

The revision list can be viewed directly by clicking the title page.

The revision list summarizes the locations of revisions and additions. Details should always be checked by referring to the relevant text.

32

H8SX Family

Software Manual

Renesas 32-Bit CISC Microcomputer
H8SX Family

Software Manual

Rev.4.00
Revision Date: Sep. 18, 2008

RenesasTechnology
www.renesas.com

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (<http://www.renesas.com>)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
 - (1) artificial life support devices or systems
 - (2) surgical implantations
 - (3) healthcare intervention (e.g., excision, administration of medication, etc.)
 - (4) any other purposes that pose a direct threat to human lifeRenesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

Configuration of This Manual

This manual comprises the following items:

1. Configuration of This Manual
2. Preface
3. Contents
5. Description of CPU
6. Description of Instructions

The configuration of the description of each instruction differs according to the instruction. However, the generic style includes the following items:

- i) Mnemonic
 - ii) Full Name
 - iii) Type
 - iv) Assembly-Language Format
 - v) Operation
 - vi) Operand Size
 - vii) Description
 - viii) Condition Code
 - ix) Available Registers
 - x) Available Addressing Modes
 - xi) Notes
6. Processing States
 7. Basic Timing

Preface

The H8SX CPU is a high-speed CPU with an internal 32-bit architecture that is upward-compatible with the H8/300, H8/300H, and H8S CPUs. The H8SX CPU has sixteen 16-bit general registers, can handle a 4-Gbyte linear address space, and is ideal for a realtime control system.

Target Users: This manual was written for users who will be using the H8SX Series in the design of application systems. Target users are expected to understand the fundamentals of microcomputers.

Objective: This manual was written to explain the instructions of the H8SX CPU to the target users.
Refer to the corresponding hardware manual for a detailed description of the hardware.

Notes on Reading This Manual:

- In order to understand the H8SX CPU
Read the manual according to the contents. This manual can be roughly categorized into parts on the CPU, description of each instruction, processing states, and basic timing.
- In order to understand the details of the instructions
Read section 2, Instruction Descriptions. The generic style includes the following items: mnemonic, full name, type, assembly-language format, operation, operand size, description, condition code, available registers, available addressing modes, and notes.

All trademarks and registered trademarks are the property of their respective owners.

Contents

Section 1	CPU	1
1.1	Features	1
1.2	CPU Operating Modes	3
1.2.1	Normal Mode	3
1.2.2	Middle Mode	5
1.2.3	Advanced Mode	6
1.2.4	Maximum Mode	7
1.3	Instruction Fetch	9
1.4	Address Space	9
1.5	Registers	10
1.5.1	General Registers	12
1.5.2	Program Counter (PC)	13
1.5.3	Condition-Code Register (CCR)	13
1.5.4	Extended Control Register (EXR)	15
1.5.5	Vector Base Register (VBR)	15
1.5.6	Short Address Base Register (SBR)	15
1.5.7	Multiply-Accumulate Register (MAC)	15
1.5.8	Initial Values of CPU Registers	16
1.6	Data Formats	16
1.6.1	General Register Data Formats	16
1.6.2	Memory Data Formats	18
1.7	Instruction Set	19
1.7.1	Instructions and Addressing Modes	20
1.7.2	Table of Instructions Classified by Function	25
1.7.3	Basic Instruction Formats	36
1.8	Addressing Modes and Effective Address Calculation	37
1.8.1	Register Direct—Rn	37
1.8.2	Register Indirect—@ERn	38
1.8.3	Register Indirect with Displacement—@(d:2, ERn), @(d:16, ERn), or @(d:32, ERn)	38
1.8.4	Index Register Indirect with Displacement—@(d:16,RnL.B), @(d:32,RnL.B), @(d:16,Rn.W), @(d:32,Rn.W), @(d:16,ERn.L), or @(d:32,ERn.L)	38
1.8.5	Register Indirect with Post-Increment, Pre-Decrement, Pre-Increment, or Post-Decrement—@ERn+, @-ERn, @+ERn, or @ERn-	38
1.8.6	Absolute Address—@aa:8, @aa:16, @aa:24, or @aa:32	40

1.8.7	Immediate—#xx	41
1.8.8	Program-Counter Relative—@(d:8, PC) or @(d:16, PC):	41
1.8.9	Program-Counter Relative with Index Register—@(RnL.B, PC), @(Rn.W, PC), or @(ERn.L, PC).....	42
1.8.10	Memory Indirect—@@aa:8	42
1.8.11	Extended Memory Indirect—@@vec:7	43
1.8.12	Effective Address Calculation	43
1.8.13	MOVA Instruction.....	45
Section 2 Instruction Descriptions.....		47
2.1	Tables and Symbols	47
2.1.1	Assembly-Language Format	48
2.1.2	Operation	49
2.1.3	Condition Code	50
2.1.4	Instruction Format	50
2.1.5	Register Specification	51
2.1.6	Bit Data Access in Bit Manipulation Instructions.....	52
2.2	Instruction Descriptions.....	53
2.2.1	ADD ADD Binary.....	53
2.2.2	ADDS ADD with Sign extension.....	56
2.2.3	ADDX ADD with eXtend carry	58
2.2.4	AND AND logical.....	60
2.2.5	ANDC AND Control register	63
2.2.6	ANDC AND Control register	64
2.2.7	BAND Bit AND.....	65
2.2.8	Bcc Branch conditionally.....	67
2.2.9	BCLR Bit CLear	69
2.2.10	BCLR/EQ Bit CLear if EQUAL	71
2.2.11	BCLR/NE Bit CLear if Not Equal	73
2.2.12	BFLD Bit Field LoAD	75
2.2.13	BFST Bit Field STore	77
2.2.14	BIAND Bit Invert AND	79
2.2.15	BILD Bit Invert LoAD.....	81
2.2.16	BIOR Bit Invert inclusive OR.....	83
2.2.17	BIST Bit Invert STore	85
2.2.18	BISTZ Bit Invert STore Zero flag	87
2.2.19	BIXOR Bit Invert eXclusive OR	89
2.2.20	BLD Bit LoAD	91
2.2.21	BNOT Bit NOT	93
2.2.22	BOR Bit inclusive OR	95

2.2.23	BRA	BRANch always	97
2.2.24	BRA/S	BRANch always with delay Slot	99
2.2.25	BRA/BC	BRANch if Bit Cleared.....	100
2.2.26	BRA/BS	BRANch if Bit Set.....	102
2.2.27	BSET	Bit SET	104
2.2.28	BSET/EQ	Bit SET if EQual.....	106
2.2.29	BSET/NE	Bit SET if Not Equal.....	108
2.2.30	BSR	Branch to SubRoutine.....	110
2.2.31	BSR/BC	Branch to SubRoutine if Bit Cleared	112
2.2.32	BSR/BS	Branch to SubRoutine if Bit Set.....	114
2.2.33	BST	Bit STore.....	116
2.2.34	BSTZ	Bit STore Zero flag.....	118
2.2.35	BTST	Bit TeST.....	120
2.2.36	BXOR	Bit eXclusive OR.....	122
2.2.37	CLRMAC	CLeaR MAC register	124
2.2.38	CMP	CoMPare.....	125
2.2.39	DAA	Decimal Adjust Add	127
2.2.40	DAS	Decimal Adjust Subtract.....	129
2.2.41	DEC (B)	DECrement	131
2.2.42	DEC (W)	DECrement	132
2.2.43	DEC (L)	DECrement	134
2.2.44	DIVS	DIVide as Signed.....	136
2.2.45	DIVU	DIVide as Unsigned.....	138
2.2.46	DIVXS	DIVide eXtend as Signed.....	140
2.2.47	DIVXU	DIVide eXtend as Unsigned	142
2.2.48	EEPMOV (B)	MOVE data to EEPROM	144
2.2.49	EEPMOV (W)	MOVE data to EEPROM	145
2.2.50	EXTS	EXTend as Signed	147
2.2.51	EXTU	EXTend as Unsigned	149
2.2.52	INC (B)	INCrement	151
2.2.53	INC (W)	INCrement	153
2.2.54	INC (L)	INCrement	155
2.2.55	JMP	JuMP.....	157
2.2.56	JSR	Jump to SubRoutine.....	159
2.2.57	LDC (B)	LoaD to Control register	161
2.2.58	LDC (B)	LoaD to Control register	163
2.2.59	LDC (W)	LoaD to Control register	165
2.2.60	LDC (W)	LoaD to Control register	167
2.2.61	LDC (L)	Load to Control register.....	169
2.2.62	LDM	LoaD to Multiple registers.....	170

2.2.63	LDMAC	LoaD to MAC register	172
2.2.64	MAC	Multiply and ACcumulate	174
2.2.65	MOV	MOVE data	178
2.2.66	MOVA	MOVE effective Address	180
2.2.67	MOVFP	MOVE From Peripheral with E clock	183
2.2.68	MOVMD (B)	MOVE Multiple Data	185
2.2.69	MOVMD (W)	MOVE Multiple Data	186
2.2.70	MOVMD (L)	MOVE Multiple Data	187
2.2.71	MOVSD	MOVE String Data	188
2.2.72	MOVTPE	MOVE To Peripheral with E clock	190
2.2.73	MULS	MULTiply as Signed	192
2.2.74	MULS/U	MULTiply as Signed	194
2.2.75	MULU	MULTiply as Unsigned	196
2.2.76	MULU/U	MULTiply as Unsigned	198
2.2.77	MULXS	MULTiply eXtend as Signed	200
2.2.78	MULXU	MULTiply eXtend as Unsigned	202
2.2.79	NEG	NEGate	204
2.2.80	NOP	No OPERATION	206
2.2.81	NOT	NOT = logical complement	207
2.2.82	OR	inclusive OR logical	209
2.2.83	ORC	inclusive OR Control register	211
2.2.84	ORC	inclusive OR Control register	212
2.2.85	POP (W)	POP data	213
2.2.86	POP (L)	POP data	215
2.2.87	PUSH (W)	PUSH data	216
2.2.88	PUSH (L)	PUSH data	218
2.2.89	ROTL	ROTate Left	220
2.2.90	ROTR	ROTate Right	222
2.2.91	ROTXL	ROTate with eXtend carry Left	224
2.2.92	ROTXR	ROTate with eXtend carry Right	226
2.2.93	RTE	ReTurn from Exception	228
2.2.94	RTE/L	Load to registers and ReTurn from Exception	229
2.2.95	RTS	ReTurn from Subroutine	231
2.2.96	RTS/L	Load to registers and ReTurn from Subroutine	232
2.2.97	SHAL	SHift Arithmetic Left	234
2.2.98	SHAR	SHift Arithmetic Right	236
2.2.99	SHLL (1)	SHift Logical Left	238
2.2.100	SHLL (2)	SHift Logical Left	240
2.2.101	SHLR (1)	SHift Logical Right	242
2.2.102	SHLR (2)	SHift Logical Right	244

2.2.103	SLEEP	SLEEP	246
2.2.104	STC (B)	STore from Control register.....	247
2.2.105	STC (B)	STore from Control register.....	248
2.2.106	STC (W)	STore from Control register.....	249
2.2.107	STC (W)	STore from Control register.....	251
2.2.108	STC (L)	STore from Control register.....	253
2.2.109	STM	STore from Multiple register	254
2.2.110	STMAC	STore from MAC register.....	256
2.2.111	SUB	SUBtract binary	258
2.2.112	SUBS	SUBtract with Sign extension.....	261
2.2.113	SUBX	SUBtract with eXtend carry.....	263
2.2.114	TAS	Test And Set	265
2.2.115	TRAPA	TRAP Always.....	266
2.2.116	XOR	eXclusive OR logical.....	268
2.2.117	XORC	eXclusive OR Control register.....	271
2.2.118	XORC	eXclusive OR Control register.....	272
2.3	List of Instruction Set		273
2.4	List of Instruction Codes.....		635
2.5	Condition Code Modification		890
Section 3 Processing States.....			897
3.1	Overview.....		897
3.2	Reset State		898
3.3	Exception-Handling State		899
3.3.1	Types of Exception Handling and Their Priority		899
3.3.2	Reset Exception Handling.....		900
3.3.3	Illegal Instruction.....		900
3.3.4	Trace		901
3.3.5	Interrupt Exception Handling and Trap Instruction Exception Handling		901
3.4	Program Execution State.....		902
3.5	Bus-Released State		902
3.6	Power-Down State		902
3.6.1	Sleep Mode		902
3.6.2	Software Standby Mode.....		902
3.6.3	Hardware Standby Mode		902
Section 4 Basic Timing.....			903
4.1	On-Chip Memory (ROM, RAM) Access Timing.....		903
4.2	On-Chip Peripheral Module Access Timing.....		904
4.3	External Address Space Access Timing		904

4.4	Bus Arbitration Timing.....	905
	Main Revisions and Additions in this Edition.....	907

Figures

Section 1 Overview

Figure 1.1	CPU Operating Modes	3
Figure 1.2	Exception Vector Table (Normal Mode).....	4
Figure 1.3	Stack Structure (Normal Mode)	4
Figure 1.4	Exception Vector Table (Middle and Advanced Modes).....	6
Figure 1.5	Stack Structure (Middle and Advanced Modes).....	7
Figure 1.6	Exception Vector Table (Maximum Modes).....	8
Figure 1.7	Stack Structure (Maximum Mode).....	8
Figure 1.8	Memory Map.....	9
Figure 1.9	CPU Registers	11
Figure 1.10	Usage of General Registers	12
Figure 1.11	Stack.....	13
Figure 1.12	General Register Data Formats	17
Figure 1.13	Memory Data Formats.....	18
Figure 1.14	Instruction Formats.....	36
Figure 1.15	Branch Address Specification in Memory Indirect Mode	43

Section 3 Processing States

Figure 3.1	Processing States	897
Figure 3.2	State Transitions	898
Figure 3.3	Stack Structure after Exception Handling (Example).....	901

Section 4 Basic Timing

Figure 4.1	On-Chip Memory Access Cycle.....	903
Figure 4.2	On-Chip Peripheral Module Access Timing	904

Tables

Section 1 CPU

Table 1.1	Instruction Classification	19
Table 1.2	Combinations of Instructions and Addressing Modes (1).....	21
Table 1.2	Combinations of Instructions and Addressing Modes (2).....	24
Table 1.3	Operation Notation	25
Table 1.4	Data Transfer Instructions.....	26
Table 1.5	Block Transfer Instructions.....	27
Table 1.6	Arithmetic Operation Instructions	28
Table 1.7	Logic Operation Instructions	30
Table 1.8	Shift Operation Instructions.....	31
Table 1.9	Bit Manipulation Instructions	32
Table 1.10	Branch Instructions	34
Table 1.11	System Control Instructions.....	35
Table 1.12	Addressing Modes	37
Table 1.13	Absolute Address Access Ranges.....	41
Table 1.14	Effective Address Calculation for Transfer and Operation Instructions.....	44
Table 1.15	Effective Address Calculation for Branch Instructions.....	45

Section 2 Instruction

Table 2.1	List of Instruction Set	273
Table 2.2	List of Instruction Codes.....	635
Table 2.3	Condition Code Modification	890

Section 3 Processing States

Table 3.1	Exception Handling Types and Priority.....	899
-----------	--	-----

Section 1 CPU

The H8SX CPU is a high-speed CPU with an internal 32-bit architecture that is upward-compatible with the H8/300, H8/300H, and H8S CPUs.

The H8SX CPU has sixteen 16-bit general registers, can handle a 4-Gbyte linear address space, and is ideal for a realtime control system.

1.1 Features

- Upward-compatible with H8/300, H8/300H, and H8S CPUs
 - Can execute H8/300, H8/300H, and H8S/2000 object programs
- Sixteen 16-bit general registers
 - Also usable as sixteen 8-bit registers or eight 32-bit registers
- 87 basic instructions
 - 8/16/32-bit arithmetic and logic instructions
 - Multiply and divide instructions
 - Bit field transfer instructions
 - Powerful bit-manipulation instructions
 - Bit condition branch instructions
 - Multiply-and-accumulate instruction
- Eleven addressing modes
 - Register direct [Rn]
 - Register indirect [@ERn]
 - Register indirect with displacement [@(d:2,ERn), @(d:16,ERn), or @(d:32,ERn)]
 - Index register indirect with displacement [@(d:16,RnL.B), @(d:32,RnL.B), @(d:16,Rn.W), @(d:32,Rn.W), @(d:16,ERn.L), or @(d:32,ERn.L)]
 - Register indirect with pre-/post-increment or pre-/post-decrement [@+ERn, @ERn+, @-ERn, or @ERn-]
 - Absolute address [@aa:8, @aa:16, @aa:24, or @aa:32]
 - Immediate [#xx:3, #xx:4, #xx:8, #xx:16, or #xx:32]
 - Program-counter relative [@(d:8,PC) or @(d:16,PC)]
 - Program-counter relative with index register [@(RnL.B,PC), @(Rn.W,PC), or @(ERn.L,PC)]
 - Memory indirect [@@aa:8]
 - Extended memory indirect [@@vec:7]

- Two base registers
 - Vector base register
 - Short address base register
- 4-Gbyte address space
 - Program: 4 Gbytes
 - Data: 4 Gbytes
- High-speed operation
 - All frequently-used instructions executed in one or two states
 - 8/16/32-bit register-register add/subtract: 1 state
 - 8×8 -bit register-register multiply: 1 state (when multiplier supported)
 - $16 \div 8$ -bit register-register divide: 10 states (when divider supported)
 - 16×16 -bit register-register multiply: 1 state (when multiplier supported)
 - $32 \div 16$ -bit register-register divide: 18 states (when divider supported)
 - 32×32 -bit register-register multiply: 5 states (when multiplier supported)
 - $32 \div 32$ -bit register-register divide: 18 states (when divider supported)
- Four CPU operating modes
 - Normal mode
 - Middle mode
 - Advanced mode
 - Maximum mode
- Power-down modes
 - Transition is made by execution of SLEEP instruction
 - Choice of CPU operating clocks

1.2 CPU Operating Modes

The H8SX CPU has four operating modes: normal, middle, advanced and maximum modes. The mode is selected with the mode pins of the LSI or other sources.

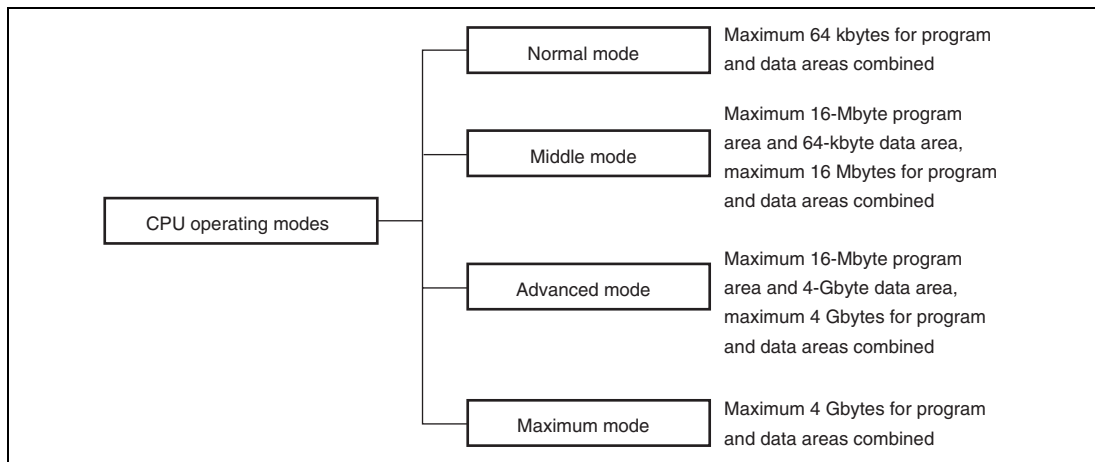


Figure 1.1 CPU Operating Modes

1.2.1 Normal Mode

The exception vector table and stack have the same structure as in the H8/300 CPU.

- Address Space

The maximum address space of 64 kbytes can be accessed.

- Extended Registers (En)

The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-bit segments of 32-bit registers. When the extended register En is used as a 16-bit register it can contain any value, even when the corresponding general register Rn is used as an address register. (If the general register Rn is referenced in the register indirect addressing mode with pre-/post-increment or pre-/post-decrement and a carry or borrow occurs, however, the value in the corresponding extended register En will be affected.)

- Instruction Set

All instructions and addressing modes can be used. Only the lower 16 bits of effective addresses (EA) are valid.

- Exception Vector Table and Memory Indirect Branch Addresses

In normal mode, the top area starting at H'0000 is allocated to the exception vector table. One branch address is stored per 16 bits. The structure of the exception vector table is shown in figure 1.2.

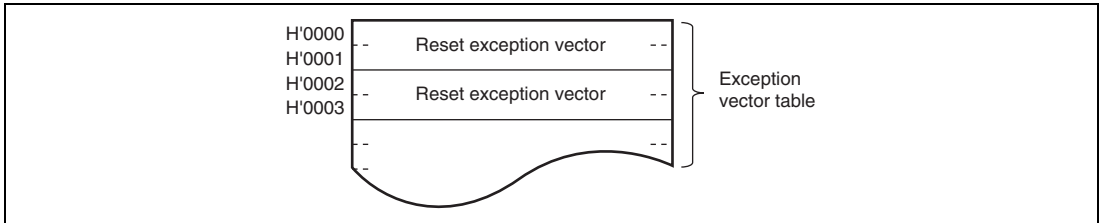


Figure 1.2 Exception Vector Table (Normal Mode)

The memory indirect (@@aa:8) and extended memory indirect (@@vec:7) addressing modes are used in the JMP and JSR instructions. An 8-bit absolute address included in the instruction code specifies a memory location. Execution branches to the contents of the memory location.

- Stack Structure

The stack structure of PC at a subroutine branch and that of PC and CCR at an exception handling are shown in figure 1.3. The PC contents are saved or restored in 16-bit units.

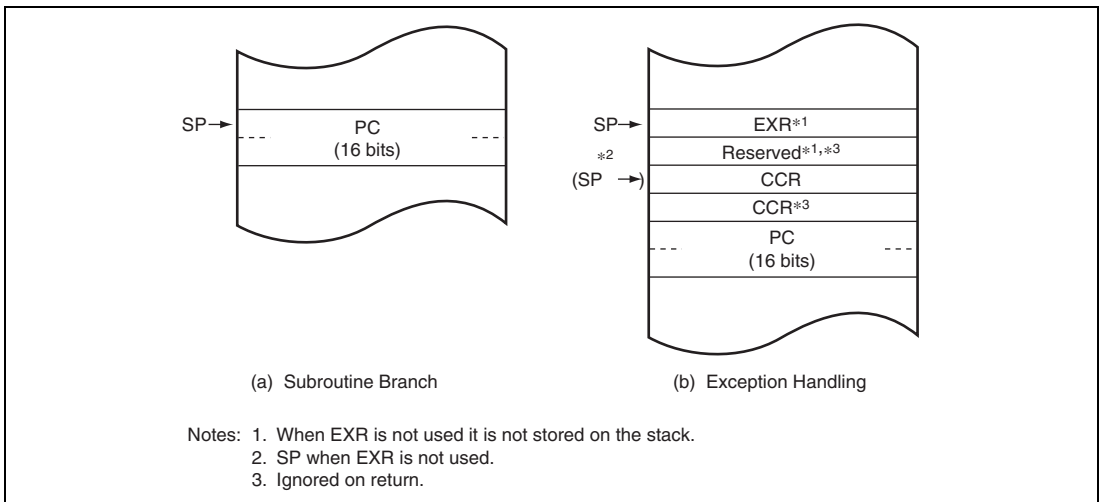


Figure 1.3 Stack Structure (Normal Mode)

1.2.2 Middle Mode

The program area in middle mode is extended to 16 Mbytes as compared with that in normal mode.

- Address Space

The maximum address space of 16 Mbytes can be accessed as a total of the program and data areas. For individual areas, up to 16 Mbytes of the program area or up to 64 kbytes of the data area can be allocated.

- Extended Registers (En)

The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-bit segments of 32-bit registers. When the extended register En is used as a 16-bit register (in other than the JMP and JSR instructions), it can contain any value even when the corresponding general register Rn is used as an address register. (If the general register Rn is referenced in the register indirect addressing mode with pre-/post-increment or pre-/post-decrement and a carry or borrow occurs, however, the value in the corresponding extended register En will be affected.)

- Instruction Set

All instructions and addressing modes can be used. Only the lower 16 bits of effective addresses (EA) are valid and the upper eight bits are sign-extended.

- Exception Vector Table and Memory Indirect Branch Addresses

In middle mode, the top area starting at H'000000 is allocated to the exception vector table. One branch address is stored per 32 bits. The upper eight bits are ignored and the lower 24 bits are stored. The structure of the exception vector table is shown in figure 1.4.

The memory indirect (@@aa:8) and extended memory indirect (@@vec:7) addressing modes are used in the JMP and JSR instructions. An 8-bit absolute address included in the instruction code specifies a memory location. Execution branches to the contents of the memory location.

In middle mode, an operand is a 32-bit (longword) operand, providing a 32-bit branch address. The upper eight bits are reserved and assumed to be H'00.

- Stack Structure

The stack structure of PC at a subroutine branch and that of PC and CCR at an exception handling are shown in figure 1.5. The PC contents are saved or restored in 24-bit units.

1.2.3 Advanced Mode

The data area is extended to 4 Gbytes as compared with that in middle mode.

- **Address Space**
The maximum address space of 4 Gbytes can be linearly accessed. For individual areas, up to 16 Mbytes of the program area and up to 4 Gbytes of the data area can be allocated.
- **Extended Registers (En)**
The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-bit segments of 32-bit registers or address registers.
- **Instruction Set**
All instructions and addressing modes can be used.
- **Exception Vector Table and Memory Indirect Branch Addresses**
In advanced mode, the top area starting at H'00000000 is allocated to the exception vector table. One branch address is stored per 32 bits. The upper eight bits are ignored and the lower 24 bits are stored. The structure of the exception vector table is shown in figure 1.4.

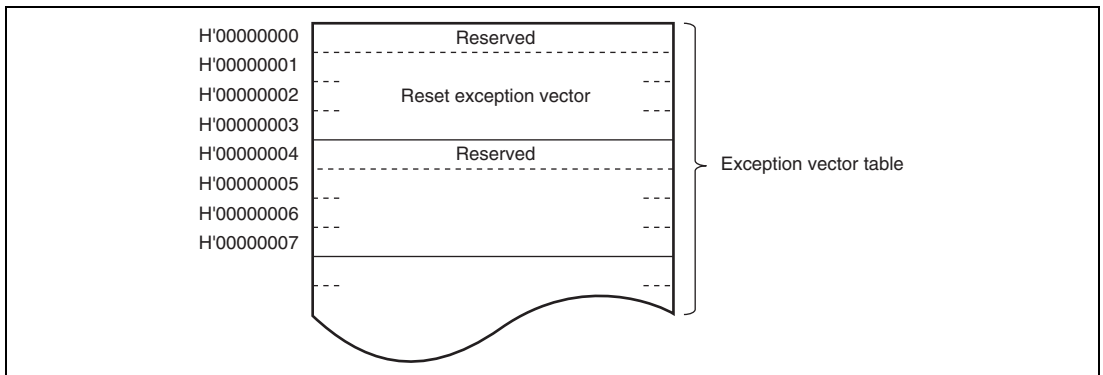


Figure 1.4 Exception Vector Table (Middle and Advanced Modes)

The memory indirect (@@aa:8) and extended memory indirect (@@vec:7) addressing modes are used in the JMP and JSR instructions. An 8-bit absolute address included in the instruction code specifies a memory location. Execution branches to the contents of the memory location. In advanced mode, an operand is a 32-bit (longword) operand, providing a 32-bit branch address. The upper eight bits are reserved and assumed to be H'00.

- Stack Structure

The stack structure of PC at a subroutine branch and that of PC and CCR at an exception handling are shown in figure 1.5. The PC contents are saved or restored in 24-bit units.

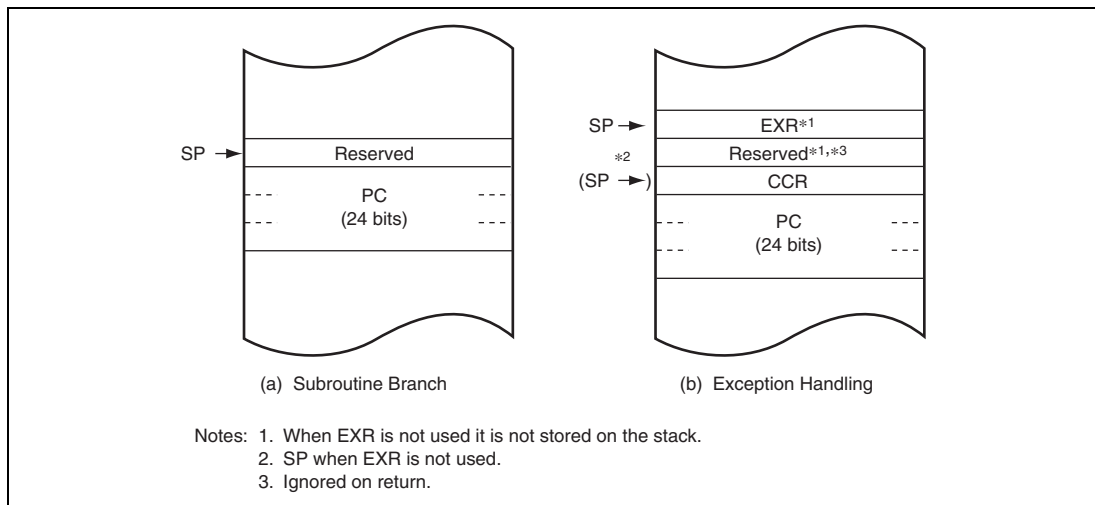


Figure 1.5 Stack Structure (Middle and Advanced Modes)

1.2.4 Maximum Mode

The program area is extended to 4 Gbytes as compared with that in advanced mode.

- Address Space

The maximum address space of 4 Gbytes can be linearly accessed.

- Extended Registers (En)

The extended registers (E0 to E7) can be used as 16-bit registers or as the upper 16-bit segments of 32-bit registers or address registers.

- Instruction Set

All instructions and addressing modes can be used.

- Exception Vector Table and Memory Indirect Branch Addresses

In maximum mode, the top area starting at H'00000000 is allocated to the exception vector table. One branch address is stored per 32 bits. The structure of the exception vector table is shown in figure 1.6.

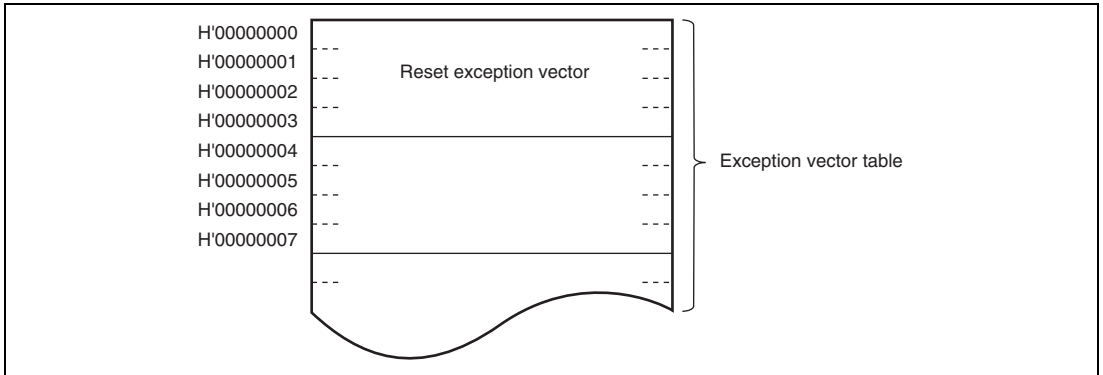


Figure 1.6 Exception Vector Table (Maximum Modes)

The memory indirect (@@aa:8) and extended memory indirect (@@vec:7) addressing modes are used in the JMP and JSR instructions. An 8-bit absolute address included in the instruction code specifies a memory location. Execution branches to the contents of the memory location. In maximum mode, an operand is a 32-bit (longword) operand, providing a 32-bit branch address.

- Stack Structure

The stack structure of PC at a subroutine branch and that of PC and CCR at an exception handling are shown in figure 1.7. The PC contents are saved or restored in 32-bit units. The EXR contents are saved or restored regardless of whether or not EXR is in use.

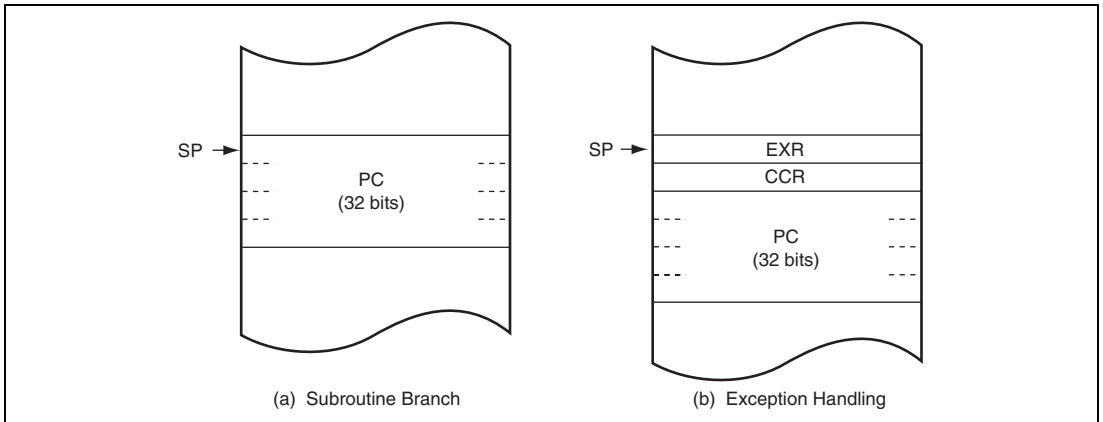


Figure 1.7 Stack Structure (Maximum Mode)

1.3 Instruction Fetch

The H8SX CPU has two modes for instruction fetch: 16-bit and 32-bit modes. It is recommended that the mode be set according to the bus width of the memory in which a program is stored. The instruction-fetch mode setting does not affect operation other than instruction fetch such as data accesses. For details, see the hardware manual for the corresponding product.

1.4 Address Space

Figure 1.8 shows a memory map of the H8SX CPU. The address space differs depending on the CPU operating mode.

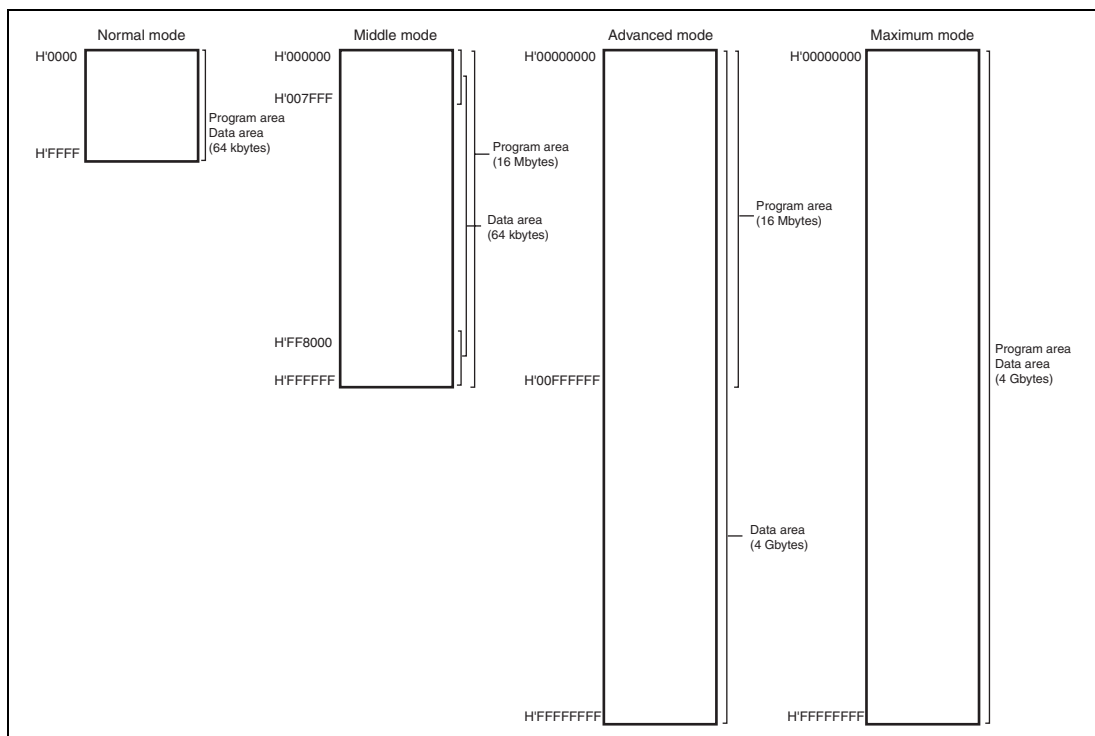


Figure 1.8 Memory Map

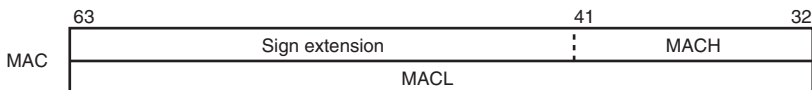
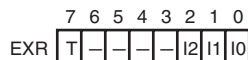
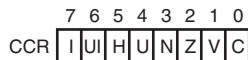
1.5 Registers

The H8SX CPU has the internal registers shown in figure 1.9. There are two types of registers: general registers and control registers. The control registers are the 32-bit program counter (PC), 8-bit extended control register (EXR), 8-bit condition-code register (CCR), 32-bit vector base register (VBR), 32-bit short address base register (SBR), and 64-bit multiply-accumulate register (MAC).

General Registers and Extended Registers

	15	0 7	0 7	0
ER0	E0		R0H	R0L
ER1	E1		R1H	R1L
ER2	E2		R2H	R2L
ER3	E3		R3H	R3L
ER4	E4		R4H	R4L
ER5	E5		R5H	R5L
ER6	E6		R6H	R6L
ER7 (SP)	E7		R7H	R7L

Control Registers



[Legend]

- | | |
|-----------------------------------|-----------------------------------|
| SP: Stack pointer | Z: Zero flag |
| PC: Program counter | V: Overflow flag |
| CCR: Condition-code register | C: Carry flag |
| I: Interrupt mask bit | EXR: Extended control register |
| U: User bit or interrupt mask bit | T: Trace bit |
| H: Half-carry flag | I2 to I0: Interrupt mask bits |
| U: User bit | VBR: Vector base register |
| N: Negative flag | SBR: Short address base register |
| | MAC: Multiply-accumulate register |

Figure 1.9 CPU Registers

1.5.1 General Registers

The H8SX CPU has eight 32-bit general registers. These general registers are all functionally alike and can be used as both address registers and data registers. When a general register is used as a data register, it can be accessed as a 32-bit, 16-bit, or 8-bit register. Figure 1.10 illustrates the usage of the general registers.

When the general registers are used as 32-bit registers or address registers, they are designated by the letters ER (ER0 to ER7).

When the general registers are used as 16-bit registers, the ER registers are divided into 16-bit general registers designated by the letters E (E0 to E7) and R (R0 to R7). These registers are functionally equivalent, providing a maximum sixteen 16-bit registers. The E registers (E0 to E7) are also referred to as extended registers.

When the general registers are used as 8-bit registers, the R registers are divided into 8-bit general registers designated by the letters RH (R0H to R7H) and RL (R0L to R7L). These registers are functionally equivalent, providing a maximum sixteen 8-bit registers.

The general registers ER (ER0 to ER7), R (R0 to R7), and RL (R0L to R7L) are also used as index registers. The size in the operand field determines which register is selected.

The usage of each register can be selected independently.

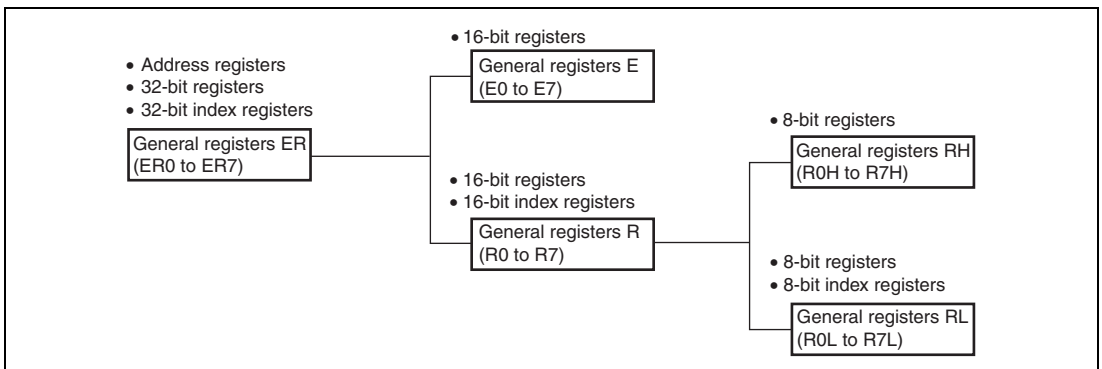


Figure 1.10 Usage of General Registers

General register ER7 has the function of stack pointer (SP) in addition to its general-register function, and is used implicitly in exception handling and subroutine branches. Figure 1.11 shows the stack.

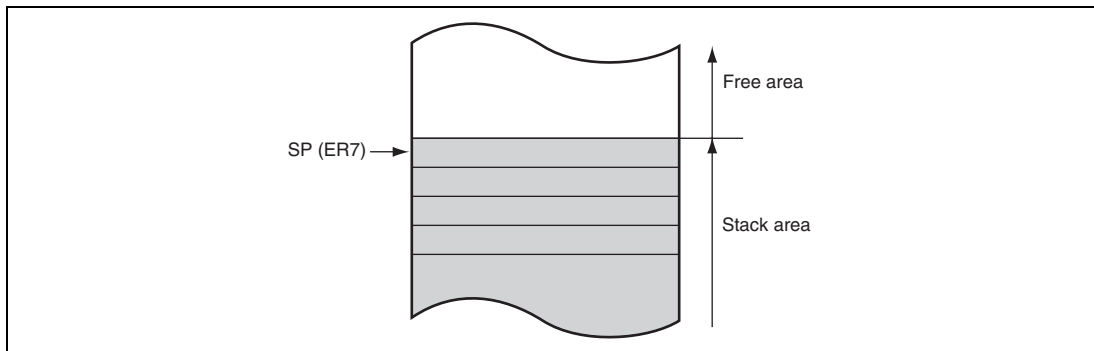


Figure 1.11 Stack

1.5.2 Program Counter (PC)

PC is a 32-bit counter that indicates the address of the next instruction the CPU will execute. The length of all CPU instructions is 16 bits (one word) or a multiple of 16 bits, so the least significant bit is ignored.

1.5.3 Condition-Code Register (CCR)

CCR is an 8-bit register that contains internal CPU status information, including an interrupt mask (I) and user (UI, U) bits and half-carry (H), negative (N), zero (Z), overflow (V), and carry (C) flags.

Operations can be performed on the CCR bits by the LDC, STC, ANDC, ORC, and XORC instructions. The N, Z, V, and C flags are used as branch conditions for conditional branch (Bcc) instructions.

Bit	Bit Name	Initial Value	R/W	Description
7	I	1	R/W	Interrupt Mask Bit Masks interrupts when set to 1. This bit is set to 1 at the start of an exception handling.
6	UI	Undefined	R/W	User Bit or Interrupt Mask Bit Can be written to and read from by software using the LDC, STC, ANDC, ORC, and XORC instructions. This bit can also be used as an interrupt mask bit.

Bit	Bit Name	Initial Value	R/W	Description
5	H	Undefined	R/W	<p>Half-Carry Flag</p> <p>When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.B, or NEG.B instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and cleared to 0 otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the ADD.L, SUB.L, CMP.L, or NEG.L instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 27, and cleared to 0 otherwise.</p>
4	U	Undefined	R/W	<p>User Bit</p> <p>Can be written to and read from by software using the LDC, STC, ANDC, ORC, and XORC instructions.</p>
3	N	Undefined	R/W	<p>Negative Flag</p> <p>Stores the value of the most significant bit (regarded as sign bit) of data.</p>
2	Z	Undefined	R/W	<p>Zero Flag</p> <p>Set to 1 to indicate zero data, and cleared to 0 to indicate non-zero data.</p>
1	V	Undefined	R/W	<p>Overflow Flag</p> <p>Set to 1 when an arithmetic overflow occurs, and cleared to 0 otherwise.</p>
0	C	Undefined	R/W	<p>Carry Flag</p> <p>Set to 1 when a carry occurs, and cleared to 0 otherwise. A carry has the following types:</p> <ul style="list-style-type: none"> • Carry from the result of addition • Borrow from the result of subtraction • Carry from the result of shift or rotation <p>The carry flag is also used as a bit accumulator by bit manipulation instructions.</p>

1.5.4 Extended Control Register (EXR)

EXR is an 8-bit register that contains the trace bit (T) and three interrupt mask bits (I2 to I0).

Operations can be performed on the EXR bits by the LDC, STC, ANDC, ORC, and XORC instructions.

For details, see the hardware manual for the corresponding product.

Bit	Bit Name	Initial Value	R/W	Description
7	T	0	R/W	Trace Bit When this bit is set to 1, a trace exception is generated each time an instruction is executed. When this bit is cleared to 0, instructions are executed in sequence.
6 to 3	—	All 1	R/W	Reserved These bits are always read as 1.
2	I2	1	R/W	Interrupt Mask Bits
1	I1	1	R/W	These bits designate the interrupt mask level (0 to 7).
0	I0	1	R/W	

1.5.5 Vector Base Register (VBR)

VBR is a 32-bit register in which the upper 20 bits are valid. The lower 12 bits of this register are read as 0s. This register is a base address of the vector area for exception handlings other than a reset and a CPU address error (extended memory indirect is also out of the target). The initial value is H'00000000. The VBR contents are changed with the LDC and STC instructions.

1.5.6 Short Address Base Register (SBR)

SBR is a 32-bit register in which the upper 24 bits are valid. The lower eight bits are read as 0s. In 8-bit absolute address addressing mode (@aa:8), this register is used as the upper address. The initial value is H'FFFFFFF0. The SBR contents are changed with the LDC and STC instructions.

1.5.7 Multiply-Accumulate Register (MAC)

MAC is a 64-bit register that stores the results of multiply-and-accumulate operations. It consists of two 32-bit registers denoted MACH and MACL. The lower 10 bits of MACH are valid; the

upper bits are sign extended. The MAC contents are changed with the MAC, CLRMAC, LDMAC, and STMAC instructions.

1.5.8 Initial Values of CPU Registers

Reset exception handling loads the start address from the vector table into the PC, clears the T bit in EXR to 0, and sets the I bits in CCR and EXR to 1. The general registers, MAC, and the other bits in CCR are not initialized. In particular, the initial value of the stack pointer (ER7) is undefined. The SP should therefore be initialized using an MOV.L instruction executed immediately after a reset.

