

Instruction set of the 68000 in alphabetical order

		0	2	3	4	5
abcd	Ds,Dd	1100	ddd100	000sss		
abcd	-(As), -(Ad)	1100	ddd100	001sss		
add.z	a,Dd	1101	ddd0zz	aaaaaa		
add.z	Ds,a	1101	sss1zz	aaaaaa		
adda.z	a,Ad	1101	dddz11	aaaaaa		
addi.z	#kz,a	0000	0110zz	aaaaaa	kkkkkkkkk	kkkkkkkkk kifz==lk kifz==lk
addq.z	#k3,a	0101	kkk0zz	aaaaaa		
addx.z	Ds,Dd	1101	ddd1zz	000sss		
addx.z	-(As), -(Ad)	1101	ddd1zz	001sss		
and.z	a,Dd	1100	ddd0zz	aaaaaa		
and.z	Ds,a	1100	sss1zz	aaaaaa		
andi.z	#kz,a	0000	0010zz	aaaaaa	kkkkkkkkk	kkkkkkkkk kifz==lk kifz==lk
asld.z	Ds,Dd	1110	sss1zz	100ddd		
asld.z	#k,Dd	1110	kkk1zz	000ddd		
asld.w	#1,a	1110	000111	aaaaaa		
asrd.z	Ds,Dd	1110	sss0zz	100ddd		
asrd.z	#k,Dd	1110	kkk0zz	000ddd		
asrd.w	#1,a	1110	000011	aaaaaa		
bcc	label	0110	ccc11	111111	if zero, a 16bit address follows	
bra	label	0110	000011	111111	if zero, a 16bit address follows	
bsr	label	0110	000111	111111	if zero, a 16bit address follows	
bchg	#n,a	0000	100001	aaaaaa	00000000	000nnnnn
bchg	Ds,a	0000	sss101	aaaaaa		
bclr	#n,a	0000	100010	aaaaaa	00000000	000nnnnn
bclr	Ds,a	0000	sss110	aaaaaa		
bset	#n,a	0000	100011	aaaaaa	00000000	000nnnnn
bset	Ds,a	0000	sss111	aaaaaa		
btst	#n,a	0000	100000	aaaaaa	00000000	000nnnnn
btst	Ds,a	0000	sss100	aaaaaa		
chk.w	a,Dd	0100	ddd110	aaaaaa		
clr.z	a	0100	0010zz	aaaaaa		
cmp.z	a,Dd	1011	ddd0zz	aaaaaa		
cmpa.z	a,Ad	1011	dddz11	aaaaaa		
cmpi.z	#kz,a	0000	1100zz	aaaaaa	kkkkkkkkk	kkkkkkkkk kifz==lk kifz==lk
cmpm.z	Ds,Dd	1011	ddd1zz	000sss		
cmpm.z	(As)+, (Ad)+	1011	ddd1zz	001sss		
db(cc)	Ds,label	0101	cccc11	001sss	11111111	11111111
divs.w	a,Dd	1000	ddd111	aaaaaa		
divu.w	a,Dd	1000	ddd011	aaaaaa		
eor.z	a,Dd	1011	ddd1zz	aaaaaa		
eori.z	#kz,a	0000	1010zz	aaaaaa	kkkkkkkkk	kkkkkkkkk kifz==lk kifz==lk
exg.l	As,Ad	1100	ddd101	001sss		
exg.l	Ds,Dd	1100	ddd101	000sss		
exg.l	As,Dd	1100	ddd110	001sss		
ext.z	Dd	0100	10001z	000ddd		
jmp	a	0100	111011	aaaaaa		
jsr	a	0100	111010	aaaaaa		

