<ea> timings

Dn | An	0(0/0)		(An)|(An)+	4(1/0)	8(2/0)	-(An)	6(1/0)	10(2/0)
d(An|PC)	8(2/0)	12(3/0)	d(An|PC,ix)	10(2/0)	14(3/0)			
xxx.W	8(2/0)	12(3/0)	xxx.L	12(3/0)	16(4/0)	#xxx	4(1/0)	8(2/0)

Arithmetic instructions

	<ea>,An	<ea>,Dn	Dn,<M>		<ea>,An	<ea>,Dn	Dn,<M>
ADD.BW	8(1/0)	4(1/0)	8(1/1)	ADD.L	6|8(1/0)	6|8(1/0)	12(1/2)
SUB.BW	8(1/0)	4(1/0)	8(1/1)	SUB.L	6|8(1/0)	6|8(1/0)	12(1/2)
CMP.BW	6(1/0)	4(1/0)	-	CMP.L	6(1/0)	6(1/0)	-
AND.BW	-	4(1/0)	8(1/1)	AND.L	-	6(1/0)	12(1/2)
OR.BW	-	4(1/0)	8(1/1)	OR.L	-	6(1/0)	12(1/2)
EOR.BW Dn,Dn		4(1/0)	8(1/1)	EOR.BW Dn,Dn		8(1/0)	12(1/1)

MULS|MULU	38..70(1/0)	DIVS	142..158(1/0)	DIVU	126..140(1/0)

Immediate instructions

	#xxx,Dn	#xxx,An	#xxx,<M>		#xxx,Dn	#xxx,An	#xxx,<M>
ADDI.BW	8(2/0)	-	12(2/1)	ADDI.L	16(3/0)	-	20(3/2)
SUBI.BW	8(2/0)	-	12(2/1)	SUBI.L	16(3/0)	-	20(3/2)
ANDI.BW	8(2/0)	-	12(2/1)	ANDI.L	16(3/0)	-	20(3/2)
EORI.BW	8(2/0)	-	12(2/1)	EORI.L	16(3/0)	-	20(3/2)
ORI.BW	8(2/0)	-	12(2/1)	ORI.L	16(3/0)	-	20(3/2)
CMPI.BW	8(2/0)	-	8(2/0)	CMPI.L	14(3/0)	-	12(3/0)
ADDQ.BW	4(2/0)	8(1/0)	8(1/1)	ADDQ.L	8(1/0)	8(1/0)	12(1/2)
SUBQ.BW	4(2/0)	8(1/0)	8(1/1)	SUBQ.L	8(1/0)	8(1/0)	12(1/2)
MOVEQ(L)	4(2/0)	-	-				

Single operand instructions

	Rn	<M>		Rn	<M>		Rn	<M>
CLR.BW	4(1/0)	8(1/1)	NOT.BW	4(1/0)	8(1/1)	TST.BW	4(1/0)	4(1/0)
CLR.L	6(1/0)	12(1/2)	NOT.L	6(1/0)	12(1/2)	TST.L	4(1/0)	4(1/0)
NEG.BW	4(1/0)	8(1/1)	NEGX.BW	4(1/0)	8(1/1)	Scc.B f	4(1/0)	8(1/1)
NEG.L	6(1/0)	12(1/2)	NEGX.L	6(1/0)	12(1/2)	Scc.B t	6(1/0)	8(1/1)
TAS(B)	4(1/0)	10(1/1)	NBCD(B)	6(1/0)	8(1/0)			

Rotate instructions

	.BW Dx|#n,Dy	.BW <ea>	.L Dx|#n,Dy
ASL,ASR,LSL,LSR,ROL,ROR,ROXL,ROXR	6+2n(1/0)	8(1/1)	8+2n(1/0)

Bit manipulation instructions

	Dn,<M>	#n,<M>		Dn,Rn	#n,Rn
BCHG(B)	8(1/1)	12(2/1)	BCHG(L)	8(1/0)	12(2/0)
BCLR(B)	8(1/1)	12(2/1)	BCLR(L)	10(1/0)	14(2/0)
BSET(B)	8(1/1)	12(2/1)	BSET(L)	8(1/0)	12(2/0)
BTST(B)	4(1/0)	8(2/0)	BTST(L)	6(1/0)	10(2/0)

