TABLE OF CONTENTS (Continued)

Paragraph Number	Title	Page Number
8.1.1	Coprocessor ID Field	8-1
8.1.2	Effective Address Field	8-1
8.1.3	Register/Memory Field	8-1
8.1.4	Source Specifier Field	8-1
8.1.5	Destination Register Field	8-2
8.1.6	Conditional Predicate Field	8-2
8.1.7	Shift and Rotate Instructions	8-2
8.1.7.1	Count Register Field	8-2
8.1.7.2	Register Field	8-2
8.1.8	Size Field	8-4
8.1.9	Opmode Field	8-4
8.1.10	Address/Data Field	8-4
8.2	Operation Code Map	8-4
	Appendix A Processor Instruction Summary	
A.1	MC68000, MC68008, MC68010 Processors	A-12
A.1.1	M68000, MC68008, and MC68010 Instruction Set	A-12
A.1.2	MC68000, MC68008, and MC68010 Addressing Modes	A-16
A.2	MC68020 Processors	A-17
A.2.1	MC68020 Instruction Set	A-17
A.2.2	MC68020 Addressing Modes	A-20
A.3	MC68030 Processors	A-21
A.3.1	MC68030 Instruction Set	A-21
A.3.2	MC68030 Addressing Modes	A-24
A.4	MC68040 Processors	A-25
A.4.1	MC68040 Instruction Set	A-25
A.4.2	MC68040 Addressing Modes	A-29
A.5	MC68881/MC68882 Coprocessors	A-30
A.5.1	MC68881/MC68882 Instruction Set	A-30
A.5.2	MC68881/MC68882 Addressing Modes	A-31
A.6	MC68851 Coprocessors	A-31
A.6.1	MC68851 Instruction Set	A-31
A.6.2	MC68851 Addressing Modes	A-31
	Appendix B Exception Processing Reference	

Figure Number	Title	Page

AC	=	Access Control Register
CAL	=	Current Access Level Register
CRP	=	CPU Root Pointer
DRP	=	DMA Root Pointer
PCSR	=	PMMU Control Register
PMMUSR	=	Paged Memory Management Unit Status Register
MMUSR	=	Memory Management Unit Status Register
SCC	=	Stack Change Control Register
SRP	=	Supervisor Root Pointer Register
TC	=	Translation Control Register
URP	=	User Root Pointer

Table 1-5. Double-Precision Real Format Summary

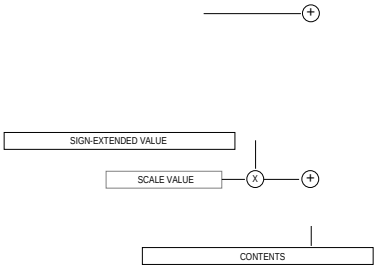
Data Format	Field Size (in Bits)
Sign (s)	1
Biased Exponent (e)	11
Fraction (f)	52
Total	64

2.2.14 Program Counter Memory Indirect Postindexed Mode

This mode is similar to the mode described in **2.2.9 Memory Indirect Postindexed Mode**, but the PC is the base register. Both the operand and operand address are in memory. The processor calculates an intermediate indirect memory address by adding a base displacement to the PC contents. The processor accesses a long word at that address and adds the scaled contents of the index register and the optional outer displacement to yield the effective address. The value of the PC used in the calculation is the address of the first extension word. This is a program reference allowed only for reads.

In the syntax for this mode, brackets enclose the values used to calculate the intermediate memory address. All four user-specified values are optional. The user must supply the assembler notation ZPC (a zero value PC) to show the PC is not used. This allows the user to access the program space without using the PC in calculating the effective address. Both the base and outer displacements may be null, word, or long word. When omitting a displacement or suppressing an element, its value is zero in the effective address calculation.

EA = (bd + PC) + Xn.SIZE*SCALE + od
([bd,PC],Xn.SIZE*SCALE,od)
111
011
1,2,3,4, or 5



Addressing Capabilities

Addressing Capabilities

NOTE: A register list includes any combination of the eight floating-point data registers or any combination of three control registers (FPCR, FPSR, and FPIAR). If a register list mask resides in a data register, only floating-point data registers may be specified.

functions, software supports remainder and integer part; the FPU also supports the nontranscendental operations of absolute value, negate, and test.

