Dy														
		<ea>			X	X	.	.	.	.	.	.	.	X	X	X
ROXR	BWL	#<1-8>,Dy	ROtate Right with eXtend	* * * 0 *												
		Dx,Dy														
		<ea>			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	- - - - -												
SBCD	B--	Dx,Dy	Subtract BCD with eXtend	* U * U *												
		-(Ax),-(Ay)														
Scc*	B--	<ea>	Set to -1 if True, 0 if False	- - - - -	.	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
		<ea>,Dn			.	1	.	.	.	.	.	.	.	.	.	.
SUBA	-WL	<ea>,An	SUBtract binary from An	- - - - -	.	.	.	.	.	.	.	.	.	.	.	.
SUBI	BWL	#x,<ea>	SUBtract Immediate	* * * * *	.	X	.	.	.	.	.	.	.	X	X	X
SUBQ	BWL	#<data>,<ea>	SUBtract 3-bit immediate	* * * * *	.	1	.	.	.	.	.	.	.	X	X	X
SUBX	BWL	Dy,Dx	SUBtract eXtended	* * * * *												
		-(Ay),-(Ax)														
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
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
UNLK		An	Deallocate Stack Frame	- - - - -												

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

Symbol	Meaning
<ea>	Effective Address Operand
<data>	Immediate data
<label>	Assembler label
<vector>	TRAP instruction Exception vector (0-15)
<rg.lst>	MOVEM instruction register list
<displ.>	LINK instruction negative displacement

Addressing Modes	Syntax
Data Register Direct	Dn
Address Register Direct	An
Address Register Indirect	(An)
Address Register Indirect with Post-Increment	(An)+
Address Register Indirect with Pre-Decrement	-(An)
Address Register Indirect with Displacement	w(An)
Address Register Indirect with Index	b(An,Rx)
Absolute Short	w
Absolute Long	l
Program Counter with Displacement	label(PC)
Program Counter with Index	label(PC,Rx)
Immediate	#x
Status Register	SR
Condition Code Register	CCR

Legend		
Dn	Data Register	(n is 0-7)
An	Address Register	(n is 0-7)
b	08-bit constant	
w	16-bit constant	
l	32-bit constant	
x	8-, 16-, 32-bit constant	
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	
X	This mode not available.	
.	This mode is available.	
1	Word & Long size only.	
2	Long size only.	
3	Byte size only.	

Condition Codes for Bcc, DBcc and Scc 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 (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	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	
DBcc Only -	F - Never Terminate (DBRA is an alternate to DBF)	
	T - Always Terminate	
Scc 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 : alien@zikzak.apana.org.au

Severely hacked around by Phil Rice 26/2/97
rice@ironbark.bendigo.latrobe.edu.au