Relative branch instructions

	taken	not taken				taken	not taken
Bcc.B	10(2/0)	8(1/0)	BSR.B	18(2/2)	DBcc(true)	-	12(2/0)
Bcc.W	10(2/0)	12(1/0)	BSR.W	18(2/2)	DBcc(false)	10(2/0)	14(3/0)

Multi precision instructions

	Dn,Dn	-(An),(-An)		Dn,Dn	-(An),(-An)
ADD|SUBX.BW	4(1/0)	18(3/1)	ADD|SUBX.L	8(1/0)	30(5/2)
ABCD|SBCD(B)	6(1/0)	18(3/1)			
CMPM.BW (An)+,(An)+		12(3/0)	CMPM.L (An)+,(An)+		20(5/0)

Other instructions

	(An)	(An)+	-(An)	d(An|PC)	d(An|PC,ix)	xxx.W	xxx.L
JMP <ea>	8(2/0)	-	-	10(2/0)	14(3/0)	10(2/0)	12(3/0)
JSR <ea>	16(2/2)	-	-	18(2/2)	22(2/2)	18(2/2)	20(3/2)
LEA <ea>,An	4(1/0)	-	-	8(2/0)	12(2/0)	8(2/0)	12(2/0)
PEA <ea>	12(1/2)	-	-	16(2/2)	20(2/2)	16(2/2)	20(2/2)
MOVEM.W <ea>,Rl	12+4n(3+n)	12+4n(3+n)	-	16+4n(4+n)	18+4n(4+n)	16+4n(4+n)	20+4n(4+n)
MOVEM.L <ea>,Rl	12+8n(3+2n)	16+8n(3+2n)	-	16+8n(4+2n)	18+8n(4+2n)	16+8n(4+2n)	20+8n(5+2n)
MOVEM.W Rl,<ea>	8+4n(2/n)	-	8+4n(2/n)	12+4n(3/n)	14+4n(3/n)	12+4n(3/n)	16+4n(4/n)
MOVEM.L Rl,<ea>	8+8n(2/2n)	-	8+8n(2/2n)	12+8n(3/2n)	14+8n(3/2n)	12+8n(3/2n)	16+8n(4/2n)

MOVEP.W Dn,(d16,An)	16(2/2)	MOVEP.L Dn,(d16,An)	24(2/4)			
MOVEP.W (d16,An),Dn	16(4/0)	MOVEP.L (d16,An),Dn	24(6/0)			
ANDI|EORI|ORI #n,CCR	20(3/0)	MOVE <ea>,CCR	12(1/0)	MOVE USP,An	4(1/0)	MOVE An,USP 4(1/0)
ANDI[EORI|ORI #n,SR	20(3/0)	MOVE <ea>,SR	12(1/0)	MOVE SR,Rn	6(1/0)	MOVE SR,<M> 8(1/1)

CHK <ea>,Dn 10(1/0)	EXG Rn,Rn	6(1/0)	EXT.WL Dn	4(1/0)	LINK An,#n	16(2/2)	NOP 4(1/0)
RTE|RTR 20(5/0)	RTS	16(4/0)	STOP	4(0/0)	SWAP Dn	4(1/0)	UNLK 12(3/0)
RESET 132(1/0)							

Exceptions

Exception	Cycles	Exception	Cycles	Exception	Cycles
Address error	50(4/7)	Bus error	50(4/7)	CHK	44(5/3)
Divide by Zero	42(5/3)	Illegal instruction	34(4/3)	Interrupt	44(5/3)
Privilege violation	34(4/3)	Reset	40(6/0)	Trace	34(4/3)
TRAP #n	38(4/3)	TRAPV (trap not taken)	4(1/0)	TRAPV (trap taken)	34(4/3)