ANDI to CCR

CCR AND Immediate
(M68000 Family)

ANDI to CCR

Operation: Source Λ CCR CCR

Assembler Syntax: ANDI # < data > ,CCR

Attributes: Size = (Byte)

Description: Performs an AND operation of the immediate operand with the condition codes and stores the result in the low-order byte of the status register.

Condition Codes:

BFCLR Test Bit Field and Clear (MC68020, MC68030, MC68040) BFCLR

CAS
CAS2

Compare and Swap with Operand
(MC68020, MC68030, MC68040)

CAS
CAS2

Instruction Format:

CAS2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	SIZE		0	1	1	1	1	1	1	0	0
D/A1		Rn1		0	0	0		Du1		0	0	0		Dc1	
D/A2		Rn2		0	0	0		Du2		0	0	0		Dc2	

Instruction Fields:

Size field—Specifies the size of the operation.
- 10 — Word operation
- 11 — Long operation

D/A1, D/A2 fields—Specify whether Rn1 and Rn2 reference data or address registers, respectively.
- 0 — The corresponding register is a data register.
- 1 — The corresponding register is an address register.

Rn1, Rn2 fields—Specify the numbers of the registers that contain the addresses of the first and second memory operands, respectively. If the operands overlap in memory, the results of any memory update are undefined.

Du1, Du2 fields—Specify the data registers that contain the update values to be written to the first and second memory operand locations if the comparison is successful.

Dc1, Dc2 fields—Specify the data registers that contain the test values to be compared to the first and second memory operands, respectively. If Dc1 and Dc2 specify the same data register and the comparison fails, memory operand 1 is stored in the data register.

NOTE

The CAS and CAS2 instructions can be used to perform secure update operations on system control data structures in a multiprocessing environment.

In the MC68040 if the operands are not equal, the destination or destination 1 operand is written back to memory to complete the locked access for CAS or CAS2, respectively.

CMP Compare (M68000 Family) CMP

Effective Address field—Specifies the source operand. All addressing modes can be used as listed in the following tables:

*Word and Long only.
**Can be used with CPU32.

NOTE

DIVU, DIVUL

Unsigned Divide
(M68000 Family)

DIVU, DIVUL

Operation: Destination Source Destination

Assembler Syntax:

DIVU.W < ea > ,Dn	32/16	16r – 16q
*DIVU.L < ea > ,Dq	32/32	32q
*DIVU.L < ea > ,Dr:Dq	64/32	32r – 32q
*DIVUL.L < ea > ,Dr:Dq	32/32	32r – 32q

*Applies to MC68020, MC68030, MC68040, CPU32 only.

Attributes:

EOR

Exclusive-OR Logical
(M68000 Family)

EOR

Operation:

EORI to CCR

Exclusive-OR Immediate to Condition Code (M68000 Family)

EORI to CCR

JMP

Jump
(M68000 Family)

JMP

Operation: Destination Address PC

Assembler Syntax: JMP < ea >

Attributes: Unsized

MOVE Move Data from Source to Destination MOVE

MOVEA

Move Address (M68000 Family)

MOVEA

Effective Address field—Specifies the location of the source operand. All addressing modes can be used as listed in the following tables:

Addressing Mode	Mode	Register	Addressing Mode	Mode	Register
Dn	000	reg. number:Dn	(xxx).W	111	000
An	001	reg. number:An	(xxx).L	111	001
(An)	010	reg. number:An	#<data>	111	100

*Can be used with CPU32.

MOVEP Move Peripheral Data MOVEP

MULU — Unsigned Multiply (M68000 Family) — MULU

Instruction Format:

LONG

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	0	0	0	0	EFFECTIVE ADDRESS					
											MODE		REGISTER		
0	REGISTER Dl			0	SIZE	0	0	0	0	0	0	0	REGISTER Dh		

Instruction Fields:

Effective Address field—Specifies the source operand. Only data addressing modes can be used as listed in the following tables:

Addressing Mode	Mode	Register	Addressing Mode	Mode	Register
Dn	000	reg. number:Dn	(xxx).W	111	000
An	—	—	(xxx).L	111	001
(An)	010	reg. number:An	#<data>	111	100
(An) +	011	reg. number:An			
– (An)	100	reg. number:An			
(d_{16},An)	101	reg. number:An	(d_{16},PC)	111	010
(d_8					

*Can be used with CPU32.