1.6 Data Formats

The H8SX CPU can process 1-bit, 4-bit BCD, 8-bit (byte), 16-bit (word), and 32-bit (longword) data.

Bit-manipulation instructions operate on 1-bit data by accessing bit n ($n = 0, 1, 2, \dots, 7$) of byte operand data. The DAA and DAS decimal-adjust instructions treat byte data as two digits of 4-bit BCD data.

1.6.1 General Register Data Formats

Figure 1.12 shows the data formats in general registers.

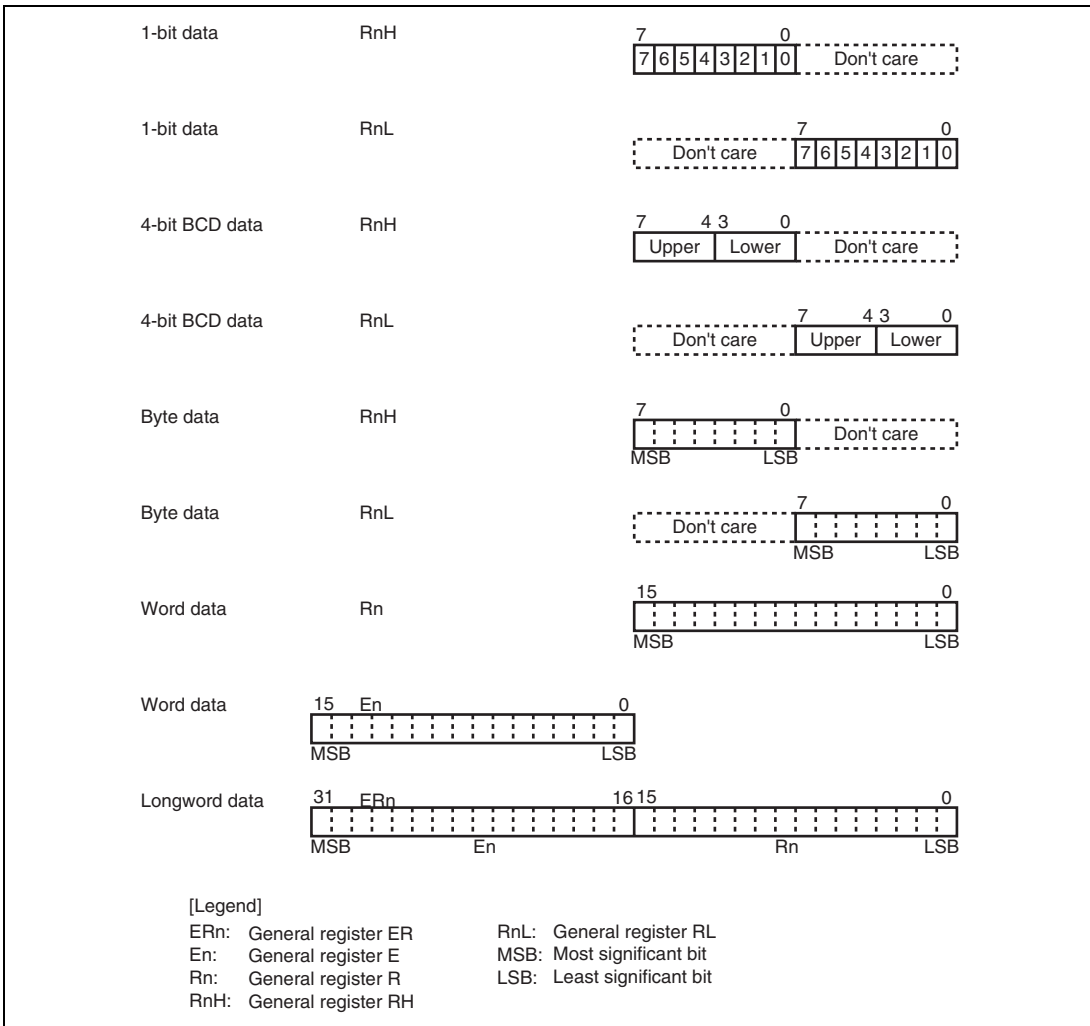


Figure 1.12 General Register Data Formats

1.6.2 Memory Data Formats

Figure 1.13 shows the data formats in memory.

The H8SX CPU can access word data and longword data which are stored at any addresses in memory. When word data begins at an odd address or longword data begins at an address other than a multiple of 4, a bus cycle is divided into two or more accesses. For example, when longword data begins at an odd address, the bus cycle is divided into byte, word, and byte accesses. In this case, these accesses are assumed to be individual bus cycles.

However, instructions to be fetched, word and longword data to be accessed during execution of the stack manipulation, branch table manipulation, block transfer instructions, and MAC instruction should be located to even addresses.

When SP (ER7) is used as an address register to access the stack, the operand size should be word size or longword size.

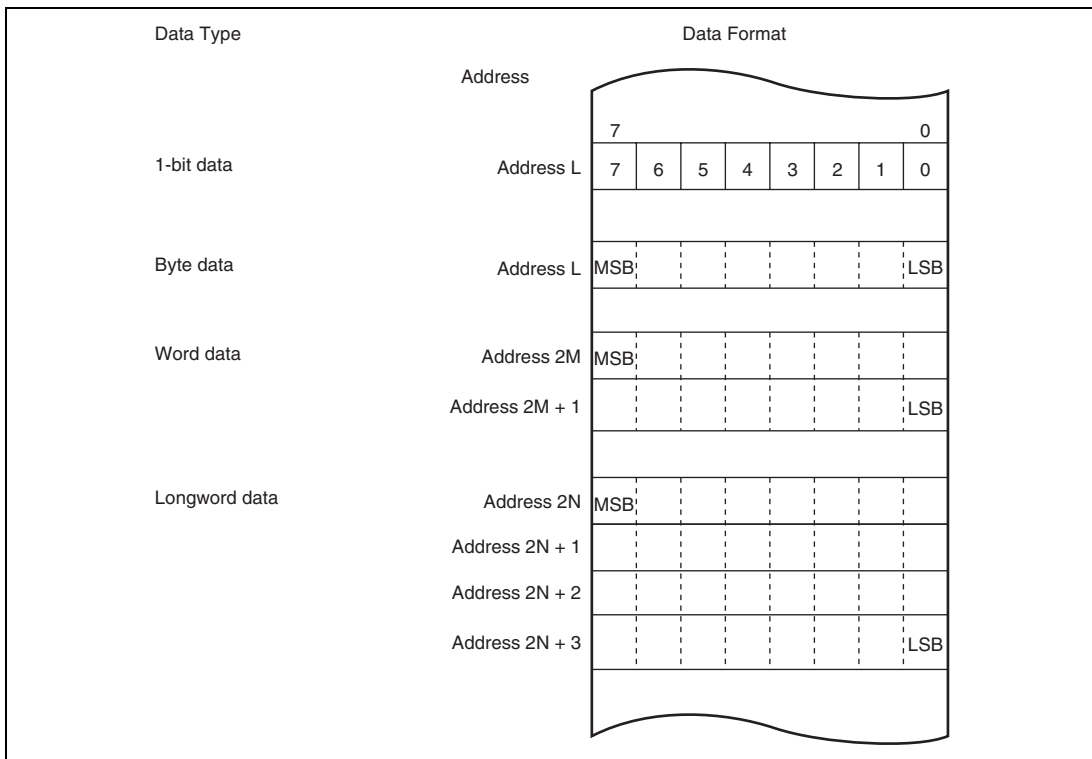


Figure 1.13 Memory Data Formats

1.7 Instruction Set

The H8SX CPU has 87 types of instructions. The instructions are classified by function as shown in table 1.1. The arithmetic operation, logic operation, shift, and bit manipulation instructions are called operation instruction in this manual.

Table 1.1 Instruction Classification

Function	Instructions	Size	Types
Data transfer	MOV	B/W/L	6
	MOVFP, MOVTPE	B	
	POP, PUSH* ¹	W/L	
	LDM, STM	L	
	MOVA	B/W* ²	
Block transfer	EEPMOV	B	3
	MOVMD	B/W/L	
	MOVSD	B	
Arithmetic operations	ADD, ADDX, SUB, SUBX, CMP, NEG, INC, DEC	B/W/L	27
	DAA, DAS	B	
	ADDS, SUBS	L	
	MULXU, DIVXU, MULXS, DIVXS	B/W	
	MULU, DIVU, MULS, DIVS	W/L	
	MULU/U* ⁶ , MULS/U* ⁶	L	
	EXTU, EXTS	W/L	
	TAS	B	
	MAC* ⁶	—	
	LDMAC* ⁶ , STMAC* ⁶	—	
	CLRMAC* ⁶	—	
Logic operations	AND, OR, XOR, NOT	B/W/L	4
Shift	SHLL, SHLR, SHAL, SHAR, ROTL, ROTR, ROTXL, ROTXR	B/W/L	8
Bit manipulation	BSET, BCLR, BNOT, BTST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR, BLD, BILD, BST, BIST	B	20
	BSET/EQ, BSET/NE, BCLR/EQ, BCLR/NE, BSTZ, BISTZ	B	
	BFLD, BFST	B	

Function	Instructions	Size	Types
Branch	BRA/BS, BRA/BC, BSR/BS, BSR/BC	B* ³	9
	Bcc* ⁵ , JMP, BSR, JSR, RTS	—	
	RTS/L	L* ⁵	
	BRA/S	—	
System control	TRAPA, RTE, SLEEP, NOP	—	10
	RTE/L	L* ⁵	
	LDC, STC, ANDC, ORC, XORC	B/W/L	
		Total	87

[Legend]

B: Byte size

W: Word size

L: Longword size

Notes: 1. POP.W Rn and PUSH.W Rn are identical to MOV.W @SP+, Rn and MOV.W Rn, @-SP.

POP.L ERn and PUSH.L ERn are identical to MOV.L @SP+, ERn and MOV.L ERn, @-SP.

2. Size of data to be added with a displacement
3. Size of data to specify a branch condition
4. Bcc is the generic designation of a conditional branch instruction.
5. Size of general register to be restored
6. Only when the multiplier is supported

1.7.1 Instructions and Addressing Modes

Table 1.2 indicates the combinations of instructions and addressing modes that the H8SX CPU can use.

Table 1.2 Combinations of Instructions and Addressing Modes (1)

Classification	Instruction	Size	#xx	Rn	Addressing Mode						
					@ERn	@(d,ERn)	@(d, ERn/L) Rn.L/B/ Rn.W/ ERn.L	@-ERn/ @ERn+/ @ERn-/ @+ERn	@aa:8	@aa:16/ @aa:32	—
Data transfer	MOV	B/W/L	S	SD	SD	SD	SD	SD	SD	SD	
		B		S/D					S/D		
	MOVFPE, MOVTPE	B		S/D						S/D* ¹	
	POP, PUSH	W/L		S/D				S/D* ²			
	LDM, STM	L		S/D				S/D* ²			
	MOVA* ⁴	B/W	S	S	S	S	S	S	S	S	
Block transfer	EEPMOV	B									SD* ³
	MOVMD	B/W/L									SD* ³
	MOVSD	B									SD* ³
Arithmetic operations	ADD, CMP	B	S	D	D	D	D	D	D	D	D
		B		S	D	D	D	D	D	D	D
		B		D	S	S	S	S	S	S	S
		B			SD	SD	SD	SD	SD	SD	SD
		W/L	S	SD	SD	SD	SD	SD	SD	SD	SD
	SUB	B	S		D	D	D	D	D	D	D
		B		S	D	D	D	D	D	D	D
		B		D	S	S	S	S	S	S	S
		B			SD	SD	SD	SD	SD	SD	SD
		W/L	S	SD	SD	SD	SD	SD	SD	SD	SD
	ADDX, SUBX	B/W/L	S	SD							
		B/W/L	S		SD						
		B/W/L	S					SD* ⁵			
	INC, DEC	B/W/L		D							
	ADDS, SUBS	L		D							
	DAA, DAS	B		D							
	MULXU, DIVXU	B/W	S:4	SD							
	MULU, DIVU	W/L	S:4	SD							

Classification	Instruction	Size	#xx	Rn	Addressing Mode							
					@ERn	@(d,ERn)	ERn.L)	@+ERn	@aa:8	@aa:16/ @aa:32	—	
Arithmetic operations	MULXS, DIVXS	B/W	S:4	SD								
	MULS, DIVS	W/L	S:4	SD								
	NEG	B		D	D	D	D	D	D	D		
		W/L		D	D	D	D	D	D	D		
	EXTU, EXTS	W/L		D	D	D	D	D	D			
	TAS	B			D							
	MAC* ¹²	—										
	CLRMAC* ¹²	—										O
	LDMAC* ¹²	—			S							
STMAC* ¹²	—			D								
Logic operations	AND, OR, XOR	B		S	D	D	D	D	D	D	D	
			B		D	S	S	S	S	S	S	
			B			SD	SD	SD	SD		SD	
			W/L	S	SD	SD	SD	SD	SD		SD	
	NOT	B		D	D	D	D	D	D	D	D	
W/L			D	D	D	D	D	D	D	D		
Shift	SHLL, SHLR	B		D	D	D	D	D	D	D	D	
			B/W/L* ⁶		D	D	D	D	D	D	D	
			B/W/L* ⁷		D							
	SHAL, SHAR ROTL, ROTR ROTXL, ROTXR	B		D	D	D	D	D	D	D	D	
			W/L		D	D	D	D	D	D	D	
Bit manipulation	BSET, BCLR, BNOT, BTST, BSET/cc, BCLR/cc	B		D	D				D	D		
			BAND, BIAND, BOR, BIOR, BXOR, BIXOR, BLD, BILD, BST, BIST, BSTZ, BISTZ	B		D	D			D	D	

Classification	Instruction	Size	#xx	Rn	Addressing Mode							
					@ERn	@(d,ERn)	ERn.L	@+ERn	@aa:8	@aa:32	—	
Bit manipulation	BFLD	B		D	S					S	S	
	BFST	B		S	D					D	D	
Branch	BRA/BS, BRA/BC* ⁸	B			S					S	S	
	BSR/BS, BSR/BC* ⁸	B			S					S	S	
System control	LDC (CCR, EXR)	B/W* ⁹	S	S	S	S			S* ¹⁰		S	
	LDC (VBR, SBR)	L			S							
	STC (CCR, EXR)	B/W* ⁹		D	D	D			D* ¹¹		D	
	STC (VBR, SBR)	L			D							
	ANDC, ORC, XORC	B	S									
	SLEEP	—										O
	NOP	—										O

[Legend]

d: d:16 or d:32

S: Can be specified as a source operand.

D: Can be specified as a destination operand.

SD: Can be specified as either a source or destination operand or both.

S/D: Can be specified as either a source or destination operand.

S:4: 4-bit immediate data can be specified as a source operand.

O: Can be used.

Notes: 1. Only @aa:16 is available.

2. @ERn+ as a source operand and @-ERn as a destination operand

3. Specified by ER5 as a source address and ER6 as a destination address for data transfer.

4. Size of data to be added with a displacement

5. Only @ERn- is available

6. When the number of bits to be shifted is 1, 2, 4, 8, or 16

7. When the number of bits to be shifted is specified by 5-bit immediate data or a general register

8. Size of data to specify a branch condition

9. Byte when immediate or register direct, otherwise, word

10. Only @ERn+ is available

11. Only @-ERn is available

12. Only when the multiplier is supported

Table 1.2 Combinations of Instructions and Addressing Modes (2)

Classification	Instruction	Size	Addressing Mode							
			@ERn	@(d,PC)	@(RnL, B/Rn.W/ ERn.L, PC)	@aa:24	@ aa:32	@@ aa:8	@@vec:7 —	
Branch	BRA/BS, BRA/BC	—		O						
	BSR/BS, BSR/BC	—		O						
	Bcc	—		O						
	BRA	—		O	O					
	BRA/S	—		O*						
	JMP	—	O			O	O	O	O	
	BSR	—		O						
	JSR	—	O			O	O	O	O	
	RTS, RTS/L	—								O
System control	TRAPA	—								O
	RTE, RTE/L	—								O

[Legend]

d: d:8 or d:16

O: Can be used.

Note: * Only @(d:8, PC) is available.

1.7.2 Table of Instructions Classified by Function

Tables 1.4 to 1.11 summarize the instructions in each functional category. The notation used in these tables is defined in table 1.3.

Table 1.3 Operation Notation

Operation Notation	Description
Rd	General register (destination)*
Rs	General register (source)*
Rn	General register*
ERn	General register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended control register
CCR	Condition-code register
VBR	Vector base register
SBR	Short address base register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
-	Subtraction
×	Multiplication
÷	Division
^	Logical AND
∨	Logical OR
⊕	Logical exclusive OR
→	Move
~	Logical not (logical complement)
:8/:16/:24/:32	8-, 16-, 24-, or 32-bit length

Note: * General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit registers (R0 to R7, E0 to E7), and 32-bit registers (ER0 to ER7).

Table 1.4 Data Transfer Instructions

Instruction	Size	Function
MOV	B/W/L	#IMM → (EAd), (EAs) → (EAd) Transfers data between immediate data, general registers, and memory.
MOVFPE	B	(EAs) → Rd
MOVTPE	B	Rs → (EAs)
POP	W/L	@SP+ → Rn Restores the data from the stack to a general register.
PUSH	W/L	Rn → @-SP Saves general register contents on the stack.
LDM	L	@SP+ → Rn (register list) Restores the data from the stack to multiple general registers. Two, three, or four general registers which have serial register numbers can be specified.
STM	L	Rn (register list) → @-SP Saves the contents of multiple general registers on the stack. Two, three, or four general registers which have serial register numbers can be specified.
MOVA	B/W	EA → Rd Zero-extends and shifts the contents of a specified general register or memory data and adds them with a displacement. The result is stored in a general register.

Table 1.5 Block Transfer Instructions

Instruction	Size	Function
EEPMOV.B EEPMOV.W	B	Transfers a data block. Transfers byte data which begins at a memory location specified by ER5 to a memory location specified by ER6. The number of byte data to be transferred is specified by R4 or R4L.
MOVMD.B	B	Transfers a data block. Transfers byte data which begins at a memory location specified by ER5 to a memory location specified by ER6. The number of byte data to be transferred is specified by R4.
MOVMD.W	W	Transfers a data block. Transfers word data which begins at a memory location specified by ER5 to a memory location specified by ER6. The number of word data to be transferred is specified by R4.
MOVMD.L	L	Transfers a data block. Transfers longword data which begins at a memory location specified by ER5 to a memory location specified by ER6. The number of longword data to be transferred is specified by R4.
MOVSD.B	B	Transfers a data block with zero data detection. Transfers byte data which begins at a memory location specified by ER5 to a memory location specified by ER6. The number of byte data to be transferred is specified by R4. When zero data is detected during transfer, the transfer stops and execution branches to a specified address.

Table 1.6 Arithmetic Operation Instructions

Instruction	Size	Function
ADD	B/W/L	$(EAd) \pm \#IMM \rightarrow (EAd)$, $(EAd) \pm (EAs) \rightarrow (EAd)$
SUB		Performs addition or subtraction on data between immediate data, general registers, and memory. Immediate byte data cannot be subtracted from byte data in a general register.
ADDX	B/W/L	$(EAd) \pm \#IMM \pm C \rightarrow (EAd)$, $(EAd) \pm (EAs) \pm C \rightarrow (EAd)$
SUBX		Performs addition or subtraction with carry on data between immediate data, general registers, and memory. The addressing mode which specifies a memory location can be specified as register indirect with post-decrement or register indirect.
INC	B/W/L	$Rd \pm 1 \rightarrow Rd$, $Rd \pm 2 \rightarrow Rd$
DEC		Increments or decrements a general register by 1 or 2. (Byte operands can be incremented or decremented by 1 only.)
ADDS	L	$Rd \pm 1 \rightarrow Rd$, $Rd \pm 2 \rightarrow Rd$, $Rd \pm 4 \rightarrow Rd$
SUBS		Adds or subtracts the value 1, 2, or 4 to or from data in a general register.
DAA	B	Rd (decimal adjust) $\rightarrow Rd$
DAS		Decimal-adjusts an addition or subtraction result in a general register by referring to the CCR to produce 2-digit 4-bit BCD data.
MULXU	B/W	$Rd \times Rs \rightarrow Rd$ Performs unsigned multiplication on data in two general registers: either 8 bits \times 8 bits \rightarrow 16 bits, or 16 bits \times 16 bits \rightarrow 32 bits.
MULU	W/L	$Rd \times Rs \rightarrow Rd$ Performs unsigned multiplication on data in two general registers: either 8 bits \times 8 bits \rightarrow 16 bits, or 16 bits \times 16 bits \rightarrow 32 bits.
MULU/U*	L	$Rd \times Rs \rightarrow Rd$ Performs unsigned multiplication on data in two general registers (32 bits \times 32 bits \rightarrow upper 32 bits).
MULXS	B/W	$Rd \times Rs \rightarrow Rd$ Performs signed multiplication on data in two general registers: either 8 bits \times 8 bits \rightarrow 16 bits, or 16 bits \times 16 bits \rightarrow 32 bits.
MULS	W/L	$Rd \times Rs \rightarrow Rd$ Performs signed multiplication on data in two general registers: either 16 bits \times 16 bits \rightarrow 16 bits, or 32 bits \times 32 bits \rightarrow 32 bits.
MULS/U*	L	$Rd \times Rs \rightarrow Rd$ Performs signed multiplication on data in two general registers (32 bits \times 32 bits \rightarrow upper 32 bits).

Instruction	Size	Function
DIVXU	B/W	$Rd \div Rs \rightarrow Rd$ Performs unsigned division on data in two general registers: either 16 bits \div 8 bits \rightarrow 8-bit quotient and 8-bit remainder, or 32 bits \div 16 bits \rightarrow 16-bit quotient and 16-bit remainder.
DIVU	W/L	$Rd \div Rs \rightarrow Rd$ Performs unsigned division on data in two general registers: either 16 bits \div 16 bits \rightarrow 16-bit quotient, or 32 bits \div 32 bits \rightarrow 32-bit quotient.
DIVXS	B/W	$Rd \div Rs \rightarrow Rd$ Performs signed division on data in two general registers: either 16 bits \div 8 bits \rightarrow 8-bit quotient and 8-bit remainder, or 32 bits \div 16 bits \rightarrow 16-bit quotient and 16-bit remainder.
DIVS	W/L	$Rd \div Rs \rightarrow Rd$ Performs signed division on data in two general registers: either 16 bits \div 16 bits \rightarrow 16-bit quotient, or 32 bits \div 32 bits \rightarrow 32-bit quotient.
CMP	B/W/L	$(EAd) - \#IMM, (EAd) - (EAs)$ Compares data between immediate data, general registers, and memory and stores the result in CCR.
NEG	B/W/L	$0 - (EAd) \rightarrow (EAd)$ Takes the two's complement (arithmetic complement) of data in a general register or the contents of a memory location.
EXTU	W/L	$(EAd) \text{ (zero extension)} \rightarrow (EAd)$ Performs zero-extension on the lower 8 or 16 bits of data in a general register or memory to word or longword size. The lower 8 bits to word or longword, or the lower 16 bits to longword can be zero-extended.
EXTS	W/L	$(EAd) \text{ (sign extension)} \rightarrow (EAd)$ Performs sign-extension on the lower 8 or 16 bits of data in a general register or memory to word or longword size. The lower 8 bits to word or longword, or the lower 16 bits to longword can be sign-extended.
TAS	B	$@ERd - 0, 1 \rightarrow \langle \text{bit } 7 \rangle \text{ of } @EAd$ Tests memory contents, and sets the most significant bit (bit 7) to 1.
MAC*	—	$(EAs) \times (EAd) + MAC \rightarrow MAC$ Performs signed multiplication on memory contents and adds the result to MAC.
CLRMAC*	—	$0 \rightarrow MAC$ Clears MAC to zero.

Instruction	Size	Function
LDMAC*	—	Rs → MAC Loads data from a general register to MAC.
STMAC*	—	MAC → Rd Stores data from MAC to a general register.

Note: * Only when the multiplier is supported.

Table 1.7 Logic Operation Instructions

Instruction	Size	Function
AND	B/W/L	$(EAd) \wedge \#IMM \rightarrow (EAd)$, $(EAd) \wedge (EAs) \rightarrow (EAd)$ Performs a logical AND operation on data between immediate data, general registers, and memory.
OR	B/W/L	$(EAd) \vee \#IMM \rightarrow (EAd)$, $(EAd) \vee (EAs) \rightarrow (EAd)$ Performs a logical OR operation on data between immediate data, general registers, and memory.
XOR	B/W/L	$(EAd) \oplus \#IMM \rightarrow (EAd)$, $(EAd) \oplus (EAs) \rightarrow (EAd)$ Performs a logical exclusive OR operation on data between immediate data, general registers, and memory.
NOT	B/W/L	$\sim (EAd) \rightarrow (EAd)$ Takes the one's complement of the contents of a general register or a memory location.

Table 1.8 Shift Operation Instructions

Instruction	Size	Function
SHLL	B/W/L	(EAd) (shift) → (EAd)
SHLR		Performs a logical shift on the contents of a general register or a memory location. The contents of a general register or a memory location can be shifted by 1, 2, 4, 8, or 16 bits. The contents of a general register can be shifted by any bits. In this case, the number of bits is specified by 5-bit immediate data or the lower 5 bits of the contents of a general register.
SHAL	B/W/L	(EAd) (shift) → (EAd)
SHAR		Performs an arithmetic shift on the contents of a general register or a memory location. 1-bit or 2-bit shift is possible.
ROTL	B/W/L	(EAd) (rotate) → (EAd)
ROTR		Rotates the contents of a general register or a memory location. 1-bit or 2-bit rotation is possible.
ROTXL	B/W/L	(EAd) (rotate) → (EAd)
ROTXR		Rotates the contents of a general register or a memory location with the carry bit. 1-bit or 2-bit rotation is possible.

Table 1.9 Bit Manipulation Instructions

Instruction	Size	Function
BSET	B	$1 \rightarrow \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle$ Sets a specified bit in the contents of a general register or a memory location to 1. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BSET/cc	B	$\text{if cc, } 1 \rightarrow \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle$ If the specified condition is satisfied, this instruction sets a specified bit in a memory location to 1. The bit number can be specified by 3-bit immediate data, or by the lower three bits of a general register. The Z flag status can be specified as a condition.
BCLR	B	$0 \rightarrow \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle$ Clears a specified bit in the contents of a general register or a memory location to 0. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BCLR/cc	B	$\text{if cc, } 0 \rightarrow \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle$ If the specified condition is satisfied, this instruction clears a specified bit in a memory location to 0. The bit number can be specified by 3-bit immediate data, or by the lower three bits of a general register. The Z flag status can be specified as a condition.
BNOT	B	$\sim \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle$ Inverts a specified bit in the contents of a general register or a memory location. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BTST	B	$\sim \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow Z$ Tests a specified bit in the contents of a general register or a memory location and sets or clears the Z flag accordingly. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BAND	B	$C \wedge \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow C$ ANDs the carry flag with a specified bit in the contents of a general register or a memory location and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
BIAND	B	$C \wedge [\sim \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle] \rightarrow C$ ANDs the carry flag with the inverse of a specified bit in the contents of a general register or a memory location and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.

Instruction	Size	Function
BOR	B	$C \vee \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow C$ ORs the carry flag with a specified bit in the contents of a general register or a memory location and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
BIOR	B	$C \vee [\sim \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle] \rightarrow C$ ORs the carry flag with the inverse of a specified bit in the contents of a general register or a memory location and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
BXOR	B	$C \oplus \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow C$ Exclusive-ORs the carry flag with a specified bit in the contents of a general register or a memory location and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
BIXOR	B	$C \oplus [\sim \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle] \rightarrow C$ Exclusive-ORs the carry flag with the inverse of a specified bit in the contents of a general register or a memory location and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
BLD	B	$\langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow C$ Transfers a specified bit in the contents of a general register or a memory location to the carry flag. The bit number is specified by 3-bit immediate data.
BILD	B	$\sim \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow C$ Transfers the inverse of a specified bit in the contents of a general register or a memory location to the carry flag. The bit number is specified by 3-bit immediate data.
BST	B	$C \rightarrow \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle$ Transfers the carry flag value to a specified bit in the contents of a general register or a memory location. The bit number is specified by 3-bit immediate data.
BSTZ	B	$Z \rightarrow \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle$ Transfers the zero flag value to a specified bit in the contents of a memory location. The bit number is specified by 3-bit immediate data.
BIST	B	$\sim C \rightarrow \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle$ Transfers the inverse of the carry flag value to a specified bit in the contents of a general register or a memory location. The bit number is specified by 3-bit immediate data.

Instruction	Size	Function
BISTZ	B	~ Z → (<bit-No.> of <EAd>) Transfers the inverse of the zero flag value to a specified bit in the contents of a memory location. The bit number is specified by 3-bit immediate data.
BFLD	B	(EAs) (bit field) → Rd Transfers a specified bit field in memory location contents to the lower bits of a specified general register.
BFST	B	Rs → (EAd) (bit field) Transfers the lower bits of a specified general register to a specified bit field in memory location contents.

Table 1.10 Branch Instructions

Instruction	Size	Function
BRA/BS BRA/BC	B	Tests a specified bit in memory location contents. If the specified condition is satisfied, execution branches to a specified address.
BSR/BS BSR/BC	B	Tests a specified bit in memory location contents. If the specified condition is satisfied, execution branches to a subroutine at a specified address.
Bcc	—	Branches to a specified address if the specified condition is satisfied.
BRA/S	—	Branches unconditionally to a specified address after executing the next instruction. The next instruction should be a 1-word instruction except for the block transfer and branch instructions.
JMP	—	Branches unconditionally to a specified address.
BSR	—	Branches to a subroutine at a specified address.
JSR	—	Branches to a subroutine at a specified address.
RTS	—	Returns from a subroutine.
RTS/L	—	Returns from a subroutine, restoring data from the stack to multiple general registers.

Table 1.11 System Control Instructions

Instruction	Size	Function
TRAPA	—	Starts trap-instruction exception handling.
RTE	—	Returns from an exception-handling routine.
RTE/L	—	Returns from an exception-handling routine, restoring data from the stack to multiple general registers.
SLEEP	—	Causes a transition to a power-down state.
LDC	B/W	#IMM → CCR, (EAs) → CCR, #IMM → EXR, (EAs) → EXR Loads immediate data or the contents of a general register or a memory location to CCR or EXR. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 8 bits are valid.
	L	Rs → VBR, Rs → SBR Transfers the general register contents to VBR or SBR.
STC	B/W	CCR → (EAd), EXR → (EAd) Transfers the contents of CCR or EXR to a general register or memory. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 8 bits are valid.
	L	VBR → Rd, SBR → Rd Transfers the contents of VBR or SBR to a general register.
ANDC	B	CCR ∧ #IMM → CCR, EXR ∧ #IMM → EXR Logically ANDs the CCR or EXR contents with immediate data.
ORC	B	CCR ∨ #IMM → CCR, EXR ∨ #IMM → EXR Logically ORs the CCR or EXR contents with immediate data.
XORC	B	CCR ⊕ #IMM → CCR, EXR ⊕ #IMM → EXR Logically exclusive-ORs the CCR or EXR contents with immediate data.
NOP	—	PC + 2 → PC Only increments the program counter.

1.7.3 Basic Instruction Formats

The H8SX CPU instructions consist of 2-byte (1-word) units. An instruction consists of an operation field (op field), a register field (r field), an effective address extension (EA field), and a condition field (cc).

Figure 1.14 shows examples of instruction formats.

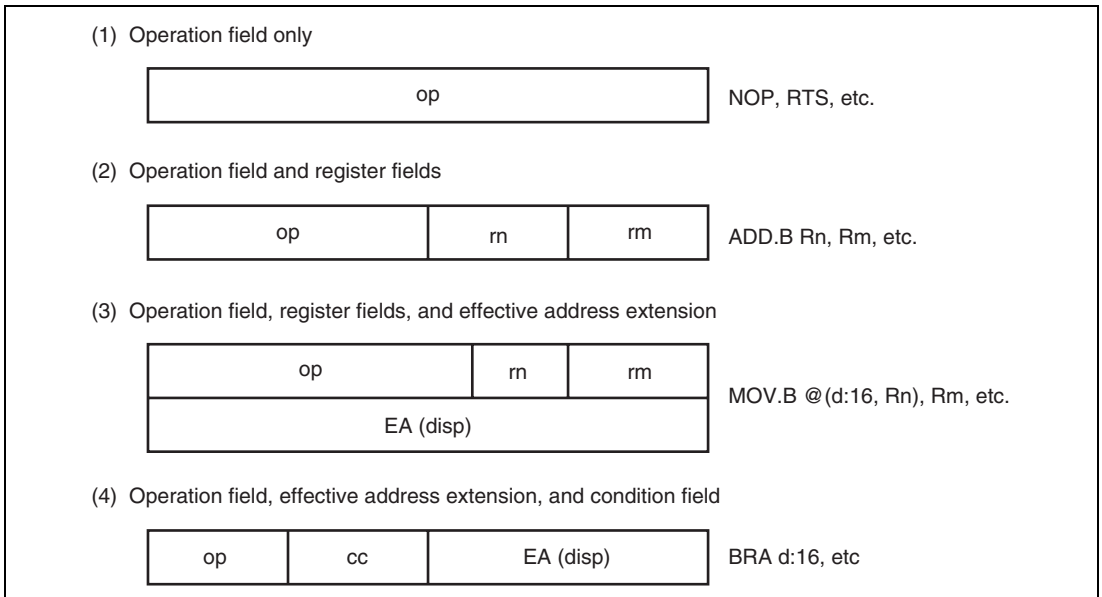


Figure 1.14 Instruction Formats

- **Operation Field**
Indicates the function of the instruction, and specifies the addressing mode and operation to be carried out on the operand. The operation field always includes the first four bits of the instruction. Some instructions have two operation fields.
- **Register Field**
Specifies a general register. Address registers are specified by 3 bits, data registers by 3 bits or 4 bits. Some instructions have two register fields. Some have no register field.
- **Effective Address Extension**
8, 16, or 32 bits specifying immediate data, an absolute address, or a displacement.
- **Condition Field**
Specifies the branch condition of Bcc instructions.

1.8 Addressing Modes and Effective Address Calculation

The H8SX CPU supports the 11 addressing modes listed in table 1.12. Each instruction uses a subset of these addressing modes.

Bit manipulation instructions use register direct, register indirect, or absolute addressing mode to specify an operand, and register direct (BSET, BCLR, BNOT, and BTST instructions) or immediate (3-bit) addressing mode to specify a bit number in the operand.

Table 1.12 Addressing Modes

No.	Addressing Mode	Symbol
1	Register direct	Rn
2	Register indirect	@ERn
3	Register indirect with displacement	@(d:2,ERn)/@(d:16,ERn)/@(d:32,ERn)
4	Index register indirect with displacement	@(d:16, RnL.B)/@(d:16,Rn.W)/@(d:16,ERn.L) @(d:32, RnL.B)/@(d:32,Rn.W)/@(d:32,ERn.L)
5	Register indirect with post-increment	@ERn+
	Register indirect with pre-decrement	@-ERn
	Register indirect with pre-increment	@+ERn
	Register indirect with post-decrement	@ERn-
6	Absolute address	@aa:8/@aa:16/@aa:24/@aa:32
7	Immediate	#xx:3/#xx:4/#xx:8/#xx:16/#xx:32
8	Program-counter relative	@(d:8,PC)/@(d:16,PC)
9	Program-counter relative with index register	@(RnL.B,PC)/@(Rn.W,PC)/@(ERn.L,PC)
10	Memory indirect	@@aa:8
11	Extended memory indirect	@@vec:7

1.8.1 Register Direct—Rn

The operand value is the contents of an 8-, 16-, or 32-bit general register which is specified by the register field in the instruction code.

R0H to R7H and R0L to R7L can be specified as 8-bit registers.

R0 to R7 and E0 to E7 can be specified as 16-bit registers.

ER0 to ER7 can be specified as 32-bit registers.

1.8.2 Register Indirect—@ERn

The operand value is the contents of the memory location which is pointed to by the contents of an address register (ERn). ERn is specified by the register field of the instruction code.

In advanced mode, if this addressing mode is used in a branch instruction, the lower 24 bits are valid and the upper 8 bits are all assumed to be 0 (H'00).

1.8.3 Register Indirect with Displacement—@(d:2, ERn), @(d:16, ERn), or @(d:32, ERn)

The operand value is the contents of a memory location which is pointed to by the sum of the contents of an address register (ERn) and a 16- or 32-bit displacement. ERn is specified by the register field of the instruction code. The displacement is included in the instruction code and the 16-bit displacement is sign-extended when added to ERn.

This addressing mode has a short format (@(d:2, ERn)). The short format can be used when the displacement is 1, 2, or 3 and the operand is byte data, when the displacement is 2, 4, or 6 and the operand is word data, or when the displacement is 4, 8, or 12 and the operand is longword data.

1.8.4 Index Register Indirect with Displacement—@(d:16,RnL.B), @(d:32,RnL.B), @(d:16,Rn.W), @(d:32,Rn.W), @(d:16,ERn.L), or @(d:32,ERn.L)

The operand value is the contents of a memory location which is pointed to by the sum of the following operation result and a 16- or 32-bit displacement: a specified bits of the contents of an address register (RnL, Rn, ERn) specified by the register field in the instruction code are zero-extended to 32-bit data and multiplied by 1, 2, or 4. The displacement is included in the instruction code and the 16-bit displacement is sign-extended when added to ERn. If the operand is byte data, ERn is multiplied by 1. If the operand is word or longword data, ERn is multiplied by 2 or 4, respectively.

1.8.5 Register Indirect with Post-Increment, Pre-Decrement, Pre-Increment, or Post-Decrement—@ERn+, @-ERn, @+ERn, or @ERn-

- Register indirect with post-increment—@ERn+

The operand value is the contents of a memory location which is pointed to by the contents of an address register (ERn). ERn is specified by the register field of the instruction code. After the memory location is accessed, 1, 2, or 4 is added to the address register contents and the sum is stored in the address register. The value added is 1 for byte access, 2 for word access, or 4 for longword access.

- Register indirect with pre-decrement—@-ERn

The operand value is the contents of a memory location which is pointed to by the following operation result: the value 1, 2, or 4 is subtracted from the contents of an address register (ERn). ERn is specified by the register field of the instruction code. After that, the operand value is stored in the address register. The value subtracted is 1 for byte access, 2 for word access, or 4 for longword access.

- Register indirect with pre-increment—@+ERn

The operand value is the contents of a memory location which is pointed to by the following operation result: the value 1, 2, or 4 is added to the contents of an address register (ERn). ERn is specified by the register field of the instruction code. After that, the operand value is stored in the address register. The value added is 1 for byte access, 2 for word access, or 4 for longword access.

- Register indirect with post-decrement—@ERn-

The operand value is the contents of a memory location which is pointed to by the contents of an address register (ERn). ERn is specified by the register field of the instruction code. After the memory location is accessed, 1, 2, or 4 is subtracted from the address register contents and the remainder is stored in the address register. The value subtracted is 1 for byte access, 2 for word access, or 4 for longword access.

If the contents of a general register which is also used as an address register is written to memory using this addressing mode, data to be written is the contents of the general register after calculating an effective address. If the same general register is specified in an instruction and two effective addresses are calculated, the contents of the general register after the first calculation of an effective address is used in the second calculation of an effective address.

Example 1:

```
MOV.W    R0, @ER0+
```

When ER0 before execution is H'12345678, H'567A is written at H'12345678.

Example 2:

```
MOV.B    @ER0+, @ER0+
```

When ER0 before execution is H'00001000, H'00001000 is read and the contents is written at

H'00001001.

After execution, ER0 is H'00001002.

Example 3:

XOR.B ROL, @ER0+

When the value in ER0 before execution is H'00001234, location H'00001234 is read, the read data is EORed with H'35, and the result is written to location H'00001234.

The value in ER0 after execution is H'00001235.

1.8.6 Absolute Address—@aa:8, @aa:16, @aa:24, or @aa:32

The operand value is the contents of a memory location which is pointed to by an absolute address included in the instruction code.

There are 8-bit (@aa:8), 16-bit (@aa:16), 24-bit (@aa:24), and 32-bit (@aa:32) absolute addresses.

To access the data area, the absolute address of 8 bits (@aa:8), 16 bits (@aa:16), or 32 bits (@aa:32) is used. For an 8-bit absolute address, the upper 24 bits are specified by SBR. For a 16-bit absolute address, the upper 16 bits are sign-extended. A 32-bit absolute address can access the entire address space.

To access the program area, the absolute address of 24 bits (@aa:24) or 32 bits (@aa:32) is used. For a 24-bit absolute address, the upper 8 bits are all assumed to be 0 (H'00).

Table 1.13 shows the accessible absolute address ranges.

Table 1.13 Absolute Address Access Ranges

Absolute Address	Normal Mode	Middle Mode	Advanced Mode	Maximum Mode
Data area	8 bits (@aa:8)	A consecutive 256-byte area (the upper address is set in SBR)		
	16 bits (@aa:16)	H'0000 to H'FFFF	H'000000 to H'007FFF,	H'00000000 to H'00007FFF, H'FFFF8000 to H'FFFFFFFF
	32 bits (@aa:32)		H'FF8000 to H'FFFFFF	H'00000000 to H'FFFFFFFF
Program area	24 bits (@aa:24)		H'000000 to H'FFFFFF	H'00000000 to H'00FFFFFF
	32 bits (@aa:32)			H'00000000 to H'00FFFFFF H'00000000 to H'FFFFFFFF

1.8.7 Immediate—#xx

The operand value is 8-bit (#xx:8), 16-bit (#xx:16), or 32-bit (#xx:32) data included in the instruction code.

This addressing mode has short formats in which 3- or 4-bit immediate data can be used.

When the size of immediate data is less than that of the destination operand value (byte, word, or longword) the immediate data is zero-extended.

The ADDS, SUBS, INC, and DEC instructions contain immediate data implicitly. Some bit manipulation instructions contain 3-bit immediate data in the instruction code, for specifying a bit number. The BFLD and BFST instructions contain 8-bit immediate data in the instruction code, for specifying a bit field. The TRAPA instruction contains 2-bit immediate data in the instruction code, for specifying a vector address.

1.8.8 Program-Counter Relative—@(d:8, PC) or @(d:16, PC):

This mode is used in the Bcc and BSR instructions. The operand value is a 32-bit branch address, which is the sum of an 8- or 16-bit displacement in the instruction code and the 32-bit address of the PC contents. The 8-bit or 16-bit displacement is sign-extended to 32 bits when added to the PC contents. The PC contents to which the displacement is added is the address of the first byte of the next instruction, so the possible branching range is -126 to $+128$ bytes (-63 to $+64$ words) or -32766 to $+32768$ bytes (-16383 to $+16384$ words) from the branch instruction. The resulting

value should be an even number. In advanced mode, only the lower 24 bits of this branch address are valid; the upper 8 bits are all assumed to be 0 (H'00).

1.8.9 Program-Counter Relative with Index Register—@(RnL.B, PC), @(Rn.W, PC), or @(ERn.L, PC)

This mode is used in the Bcc and BSR instructions. The operand value is a 32-bit branch address, which is the sum of the following operation result and the 32-bit address of the PC contents: the contents of an address register specified by the register field in the instruction code (RnL, Rn, or ERn) is zero-extended and multiplied by 2. The PC contents to which the displacement is added is the address of the first byte of the next instruction. In advanced mode, only the lower 24 bits of this branch address are valid; the upper 8 bits are all assumed to be 0 (H'00).

1.8.10 Memory Indirect—@@aa:8

This mode can be used by the JMP and JSR instructions. The operand value is a branch address, which is the contents of a memory location pointed to by an 8-bit absolute address in the instruction code.

The upper bits of an 8-bit absolute address are all assumed to be 0, so the address range is 0 to 255 (H'0000 to H'00FF in normal mode, H'000000 to H'0000FF in other modes).

In normal mode, the memory location is pointed to by word-size data and the branch address is 16 bits long. In other modes, the memory location is pointed to by longword-size data. In middle or advanced mode, the first byte of the longword-size data is assumed to be all 0 (H'00).

Note that the top part of the address range is also used as the exception handling vector area. A vector address of an exception handling other than a reset or a CPU address error can be changed by VBR.

Figure 1.15 shows an example of specification of a branch address using this addressing mode.

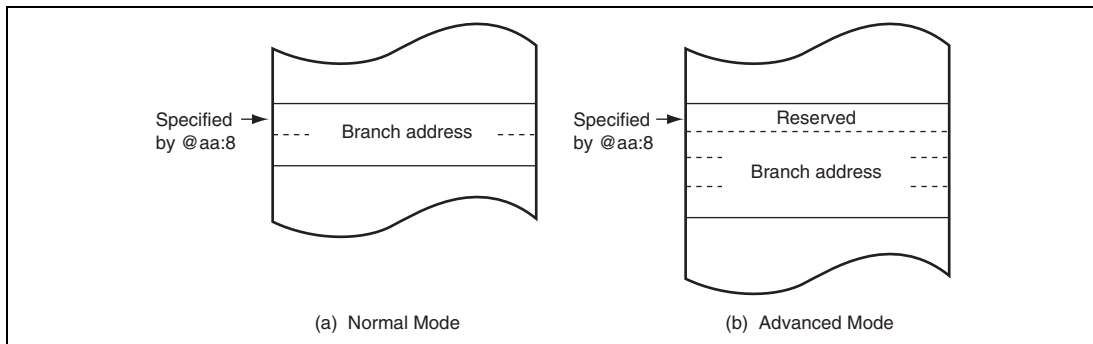


Figure 1.15 Branch Address Specification in Memory Indirect Mode

1.8.11 Extended Memory Indirect—@@vec:7

This mode can be used by the JMP and JSR instructions. The operand value is a branch address, which is the contents of a memory location pointed to by the following operation result: the sum of 7-bit data in the instruction code and the value of H'80 is multiplied by 2 or 4.

The address range to store a branch address is H'0100 to H'01FF in normal mode and H'000200 to H'0003FF in other modes. In assembler notation, an address to store a branch address is specified.

In normal mode, the memory location is pointed to by word-size data and the branch address is 16 bits long. In other modes, the memory location is pointed to by longword-size data. In middle or advanced mode, the first byte of the longword-size data is assumed to be all 0 (H'00).

1.8.12 Effective Address Calculation

Tables 1.14 and 1.15 show how effective addresses are calculated in each addressing mode. The lower bits of the effective address are valid and the upper bits are ignored (zero extended or sign extended) according to the CPU operating mode.

The valid bits in middle mode are as follows:

- The lower 16 bits of the effective address are valid and the upper 16 bits are sign-extended for the transfer and operation instructions.
- The lower 24 bits of the effective address are valid and the upper eight bits are zero-extended for the branch instructions.