68000 instruction timing cheat sheet

Move Byte and Word Instruction Execution Times

	Dn	An	(An)	(An)+	-(An)	d(An)	d(An,ix)	xxx.W	|xxx.L
Dn	4(1/0)	4(1/0)	8(1/1)	8(1/1)	8(1/1)	12(2/1)	14(2/1)	12(2/1)	16(3/1)
An	4(1/0)	4(1/0)	8(1/1)	8(1/1)	8(1/1)	12(2/1)	14(2/1)	12(2/1)	16(3/1)
(An)	8(2/0)	8(2/0)	12(2/1)	12(2/1)	12(2/1)	16(3/1)	18(3/1)	16(3/1)	20(4/1)
(An)+	8(2/0)	8(2/0)	12(2/1)	12(2/1)	12(2/1)	16(3/1)	18(3/1)	16(3/1)	20(4/1)
-(An)	10(2/0)	10(2/0)	14(2/1)	14(2/1)	14(2/1)	18(3/1)	20(4/1)	18(3/1)	22(4/1)
d(An)	12(3/0)	12(3/0)	16(3/1)	16(3/1)	16(3/1)	20(4/1)	22(4/1)	20(4/1)	24(5/1)
d(An,ix)	14(3/0)	14(3/0)	18(3/1)	18(3/1)	18(3/1)	22(4/1)	24(4/1)	22(4/1)	26(5/1)
xxx.W	12(3/0)	12(3/0)	16(3/1)	16(3/1)	16(3/1)	20(4/1)	22(4/1)	20(4/1)	24(5/1)
xxx.L	16(4/0)	16(4/0)	20(4/1)	20(4/1)	20(4/1)	24(5/1)	26(5/1)	24(5/1)	28(6/1)
d(PC)	12(3/0)	12(3/0)	16(3/1)	16(3/1)	16(3/1)	20(4/1)	22(4/1)	20(4/1)	24(5/1)
d(PC,ix)	14(3/0)	14(3/0)	18(3/1)	18(3/1)	18(3/1)	22(4/1)	24(4/1)	22(4/1)	26(5/1)
#xxx	8(2/0)	8(2/0)	12(2/1)	12(2/1)	12(2/1)	16(3/1)	18(3/1)	16(3/1)	20(4/1)

The size of the index register (ix) does not affect execution time

Move Long Instruction Execution Times

	Dn	An	(An)	(An)+	-(An)	d(An)	d(An,ix)	xxx.W	|xxx.L
	4(1/0)		8(1/1)						
Dn	4(1/0)	4(1/0)	12(1/2)	12(1/2)	12(1/2)	16(2/2)	18(2/2)	16(2/2)	20(3/2)
An	4(1/0)	4(1/0)	12(1/2)	12(1/2)	12(1/2)	16(2/2)	18(2/2)	16(2/2)	20(3/2)
(An)	12(3/0)	12(3/0)	20(3/2)	20(3/2)	20(3/2)	24(4/2)	26(4/2)	24(4/2)	28(5/2)
(An)+	12(3/0)	12(3/0)	20(3/2)	20(3/2)	20(3/2)	24(4/2)	26(4/2)	24(4/2)	28(5/2)
-(An)	14(3/0)	14(3/0)	22(3/2)	22(3/2)	22(3/2)	26(4/2)	28(4/2)	26(4/2)	30(5/2)
d(An)	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	28(5/2)	30(5/2)	28(5/2)	32(6/2)
d(An,ix)	18(4/0)	18(4/0)	26(4/2)	26(4/2)	26(4/2)	30(5/2)	32(5/2)	30(5/2)	34(6/2)
xxx.W	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	28(5/2)	30(5/2)	28(5/2)	32(6/2)
xxx.L	20(5/0)	20(5/0)	28(5/2)	28(5/2)	28(5/2)	32(6/2)	34(6/2)	32(6/2)	36(7/2)
d(PC)	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	28(5/2)	30(5/2)	28(5/2)	32(5/2)
d(PC,ix)	18(4/0)	18(4/0)	26(4/2)	26(4/2)	26(4/2)	30(5/2)	32(5/2)	30(5/2)	34(6/2)
#xxx	12(3/0)	12(3/0)	20(3/2)	20(3/2)	20(3/2)	24(4/2)	26(4/2)	24(4/2)	28(5/2)

The size of the index register (ix) does not affect execution time