Register Dl field—Specifies a data register for the destination operand. The 32-bit multiplicand comes from this register, and the low-order 32 bits of the product are loaded into this register.

Size field—Selects a 32- or 64-bit product.
0 — 32-bit product to be returned to register Dl.
1 — 64-bit product to be returned to Dh – Dl.

Register Dh field—If size is one, specifies the data register into which the high-order 32 bits of the product are loaded. If Dh = Dl and size is one, the results of the operation are undefined. Otherwise, this field is unused.

NOT — Logical Complement (M68000 Family) — NOT

Operation: ~ Destination → Destination

Assembler Syntax: NOT < ea >

Attributes: Size = (Byte, Word, Long)

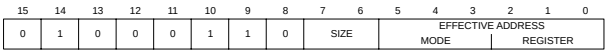
Description: Calculates the ones complement of the destination operand and stores the result in the destination location. The size of the operation is specified as byte, word, or long.

Condition Codes:

X	N	Z	V	C
—	*	*	0	0

X — Not affected.
N — Set if the result is negative; cleared otherwise.
Z — Set if the result is zero; cleared otherwise.
V — Always cleared.
C — Always cleared.

Instruction Format:



NOT — Logical Complement (M68000 Family) — NOT

Instruction Fields:

Size field—Specifies the size of the operation.
00— Byte operation
01— Word operation
10— Long operation

Effective Address field—Specifies the destination operand. Only data alterable addressing modes can be used as listed in the following tables:

PACK Pack PACK

TRAP

Trap
(M68000 Family)

TRAP

Operation: 1

UNPK Unpack BCD UNPK

Table 5-1. Directly Supported Floating-Point Instructions

Mnemonic	Description
FABS	Floating-Point Absolute Value
FADD	Floating-Point Add
FBcc	Floating-Point Branch Conditionally
FCMP	Floating-Point Compare
FDBcc	Floating-Point Test Condition, Decrement, and Branch
FDIV	Floating-Point Divide
FMOVE	Move Floating-Point Data Register
FMOVE	Move Floating-Point System Control Register
FMOVEM	Move Multiple Floating-Point System Data Register
FMOVEM	Move Multiple Floating-Point Control Data Register
FMUL	Floating-Point Multiply
FNEG	Floating-Point Negate
FNOP	No Operation
FRESTORE*	Restore Internal Floating-Point State*
FSAVE*	Save Internal Floating-Point State*
FScc	Set According to Floating-Point Condition
FSORT	Floating-Point Square Root
FSUB	Floating-Point Subtract
FSGLDIV	Floating-Point Single-Precision Divide
FSFLMUL	Floating-Point Single-Precision Multiply
FTRAPcc	Trap on Floating-Point Condition
FTST	Test Floating-Point Operand

*These are privileged instructions; refer to **Section 6 Supervisor (Privaleged) Instructions** for detailed information.

FASIN

Arc Sine
(MC6888X, M68040FPSP)

FASIN

FATAN

Arc Tangent
(MC6888X, M68040FPSP)

FATAN

Instruction Format:

Instruction Fields:

Coprocessor ID field—Specifies which coprocessor in the system is to execute this instruction. Motorola assemblers default to ID = 1 for the floating-point coprocessor.

FATANH Hyperbolic Arc Tangent (MC6888X, M68040FPSP) FATANH

FDBcc Floating-Point Test Condition, Decrement, and Branch FDBcc

FETOX e^x

(MC6888X, M68040FPSP)

R/M field—Specifies the source operand address mode.
0 — The operation is register to register.
1 — The operation is < ea > to register.

Source Specifier Field—Specifies the source register or data format.
If R/M = 0, specifies the source floating-point data register.
If R/M = 1, specifies the source data format:
000 — Long-Word Integer (L)
001 — Single-Precision Real (S)
010 — Extended-Precision Real (X)
011 — Packed-Decimal Real (P)*
100 — Word Integer (W)
101 — Double-Precision Real (D)
110 — Byte Integer (B)

Destination Register field—Specifies the destination floating- point data register. If R/M = 0 and the source and destination fields are equal, then the input operand is taken from the specified floating-point data register, and the result is written into the same register. If the single register syntax is used, Motorola assemblers set the source and destination fields to the same value.