Table 1.14 Effective Address Calculation for Transfer and Operation Instructions


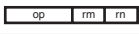
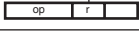

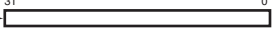
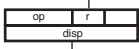
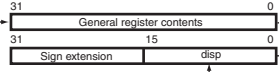
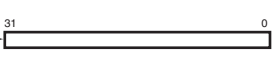

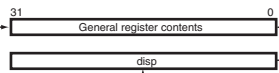
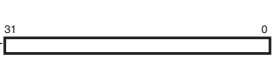
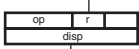
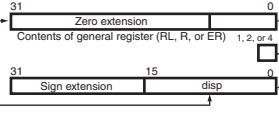
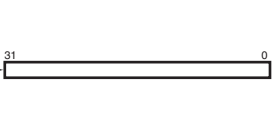

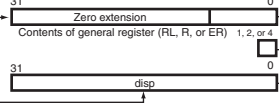
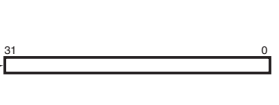

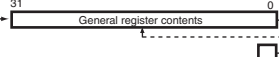
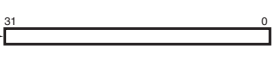

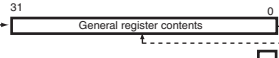
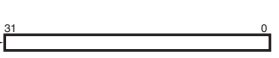
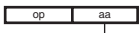


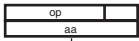
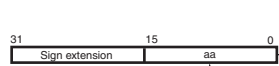
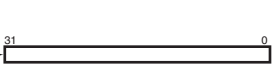
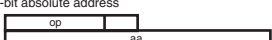
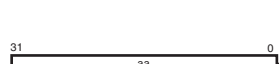





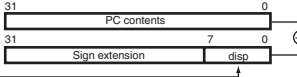

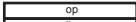
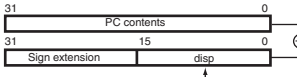


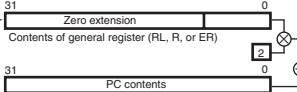


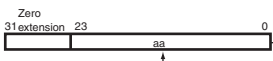


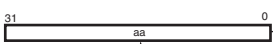


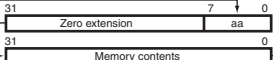


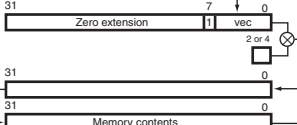

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
1	Immediate 		
2	Register direct 		
3	Register indirect 		
4	Register indirect with 16-bit displacement 		
	Register indirect with 32-bit displacement 		
5	Index register indirect with 16-bit displacement 		
	Index register indirect with 32-bit displacement 		
6	Register indirect with post-increment or post-decrement 		
	Register indirect with pre-increment or pre-decrement 		
7	8-bit absolute address 		
	16-bit absolute address 		
	32-bit absolute address 		

Table 1.15 Effective Address Calculation for Branch Instructions

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
1	Register indirect 		
2	Program-counter relative with 8-bit displacement 		
	Program-counter relative with 16-bit displacement 		
3	Program-counter relative with index register 		
4	24-bit absolute address 		
	32-bit absolute address 		
5	Memory indirect 		
6	Extended memory indirect 		

1.8.13 MOVA Instruction

The MOVA instruction stores the effective address in a general register.

1. Firstly, data is obtained by the addressing mode shown in item 2 of table 1.14.
2. Next, the effective address is calculated using the obtained data as the index by the addressing mode shown in item 5 of table 1.14. The obtained data is used instead of the general register shown in item 5 of table 1.14. For details, see section 2.2.66, MOVA MOVE Effective Address.

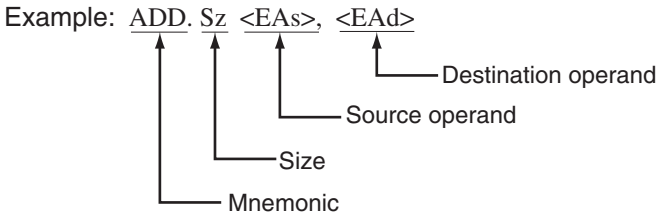
Section 2 Instruction Descriptions

2.1 Tables and Symbols

This section explains how to read the tables in section 2.2, Instruction Descriptions describing each instruction.

Mnemonic	Full Name
Gives the mnemonic of the instruction.	Gives the full name of the instruction.
<p>[1] Assembly-Language Format Indicates the assembly-language format of the instruction. (See section 2.1.1, Assembly-Language Format.)</p>	
<p>[2] Operation Describes the instruction in symbolic notation. (See section 2.1.2, Operation.)</p>	
<p>[3] Operand Size Indicates the available operand sizes.</p>	
<p>[4] Description Describes the operation of the instruction in detail.</p>	
<p>[5] Condition Code Indicates the effect of instruction execution on the flags in CCR. (See section 2.1.3, Condition Code.)</p>	
<p>[6] Available Registers Indicates which registers can be selected by the register field of the instruction code.</p>	
<p>[7] Available Addressing Modes Indicates which addressing modes can be selected.</p>	
<p>[8] Notes Gives notes concerning execution of the instruction.</p>	

2.1.1 Assembly-Language Format



The operand size is byte (B), word (W), or longword (L). Some instructions are restricted to a limited set of operand sizes.

The symbol <EA> indicates that two or more addressing modes can be used. The H8S/2600 CPU supports the 11 addressing modes listed next. Effective address calculation is described in section 1.8, Addressing Modes and Effective Address Calculation.

Symbol	Addressing Mode
Rn	Register direct
@ERn	Register indirect
@(d:2, ERn)/@(d:16, ERn)/ @(d:32, ERn)	Register indirect with displacement
@(d:16, RnL.B)/@(d:16, Rn.W)/ @(d:16, ERn.L)	Index register indirect with displacement
@(d:32, RnL.B)/@(d:32, Rn.W)/ @(d:32, ERn.L)	Index register indirect with displacement
@ERn+	Register indirect with post-increment
@-ERn	Register indirect with pre-decrement
@+ERn	Register indirect with pre-increment
@ERn-	Register indirect with post-decrement
@aa:8/@aa:16/@aa:24/@aa:32	Absolute address
#xx:3/#xx:4/#xx:8/#xx:16/#xx:32	Immediate
@(d:8, PC)/@(d:16, PC)	Program-counter relative
@(RnL.B, PC)/@(Rn.W, PC)/ @(ERn.L, PC)	Program-counter relative with index register
@@aa:8	Memory indirect
@@vec:7	Extended memory indirect

The suffixes :8, :16, :24, and :32 may be omitted. In particular, if the :8, :16, :24, or :32 designation is omitted in an absolute address or displacement, the assembler will optimize the length according to the value range.

Note: ":2" and ":3" in "#xx (:2)" and "#xx (:3)" indicate the specifiable bit length. Do not include (:2) or (:3) in the assembler notation.

Example: TRAPA #3

2.1.2 Operation

The symbols used in the operation descriptions are defined as follows.

Symbol	Description
Rd	General register (destination)*
Rs	General register (source)*
Rn	General register*
ERn	General register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended control register
CCR	Condition-code register
VBR	Vector base register
SBR	Short address base register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Add
-	Subtract
×	Multiply
÷	Divide
^	Logical AND

Symbol	Description
\vee	Logical OR
\oplus	Logical exclusive OR
\rightarrow	Transfer
\sim	Logical NOT (logical complement)
:8/:16/:24/:32	8-, 16-, 24-, or 32-bit length

Note: * General registers include 8-bit registers (R0H to R7H and R0L to R7L), 16-bit registers (R0 to R7 and E0 to E7), and 32-bit registers (ER0 to ER7).

2.1.3 Condition Code

The symbols used in the condition-code description are defined as follows.

Symbol	Meaning
\updownarrow	Changes according to the result of instruction execution
*	Undetermined (no guaranteed value)
0	Always cleared to 0
1	Always set to 1
—	Not affected by execution of the instruction
Δ	Varies depending on conditions; see the notes

2.1.4 Instruction Format

The symbols used in the instruction format descriptions are listed below.

Symbol	Meaning
IMM	Immediate data (2, 3, 8, 16, or 32 bits)
abs	Absolute address (8, 16, 24, or 32 bits)
disp	Displacement (8, 16, or 32 bits)
rs, rd, rn	Register field (4 bits). The symbols rs, rd, and rn correspond to operand symbols Rs, Rd, and Rn.
ers, erd, ern	Register field (3 bits). The symbols ers, erd, and ern correspond to operand symbols ERs, ERd, and ERn.

2.1.5 Register Specification

Address Register Specification: When a general register is used as an address register, the register is specified by a 3-bit register field (ers or erd).

Data Register Specification: A general register can be used as a 32-bit, 16-bit, or 8-bit data register.

When used as a 32-bit register, it is specified by a 3-bit register field (ers, erd, or ern).

When used as a 16-bit register, it is specified by a 4-bit register field (rs, rd, or m). The lower 3 bits of the register field specify the register number. The upper bit is set to 1 to specify an extended register (En) or cleared to 0 to specify a general register (Rn).

When used as an 8-bit register, it is specified by a 4-bit register field (rs, rd, or m). The lower 3 bits of the register field specify the register number. The upper bit is set to 1 to specify a low register (RnL) or cleared to 0 to specify a high register (RnH). The correspondence is shown next.

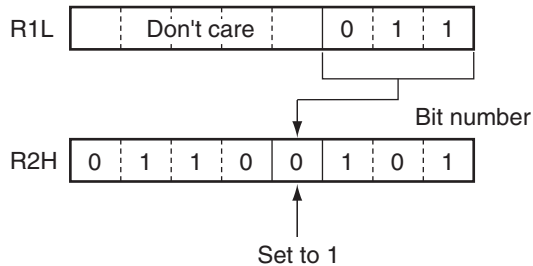
Address Register 32-Bit Register		16-Bit Register		8-Bit Register	
Register Field	General Register	Register Field	General Register	Register Field	General Register
000	ER0	0000	R0	0000	R0H
001	ER1	0001	R1	0001	R1H
.
.
111	ER7	0111	R7	0111	R7H
		1000	E0	1000	R0L
		1001	E1	1001	R1L
	
	
		1111	E7	1111	R7L

2.1.6 Bit Data Access in Bit Manipulation Instructions

Bit data is accessed as the n-th bit ($n = 0, 1, 2, 3, \dots, 7$) of byte data in a general register or the contents of a memory location. The bit number is given by 3-bit immediate data or the lower 3 bits of a general register.

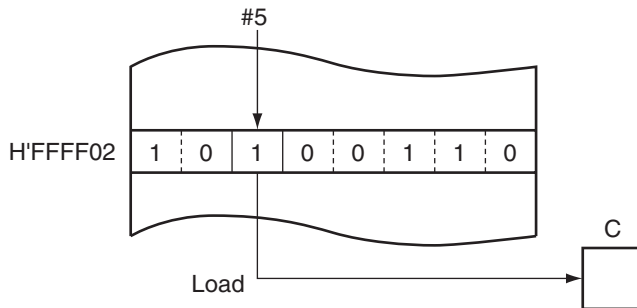
Example 1: To set bit 3 in R2H to 1

BSET R1L, R2H



Example 2: To load bit 5 at address H'FFFF02 into the bit accumulator

BLD #5, @H'FFFF02



The operand size and addressing mode of a bit manipulation instruction are as indicated for the data in a register or the contents of a memory location.

2.2 Instruction Descriptions

2.2.1 ADD

ADD Binary

[1] Assembly-Language Format

ADD.Sz <EAs>, <EAd>

[2] Operation

<EAd> + <EAs> → <EAd>

[3] Operand Size

Byte, word, longword

[4] Description

This instruction adds the source operand <EAs> to the contents of the destination location <EAd> (destination operand) and stores the result in the destination location.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	↓	—	↓	↓	↓	↓

H: Set to 1 if there is a carry at the 4th bit from the MSB (bit 27, bit 11, or bit 3); otherwise cleared to 0.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Set to 1 if an overflow occurs; otherwise cleared to 0.

C: Set to 1 if there is a carry at the MSB (bit 31, bit 15, or bit 7); otherwise cleared to 0.

[6] Available General Registers

Usage			General Register	
Source	Address register		ER0 to ER7	
	Index register		R0L to R7L, R0 to R7, ER0 to ER7	
	Data register	Byte		R0L to R7L, R0H to R7H
		Word		R0 to R7, E0 to E7
Longword			ER0 to ER7	
Destination	Address register		ER0 to ER7	
	Index register		R0L to R7L, R0 to R7, ER0 to ER7	
	Data register	Byte		R0L to R7L, R0H to R7H
		Word		R0 to R7, E0 to E7
Longword			ER0 to ER7	

[7] Available Addressing Modes

Source	Destination																	
	Rd	@ERd	@(d:2,ERd)	@(d:16,ERd)	@(d:32,ERd)	@ERd+	@ERd-	@+ERd	@-ERd	@(d:16,RdL.B)	@(d:16,Rd.W)	@(d:16,ERd.L)	@(d:32,RdL.B)	@(d:32,Rd.W)	@(d:32,ERd.L)	@aa:8	@aa:16	@aa:32
#x:3(1-7)	WL	W															W	W
#x:8	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
#x:16	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL		WL	WL
#x:32	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L		L	L
Rs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL
@ERs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:2,ERs)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:16,ERs)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:32,ERs)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@ERs+	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@ERs-	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@+ERs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@-ERs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:16,RsL.B)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:16,Rs.W)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:16,ERs.L)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:32,RsL.B)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:32,Rs.W)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:32,ERs.L)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@aa:8	B																	
@aa:16	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@aa:32	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL

2.2.2 ADDS**ADD with Sign extension****[1] Assembly-Language Format**

ADDS #1, ERd

ADDS #2, ERd

ADDS #4, ERd

[2] Operation

ERd + 1 → ERd

ERd + 2 → ERd

ERd + 4 → ERd

[3] Operand size

Longword

[4] Description

This instruction adds the immediate value 1, 2, or 4 to the contents of a 32-bit register ERd (destination operand). Unlike the ADD instruction, the condition code remains unchanged.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Destination	ER0 to ER7

[7] Available Addressing Mode

Usage	Addressing Mode
Destination	Register direct

2.2.3

ADDX**ADD with eXtend carry****[1] Assembly-Language Format**

ADDX.Sz <EAs>, <EAd>

[2] Operation

<EAd> + <EAs> + C → <EAd>

[3] Operand Size

Byte, word, longword

[4] Description

This instruction adds the source operand <EAs> and the carry flag to the contents of the destination location <EAd> (destination operand) and stores the result in the destination location.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	↕	—	↕	↕	↕	↕

H: Set to 1 if there is a carry at the 4th bit from the MSB (bit 27, bit 11, or bit 3); otherwise cleared to 0.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Previous value remains unchanged if the result is zero; otherwise cleared to 0.

V: Set to 1 if an overflow occurs; otherwise cleared to 0.

C: Set to 1 if there is a carry at the MSB (bit 31, bit 15, or bit 7); otherwise cleared to 0.

[6] Available General Registers

Usage		General Register	
Source	Address register		ER0 to ER7
	Data register	Byte	R0L to R7L, R0H to R7H
		Word	R0 to R7, E0 to E7
		Longword	ER0 to ER7
Destination	Address register		ER0 to ER7
	Data register	Byte	R0L to R7L, R0H to R7H
		Word	R0 to R7, E0 to E7
		Longword	ER0 to ER7

[7] Available Addressing Modes

Source	Destination																		
	Rd	@ERd	@(d:2,ERd)	@(d:16,ERd)	@(d:32,ERd)	@ERd+	@ERd-	@+ERd	@-ERd	@(d:16,RdL.B)	@(d:16,Rd.W)	@(d:16,ERd.L)	@(d:32,RdL.B)	@(d:32,Rd.W)	@(d:32,ERd.L)	@aa:8	@aa:16	@aa:32	
#x:3(1-7)																			
#x:8	B	B						B											
#x:16	W	W						W											
#x:32	L	L						L											
Rs	BWL	BWL						BWL											
@ERs	BWL	BWL																	
@(d:2,ERs)																			
@(d:16,ERs)																			
@(d:32,ERs)																			
@ERs+																			
@ERs-	BWL							BWL											
@+ERs																			
@-ERs																			
@(d:16,RsL.B)																			
@(d:16,Rs.W)																			
@(d:16,ERs.L)																			
@(d:32,RsL.B)																			
@(d:32,Rs.W)																			
@(d:32,ERs.L)																			
@aa:8																			
@aa:16																			
@aa:32																			

2.2.4

AND

AND logical

[1] Assembly-Language Format

AND.Sz <EAs>, <EAd>

[2] Operation

<EAd> \wedge (EAs) \rightarrow <EAd>

[3] Operand Size

Byte, word, longword

[4] Description

This instruction ANDs the source operand <EAs> with the contents of the destination location <EAd> (destination operand) and stores the result in the destination location.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↑	↑	0	—

H: Previous value remains unchanged.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Always cleared to 0.

C: Previous value remains unchanged.

[6] Available General Registers

Usage		General Register	
Source	Address register	ER0 to ER7	
	Index register	R0L to R7L, R0 to R7, ER0 to ER7	
	Data register	Byte	R0L to R7L, R0H to R7H
		Word	R0 to R7, E0 to E7
Longword		ER0 to ER7	
Destination	Address register	ER0 to ER7	
	Index register	R0L to R7L, R0 to R7, ER0 to ER7	
	Data register	Byte	R0L to R7L, R0H to R7H
		Word	R0 to R7, E0 to E7
Longword		ER0 to ER7	

[7] Available Addressing Modes

Source	Destination																	
	Rd	@ERd	@(d:2,ERd)	@(d:16,ERd)	@(d:32,ERd)	@ERd+	@ERd-	@+ERd	@-ERd	@(d:16,RdL.B)	@(d:16,Rd.W)	@(d:16,ERd.L)	@(d:32,RdL.B)	@(d:32,Rd.W)	@(d:32,ERd.L)	@aa:8	@aa:16	@aa:32
#x:3(1-7)																		
#x:8	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
#x:16	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL		WL	WL
#x:32	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L		L	L
Rs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL
@ERs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:2,ERs)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:16,ERs)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:32,ERs)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@ERs+	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@ERs-	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@+ERs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@-ERs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:16,RsL.B)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:16,Rs.W)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:16,ERs.L)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:32,RsL.B)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:32,Rs.W)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:32,ERs.L)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@aa:8	B																	
@aa:16	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@aa:32	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL

2.2.5 ANDC

AND Control register

[1] Assembly-Language Format

ANDC #xx:8, CCR

[2] Operation $CCR \wedge \#IMM \rightarrow CCR$ **[3] Operand Size**

Byte

[4] Description

This instruction ANDs the contents of the condition-code register (CCR) with immediate data and stores the result in CCR. No interrupt requests, including NMI, are accepted immediately after execution of this instruction.

[5] Condition Code

I	UI	H	U	N	Z	V	C
↓	↓	↓	↓	↓	↓	↓	↓

I: Stores the corresponding bit of the result.

UI: Stores the corresponding bit of the result.

H: Stores the corresponding bit of the result.

U: Stores the corresponding bit of the result.

N: Stores the corresponding bit of the result.

Z: Stores the corresponding bit of the result.

V: Stores the corresponding bit of the result.

C: Stores the corresponding bit of the result.

[6] Available Addressing Mode

Immediate

2.2.6 ANDC**AND Control register****[1] Assembly-Language Format**

ANDC #xx:8, EXR

[2] Operation $EXR \wedge \#IMM \rightarrow EXR$ **[3] Operand Size**

Byte

[4] Description

This instruction ANDs the contents of the extended control register (EXR) with immediate data and stores the result in EXR. No interrupt requests, including NMI, are accepted immediately after execution of this instruction.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available Addressing Mode

Immediate

2.2.7

BAND

Bit AND

[1] Assembly-Language Format

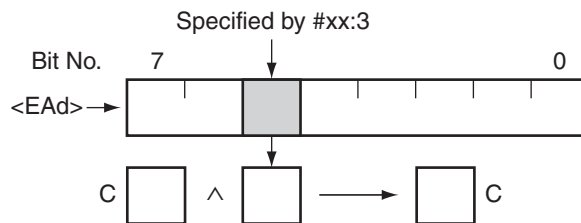
BAND #xx:3, <EAd>

[2] Operation $C \wedge (\text{bit No. of } \langle \text{EAd} \rangle) \rightarrow C$ **[3] Operand Size**

Byte

[4] Description

This instruction ANDs a specified bit in the destination operand with the carry flag and stores the result in the carry flag. The bit number is specified by 3-bit immediate data. The destination operand contents remain unchanged.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	↕

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Stores the result of the operation.

[6] Available General Registers

Usage	General Register
Destination	R0L to R7L, R0H to R7H
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
Destination	Register direct
	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

[8] Note

For details on the accessible ranges for @aa:8 or @aa:16, see the hardware manual for the corresponding product.

2.2.8

Bcc**Branch conditionally****[1] Assembly-Language Format**

Bcc disp

(cc: Condition field)

[2] Operation

if condition is true, then

PC + disp → PC

else

next

[3] Operand Size

—

[4] Description

If the condition specified in the condition field (cc) is satisfied, a displacement is added to the program counter (PC) and execution branches to the resulting address. If the condition is not satisfied, the next instruction is executed. The PC value used in the address calculation is the starting address of the instruction immediately following the Bcc instruction. The displacement is a signed 8-bit or 16-bit value. The branch destination address can be located in the range from -126 to +128 bytes or -32766 to +32768 bytes from the Bcc instruction.

Mnemonic	Meaning	cc	Condition	Signed/Unsigned*
BRA (BT)	Always (true)	0000	True	
BRN (BF)	Never (false)	0001	False	
BHI	High	0010	$C \vee Z = 0$	$X > Y$ (unsigned)
BLS	Low or Same	0011	$C \vee Z = 1$	$X \leq Y$ (unsigned)
BCC (BHS)	Carry Clear (High or Same)	0100	$C = 0$	$X \geq Y$ (unsigned)
BCS (BLO)	Carry Set (LOW)	0101	$C = 1$	$X < Y$ (unsigned)
BNE	Not Equal	0110	$Z = 0$	$X \neq Y$ (unsigned or signed)
BEQ	Equal	0111	$Z = 1$	$X = Y$ (unsigned or signed)
BVC	oVerflow Clear	1000	$V = 0$	
BVS	oVerflow Set	1001	$V = 1$	

Mnemonic	Meaning	cc	Condition	Signed/Unsigned*
BPL	PLus	1010	$N = 0$	
BMI	MInus	1011	$N = 1$	
BGE	Greater or Equal	1100	$N \oplus V = 0$	$X \geq Y$ (signed)
BLT	Less Than	1101	$N \oplus V = 1$	$X < Y$ (signed)
BGT	Greater Than	1110	$Z \vee (N \oplus V) = 0$	$X > Y$ (signed)
BLE	Less or Equal	1111	$Z \vee (N \oplus V) = 1$	$X \leq Y$ (signed)

Note: * If the immediately preceding instruction is a CMP instruction, X is the general register contents (destination operand) and Y is the source operand.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Notes

1. The branch destination address must be even.
2. In machine language, BRA, BRN, BCC, and BCS are identical to BT, BF, BHS, and BLO, respectively.

2.2.9

BCLR**Bit CLear****[1] Assembly-Language Format**

BCLR #xx:3, <EAd>

BCLR Rn, <EAd>

[2] Operation

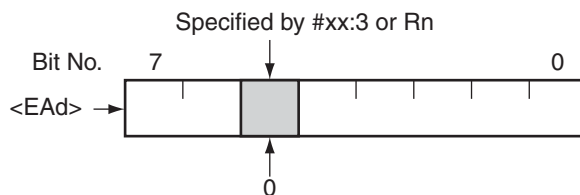
0 → (<bit No.> of <EAd>)

[3] Operand Size

Byte

[4] Description

This instruction clears a specified bit in the destination operand to 0. The bit number can be specified by 3-bit immediate data, or by the lower three bits of an 8-bit register Rn. The specified bit is not tested. (The condition code remains unchanged.)

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Bit number specification	R0L to R7L, R0H to R7H
Destination	R0L to R7L, R0H to R7H
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
	Register direct
Destination	Register direct
	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

[8] Note

For details on the accessible ranges for @aa:8 or @aa:16, see the hardware manual for the corresponding product.

2.2.10 BCLR/EQ

Bit CLear if EQual

[1] Assembly-Language Format

BCLR/EQ #xx:3, <EAd>

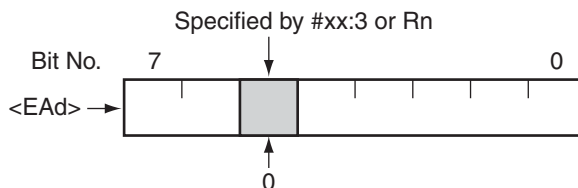
BCLR/EQ Rn, <EAd>

[2] Operationif equal ($Z = 1$), $0 \rightarrow$ (<bit No.> of <EAd>)**[3] Operand Size**

Byte

[4] Description

If the specified condition is satisfied, this instruction clears a specified bit in the destination operand to 0. The bit number can be specified by 3-bit immediate data, or by the lower three bits of an 8-bit register Rn. (The condition code remains unchanged.)

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Bit number specification	R0L to R7L, R0H to R7H
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
	Register direct
Destination	Register direct
	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

[8] Note

For details on the accessible ranges for @aa:8 or @aa:16, see the hardware manual for the corresponding product.

[6] Available General Registers

Usage	General Register
Bit number specification	R0L to R7L, R0H to R7H
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
	Register direct
Destination	Register direct
	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

[8] Note

For details on the accessible ranges for @aa:8 or @aa:16, see the hardware manual for the corresponding product.

2.2.12

BFLD

Bit Field Load

[1] Assembly-Language Format

BFLD #xx:8, <EAs>,Rd

[2] Operation

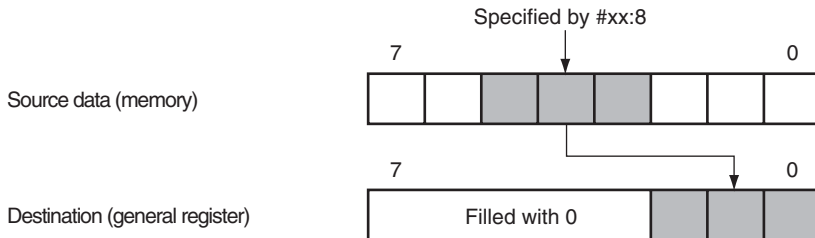
(<bit field> of <EAs>) → Rd

[3] Operand Size

Byte

[4] Description

This instruction loads a specified field in the source operand into an 8-bit register Rd. The specified field is right-aligned and the upper bits are filled with 0s. The bit field is specified by the bits in 8-bit immediate data which are set to 1. (The condition code remains unchanged.)

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Destination	R0L to R7L, R0H to R7H
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
Source	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)
Destination	Register direct

[8] Note

For details on the accessible ranges for @aa:8 or @aa:16, see the hardware manual for the corresponding product.

The immediate data that specifies a bit field is limited to consecutive bits with the value 1.

2.2.13 BFST

Bit Field Store

[1] Assembly-Language Format

BFST Rs, #xx:8, <EAd>

[2] Operation

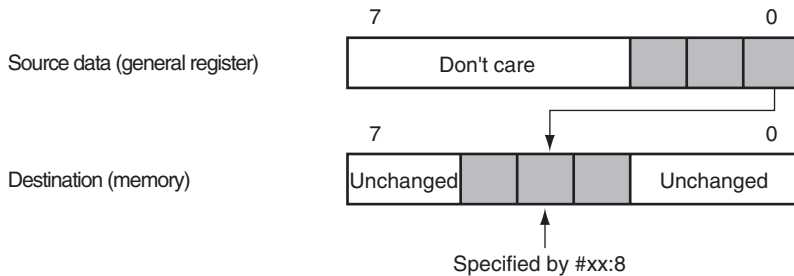
Rs → (<bit field> of <EAd>)

[3] Operand Size

Byte

[4] Description

This instruction stores the contents of an 8-bit register (Rs) (data is right-aligned) into a specified field in the destination operand. The bit field is specified by the bits in 8-bit immediate data which are set to 1. (The condition code remains unchanged.)

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Source	R0L to R7L, R0H to R7H
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
Source	Register direct
Destination	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

[8] Note

For details on the accessible ranges for @aa:8 or @aa:16, see the hardware manual for the corresponding product.

The immediate data that specifies a bit field is limited to consecutive bits with the value 1.

2.2.14 BIAND

Bit Invert AND

[1] Assembly-Language Format

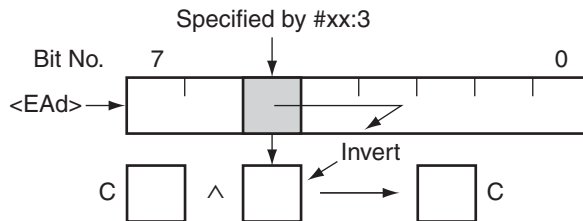
BIAND #xx:3, <EAd>

[2] Operation $C \wedge [\sim (\text{bit No. of } \langle \text{EAd} \rangle)] \rightarrow C$ **[3] Operand Size**

Byte

[4] Description

This instruction ANDs the inverse of a specified bit in the destination operand with the carry flag and stores the result in the carry flag. The bit number is specified by 3-bit immediate data. The destination operand contents remain unchanged.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	↕

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Stores the result of the operation.

[6] Available General Registers

Usage	General Register
Destination	R0L to R7L, R0H to R7H
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
Destination	Register direct
	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

[8] Note

For details on the accessible ranges for @aa:8 or @aa:16, see the hardware manual for the corresponding product.

2.2.15 BILD

Bit Invert Load

[1] Assembly-Language Format

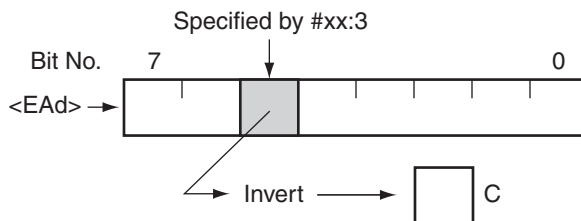
BILD #xx:3, <EAd>

[2] Operation \sim (<bit No.> of <EAd>) \rightarrow C**[3] Operand Size**

Byte

[4] Description

This instruction loads the inverse of a specified bit from the destination operand into the carry flag. The bit number is specified by 3-bit immediate data. The destination operand contents remain unchanged.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	↕

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Loaded with the inverse of a specified bit.

[6] Available General Registers

Usage	General Register
Destination	R0L to R7L, R0H to R7H
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
Destination	Register direct
	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

[8] Note

For details on the accessible ranges for @aa:8 or @aa:16, see the hardware manual for the corresponding product.

2.2.16 BIOR

Bit Invert inclusive OR

[1] Assembly-Language Format

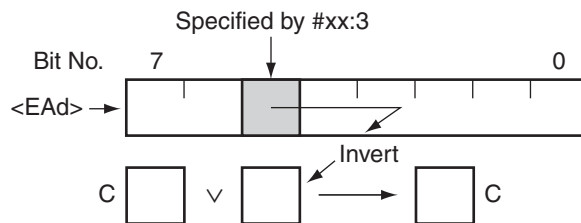
BIOR #xx:3, <EAd>

[2] Operation $C \vee [\sim (\text{<bit No.> of <EAd>})] \rightarrow C$ **[3] Operand Size**

Byte

[4] Description

This instruction ORs the inverse of a specified bit in the destination operand with the carry flag and stores the result in the carry flag. The bit number is specified by 3-bit immediate data. The destination operand contents remain unchanged.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	↕

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Stores the result of the operation.

[6] Available General Registers

Usage	General Register
Destination	R0L to R7L, R0H to R7H
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
Destination	Register direct
	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

[8] Note

For details on the accessible ranges for @aa:8 or @aa:16, see the hardware manual for the corresponding product.

2.2.17 BIST

Bit Invert Store

[1] Assembly-Language Format

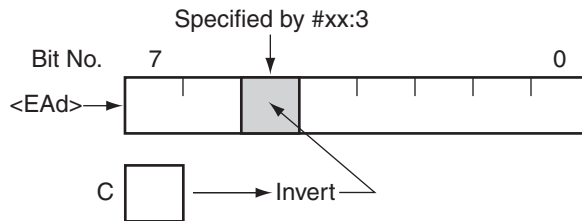
BIST #xx:3, <EAd>

[2] Operation $\sim C \rightarrow$ (<bit No.> of <EAd>)**[3] Operand Size**

Byte

[4] Description

This instruction stores the inverse of the carry flag in a specified bit location in the destination operand. The bit number is specified by 3-bit immediate data. Other bits in the destination operand remain unchanged.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Destination	R0L to R7L, R0H to R7H
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
Destination	Register direct
	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

[8] Note

For details on the accessible ranges for @aa:8 or @aa:16, see the hardware manual for the corresponding product.

2.2.18 BISTZ

Bit Invert STore Zero flag

[1] Assembly-Language Format

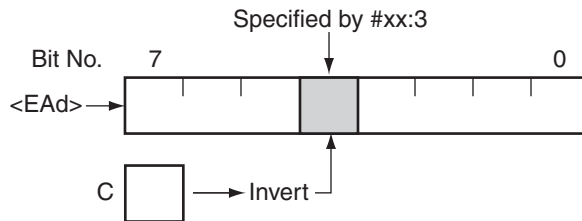
BISTZ #xx:3, <EAd>

[2] Operation $\sim Z \rightarrow$ (<bit No.> of <EAd>)**[3] Operand Size**

Byte

[4] Description

This instruction stores the inverse of the zero flag in a specified bit location in the destination operand. The bit number is specified by 3-bit immediate data. Other bits in the destination operand remain unchanged.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Stores the result of the operation.

[6] Available General Registers

Usage	General Register
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
Destination	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

[8] Note

For details on the accessible ranges for @aa:8 or @aa:16, see the hardware manual for the corresponding product.

2.2.19 BIXOR

Bit Invert eXclusive OR

[1] Assembly-Language Format

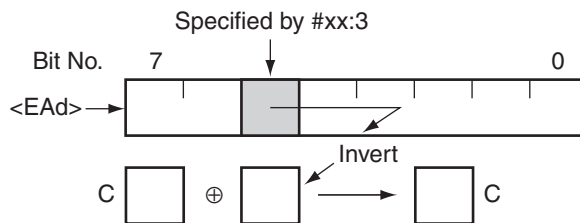
BIXOR #xx:3, <EAd>

[2] Operation $C \oplus [\sim (\text{bit No. of } \langle \text{EAd} \rangle)] \rightarrow C$ **[3] Operand Size**

Byte

[4] Description

This instruction exclusively ORs the inverse of a specified bit in the destination operand with the carry flag and stores the result in the carry flag. The bit number is specified by 3-bit immediate data. The destination operand contents remain unchanged.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	↕

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Stores the result of the operation.

[6] Available General Registers

Usage	General Register
Destination	R0L to R7L, R0H to R7H
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
Destination	Register direct
	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

[8] Note

For details on the accessible ranges for @aa:8 or @aa:16, see the hardware manual for the corresponding product.

2.2.20 BLD**Bit Load****[1] Assembly-Language Format**

BLD #xx:3, <EAd>

[2] Operation

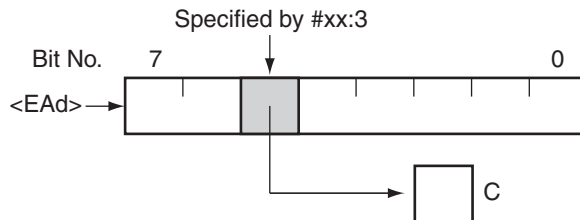
(<bit No.> of <EAd>) → C

[3] Operand Size

Byte

[4] Description

This instruction loads a specified bit from the destination operand into the carry flag. The bit number is specified by 3-bit immediate data. The destination operand contents remain unchanged.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	↑↓

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: The content of a specified bit is loaded.

[6] Available General Registers

Usage	General Register
Destination	R0L to R7L, R0H to R7H
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
Destination	Register direct
	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

[8] Note

For details on the accessible ranges for @aa:8 or @aa:16, see the hardware manual for the corresponding product.

2.2.21 BNOT**Bit NOT****[1] Assembly-Language Format**

BNOT #xx:3, <EAd>

BNOT Rn, <EAd>

[2] Operation

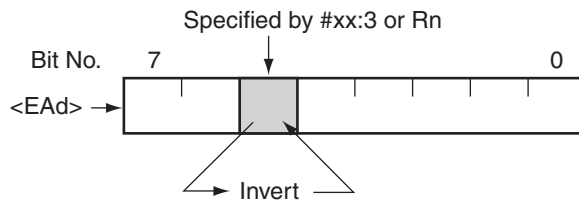
~ (<bit No.> of <EAd>) → (<bit No.> of <EAd>)

[3] Operand Size

Byte

[4] Description

This instruction inverts a specified bit in the destination operand. The bit number is specified by 3-bit immediate data or by the lower 3 bits of an 8-bit register (Rn). The specified bit is not tested. The condition code remains unchanged.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Bit number specification	R0L to R7L, R0H to R7H
Destination	R0L to R7L, R0H to R7H
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
	Register direct
Destination	Register direct
	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

[8] Note

For details on the accessible ranges for @aa:8 or @aa:16, see the hardware manual for the corresponding product.

2.2.22 BOR

Bit inclusive OR

[1] Assembly-Language Format

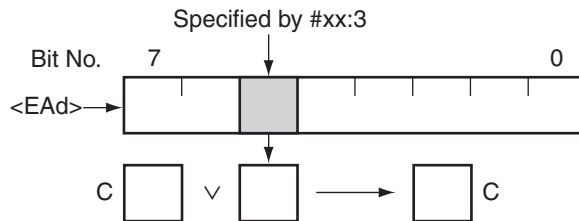
BOR #xx:3, <EAd>

[2] Operation $C \vee (\text{<bit No.> of <EAd>}) \rightarrow C$ **[3] Operand Size**

Byte

[4] Description

This instruction ORs a specified bit in the destination operand with the carry flag and stores the result in the carry flag. The bit number is specified by 3-bit immediate data. The destination operand contents remain unchanged.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	↕

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Stores the result of the operation.

[6] Available General Registers

Usage	General Register
Destination	R0L to R7L, R0H to R7H
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
Destination	Register direct
	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

[8] Note

For details on the accessible ranges for @aa:8 or @aa:16, see the hardware manual for the corresponding product.

2.2.23**BRA****BRAnch always****[1] Assembly-Language Format**

BRA Rn.Sz

Examples: BRA R0L.B
 BRA R0.W
 BRA ER0.L

[2] Operation

PC + Rn → PC

[3] Operand Size

—

[4] Description

This instruction branches unconditionally to an address which is the result of the sum of the PC and Rn values. The PC value used in the address calculation is the start address of the instruction immediately after this instruction.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage		General Register
Data register	Byte	R0L to R7L
	Word	R0 to R7
	Longword	ER0 to ER7

[7] Notes

1. The branch destination address must be even.
2. For details on BRA disp, see section 2.2.8, Bcc Branch conditionally.

2.2.24 BRA/S**BRAnch always with delay Slot****[1] Assembly-Language Format**

BRA/S disp

[2] Operation

PC + disp → PC

[3] Operand Size

—

[4] Description

This instruction branches unconditionally to a specified address after the next instruction (delay slot) is executed. The next instruction should be a 1-word instruction other than the block transfer, branch, TRAPA, RTE, RTE/L, and SLEEP instructions. The displacement is a signed 8-bit value. The branch destination address can be located in the range from -126 to +128 bytes from this instruction.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Notes

1. The branch destination address must be even.
2. Interrupts are not accepted between this instruction and the next instruction.

2.2.25 BRA/BC

BRANch if Bit Cleared

[1] Assembly-Language Format

BRA/BC #xx:3, <EAs>, disp

[2] Operation

if bit cleared

PC + disp → PC

else

next

[3] Operand Size

Byte

[4] Description

If a specified bit in the source operand is cleared to 0, a displacement is added to the program counter (PC) and execution branches to the resulting address. If the specified bit is set to 1, the next instruction is executed. The PC value used in the address calculation is the start address of the instruction immediately after this instruction. The displacement is a signed 8-bit or 16-bit value. The signed 8- or 16-bit displacement is capable of specifying branch destination addresses in the range from -128 to +126 bytes or from -32768 to +32766 bytes relative to the address that follows this instruction.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
Source	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

2.2.26

BRA/BS

BRANch if Bit Set

[1] Assembly-Language Format

BRA/BS #xx:3, <EAs>, disp

[2] Operation

if bit set

PC + disp → PC

else

next

[3] Operand Size

Byte

[4] Description

If a specified bit in the source operand is set to 1, a displacement is added to the program counter (PC) and execution branches to the resulting address. If the specified bit is cleared to 0, the next instruction is executed. The PC value used in the address calculation is the start address of the instruction immediately after this instruction. The displacement is a signed 8-bit or 16-bit value. The signed 8- or 16-bit displacement is capable of specifying branch destination addresses in the range from -128 to +126 bytes or from -32768 to +32766 bytes relative to the address that follows this instruction.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
Source	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

2.2.27

BSET**Bit SET****[1] Assembly-Language Format**

BSET #xx:3, <EAd>

BSET Rn, <EAd>

[2] Operation

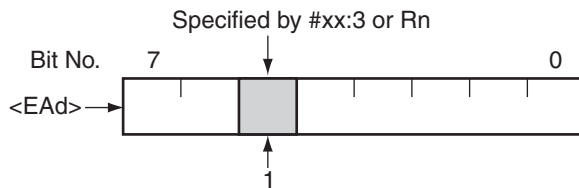
1 → (<bit No.> of <EAd>)

[3] Operand Size

Byte

[4] Description

This instruction sets a specified bit in the destination operand to 1. The bit number can be specified by 3-bit immediate data, or by the lower three bits of an 8-bit register (Rn). The specified bit is not tested. The condition code remains unchanged.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Bit number specification	R0L to R7L, R0H to R7H
Destination	R0L to R7L, R0H to R7H
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
	Register direct
Destination	Register direct
	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

[8] Note

For details on the accessible ranges for @aa:8 or @aa:16, see the hardware manual for the corresponding product.

2.2.28 BSET/EQ

Bit SET if Equal

[1] Assembly-Language Format

BSET/EQ #xx:3, <EAd>

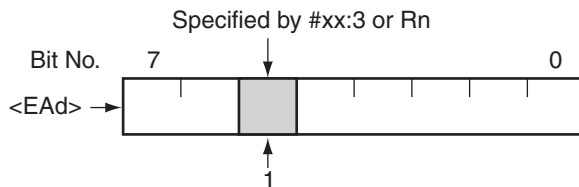
BSET/EQ Rn, <EAd>

[2] Operationif equal ($Z = 1$), $1 \rightarrow$ (<bit No.> of <EAd>)**[3] Operand Size**

Byte

[4] Description

If the specified condition is satisfied, this instruction sets a specified bit in the destination operand to 1. The bit number can be specified by 3-bit immediate data, or by the lower three bits of an 8-bit register (Rn). The specified bit is not tested. The condition code remains unchanged.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Bit number specification	R0L to R7L, R0H to R7H
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
	Register direct
Destination	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

[8] Note

For details on the accessible ranges for @aa:8 or @aa:16, see the hardware manual for the corresponding product.

2.2.29

BSET/NE

Bit SET if Not Equal

[1] Assembly-Language Format

BSET/NE #xx:3, <EAd>

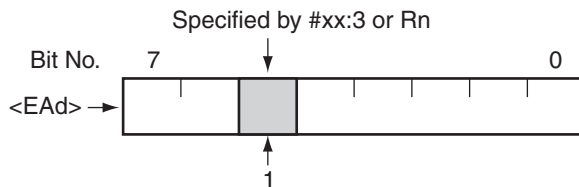
BSET/NE Rn, <EAd>

[2] Operationif not equal ($Z = 0$), 1 \rightarrow (<bit No.> of <EAd>)**[3] Operand Size**

Byte

[4] Description

If the specified condition is satisfied, this instruction sets a specified bit in the destination operand to 1. The bit number can be specified by 3-bit immediate data, or by the lower three bits of an 8-bit register (Rn). The specified bit is not tested. The condition code remains unchanged.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Bit number specification	R0L to R7L, R0H to R7H
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
	Register direct
Destination	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

[8] Note

For details on the accessible ranges for @aa:8 or @aa:16, see the hardware manual for the corresponding product.

2.2.30 BSR**Branch to SubRoutine**

[1] Assembly-Language Format

BSR <EA>

Examples: BSR Label
 BSR ROL.B**[2] Operation**

PC → @-SP

Effective address → PC

[3] Operand Size

—

[4] Description

This instruction pushes the program counter (PC) value onto the stack as a restart address, then branches to a specified effective address. The PC value used in the address calculation is the start address of the instruction immediately after this instruction. The PC value pushed onto the stack is the start address of the instruction immediately after this instruction.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register	
Data register	Byte	R0L to R7L
	Word	R0 to R7
	Longword	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Branch address	Program-counter relative (8-bit displacement)
	Program-counter relative (16-bit displacement)
	Program-counter relative with index register (8-bit index)
	Program-counter relative with index register (16-bit index)
	Program-counter relative with index register (32-bit index)

2.2.31 BSR/BC**Branch to SubRoutine if Bit Cleared****[1] Assembly-Language Format**

BSR/BC #xx:3, <EAs>, disp

[2] Operation

if bit cleared

PC → @SP

PC + disp → PC

else

next

[3] Operand Size

Byte

[4] Description

If a specified bit in the source operand is cleared to 0, a displacement is added to the program counter (PC) and execution branches to the subroutine. If the specified bit is set to 1, the next instruction is executed. The PC value used in the address calculation is the start address of the instruction immediately after this instruction. The displacement is a signed 16-bit value. The branch destination address can be located in the range from -32766 to +32768 bytes from this instruction.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
Source	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

2.2.32 BSR/BS**Branch to SubRoutine if Bit Set****[1] Assembly-Language Format**

BSR/BS #xx:3, <EAs>, disp

[2] Operation

if bit set

PC → @-SP

PC + disp → PC

else

next

[3] Operand Size

Byte

[4] Description

If a specified bit in the source operand is set to 1, a displacement is added to the program counter (PC) and execution branches to the subroutine. If the specified bit is cleared to 0, the next instruction is executed. The PC value used in the address calculation is the start address of the instruction immediately after this instruction. The displacement is a signed 16-bit value. The branch destination address can be located in the range from -32766 to +32768 bytes from this instruction.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
Source	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

2.2.33 **BST****Bit Store****[1] Assembly-Language Format**

BST #xx:3, <EAd>

[2] Operation

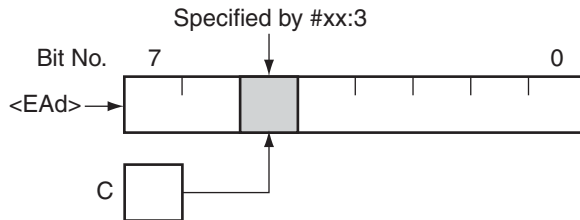
C → (<bit No.> of <EAd>)

[3] Operand Size

Byte

[4] Description

This instruction stores the carry flag in a specified bit location in the destination operand. The bit number is specified by 3-bit immediate data. Other bits in the destination operand remain unchanged.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Destination	R0L to R7L, R0H to R7H
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
Destination	Register direct
	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

[8] Note

For details on the accessible ranges for @aa:8 or @aa:16, see the hardware manual for the corresponding product.