lea	a,Ad	0100	ddd111	aaaaaa				
link	As,d16	0100	111001	010sss	dddddddd	dddddddd		
lsl.d.z	Ds,Dd	1110	sss1zz	101ddd				
lsl.d.z	#k,Dd	1110	kkk1zz	001ddd				
lsl.d.w	#1,a	1110	001111	aaaaaa				
lsrd.z	Ds,Dd	1110	sss0zz	101ddd				
lsrd.z	#k,Dd	1110	kkk0zz	001ddd				
lsrd.w	#1,a	1110	001011	aaaaaa				
move	SR,a	0100	000011	aaaaaa				
move	a,CCR	0100	010011	aaaaaa				
move	a,SR	0100	011011	aaaaaa				
move	As,USP	0100	111001	100sss				
move	USP,Ad	0100	111001	101ddd				
move.b	as,ad	0001	dddddd	ssssss	(dddddd	reversed)		
move.l	as,ad	0010	dddddd	ssssss	(dddddd	reversed)		
move.w	as,ad	0011	dddddd	ssssss	(dddddd	reversed)		
movem.z	a,reg-list	0100	11001z	aaaaaa	a6543210	d6543210		
movem.z	reg-list,a	0100	10001z	aaaaaa	a6543210	d6543210		
movep.z	Ds,d16(Ad)	0000	sss11z	001ddd	dddddddd	dddddddd		
movep.z	d16(As),Dd	0000	ddd10z	001sss	dddddddd	dddddddd		
moveq.l	#k8,Dd	0111	ddd0kk	kkkkkk				
mul.s.w	a,Dd	1100	ddd111	aaaaaa				
mul.w	a,Dd	1100	ddd011	aaaaaa				
nbcd	a	0100	100000	aaaaaa				
neg.z	a	0100	0100zz	aaaaaa				
negx.z	a	0100	0000zz	aaaaaa				
nop		0100	111001	110001				
not.z	a	0100	0110zz	aaaaaa				
or.z	a,Dd	1000	ddd0zz	aaaaaa				
or.z	Ds,a	1000	sss1zz	aaaaaa				
ori.z	#kz,a	0000	0000zz	aaaaaa	kkkkkkkk	kkkkkkkk	kifz==lk	kifz==lk
pea	a	0100	100001	aaaaaa				
reset		0100	111001	110000				
rold.z	Ds,Dd	1110	sss1zz	111ddd				
rold.z	#k,Dd	1110	kkk1zz	011ddd				
rold.w	#1,a	1110	011111	aaaaaa				
rord.z	Ds,Dd	1110	sss0zz	111ddd				
rord.z	#k,Dd	1110	kkk0zz	011ddd				
rord.w	#1,a	1110	011011	aaaaaa				
roxld.z	Ds,Dd	1110	sss1zz	110ddd				
roxld.z	#k,Dd	1110	kkk1zz	010ddd				
roxld.w	#1,a	1110	010111	aaaaaa				
roxrd.z	Ds,Dd	1110	sss0zz	110ddd				
roxrd.z	#k,Dd	1110	kkk0zz	010ddd				
roxrd.w	#1,a	1110	010011	aaaaaa				
rte		0100	111001	110011				
rtr		0100	111001	110111				
rts		0100	111001	110101				
sbcd	Ds,Dd	1000	ddd100	000sss				
sbcd	-(As),-(Ad)	1000	ddd100	001sss				
stcc.b	a	0101	cccc11	aaaaaa				
stop	#k16	0100	111001	110010	kkkkkkkk	kkkkkkkk		

```

sub.z   Ds,a           1001 sss1zz aaaaaa
sub.z   a,Dd          1001 ddd0zz aaaaaa
subi.z  #kz,a         0000 0100zz aaaaaa kkkkkkkk kkkkkkkk kifz==lk kifz==lk
suba.z  a,Ad          1001 dddz1l aaaaaa
subq.z  #k3,a         0101 kkk1zz aaaaaa
subx.z  Ds,Dd         1001 ddd1zz 000sss
subx.z  -(As), -(Ad) 1001 ddd1zz 001sss

swap.w  Dd            0100 100001 000ddd
tas.b   a              0100 101011 aaaaaa
trap    vector        0100 111001 00vvvv
trapv   0100 111001 110110
tst.z   a              0100 1010zz aaaaaa

unlk    Ad             0100 111001 011ddd

```

General subfields

```

zz      00=b, 01=w, 10=l
z       0=w, 1=l
kkk     immediate data in addq etc.: 0==8
        immediate data in movq is sign extended to 32bits
        shift count also represents 1..8 (how?)

```

Addresssing modes (aaaaaa):

```

000 rrr  Dr
001 rrr  Ar
010 rrr  (Ar)
011 rrr  (Ar)+
100 rrr  -(Ar)
101 rrr  d16(Ar)      dddddddd dddddddd
110 rrr  d8(Ar,ix)   aiiiz000 dddddddd
111 000  addr16      dddddddd dddddddd
111 001  addr32      dddddddd dddddddd dddddddd dddddddd
111 010  d16(PC)     dddddddd dddddddd
111 011  d8(PC,ix)   aiiiz000 dddddddd
111 100  imm/implicit

```

Flags:

```

0000 t   always or bra ipv btr
0001 f   never or bsr ipv bnv
0010 hi
0011 ls
0100 cc
0101 cs
0110 ne
0111 eq
1000 vc
1001 vs
1010 pl
1011 mi
1100 ge
1101 lt
1110 gt
1111 le

```