FETOXM1 e

FLOG2 Log_2 (MC6888X, M68040FPSP) FLOG2

Instruction Format:

Instruction Fields:

Coprocessor ID field—Specifies which coprocessor in the system is to execute this

FMOVEM Move Multiple Floating-Point Data Registers (MC6888X, MC68040) FMOVEM

An example of the calculation of this mask is as follows:

Nested calling sequence...

Upon return from a procedure, the restoration of the necessary registers follows the same convention, and the register mask generated during the save operation on entry is used to restore the required floating-point data registers:

FMUL

Floating-Point Multiply
(MC6888X, MC68040)

FMUL

Operation: Source x FPn FPn

Assembler Syntax:
FMUL. < fmt > < ea > ,FPn
FMUL.X FPm,FPn
*FrMUL < fmt > < ea > ,FPn
*FrMUL.X FPm,FPn
where r is rounding precision, S or D
*Supported by MC68040 only

Attributes: Format = (Byte, Word, Long, Single, Double, Extended, Packed)

Description: Converts the source operand to extended precision (if necessary) and multiplies that number by the number in the destination floating-point data register. Stores the result in the destination floating-point data register.

FMUL will round the result to the precision selected in the floating-point control register. FSMUL and FDMUL will round the result to single or double precision, respectively, regardless of the rounding precision selected in the floating-point control register.

FREM IEEE Remainder (MC6888X, M68040FPSP) FREM

Effective Address field—Determines the addressing mode for external operands.
*193. 1 3 <75he 29EDes25M000 rbd,Av,Xv)B—47e1Prec132ti11[(EDespec733(000)-6994(r[bd,Av,Xv],od)B)3004te1d-Pt r17(—)POr0 adetAveC3910r[bd,PC,Xv],od)B)29r17pec733(000)-6994(r 6St

FTWOTOX 2^x (MC6888X, M68040FPSP) FTWOTOX

Operation: 2^{Source}

FSAVE

MOVE from SR

Move from the Status Register (MC68EC000, MC68010, MC68020, MC68030, MC68040, CPU32)

MOVE from SR

Instruction Field:

Effective Address field—Specifies the destination location. Only data alterable addressing modes can be used as listed in the following tables:

*Available for the CPU32.

NOTE

Use the MOVE from CCR instruction to access only the condition codes.

PFLUSH PFLUSHA PFLUSHS — Invalidate Entries in the ATC (MC68851)

PFLUSH PFLUSHA PFLUSHS

Operation: If Supervisor State
Then Address Translation Cache Entries For Destination Address

PMOVE Move PMMU Register (MC68851) PMOVE

PMMU Status Register: Not affected unless the PMMU status register is written to by the instruction.

Instruction Format 1:

PMOVE to/from TC, CRP, DRP, SRP, CAL, VAL, SCC, AC

Instruction Fields:

Effective Address field—for memory-to-register transfers, any addressing mode is allowed as listed in the following table:

*PMOVE to CRP, SRP, and DMA root pointer not allowed with these modes

PScc — Set on PMMU Condition (MC68851) — PScc

Instruction Format:

Instruction Fields:

Effective Address field—Specifies the destination location. Only data alterable addressing modes can be used as listed in the following table:

Addressing Mode	Mode	Register	Addressing Mode	Mode	Register
Dn	000	reg. number:Dn	(xxx).W	111	000
An	—	—	(xxx).L	111	001
(An)	010	reg. number:An	# < data >	—	—
(An) +	011	reg. number:An			
—(An)	100	reg. number:An			
(d_{16},An)	101	reg. number:An	(d_{16},PC),AU8		

MC68851 Condition field—Specifies the coprocessor condition to be tested. This field is passed to the MC68851, which provides directives to the main processor for processing this instruction.

PTEST Test a Logical Address (MC68030 only) PTEST

Instruction Format:

Instruction Fields:

Effective Address field—Specifies the logical address to be tested. Only control alterable addressing modes can be used as listed in the following table:

Level field—Specifies the highest numbered level to be searched in the table. When this field contains 0, the A field and the register field must also be 0. The instruction takes an F-line exception when the level field is 0 and the A field is not 0.

R/W field—Specifies simulating a read or write bus cycle (no difference for MC68030 MMU).
0—Write
1—Read

A field—Specifies the address register option.
0—No address register.