2.2.34 **BSTZ****Bit STore Zero flag****[1] Assembly-Language Format**

BSTZ #xx:3, <EAd>

[2] Operation

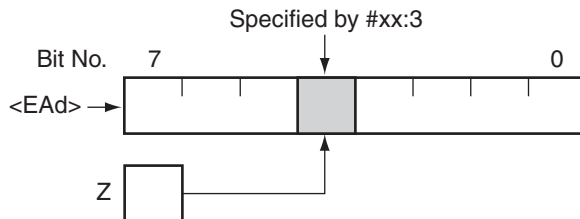
Z → (<bit No.> of <EAd>)

[3] Operand Size

Byte

[4] Description

This instruction stores the zero flag in a specified bit location in the destination operand. The bit number is specified by 3-bit immediate data. Other bits in the destination operand remain unchanged.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
Destination	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

[8] Note

For details on the accessible ranges for @aa:8 or @aa:16, see the hardware manual for the corresponding product.

2.2.35

BTST

Bit TeST

[1] Assembly-Language Format

BTST #xx:3, <EAd>

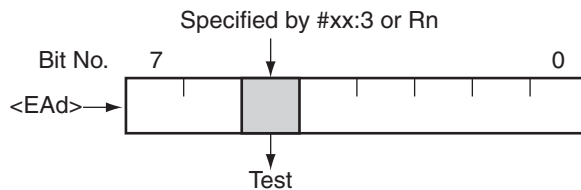
BTST Rn, <EAd>

[2] Operation \sim (<bit No.> of <EAd>) \rightarrow Z**[3] Operand Size**

Byte

[4] Description

This instruction tests a specified bit in the destination operand and sets or clears the zero flag according to the result. The bit number can be specified by 3-bit immediate data, or by the lower three bits of an 8-bit register (Rn). The destination operand contents remain unchanged.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	—	↕	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Set to 1 if a specified bit is zero; otherwise cleared to 0.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Bit number specification	R0L to R7L, R0H to R7H
Destination	R0L to R7L, R0H to R7H
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
Destination	Register direct
	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

[8] Note

For details on the accessible ranges for @aa:8 or @aa:16, see the hardware manual for the corresponding product.

2.2.36 **BXOR****Bit eXclusive OR****[1] Assembly-Language Format**

BXOR #xx:3, <EAd>

[2] Operation

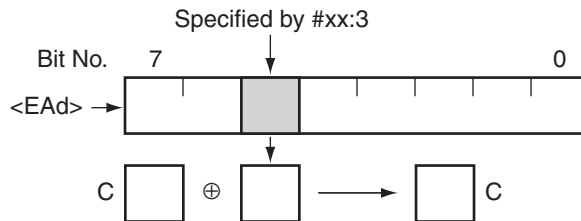
$C \oplus (\text{bit No. of } \langle \text{EAd} \rangle) \rightarrow C$

[3] Operand Size

Byte

[4] Description

This instruction exclusively ORs a specified bit in the destination operand with the carry flag and stores the result in the carry flag. The bit number is specified by 3-bit immediate data. The destination operand contents remain unchanged.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	↕

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Stores the result of the operation.

[6] Available General Registers

Usage	General Register
Destination	R0L to R7L, R0H to R7H
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Bit number specification	Immediate
Destination	Register direct
	Register indirect
	Absolute address (8 bits)
	Absolute address (16 bits)
	Absolute address (32 bits)

[8] Note

For details on the accessible ranges for @aa:8 or @aa:16, see the hardware manual for the corresponding product.

2.2.37

CLRMAC

CLear MAC register

[1] Assembly-Language Format

CLRMAC

[2] Operation

0 → MACH, MACL

[3] Operand Size

—

[4] Description

This instruction simultaneously clears registers MACH and MACL. It is available only when the multiplier is supported.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Note

Execution of this instruction also clears the overflow flag in the multiplier to 0.

2.2.38 CMP**CoMPare****[1] Assembly-Language Format**

CMP.Sz <EAs>, EAd

[2] Operation

<EAd> – <EAs>, set/clear CCR

[3] Operand Size

Byte, word, longword

[4] Description

This instruction subtracts the source operand <EAs> from the contents of the destination location <EAd> (destination operand) and sets or clears the condition code according to the result. The contents of the destination location remain unchanged.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	↕	—	↕	↕	↕	↕

H: Set to 1 if there is a borrow at the 4th bit from the MSB (bit 27, bit 11, or bit 3); otherwise cleared to 0.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Set to 1 if an overflow occurs; otherwise cleared to 0.

C: Set to 1 if there is a borrow at the MSB (bit 31, bit 15, or bit 7); otherwise cleared to 0.

[6] Available General Registers

Usage	General Register	
Address register	ER0 to ER7	
Index register	R0L to R7L, R0 to R7, ER0 to ER7	
Data register	Byte	R0L to R7L, R0H to R7H
	Word	R0 to R7, E0 to E7
	Longword	ER0 to ER7

[7] Available Addressing Modes

Source	Destination																	
	Rd	@ERd	@(d:2,ERd)	@(d:16,ERd)	@(d:32,ERd)	@ERd+	@ERd-	@+ERd	@-ERd	@(d:16,RdL.B)	@(d:16,Rd.W)	@(d:16,ERd.L)	@(d:32,RdL.B)	@(d:32,Rd.W)	@(d:32,ERd.L)	@aa:8	@aa:16	@aa:32
#x:3(1-7)	WL	W															W	W
#x:8	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
#x:16	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL
#x:32	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
Rs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL
@ERs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:2,ERs)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:16,ERs)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:32,ERs)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@ERs+	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@ERs-	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@+ERs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@-ERs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:16,RsL.B)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:16,Rs.W)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:16,ERs.L)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:32,RsL.B)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:32,Rs.W)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:32,ERs.L)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@aa:8	B																	
@aa:16	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@aa:32	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL

2.2.39

DAA

Decimal Adjust Add

[1] Assembly-Language Format

DAA Rd

[2] Operation

Rd (decimal adjust) → Rd

[3] Operand Size

Byte

[4] Description

Given that the result of an addition operation performed by an ADD.B or ADDX.B instruction on 4-bit BCD data is contained in an 8-bit register (Rd) and the carry and half-carry flags, the DAA instruction adjusts the contents of the 8-bit register (Rd) (destination operand) by adding H'00, H'06, H'60, or H'66 according to the table below.

C Flag before Adjustment	Upper 4 Bits before Adjustment	H Flag before Adjustment	Lower 4 Bits before Adjustment	Value Added (Hexadecimal)	C Flag after Adjustment
0	0 to 9	0	0 to 9	00	0
0	0 to 8	0	A to F	06	0
0	0 to 9	1	0 to 3	06	0
0	A to F	0	0 to 9	60	1
0	9 to F	0	A to F	66	1
0	A to F	1	0 to 3	66	1
1	0 to 2	0	0 to 9	60	1
1	0 to 2	0	A to F	66	1
1	0 to 3	1	0 to 3	66	1

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	*	—	↕	↕	*	↕

H: Undetermined (no guaranteed value).

N: Set to 1 if the adjusted result is negative; otherwise cleared to 0.

Z: Set to 1 if the adjusted result is zero; otherwise cleared to 0.

V: Undetermined (no guaranteed value).

C: Set to 1 if there is a carry at bit 7; otherwise remain unchanged.

[6] Available General Registers

Usage	General Register
Destination	R0L to R7L, R0H to R7H

[7] Available Addressing Mode

Usage	Addressing Mode
Destination	Register direct

[8] Note

Valid results (8-bit register Rd contents, and C, V, Z, N, and H flags) are not assured if this instruction is executed under conditions other than those described above.

2.2.40 DAS**Decimal Adjust Subtract****[1] Assembly-Language Format**

DAS Rd

[2] Operation

Rd (decimal adjust) → Rd

[3] Operand Size

Byte

[4] Description

Given that the result of a subtraction operation performed by a SUB.B, SUBX.B, or NEG.B instruction on 4-bit BCD data is contained in an 8-bit register Rd and the carry and half-carry flags, the DAS instruction adjusts the contents of the 8-bit register Rd (destination operand) by adding H'00, H'FA, H'A0, or H'9A according to the table below.

C Flag before Adjustment	Upper 4 Bits before Adjustment	H Flag before Adjustment	Lower 4 Bits before Adjustment	Value Added (Hexadecimal)	C Flag after Adjustment
0	0 to 9	0	0 to 9	00	0
0	0 to 8	1	6 to F	FA	0
1	7 to F	0	0 to 9	A0	1
1	6 to F	1	6 to F	9A	1

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	*	—	↕	↕	*	—

H: Undetermined (no guaranteed value).

N: Set to 1 if the adjusted result is negative; otherwise cleared to 0.

Z: Set to 1 if the adjusted result is zero; otherwise cleared to 0.

V: Undetermined (no guaranteed value).

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Destination	R0L to R7L, R0H to R7H

[7] Available Addressing Mode

Usage	Addressing Mode
Destination	Register direct

[8] Note

Valid results (8-bit register Rd contents, and C, V, Z, N, and H flags) are not assured if this instruction is executed under conditions other than those described above.

2.2.41 DEC (B)**DECrement****[1] Assembly-Language Format**

DEC.B Rd

[2] Operation $Rd - 1 \rightarrow Rd$ **[3] Operand Size**

Byte

[4] Description

This instruction subtracts 1 from the contents of an 8-bit register (Rd) (destination operand) and stores the result in the 8-bit register (Rd).

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↕	↕	↕	—

H: Previous value remains unchanged.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Set to 1 if an overflow occurs; otherwise cleared to 0.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Destination	R0L to R7L, R0H to R7H

[7] Available Addressing Mode

Usage	Addressing Mode
Destination	Register direct

[8] Note

An overflow is caused by the operation $H'80 - 1 \rightarrow H'7F$.

2.2.42 DEC (W)**DECrement****[1] Assembly-Language Format**

DEC.W #1, Rd

DEC.W #2, Rd

[2] Operation

Rd - 1 → Rd

Rd - 2 → Rd

[3] Operand Size

Word

[4] Description

This instruction subtracts the immediate value 1 or 2 from the contents of a 16-bit register (Rd) (destination operand) and stores the result in the 16-bit register (Rd).

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↑	↑	↑	—

H: Previous value remains unchanged.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Set to 1 if an overflow occurs; otherwise cleared to 0.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Destination	R0 to R7, E0 to E7

[7] Available Addressing Mode

Usage	Addressing Mode
Destination	Register direct

[8] Note

An overflow is caused by the operation $H'8000 - 1 \rightarrow H'7FFF$, $H'8000 - 2 \rightarrow H'7FFE$, or $H'8001 - 2 \rightarrow H'7FFF$.

2.2.43 DEC (L)**DECrement****[1] Assembly-Language Format**

DEC.L #1, ERd

DEC.L #2, ERd

[2] Operation

ERd - 1 → ERd

ERd - 2 → ERd

[3] Operand Size

Longword

[4] Description

This instruction subtracts the immediate value 1 or 2 from the contents of a 32-bit register (ERd) (destination operand) and stores the result in the 32-bit register (ERd).

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↕	↕	↕	—

H: Previous value remains unchanged.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Set to 1 if an overflow occurs; otherwise cleared to 0.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Destination	ER0 to ER7

[7] Available Addressing Mode

Usage	Addressing Mode
Destination	Register direct

[8] Note

An overflow is caused by the operation $H'80000000 - 1 \rightarrow H'7FFFFFFF$, $H'80000000 - 2 \rightarrow H'7FFFFFFE$, or $H'80000001 - 2 \rightarrow H'7FFFFFFF$.

2.2.44 DIVS**DIVide as Signed****[1] Assembly-Language Format**

DIVS.W/L Rs, Rd

DIVS.W/L #xx:4, Rd

[2] Operation $Rd \div Rs \rightarrow Rd$ $Rd \div \#IMM \rightarrow Rd$ **[3] Operand Size**

Word, longword

[4] Description

This instruction divides the contents of a 16- or 32-bit register (Rd) by the contents of a 16- or 32-bit register (Rs) or a 4-bit immediate value and stores the result in the 16- or 32-bit register Rd. The division is signed. Valid results are not assured if division by zero is attempted or an overflow occurs. A 4-bit immediate data in the source operand is extended to a 16- or 32-bit unsigned data.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↕	↕	—	—

H: Previous value remains unchanged.

N: Set to 1 if the quotient is negative; otherwise cleared to 0.

Z: Set to 1 if the divisor is zero; otherwise cleared to 0.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register	
Source	Word	R0 to R7, E0 to E7
	Longword	ER0 to ER7
Destination	Word	R0 to R7, E0 to E7
	Longword	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Source	Register direct
	4-bit immediate
Destination	Register direct

[8] Note

The N flag is set to 1 when the signs of the dividend and divisor are different and cleared to 0 when they are the same. Accordingly, the N flag will be set to 1 in some cases where the quotient is 0.

2.2.45 DIVU**DIVide as Unsigned****[1] Assembly-Language Format**

DIVU.W/L Rs, Rd

DIVU.W/L #xx:4, Rd

[2] Operation $Rd \div Rs \rightarrow Rd$ $Rd \div \#IMM \rightarrow Rd$ **[3] Operand Size**

Word, longword

[4] Description

This instruction divides the contents of a 16- or 32-bit register (Rd) by the contents of a 16- or 32-bit register (Rs) or a 4-bit immediate value and stores the result in the 16- or 32-bit register Rd. The division is unsigned. Valid results are not assured if division by zero is attempted. A 4-bit immediate data in the source operand is extended to a 16- or 32-bit unsigned data.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↕	↕	—	—

H: Previous value remains unchanged.

N: Set to 1 if the divisor is negative; otherwise cleared to 0.

Z: Set to 1 if the divisor is zero; otherwise cleared to 0.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage		General Register
Source	Word	R0 to R7, E0 to E7
	Longword	ER0 to ER7
Destination	Word	R0 to R7, E0 to E7
	Longword	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Source	Register direct
	4-bit immediate
Destination	Register direct

[8] Note

The N flag is set to 1 when the signs of the dividend and divisor are different and cleared to 0 when they are the same. Accordingly, the N flag will be set to 1 in some cases where the quotient is 0.

2.2.46 DIVXS**DIVide eXtend as Signed****[1] Assembly-Language Format**

DIVXS.B/W Rs, Rd

DIVXS.B/W #xx:4, Rd

[2] Operation $Rd \div Rs \rightarrow Rd$ $Rd \div \#IMM \rightarrow Rd$ **[3] Operand Size**

Byte, word

[4] Description

This instruction divides the lower 16 or 32 bits of the contents of the destination register (Rd) by the source operand and stores the 16- or 32-bit result in the destination register. The division is signed. The operation performed on 8-bit operands is 16 bits \div 8 bits \rightarrow 8-bit quotient and 8-bit remainder, and the operation performed on 16-bit operands is 32 bits \div 16 bits \rightarrow 16-bit quotient and 16-bit remainder. The quotient is placed in the lower 8 or 16 bits of Rd. The remainder is placed in the upper 8 or 16 bits of Rd. The sign of the remainder matches the sign of the dividend. A 4-bit immediate data in the source operand is extended to an 8- or 16-bit unsigned data. Valid results are not assured if division by zero is attempted or an overflow occurs.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↕	↕	—	—

H: Previous value remains unchanged.

N: Set to 1 if the quotient is negative; otherwise cleared to 0.

Z: Set to 1 if the divisor is zero; otherwise cleared to 0.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage		General Register
Source	Byte	R0L to R7L, R0H to R7H
	Word	R0 to R7, E0 to E7
Destination	Byte	R0 to R7, E0 to E7
	Word	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Source	Register direct
	4-bit immediate
Destination	Register direct

2.2.47 DIVXU**DIVide eXtend as Unsigned****[1] Assembly-Language Format**

DIVXU.B/W Rs, Rd

DIVXU.B/W #xx:4, Rd

[2] Operation $Rd \div Rs \rightarrow Rd$ $Rd \div \#IMM \rightarrow Rd$ **[3] Operand Size**

Byte, word

[4] Description

This instruction divides the lower 16 or 32 bits of the contents of the destination register (Rd) by the source operand and stores the 16- or 32-bit result in the destination register. The division is unsigned. The operation performed on 8-bit operands is 16 bits \div 8 bits \rightarrow 8-bit quotient and 8-bit remainder, and the operation performed on 16-bit operands is 32 bits \div 16 bits \rightarrow 16-bit quotient and 16-bit remainder. The quotient is placed in the lower 8 or 16 bits of Rd. The remainder is placed in the upper 8 or 16 bits of Rd. A 4-bit immediate data in the source operand is extended to an 8- or 16-bit unsigned data. Valid results are not assured if division by zero is attempted or an overflow occurs.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↕	↕	—	—

H: Previous value remains unchanged.

N: Set to 1 if the divisor is negative; otherwise cleared to 0.

Z: Set to 1 if the divisor is zero; otherwise cleared to 0.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage		General Register
Source	Byte	R0L to R7L, R0H to R7H
	Word	R0 to R7, E0 to E7
Destination	Byte	R0 to R7, E0 to E7
	Word	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Source	Register direct
	4-bit immediate
Destination	Register direct

2.2.48 EEPMOV (B)**MOVE data to EEPROM****[1] Assembly-Language Format**

EEPMOV.B

[2] Operationif R4L \neq 0 thenRepeat @ER5+ \rightarrow @ER6+R4L - 1 \rightarrow R4L

Until R4L = 0

else

next

[3] Operand Size

—

[4] Description

This instruction performs a block data transfer. It moves data from a memory location specified in ER5 to a memory location specified in ER6, increments ER5 and ER6, decrements R4L, and repeats these operations until R4L reaches zero. Execution then proceeds to the next instruction. Byte data is transferred for each transfer. R4L contains the number of bytes to be transferred. The byte symbol (B) in the assembly-language format above designates an 8-bit register R4L used in this instruction to indicate the amount of transfer data (allowing a maximum 255 bytes to be transferred). No interrupts are detected while the block transfer is in progress. When the EEPMOV.B instruction ends, R4L contains 0 (zero), and ER5 and ER6 contain the last transfer address + 1.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

2.2.49 EEPMOV (W)**MOVE data to EEPROM****[1] Assembly-Language Format**

EEPMOV.W

[2] Operation

```

if R4 ≠ 0 then
    Repeat @ER5+ → @ER6+
           R4 - 1 → R4
    Until R4 = 0
else
    next

```

[3] Operand Size

—

[4] Description

This instruction performs a block data transfer. It moves data from a memory location specified in ER5 to a memory location specified in ER6, increments ER5 and ER6, decrements R4, and repeats these operations until R4 reaches zero. Execution then proceeds to the next instruction. Byte data is transferred for each transfer. R4 contains the number of bytes to be transferred. The word symbol (W) in the assembly-language format above designates a 16-bit register R4 used in this instruction to indicate the amount of transfer data (allowing a maximum 65535 bytes to be transferred). All interrupts are detected while the block transfer is in progress.

If no interrupt occurs while the EEPMOV.W instruction is executing, R4 contains 0 (zero), and ER5 and ER6 contain the last transfer address + 1 when the EEPMOV.W instruction ends. If an interrupt occurs, interrupt exception handling begins after the current byte has been transferred. At this time, R4 indicates the number of bytes remaining to be transferred. ER5 and ER6 indicate the next transfer addresses. The PC value pushed onto the stack in interrupt exception handling is the start address of the instruction immediately after the EEPMOV.W instruction.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] EEPMOV.W Instruction and Interrupts

If an interrupt request occurs while the EEPMOV.W instruction is being executed, interrupt exception handling is carried out after the current byte has been transferred. Register contents are then as follows:

ER5: Start address of remaining source data to be transferred

ER6: Start address of remaining destination data to be transferred

R4: Number of bytes remaining to be transferred

The PC value pushed on the stack in interrupt exception handling is the start address of the instruction immediately after the EEPMOV.W instruction. Programs should be coded as follows to allow generation of interrupts during execution of the EEPMOV.W instruction.

Example:

```
L1: EEPMOV.W
      MOV.W   R4,R4
      BNE     L1
```

Interrupt requests other than NMI are not accepted if they are masked in the CPU.

During execution of the EEPMOV.B instruction no interrupts are accepted, including NMI.

2.2.50 EXTS

EXTend as Signed

[1] Assembly-Language Format

EXTS.W/L <EAd>

EXTS.L #2, <EAd>

[2] Operation

Sign extension <EAd> → <EAd>

[3] Operand Size

Word, longword

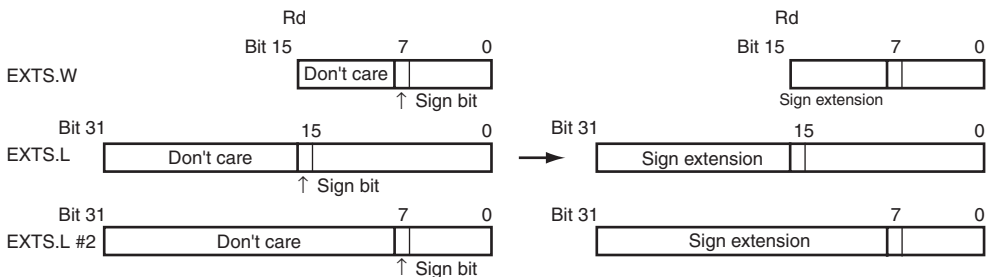
[4] Description

This instruction sign-extends the contents of the destination location <EAd> (destination operand).

EXTS.W <EAd> copies the sign bit (bit 7) of the lower 8 bits in a 16-bit destination operand in the upward direction to extend the data to word data.

EXTS.L <EAd> copies the sign bit (bit 15) of the lower 16 bits in a 32-bit destination operand in the upward direction to extend the data to longword data.

EXTS.L #2, <EAd> copies the sign bit (bit 7) of the lower 8 bits in a 32-bit destination operand in the upward direction to extend the data to longword data.



[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↕	↕	0	—

H: Previous value remains unchanged.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Always cleared to 0.

C: Previous value remains unchanged.

[6] Available General Registers

Usage		General Register	
Destination	Address register	ER0 to ER7	
	Index register	R0L to R7L, R0 to R7, ER0 to ER7	
	Data register	Byte	R0L to R7L, R0H to R7H
		Word	R0 to R7, E0 to E7
	Longword	ER0 to ER7	

[7] Available Addressing Modes

Source	Destination																	
	—	Rd	@ERd	@(d:2,ERd)	@(d:16,ERd)	@(d:32,ERd)	@ERd+	@ERd-	@+ERd	@-ERd	@(d:16,RdL.B)	@(d:16,Rd.W)	@(d:16,ERdL)	@(d:32,RdL.B)	@(d:32,Rd.W)	@(d:32,ERdL)	@aa:8	@aa:16
	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL		WL	WL

2.2.51 EXTU

EXTend as Unsigned

[1] Assembly-Language Format

EXTU.W/L <EAd>

EXTU.L #2, <EAd>

[2] Operation

Zero extension <EAd> → <EAd>

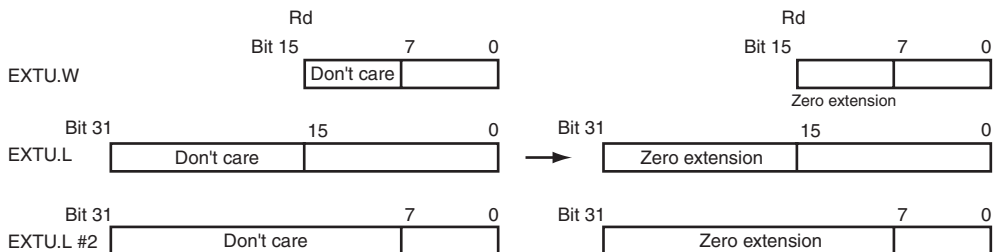
[3] Operand Size

Word, longword

[4] Description

This instruction zero-extends the contents of the destination location <EAd> (destination operand).

EXTU.W <EAd> fills the upper 8 bits of a 16-bit destination operand with 0s. EXTU.L <EAd> fills the upper 16 bits of a 32-bit destination operand with 0s. EXTU.L #2, <EAd> fills the upper 24 bits of a 32-bit destination operand with 0s.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	0	↕	0	—

H: Previous value remains unchanged.

N: Always cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Always cleared to 0.

C: Previous value remains unchanged.

[6] Available General Registers

Usage		General Register	
Destination	Address register	ER0 to ER7	
	Index register	R0L to R7L, R0 to R7, ER0 to ER7	
	Data register	Byte	R0L to R7L, R0H to R7H
		Word	R0 to R7, E0 to E7
		Longword	ER0 to ER7

Available Addressing Modes

Source	Destination																	
	Rd	@ERd	@(d:2,ERd)	@(d:16,ERd)	@(d:32,ERd)	@ERd+	@ERd-	@+ERd	@-ERd	@(d:16,RdL,B)	@(d:16,Rd,W)	@(d:16,ERd,L)	@(d:32,RdL,B)	@(d:32,Rd,W)	@(d:32,ERd,L)	@aa:8	@aa:16	@aa:32
—	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL		WL	WL

2.2.52 INC (B)**INCRement****[1] Assembly-Language Format**

INC.B Rd

[2] Operation

Rd + 1 → Rd

[3] Operand Size

Byte

[4] Description

This instruction increments an 8-bit register (Rd) (destination operand) and stores the result in the 8-bit register (Rd).

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↕	↕	↕	—

H: Previous value remains unchanged.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Set to 1 if an overflow occurs; otherwise cleared to 0.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Destination	R0L to R7L, R0H to R7H

[7] Available Addressing Mode

Usage	Addressing Mode
Destination	Register direct

[8] Note

An overflow is caused by the operation $H'7F + 1 \rightarrow H'80$.

2.2.53 INC (W)**INCrement****[1] Assembly-Language Format**

INC.W #1, Rd
 INC.W #2, Rd

[2] Operation

Rd + 1 → Rd
 Rd + 2 → Rd

[3] Operand Size

Word

[4] Description

This instruction adds the immediate value 1 or 2 to the contents of a 16-bit register (Rd) (destination operand) and stores the result in the 16-bit register (Rd).

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↕	↕	↕	—

H: Previous value remains unchanged.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Set to 1 if an overflow occurs; otherwise cleared to 0.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Destination	R0 to R7, E0 to E7

[7] Available Addressing Mode

Usage	Addressing Mode
Destination	Register direct

[8] Note

An overflow is caused by the operation $H'7FFF + 1 \rightarrow H'8000$, $H'7FFF + 2 \rightarrow H'8001$, or $H'7FFE + 2 \rightarrow H'8000$.

2.2.54 INC (L)**INCRement****[1] Assembly-Language Format**

INC.L #1, ERd

INC.L #2, ERd

[2] Operation

ERd + 1 → ERd

ERd + 2 → ERd

[3] Operand Size

Longword

[4] Description

This instruction adds the immediate value 1 or 2 to the contents of a 32-bit register (ERd) (destination operand) and stores the result in the 32-bit register (ERd).

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↕	↕	↕	—

H: Previous value remains unchanged.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Set to 1 if an overflow occurs; otherwise cleared to 0.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Destination	ER0 to ER7

[7] Available Addressing Mode

Usage	Addressing Mode
Destination	Register direct

[8] Note

An overflow is caused by the operation $H'7FFFFFFF + 1 \rightarrow H'80000000$, $H'7FFFFFFF + 2 \rightarrow H'80000001$, or $H'7FFFFFFE + 2 \rightarrow H'80000000$.

2.2.55

JMP**JuMP****[1] Assembly-Language Format**

JMP <EA>

Examples:

JMP @label:24

JMP @@label:7

[2] Operation

Effective address → PC

[3] Operand Size

—

[4] Description

This instruction branches unconditionally to a specified effective address.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Branch address	Register indirect
	Absolute address (24 bits)
	Absolute address (32 bits)
	Memory indirect
	Extended memory indirect

2.2.56 JSR**Jump to SubRoutine****[1] Assembly-Language Format**

JSR <EA>

Examples:

JSR @label:32

JSR @@label:8

[2] Operation

PC → @-SP

Effective address → PC

[3] Operand Size

—

[4] Description

This instruction pushes the PC value onto the stack as a restart address, then branches to a specified effective address. The PC value pushed onto the stack is the start address of the instruction immediately after this instruction.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Branch address	Register indirect
	Absolute address (24 bits)
	Absolute address (32 bits)
	Memory indirect
	Extended memory indirect

2.2.57 LDC (B)**LoaD to Control register****[1] Assembly-Language Format**

LDC.B <EAs>, CCR

[2] Operation

<EAs> → CCR

[3] Operand Size

Byte

[4] Description

This instruction loads the source operand contents into the condition-code register (CCR).

No interrupt requests, including NMI, are accepted immediately after execution of this instruction.

[5] Condition Code

I	UI	H	U	N	Z	V	C
↓	↓	↓	↓	↓	↓	↓	↓

I: Loaded from the corresponding bit in the source operand.

H: Loaded from the corresponding bit in the source operand.

N: Loaded from the corresponding bit in the source operand.

Z: Loaded from the corresponding bit in the source operand.

V: Loaded from the corresponding bit in the source operand.

C: Loaded from the corresponding bit in the source operand.

[6] Available General Registers

Usage	General Register
Source	R0L to R7L, R0H to R7H

[7] Available Addressing Modes

Usage	Addressing Mode
Source	Register direct
	8-bit immediate

2.2.58 LDC (B)**LoaD to Control register****[1] Assembly-Language Format**

LDC.B <EAs>, EXR

[2] Operation

<EAs> → EXR

[3] Operand Size

Byte

[4] Description

This instruction loads the source operand contents into the extended control register (EXR).

No interrupt requests, including NMI, are accepted for three states after execution of this instruction.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Source	R0L to R7L, R0H to R7H

[7] Available Addressing Modes

Usage	Addressing Mode
Source	Register direct
	8-bit immediate

2.2.59 LDC (W)**LoaD to Control register****[1] Assembly-Language Format**

LDC.W <EAs>, CCR

[2] Operation

(EAs) → CCR

[3] Operand Size

Word

[4] Description

This instruction loads the source operand contents into the condition-code register (CCR). Although CCR is a byte register, the load operation is performed in words. The contents of the start address are loaded into CCR.

No interrupt requests, including NMI, are accepted immediately after execution of this instruction.

[5] Condition Code

I	UI	H	U	N	Z	V	C
↕	↕	↕	↕	↕	↕	↕	↕

I: Loaded from the corresponding bit in the source operand.

H: Loaded from the corresponding bit in the source operand.

N: Loaded from the corresponding bit in the source operand.

Z: Loaded from the corresponding bit in the source operand.

V: Loaded from the corresponding bit in the source operand.

C: Loaded from the corresponding bit in the source operand.

[6] Available General Registers

Usage	General Register
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Source	Register indirect
	Register indirect with displacement
	Register indirect with post-increment
	Absolute address (16 bits)
	Absolute address (32 bits)

2.2.60 LDC (W)**LoaD to Control register****[1] Assembly-Language Format**

LDC.W <EAs>, EXR

[2] Operation

(EAs) → EXR

[3] Operand Size

Word

[4] Description

This instruction loads the source operand contents into the extended control register (EXR). Although EXR is a byte register, the load operation is performed in word units. The contents of the start address are loaded into EXR.

No interrupt requests, including NMI, are accepted immediately after execution of this instruction.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Source	Register indirect
	Register indirect with displacement
	Register indirect with post-increment
	Absolute address (16 bits)
	Absolute address (32 bits)

2.2.61 LDC (L)**Load to Control register****[1] Assembly-Language Format**

LDC.L ERs, VBR

LDC.L ERs, SBR

[2] Operation

ERs → VBR

ERs → SBR

[3] Operand Size

Longword

[4] Description

This instruction transfers the contents of a 32-bit register (ERs) into a base register (VBR or SBR).

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Destination	ER0 to ER7

[7] Available Addressing Mode

Usage	Addressing Mode
Destination	Register direct

2.2.62 LDM**Load to Multiple registers****[1] Assembly-Language Format**

LDM.L @SP+, <register list>

[2] Operation

@SP+ → ERn (register list)

[3] Operand Size

Longword

[4] Description

This instruction restores data saved on the stack to the multiple registers specified in a register list. Registers are restored in descending order of the register numbers. Two, three, or four registers can be restored by one LDM instruction. The register numbers should be serial.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Register list	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Register list	Register direct ($ERn - ERn + 1$)
	Register direct ($ERn - ERn + 2$)
	Register direct ($ERn - ERn + 3$)

[8] Notes

1. If ER7 is selected and saved on the stack, the saved data is read from the stack, however, it is not restored to ER7.
2. The register numbers of ER0 and ER7 are not considered as serial numbers.

2.2.63 LDMAC**Load to MAC register****[1] Assembly-Language Format**

LDMAC ERs, MAC register

[2] Operation

ERs → MACH

or

ERs → MACL

[3] Operand Size

Longword

[4] Description

This instruction transfers the contents of a general register to a MAC register (MACH or MACL).

If the transfer is to MACH, only the lowest 10 bits of the general register are transferred.

This instruction is available only when the multiplier is supported.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Source	ER0 to ER7

[7] Available Addressing Mode

Usage	Addressing Mode
Source	Register direct

[8] Note

Execution of this instruction clears the overflow flag in the multiplier to 0.

2.2.64 MAC**Multiply and ACcumulate****[1] Assembly-Language Format**

MAC @ERn+, @ERm+

[2] Operation

$(EAn) \times (EAm) + \text{MAC register} \rightarrow \text{MAC register}$

$ERn + 2 \rightarrow ERn$

$ERm + 2 \rightarrow ERm$

[3] Operand Size

—

[4] Description

This instruction performs signed multiplication on two 16-bit operands at addresses given by the contents of general registers (ERn and ERm), adds the 32-bit result to the contents of the MAC register, and stores the sum in the MAC register. After this operation, ERn and ERm are both incremented by 2. The operation can be carried out in saturating or non-saturating mode, depending on the MACS bit in the system control register (SYSCR).

In non-saturating mode, MACH and MACL are concatenated to store a 42-bit result. The value of bit 41 is copied into the upper 22 bits of MACH as a sign extension.

In saturating mode, only MACL is valid, and the result is limited to the range from H'80000000 (minimum value) to H'7FFFFFFF (maximum value). If the result overflows in the negative direction, H'80000000 (minimum value) is stored in MACL. If the result overflows in the positive direction, H'7FFFFFFF (maximum value) is stored in MACL. The LSB of MACH indicates the status of the overflow flag (V-MULT) in the multiplier. Other bits retain their previous contents. For details, see the hardware manual for the corresponding product.

This instruction is available only when the multiplier is supported.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Address register	ER0 to ER7

[7] Available Addressing Mode

Usage	Addressing Mode
General register	Register indirect with post-increment

[8] Notes

1. Flags (N, Z, and V) indicating the result of the MAC instruction can be set in the condition-code register (CCR) by the STMAC instruction.
2. If ER_n and ER_m are the same register, the execution addresses are ER_n and ER_n + 2. After execution, the value of ER_n is ER_n + 4.
3. If the MACS bit is modified during execution of a MAC instruction, the result cannot be guaranteed.

[9] Modification of Flags

The multiplier has N-MULT, Z-MULT, and V-MULT flags that indicate the results of MAC instructions. These flags are separated from the condition-code register (CCR). The values of these flags can be set in the N, Z, and V flags of CCR only by the STMAC instruction.

N-MULT and Z-MULT are modified only by MAC instructions. V-MULT retains a value indicating whether an overflow has occurred in the past, until it is cleared by execution of the CLRMAC or LDMAC instruction.

The setting and clearing conditions for these flags are given below.

- **N-MULT (Negative Flag)**

Saturating mode	Setting condition	Set when bit 31 of MACL is set to 1 by execution of a MAC instruction
	Clearing condition	Cleared when bit 31 of MACL is cleared to 0 by execution of a MAC instruction
Non-saturating mode	Setting condition	Set when bit 41 of MACH is set to 1 by execution of a MAC instruction
	Clearing condition	Cleared when bit 41 of MACH is cleared to 0 by execution of a MAC instruction

- **Z-MULT (Zero Flag)**

Saturating mode	Setting condition	Set when MACL is cleared to 0 by execution of a MAC instruction
	Clearing condition	Cleared when MACL is not cleared to 0 by execution of a MAC instruction
Non-saturating mode	Setting condition	Set when MACH and MACL are both cleared to 0 by execution of a MAC instruction
	Clearing condition	Cleared when MACH or MACL is not cleared to 0 by execution of a MAC instruction

- **V-MULT (Overflow Flag)**

Saturating mode	Setting condition	Set when the result of a MAC instruction overflows the range from H'80000000 (minimum) to H'7FFFFFFF (maximum)
	Clearing condition	Cleared when a CLRMAC or LDMAC instruction is executed*
Non-saturating mode	Setting condition	Set when the result of a MAC instruction overflows the range from H'2000000000 (minimum) to H'1FFFFFFFFF (maximum)
	Clearing condition	Cleared when a CLRMAC or LDMAC instruction is executed*

Note: * Not cleared even when the result of a MAC instruction is within the above range.

The N-MULT, Z-MULT, and V-MULT flags are not modified by switching between saturating and non-saturating modes, or by execution of a multiply instruction (MULXU or MULXS).

- **Example**

CLRMAC

MAC @ER1+,@ER2+

MAC @ER1+,@ER2+ ←—— Overflow occurs
:
MAC @ER1+,@ER2+ ←—— Result = 0
NOP
STMAC MACH,ER3 ←—— CCR (N = 0, Z = 1, V = 1)
CLRMAC
STMAC MACH,ER3 ←—— CCR (N = 0, Z = 1, V = 0)

2.2.65 MOV**MOVE data****[1] Assembly-Language Format**

MOV.Sz <EAs>, <EAd>

[2] Operation

<EAs> → <EAd>

[3] Operand Size

Byte, word, longword

[4] Description

This instruction transfers the source operand <EAs> into the destination location, tests the transferred data, and sets CCR according to the result.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↑	↑	0	—

H: Previous value remains unchanged.

N: Set to 1 if the transferred data is negative; otherwise cleared to 0.

Z: Set to 1 if the transferred data is zero; otherwise cleared to 0.

V: Always cleared to 0.

C: Previous value remains unchanged.

[6] Available General Registers

Usage		General Register
Address register		ER0 to ER7
Index register		R0L to R7L, R0 to R7, ER0 to ER7
Data register	Byte	R0L to R7L, R0H to R7H
	Word	R0 to R7, E0 to E7
	Longword	ER0 to ER7

[7] Available Addressing Modes

Source	Destination																		
	Rd	@ERd	@(d:2,ERd)	@(d:16,ERd)	@(d:32,ERd)	@ERd+	@ERd-	@+ERd	@-ERd	@(d:16,RdL,B)	@(d:16,Rd,W)	@(d:16,ERd,L)	@(d:32,RdL,B)	@(d:32,Rd,W)	@(d:32,ERd,L)	@aa:8	@aa:16	@aa:32	
#x:3(1-7)	WL																		
#x:4																		BW	BW
#x:8	B	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	
#x:16	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL		WL	WL	
#x:32	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L		L	L	
Rs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL	
@ERs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	
@(d:2,ERs)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	
@(d:16,ERs)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	
@(d:32,ERs)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	
@ERs+	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	
@ERs-	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	
@+ERs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	
@-ERs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	
@(d:16,RsL,B)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	
@(d:16,Rs,W)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	
@(d:16,ERs,L)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	
@(d:32,RsL,B)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	
@(d:32,Rs,W)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	
@(d:32,ERs,L)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	
@aa:8	B																		
@aa:16	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	
@aa:32	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	

2.2.66 MOVA**MOVE effective Address****[1] Assembly-Language Format**

MOVA/Sz @(d, <EAs>), ERd

[2] Operation

d + <EAs> → ERd

[3] Operand Size

Index: Byte, word

[4] Description

This instruction zero-extends the source operand (index), shifts the data by the specified bits to the left, and adds the shifted data with the displacement to obtain an effective address. The result is stored in a 32-bit register (ERd).

MOVA/B is used when accessing the obtained effective address in bytes and the data is not shifted to the left. MOVA/W is used when accessing the obtained effective address in words and the data is shifted by 1. MOVA/L is used when accessing the obtained effective address in longwords and the data is shifted by 2.

- Example

```
MOVA/L    @(H'FFFFFF00, R0L.B), ER1
MOV.L     @ER1, ER2
```

The source operand can be selected from byte data or word data independent of the size in MOVA/B, MOVA/W, or MOVA/L. An example is shown below.

- Example

```
MOVA/L    @(H'FFFFFF00, @(H'FFFFFF00, R0.W).B), ER1
```

Selects the size for an access to the effective address.	Selects the source operand.	Selects the size of the source operand.
--	--------------------------------	--

In this case, the operation is identical to the following operation.

```
MOV.B     @(H'FFFFFF00, R0.W), R1L
MOVA/L    @(H'FFFFFF00, R1L.B), ER1
```


[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage		General Register		
Source (index)	Address register	ER0 to ER7		
	Index register	R0L to R7L, R0 to R7, ER0 to ER7		
	Data register	Byte	R0L to R7L, R0H to R7H	
		Word	R0 to R7, E0 to E7	
Longword		ER0 to ER7		
Destination		ER0 to ER7		

[7] Available Addressing Modes

Usage	Addressing Mode
Source(index)	Register direct
	Register indirect
	Register indirect with displacement
	Index register indirect with displacement
	Register indirect with post-increment
	Register indirect with post-decrement
	Register indirect with pre-increment
	Register indirect with pre-decrement
Destination	Register direct

[8] Notes

When the addressing mode of the source operand (index) is register direct and the numbers of the source and destination general registers are the same, the short format can be used. For details, see section 2.4, List of Instruction Codes.

Example:

```
MOVA/W @(H'FFFF8000, R1L,B), ER1
```

2.2.67 MOVFPE**MOVE From Peripheral with E clock****[1] Assembly-Language Format**

MOVFPE @aa:16, Rd

[2] Operation

(EAs) → Rd

Synchronized with E clock

[3] Operand Size

Byte

[4] Description

This instruction transfers memory contents specified by a 16-bit absolute address to a general register Rd in synchronization with an E clock, tests the transferred data, and sets CCR according to the result.

Note: Avoid using this instruction in microcomputers without an E clock output pin, or in single-chip mode.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↑↓	↑↓	0	—

H: Previous value remains unchanged.

N: Set to 1 if the transferred data is negative; otherwise cleared to 0.

Z: Set to 1 if the transferred data is zero; otherwise cleared to 0.

V: Always cleared to 0.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Destination	R0L to R7L, R0H to R7H

[7] Available Addressing Modes

Usage	Addressing Mode
Source	Absolute address (16 bits)
Destination	Register direct

[8] Notes

1. This instruction cannot be used with addressing modes other than the above, and cannot transfer word data or longword data.
2. For details on data transfer using this instruction, see the hardware manual for the corresponding product.

2.2.68 MOVMD (B)**MOVE Multiple Data****[1] Assembly-Language Format**

MOVMD.B

[2] Operation

Repeat @ER5+ → @ER6+

R4 - 1 → R4

Until R4 = 0

[3] Operand Size

Transfer data: Byte

[4] Description

This instruction performs a block data transfer. Byte data is consecutively transferred for each data transfer. This instruction transfers the contents of a memory location pointed to by ER5 to a memory location pointed to by ER6, increments the contents of ER5 and ER6, decrements the R4 contents, and repeats these operations until R4 reaches zero. Execution then proceeds to the next instruction. R4 contains the number of transfers to be carried out. (When setting 0 in R4, it is considered as 65536.)

If an interrupt request occurs while the MOVMD instruction is being executed, interrupt exception handling is carried out after the current byte has been transferred. The PC value pushed onto the stack in interrupt exception handling is the address of this instruction. The remaining data continues to be transferred after returning from the interrupt handling routine.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

2.2.69 MOVMD (W)**MOVE Multiple Data****[1] Assembly-Language Format**

MOVMD.W

[2] Operation

Repeat @ER5+ → @ER6+

R4 - 1 → R4

Until R4 = 0

[3] Operand Size

Transfer data: Word

[4] Description

This instruction performs a block data transfer. Word data is consecutively transferred for each data transfer. This instruction transfers the contents of a memory location pointed to by ER5 to a memory location pointed to by ER6, increments the contents of ER5 and ER6 by 2, decrements the R4 contents, and repeats these operations until R4 reaches zero. Execution then proceeds to the next instruction. R4 contains the number of transfers to be carried out. (When setting 0 in R4, it is considered as 65536.)

If an interrupt request occurs while the MOVMD instruction is being executed, interrupt exception handling is carried out after the current word has been transferred. The PC value pushed on the stack in interrupt exception handling is the address of this instruction. The remaining data continues to be transferred after returning from the interrupt handling routine.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

2.2.70 MOVMD (L)**MOVE Multiple Data****[1] Assembly-Language Format**

MOVMD.L

[2] Operation

Repeat @ER5+ → @ER6+

R4 - 1 → R4

Until R4 = 0

[3] Operand Size

Transfer data: Longword

[4] Description

This instruction performs a block data transfer. Longword data is consecutively transferred for each data transfer. This instruction transfers the contents of a memory location pointed to by ER5 to a memory location pointed to by ER6, increments the contents of ER5 and ER6 by 4, decrements the R4 contents, and repeats these operations until R4 reaches zero. Execution then proceeds to the next instruction. R4 contains the number of transfers to be carried out. (When setting 0 in R4, it is considered as 65536.)

If an interrupt request occurs while the MOVMD instruction is being executed, interrupt exception handling is carried out after the current longword has been transferred. The PC value pushed on the stack in interrupt exception handling is the address of this instruction. The remaining data continues to be transferred after returning from the interrupt handling routine.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

2.2.71 MOVSD**MOVe String Data****[1] Assembly-Language Format**

MOVSD disp

[2] Operation

Repeat @ER5+ → @ER6

if @ER6 == 0 then {

ER6 + 1 → ER6

R4 - 1 → R4

goto next + disp

}

ER6 + 1 → ER6

R4 - 1 → R4

Until R4 = 0

[3] Operand Size

Transfer data: Byte

[4] Description

This instruction performs a block data transfer. Byte data is consecutively transferred for each data transfer. This instruction transfers the contents of a memory location pointed to by ER5 to a memory location pointed to by ER6, increments the contents of ER5 and ER6, and decrements the R4 contents. Execution then proceeds to the next instruction. Transferring zero data (H'00) stops the transfer. A displacement is added to the program counter (PC) and execution branches to the resulting address. The PC value used in the address calculation is the start address of the instruction immediately after this instruction. The displacement is a signed 16-bit value. The maximum number of transfers is specified by R4 (when setting 0 in R4, it is considered as 65536). The above operations are repeated until R4 reaches zero.

If an interrupt request occurs while the MOVSD instruction is being executed, interrupt exception handling is carried out after the current byte has been transferred. The PC value pushed on the stack in interrupt exception handling is the address of this instruction. The remaining data continues to be transferred after returning from the interrupt handling routine.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

2.2.72 MOVTPPE**MOVE To Peripheral with E clock****[1] Assembly-Language Format**

MOVTPPE Rs, @aa:16

[2] Operation

Rs → (EAd)

Synchronized with E clock

[3] Operand Size

Byte

[4] Description

This instruction stores the contents of a general register Rs (source operand) to the location of an operand specified by a 16-bit absolute address in synchronization with an E clock, tests the transferred data, and sets CCR according to the result.

Note: Avoid using this instruction in microcomputers without an E clock output pin, or in single-chip mode.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↕	↕	0	—

H: Previous value remains unchanged.

N: Set to 1 if the transferred data is negative; otherwise cleared to 0.

Z: Set to 1 if the transferred data is zero; otherwise cleared to 0.

V: Always cleared to 0.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Source	R0L to R7L, R0H to R7H

[7] Available Addressing Modes

Usage	Addressing Mode
Source	Register direct
Destination	Absolute address (16 bits)

[8] Notes

1. This instruction cannot be used with addressing modes other than the above, and cannot load word data or longword data.
2. For details on data transfer using this instruction, see the hardware manual for the corresponding product.

2.2.73 MULS**MULTIPLY as Signed****[1] Assembly-Language Format**

MULS.W/L Rs, Rd

MULS.W/L #xx:4, Rd

[2] Operation $Rd \times Rs \rightarrow Rd$ $Rd \times \#IMM \rightarrow Rd$ **[3] Operand Size**

Word, longword

[4] Description

This instruction multiplies the contents of a 16- or 32-bit register (Rd) by the contents of a 16- or 32-bit register (Rs) or 4-bit immediate value and stores the result (only the lower 16 or 32 bits) in the 16- or 32-bit register (Rd). (The upper 16 or 32 bits of the result are ignored. The Z or N flag also changes according to the contents of the lower 16 or 32 bits regardless of the contents of the upper 16 or 32 bits.) The multiplication is signed.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↑	↓	—	—

H: Previous value remains unchanged.

N: Set to 1 if the result (the data stored in Rd) is negative; otherwise cleared to 0.

Z: Set to 1 if the result (the data stored in Rd) is zero; otherwise cleared to 0.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register	
Source	Word	R0 to R7, E0 to E7
	Longword	ER0 to ER7
Destination	Word	R0 to R7, E0 to E7
	Longword	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Source	Register direct
	4-bit immediate
Destination	Register direct

2.2.74 MULS/U**MULTIPLY as Signed****[1] Assembly-Language Format**

MULS/U.L Rs, Rd

MULS/U.L #xx:4, Rd

[2] Operation $Rd \times Rs \rightarrow Rd$ (upper 32 bits) $Rd \times \#IMM \rightarrow Rd$ (upper 32 bits)**[3] Operand Size**

Longword

[4] Description

This instruction multiplies the contents of a 32-bit register (Rd) by the contents of a 32-bit register (Rs) or 4-bit immediate value and stores the upper 32 bits of the result in the 32-bit register (Rd). (The lower 32 bits of the result are ignored. The Z or N flag also changes according to the contents of the upper 32 bits regardless of the contents of the lower 32 bits.) The multiplication is signed.

This instruction is available only when the multiplier is supported.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↑	↑	—	—

H: Previous value remains unchanged.

N: Set to 1 if the result (the data stored in Rd) is negative; otherwise cleared to 0.

Z: Set to 1 if the result (the data stored in Rd) is zero; otherwise cleared to 0.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Source	ER0 to ER7
Destination	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Source	Register direct
	4-bit immediate
Destination	Register direct

2.2.75 MULU**MULTIPLY as Unsigned****[1] Assembly-Language Format**

MULU.W/L Rs, Rd

MULU.W/L #xx:4, Rd

[2] Operation $Rd \times Rs \rightarrow Rd$ $Rd \times \#IMM \rightarrow Rd$ **[3] Operand Size**

Word, longword

[4] Description

This instruction multiplies the contents of a 16- or 32-bit register (Rd) by the contents of a 16- or 32-bit register (Rs) or 4-bit immediate value and stores the result (only the lower 16 or 32 bits) in the 16- or 32-bit register Rd. The multiplication is unsigned.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register	
Source	Word	R0 to R7, E0 to E7
	Longword	ER0 to ER7
Destination	Word	R0 to R7, E0 to E7
	Longword	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Source	Register direct
	4-bit immediate
Destination	Register direct

2.2.76 MULU/U**MULTIPLY as Unsigned****[1] Assembly-Language Format**

MULU/U.L Rs, Rd

MULU/U.L #xx:4, Rd

[2] Operation $Rd \times Rs \rightarrow Rd$ (upper 32 bits) $Rd \times \#IMM \rightarrow Rd$ (upper 32 bits)**[3] Operand Size**

Longword

[4] Description

This instruction multiplies the contents of a 32-bit register (Rd) by the contents of a 32-bit register (Rs) or 4-bit immediate value and stores the upper 32 bits of the result in the 32-bit register (Rd). The multiplication unsigned.

This instruction is available only when the multiplier is supported.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Source	ER0 to ER7
Destination	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Source	Register direct
	4-bit immediate
Destination	Register direct

2.2.77 MULXS

MULTiply eXtend as Signed

[1] Assembly-Language Format

MULXS.B/W Rs, Rd
 MULXS.B/W #xx:4, Rd

[2] Operation

$Rd \times Rs \rightarrow Rd$
 $Rd \times \#IMM \rightarrow Rd$

[3] Operand Size

Byte, word

[4] Description

This instruction multiplies the source operand by the lower 8 or 16 bits of the contents of the destination register (Rd) and stores the 16- or 32-bit result in the destination register. The multiplication is signed. The operation performed on 8-bit operands is 8 bits \times 8 bits \rightarrow 16-bit product, and the operation performed on 16-bit operands is 16 bits \times 16 bits \rightarrow 32-bit product.

A 4-bit immediate data in the source operand is extended to an 8- or 16-bit unsigned data.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↑	↑	—	—

H: Previous value remains unchanged.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage		General Register
Source	Byte	R0L to R7L, R0H to R7H
	Word	R0 to R7, E0 to E7
Destination	Byte	R0 to R7, E0 to E7
	Word	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Source	Register direct
	4-bit immediate
Destination	Register direct

2.2.78 MULXU**MULTiply eXtend as Unsigned****[1] Assembly-Language Format**

MULXU.B/W Rs, Rd

MULXU.B/W #xx:4, Rd

[2] Operation $Rd \times Rs \rightarrow Rd$ $Rd \times \#IMM \rightarrow Rd$ **[3] Operand Size**

Byte, word

[4] Description

This instruction multiplies the source operand by the lower 8 or 16 bits of the contents of the destination register (Rd) and stores the 16- or 32-bit result in the destination register. The multiplication is unsigned. The operation performed on 8-bit operands is 8 bits \times 8 bits \rightarrow 16-bit product and the operation performed on 16-bit operands is 16 bits \times 16 bits \rightarrow 32-bit product.

The 4-bit immediate data in the source operand is extended to an 8- or 16-bit unsigned data.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage		General Register
Source	Byte	R0L to R7L, R0H to R7H
	Word	R0 to R7, E0 to E7
Destination	Byte	R0 to R7, E0 to E7
	Word	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Source	Register direct
	4-bit immediate
Destination	Register direct

2.2.79 NEG**NEGate****[1] Assembly-Language Format**

NEG.Sz <EAd>

[2] Operation

0 – <EAd> → <EAd>

[3] Operand Size

Byte, word, longword

[4] Description

This instruction takes the two's complement of the contents of the destination location <EAd> (destination operand) and stores the result in the destination location. If the original contents of the destination operand are H'80, the result remains H'80 and an overflow occurs. Similarly, if the original contents are H'8000 or H'80000000, the result remains H'8000 or H'80000000, respectively and an overflow occurs.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	↕	—	↕	↕	↕	↕

H: Set to 1 if there is a borrow at the 4th bit from the MSB (bit 27, bit 11, or bit 3); otherwise cleared to 0.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Set to 1 if an overflow occurs; otherwise cleared to 0.

C: Set to 1 if there is a borrow at the MSB (bit 31, bit 15, or bit 7); otherwise cleared to 0.

[6] Available General Registers

Usage	General Register		
Destination	Address register	ER0 to ER7	
	Index register	R0L to R7L, R0 to R7, ER0 to ER7	
	Data register	Byte	R0L to R7L, R0H to R7H
		Word	R0 to R7, E0 to E7
Longword		ER0 to ER7	

[7] Available Addressing Modes

Source	Destination																	
	—	Rd	@ERd	@(d:2,ERd)	@(d:16,ERd)	@(d:32,ERd)	@ERd+	@ERd-	@+ERd	@-ERd	@(d:16,RdL,B)	@(d:16,Rd,W)	@(d:16,ERd,L)	@(d:32,RdL,B)	@(d:32,Rd,W)	@(d:32,ERd,L)	@aa:8	@aa:16
	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL

2.2.80**NOP****No Operation**

[1] Assembly-Language Format

NOP

[2] Operation

PC + 2 → PC

[3] Operand Size

—

[4] Description

This instruction only increments the program counter, causing the next instruction to be executed. The internal state of the CPU does not change.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

2.2.81 NOT**NOT = logical complement****[1] Assembly-Language Format**

NOT.Sz <EAd>

[2] Operation $\sim \langle \text{EAd} \rangle \rightarrow \langle \text{EAd} \rangle$ **[3] Operand Size**

Byte, word, longword

[4] Description

This instruction takes the one's complement of the contents of the destination location <EAd> (destination operand) and stores the result in the destination location.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↕	↕	0	—

H: Previous value remains unchanged.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Always cleared to 0.

C: Previous value remains unchanged.

[6] Available General Registers

Usage		General Register
Destination	Address register	ER0 to ER7
	Index register	R0L to R7L, R0 to R7, ER0 to ER7
	Data register	Byte R0L to R7L, R0H to R7H
	Word	R0 to R7, E0 to E7
	Longword	ER0 to ER7

[7] Available Addressing Modes

Source	Destination																	
	Rd	@ ERd	@ (dt:2,ERd)	@ (dt:16,ERd)	@ (dt:32,ERd)	@ ERd+	@ ERd-	@ +ERd	@ -ERd	@ (dt:16,RdL.B)	@ (dt:16,Rd.W)	@ (dt:16,ERd.L)	@ (dt:32,RdL.B)	@ (dt:32,Rd.W)	@ (dt:32,ERd.L)	@ aa:8	@ aa:16	@ aa:32
—	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL

2.2.82

OR

inclusive OR logical

[1] Assembly-Language Format

OR.Sz <EAs>, <EAd>

[2] Operation

<EAd> ∨ <EAs> → <EAd>

[3] Operand Size

Byte, word, longword

[4] Description

This instruction ORs the contents of the destination location <EAd> (destination operand) with the source operand <EAs> and stores the result in the destination location.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↑	↑	0	—

H: Previous value remains unchanged.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Always cleared to 0.

C: Previous value remains unchanged.

[6] Available General Registers

Usage		General Register
Address register		ER0 to ER7
Index register		R0L to R7L, R0 to R7, ER0 to ER7
Data register	Byte	R0L to R7L, R0H to R7H
	Word	R0 to R7, E0 to E7
	Longword	ER0 to ER7

[7] Available Addressing Modes

Source	Destination																		
	Rd	@ERd	@(d:2,ERd)	@(d:16,ERd)	@(d:32,ERd)	@ERd+	@ERd-	@+ERd	@-ERd	@(d:16,RdL.B)	@(d:16,Rd.W)	@(d:16,ERd.L)	@(d:32,RdL.B)	@(d:32,Rd.W)	@(d:32,ERd.L)	@aa:8	@aa:16	@aa:32	
#x:3(1-7)																			
#x:8	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
#x:16	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL		WL	WL	
#x:32	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L		L	L	
Rs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL	BWL
@ERs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@(d:2,ERs)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@(d:16,ERs)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@(d:32,ERs)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@ERs+	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@ERs-	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@+ERs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@-ERs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@(d:16,RsL.B)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@(d:16,Rs.W)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@(d:16,ERs.L)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@(d:32,RsL.B)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@(d:32,Rs.W)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@(d:32,ERs.L)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@aa:8	B																		
@aa:16	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@aa:32	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL

2.2.83 ORC**inclusive OR Control register****[1] Assembly-Language Format**

ORC #xx:8, CCR

[2] OperationCCR \vee #IMM \rightarrow CCR**[3] Operand Size**

Byte

[4] Description

This instruction ORs the contents of the condition-code register (CCR) with immediate data and stores the result in CCR. No interrupt requests, including NMI, are accepted immediately after execution of this instruction.

[5] Condition Code

I	UI	H	U	N	Z	V	C
↕	↕	↕	↕	↕	↕	↕	↕

I: Stores the corresponding bit of the result.

UI: Stores the corresponding bit of the result.

H: Stores the corresponding bit of the result.

U: Stores the corresponding bit of the result.

N: Stores the corresponding bit of the result.

Z: Stores the corresponding bit of the result.

V: Stores the corresponding bit of the result.

C: Stores the corresponding bit of the result.

[6] Available Addressing Mode

Immediate

2.2.84 ORC**inclusive OR Control register**

[1] Assembly-Language Format

ORC #xx:8, EXR

[2] OperationEXR \vee #IMM \rightarrow EXR**[3] Operand Size**

Byte

[4] Description

This instruction ORs the contents of EXR with immediate data and stores the result in EXR. No interrupt requests, including NMI, are accepted immediately after execution of this instruction.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available Addressing Mode

Immediate

2.2.85 POP (W)**POP data****[1] Assembly-Language Format**

POP.W Rn

[2] Operation

@SP+ → Rn

[3] Operand Size

Word

[4] Description

This instruction restores data from the stack to a 16-bit general register Rn, tests the restored data, and sets CCR according to the result.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↕	↕	0	—

H: Previous value remains unchanged.

N: Set to 1 if the transferred data is negative; otherwise cleared to 0.

Z: Set to 1 if the transferred data is zero; otherwise cleared to 0.

V: Always cleared to 0.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Destination	R0 to R7, E0 to E7

[7] Available Addressing Mode

Usage	Addressing Mode
Destination	Register direct

[8] Note

POP.W Rn is identical to MOV.W @SP+, Rn.

2.2.86 POP (L)**POP data****[1] Assembly-Language Format**

POP.L ERn

[2] Operation

@SP+ → ERn

[3] Operand Size

Longword

[4] Description

This instruction restores data from the stack to a 32-bit general register ERn, tests the restored data, and sets CCR according to the result.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↑	↑	0	—

H: Previous value remains unchanged.

N: Set to 1 if the transferred data is negative; otherwise cleared to 0.

Z: Set to 1 if the transferred data is zero; otherwise cleared to 0.

V: Always cleared to 0.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Address register	ER0 to ER7

[7] Available Addressing Mode

Usage	Addressing Mode
Destination	Register direct

[8] Note

POP.L ERn is identical to MOV.L @SP+, ERn.

2.2.87**PUSH (W)****PUSH data****[1] Assembly-Language Format**

PUSH.W Rn

[2] Operation

Rn → @-SP

[3] Operand Size

Word

[4] Description

This instruction saves data from a 16-bit register Rn onto the stack, tests the saved data, and sets CCR according to the result.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↑	↑	0	—

H: Previous value remains unchanged.

N: Set to 1 if the transferred data is negative; otherwise cleared to 0.

Z: Set to 1 if the transferred data is zero; otherwise cleared to 0.

V: Always cleared to 0.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Destination	R0 to R7, E0 to E7

[7] Available Addressing Mode

Usage	Addressing Mode
Destination	Register direct

[8] Notes

1. PUSH.W Rn is identical to MOV.W Rn, @-SP.
2. When PUSH.W R7 or PUSH.W E7 is executed, the value saved onto the stack is the R7 or E7 value after effective address calculation (ER7 is decremented by 2).

2.2.88 PUSH (L)**PUSH data****[1] Assembly-Language Format**PUSH.L ER_n**[2] Operation**ER_n → @-SP**[3] Operand Size**

Longword

[4] Description

This instruction pushes data from a 32-bit register (ER_n) onto the stack, tests the saved data, and sets CCR according to the result.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↑	↑	0	—

H: Previous value remains unchanged.

N: Set to 1 if the transferred data is negative; otherwise cleared to 0.

Z: Set to 1 if the transferred data is zero; otherwise cleared to 0.

V: Always cleared to 0.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Address register	ER0 to ER7

[7] Available Addressing Mode

Usage	Addressing Mode
Destination	Register direct

[8] Notes

1. PUSH.L ERn is identical to MOV.L ERn, @-SP.
2. When PUSH.L ER7 is executed, the value saved onto the stack is the ER7 value after effective address calculation (ER7 is decremented by 4).

2.2.89

ROTL

ROTate Left

[1] Assembly-Language Format

ROTL.Sz <EAd>

ROTL.Sz #2, <EAd>

[2] Operation

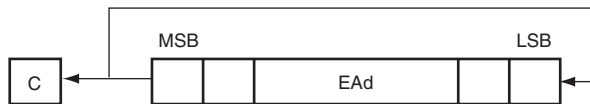
<EAd> (left rotation) → <EAd>

[3] Operand Size

Byte, word, longword

[4] Description

This instruction rotates the bits in the contents of the destination location <EAd> (destination operand) by one or two bits to the left. The result is stored in the destination location. The shifted out bits are returned to the lower bits and the last bit shifted out is also copied to the carry flag.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	↑	↑	0	↑

H: Previous value remains unchanged.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Always cleared to 0.

C: Stores the last bit shifted out.

[6] Available General Registers

Usage		General Register	
Destination	Address register	ER0 to ER7	
	Index register	R0L to R7L, R0 to R7, ER0 to ER7	
	Data register	Byte	R0L to R7L, R0H to R7H
		Word	R0 to R7, E0 to E7
Longword		ER0 to ER7	

[7] Available Addressing Modes

Number of rotated bits: Immediate

Source	Destination																	
	—	Rd	@ERd	@(d:2,ERd)	@(d:16,ERd)	@(d:32,ERd)	@ERd+	@ERd-	@+ERd	@-ERd	@(d:16,RdL,B)	@(d:16,Rd,W)	@(d:16,ERd,L)	@(d:32,RdL,B)	@(d:32,Rd,W)	@(d:32,ERd,L)	@aa:8	@aa:16
	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL

[8] Note

For details on the accessible ranges for @aa:8 or @aa:16, see the hardware manual for the corresponding product.

2.2.90 ROTR**ROTate Right****[1] Assembly-Language Format**

ROTR.Sz <EAd>

ROTR.Sz #2, <EAd>

[2] Operation

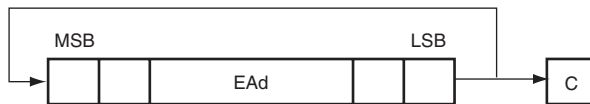
<EAd> (right rotation) → <EAd>

[3] Operand Size

Byte, word, longword

[4] Description

This instruction rotates the bits in the contents of the destination location <EAd> (destination operand) by one or two bits to the right. The result is stored in the destination location. The shifted out bits are returned to the upper bits and the last bit shifted out is also copied to the carry flag.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	↕	↕	0	↕

H: Previous value remains unchanged.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Always cleared to 0.

C: Stores the last bit shifted out.

[6] Available General Registers

Usage		General Register	
Destination	Address register	ER0 to ER7	
	Index register	R0L to R7L, R0 to R7, ER0 to ER7	
	Data register	Byte	R0L to R7L, R0H to R7H
		Word	R0 to R7, E0 to E7
Longword		ER0 to ER7	

[7] Available Addressing Modes

Number of rotated bits: Immediate

Source	Destination																	
	—	Rd	@ERd	@(d:2,ERd)	@(d:16,ERd)	@(d:32,ERd)	@ERd+	@ERd-	@+ERd	@-ERd	@(d:16,RdL,B)	@(d:16,Rd,W)	@(d:16,ERd,L)	@(d:32,RdL,B)	@(d:32,Rd,W)	@(d:32,ERd,L)	@aa:8	@aa:16
	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL

2.2.91 ROTXL

ROTate with eXtend carry Left

[1] Assembly-Language Format

ROTXL.Sz <EAd>
 ROTXL.Sz #2, <EAd>

[2] Operation

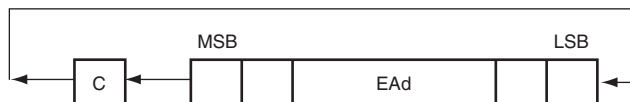
<EAd> (left rotation with carry) → <EAd>

[3] Operand Size

Byte, word, longword

[4] Description

This instruction rotates the bits in the contents of the destination location <EAd> (destination operand) by one or two bits to the left. The shifted bits include the carry flag. The result is stored in the destination location. When 2-bit rotation with the carry flag is operated, the shifted out bits are returned to the lower bits and the last bit shifted out is also copied to the carry flag.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	↕	↕	0	↕

H: Previous value remains unchanged.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Always cleared to 0.

C: Stores the last bit shifted out.

[6] Available General Registers

Usage		General Register	
Destination	Address register	ER0 to ER7	
	Index register	R0L to R7L, R0 to R7, ER0 to ER7	
	Data register	Byte	R0L to R7L, R0H to R7H
		Word	R0 to R7, E0 to E7
Longword		ER0 to ER7	

[7] Available Addressing Modes

Number of rotated bits: Immediate

Source	Destination																	
	Rd	@ERd	@(d:2,ERd)	@(d:16,ERd)	@(d:32,ERd)	@ERd+	@ERd-	@+ERd	@-ERd	@(d:16,RdL,B)	@(d:16,Rd,W)	@(d:16,ERd,L)	@(d:32,RdL,B)	@(d:32,Rd,W)	@(d:32,ERd,L)	@aa:8	@aa:16	@aa:32
—	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL

2.2.92 ROTXR

ROTate with eXtend carry Right

[1] Assembly-Language Format

ROTXR.Sz <EAd>

ROTXR.Sz #2, <EAd>

[2] Operation

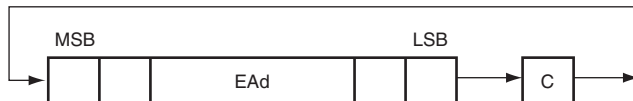
<EAd> (right rotation with carry) → <EAd>

[3] Operand Size

Byte, word, longword

[4] Description

This instruction rotates the bits in the contents of the destination location <EAd> (destination operand) by one or two bits to the right. The shifted bits include the carry flag. The result is stored in the destination location. When 2-bit rotation with the carry flag is operated, the shifted out bits are returned to the upper bits and the last bit shifted out is also copied to the carry flag.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	↕	↕	0	↕

H: Previous value remains unchanged.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Always cleared to 0.

C: Stores the last bit shifted out.

[6] Available General Registers

Usage	General Register		
Destination	Address register	ER0 to ER7	
	Index register	R0L to R7L, R0 to R7, ER0 to ER7	
	Data register	Byte	R0L to R7L, R0H to R7H
		Word	R0 to R7, E0 to E7
Longword		ER0 to ER7	

[7] Available Addressing Modes

Number of rotated bits: Immediate

Source	Destination																	
	Rd	@ERd	@(d:2,ERd)	@(d:16,ERd)	@(d:32,ERd)	@ERd+	@ERd-	@+ERd	@-ERd	@(d:16,RdL,B)	@(d:16,Rd,W)	@(d:16,ERd,L)	@(d:32,RdL,B)	@(d:32,Rd,W)	@(d:32,ERd,L)	@aa:8	@aa:16	@aa:32
—	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL

2.2.93 RTE**ReTurn from Exception****[1] Assembly-Language Format**

RTE

[2] Operation

- When EXR is valid or this LSI is in maximum mode
 - @SP+ → EXR
 - @SP+ → CCR
 - @SP+ → PC
- Other than above
 - @SP+ → CCR
 - @SP+ → PC

[3] Operand Size

—

[4] Description

This instruction returns from an exception-handling routine by restoring the EXR, condition-code register (CCR) and program counter (PC) from the stack. Program execution continues from the address restored in PC. The CCR and PC contents at the time of execution of this instruction are lost.

[5] Condition Code

I	UI	H	U	N	Z	V	C
↑	↓	↓	↓	↓	↓	↓	↓

I: Restored from the corresponding bit on the stack.

UI: Restored from the corresponding bit on the stack.

H: Restored from the corresponding bit on the stack.

U: Restored from the corresponding bit on the stack.

N: Restored from the corresponding bit on the stack.

Z: Restored from the corresponding bit on the stack.

V: Restored from the corresponding bit on the stack.

C: Restored from the corresponding bit on the stack.

2.2.94 RTE/L**Load to registers and ReTurn from Exception****[1] Assembly-Language Format**

RTE/L <register list>

[2] Operation

- When EXR is valid or this LSI is in maximum mode
 - @SP+ → ERn (register list)
 - @SP+ → EXR
 - @SP+ → CCR
 - @SP+ → PC
- Other than above
 - @SP+ → ERn (register list)
 - @SP+ → CCR
 - @SP+ → PC

[3] Operand Size

Longword

[4] Description

This instruction restores the data from the stack to multiple registers selected by the register list. In addition, this instruction restores the EXR, condition-code register (CCR) and program counter (PC) from the stack. Program execution continues from the address restored in PC. General registers are restored in descending order of the register number. The number of registers to be restored is within the range of 1 to 4. The register numbers should be serial.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	↓	—	↓	↓	↓	↓

H: Restored from the corresponding bit on the stack.

N: Restored from the corresponding bit on the stack.

Z: Restored from the corresponding bit on the stack.

V: Restored from the corresponding bit on the stack.

C: Restored from the corresponding bit on the stack.

[6] Available General Registers

Usage	General Register
Register list	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Register list	Register direct (ERn)
	Register direct (ERn – ERn+1)
	Register direct (ERn – ERn+2)
	Register direct (ERn – ERn+3)

[8] Notes

1. If ER7 is selected and saved on the stack, the saved data is read from the stack, however, it is not restored to ER7.
2. The register numbers of ER0 and ER7 are not considered as serial numbers.

2.2.95 RTS

ReTurn from Subroutine

[1] Assembly-Language Format

RTS

[2] Operation

@SP+ → PC

[3] Operand Size

—

[4] Description

This instruction returns from a subroutine by restoring the program counter (PC) from the stack. Program execution continues from the address restored in PC. The PC contents at the time of execution of this instruction are lost.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

2.2.96 RTS/L**Load to registers and ReTurn from Subroutine****[1] Assembly-Language Format**

RTS/L <register list>

[2] Operation

@SP+ → ERn (register list)

@SP+ → PC

[3] Operand Size

Longword

[4] Description

This instruction restores the data from the stack to multiple registers selected by the register list. In addition, this instruction restores the program counter (PC) from the stack. Program execution continues from the address restored in PC. General registers are restored in descending order of the register number. The number of registers to be restored is within the range of 1 to 4. The register numbers should be serial.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Register list	ER0 to ER6

[7] Available Addressing Modes

Usage	Addressing Mode
Register list	Register direct (ERn)
	Register direct (ERn – ERn+1)
	Register direct (ERn – ERn+2)
	Register direct (ERn – ERn+3)

[8] Notes

1. If ER7 is selected and saved on the stack, the saved data is read from the stack, however, it is not restored to ER7.
2. The register numbers of ER0 and ER7 are not considered as serial numbers.

2.2.97

SHAL

SHift Arithmetic Left

[1] Assembly-Language Format

SHAL.Sz <EAd>

SHAL.Sz #2, <EAd>

[2] Operation

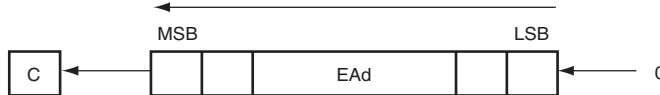
<EAd> (left arithmetic shift) → <EAd>

[3] Operand Size

Byte, word, longword

[4] Description

This instruction arithmetically shifts the bits in the contents of the destination location <EAd> (destination operand) by one or two bits to the left. The result is stored in the destination location. The lower bits are filled with 0s. The last bit shifted out is stored in the carry flag.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	↕	↕	↕	↕

H: Previous value remains unchanged.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Set to 1 if an overflow occurs; otherwise cleared to 0.

C: Stores the last bit shifted out.

[6] Available General Registers

Usage		General Register	
Destination	Address register	ER0 to ER7	
	Index register	R0L to R7L, R0 to R7, ER0 to ER7	
	Data register	Byte	R0L to R7L, R0H to R7H
		Word	R0 to R7, E0 to E7
Longword		ER0 to ER7	

[7] Available Addressing Modes

Number of shifted bits: Immediate

Source	Destination																	
	—	Rd	@ERd	@(d:2,ERd)	@(d:16,ERd)	@(d:32,ERd)	@ERd+	@ERd-	@+ERd	@-ERd	@(d:16,RdL,B)	@(d:16,Rd,W)	@(d:16,ERd,L)	@(d:32,RdL,B)	@(d:32,Rd,W)	@(d:32,ERd,L)	@aa:8	@aa:16
	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL

2.2.98 SHAR**SHift Arithmetic Right****[1] Assembly-Language Format**

SHAR.Sz <EAd>

SHAR.Sz #2, <EAd>

[2] Operation

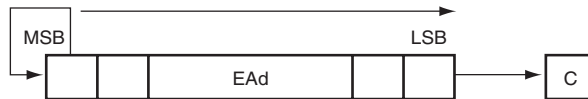
<EAd> (right arithmetic shift) → <EAd>

[3] Operand Size

Byte, word, longword

[4] Description

This instruction arithmetically shifts the bits in the contents of the destination location <EAd> (destination operand) by one or two bits to the right. The result is stored in the destination location. The upper bits are filled with the previous MSB (sign bit) value, therefore, the sign of the data is not changed. The last bit shifted out is stored in the carry flag.

**[5] Condition Code**

I	UI	H	U	N	Z	V	C
—	—	—	—	↕	↕	0	↕

H: Previous value remains unchanged.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Always cleared to 0.

C: Stores the last bit shifted out.

[6] Available General Registers

Usage	General Register		
Destination	Address register	ER0 to ER7	
	Index register	R0L to R7L, R0 to R7, ER0 to ER7	
	Data register	Byte	R0L to R7L, R0H to R7H
		Word	R0 to R7, E0 to E7
Longword		ER0 to ER7	

[7] Available Addressing Modes

Number of shifted bits: Immediate

Source	Destination																	
	—	Rd	@ERd	@(d:2,ERd)	@(d:16,ERd)	@(d:32,ERd)	@ERd+	@ERd-	@+ERd	@-ERd	@(d:16,RdL,B)	@(d:16,Rd,W)	@(d:16,ERd,L)	@(d:32,RdL,B)	@(d:32,Rd,W)	@(d:32,ERd,L)	@aa:8	@aa:16
	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL

2.2.99 SHLL (1)**SHift Logical Left****[1] Assembly-Language Format**

SHLL.Sz Rn, Rd
 SHLL.Sz #xx:5, Rd

[2] Operation

Rd (left logical shift) → Rd

[3] Operand Size

Byte, word, longword

[4] Description

This instruction shifts the bits in the contents of the destination register Rd (destination operand) by the specified bits to the left. The result is stored in the destination location. The lower bits are filled with 0s. The last bit shifted out is stored in the carry flag.

The number of bits to be shifted is specified by the immediate data or the contents of a general register Rn in which the lower 5 bits are valid. If a general register Rn is used for specifying the number of bits to be shifted, it should be specified as an 8-bit register.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↕	↕	0	↕

H: Previous value remains unchanged.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Always cleared to 0.

C: Stores the last bit shifted out.

[6] Available General Registers

Usage		General Register
Number of shifted bits		R0L to R7L, R0H to R7H
Destination	Byte	R0L to R7L, R0H to R7H
	Word	R0 to R7, E0 to E7
	Longword	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Number of shifted bits	Immediate
	Register direct
Destination	Register direct

[8] Note

The number of states for execution differs according to the number of shifted bits.

Number of Shifted Bits	Execution States*
0, 1, 2, 4, 5, 6, 8, 9, 10, 16, 17, 18	2
3, 7, 11, 12, 13, 14, 15, 19, 20, 21, 22, 23, 24, 25, 26, 27	3
28, 29, 30, 31	4

Note: * When the instruction code is stored in the on-chip memory.

2.2.100 SHLL (2)**SHift Logical Left****[1] Assembly-Language Format**

SHLL.Sz <EAd>

SHLL.Sz #n, <EAd>

n = 2, 4, 8, 16

[2] Operation

<EAd> (left logical shift) → <EAd>

[3] Operand Size

Byte, word, longword

[4] Description

This instruction shifts the bits in the contents of the destination location <EAd> (destination operand) by one or specified bits to the left. The result is stored in the destination location. The lower bits are filled with 0s. The last bit shifted out is stored in the carry flag.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↕	↕	0	↕

H: Previous value remains unchanged.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Always cleared to 0.

C: Stores the last bit shifted out.

[6] Available General Registers

Usage	General Register	
Destination	Address register	ER0 to ER7
	Index register	R0L to R7L, R0 to R7, ER0 to ER7
Data register	Byte	R0L to R7L, R0H to R7H
	Word	R0 to R7, E0 to E7
	Longword	ER0 to ER7

[7] Available Addressing Modes

Number of shifted bits: Immediate

Source	Destination																	
	—	Rd	@ERd	@(d:2,ERd)	@(d:16,ERd)	@(d:32,ERd)	@ERd+	@ERd-	@+ERd	@-ERd	@(d:16,RdL:B)	@(d:16,Rd:W)	@(d:16,ERd:L)	@(d:32,RdL:B)	@(d:32,Rd:W)	@(d:32,ERd:L)	@aa:8	@aa:16
	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL

2.2.101 SHLR (1)**SHift Logical Right****[1] Assembly-Language Format**

SHLR.Sz Rn, Rd

SHLR.Sz #xx:5, Rd

[2] Operation

Rd (shift logical right) → Rd

[3] Operand Size

Byte, word, longword

[4] Description

This instruction shifts the bits in the contents of the destination register (Rd) (destination operand) by the specified bits to the right. The result is stored in the destination location. The upper bits are filled with 0s. The last bit shifted out is stored in the carry flag.

The number of bits to be shifted is specified by the immediate data or the contents of a general register Rn in which the lower 5 bits are valid. If a general register Rn is used for specifying the number of bits, it should be specified as an 8-bit register.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↑	↑	0	↑

H: Previous value remains unchanged.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Always cleared to 0.

C: Stores the last bit shifted out.

[6] Available General Registers

Usage		General Register
Number of shifted bits		R0L to R7L, R0H to R7H
Destination	Byte	R0L to R7L, R0H to R7H
	Word	R0 to R7, E0 to E7
	Longword	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Number of shifted bits	Immediate
	Register direct
Destination	Register direct

[8] Note

The number of states for execution differs according to the number of shifted bits.

Number of Shifted Bits	Execution States*
0, 1, 2, 4, 5, 6, 8, 9, 10, 16, 17, 18	2
3, 7, 11, 12, 13, 14, 15, 19, 20, 21, 22, 23, 24, 25, 26, 27	3
28, 29, 30, 31	4

Note: * When the instruction code is stored in the on-chip memory.

2.2.102 SHLR (2)**SHift Logical Right****[1] Assembly-Language Format**

SHLR.Sz <EAd>

SHLR.Sz #2, <EAd>

n = 2, 4, 8, 16

[2] Operation

<EAd> (right logical shift) → <EAd>

[3] Operand Size

Byte, word, longword

[4] Description

This instruction shifts the bits in the contents of the destination location <EAd> (destination operand) by one or specified bits to the right. The result is stored in the destination location. The upper bits are filled with 0s. The last bit shifted out is stored in the carry flag.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↕	↕	0	↕

H: Previous value remains unchanged.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Always cleared to 0.

C: Stores the last bit shifted out.

[6] Available General Registers

Usage	General Register		
Destination	Address register	ER0 to ER7	
	Index register	R0L to R7L, R0 to R7, ER0 to ER7	
	Data register	Byte	R0L to R7L, R0H to R7H
		Word	R0 to R7, E0 to E7
Longword		ER0 to ER7	

[7] Available Addressing Modes

Number of shifted bits: Immediate

Source	Destination																	
	Rd	@ERd	@(d:2,ERd)	@(d:16,ERd)	@(d:32,ERd)	@ERd+	@ERd-	@+ERd	@-ERd	@(d:16,RdL,B)	@(d:16,Rd,W)	@(d:16,ERd,L)	@(d:32,RdL,B)	@(d:32,Rd,W)	@(d:32,ERd,L)	@aa:8	@aa:16	@aa:32
—	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL

2.2.103 SLEEP**SLEEP**

[1] Assembly-Language Format

SLEEP

[2] Operation

Program execution state → power-down mode

[3] Operand Size

—

[4] Description

When the SLEEP instruction is executed, the CPU enters power-down mode. Its internal state remains unchanged, but the CPU stops executing instructions and waits for an exception-handling request. When it receives an exception-handling request, the CPU exits the power-down mode and begins the exception-handling sequence. Interrupt requests other than NMI cannot clear the power-down mode if they are masked in the CPU.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Note

For details on power-down modes, see the hardware manual for the corresponding product.

2.2.104 STC (B)**STore from Control register****[1] Assembly-Language Format**

STC.B CCR, Rd

[2] Operation

CCR → Rd

[3] Operand Size

Byte

[4] Description

This instruction stores the CCR contents to an 8-bit register Rd.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Destination	R0L to R7L, R0H to R7H

[7] Available Addressing Mode

Usage	Addressing Mode
Destination	Register direct

2.2.105 STC (B)**STore from Control register****[1] Assembly-Language Format**

STC.B EXR, Rd

[2] Operation

EXR → Rd

[3] Operand Size

Byte

[4] Description

This instruction stores the EXR contents to an 8-bit register Rd.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Destination	R0L to R7L, R0H to R7H

[7] Available Addressing Mode

Usage	Addressing Mode
Destination	Register direct

2.2.106 STC (W)**STore from Control register****[1] Assembly-Language Format**

STC.W CCR, <EAd>

[2] Operation

CCR → <EAd>

[3] Operand Size

Word

[4] Description

This instruction stores the CCR contents to a destination location. Although CCR is a byte register, the store operation is performed in words. The CCR contents are stored in the upper byte. Undetermined data is stored in the lower byte.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Destination	Register direct
	Register indirect with displacement
	Register indirect with pre-decrement
	Absolute address (16 bits)
	Absolute address (32 bits)

2.2.107 STC (W)**STore from Control register****[1] Assembly-Language Format**

STC.W EXR, <EAd>

[2] Operation

EXR → <EAd>

[3] Operand Size

Word

[4] Description

This instruction stores the EXR contents to a destination location. Although EXR is a byte register, the store operation is performed in words. The EXR contents are stored in the upper byte. Undetermined data is stored in the lower byte.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Address register	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Destination	Register direct
	Register indirect with displacement
	Register indirect with pre-decrement
	Absolute address (16 bits)
	Absolute address (32 bits)

2.2.108 STC (L)**SStore from Control register****[1] Assembly-Language Format**

STC.L VBR,ERd

STC.L SBR,ERd

[2] Operation

VBR → ERd

SBR → ERd

[3] Operand Size

Longword

[4] Description

This instruction stores the contents of the base register (VBR or SBR) to a 32-bit register ERd.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Destination	ER0 to ER7

[7] Available Addressing Mode

Usage	Addressing Mode
Destination	Register direct

2.2.109 STM**STore from Multiple register****[1] Assembly-Language Format**

STM.L <register list>, @-SP

[2] Operation

ERn (register list) → @-SP

[3] Operand Size

Longword

[4] Description

This instruction saves multiple registers specified by a register list onto the stack. The registers are saved in ascending order of the register numbers. Two, three, or four registers can be saved by one STM instruction. The register numbers should be serial.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Source	ER0 to ER7

[7] Available Addressing Modes

Usage	Addressing Mode
Source	Register direct (ERn – ERn+1)
	Register direct (ERn – ERn+2)
	Register direct (ERn – ERn+3)

[8] Note

1. When ER7 is to be saved, the value after effective address calculation (ER7 is decremented by 4) is saved onto the stack.
2. The register numbers of ER0 and ER7 are not considered as serial numbers.

2.2.110 STMAC**STore from MAC register****[1] Assembly-Language Format**

STMAC MAC register, ERd

[2] Operation

MACH → ERd

or

MACL → ERd

[3] Operand Size

Longword

[4] Description

This instruction transfers the contents of a MAC register (MACH or MACL) to a general register. If the transfer is from MACH, the upper 22 bits are sign extended and then transferred to the general register.

This instruction is available only when the multiplier is supported.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↑*	↑*	↑*	—

H: Previous value remains unchanged.

N: Set to 1 if a MAC instruction resulted in a negative MAC register value; otherwise cleared to 0.

Z: Set to 1 if a MAC instruction resulted in a MAC register value of zero; otherwise cleared to 0.

V: Set to 1 if a MAC instruction resulted in an overflow; otherwise cleared to 0.

C: Previous value remains unchanged.

Note: * Execution of this instruction copies the N, Z, and V flag values from the multiplier to the condition-code register (CCR). If the STMAC instruction is executed after a CLRMAC or LDMAC instruction without executing the MAC instruction, the V flag will be 0 and the N and Z flags will have undetermined values.

[6] Available General Registers

Usage	General Register
Destination	ER0 to ER7

[7] Available Addressing Mode

Usage	Addressing Mode
Destination	Register direct

2.2.111 SUB**SUBtract binary****[1] Assembly-Language Format**

SUB.Sz <EAs>, <EAd>

[2] Operation

<EAd> - <EAs> → <EAd>

[3] Operand Size

Byte, word, longword

[4] Description

This instruction subtracts the source operand <EAs> from the contents of the destination location <EAd> (destination operand) and stores the result in the destination location.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	↑	—	↑	↑	↑	↑

H: Set to 1 if there is a borrow at the 4th bit from the MSB (bit 27, bit 11, or bit 3); otherwise cleared to 0.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Set to 1 if an overflow occurs; otherwise cleared to 0.

C: Set to 1 if there is a borrow at the MSB (bit 31, bit 15, or bit 7); otherwise cleared to 0.

[6] Available General Registers

Usage		General Register	
Source	Address register	ER0 to ER7	
	Index register	R0L to R7L, R0 to R7, ER0 to ER7	
	Data register	Byte	R0L to R7L, R0H to R7H
		Word	R0 to R7, E0 to E7
	Longword	ER0 to ER7	
Destination	Address register	ER0 to ER7	
	Index register	R0L to R7L, R0 to R7, ER0 to ER7	
	Data register	Byte	R0L to R7L, R0H to R7H
		Word	R0 to R7, E0 to E7
	Longword	ER0 to ER7	

[7] Available Addressing Modes

Source	Destination																	
	Rd	@ERd	@(d:2,ERd)	@(d:16,ERd)	@(d:32,ERd)	@ERd+	@ERd-	@+ERd	@-ERd	@(d:16,RdL.B)	@(d:16,Rd.W)	@(d:16,ERd.L)	@(d:32,RdL.B)	@(d:32,Rd.W)	@(d:32,ERd.L)	@aa:8	@aa:16	@aa:32
#x:3(1-7)	WL	W															W	W
#x:8		B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
#x:16	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL
#x:32	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
Rs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL
@ERs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:2,ERs)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:16,ERs)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:32,ERs)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@ERs+	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@ERs-	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@+ERs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@-ERs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:16,RsL.B)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:16,Rs.W)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:16,ERs.L)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:32,RsL.B)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:32,Rs.W)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@(d:32,ERs.L)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@aa:8	B																	
@aa:16	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL
@aa:32	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL

2.2.112 SUBS**SUBtract with Sign extension****[1] Assembly-Language Format**

SUBS #1, ERd

SUBS #2, ERd

SUBS #4, ERd

[2] Operation

ERd - 1 → ERd

ERd - 2 → ERd

ERd - 4 → ERd

[3] Operand Size

Longword

[4] Description

This instruction subtracts the immediate value 1, 2, or 4 from the contents of a 32-bit register ERd (destination operand). Unlike the SUB instruction, the condition code remains unchanged.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Destination	ER0 to ER7

[7] Available Addressing Mode

Usage	Addressing Mode
Destination	Register direct

2.2.113 SUBX**SUBtract with eXtend carry****[1] Assembly-Language Format**

SUBX.Sz <EAs>, <EAd>

[2] Operation $\langle EAd \rangle - \langle EAs \rangle - C \rightarrow \langle EAd \rangle$ **[3] Operand Size**

Byte, word, longword

[4] Description

This instruction subtracts the source operand <EAs> and the carry flag from the contents of the destination location <EAd> (destination operand) and stores the result in the destination location.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	↑	—	↑	↑	↑	↑

H: Set to 1 if there is a borrow at the 4th bit from the MSB (bit 27, bit 11, or bit 3); otherwise cleared to 0.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Previous value remains unchanged if the result is zero; otherwise cleared to 0.

V: Set to 1 if an overflow occurs; otherwise cleared to 0.

C: Set to 1 if there is a borrow at the MSB (bit 31, bit 15, or bit 7); otherwise cleared to 0.

[6] Available General Registers

Usage		General Register	
Source	Address register		ER0 to ER7
	Data register	Byte	R0L to R7L, R0H to R7H
		Word	R0 to R7, E0 to E7
		Longword	ER0 to ER7
Destination	Address register		ER0 to ER7
	Data register	Byte	R0L to R7L, R0H to R7H
		Word	R0 to R7, E0 to E7
		Longword	ER0 to ER7

[7] Available Addressing Modes

Source	Destination																		
	Rd	@ERd	@(d:2,ERd)	@(d:16,ERd)	@(d:32,ERd)	@ERd+	@ERd-	@+ERd	@-ERd	@(d:16,RdL,B)	@(d:16,Rd,W)	@(d:16,ERd,L)	@(d:32,RdL,B)	@(d:32,Rd,W)	@(d:32,ERd,L)	@aa:8	@aa:16	@aa:32	
#x:3(1-7)																			
#x:8	B	B						B											
#x:16	W	W						W											
#x:32	L	L						L											
Rs	BWL	BWL						BWL											
@ERs	BWL	BWL																	
@(d:2,ERs)																			
@(d:16,ERs)																			
@(d:32,ERs)																			
@ERs+																			
@ERs-	BWL							BWL											
@+ERs																			
@-ERs																			
@(d:16,RsL,B)																			
@(d:16,Rs,W)																			
@(d:16,ERs,L)																			
@(d:32,RsL,B)																			
@(d:32,Rs,W)																			
@(d:32,ERs,L)																			
@aa:8																			
@aa:16																			
@aa:32																			

2.2.114 TAS**Test And Set****[1] Assembly-Language Format**

TAS @ERd

[2] Operation

@ERd – 0 → set CCR

1 → (<bit 7> of @ERd)

[3] Operand Size

Byte

[4] Description

This instruction tests the contents of a memory location by comparing it with zero, and sets the condition-code register according to the result. Then it sets the MSB (bit 7) of the operand to 1.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↕	↕	0	—

H: Previous value remains unchanged.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Always cleared to 0.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register
Address register	ER0 to ER7

[7] Available Addressing Mode

Usage	Addressing Mode
Destination	Register indirect

2.2.115 TRAPA**TRAP Always****[1] Assembly-Language Format**

TRAPA #x:2

[2] Operation

- When EXR is valid or this LSI is in maximum mode
 PC → @-SP
 CCR → @-SP
 EXR → @-SP
 <Vector> → PC
- Other than above
 PC → @-SP
 CCR → @-SP
 <Vector> → PC

[3] Operand Size**[4] Description**

This instruction pushes the program counter (PC) and condition-code register (CCR) onto the stack, then sets the I bit to 1. If the extended control register (EXR) is valid, EXR is also saved onto the stack. However, bits I2 to I0 are not modified. Next execution branches to a new address given by the contents of the vector address corresponding to a specified number. The PC value pushed onto the stack is the start address of the instruction immediately after this instruction.