Supervisor (Privileged) Instructions

Supervisor (Privileged) Instructions

TBLS TBLSN — Table Lookup and Interpolate (Signed) (CPU32)

If R = 1 (TABLENR), the result is returned in register Dx without rounding. If the size is byte, the integer portion of the result is returned in Dx 15 – 8; the integer portion of a word result is stored in Dx 23 – 8; the least significant 24 bits of a long result are stored in Dx 31 – 8. Byte and word results are sign-extended to fill the entire 32-bit register.

	31 – 24	23 – 16	15 – 8	7 – 0
BYTE	SIGN-EXTENDED	SIGN-EXTENDED	RESULT	FRACTION
WORD	SIGN-EXTENDED	RESULT	RESULT	FRACTION
LONG	RESULT	RESULT	RESULT	FRACTION

NOTE

The long-word result contains only the least significant 24 bits of integer precision.

For all sizes, the 8-bit fractional portion of the result is returned to the low byte of the data register, Dx 7 – 0. User software can make use of the fractional data to reduce cumulative errors in lengthy calculations or implement rounding algorithms different from that provided by other forms of TBLS. The previously described assumed radix point places two restrictions on the programmer:

1. Tables are limited to 257 entries in length.
2. Interpolation resolution is limited to 1/256, the distance between consecutive table entries. The assumed radix point should not, however, be construed by the programmer as a requirement that the independent variable be calculated as a fractional number in the range 0 < < 255. On the contrary, X should be considered an integer in the range 0 < < 65535, realizing that the table is actually a compressed representation of a linearized function in which only every 256th value is actually stored in memory.

TBLS TBLSN — Table Lookup and Interpolate (Signed) (CPU32)

Condition Codes: CXR—-2750(No arf)7(Rfeced)]TJ0 -1.567 TD[(CN)-401R—-2750(Se is the fmot sign

SECTION 8
INSTRUCTION FORMAT SUMMARY

This section contains a listing of the M68000 family instructions in binary format. It is listed in opcode order for the M68000 family instruction set.

8.1 INSTRUCTION FORMAT

i1 i f67i1 707 334.8999

APPENDIX A
PROCESSOR INSTRUCTION SUMMARY

Table A-2 lists the M68000 family instructions by mnemonics, followed by the descriptive name.**name.**

A.2.2 MC68020 Addressing Modes

The MC68020 supports 18 addressing modes as shown in Table A-7.

Table A-7. MC68020 Data Addressing Modes

A.3.2 MC68030 Addressing Modes

The MC68030 supports 18 addressing modes as shown in Table A-9.

Table A-9. MC68030 Data Addressing Modes

Addressing Modes	Syntax
Register Direct	
Data Register Direct	Dn
Address Register Direct	An
Register Indirect	
Address Register Indirect	(An)
Address Register Indirect with Postincrement	(An)+
Address Register Indirect with Predecrement	–(An)
Address Register Indirect with Displacement	(d_{16},An)
Register Indirect with Index	
Address Register Indirect with Index (8-Bit Displacement)	(d_8,An,Xn)
Address Register Indirect with Index (Base Displacement)	(bd,An,Xn)
Memory Indirect	
Memory Indirect Postindexed	([bd,An],Xn,od)
Memory Indirect Preindexed	([bd,An,Xn],od)
Program Counter Indirect with Displacement	(d_{16},PC)
Program Counter Indirect with Index	
PC Indirect with Index (8-Bit Displacement)	(d_8,PC,Xn)
PC Indirect with Index (Base Displacement)	(bd,PC,Xn)
Program Counter Memory Indirect	
PC Memory Indirect Postindexed	([bd,PC],Xn,od)
PC Memory Indirect Preindexed	([bd,PC,Xn],od)
Absolute	
Absolute Short	(xxx).W
Absolute Long	(xxx).L
Immediate	#<data>

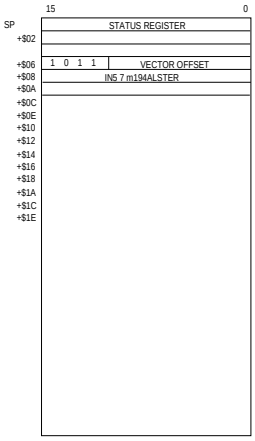


Figure B-12. MC68020 and MC68030 Long Bus Cycle Stack Frame, Format $B

Figure B-13. CPU32 Bus Error for Prefetches and Operands Stack Frame, Format $C