#x	Vector Address	
	Normal Mode	Other than Normal Mode
0	H'0010 to H'0011	H'000020 to H'000023
1	H'0012 to H'0013	H'000024 to H'000027
2	H'0014 to H'0015	H'000028 to H'00002B
3	H'0016 to H'0017	H'00002C to H'00002F

[5] Condition Code

I	UI	H	U	N	Z	V	C
1	Δ^*	—	—	—	—	—	—

I: Always set to 1.

UI: See note.

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

Note: * The UI bit is set to 1 when used as an interrupt mask bit. However, it retains the previous value when used as a user bit. For details, see the hardware manual for the corresponding product.

[6] Available Addressing Mode

Usage	Addressing Mode
Bit number specification	Immediate

[7] Note

The stack and vector structure differ according to the CPU operating mode and whether EXR is valid or invalid.

2.2.116 XOR**eXclusive OR logical**

[1] Assembly-Language Format

XOR.Sz <EAs>, <EAd>

[2] Operation<EAd> \oplus <EAs> \rightarrow <EAd>**[3] Operand Size**

Byte, word, longword

[4] Description

This instruction exclusively ORs the contents of the destination location <EAd> (destination operand) with the source operand and stores the result in the destination location.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	↕	↕	0	—

H: Previous value remains unchanged.

N: Set to 1 if the result is negative; otherwise cleared to 0.

Z: Set to 1 if the result is zero; otherwise cleared to 0.

V: Always cleared to 0.

C: Previous value remains unchanged.

[6] Available General Registers

Usage	General Register		
Source	Address register	ER0 to ER7	
	Index register	R0L to R7L, R0 to R7, ER0 to ER7	
	Data register	Byte	R0L to R7L, R0H to R7H
		Word	R0 to R7, E0 to E7
	Longword	ER0 to ER7	
Destination	Address register	ER0 to ER7	
	Index register	R0L to R7L, R0 to R7, ER0 to ER7	
	Data register	Byte	R0L to R7L, R0H to R7H
		Word	R0 to R7, E0 to E7
	Longword	ER0 to ER7	

[7] Available Addressing Modes

Source	Destination																		
	Rd	@ERd	@(d:2,ERd)	@(d:16,ERd)	@(d:32,ERd)	@ERd+	@ERd-	@+ERd	@-ERd	@(d:16,RdL.B)	@(d:16,Rd.W)	@(d:16,ERd.L)	@(d:32,RdL.B)	@(d:32,Rd.W)	@(d:32,ERd.L)	@aa:8	@aa:16	@aa:32	
#x:3(1-7)																			
#x:8	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
#x:16	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL	WL		WL	WL	
#x:32	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L		L	L	
Rs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL	BWL
@ERs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@(d:2,ERs)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@(d:16,ERs)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@(d:32,ERs)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@ERs+	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@ERs-	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@+ERs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@-ERs	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@(d:16,RsL.B)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@(d:16,Rs.W)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@(d:16,ERs.L)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@(d:32,RsL.B)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@(d:32,Rs.W)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@(d:32,ERs.L)	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@aa:8	B																		
@aa:16	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL
@aa:32	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL	BWL		BWL	BWL	BWL

2.2.117 XORC**eXclusive OR Control register****[1] Assembly-Language Format**

XORC #xx:8, CCR

[2] OperationCCR \oplus #IMM \rightarrow CCR**[3] Operand Size**

Byte

[4] Description

This instruction exclusively ORs the contents of the condition-code register (CCR) with immediate data and stores the result in CCR. No interrupt requests, including NMI, are accepted immediately after execution of this instruction.

[5] Condition Code

I	UI	H	U	N	Z	V	C
↑	↑	↑	↑	↑	↑	↑	↑

I: Stores the corresponding bit of the result.

UI: Stores the corresponding bit of the result.

H: Stores the corresponding bit of the result.

U: Stores the corresponding bit of the result.

N: Stores the corresponding bit of the result.

Z: Stores the corresponding bit of the result.

V: Stores the corresponding bit of the result.

C: Stores the corresponding bit of the result.

[6] Available Addressing Mode

Immediate

2.2.118 XORC**eXclusive OR Control register****[1] Assembly-Language Format**

XORC #xx:8, EXR

[2] Operation $EXR \oplus \#IMM \rightarrow EXR$ **[3] Operand Size**

Byte

[4] Description

This instruction exclusively ORs the contents of the extended control register (EXR) with immediate data and stores the result in EXR. No interrupt requests, including NMI, are accepted immediately after execution of this instruction.

[5] Condition Code

I	UI	H	U	N	Z	V	C
—	—	—	—	—	—	—	—

H: Previous value remains unchanged.

N: Previous value remains unchanged.

Z: Previous value remains unchanged.

V: Previous value remains unchanged.

C: Previous value remains unchanged.

[6] Available Addressing Mode

Immediate

Data transfer (5)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode										Operation ⁵					Condition Codes ^{6,2}					
				Min.	16-Bit Instruction Fetch #xxx	Rn	@ERn	@ (d,ERn)	@ (d,Rn),W(Rn,L)	@ ERn/@ ERn+/@ ERn+/@ +ERn	@ aa:9/@ aa:16/@ aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C		
																							Number of Execution Stages ¹	
MOV	MOV.B @ERs, @(d:2,ERd)	B	3	4				D	S				@ERs → @(1/2/3+ERd)	ERs32-1→ERs32			↑	↓	0	—				
	MOV.B @ERs, @(d:16,ERd)	B	3	5			D	S					@ERs → @(d:16+ERd)	ERs32-1→ERs32			↑	↓	0	—				
	MOV.B @ERs, @(d:32,ERd)	B	4	3	6			D	S				@ERs → @(d:32+ERd)	ERs32-1→ERs32			↑	↓	0	—				
	MOV.B @ERs, @(d:16,Rd,B)	B	3	3	5			D	S				@ERs → @(d:16+RdL)	ERs32-1→ERs32			↑	↓	0	—				
	MOV.B @ERs, @(d:16,Rd,W)	B	3	3	5			D	S				@ERs → @(d:16+Rd)	ERs32-1→ERs32			↑	↓	0	—				
	MOV.B @ERs, @(d:16,ERd,L)	B	3	3	5			D	S				@ERs → @(d:16+ERd)	ERs32-1→ERs32			↑	↓	0	—				
	MOV.B @ERs, @(d:32,Rd,B)	B	4	3	6			D	S				@ERs → @(d:32+RdL)	ERs32-1→ERs32			↑	↓	0	—				
	MOV.B @ERs, @(d:32,Rd,W)	B	4	3	6			D	S				@ERs → @(d:32+Rd)	ERs32-1→ERs32			↑	↓	0	—				
	MOV.B @ERs, @(d:32,ERd,L)	B	4	3	6			D	S				@ERs → @(d:32+ERd)	ERs32-1→ERs32			↑	↓	0	—				
	MOV.B @ERs, @aa:16	B	3	3	5				S	D			@ERs → @aa:16	ERs32-1→ERs32			↑	↓	0	—				
	MOV.B @ERs, @aa:32	B	4	3	6				S	D			@ERs → @aa:32	ERs32-1→ERs32			↑	↓	0	—				
	MOV.B @+ERs, @ERd	B	2	4	4			D	S		ERs32+1→ERs32		@ERs → @ERd				↑	↓	0	—				
	MOV.B @+ERs, @ERd+	B	2	4	4				S		ERs32+1→ERs32		@ERs → @ERd	ERd32+1→ERd32			↑	↓	0	—				
	MOV.B @+ERs, @ERd-	B	2	4	4				S		ERs32+1→ERs32		@ERs → @ERd	ERd32-1→ERd32			↑	↓	0	—				
	MOV.B @+ERs, @+ERd	B	2	4	4				S		ERs32+1→ERs32	ERd32+1→ERd32	@ERs → @ERd				↑	↓	0	—				
	MOV.B @+ERs, @-ERd	B	2	4	4				S		ERs32+1→ERs32	ERd32-1→ERd32	@ERs → @ERd				↑	↓	0	—				
	MOV.B @+ERs, @(d:2,ERd)	B	2	4	4			D	S		ERs32+1→ERs32		@ERs → @(1/2/3+ERd)				↑	↓	0	—				
	MOV.B @+ERs, @(d:16,ERd)	B	3	4	5			D	S		ERs32+1→ERs32		@ERs → @(d:16+ERd)				↑	↓	0	—				
	MOV.B @+ERs, @(d:32,ERd)	B	4	4	6			D	S		ERs32+1→ERs32		@ERs → @(d:32+ERd)				↑	↓	0	—				
	MOV.B @+ERs, @(d:16,Rd,B)	B	3	4	5			D	S		ERs32+1→ERs32		@ERs → @(d:16+RdL)				↑	↓	0	—				
	MOV.B @+ERs, @(d:16,Rd,W)	B	3	4	5				S		ERs32+1→ERs32		@ERs → @(d:16+Rd)				↑	↓	0	—				
	MOV.B @+ERs, @(d:16,ERd,L)	B	3	4	5			D	S		ERs32+1→ERs32		@ERs → @(d:16+ERd)				↑	↓	0	—				
	MOV.B @+ERs, @(d:32,Rd,B)	B	4	4	6			D	S		ERs32+1→ERs32		@ERs → @(d:32+RdL)				↑	↓	0	—				
	MOV.B @+ERs, @(d:32,Rd,W)	B	4	4	6			D	S		ERs32+1→ERs32		@ERs → @(d:32+Rd)				↑	↓	0	—				
	MOV.B @+ERs, @(d:32,ERd,L)	B	4	4	6			D	S		ERs32+1→ERs32		@ERs → @(d:32+ERd)				↑	↓	0	—				

Data transfer (18)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹ Fxx	Addressing Mode							Operation ⁵					Condition Codes ^{6,2}					
						Rn	@(d,ERn)	@(d,Rn,WRn,LL)	@(d,ERn)/@ERn+/@ERn+/@+ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C		
MOV	MOV.W @ERs-, @(d:2,ERd)	W	3	4				D	S					@ERs → @(2/4/6+ERd)	ERs32-2→ERs32				↑	↓	0	—	
	MOV.W @ERs-, @(d:16,ERd)	W	3	5				D	S					@ERs → @(d:16+ERd)	ERs32-2→ERs32				↑	↓	0	—	
	MOV.W @ERs-, @(d:32,ERd)	W	4	3	6			D	S					@ERs → @(d:32+ERd)	ERs32-2→ERs32				↑	↓	0	—	
	MOV.W @ERs-, @(d:16,Rd,B)	W	3	3	5			D	S					@ERs → @(d:16+RdL<<1)	ERs32-2→ERs32				↑	↓	0	—	
	MOV.W @ERs-, @(d:16,Rd,W)	W	3	3	5			D	S					@ERs → @(d:16+Rd<<1)	ERs32-2→ERs32				↑	↓	0	—	
	MOV.W @ERs-, @(d:16,ERd,L)	W	3	3	5			D	S					@ERs → @(d:16+ERd<<1)	ERs32-2→ERs32				↑	↓	0	—	
	MOV.W @ERs-, @(d:32,Rd,B)	W	4	3	6			D	S					@ERs → @(d:32+RdL<<1)	ERs32-2→ERs32				↑	↓	0	—	
	MOV.W @ERs-, @(d:32,Rd,W)	W	4	3	6			D	S					@ERs → @(d:32+Rd<<1)	ERs32-2→ERs32				↑	↓	0	—	
	MOV.W @ERs-, @(d:32,ERd,L)	W	4	3	6			D	S					@ERs → @(d:32+ERd<<1)	ERs32-2→ERs32				↑	↓	0	—	
	MOV.W @ERs-, @aa:16	W	3	3	5				S	D				@ERs → @aa:16	ERs32-2→ERs32				↑	↓	0	—	
	MOV.W @ERs-, @aa:32	W	4	3	6				S	D				@ERs → @aa:32	ERs32-2→ERs32				↑	↓	0	—	
	MOV.W @+ERs, @ERd	W	2	4	4				S		ERs32+2→ERs32			@ERs → @ERd				↑	↓	0	—		
	MOV.W @+ERs, @ERd+	W	2	4	4				S		ERs32+2→ERs32	ERd32		@ERs → @ERd	ERd32+2→ERd32				↑	↓	0	—	
	MOV.W @+ERs, @ERd-	W	2	4	4				S		ERs32+2→ERs32			@ERs → @ERd	ERd32-2→ERd32				↑	↓	0	—	
	MOV.W @+ERs, @+ERd	W	2	4	4				S		ERs32+2→ERs32	ERd32+2→ERd32		@ERs → @ERd				↑	↓	0	—		
	MOV.W @+ERs, @-ERd	W	2	4	4				S		ERs32+2→ERs32	ERd32-2→ERd32		@ERs → @ERd				↑	↓	0	—		
	MOV.W @+ERs, @(d:2,ERd)	W	2	4	4			D	S		ERs32+2→ERs32			@ERs → @(2/4/6+ERd)				↑	↓	0	—		
	MOV.W @+ERs, @(d:16,ERd)	W	3	4	5			D	S		ERs32+2→ERs32			@ERs → @(d:16+ERd)				↑	↓	0	—		
	MOV.W @+ERs, @(d:32,ERd)	W	4	4	6			D	S		ERs32+2→ERs32			@ERs → @(d:32+ERd)				↑	↓	0	—		
	MOV.W @+ERs, @(d:16,Rd,B)	W	3	4	5			D	S		ERs32+2→ERs32			@ERs → @(d:16+RdL<<1)				↑	↓	0	—		
	MOV.W @+ERs, @(d:16,Rd,W)	W	3	4	5			D	S		ERs32+2→ERs32			@ERs → @(d:16+Rd<<1)				↑	↓	0	—		
	MOV.W @+ERs, @(d:16,ERd,L)	W	3	4	5			D	S		ERs32+2→ERs32			@ERs → @(d:16+ERd<<1)				↑	↓	0	—		
	MOV.W @+ERs, @(d:32,Rd,B)	W	4	4	6			D	S		ERs32+2→ERs32			@ERs → @(d:32+RdL<<1)				↑	↓	0	—		
	MOV.W @+ERs, @(d:32,Rd,W)	W	4	4	6			D	S		ERs32+2→ERs32			@ERs → @(d:32+Rd<<1)				↑	↓	0	—		
	MOV.W @+ERs, @(d:32,ERd,L)	W	4	4	6			D	S		ERs32+2→ERs32			@ERs → @(d:32+ERd<<1)				↑	↓	0	—		

Data transfer (29)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode							Operation ^s					Condition Codes ^{s2}								
				Min.	16-Bit Instruction Fetch Execution Status ¹	FX	Rn	@ERn	@(d,ERn)	@(d,Rn,ERn,W,ERn,L)	@ERn/@ERn+/@ERn+/@+ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C	
MOV	MOV.L ERs, @ERd	L	2 3 3	S								ERd32-4 → ERd32			ERs → @ERd									
	MOV.L ERs, @(d:2,ERd)	L	2 3 3	S				D							ERs → @(4/8/12+ERd)									
	MOV.L ERs, @(d:16,ERd)	L	3 3 4	S				D							ERs → @(d:16+ERd)									
	MOV.L ERs, @(d:32,ERd)	L	4 3 5	S				D							ERs → @(d:32+ERd)									
	MOV.L ERs, @(d:16,Rd.B)	L	3 3 4	S				D							ERs → @(d:16+RdL<<2)									
	MOV.L ERs, @(d:16,Rd.W)	L	3 3 4	S				D							ERs → @(d:16+Rd<<2)									
	MOV.L ERs, @(d:16,ERdL)	L	3 3 4	S				D							ERs → @(d:16+ERd<<2)									
	MOV.L ERs, @(d:32,Rd.B)	L	4 3 5	S				D							ERs → @(d:32+RdL<<2)									
	MOV.L ERs, @(d:32,Rd.W)	L	4 3 5	S				D							ERs → @(d:32+Rd<<2)									
	MOV.L ERs, @(d:32,ERdL)	L	4 3 5	S				D							ERs → @(d:32+ERd<<2)									
	MOV.L ERs, @aa:16	L	3 2 4	S						D					ERs → @aa:16									
	MOV.L ERs, @aa:32	L	4 2 5	S						D					ERs → @aa:32									
	MOV.L @ERs, ERd	L	2 2 3	D	S										@ERs → ERd									
	MOV.L @ERs+, ERd	L	2 2 3	D	S										@ERs → ERd	ERs32+4 → ERs32								
	MOV.L @ERs-, ERd	L	2 2 3	D					S						@ERs → ERd	ERs32-4 → ERs32								
	MOV.L @+ERs, ERd	L	2 3 3	D					S			ERs32+4 → ERs32			@ERs → ERd									
	MOV.L @-ERs, ERd	L	2 3 3	D					S			ERs32-4 → ERs32			@ERs → ERd									
	MOV.L @(d:2,ERs), ERd	L	2 3 3	D	S										@(4/8/12+ERs) → ERd									
	MOV.L @(d:16,ERs), ERd	L	3 3 4	D	S										@(d:16+ERs) → ERd									
	MOV.L @(d:32,ERs), ERd	L	4 3 5	D	S										@(d:32+ERs) → ERd									
	MOV.L @(d:16,Rs.B), ERd	L	3 3 4	D					S						@(d:16+RsL<<2) → ERd									
	MOV.L @(d:16,Rs.W), ERd	L	3 3 4	D					S						@(d:16+Rs<<2) → ERd									
	MOV.L @(d:16,ERs.L), ERd	L	3 3 4	D					S						@(d:16+ERs<<2) → ERd									
	MOV.L @(d:32,Rs.B), ERd	L	4 3 5	D					S						@(d:32+RsL<<2) → ERd									
	MOV.L @(d:32,Rs.W), ERd	L	4 3 5	D					S						@(d:32+Rs<<2) → ERd									

Data transfer (30)

Instruction	Mnemonic	Size	Instruction Length			Addressing Mode								Operation ⁵					Condition Code ^{6,2}																			
			Min.	Max.	16-Bit Instruction Fetch Execution Status ¹	Rn	@ERn	@ (d, ERn)	@ (d, Rn.L, ERn.W, ERn.L)	@ ERn / @ ERn + / @ ERn - / @ ERn	@ aa:8 / @ aa:16 / @ aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C																
			#xx	D	S																		D	S	D	S	D	S	D									
MOV	MOV.L @ (d:32, ERs.L), ERd	L	4	3	5	D		S																														
	MOV.L @aa:16, ERd	L	3	2	4	D																																
	MOV.L @aa:32, ERd	L	4	2	5	D				S																												
	MOV.L @ERs, @ERd	L	2	3	4			S																														
	MOV.L @ERs, @ERd+	L	2	3	4			S		D																												
	MOV.L @ERs, @ERd-	L	2	3	4			S		D																												
	MOV.L @ERs, @+ERd	L	2	3	4			S		D				ERd32+4→ERd32																								
	MOV.L @ERs, @-ERd	L	2	3	4			S		D				ERd32-4→ERd32																								
	MOV.L @ERs, @ (d:2, ERd)	L	2	3	4			S	D					@ERs → @ (4/8/12+ERd)																								
	MOV.L @ERs, @ (d:16, ERd)	L	3	3	5			S	D					@ERs → @ (d:16+ERd)																								
	MOV.L @ERs, @ (d:32, ERd)	L	4	3	6			S	D					@ERs → @ (d:32+ERd)																								
	MOV.L @ERs, @ (d:16, Rd.B)	L	3	3	5			S	D					@ERs → @ (d:16+RdL<<2)																								
	MOV.L @ERs, @ (d:16, Rd.W)	L	3	3	5			S	D					@ERs → @ (d:16+Pd<<2)																								
	MOV.L @ERs, @ (d:16, ERd.L)	L	3	3	5			S	D					@ERs → @ (d:16+ERd<<2)																								
	MOV.L @ERs, @ (d:32, Rd.B)	L	4	3	6			S	D					@ERs → @ (d:32+RdL<<2)																								
	MOV.L @ERs, @ (d:32, Rd.W)	L	4	3	6			S	D					@ERs → @ (d:32+Pd<<2)																								
	MOV.L @ERs, @ (d:32, ERd.L)	L	4	3	6			S	D					@ERs → @ (d:32+ERd<<2)																								
	MOV.L @ERs, @aa:16	L	3	3	5			S		D				@ERs → @aa:16																								
	MOV.L @ERs, @aa:32	L	4	3	6			S		D				@ERs → @aa:32																								
	MOV.L @ERs+, @ERd	L	2	3	4			D		S				@ERs → @ERd	ERs32+4→ERs32																							
	MOV.L @ERs+, @ERd+	L	2	3	4					S				@ERs → @ERd	ERs32+4→ERs32	ERd32+4→ERd32																						
	MOV.L @ERs+, @ERd-	L	2	3	4					S				@ERs → @ERd	ERs32+4→ERs32	ERd32-4→ERd32																						
	MOV.L @ERs+, @+ERd	L	2	3	4					S				@ERs → @ERd	ERs32+4→ERs32																							
	MOV.L @ERs+, @-ERd	L	2	3	4					S				@ERs → @ERd	ERd32-4→ERd32																							

Data transfer (31)

Instruction n	Mnemonic	Size	Instruction Length	Addressing Mode								Operation ⁵					Condition Codes ^{6,7}							
				Min.	16-Bit Instruction Fetch Execution Status ¹ #xx	Rn	@ERn	@ (d,ERn)	@ (d,Rn,ERn,W,ERn,L)	@ ERn/@ ERn+/@ ERn+/@ +ERn	@ aa:9/@ aa:16/@ aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C		
																							Number of Execution Stages ²	Operation 1
MOV	MOV.L @ERs+, @(d:2,ERd)	L	2	3	4			D		S			@ERs → @(4/8/12+ERd)	ERs32+4→ERs32										
	MOV.L @ERs+, @(d:16,ERd)	L	3	3	5			D		S			@ERs → @(d:16+ERd)	ERs32+4→ERs32										
	MOV.L @ERs+, @(d:32,ERd)	L	4	3	6			D		S			@ERs → @(d:32+ERd)	ERs32+4→ERs32										
	MOV.L @ERs+, @(d:16,Rd,B)	L	3	3	5			D		S			@ERs → @(d:16+RdL<<2)	ERs32+4→ERs32										
	MOV.L @ERs+, @(d:16,Rd,W)	L	3	3	5			D		S			@ERs → @(d:16+Rd<<2)	ERs32+4→ERs32										
	MOV.L @ERs+, @(d:16,ERd,L)	L	3	3	5			D		S			@ERs → @(d:16+ERd<<2)	ERs32+4→ERs32										
	MOV.L @ERs+, @(d:32,Rd,B)	L	4	3	6			D		S			@ERs → @(d:32+RdL<<2)	ERs32+4→ERs32										
	MOV.L @ERs+, @(d:32,Rd,W)	L	4	3	6			D		S			@ERs → @(d:32+Rd<<2)	ERs32+4→ERs32										
	MOV.L @ERs+, @(d:32,ERd,L)	L	4	3	6			D		S			@ERs → @(d:32+ERd<<2)	ERs32+4→ERs32										
	MOV.L @ERs+, @aa:16	L	3	3	5					S	D		@ERs → @aa:16	ERs32+4→ERs32										
	MOV.L @ERs+, @aa:32	L	4	3	6					S	D		@ERs → @aa:32	ERs32+4→ERs32										
	MOV.L @ERs-, @ERd	L	2	3	4			D		S			@ERs → @ERd	ERs32-4→ERs32										
	MOV.L @ERs-, @ERd+	L	2	3	4					S			@ERs → @ERd	ERs32-4→ERs32	ERd32+4→ERd32									
	MOV.L @ERs-, @ERd-	L	2	3	4					S			@ERs → @ERd	ERs32-4→ERs32	ERd32-4→ERd32									
	MOV.L @ERs-, @+ERd	L	2	3	4					S		ERd32+4→ERd32	@ERs → @ERd	ERs32-4→ERs32										
	MOV.L @ERs-, @-ERd	L	2	3	4					S		ERd32-4→ERd32	@ERs → @ERd	ERs32-4→ERs32										
	MOV.L @ERs-, @(d:2,ERd)	L	2	3	4				D	S			@ERs → @(4/8/12+ERd)	ERs32-4→ERs32										
	MOV.L @ERs-, @(d:16,ERd)	L	3	3	5			D		S			@ERs → @(d:16+ERd)	ERs32-4→ERs32										
	MOV.L @ERs-, @(d:32,ERd)	L	4	3	6			D		S			@ERs → @(d:32+ERd)	ERs32-4→ERs32										
	MOV.L @ERs-, @(d:16,Rd,B)	L	3	3	5			D		S			@ERs → @(d:16+RdL<<2)	ERs32-4→ERs32										
	MOV.L @ERs-, @(d:16,Rd,W)	L	3	3	5			D		S			@ERs → @(d:16+Rd<<2)	ERs32-4→ERs32										
	MOV.L @ERs-, @(d:16,ERd,L)	L	3	3	5			D		S			@ERs → @(d:16+ERd<<2)	ERs32-4→ERs32										
	MOV.L @ERs-, @(d:32,Rd,B)	L	4	3	6			D		S			@ERs → @(d:32+RdL<<2)	ERs32-4→ERs32										
	MOV.L @ERs-, @(d:32,Rd,W)	L	4	3	6			D		S			@ERs → @(d:32+Rd<<2)	ERs32-4→ERs32										
	MOV.L @ERs-, @(d:32,ERd,L)	L	4	3	6			D		S			@ERs → @(d:32+ERd<<2)	ERs32-4→ERs32										

Data transfer (32)

Instruction	Mnemonic	Size	Instruction Length	Min.	Addressing Mode										Operation ^s					Condition Codes ^{s2}																		
					16-Bit Instruction Fetch Execution Status ¹	Prx	Rn	@ERn	@(d,ERn)	@ (d,Rn),Rn,W(ERn,L)	@ ERn/ @ ERn+/ @ ERn+/ @ ERn	@ aa:9/ @ aa:16/ @ aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C															
																								Number of Execution Stages ¹														
MOV	MOV.L @ERs-, @aa:16	L	3	3	5																																	
	MOV.L @ERs-, @aa:32	L	4	3	6																																	
	MOV.L @+ERs, @ERd	L	2	4	4																																	
	MOV.L @+ERs, @ERd+	L	2	4	4																																	
	MOV.L @+ERs, @ERd-	L	2	4	4																																	
	MOV.L @+ERs, @+ERd	L	2	4	4																																	
	MOV.L @+ERs, @-ERd	L	2	4	4																																	
	MOV.L @+ERs, @(d:2,ERd)	L	2	4	4																																	
	MOV.L @+ERs, @(d:16,ERd)	L	3	4	5																																	
	MOV.L @+ERs, @(d:32,ERd)	L	4	4	6																																	
	MOV.L @+ERs, @(d:16,Rd,B)	L	3	4	5																																	
	MOV.L @+ERs, @(d:16,Rd,W)	L	3	4	5																																	
	MOV.L @+ERs, @(d:16,ERd,L)	L	3	4	5																																	
	MOV.L @+ERs, @(d:32,Rd,B)	L	4	4	6																																	
	MOV.L @+ERs, @(d:32,Rd,W)	L	4	4	6																																	
	MOV.L @+ERs, @(d:32,ERd,L)	L	4	4	6																																	
	MOV.L @+ERs, @aa:16	L	3	4	5																																	
	MOV.L @+ERs, @aa:32	L	4	4	6																																	
	MOV.L @-ERs, @ERd	L	2	4	4																																	
	MOV.L @-ERs, @ERd+	L	2	4	4																																	
	MOV.L @-ERs, @ERd-	L	2	4	4																																	
	MOV.L @-ERs, @+ERd	L	2	4	4																																	
	MOV.L @-ERs, @-ERd	L	2	4	4																																	
	MOV.L @-ERs, @(d:2,ERd)	L	2	4	4																																	
	MOV.L @-ERs, @(d:16,ERd)	L	3	4	5																																	

Data transfer (33)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	Number of 16-Bit Instruction Fetch Execution Stages ¹	Addressing Mode							Operation ⁵					Condition Codes ²												
							Prx	Rn	@(ERn)	@(d,ERn)	@(d,Rn,ERn,W,ERn,L)	@(ERn)/@ERn+/@ERn-/@+ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C							
MOV	MOV.L @(ERs),@(d:32,ERd)	L	4	4	6				D	S				ERs32-4→ERs32			@ERs	→	@(d:32+ERd)												
	MOV.L @(ERs),@(d:16,Rd.B)	L	3	4	5				D	S				ERs32-4→ERs32			@ERs	→	@(d:16+RdL<<2)												
	MOV.L @(ERs),@(d:16,Rd.W)	L	3	4	5				D	S				ERs32-4→ERs32			@ERs	→	@(d:16+Pd<<2)												
	MOV.L @(ERs),@(d:16,ERd.L)	L	3	4	5				D	S				ERs32-4→ERs32			@ERs	→	@(d:16+ERd<<2)												
	MOV.L @(ERs),@(d:32,Rd.B)	L	4	4	6				D	S				ERs32-4→ERs32			@ERs	→	@(d:32+RdL<<2)												
	MOV.L @(ERs),@(d:32,Rd.W)	L	4	4	6				D	S				ERs32-4→ERs32			@ERs	→	@(d:32+Pd<<2)												
	MOV.L @(ERs),@(d:32,ERd.L)	L	4	4	6				D	S				ERs32-4→ERs32			@ERs	→	@(d:32+ERd<<2)												
	MOV.L @(ERs),@aa:16	L	3	4	5					S	D			ERs32-4→ERs32			@ERs	→	@aa:16												
	MOV.L @(ERs),@aa:32	L	4	4	6					S	D			ERs32-4→ERs32			@ERs	→	@aa:32												
	MOV.L @(d:2,ERs),@ERd	L	2	4	4				D	S							@(4/8/12+ERs)	→	@ERd												
	MOV.L @(d:2,ERs),@ERd+	L	2	4	4				S	D							@(4/8/12+ERs)	→	@ERd			ERd32+4→ERd32									
	MOV.L @(d:2,ERs),@ERd-	L	2	4	4				S	D							@(4/8/12+ERs)	→	@ERd			ERd32-4→ERd32									
	MOV.L @(d:2,ERs),@+ERd	L	2	4	4				S	D				ERd32+4→ERd32			@(4/8/12+ERs)	→	@ERd												
	MOV.L @(d:2,ERs),@-ERd	L	2	4	4				S	D				ERd32-4→ERd32			@(4/8/12+ERs)	→	@ERd												
	MOV.L @(d:2,ERs),@(d:2,ERd)	L	2	4	4				S	D							@(4/8/12+ERs)	→	@(4/8/12+ERd)												
	MOV.L @(d:2,ERs),@(d:16,ERd)	L	3	4	5				S	D							@(4/8/12+ERs)	→	@(d:16+ERd)												
	MOV.L @(d:2,ERs),@(d:32,ERd)	L	4	4	6				S	D							@(4/8/12+ERs)	→	@(d:32+ERd)												
	MOV.L @(d:2,ERs),@(d:16,Rd.B)	L	3	4	5				S	D							@(4/8/12+ERs)	→	@(d:16+RdL<<2)												
	MOV.L @(d:2,ERs),@(d:16,Rd.W)	L	3	4	5				S	D							@(4/8/12+ERs)	→	@(d:16+Pd<<2)												
	MOV.L @(d:2,ERs),@(d:16,ERd.L)	L	3	4	5				S	D							@(4/8/12+ERs)	→	@(d:16+ERd<<2)												
	MOV.L @(d:2,ERs),@(d:32,Rd.B)	L	4	4	6				S	D							@(4/8/12+ERs)	→	@(d:32+RdL<<2)												
	MOV.L @(d:2,ERs),@(d:32,Rd.W)	L	4	4	6				S	D							@(4/8/12+ERs)	→	@(d:32+Pd<<2)												
	MOV.L @(d:2,ERs),@(d:32,ERd.L)	L	4	4	6				S	D							@(4/8/12+ERs)	→	@(d:32+ERd<<2)												
	MOV.L @(d:2,ERs),@aa:16	L	3	4	5				S	D							@(4/8/12+ERs)	→	@aa:16												
	MOV.L @(d:2,ERs),@aa:32	L	4	4	6				S	D							@(4/8/12+ERs)	→	@aa:32												

Data transfer (36)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode											Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	Condition Codes ^{s2}							
				Min.	16-Bit Instruction Fetch Execution Status ¹	Rn	@ERn	@d.ERn	@d.Rn.L/Rn.W/ERn.L	@ERn/@ERn+/@ERn+/@+ERn	@aa:9/@aa:16/@aa:32	Operation ^{s5}								I	H	N	Z	V	C		
				Max.	xxx							Operation 1	Operation 2	Operation 3												Operation 4	Operation 5
MOV	MOV.L @(d:16,Rs,W),@ERd-	L	3	4	5					S	D								ERd32-4→ERd32	—	—	↑	↓	0	—		
	MOV.L @(d:16,Rs,W),@+ERd	L	3	4	5					S	D		ERd32+4→ERd32							—	—	↑	↓	0	—		
	MOV.L @(d:16,Rs,W),@-ERd	L	3	4	5					S	D		ERd32-4→ERd32							—	—	↑	↓	0	—		
	MOV.L @(d:16,Rs,W),@(d:2,ERd)	L	3	4	5					D	S									—	—	↑	↓	0	—		
	MOV.L @(d:16,Rs,W),@(d:16,ERd)	L	4	4	6					D	S									—	—	↑	↓	0	—		
	MOV.L @(d:16,Rs,W),@(d:32,ERd)	L	5	4	7					D	S									—	—	↑	↓	0	—		
	MOV.L @(d:16,Rs,W),@(d:16,Rd,B)	L	4	4	6						S									—	—	↑	↓	0	—		
	MOV.L @(d:16,Rs,W),@(d:16,Rd,W)	L	4	4	6						S									—	—	↑	↓	0	—		
	MOV.L @(d:16,Rs,W),@(d:16,ERd,L)	L	4	4	6						S									—	—	↑	↓	0	—		
	MOV.L @(d:16,Rs,W),@(d:32,Rd,B)	L	5	4	7						S									—	—	↑	↓	0	—		
	MOV.L @(d:16,Rs,W),@(d:32,Rd,W)	L	5	4	7						S									—	—	↑	↓	0	—		
	MOV.L @(d:16,Rs,W),@(d:32,ERd,L)	L	5	4	7						S									—	—	↑	↓	0	—		
	MOV.L @(d:16,Rs,W),@aa:16	L	4	4	6						S	D								—	—	↑	↓	0	—		
	MOV.L @(d:16,Rs,W),@aa:32	L	5	4	7						S	D								—	—	↑	↓	0	—		
	MOV.L @(d:16,ERs,L),@ERd	L	3	4	5					D	S									—	—	↑	↓	0	—		
	MOV.L @(d:16,ERs,L),@ERd+	L	3	4	5						S	D								ERd32+4→ERd32	—	—	↑	↓	0	—	
	MOV.L @(d:16,ERs,L),@ERd-	L	3	4	5						S	D								ERd32-4→ERd32	—	—	↑	↓	0	—	
	MOV.L @(d:16,ERs,L),@+ERd	L	3	4	5						S	D	ERd32+4→ERd32							—	—	↑	↓	0	—		
	MOV.L @(d:16,ERs,L),@-ERd	L	3	4	5						S	D	ERd32-4→ERd32							—	—	↑	↓	0	—		
	MOV.L @(d:16,ERs,L),@(d:2,ERd)	L	3	4	5					D	S									—	—	↑	↓	0	—		
	MOV.L @(d:16,ERs,L),@(d:16,ERd)	L	4	4	6					D	S									—	—	↑	↓	0	—		
	MOV.L @(d:16,ERs,L),@(d:32,ERd)	L	5	4	7					D	S									—	—	↑	↓	0	—		
	MOV.L @(d:16,ERs,L),@(d:16,Rd,B)	L	4	4	6						S									—	—	↑	↓	0	—		
	MOV.L @(d:16,ERs,L),@(d:16,Rd,W)	L	4	4	6						S									—	—	↑	↓	0	—		
	MOV.L @(d:16,ERs,L),@(d:16,ERd,L)	L	4	4	6						S									—	—	↑	↓	0	—		

Data transfer (38)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode										Operation ⁵					Condition Codes ^{6,7}						
				Min.	16-Bit Instruction Fetch Execution Status ¹	2-Byte Instruction Fetch Execution Status ¹	4-Byte Instruction Fetch Execution Status ¹	Rn	@ERn	@(d,ERn)	@(d,Rn),#ERn,W,ERn,L	@ERn/@ERn+/@ERn-/@ERn+/@+ERn	@aa:0/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C	
MOV	MOV.L @(d:32,Rs,W),@(d:32,Rd,W)	L	6 4 8																						
	MOV.L @(d:32,Rs,W),@(d:32,ERd,L)	L	6 4 8																						
	MOV.L @(d:32,Rs,W),@aa:16	L	5 4 7																						
	MOV.L @aa:16,@(d:32,ERd)	L	5 4 8																						
	MOV.L @aa:16,@(d:16,Rd,B)	L	4 4 7																						
	MOV.L @aa:16,@(d:16,Rd,W)	L	4 4 7																						
	MOV.L @aa:16,@(d:16,ERd,L)	L	4 4 7																						
	MOV.L @aa:16,@(d:32,Rd,B)	L	5 4 8																						
	MOV.L @aa:16,@(d:32,Rd,W)	L	5 4 8																						
	MOV.L @aa:16,@(d:32,ERd,L)	L	5 4 8																						
	MOV.L @aa:16,@aa:16	L	4 4 7																						
	MOV.L @aa:16,@aa:32	L	5 4 8																						
	MOV.L @aa:32,@ERd	L	4 4 7																						
	MOV.L @aa:32,@ERd+	L	4 4 7																						
	MOV.L @aa:32,@ERd-	L	4 4 7																						
	MOV.L @aa:32,@+ERd	L	4 4 7																						
	MOV.L @aa:32,@-ERd	L	4 4 7																						
	MOV.L @aa:32,@(d:2,ERd)	L	4 4 7																						
	MOV.L @aa:32,@(d:16,ERd)	L	5 4 8																						
	MOV.L @aa:32,@(d:32,ERd)	L	6 4 9																						
	MOV.L @aa:32,@(d:16,Rd,B)	L	5 4 8																						
	MOV.L @aa:32,@(d:16,Rd,W)	L	5 4 8																						
	MOV.L @aa:32,@(d:16,ERd,L)	L	5 4 8																						
	MOV.L @aa:32,@(d:32,Rd,B)	L	6 4 9																						

Data transfer (45)

Instruction	Mnemonic	Size	Instruction Length	Min	Max	16-Bit Instruction Fetch Execution Status ¹	Addressing Mode									Operation ⁵					Condition Codes ³						
							Prx	Rn	@ERN	@(d,ERN)	@(d,Rn,LRn,WRn,LL)	@(d,ERn/ERN+/@ERN+/@ERN+/@+ERn)	@aa:R/@aa:16//@aa:32	Operation 1	Operation 2	Operation 3			Operation 4	Operation 5	I	H	N	Z	V	C	
MOVA	MOVA,WL @(d:16,@(d:16,ERs),B),ERd	B	4	4	5		D		S								d:16 + (@(d:16+ERd))8<<1 → ERd										
	MOVA,WL @(d:16,@(d:32,ERs),B),ERd	B	5	4	6		D		S								d:16 + (@(d:32+ERs))8<<1 → ERd										
	MOVA,WL @(d:16,@(d:16,Rs,B),B),ERd	B	4	4	5		D		S								d:16 + (@(d:16+RsL))8<<1 → ERd										
	MOVA,WL @(d:16,@(d:16,Rs,W),B),ERd	B	4	4	5		D		S								d:16 + (@(d:16+Rs))8<<1 → ERd										
	MOVA,WL @(d:16,@(d:16,ERs,L),B),ERd	B	4	4	5		D		S								d:16 + (@(d:16+ERd))8<<1 → ERd										
	MOVA,WL @(d:16,@(d:32,Rs,B),B),ERd	B	5	4	6		D		S								d:16 + (@(d:32+RsL))8<<1 → ERd										
	MOVA,WL @(d:16,@(d:32,Rs,W),B),ERd	B	5	4	6		D		S								d:16 + (@(d:32+Rs))8<<1 → ERd										
	MOVA,WL @(d:16,@(d:32,ERs,L),B),ERd	B	5	4	6		D		S								d:16 + (@(d:32+ERd))8<<1 → ERd										
	MOVA,WL @ (d:16,@aa:16,B),ERd	B	4	3	5		D				S						d:16 + (@aa:16)8<<1 → ERd										
	MOVA,WL @ (d:16,@aa:32,B),ERd	B	5	3	6		D				S						d:16 + (@aa:32)8<<1 → ERd										
	MOVA,WL @ (d:16,@ERs,W),ERd	W	3	3	4		D	S									d:16 + (@ERs)16<<1 → ERd										
	MOVA,WL @ (d:16,@ERs+,W),ERd	W	3	3	4		D			S							d:16 + (@ERs)16<<1 → ERd	ERs32+2→ERs32									
MOVA,WL @ (d:16,@ERs-,W),ERd	W	3	3	4		D			S							d:16 + (@ERs)16<<1 → ERd	ERs32-2→ERs32										
MOVA,WL @ (d:16,@+ERs,W),ERd	W	3	4	4		D			S							ERs32+2→ERs32	d:16 + (@ERs)16<<1 → ERd										

Data transfer (46)

Instruction	Mnemonic	Size	Instruction Length	Number of			Addressing Mode									Operation ⁵					Condition Code ^{s2}													
				Min.	Max.	16-Bit Instruction Fetch Execution Status ¹	Immediate	Register	Register with Displacement	Register with Displacement and Immediate	Register with Displacement and Register	Register with Displacement and Register Indirect	Register with Displacement and Register Indirect with Scaling	Register with Displacement and Register Indirect with Scaling and Register	Register with Displacement and Register Indirect with Scaling and Register Indirect	Register with Displacement and Register Indirect with Scaling and Register Indirect with Register	Register with Displacement and Register Indirect with Scaling and Register Indirect with Register and Register	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C						
MOVA	MOVA/W,L @(d:16,@ERs.W),ERd	W	3	4	4	D												ERs32-2→ERs32		d:16 + (@ERs)16<<1 → ERd														
	MOVA/W,L @(d:16,@(d:2,ERs.W)),ERd	W	3	4	4	D	S													d:16 + (@(2/4/6+ERs))16 <<1 → ERd														
	MOVA/W,L @(d:16,@(d:16,ERs.W)),ERd	W	4	4	5	D	S													d:16 + (@(d:16+ERd))16 <<1 → ERd														
	MOVA/W,L @(d:16,@(d:32,ERs.W)),ERd	W	5	4	6	D	S													d:16 + (@(d:32+ERd))16 <<1 → ERd														
	MOVA/W,L @(d:16,@(d:16,Rs.B.W)),ERd	W	4	4	5	D	S													d:16 + (@(d:16+Rs<<1))16 <<1 → ERd														
	MOVA/W,L @(d:16,@(d:16,Rs.W.W)),ERd	W	4	4	5	D	S													d:16 + (@(d:16+Rs<<1))16 <<1 → ERd														
	MOVA/W,L @(d:16,@(d:16,ERs.L.W)),ERd	W	4	4	5	D	S													d:16 + (@(d:16+ERd<<1)) → ERd														
	MOVA/W,L @(d:16,@(d:32,Rs.B.W)),ERd	W	5	4	6	D	S													d:16 + (@(d:32+Rs<<1))16 <<1 → ERd														
	MOVA/W,L @(d:16,@(d:32,Rs.W.W)),ERd	W	5	4	6	D	S													d:16 + (@(d:32+Rs<<1))16 <<1 → ERd														

Data transfer (48)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode							Operation ^s					Condition Codes ^{s2}																	
				Min.	Max.	16-Bit Instruction Fetch Execution Status ^{s1}	Rn	@ERn	@(d, ERn)	@(d, Rn, ERn, W, ERn, L)	@ERn/@ERn+/@ERn-/@ERn+/@ERn-/@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C											
MOVA	MOVA/LL @(d:16,@(d:32,Rs.B),B),ERd	B	5	4	6		D		S																								
MOVA	MOVA/LL @(d:16,@(d:32,Rs.W),B),ERd	B	5	4	6		D		S																								
MOVA	MOVA/LL @(d:16,@(d:32,ERs.L),B),ERd	B	5	4	6		D		S																								
MOVA	MOVA/LL @(d:16,@aa:16.B),ERd	B	4	3	5		D			S																							
MOVA	MOVA/LL @(d:16,@aa:32.B),ERd	B	5	3	6		D			S																							
MOVA	MOVA/LL @(d:16,@ERs.W),ERd	W	3	3	4		D	S																									
MOVA	MOVA/LL @(d:16,@ERs+.W),ERd	W	3	3	4		D		S																								
MOVA	MOVA/LL @(d:16,@ERs-.W),ERd	W	3	3	4		D		S																								
MOVA	MOVA/LL @(d:16,@+ERs.W),ERd	W	3	4	4		D		S					ERs32+2→ERs32																			
MOVA	MOVA/LL @(d:16,@-ERs.W),ERd	W	3	4	4		D		S					ERs32-2→ERs32																			
MOVA	MOVA/LL @(d:16,@(d2,ERs),W),ERd	W	3	4	4		D	S																									
MOVA	MOVA/LL @(d:16,@(d16,ERs),W),ERd	W	4	4	5		D	S																									
MOVA	MOVA/LL @(d:16,@(d32,ERs),W),ERd	W	5	4	6		D	S																									
MOVA	MOVA/LL @(d:16,@(d16,Rs.B),W),ERd	W	4	4	5		D		S																								

Data transfer (49)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode						Operation ⁵					Condition Codes ^{6,7}								
				Min.	Max.	16-Bit Instruction Fetch Execution Status ¹	Prx	Rn	@(d,ERn)	@(d,Rn,LRn,WRn,L)	@(ERn/ERn+/@ERn+/@ERn/@+ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C
MOVA	MOVA/LL @(d:16,@(d:16,Rs.W),W),ERd	W	4	4	5		D		S			d:16 + (@(d:16+Rs<<1))16 → ERd <<2											
	MOVA/LL @(d:16,@(d:16,ERs.L),W),ERd	W	4	4	5		D		S			d:16 + (@(d:16+ERd<<1))16 → ERd <<2											
	MOVA/LL @(d:16,@(d:32,Rs.B),W),ERd	W	5	4	6		D		S			d:16 + (@(d:32+RsL<<1))16 → ERd <<2											
	MOVA/LL @(d:16,@(d:32,Rs.W),W),ERd	W	5	4	6		D		S			d:16 + (@(d:32+Rs<<1))16 → ERd <											
	MOVA/LL @(d:16,@(d:32,ERs.L),W),ERd	W	5	4	6		D		S			d:16 + (@(d:32+ERd<<1))16 → ERd <<2											
	MOVA/LL @(d:16,@aa:16,W),ERd	W	4	3	5		D			S			d:16 + (@aa:16)16<<2 → ERd										
	MOVA/LL @(d:16,@aa:32,W),ERd	W	5	3	6		D			S			d:16 + (@aa:32)16<<2 → ERd										
	MOVA/B.L @(d:32,@ERs.B),ERd	B	4	3	5		D	S					d:32 + (@ERs)8 → ERd										
	MOVA/B.L @(d:32,@ERs+.B),ERd	B	4	3	5		D			S			d:32 + (@ERs)8 → ERd	ERs32+1→ERs32									
	MOVA/B.L @(d:32,@ERs-.B),ERd	B	4	3	5		D			S			d:32 + (@ERs)8 → ERd	ERs32-1→ERs32									
	MOVA/B.L @(d:32,@+ERs.B),ERd	B	4	4	5		D			S		ERs32+1→ERs32	d:32 + (@ERs)8 → ERd										
	MOVA/B.L @(d:32,@-ERs.B),ERd	B	4	4	5		D			S		ERs32-1→ERs32	d:32 + (@ERs)8 → ERd										
MOVA/B.L @(d:32,@(d:2,ERs.B),ERd	B	4	4	5		D		S				d:32 + (@((1/2/3+ERs)))8 → ERd											

Data transfer (50)

Instruction n	Mnemonic	Size	Instruction Length	Addressing Mode								Operation ⁵					Condition Codes ²					
				Min.	16-Bit Instruction Fetch Execution Status ⁴ Fxx	Rn	@ERN	@ (d,ERN)	@ (d,Rn,WRn,L)	@ ERn/ERn+/ERn-/ERn+/@+ERn	@ aa:R/ @ aa:16/ @ aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C
MOVA	MOVA/B.L @(d:32,@(d:16,ERs),B),ERd	B	5	4	6	D		S				d:32 + (@(d:16+ERd))8 → ERd			—	—	—	—	—	—	—	
	MOVA/B.L @(d:32,@(d:32,ERs),B),ERd	B	6	4	7	D		S				d:32 + (@(d:32+ERs))8 → ERd			—	—	—	—	—	—	—	
	MOVA/B.L @(d:32,@(d:16,Rs,B),ERd	B	5	4	6	D		S				d:32 + (@(d:16+RsL))8 → ERd			—	—	—	—	—	—	—	
	MOVA/B.L @(d:32,@(d:16,Rs,W),B),ERd	B	5	4	6	D		S				d:32 + (@(d:16+Rs))8 → ERd			—	—	—	—	—	—	—	
	MOVA/B.L @(d:32,@(d:16,ERs,L),B),ERd	B	5	4	6	D		S				d:32 + (@(d:16+ERd))8 → ERd			—	—	—	—	—	—	—	
	MOVA/B.L	B	6	4	7	D		S				d:32 + (@(d:32+RsL))8 → ERd			—	—	—	—	—	—	—	
	MOVA/B.L @(d:32,@(d:32,Rs,W),B),ERd	B	6	4	7	D		S				d:32 + (@(d:32+Rs))8 → ERd			—	—	—	—	—	—	—	
	MOVA/B.L @(d:32,@(d:32,ERs,L),B),ERd	B	6	4	7	D		S				d:32 + (@(d:32+ERd))8 → ERd			—	—	—	—	—	—	—	
	MOVA/B.L @(d:32,@aa:16,B),ERd	B	5	3	6	D			S			d:32 + (@aa:16)8 → ERd			—	—	—	—	—	—	—	
	MOVA/B.L @(d:32,@aa:32,B),ERd	B	6	3	7	D			S			d:32 + (@aa:32)8 → ERd			—	—	—	—	—	—	—	
	MOVA/B.L @(d:32,@ERs,W),ERd	W	4	3	5	D	S					d:32 + (@ERs)16 → ERd			—	—	—	—	—	—	—	
	MOVA/B.L @(d:32,@ERs+,W),ERd	W	4	3	5	D			S			d:32 + (@ERs)16 → ERd	ERs32+2→ERs32		—	—	—	—	—	—	—	
	MOVA/B.L @(d:32,@ERs-,W),ERd	W	4	3	5	D			S			d:32 + (@ERs)16 → ERd	ERs32-2→ERs32		—	—	—	—	—	—	—	
	MOVA/B.L @(d:32,@+ERs,W),ERd	W	4	4	5	D			S		ERs32+2→ERs32	d:32 + (@ERs)16 → ERd			—	—	—	—	—	—	—	
	MOVA/B.L @(d:32,@-ERs,W),ERd	W	4	4	5	D			S		ERs32-2→ERs32	d:32 + (@ERs)16 → ERd			—	—	—	—	—	—	—	
	MOVA/B.L @(d:32,@(d:2,ERs),W),ERd	W	4	4	5	D		S				d:32 + (@(2/4/6+ERs))1 → ERd			—	—	—	—	—	—	—	
	MOVA/B.L @(d:32,@(d:16,ERs),W),ERd	W	5	4	6	D		S				d:32 + (@(d:16+ERd))16 → ERd			—	—	—	—	—	—	—	
	MOVA/B.L @(d:32,@(d:32,ERs),W),ERd	W	6	4	7	D		S				d:32 + (@(d:32+ERd))16 → ERd			—	—	—	—	—	—	—	

Data transfer (51)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode										Operation ^s					Condition Codes ^{s2}													
				Number of Execution Stages ¹		Rn	@ERn	@ERn	@ERn/@ERn+/@ERn-/@ERn-@ERn	@ERn/@ERn+/@ERn-/@ERn-@ERn	@aa:8/@aa:16/@aa:32	Rn	@ERn												@ERn/@ERn+/@ERn-/@ERn-@ERn	@aa:8/@aa:16/@aa:32						
				Min.	Max.									16-Bit Instruction Fetch	16-Bit Instruction Fetch																	
n				Min.	Max.	16-Bit Instruction Fetch	16-Bit Instruction Fetch	pxx	Rn	@ERn	@ERn	@ERn/@ERn+/@ERn-/@ERn-@ERn	@ERn/@ERn+/@ERn-/@ERn-@ERn	@aa:8/@aa:16/@aa:32	Rn	@ERn	@ERn/@ERn+/@ERn-/@ERn-@ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C			
MOVA	MOVA/B.L @(d:32,@(d:16,Rs.B),W),ERd	W	5	4	6				D		S											d:32 + (@(d:16+RsL<<1))1 → ERd 6										
	MOVA/B.L @(d:32,@(d:16,Rs.W),W),ERd	W	5	4	6				D		S											d:32 + (@(d:16+Rs<<1))16 → ERd										
	MOVA/B.L @(d:32,@(d:16,ERs.L),W),ERd	W	5	4	6				D		S											d:32 + (@(d:16+ERd<<1)) → ERd 16										
	MOVA/B.L @(d:32,@(d:32,Rs.B),W),ERd	W	6	4	7				D		S											d:32 + (@(d:32+RsL<<1))1 → ERd 6										
	MOVA/B.L @(d:32,@(d:32,Rs.W),W),ERd	W	6	4	7				D		S											d:32 + (@(d:32+Rs<<1))16 → ERd										
	MOVA/B.L @(d:32,@(d:32,ERs.L),W),ERd	W	6	4	7				D		S											d:32 + (@(d:32+ERd<<1)) → ERd 16										
	MOVA/B.L @ (d:32,@aa:16,W),ERd	W	5	3	6				D					S								d:32 + (@aa:16)16 → ERd										
	MOVA/B.L @ (d:32,@aa:32,W),ERd	W	6	3	7				D					S								d:32 + (@aa:32)16 → ERd										
	MOVA/W.L @ (d:32,@ERs.B),ERd	B	4	3	5				D	S												d:32 + (@ERs)8<<1 → ERd										
	MOVA/W.L @ (d:32,@ERs+.B),ERd	B	4	3	5				D		S											d:32 + (@ERs)8<<1 → ERd	ERs32+1→ERs32									
	MOVA/W.L @ (d:32,@ERs-.B),ERd	B	4	3	5				D		S											d:32 + (@ERs)8<<1 → ERd	ERs32-1→ERs32									
	MOVA/W.L @ (d:32,@+EPs.B),ERd	B	4	4	5				D		S									ERs32+1→ERs32		d:32 + (@ERs)8<<1 → ERd										
	MOVA/W.L @ (d:32,@-ERs.B),ERd	B	4	4	5				D		S									ERs32-1→ERs32		d:32 + (@ERs)8<<1 → ERd										
	MOVA/W.L @(d:32,@(d:2,ERs.B),ERd	B	4	4	5				D	S												d:32 + (@(1/2/3+ERs))8 → ERd <<1										

Data transfer (52)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	Addressing Mode							Operation ⁵					Condition Code ^{6,7}									
						Number of Execution Stages ¹	16-Bit Instruction Fetch	Fetch	Execution	Stage ²	Rn	@ERN	@(d,ERN)	@(d,Rn,ERN,W,ERN,L)	@ERN/@ERN+/@ERN-/@+ERN	@aa:R/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C
MOVA	MOVA/WL @(d:32,@(d:16,ERs),B),ERd	B	5	4	6		D		S							d:32 + (@(d:16+ERd))8 → ERd <<1											
MOVA	MOVA/WL @(d:32,@(d:32,ERs),B),ERd	B	6	4	7		D		S							d:32 + (@(d:32+ERd))8 → ERd <<1											
MOVA	MOVA/WL @(d:32,@(d:16,Rs,B),B),ERd	B	5	4	6		D		S							d:32 + (@(d:16+RsL))8 → ERd <<1											
MOVA	MOVA/WL @(d:32,@(d:16,Rs,W),B),ERd	B	5	4	6		D		S							d:32 + (@(d:16+Rs))8 → ERd <<1											
MOVA	MOVA/WL @(d:32,@(d:16,ERs,L),B),ERd	B	5	4	6		D		S							d:32 + (@(d:16+ERd))8 → ERd <<1											
MOVA	MOVA/WL @(d:32,@(d:32,Rs,B),B),ERd	B	6	4	7		D		S							d:32 + (@(d:32+RsL))8 → ERd <<1											
MOVA	MOVA/WL @(d:32,@(d:32,Rs,W),B),ERd	B	6	4	7		D		S							d:32 + (@(d:32+Rs))8 → ERd <<1											
MOVA	MOVA/WL @(d:32,@(d:32,ERs,L),B),ERd	B	6	4	7		D		S							d:32 + (@(d:32+ERd))8 → ERd <<1											
MOVA	MOVA/WL @ (d:32,@aa:16,B),ERd	B	5	3	6		D				S					d:32 + (@aa:16)8<<1 → ERd											
MOVA	MOVA/WL @ (d:32,@aa:32,B),ERd	B	6	3	7		D				S					d:32 + (@aa:32)8<<1 → ERd											
MOVA	MOVA/WL @ (d:32,@ERs,W),ERd	W	4	3	5		D	S								d:32 + (@ERs)16<<1 → ERd											
MOVA	MOVA/WL @ (d:32,@ERs+,W),ERd	W	4	3	5		D				S					d:32 + (@ERs)16<<1 → ERd	ERs32+2→ERs32										
MOVA	MOVA/WL @ (d:32,@ERs-,W),ERd	W	4	3	5		D				S					d:32 + (@ERs)16<<1 → ERd	ERs32-2→ERs32										
MOVA	MOVA/WL @ (d:32,@+ERs,W),ERd	W	4	4	5		D				S					ERs32+2→ERs32 d:32 + (@ERs)16<<1 → ERd											



Data transfer (53)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	16-Bit Instruction Fetch Execution Status ¹	Addressing Mode					Operation ⁵					Condition Codes ³															
							Rn	@(d,ERn)	@(d,Rn.L,Rn.W,ERn.L)	@(d,ERn)/@ERn+/@ERn-/@+ERn	@aa:R/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C										
MOVA	MOVA.W.L @(d:32,@-ERs.W),ERd	W	4	4	5		D			S			ERs32-2 → ERs32			d:32 + (@ERs)16<<1 → ERd																
	MOVA.W.L @(d:32,@(d:2,ERs),W),ERd	W	4	4	5		D		S							d:32 + (@(2/4/6+ERs))16 <<1 → ERd																
	MOVA.W.L @(d:32,@(d:16,ERs),W),ERd	W	5	4	6		D		S							d:32 + (@(d:16+ERd))16 <<1 → ERd																
	MOVA.W.L @(d:32,@(d:32,ERs),W),ERd	W	6	4	7		D		S							d:32 + (@(d:32+ERd))16 <<1 → ERd																
	MOVA.W.L @(d:32,@(d:16,Rs.B),W),ERd	W	5	4	6		D		S							d:32 + (@(d:16+RsL<<1))16 <<1 → ERd																
	MOVA.W.L @(d:32,@(d:16,Rs.W),ERd	W	5	4	6		D		S							d:32 + (@(d:16+Rs<<1))16 <<1 → ERd																
	MOVA.W.L @(d:32,@(d:16,ERs.L),W),ERd	W	5	4	6		D		S							d:32 + (@(d:16+ERd<<1))16 <<1 → ERd																
	MOVA.W.L @(d:32,@(d:32,Rs.B),W),ERd	W	6	4	7		D		S							d:32 + (@(d:32+RsL<<1))16 <<1 → ERd																
	MOVA.W.L @(d:32,@(d:32,Rs.W),W),ERd	W	6	4	7		D		S							d:32 + (@(d:32+Rs<<1))16 <<1 → ERd																

Data transfer (54)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode								Operation ⁵					Condition Codes ²								
				Number of		Execution Status ¹		Rn	@ERn	@ (d, ERn)	@ (d, Rn, ERn, W, ERn, L)	@ ERn / @ ERn+ / @ ERn+ / @ ERn	@ aa:R / @ aa:16 / @ aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C	
				Min.	Max.	16-Bit Instruction Fetch	Execution Status ¹																		
MOVA	MOVA/WL @ (d:32, @ (d:32, ERs, L), W), ERd	W	6	4	7	D		S						d:32 + (@ (d:32+ERd<<1)) → ERd 16 <<1											
	MOVA/WL @ (d:32, @ aa:16, W), ERd	W	5	3	6	D				S				d:32 + (@ aa:16)16<<1 → ERd											
	MOVA/WL @ (d:32, @ aa:32, W), ERd	W	6	3	7	D				S				d:32 + (@ aa:32)16<<1 → ERd											
	MOVA/LL @ (d:32, @ ERs, B), ERd	B	4	3	5	D	S							d:32 + (@ ERs)8<<2 → ERd											
	MOVA/LL @ (d:32, @ ERs+ B), ERd	B	4	3	5	D				S				d:32 + (@ ERs)8<<2 → ERd	ERs32+1 → ERs32										
	MOVA/LL @ (d:32, @ ERs- B), ERd	B	4	3	5	D				S				d:32 + (@ ERs)8<<2 → ERd	ERs32-1 → ERs32										
	MOVA/LL @ (d:32, @ +ERs, B), ERd	B	4	4	5	D				S		ERs32+1 → ERs32		d:32 + (@ ERs)8<<2 → ERd											
	MOVA/LL @ (d:32, @ -ERs, B), ERd	B	4	4	5	D				S		ERs32-1 → ERs32		d:32 + (@ ERs)8<<2 → ERd											
	MOVA/LL @ (d:32, @ (d:2, ERs), B), ERd	B	4	4	5	D	S							d:32 + (@ (1/2/3+ERs))8 → ERd <<2											
	MOVA/LL @ (d:32, @ (d:16, ERs), B), ERd	B	5	4	6	D	S							d:32 + (@ (d:16+ERd))8 → ERd <<2											
	MOVA/LL @ (d:32, @ (d:32, ERs), B), ERd	B	6	4	7	D	S							d:32 + (@ (d:32+ERd))8 → ERd <<2											
	MOVA/LL @ (d:32, @ (d:16, Rs, B), B), ERd	B	5	4	6	D		S						d:32 + (@ (d:16+Rs))8 → ERd <<2											
	MOVA/LL @ (d:32, @ (d:16, Rs, W), B), ERd	B	5	4	6	D		S						d:32 + (@ (d:16+Rs))8 → ERd <<2											
	MOVA/LL @ (d:32, @ (d:16, ERs, L), B), ERd	B	5	4	6	D		S						d:32 + (@ (d:16+ERd))8 → ERd <<2											



Data transfer (55)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	Number of Execution Stages ¹	Addressing Mode						Operation ⁵					Condition Code ²																			
							16-Bit Instruction Fetch	Fetch	Rn	@(d,ERn)	@(d,Rn,ERn,W,ERn,L)	@(ERn)/@ERn+/@ERn-/@ERn+/@+ERn	@aa:R/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C													
MOVA	MOVA/LL @(d:32,@(d:32,Rs.B),B),ERd	B	6	4	7		D		S						d:32 + ((d:32+RsL)8 <<2 → ERd																						
	MOVA/LL @(d:32,@(d:32,Rs.W),B),ERd	B	6	4	7		D		S						d:32 + ((d:32+Rs)8 <<2 → ERd																						
	MOVA/LL @(d:32,@(d:32,ERs.L),B),ERd	B	6	4	7		D		S						d:32 + ((d:32+ERd)8 <<2 → ERd																						
	MOVA/LL @(d:32,@aa:16.B),ERd	B	5	3	6		D				S				d:32 + (@aa:16)8<<2 → ERd																						
	MOVA/LL @(d:32,@aa:32.B),ERd	B	6	3	7		D				S				d:32 + (@aa:32)8<<2 → ERd																						
	MOVA/LL @(d:32,@ERs.W),ERd	W	4	3	5		D	S							d:32 + (@ERs)16<<2 → ERd																						
	MOVA/LL @(d:32,@ERs+.W),ERd	W	4	3	5		D		S						d:32 + (@ERs)16<<2 → ERd	ERs32+2→ERs32																					
	MOVA/LL @(d:32,@ERs-.W),ERd	W	4	3	5		D		S						d:32 + (@ERs)16<<2 → ERd	ERs32-2→ERs32																					
	MOVA/LL @(d:32,@+ERs.W),ERd	W	4	4	5		D		S					ERs32+2→ERs32	d:32 + (@ERs)16<<2 → ERd																						
	MOVA/LL @(d:32,@-ERs.W),ERd	W	4	4	5		D		S					ERs32-2→ERs32	d:32 + (@ERs)16<<2 → ERd																						
	MOVA/LL @(d:32,@(d2,ERs),W),ERd	W	4	4	5		D	S							d:32 + (@(2/4/6+ERs)16 <<2 → ERd																						
	MOVA/LL @(d:32,@(d:16,ERs),W),ERd	W	5	4	6		D	S							d:32 + (@(d:16+ERd)16 <<2 → ERd																						
	MOVA/LL @(d:32,@(d:32,ERs),W),ERd	W	6	4	7		D	S							d:32 + (@(d:32+ERd)16 <<2 → ERd																						
	MOVA/LL @(d:32,@(d:16,Rs.B),W),ERd	W	5	4	6		D		S						d:32 + (@(d:16+RsL<<1)16 <<2 → ERd																						

Data transfer (56)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	16-Bit Instruction Fetch Execution Status ¹	Addressing Mode							Operation ⁵					Condition Code ^{6,2}									
							ppx	Rn	@(d,ERn)	@(d,Rn,ERn,W,ERn,L)	@(d,ERn)/@ERn+/@ERn+/@+ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C					
MOVA	MOVA/LL @(d:32,@(d:16,Rs,W),W),ERd	W	5	4	6		D		S																			
	MOVA/LL @(d:32,@(d:16,ERs,L),W),ERd	W	5	4	6		D		S																			
	MOVA/LL @(d:32,@(d:32,Rs,B),W),ERd	W	6	4	7		D		S																			
	MOVA/LL @(d:32,@(d:32,Rs,W),W),ERd	W	6	4	7		D		S																			
	MOVA/LL @(d:32,@(d:32,ERs,L),W),ERd	W	6	4	7		D		S																			
	MOVA/LL @ (d:32,@aa:16,W),ERd	W	5	3	6		D			S																		
	MOVA/LL @ (d:32,@aa:32,W),ERd	W	6	3	7		D			S																		

Block transfer (1)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode								Operation ^{s5}					Condition Codes ^{s2}							
				Min.	Max.	16-Bit Instruction Fetch Execution Status ¹	pxx	Rn	@Ern	@(d,Ern)	@(d,Rn,Ern,WEm,L)	@Ern@Ern+@Ern-@+Ern	@aa:R@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C
EEPMOV	EEPMOV.B	B	2	4	4							S	if R4L→0	Repeat	@ER5+	→ @ER6+								
				+	+	2n	2n					D				(Byte-size transfer)	→ R4L							
EEPMOV	EEPMOV.W	B	2	4	4							S	if R4→0	Repeat	@ER5+	→ @ER6+								
				+	+	2n	2n					D				(Byte-size transfer)	→ R4							
MOVMD	MOVMD.B	B	1	1	1							S		Repeat	@ER5+	→ @ER6+								
				+	+	2n	2n					D				(Byte-size transfer)	→ R4							
MOVMD	MOVMD.W	W	1	1	1							S		Repeat	@ER5+	→ @ER6+								
				+	+	2n	2n					D				(Word-size transfer)	→ R4							

Block transfer (2)

Instruction	Mnemonic	Size	Instruction Length	Number of Execution Stages ¹							Addressing Mode	Operation ⁵					Condition Code ^{6,2}					
				Min.	16-Bit Instruction Fetch	Instruction Decode	Execute	Write Back	Write Back	Write Back		Write Back	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V
MOVMD	MOVMD.L	L	1	1	1	1	1	1	1	1	S D S ³	Repeat Until R4=0	@ER5+ R4	→ @ER6+ (Longword-size transfer) - 1 → R4								
MOVSD	MOVSD.B d:16	B	2	4	4	4	4	4	4	4	S D S ³	Repeat Until R4=0	@ER5+ if @ER6==0 then { ER6+1→ER6 R4-1→R4 goto next+disp } ER6+1→ER6 R4-1→R4	→@ER6300								

Arithmetic operations (1)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch #xx	Execution Status ¹	Addressing Mode							Operation ⁵					Condition Codes ²																							
							Rn	@ERn	@(d,ERn)	@(d,Rn,WRn.L)	@ERn/@ERn+/@ERn-/@+ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C																			
ADD	ADD.B #xx:8,Rd	B	1	1	1	S	D									Rd8 + #xx → Rd8																										
	ADD.B #xx:8,@ERd	B	2	3	4	S	D									@ERd + #xx → @ERd																										
	ADD.B #xx:8,@ERd+	B	3	4	5	S				D						@ERd + #xx → @ERd						ERd32+1→ERd32																				
	ADD.B #xx:8,@ERd-	B	3	4	5	S				D						@ERd + #xx → @ERd						ERd32-1→ERd32																				
	ADD.B #xx:8,@+ERd	B	3	5	5	S				D						ERd32+1→ERd32	@ERd + #xx → @ERd																									
	ADD.B #xx:8,@-ERd	B	3	5	5	S				D						ERd32-1→ERd32	@ERd + #xx → @ERd																									
	ADD.B #xx:8,@(d:2,ERd)	B	3	5	5	S				D						@(1/2/3+ERd) + #xx → @(1/2/3+ERd)																										
	ADD.B #xx:8,@(d:16,ERd)	B	4	5	6	S				D						@(d:16+ERd) + #xx → @(d:16+ERd)																										
	ADD.B #xx:8,@(d:32,ERd)	B	5	5	7	S				D						@(d:32+ERd) + #xx → @(d:32+ERd)																										
	ADD.B #xx:8,@(d:16,Rd,B)	B	4	5	6	S				D						@(d:16+RdL) + #xx → @(d:16+RdL)																										
	ADD.B #xx:8,@(d:16,Rd,W)	B	4	5	6	S				D						@(d:16+Rd) + #xx → @(d:16+Rd)																										
	ADD.B #xx:8,@(d:16,ERd,L)	B	4	5	6	S				D						@(d:16+ERd) + #xx → @(d:16+ERd)																										
	ADD.B #xx:8,@(d:32,Rd,B)	B	5	5	7	S				D						@(d:32+RdL) + #xx → @(d:32+RdL)																										
	ADD.B #xx:8,@(d:32,Rd,W)	B	5	5	7	S				D						@(d:32+Rd) + #xx → @(d:32+Rd)																										
	ADD.B #xx:8,@(d:32,ERd,L)	B	5	5	7	S				D						@(d:32+ERd) + #xx → @(d:32+ERd)																										
	ADD.B #xx:8,@aa:8	B	2	3	4	S				D						@aa:8 + #xx → @aa:8																										
	ADD.B #xx:8,@aa:16	B	3	3	5	S				D						@aa:16 + #xx → @aa:16																										
	ADD.B #xx:8,@aa:32	B	4	3	6	S				D						@aa:32 + #xx → @aa:32																										
	ADD.B Rs,Rd	B	1	1	1	S										Rd8 + Rs8 → Rd8																										
	ADD.B Rs,@ERd	B	2	3	4	S	D									@ERd + Rs8 → @ERd																										
	ADD.B Rs,@ERd+	B	2	4	4	S				D						@ERd + Rs8 → @ERd							ERd32+1→ERd32																			
	ADD.B Rs,@ERd-	B	2	4	4	S				D						@ERd + Rs8 → @ERd							ERd32-1→ERd32																			
	ADD.B Rs,@+ERd	B	2	5	5	S				D						ERd32+1→ERd32	@ERd + Rs8 → @ERd																									
	ADD.B Rs,@-ERd	B	2	5	5	S				D						ERd32-1→ERd32	@ERd + Rs8 → @ERd																									
	ADD.B Rs,@(d:2,ERd)	B	2	5	5	S	D									@(1/2/3+ERd) + Rs8 → @(1/2/3+ERd)																										

Arithmetic operations (2)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode								Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	Condition Code ^{s2}																					
				Number of Execution Stages ¹		Rn	@ERN	@d(ERN)	@d(Rn,ERN,WERN,L)	@ERN/@ERN+/@ERN+/@ERN	@aa:8/@aa:16/@aa:32						Rn	@ERN	@d(ERN)	@d(Rn,ERN,WERN,L)	@ERN/@ERN+/@ERN+/@ERN	@aa:8/@aa:16/@aa:32	I	H	N	Z	V	C										
				16-Bit Instruction Fetch	Execute																																	
ADD	ADD.B Rs,@(d:16,ERd)	B	3	5	5	S		D																														
	ADD.B Rs,@(d:32,ERd)	B	4	5	6	S		D																														
	ADD.B Rs,@(d:16,Rd,B)	B	3	5	5	S		D																														
	ADD.B Rs,@(d:16,Rd,W)	B	3	5	5	S		D																														
	ADD.B Rs,@(d:16,ERd,L)	B	3	5	5	S		D																														
	ADD.B Rs,@(d:32,Rd,B)	B	4	5	6	S		D																														
	ADD.B Rs,@(d:32,Rd,W)	B	4	5	6	S		D																														
	ADD.B Rs,@(d:32,ERd,L)	B	4	5	6	S		D																														
	ADD.B Rs,@aa:8	B	2	3	4	S				D																												
	ADD.B Rs,@aa:16	B	3	3	5	S				D																												
	ADD.B Rs,@aa:32	B	4	3	6	S				D																												
	ADD.B @ERs,Rd	B	2	3	3	D	S																															
	ADD.B @ERs+,Rd	B	2	3	3	D			S																													
	ADD.B @ERs-,Rd	B	2	3	3	D			S																													
	ADD.B @+ERs,Rd	B	2	4	4	D			S				ERs32+1→ERs32																									
	ADD.B @-ERs,Rd	B	2	4	4	D			S				ERs32-1→ERs32																									
	ADD.B @(d:2,ERs),Rd	B	2	4	4	D		S																														
	ADD.B @(d:16,ERs),Rd	B	3	4	4	D		S																														
	ADD.B @(d:32,ERs),Rd	B	4	4	5	D		S																														
	ADD.B @(d:16,Rs,B),Rd	B	3	4	4	D		S																														
	ADD.B @(d:16,Rs,W),Rd	B	3	4	4	D		S																														
	ADD.B @(d:16,ERs,L),Rd	B	3	4	4	D		S																														
	ADD.B @(d:32,Rs,B),Rd	B	4	4	5	D		S																														
	ADD.B @(d:32,Rs,W),Rd	B	4	4	5	D		S																														
	ADD.B @(d:32,ERs,L),Rd	B	4	4	5	D		S																														

Arithmetic operations (4)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹ #px	Addressing Mode							Operation ⁵					Condition Codes ²						
						Rn	@(d,ERn)	@(d,Rn,WRn,L)	@(ERn/ERn+/@ERn/@@ERn	@aa:R/ @aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C			
ADD	ADD.B @ERs+, @(d:16,ERd)	B	4	6	7				D	S				@(d:16+ERd) + @ERs → @(d:16+ERd)	ERs32+1→ERs32			—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs+, @(d:32,ERd)	B	5	6	8				D	S				@(d:32+ERd) + @ERs → @(d:32+ERd)	ERs32+1→ERs32			—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs+, @(d:16,Rd.B)	B	4	6	7				D	S				@(d:16+RdL) + @ERs → @(d:16+RdL)	ERs32+1→ERs32			—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs+, @(d:16,Rd.W)	B	4	6	7				D	S				@(d:16+Rd) + @ERs → @(d:16+Rd)	ERs32+1→ERs32			—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs+, @(d:16,ERd.L)	B	4	6	7				D	S				@(d:16+ERd) + @ERs → @(d:16+ERd)	ERs32+1→ERs32			—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs+, @(d:32,Rd.B)	B	5	6	8				D	S				@(d:32+RdL) + @ERs → @(d:32+RdL)	ERs32+1→ERs32			—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs+, @(d:32,Rd.W)	B	5	6	8				D	S				@(d:32+Rd) + @ERs → @(d:32+Rd)	ERs32+1→ERs32			—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs+, @(d:32,ERd.L)	B	5	6	8				D	S				@(d:32+ERd) + @ERs → @(d:32+ERd)	ERs32+1→ERs32			—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs+, @aa:16	B	4	5	7				S	D				@aa:16 + @ERs → @aa:16	ERs32+1→ERs32			—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs+, @aa:32	B	5	5	8				S	D				@aa:32 + @ERs → @aa:32	ERs32+1→ERs32			—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs-, @ERd	B	3	5	6				D	S				@ERd + @ERs → @ERd	ERs32-1→ERs32			—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs-, @ERd+	B	3	5	6				S					@ERd + @ERs → @ERd	ERs32-1→ERs32	ERd32+1→ERd32		—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs-, @ERd-	B	3	5	6				S					@ERd + @ERs → @ERd	ERs32-1→ERs32	ERd32-1→ERd32		—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs-, @+ERd	B	3	6	7				S					ERd32+1→ERd32	@ERd + @ERs → @ERd	ERs32-1→ERs32		—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs-, @-ERd	B	3	6	7				S					ERd32-1→ERd32	@ERd + @ERs → @ERd	ERs32-1→ERs32		—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs-, @(d:2,ERd)	B	3	6	7				D	S				@(1/2/3+ERd) + @ERs → @(1/2/3+ERd)	ERs32-1→ERs32			—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs-, @(d:16,ERd)	B	4	6	7				D	S				@(d:16+ERd) + @ERs → @(d:16+ERd)	ERs32-1→ERs32			—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs-, @(d:32,ERd)	B	5	6	8				D	S				@(d:32+ERd) + @ERs → @(d:32+ERd)	ERs32-1→ERs32			—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs-, @(d:16,Rd.B)	B	4	6	7				D	S				@(d:16+RdL) + @ERs → @(d:16+RdL)	ERs32-1→ERs32			—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs-, @(d:16,Rd.W)	B	4	6	7				D	S				@(d:16+Rd) + @ERs → @(d:16+Rd)	ERs32-1→ERs32			—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs-, @(d:16,ERd.L)	B	4	6	7				D	S				@(d:16+ERd) + @ERs → @(d:16+ERd)	ERs32-1→ERs32			—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs-, @(d:32,Rd.B)	B	5	6	8				D	S				@(d:32+RdL) + @ERs → @(d:32+RdL)	ERs32-1→ERs32			—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs-, @(d:32,Rd.W)	B	5	6	8				D	S				@(d:32+Rd) + @ERs → @(d:32+Rd)	ERs32-1→ERs32			—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs-, @(d:32,ERd.L)	B	5	6	8				D	S				@(d:32+ERd) + @ERs → @(d:32+ERd)	ERs32-1→ERs32			—	↓	↑	↓	↑	↓	↑
	ADD.B @ERs-, @aa:16	B	4	5	7				S	D				@aa:16 + @ERs → @aa:16	ERs32-1→ERs32			—	↓	↑	↓	↑	↓	↑



Arithmetic operations (9)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	Addressing Mode											Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	Condition Code ^{s2}																						
						Number of 16-Bit Instruction Fetch Execution Stages ¹	pxx	Rn	@ERn	@(d,ERn)	@(d,Rn,WRn,LRn)	@ERn/@ERn+/@ERn+/@ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3						Operation 4	Operation 5	I	H	N	Z	V	C															
ADD	ADD.B @(d:16,Rs,W),@ERd-	B	4	6	7								S	D			@ERd + @(d:16+Rs) → @ERd		ERd32-1→ERd32																									
	ADD.B @(d:16,Rs,W),@+ERd	B	4	7	8								S	D		ERd32+1→ERd32	@ERd + @(d:16+Rs) → @ERd																											
	ADD.B @(d:16,Rs,W),@-ERd	B	4	7	8								S	D		ERd32-1→ERd32	@ERd + @(d:16+Rs) → @ERd																											
	ADD.B @(d:16,Rs,W),@(d:2,ERd)	B	4	7	8								D	S			@(1/2/3+ERd) + @(d:16+Rs) → @(1/2/3+ERd)																											
	ADD.B @(d:16,Rs,W),@(d:16,ERd)	B	5	7	8								D	S			@(d:16+ERd) + @(d:16+Rs) → @(d:16+ERd)																											
	ADD.B @(d:16,Rs,W),@(d:32,ERd)	B	6	7	9								D	S			@(d:32+ERd) + @(d:16+Rs) → @(d:32+ERd)																											
	ADD.B @(d:16,Rs,W),@(d:16,Rd,B)	B	5	7	8									S			@(d:16+RdL) + @(d:16+Rs) → @(d:16+RdL)																											
	ADD.B @(d:16,Rs,W),@(d:16,Rd,W)	B	5	7	8									S			@(d:16+Rd) + @(d:16+Rs) → @(d:16+Rd)																											
	ADD.B @(d:16,Rs,W),@(d:16,ERdL)	B	5	7	8									S			@(d:16+ERd) + @(d:16+Rs) → @(d:16+ERd)																											
	ADD.B @(d:16,Rs,W),@(d:32,Rd,B)	B	6	7	9									S			@(d:32+RdL) + @(d:16+Rs) → @(d:32+RdL)																											
	ADD.B @(d:16,Rs,W),@(d:32,Rd,W)	B	6	7	9									S			@(d:32+Rd) + @(d:16+Rs) → @(d:32+Rd)																											
	ADD.B @(d:16,Rs,W),@(d:32,ERdL)	B	6	7	9									S			@(d:32+ERd) + @(d:16+Rs) → @(d:32+ERd)																											
	ADD.B @(d:16,Rs,W),@aa:16	B	5	6	8									S	D		@aa:16 + @(d:16+Rs) → @aa:16																											
	ADD.B @(d:16,Rs,W),@aa:32	B	6	6	9									S	D		@aa:32 + @(d:16+Rs) → @aa:32																											
	ADD.B @(d:16,ERs,L),@ERd	B	4	6	7							D		S			@ERd + @(d:16+ERs) → @ERd																											
	ADD.B @(d:16,ERs,L),@ERd+	B	4	6	7									S	D		@ERd + @(d:16+ERs) → @ERd																											
	ADD.B @(d:16,ERs,L),@ERd-	B	4	6	7									S	D		@ERd + @(d:16+ERs) → @ERd																											
	ADD.B @(d:16,ERs,L),@+ERd	B	4	7	8									S	D			@ERd + @(d:16+ERs) → @ERd																										
	ADD.B @(d:16,ERs,L),@-ERd	B	4	7	8									S	D			@ERd + @(d:16+ERs) → @ERd																										
	ADD.B @(d:16,ERs,L),@(d:2,ERd)	B	4	7	8									D	S		@(1/2/3+ERd) + @(d:16+ERs) → @(1/2/3+ERd)																											
	ADD.B @(d:16,ERs,L),@(d:16,ERd)	B	5	7	8									D	S		@(d:16+ERd) + @(d:16+ERs) → @(d:16+ERd)																											
	ADD.B @(d:16,ERs,L),@(d:32,ERd)	B	6	7	9									D	S		@(d:32+ERd) + @(d:16+ERs) → @(d:32+ERd)																											
	ADD.B @(d:16,ERs,L),@(d:16,Rd,B)	B	5	7	8									S			@(d:16+RdL) + @(d:16+ERs) → @(d:16+RdL)																											
	ADD.B @(d:16,ERs,L),@(d:16,Rd,W)	B	5	7	8									S			@(d:16+Rd) + @(d:16+ERs) → @(d:16+Rd)																											
	ADD.B @(d:16,ERs,L),@(d:16,ERdL)	B	5	7	8									S			@(d:16+ERd) + @(d:16+ERs) → @(d:16+ERd)																											

Arithmetic operations (11)

Instruction n	Mnemonic	Size	Instruction Length	Addressing Mode										Operation ⁵					Condition Codes ²										
				Min.	16-Bit Instruction Fetch Execution Stages ¹	pxx	Rn	@ERN	@ (d,ERN)	@ (d,Rn,ERn,WERn,L)	@ ERn/@ ERn+/@ ERn+/@ +ERn	@ aa:8/@ aa:16/@ aa:32				Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C			
ADD	ADD.B @(d:32,Rs,W),@-ERd	B	5 7 9						S	D					ERd32-1→ERd32	@ERd + @(d:32+Rs) → @ERd							—	↓	↑	↓	↑	↓	
	ADD.B @(d:32,Rs,W),@(d:2,ERd)	B	5 7 9						D	S						@(1/2/3+ERd) + @(d:32+Rs) → @(1/2/3+ERd)							—	↓	↑	↓	↑	↓	
	ADD.B @(d:32,Rs,W),@(d:16,ERd)	B	6 7 9						D	S						@(d:16+ERd) + @(d:32+Rs) → @(d:16+ERd)							—	↓	↑	↓	↑	↓	
	ADD.B @(d:32,Rs,W),@(d:32,ERd)	B	7 7 1						D	S						@(d:32+ERd) + @(d:32+Rs) → @(d:32+ERd)							—	↓	↑	↓	↑	↓	
	ADD.B @(d:32,Rs,W),@(d:16,Rd,B)	B	6 7 9							S						@(d:16+RdL) + @(d:32+Rs) → @(d:16+RdL)							—	↓	↑	↓	↑	↓	
	ADD.B @(d:32,Rs,W),@(d:16,Rd,W)	B	6 7 9							S						@(d:16+Rd) + @(d:32+Rs) → @(d:16+Rd)							—	↓	↑	↓	↑	↓	
	ADD.B @(d:32,Rs,W),@(d:16,ERd,L)	B	6 7 9							S						@(d:16+ERd) + @(d:32+Rs) → @(d:16+ERd)							—	↓	↑	↓	↑	↓	
	ADD.B @(d:32,Rs,W),@(d:32,Rd,B)	B	7 7 1							S						@(d:32+RdL) + @(d:32+Rs) → @(d:32+RdL)							—	↓	↑	↓	↑	↓	
	ADD.B @(d:32,Rs,W),@(d:32,Rd,W)	B	7 7 1							S						@(d:32+Rd) + @(d:32+Rs) → @(d:32+Rd)							—	↓	↑	↓	↑	↓	
	ADD.B @(d:32,Rs,W),@(d:32,ERd,L)	B	7 7 1							S						@(d:32+ERd) + @(d:32+Rs) → @(d:32+ERd)							—	↓	↑	↓	↑	↓	
	ADD.B @(d:32,Rs,W),@aa:16	B	6 6 9							S	D					@aa:16 + @(d:32+Rs) → @aa:16							—	↓	↑	↓	↑	↓	
	ADD.B @(d:32,Rs,W),@aa:32	B	7 6 1							S	D					@aa:32 + @(d:32+Rs) → @aa:32							—	↓	↑	↓	↑	↓	
	ADD.B @(d:32,ERs,L),@ERd	B	5 6 8						D	S						@ERd + @(d:32+ERs) → @ERd							—	↓	↑	↓	↑	↓	
	ADD.B @(d:32,ERs,L),@ERd+	B	5 6 8							S	D					@ERd + @(d:32+ERs) → @ERd							ERd32+1→ERd32	—	↓	↑	↓	↑	↓
	ADD.B @(d:32,ERs,L),@ERd-	B	5 6 8							S	D					@ERd + @(d:32+ERs) → @ERd							ERd32-1→ERd32	—	↓	↑	↓	↑	↓
	ADD.B @(d:32,ERs,L),@+ERd	B	5 7 9							S	D			ERd32+1→ERd32	@ERd + @(d:32+ERs) → @ERd								—	↓	↑	↓	↑	↓	
	ADD.B @(d:32,ERs,L),@-ERd	B	5 7 9							S	D			ERd32-1→ERd32	@ERd + @(d:32+ERs) → @ERd								—	↓	↑	↓	↑	↓	
	ADD.B @(d:32,ERs,L),@(d:2,ERd)	B	5 7 9						D	S						@(1/2/3+ERd) + @(d:32+ERs) → @(1/2/3+ERd)							—	↓	↑	↓	↑	↓	
	ADD.B @(d:32,ERs,L),@(d:16,ERd)	B	6 7 9						D	S						@(d:16+ERd) + @(d:32+ERs) → @(d:16+ERd)							—	↓	↑	↓	↑	↓	
	ADD.B @(d:32,ERs,L),@(d:32,ERd)	B	7 7 1						D	S						@(d:32+ERd) + @(d:32+ERs) → @(d:32+ERd)							—	↓	↑	↓	↑	↓	
	ADD.B @(d:32,ERs,L),@(d:16,Rd,B)	B	6 7 9							S						@(d:16+RdL) + @(d:32+ERs) → @(d:16+RdL)							—	↓	↑	↓	↑	↓	
	ADD.B @(d:32,ERs,L),@(d:16,Rd,W)	B	6 7 9							S						@(d:16+Rd) + @(d:32+ERs) → @(d:16+Rd)							—	↓	↑	↓	↑	↓	
	ADD.B @(d:32,ERs,L),@(d:16,ERd,L)	B	6 7 9							S						@(d:16+ERd) + @(d:32+ERs) → @(d:16+ERd)							—	↓	↑	↓	↑	↓	
	ADD.B @(d:32,ERs,L),@(d:32,Rd,B)	B	7 7 1							S						@(d:32+RdL) + @(d:32+ERs) → @(d:32+RdL)							—	↓	↑	↓	↑	↓	
	ADD.B @(d:32,ERs,L),@(d:32,Rd,W)	B	7 7 1							S						@(d:32+Rd) + @(d:32+ERs) → @(d:32+Rd)							—	↓	↑	↓	↑	↓	

Arithmetic operations (12)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	Addressing Mode						Operation ⁵					Condition Codes ^{6,2}													
						Number of 16-Bit Instruction Fetch Execution Stages ¹	Rn	@(d,ERn)	@(d,ERn,W,ERn.L)	@(d,ERn)/@ERn+/@ERn-/@+ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C								
ADD	ADD.B @(d:32,ERs.L),@(d:32,ERd.L)	B	7	1						S							@(d:32+ERd) + @(d:32+ERs) → @(d:32+ERd)							—	↑	↑	↑	↑	↑	
	ADD.B @(d:32,ERs.L),@aa:16	B	6	9						S	D						@aa:16 + @(d:32+ERs) → @aa:16							—	↑	↑	↑	↑	↑	
	ADD.B @(d:32,ERs.L),@aa:32	B	7	6	1					S	D						@aa:32 + @(d:32+ERs) → @aa:32							—	↑	↑	↑	↑	↑	
	ADD.B @aa:16,@ERd	B	3	4	6				D			S					@ERd + @aa:16 → @ERd							—	↑	↑	↑	↑	↑	
	ADD.B @aa:16,@ERd+	B	3	4	6					D	S						@ERd + @aa:16 → @ERd							ERd32+1→ERd32	—	↑	↑	↑	↑	↑
	ADD.B @aa:16,@ERd-	B	3	4	6					D	S						@ERd + @aa:16 → @ERd							ERd32-1→ERd32	—	↑	↑	↑	↑	↑
	ADD.B @aa:16,@+ERd	B	3	5	7					D	S						ERd32+1→ERd32	@ERd + @aa:16 → @ERd							—	↑	↑	↑	↑	↑
	ADD.B @aa:16,@-ERd	B	3	5	7					D	S						ERd32-1→ERd32	@ERd + @aa:16 → @ERd							—	↑	↑	↑	↑	↑
	ADD.B @aa:16,@(d:2,ERd)	B	3	5	7					D	S						@(1/2/3+ERd) + @aa:16 → @(1/2/3+ERd)							—	↑	↑	↑	↑	↑	
	ADD.B @aa:16,@(d:16,ERd)	B	4	5	7					D	S						@(d:16+ERd) + @aa:16 → @(d:16+ERd)							—	↑	↑	↑	↑	↑	
	ADD.B @aa:16,@(d:32,ERd)	B	5	5	8					D	S						@(d:32+ERd) + @aa:16 → @(d:32+ERd)							—	↑	↑	↑	↑	↑	
	ADD.B @aa:16,@(d:16,Rd.B)	B	4	5	7					D	S						@(d:16+RdL) + @aa:16 → @(d:16+RdL)							—	↑	↑	↑	↑	↑	
	ADD.B @aa:16,@(d:16,Rd.W)	B	4	5	7					D	S						@(d:16+Rd) + @aa:16 → @(d:16+Rd)							—	↑	↑	↑	↑	↑	
	ADD.B @aa:16,@(d:16,ERd.L)	B	4	5	7					D	S						@(d:16+ERd) + @aa:16 → @(d:16+ERd)							—	↑	↑	↑	↑	↑	
	ADD.B @aa:16,@(d:32,Rd.B)	B	5	5	8					D	S						@(d:32+RdL) + @aa:16 → @(d:32+RdL)							—	↑	↑	↑	↑	↑	
	ADD.B @aa:16,@(d:32,Rd.W)	B	5	5	8					D	S						@(d:32+Rd) + @aa:16 → @(d:32+Rd)							—	↑	↑	↑	↑	↑	
	ADD.B @aa:16,@(d:32,ERd.L)	B	5	5	8					D	S						@(d:32+ERd) + @aa:16 → @(d:32+ERd)							—	↑	↑	↑	↑	↑	
	ADD.B @aa:16,@aa:16	B	4	4	7						S						@aa:16 + @aa:16 → @aa:16							—	↑	↑	↑	↑	↑	
	ADD.B @aa:16,@aa:32	B	5	4	8						S						@aa:32 + @aa:16 → @aa:32							—	↑	↑	↑	↑	↑	
	ADD.B @aa:32,@ERd	B	4	4	7				D			S					@ERd + @aa:32 → @ERd							—	↑	↑	↑	↑	↑	
	ADD.B @aa:32,@ERd+	B	4	4	7					D	S						@ERd + @aa:32 → @ERd							ERd32+1→ERd32	—	↑	↑	↑	↑	↑
	ADD.B @aa:32,@ERd-	B	4	4	7					D	S						@ERd + @aa:32 → @ERd							ERd32-1→ERd32	—	↑	↑	↑	↑	↑
	ADD.B @aa:32,@+ERd	B	4	5	8					D	S						ERd32+1→ERd32	@ERd + @aa:32 → @ERd							—	↑	↑	↑	↑	↑
	ADD.B @aa:32,@-ERd	B	4	5	8					D	S						ERd32-1→ERd32	@ERd + @aa:32 → @ERd							—	↑	↑	↑	↑	↑
	ADD.B @aa:32,@(d:2,ERd)	B	4	5	8					D	S						@(1/2/3+ERd) + @aa:32 → @(1/2/3+ERd)							—	↑	↑	↑	↑	↑	

Arithmetic operations (15)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode								Operation ⁵					Condition Codes ^{6,7}																										
				Min.	Max.	16-Bit Instruction Fetch Execution Status ¹	Fxx	Rn	@ERn	@ERn	@ERn.W/Rn.L											@ERn/ERn+/ERn-/@ERn-/@ERn	@aa:16/aa:32																				
												Number of Instructions	Fetch																														
n																																											
ADD	ADD.W @ERs,Rd	W	2	3	3		D																																				
	ADD.W @+ERs,Rd	W	2	4	4		D										ERs32+2→ERs32																										
	ADD.W @-ERs,Rd	W	2	4	4		D										ERs32-2→ERs32																										
	ADD.W @(d:2,ERs),Rd	W	2	4	4		D		S																																		
	ADD.W @(d:16,ERs),Rd	W	3	4	4		D		S																																		
	ADD.W @(d:32,ERs),Rd	W	4	4	5		D		S																																		
	ADD.W @(d:16,Rs,B),Rd	W	3	4	4		D		S																																		
	ADD.W @(d:16,Rs,W),Rd	W	3	4	4		D		S																																		
	ADD.W @(d:16,ERs,L),Rd	W	3	4	4		D		S																																		
	ADD.W @(d:32,Rs,B),Rd	W	4	4	5		D		S																																		
	ADD.W @(d:32,Rs,W),Rd	W	4	4	5		D		S																																		
	ADD.W @(d:32,ERs,L),Rd	W	4	4	5		D		S																																		
	ADD.W @aa:16,Rd	W	3	3	4		D				S																																
	ADD.W @aa:32,Rd	W	4	3	5		D				S																																
	ADD.W @ERs,@ERd	W	2	4	5			S																																			
	ADD.W @ERs,@ERd+	W	2	4	5			S			D																																
	ADD.W @ERs,@ERd-	W	2	4	5			S			D																																
	ADD.W @ERs,@+ERd	W	2	5	6			S			D							ERd32+2→ERd32																									
	ADD.W @ERs,@-ERd	W	2	5	6			S			D							ERd32-2→ERd32																									
	ADD.W @ERs,@(d:2,ERd)	W	2	5	6			S		D																																	
ADD.W @ERs,@(d:16,ERd)	W	3	5	6			S		D																																		
ADD.W @ERs,@(d:32,ERd)	W	4	5	7			S		D																																		
ADD.W @ERs,@(d:16,Rd,B)	W	3	5	6			S		D																																		
ADD.W @ERs,@(d:16,Rd,W)	W	3	5	6			S		D																																		

Arithmetic operations (16)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹ #Fxx	Addressing Mode								Operation ⁵					Condition Codes ²												
						Rn	@ERN	@(d,ERN)	@(d,Rn,ERN,WERN,L)	@ERN/ERN+/ERN-/@ERN-/@+ERN	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C									
ADD	ADD.W @ERs,@(d:16,ERd.L)	W	3	5	6		S	D							@(d:16+ERd<<1) + @ERs	→	@(d:16+ERd<<1)							—	↑	↓	↑	↓	↑	↓	
	ADD.W @ERs,@(d:32,Rd.B)	W	4	5	7		S	D							@(d:32+RdL<<1) + @ERs	→	@(d:32+RdL<<1)								—	↑	↓	↑	↓	↑	↓
	ADD.W @ERs,@(d:32,Rd.W)	W	4	5	7		S	D							@(d:32+Rd<<1) + @ERs	→	@(d:32+Rd<<1)								—	↑	↓	↑	↓	↑	↓
	ADD.W @ERs,@(d:32,ERd.L)	W	4	5	7		S	D							@(d:32+ERd<<1) + @ERs	→	@(d:32+ERd<<1)								—	↑	↓	↑	↓	↑	↓
	ADD.W @ERs,@aa:16	W	3	4	6		S			D					@aa:16 + @ERs	→	@aa:16								—	↑	↓	↑	↓	↑	↓
	ADD.W @ERs,@aa:32	W	4	4	7		S			D					@aa:32 + @ERs	→	@aa:32								—	↑	↓	↑	↓	↑	↓
	ADD.W @ERs+,@ERd	W	3	5	6			D	S						@ERd + @ERs	→	@ERd	ERs32+2→ERs32							—	↑	↓	↑	↓	↑	↓
	ADD.W @ERs+,@ERd+	W	3	5	6				S						@ERd + @ERs	→	@ERd	ERs32+2→ERs32	ERd32+2→ERd32						—	↑	↓	↑	↓	↑	↓
	ADD.W @ERs+,@ERd-	W	3	5	6				S						@ERd + @ERs	→	@ERd	ERs32+2→ERs32	ERd32-2→ERd32						—	↑	↓	↑	↓	↑	↓
	ADD.W @ERs+,@+ERd	W	3	6	7				S					ERd32+2→ERd32	@ERd + @ERs	→	@ERd	ERs32+2→ERs32							—	↑	↓	↑	↓	↑	↓
	ADD.W @ERs+,@-ERd	W	3	6	7				S				ERd32-2→ERd32	@ERd + @ERs	→	@ERd	ERs32+2→ERs32								—	↑	↓	↑	↓	↑	↓
	ADD.W @ERs+,@(d:2,ERd)	W	3	6	7			D	S					@(2/4/6+ERd) + @ERs	→	@(2/4/6+ERd)	ERs32+2→ERs32								—	↑	↓	↑	↓	↑	↓
	ADD.W @ERs+,@(d:16,ERd)	W	4	6	7			D	S					@(d:16+ERd) + @ERs	→	@(d:16+ERd)	ERs32+2→ERs32								—	↑	↓	↑	↓	↑	↓
	ADD.W @ERs+,@(d:32,ERd)	W	5	6	8			D	S					@(d:32+ERd) + @ERs	→	@(d:32+ERd)	ERs32+2→ERs32								—	↑	↓	↑	↓	↑	↓
	ADD.W @ERs+,@(d:16,Rd.B)	W	4	6	7				D	S				@(d:16+RdL<<1) + @ERs	→	@(d:16+RdL<<1)	ERs32+2→ERs32								—	↑	↓	↑	↓	↑	↓
	ADD.W @ERs+,@(d:16,Rd.W)	W	4	6	7				D	S				@(d:16+Rd<<1) + @ERs	→	@(d:16+Rd<<1)	ERs32+2→ERs32								—	↑	↓	↑	↓	↑	↓
	ADD.W @ERs+,@(d:16,ERd.L)	W	4	6	7				D	S				@(d:16+ERd<<1) + @ERs	→	@(d:16+ERd<<1)	ERs32+2→ERs32								—	↑	↓	↑	↓	↑	↓
	ADD.W @ERs+,@(d:32,Rd.B)	W	5	6	8				D	S				@(d:32+RdL<<1) + @ERs	→	@(d:32+RdL<<1)	ERs32+2→ERs32								—	↑	↓	↑	↓	↑	↓
	ADD.W @ERs+,@(d:32,Rd.W)	W	5	6	8				D	S				@(d:32+Rd<<1) + @ERs	→	@(d:32+Rd<<1)	ERs32+2→ERs32								—	↑	↓	↑	↓	↑	↓
	ADD.W @ERs+,@(d:32,ERd.L)	W	5	6	8				D	S				@(d:32+ERd<<1) + @ERs	→	@(d:32+ERd<<1)	ERs32+2→ERs32								—	↑	↓	↑	↓	↑	↓
	ADD.W @ERs+,@aa:16	W	4	5	7				S	D				@aa:16 + @ERs	→	@aa:16	ERs32+2→ERs32								—	↑	↓	↑	↓	↑	↓
	ADD.W @ERs+,@aa:32	W	5	5	8				S	D				@aa:32 + @ERs	→	@aa:32	ERs32+2→ERs32								—	↑	↓	↑	↓	↑	↓
	ADD.W @ERs-,@ERd	W	3	5	6			D	S					@ERd + @ERs	→	@ERd	ERs32-2→ERs32								—	↑	↓	↑	↓	↑	↓
	ADD.W @ERs-,@ERd+	W	3	5	6				S					@ERd + @ERs	→	@ERd	ERs32-2→ERs32	ERd32+2→ERd32							—	↑	↓	↑	↓	↑	↓
	ADD.W @ERs-,@ERd-	W	3	5	6				S					@ERd + @ERs	→	@ERd	ERs32-2→ERs32	ERd32-2→ERd32							—	↑	↓	↑	↓	↑	↓



Arithmetic operations (17)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹ #xx	Addressing Mode					Operation ⁵					Condition Codes ²															
						Rn	@(d,ERn)	@(d,Rn,ERn,WERn,L)	@(d,Rn/ERn+/ERn+/@ERn/@@+ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C										
ADD	ADD.W @ERs-, @+ERd	W	3	6	7							ERd32+2→ERd32	@ERd	+	@ERs	→	@ERd	ERs32-2→ERs32													
	ADD.W @ERs-, @-ERd	W	3	6	7							ERd32-2→ERd32	@ERd	+	@ERs	→	@ERd	ERs32-2→ERs32													
	ADD.W @ERs-, @(d:2,ERd)	W	3	6	7					D	S				@ERs	→	@(2/4/6+ERd)	ERs32-2→ERs32													
	ADD.W @ERs-, @(d:16,ERd)	W	4	6	7					D	S				@ERs	→	@(d:16+ERd)	ERs32-2→ERs32													
	ADD.W @ERs-, @(d:32,ERd)	W	5	6	8					D	S				@ERs	→	@(d:32+ERd)	ERs32-2→ERs32													
	ADD.W @ERs-, @(d:16,Rd,B)	W	4	6	7					D	S				@ERs	→	@(d:16+RdL<<1)	ERs32-2→ERs32													
	ADD.W @ERs-, @(d:16,Rd,W)	W	4	6	7					D	S				@ERs	→	@(d:16+RdL<<1)	ERs32-2→ERs32													
	ADD.W @ERs-, @(d:16,ERd,L)	W	4	6	7					D	S				@ERs	→	@(d:16+ERdL<<1)	ERs32-2→ERs32													
	ADD.W @ERs-, @(d:32,Rd,B)	W	5	6	8					D	S				@ERs	→	@(d:32+RdL<<1)	ERs32-2→ERs32													
	ADD.W @ERs-, @(d:32,Rd,W)	W	5	6	8					D	S				@ERs	→	@(d:32+RdL<<1)	ERs32-2→ERs32													
	ADD.W @ERs-, @(d:32,ERd,L)	W	5	6	8					D	S				@ERs	→	@(d:32+ERdL<<1)	ERs32-2→ERs32													
	ADD.W @ERs-, @aa:16	W	4	5	7						S	D			@aa:16	+	@ERs	→	@aa:16	ERs32-2→ERs32											
	ADD.W @ERs-, @aa:32	W	5	5	8						S	D			@aa:32	+	@ERs	→	@aa:32	ERs32-2→ERs32											
	ADD.W @+ERs, @ERd	W	3	6	6						D	S		ERs32+2→ERs32	@ERd	+	@ERs	→	@ERd												
	ADD.W @+ERs, @ERd+	W	3	6	6						S	S		ERs32+2→ERs32	@ERd	+	@ERs	→	@ERd		ERd32+2→ERd32										
	ADD.W @+ERs, @ERd-	W	3	6	6						S	S		ERs32+2→ERs32	@ERd	+	@ERs	→	@ERd		ERd32-2→ERd32										
	ADD.W @+ERs, @+ERd	W	3	7	7						S	S		ERs32+2→ERs32	ERd32+2→ERd32	@ERd	+	@ERs	→	@ERd											
	ADD.W @+ERs, @-ERd	W	3	7	7						S	S		ERs32+2→ERs32	ERd32-2→ERd32	@ERd	+	@ERs	→	@ERd											
	ADD.W @+ERs, @(d:2,ERd)	W	3	7	7					D	S			ERs32+2→ERs32	@(2/4/6+ERd)	+	@ERs	→	@(2/4/6+ERd)												
	ADD.W @+ERs, @(d:16,ERd)	W	4	7	7					D	S			ERs32+2→ERs32	@(d:16+ERd)	+	@ERs	→	@(d:16+ERd)												
	ADD.W @+ERs, @(d:32,ERd)	W	5	7	8					D	S			ERs32+2→ERs32	@(d:32+ERd)	+	@ERs	→	@(d:32+ERd)												
	ADD.W @+ERs, @(d:16,Rd,B)	W	4	7	7					D	S			ERs32+2→ERs32	@(d:16+RdL<<1)	+	@ERs	→	@(d:16+RdL<<1)												
	ADD.W @+ERs, @(d:16,Rd,W)	W	4	7	7					D	S			ERs32+2→ERs32	@(d:16+RdL<<1)	+	@ERs	→	@(d:16+RdL<<1)												
	ADD.W @+ERs, @(d:16,ERd,L)	W	4	7	7					D	S			ERs32+2→ERs32	@(d:16+ERdL<<1)	+	@ERs	→	@(d:16+ERdL<<1)												
	ADD.W @+ERs, @(d:32,Rd,B)	W	5	7	8					D	S			ERs32+2→ERs32	@(d:32+RdL<<1)	+	@ERs	→	@(d:32+RdL<<1)												

Arithmetic operations (18)

Instruction n	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹ #xx	Addressing Mode							Operation ⁵					Condition Codes ²									
						Rn	@ERn	@(d,ERn)	@(d,Rn,ERn,WERn,L)	@ERn/@ERn+/@ERn-/@+ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C					
ADD	ADD.W @+ERs,@(d:32,Rd,W)	W	5	7	8				D	S				ERs32-2→ERs32		@(d:32+Rd<<1) + @ERs	→	@(d:32+Rd<<1)				—	↑	↑	↑	↑	↑
	ADD.W @+ERs,@(d:32,ERd,L)	W	5	7	8				D	S				ERs32-2→ERs32		@(d:32+ERd<<1) + @ERs	→	@(d:32+ERd<<1)				—	↑	↑	↑	↑	↑
	ADD.W @+ERs,@aa:16	W	4	6	7						S	D		ERs32-2→ERs32		@aa:16 + @ERs	→	@aa:16				—	↑	↑	↑	↑	↑
	ADD.W @+ERs,@aa:32	W	5	6	8						S	D		ERs32-2→ERs32		@aa:32 + @ERs	→	@aa:32				—	↑	↑	↑	↑	↑
	ADD.W @-ERs,@ERd	W	3	6	6				D		S			ERs32-2→ERs32		@ERd + @ERs	→	@ERd				—	↑	↑	↑	↑	↑
	ADD.W @-ERs,@ERd+	W	3	6	6						S			ERs32-2→ERs32		@ERd + @ERs	→	@ERd		ERd32+2→ERd32		—	↑	↑	↑	↑	↑
	ADD.W @-ERs,@ERd-	W	3	6	6						S			ERs32-2→ERs32		@ERd + @ERs	→	@ERd		ERd32-2→ERd32		—	↑	↑	↑	↑	↑
	ADD.W @-ERs,@+ERd	W	3	7	7						S			ERs32-2→ERs32	ERd32+2→ERd32	@ERd + @ERs	→	@ERd				—	↑	↑	↑	↑	↑
	ADD.W @-ERs,@-ERd	W	3	7	7						S			ERs32-2→ERs32	ERd32-2→ERd32	@ERd + @ERs	→	@ERd				—	↑	↑	↑	↑	↑
	ADD.W @-ERs,@(d:2,ERd)	W	3	7	7				D		S			ERs32-2→ERs32		@(2/4/6+ERd) + @ERs	→	@(2/4/6+ERd)				—	↑	↑	↑	↑	↑
	ADD.W @-ERs,@(d:16,ERd)	W	4	7	7				D		S			ERs32-2→ERs32		@(d:16+ERd) + @ERs	→	@(d:16+ERd)				—	↑	↑	↑	↑	↑
	ADD.W @-ERs,@(d:32,ERd)	W	5	7	8				D		S			ERs32-2→ERs32		@(d:32+ERd) + @ERs	→	@(d:32+ERd)				—	↑	↑	↑	↑	↑
	ADD.W @-ERs,@(d:16,Rd,B)	W	4	7	7					D	S			ERs32-2→ERs32		@(d:16+RdL<<1) + @ERs	→	@(d:16+RdL<<1)				—	↑	↑	↑	↑	↑
	ADD.W @-ERs,@(d:16,Rd,W)	W	4	7	7					D	S			ERs32-2→ERs32		@(d:16+ERd<<1) + @ERs	→	@(d:16+ERd<<1)				—	↑	↑	↑	↑	↑
	ADD.W @-ERs,@(d:16,ERd,L)	W	4	7	7					D	S			ERs32-2→ERs32		@(d:16+ERd<<1) + @ERs	→	@(d:16+ERd<<1)				—	↑	↑	↑	↑	↑
	ADD.W @-ERs,@(d:32,Rd,B)	W	5	7	8					D	S			ERs32-2→ERs32		@(d:32+RdL<<1) + @ERs	→	@(d:32+RdL<<1)				—	↑	↑	↑	↑	↑
	ADD.W @-ERs,@(d:32,Rd,W)	W	5	7	8					D	S			ERs32-2→ERs32		@(d:32+Rd<<1) + @ERs	→	@(d:32+Rd<<1)				—	↑	↑	↑	↑	↑
	ADD.W @-ERs,@(d:32,ERd,L)	W	5	7	8					D	S			ERs32-2→ERs32		@(d:32+ERd<<1) + @ERs	→	@(d:32+ERd<<1)				—	↑	↑	↑	↑	↑
	ADD.W @-ERs,@aa:16	W	4	6	7						S	D		ERs32-2→ERs32		@aa:16 + @ERs	→	@aa:16				—	↑	↑	↑	↑	↑
	ADD.W @-ERs,@aa:32	W	5	6	8						S	D		ERs32-2→ERs32		@aa:32 + @ERs	→	@aa:32				—	↑	↑	↑	↑	↑
ADD.W @(d:2,ERs),@ERd	W	3	6	6				D	S						@ERd + @ (2/4/6+ERs)	→	@ERd				—	↑	↑	↑	↑	↑	
ADD.W @(d:2,ERs),@ERd+	W	3	6	6					S	D					@ERd + @ (2/4/6+ERs)	→	@ERd		ERd32+2→ERd32		—	↑	↑	↑	↑	↑	
ADD.W @(d:2,ERs),@ERd-	W	3	6	6					S	D					@ERd + @ (2/4/6+ERs)	→	@ERd		ERd32-2→ERd32		—	↑	↑	↑	↑	↑	
ADD.W @(d:2,ERs),@+ERd	W	3	7	7					S	D				ERd32+2→ERd32	@ERd + @ (2/4/6+ERs)	→	@ERd				—	↑	↑	↑	↑	↑	
ADD.W @(d:2,ERs),@-ERd	W	3	7	7					S	D				ERd32-2→ERd32	@ERd + @ (2/4/6+ERs)	→	@ERd				—	↑	↑	↑	↑	↑	



Arithmetic operations (19)

Instruction	Mnemonic	Size	Instruction Length			Addressing Mode								Operation ^s					Condition Codes ^{s2}																		
			Min.	Max.	Number of 16-bit Instruction Fetch Execution Stages ¹	Rn	@ERn	@(d,ERn)	@(d,Rn,WRn,L)	@ERn/@ERn+/@ERn+/@+ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2												Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C				
ADD	ADD.W @(d:2,ERs),@(d:2,ERd)	W	3	7	7						S																										
	ADD.W @(d:2,ERs),@(d:16,ERd)	W	4	7	7						S																										
	ADD.W @(d:2,ERs),@(d:32,ERd)	W	5	7	8						S																										
	ADD.W @(d:2,ERs),@(d:16,Rd.B)	W	4	7	7						S	D																									
	ADD.W @(d:2,ERs),@(d:16,Rd.W)	W	4	7	7						S	D																									
	ADD.W @(d:2,ERs),@(d:16,ERd.L)	W	4	7	7						S	D																									
	ADD.W @(d:2,ERs),@(d:32,Rd.B)	W	5	7	8						S	D																									
	ADD.W @(d:2,ERs),@(d:32,Rd.W)	W	5	7	8						S	D																									
	ADD.W @(d:2,ERs),@(d:32,ERd.L)	W	5	7	8						S	D																									
	ADD.W @(d:2,ERs),@aa:16	W	4	6	7						S			D																							
	ADD.W @(d:2,ERs),@aa:32	W	5	6	8						S			D																							
	ADD.W @(d:16,ERs),@ERd	W	4	6	7						D	S																									
	ADD.W @(d:16,ERs),@ERd+	W	4	6	7						S	D																									
	ADD.W @(d:16,ERs),@ERd-	W	4	6	7						S	D																									
	ADD.W @(d:16,ERs),@+ERd	W	4	7	8						S	D				ERd32+2→ERd32	@ERd	+	@(d:16+ERs)	→	@ERd																
	ADD.W @(d:16,ERs),@-ERd	W	4	7	8						S	D				ERd32-2→ERd32	@ERd	+	@(d:16+ERs)	→	@ERd																
	ADD.W @(d:16,ERs),@(d:2,ERd)	W	4	7	8						S										@(2/4+ERd)	+	@(d:16+ERs)	→	@(2/4+ERd)												
	ADD.W @(d:16,ERs),@(d:16,ERd)	W	5	7	8						S										@(d:16+ERd)	+	@(d:16+ERs)	→	@(d:16+ERd)												
	ADD.W @(d:16,ERs),@(d:32,ERd)	W	6	7	9						S										@(d:32+ERd)	+	@(d:16+ERs)	→	@(d:32+ERd)												
	ADD.W @(d:16,ERs),@(d:16,Rd.B)	W	5	7	8						S	D									@(d:16+RdL<<1)	+	@(d:16+ERs)	→	@(d:16+RdL<<1)												
	ADD.W @(d:16,ERs),@(d:16,Rd.W)	W	5	7	8						S	D									@(d:16+Rd<<1)	+	@(d:16+ERs)	→	@(d:16+Rd<<1)												
	ADD.W @(d:16,ERs),@(d:16,ERd.L)	W	5	7	8						S	D									@(d:16+ERd<<1)	+	@(d:16+ERs)	→	@(d:16+ERd<<1)												
	ADD.W @(d:16,ERs),@(d:32,Rd.B)	W	6	7	9						S	D									@(d:32+RdL<<1)	+	@(d:16+ERs)	→	@(d:32+RdL<<1)												
	ADD.W @(d:16,ERs),@(d:32,Rd.W)	W	6	7	9						S	D									@(d:32+Rd<<1)	+	@(d:16+ERs)	→	@(d:32+Rd<<1)												
	ADD.W @(d:16,ERs),@(d:32,ERd.L)	W	6	7	9						S	D									@(d:32+ERd<<1)	+	@(d:16+ERs)	→	@(d:32+ERd<<1)												

Arithmetic operations (20)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode								Operation ⁵					Condition Codes ^{6,7}								
				Min.	16-Bit Instruction Fetch Execution Stages ¹	16-Bit Instruction Fetch Execution Stages ¹	16-Bit Instruction Fetch Execution Stages ¹	Rn	Rs	Rs	Rs														
ADD	ADD.W @(d:16,ERs),@aa:16	W	5 6 8					S		D															
	ADD.W @(d:16,ERs),@aa:32	W	6 6 9					S		D															
	ADD.W @(d:32,ERs),@ERd	W	5 6 8					D S																	
	ADD.W @(d:32,ERs),@ERd+	W	5 6 8					S		D											ERd32+2→ERd32				
	ADD.W @(d:32,ERs),@ERd-	W	5 6 8					S		D											ERd32-2→ERd32				
	ADD.W @(d:32,ERs),@+ERd	W	5 7 9					S		D		ERd32+2→ERd32													
	ADD.W @(d:32,ERs),@-ERd	W	5 7 9					S		D		ERd32-2→ERd32													
	ADD.W @(d:32,ERs),@(d:2,ERd)	W	5 7 9					S				@(2/4+ERd) + @(d:32+ERs)													
	ADD.W @(d:32,ERs),@(d:16,ERd)	W	6 7 9					S				@(d:16+ERd) + @(d:32+ERs)													
	ADD.W @(d:32,ERs),@(d:32,ERd)	W	7 7 1					S				@(d:32+ERd) + @(d:32+ERs)													
	ADD.W @(d:32,ERs),@(d:16,Rd,B)	W	6 7 9					S D				@(d:16+Rd<<1) + @(d:32+ERs)													
	ADD.W @(d:32,ERs),@(d:16,Rd,W)	W	6 7 9					S D				@(d:16+Rd<<1) + @(d:32+ERs)													
	ADD.W @(d:32,ERs),@(d:16,ERdL)	W	6 7 9					S D				@(d:16+ERd<<1) + @(d:32+ERs)													
	ADD.W @(d:32,ERs),@(d:32,Rd,B)	W	7 7 1					S D				@(d:32+RdL<<1) + @(d:32+ERs)													
	ADD.W @(d:32,ERs),@(d:32,Rd,W)	W	7 7 1					S D				@(d:32+Rd<<1) + @(d:32+ERs)													
	ADD.W @(d:32,ERs),@(d:32,ERdL)	W	7 7 1					S D				@(d:32+ERd<<1) + @(d:32+ERs)													
	ADD.W @(d:32,ERs),@aa:16	W	6 6 9					S		D		@aa:16 + @(d:32+ERs)													
	ADD.W @(d:32,ERs),@aa:32	W	7 6 1					S		D		@aa:32 + @(d:32+ERs)													
	ADD.W @(d:16,Rs,B),@ERd	W	4 6 7					D S				@ERd + @(d:16+RsL<<1)													
	ADD.W @(d:16,Rs,B),@ERd+	W	4 6 7					S D				@ERd + @(d:16+RsL<<1)									ERd32+2→ERd32				
	ADD.W @(d:16,Rs,B),@ERd-	W	4 6 7					S D				@ERd + @(d:16+RsL<<1)									ERd32-2→ERd32				
	ADD.W @(d:16,Rs,B),@+ERd	W	4 7 8					S D			ERd32+2→ERd32	@ERd + @(d:16+RsL<<1)													
	ADD.W @(d:16,Rs,B),@-ERd	W	4 7 8					S D			ERd32-2→ERd32	@ERd + @(d:16+RsL<<1)													
	ADD.W @(d:16,Rs,B),@(d:2,ERd)	W	4 7 8					D S				@(2/4+ERd) + @(d:16+RsL<<1)													
	ADD.W @(d:16,Rs,B),@(d:16,ERd)	W	5 7 8					D S				@(d:16+ERd) + @(d:16+RsL<<1)													

Arithmetic operations (22)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode											Operation ⁶					Condition Codes ^{5,2}																					
				Min.	16-Bit Instruction Fetch Execution States ¹ #xx	Number of Execution States ¹	Rn	@ERN	@(d, ERn)	@(d, ERn), W(ERn, L)	@(d, ERn)/@ERn+/@ERn+/@ERN+/@ERN+/@ERN+/@ERN+/@ERN	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C																		
ADD	ADD.W @(d:16,ERs.L),@ERd	W	4	6	7					D	S									@ERd + @(d:16+ERs<<1) → @ERd											—	↑	↑	↑	↑	↑					
	ADD.W @(d:16,ERs.L),@ERd+	W	4	6	7						S	D								@ERd + @(d:16+ERs<<1) → @ERd				ERd32+2→ERd32																	
	ADD.W @(d:16,ERs.L),@ERd-	W	4	6	7						S	D								@ERd + @(d:16+ERs<<1) → @ERd				ERd32-2→ERd32																	
	ADD.W @(d:16,ERs.L),@+ERd	W	4	7	8						S	D			ERd32+2→ERd32					@ERd + @(d:16+ERs<<1) → @ERd																					
	ADD.W @(d:16,ERs.L),@-ERd	W	4	7	8						S	D			ERd32-2→ERd32					@ERd + @(d:16+ERs<<1) → @ERd																					
	ADD.W @(d:16,ERs.L),@(d:2,ERd)	W	4	7	8					D	S									@(2/4/6+ERd) + @(d:16+ERs<<1) → @(2/4/6+ERd)																					
	ADD.W @(d:16,ERs.L),@(d:16,ERd)	W	5	7	8						D	S								@(d:16+ERd) + @(d:16+ERs<<1) → @(d:16+ERd)																					
	ADD.W @(d:16,ERs.L),@(d:32,ERd)	W	6	7	9						D	S								@(d:32+ERd) + @(d:16+ERs<<1) → @(d:32+ERd)																					
	ADD.W @(d:16,ERs.L),@(d:16,Rd.B)	W	5	7	8						S									@(d:16+RdL<<1) + @(d:16+ERs<<1) → @(d:16+RdL<<1)																					
	ADD.W @(d:16,ERs.L),@(d:16,Rd.W)	W	5	7	8						S									@(d:16+Rd<<1) + @(d:16+ERs<<1) → @(d:16+Rd<<1)																					
	ADD.W @(d:16,ERs.L),@(d:16,ERd.L)	W	5	7	8						S									@(d:16+ERd<<1) + @(d:16+ERs<<1) → @(d:16+ERd<<1)																					
	ADD.W @(d:16,ERs.L),@(d:32,Rd.B)	W	6	7	9						S									@(d:32+RdL<<1) + @(d:16+ERs<<1) → @(d:32+RdL<<1)																					
	ADD.W @(d:16,ERs.L),@(d:32,Rd.W)	W	6	7	9						S									@(d:32+Rd<<1) + @(d:16+ERs<<1) → @(d:32+Rd<<1)																					
	ADD.W @(d:16,ERs.L),@(d:32,ERd.L)	W	6	7	9						S									@(d:32+ERd<<1) + @(d:16+ERs<<1) → @(d:32+ERd<<1)																					
	ADD.W @(d:16,ERs.L),@aa:16	W	5	6	8						S	D								@aa:16 + @(d:16+ERs<<1) → @aa:16																					
	ADD.W @(d:16,ERs.L),@aa:32	W	6	6	9						S	D								@aa:32 + @(d:16+ERs<<1) → @aa:32																					
	ADD.W @(d:32,Rs.B),@ERd	W	5	6	8					D	S									@ERd + @(d:32+RsL<<1) → @ERd																					
	ADD.W @(d:32,Rs.B),@ERd+	W	5	6	8						S	D								@ERd + @(d:32+RsL<<1) → @ERd				ERd32+2→ERd32																	
	ADD.W @(d:32,Rs.B),@ERd-	W	5	6	8						S	D								@ERd + @(d:32+RsL<<1) → @ERd				ERd32-2→ERd32																	
	ADD.W @(d:32,Rs.B),@+ERd	W	5	7	9						S	D			ERd32+2→ERd32					@ERd + @(d:32+RsL<<1) → @ERd																					
	ADD.W @(d:32,Rs.B),@-ERd	W	5	7	9						S	D			ERd32-2→ERd32					@ERd + @(d:32+RsL<<1) → @ERd																					
	ADD.W @(d:32,Rs.B),@(d:2,ERd)	W	5	7	9					D	S									@(2/4/6+ERd) + @(d:32+RsL<<1) → @(2/4/6+ERd)																					
	ADD.W @(d:32,Rs.B),@(d:16,ERd)	W	6	7	9						D	S								@(d:16+ERd) + @(d:32+RsL<<1) → @(d:16+ERd)																					
	ADD.W @(d:32,Rs.B),@(d:32,ERd)	W	7	7	1						D	S								@(d:32+ERd) + @(d:32+RsL<<1) → @(d:32+ERd)																					

Arithmetic operations (23)

Instruction	Mnemonic	Size	Instruction Length	Min.	Number of 16-Bit Instruction Fetch Execution Stages ¹	Addressing Mode							Operation ⁵					Condition Codes ²																												
						16	17	18	19	20	21	22	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C																							
ADD	ADD.W @ (d:32, Rs.B), @ (d:16, Rd.B)	W	6	7	9						S																																			
	ADD.W @ (d:32, Rs.B), @ (d:16, Rd.W)	W	6	7	9						S																																			
	ADD.W @ (d:32, Rs.B), @ (d:16, ERd.L)	W	6	7	9						S																																			
	ADD.W @ (d:32, Rs.B), @ (d:32, Rd.B)	W	7	7	1						S																																			
	ADD.W @ (d:32, Rs.B), @ (d:32, Rd.W)	W	7	7	1						S																																			
	ADD.W @ (d:32, Rs.B), @ (d:32, ERd.L)	W	7	7	1						S																																			
	ADD.W @ (d:32, Rs.B), @ aa:16	W	6	6	9						S	D																																		
	ADD.W @ (d:32, Rs.B), @ aa:32	W	7	6	1						S	D																																		
	ADD.W @ (d:32, Rs.W), @ ERd	W	5	6	8				D		S																																			
	ADD.W @ (d:32, Rs.W), @ ERd+	W	5	6	8						S	D																																		
	ADD.W @ (d:32, Rs.W), @ ERd-	W	5	6	8						S	D																																		
	ADD.W @ (d:32, Rs.W), @ +ERd	W	5	7	9						S	D																																		
	ADD.W @ (d:32, Rs.W), @ -ERd	W	5	7	9						S	D																																		
	ADD.W @ (d:32, Rs.W), @ (d:2, ERd)	W	5	7	9						D	S																																		
	ADD.W @ (d:32, Rs.W), @ (d:16, ERd)	W	6	7	9						D	S																																		
	ADD.W @ (d:32, Rs.W), @ (d:32, ERd)	W	7	7	1						D	S																																		
	ADD.W @ (d:32, Rs.W), @ (d:16, Rd.B)	W	6	7	9							S																																		
	ADD.W @ (d:32, Rs.W), @ (d:16, Rd.W)	W	6	7	9							S																																		
	ADD.W @ (d:32, Rs.W), @ (d:16, ERd.L)	W	6	7	9							S																																		
	ADD.W @ (d:32, Rs.W), @ (d:32, Rd.B)	W	7	7	1							S																																		
	ADD.W @ (d:32, Rs.W), @ (d:32, Rd.W)	W	7	7	1							S																																		
	ADD.W @ (d:32, Rs.W), @ (d:32, ERd.L)	W	7	7	1							S																																		
	ADD.W @ (d:32, Rs.W), @ aa:16	W	6	6	9							S	D																																	
	ADD.W @ (d:32, Rs.W), @ aa:32	W	7	6	1							S	D																																	
	ADD.W @ (d:32, ERs.L), @ ERd	W	5	6	8				D			S																																		

Arithmetic operations (24)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode										Operation ⁵					Condition Codes ²						
				Min.	16-Bit Instruction Fetch Execution Status ¹	Prx	Rn	@ERn	@(d,ERn)	@(d,Rn,WRn,L)	@ERn/@ERn+/@ERn+/@ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C		
ADD	ADD.W @(d:32,ERs.L),@ERd+	W	5 6 8							S	D					@ERd + @(d:32+ERs<<1) → @ERd			ERd32+2→ERd32	—	↓	↑	↓	↑	↓
	ADD.W @(d:32,ERs.L),@ERd-	W	5 6 8							S	D					@ERd + @(d:32+ERs<<1) → @ERd			ERd32-2→ERd32	—	↓	↑	↓	↑	↓
	ADD.W @(d:32,ERs.L),@+ERd	W	5 7 9							S	D			ERd32+2→ERd32		@ERd + @(d:32+ERs<<1) → @ERd				—	↓	↑	↓	↑	↓
	ADD.W @(d:32,ERs.L),@-ERd	W	5 7 9							S	D			ERd32-2→ERd32		@ERd + @(d:32+ERs<<1) → @ERd				—	↓	↑	↓	↑	↓
	ADD.W @(d:32,ERs.L),@(d:2,ERd)	W	5 7 9							D	S					@(2/4/6+ERd) + @(d:32+ERs<<1) → @(2/4/6+ERd)				—	↓	↑	↓	↑	↓
	ADD.W @(d:32,ERs.L),@(d:16,ERd)	W	6 7 9							D	S					@(d:16+ERd) + @(d:32+ERs<<1) → @(d:16+ERd)				—	↓	↑	↓	↑	↓
	ADD.W @(d:32,ERs.L),@(d:32,ERd)	W	7 7 1							D	S					@(d:32+ERd) + @(d:32+ERs<<1) → @(d:32+ERd)				—	↓	↑	↓	↑	↓
	ADD.W @(d:32,ERs.L),@(d:16,Rd.B)	W	6 7 9							S	S					@(d:16+Rd<<1) + @(d:32+ERs<<1) → @(d:16+Rd<<1)				—	↓	↑	↓	↑	↓
	ADD.W @(d:32,ERs.L),@(d:16,Rd.W)	W	6 7 9							S	S					@(d:16+Rd<<1) + @(d:32+ERs<<1) → @(d:16+Rd<<1)				—	↓	↑	↓	↑	↓
	ADD.W @(d:32,ERs.L),@(d:16,ERd.L)	W	6 7 9							S	S					@(d:16+ERd<<1) + @(d:32+ERs<<1) → @(d:16+ERd<<1)				—	↓	↑	↓	↑	↓
	ADD.W @(d:32,ERs.L),@(d:32,Rd.B)	W	7 7 1							S	S					@(d:32+RdL<<1) + @(d:32+ERs<<1) → @(d:32+RdL<<1)				—	↓	↑	↓	↑	↓
	ADD.W @(d:32,ERs.L),@(d:32,Rd.W)	W	7 7 1							S	S					@(d:32+Rd<<1) + @(d:32+ERs<<1) → @(d:32+Rd<<1)				—	↓	↑	↓	↑	↓
	ADD.W @(d:32,ERs.L),@(d:32,ERd.L)	W	7 7 1							S	S					@(d:32+ERd<<1) + @(d:32+ERs<<1) → @(d:32+ERd<<1)				—	↓	↑	↓	↑	↓
	ADD.W @(d:32,ERs.L),@aa:16	W	6 6 9							S	D					@aa:16 + @(d:32+ERs<<1) → @aa:16				—	↓	↑	↓	↑	↓
	ADD.W @(d:32,ERs.L),@aa:32	W	7 6 1							S	D					@aa:32 + @(d:32+ERs<<1) → @aa:32				—	↓	↑	↓	↑	↓
	ADD.W @aa:16,@ERd	W	3 4 6							D	S					@ERd + @aa:16 → @ERd				—	↓	↑	↓	↑	↓
	ADD.W @aa:16,@ERd+	W	3 4 6							D	S					@ERd + @aa:16 → @ERd			ERd32+2→ERd32	—	↓	↑	↓	↑	↓
	ADD.W @aa:16,@ERd-	W	3 4 6							D	S					@ERd + @aa:16 → @ERd			ERd32-2→ERd32	—	↓	↑	↓	↑	↓
	ADD.W @aa:16,@+ERd	W	3 5 7							D	S			ERd32+2→ERd32		@ERd + @aa:16 → @ERd				—	↓	↑	↓	↑	↓
	ADD.W @aa:16,@-ERd	W	3 5 7							D	S			ERd32-2→ERd32		@ERd + @aa:16 → @ERd				—	↓	↑	↓	↑	↓
	ADD.W @aa:16,@(d:2,ERd)	W	3 5 7							D	S					@(2/4/6+ERd) + @aa:16 → @(2/4/6+ERd)				—	↓	↑	↓	↑	↓
	ADD.W @aa:16,@(d:16,ERd)	W	4 5 7							D	S					@(d:16+ERd) + @aa:16 → @(d:16+ERd)				—	↓	↑	↓	↑	↓
	ADD.W @aa:16,@(d:32,ERd)	W	5 5 8							D	S					@(d:32+ERd) + @aa:16 → @(d:32+ERd)				—	↓	↑	↓	↑	↓
	ADD.W @aa:16,@(d:16,Rd.B)	W	4 5 7							D	S					@(d:16+RdL<<1) + @aa:16 → @(d:16+RdL<<1)				—	↓	↑	↓	↑	↓
	ADD.W @aa:16,@(d:16,Rd.W)	W	4 5 7							D	S					@(d:16+Rd<<1) + @aa:16 → @(d:16+Rd<<1)				—	↓	↑	↓	↑	↓



Arithmetic operations (25)

Instruction	Mnemonic	Size	Instruction Length	Number of		Addressing Mode											Operation ^s					Condition Code ^{s2}																		
				Min.	Max.	16-Bit Instruction Fetch	Execution Status ¹	#xx	Rn	@ERN	@(d,ERn)	@(d,Rn,LRn,WERn,L)	@(d,ERn)/@ERN+/@ERN+/@ERN+/@+ERn	@ERn/@aa:16/@aa:32																										
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24													
ADD	ADD.W @aa:16, @(d:16,ERd,L)	W	4	5	7										D	S				@(d:16+ERd<<1) + @aa:16	→	@(d:16+ERd<<1)																		
	ADD.W @aa:16, @(d:32,Rd,B)	W	5	5	8										D	S				@(d:32+RdL<<1) + @aa:16	→	@(d:32+RdL<<1)																		
	ADD.W @aa:16, @(d:32,Rd,W)	W	5	5	8										D	S				@(d:32+Rd<<1) + @aa:16	→	@(d:32+Rd<<1)																		
	ADD.W @aa:16, @(d:32,ERd,L)	W	5	5	8										D	S				@(d:32+ERd<<1) + @aa:16	→	@(d:32+ERd<<1)																		
	ADD.W @aa:16, @aa:16	W	4	4	7											S			@aa:16	+	@aa:16	→	@aa:16																	
	ADD.W @aa:16, @aa:32	W	5	4	8											S			@aa:32	+	@aa:16	→	@aa:32																	
	ADD.W @aa:32, @ERd	W	4	4	7										D	S				@ERd	+	@aa:32	→	@ERd																
	ADD.W @aa:32, @ERd+	W	4	4	7										D	S				@ERd	+	@aa:32	→	@ERd			ERd32+2→ERd32													
	ADD.W @aa:32, @ERd-	W	4	4	7										D	S				@ERd	+	@aa:32	→	@ERd			ERd32-2→ERd32													
	ADD.W @aa:32, @+ERd	W	4	5	8										D	S				ERd32+2→ERd32	@ERd	+	@aa:32	→	@ERd															
	ADD.W @aa:32, @-ERd	W	4	5	8										D	S				ERd32-2→ERd32	@ERd	+	@aa:32	→	@ERd															
	ADD.W @aa:32, @(d:2,ERd)	W	4	5	8										D	S				@(2/4/6+ERd)	+	@aa:32	→	@(2/4/6+ERd)																
	ADD.W @aa:32, @(d:16,ERd)	W	5	5	8										D	S				@(d:16+ERd)	+	@aa:32	→	@(d:16+ERd)																
	ADD.W @aa:32, @(d:32,ERd)	W	6	5	9										D	S				@(d:32+ERd)	+	@aa:32	→	@(d:32+ERd)																
	ADD.W @aa:32, @(d:16,Rd,B)	W	5	5	8										D	S				@(d:16+RdL<<1) + @aa:32	→	@(d:16+RdL<<1)																		
	ADD.W @aa:32, @(d:16,Rd,W)	W	5	5	8										D	S				@(d:16+Rd<<1) + @aa:32	→	@(d:16+Rd<<1)																		
	ADD.W @aa:32, @(d:16,ERd,L)	W	5	5	8										D	S				@(d:16+ERd<<1) + @aa:32	→	@(d:16+ERd<<1)																		
	ADD.W @aa:32, @(d:32,Rd,B)	W	6	5	9										D	S				@(d:32+RdL<<1) + @aa:32	→	@(d:32+RdL<<1)																		
	ADD.W @aa:32, @(d:32,Rd,W)	W	6	5	9										D	S				@(d:32+Rd<<1) + @aa:32	→	@(d:32+Rd<<1)																		
	ADD.W @aa:32, @(d:32,ERd,L)	W	6	5	9										D	S				@(d:32+ERd<<1) + @aa:32	→	@(d:32+ERd<<1)																		
	ADD.W @aa:32, @aa:16	W	5	4	8											S			@aa:16	+	@aa:32	→	@aa:16																	
	ADD.W @aa:32, @aa:32	W	6	4	9											S			@aa:32	+	@aa:32	→	@aa:32																	
	ADD.L #xx:3,ERd	L	1	1	1	S	D													ERd	+	#xx:3	→	ERd																
	ADD.L #xx:16,ERd	L	2	1	2	S	D													ERd	+	#xx:16	→	ERd																
	ADD.L #xx:32,ERd	L	3	1	3	S	D													ERd	+	#xx	→	ERd																



Arithmetic operations (26)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode								Operation ⁵					Condition Code ^{6,7}																				
				Min. #xx	16-Bit Instruction Fetch Execution Status ⁸	Number of Fetch Execution Status ⁸	Rn	@(d,ERn)	@(d,Rn),W(ERn,L)	@(Rn)/@ERn//@ERn//@ERn//@ERn//@ERn	@Rn//@Rn//@Rn//@Rn//@Rn//@Rn	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C															
ADD	ADD.L #xx:16, @ERd	L	3	4	5	S		D													—	↓	↑	↓	↑	↓	↑										
	ADD.L #xx:16, @ERd+	L	3	4	5	S					D												ERd32+4→ERd32	—	↓	↑	↓	↑	↓	↑							
	ADD.L #xx:16, @ERd-	L	3	4	5	S						D												ERd32-4→ERd32	—	↓	↑	↓	↑	↓	↑						
	ADD.L #xx:16, @+ERd	L	3	5	5	S					D																										
	ADD.L #xx:16, @-ERd	L	3	5	5	S						D																									
	ADD.L #xx:16, @(d:2,ERd)	L	3	5	5	S						D																									
	ADD.L #xx:16, @(d:16,ERd)	L	4	5	6	S						D																									
	ADD.L #xx:16, @(d:32,ERd)	L	5	5	7	S						D																									
	ADD.L #xx:16, @(d:16,Rd,B)	L	4	5	6	S						D																									
	ADD.L #xx:16, @(d:16,Rd,W)	L	4	5	6	S						D																									
	ADD.L #xx:16, @(d:16,ERd,L)	L	4	5	6	S						D																									
	ADD.L #xx:16, @(d:32,Rd,B)	L	5	5	7	S						D																									
	ADD.L #xx:16, @(d:32,Rd,W)	L	5	5	7	S						D																									
	ADD.L #xx:16, @(d:32,ERd,L)	L	5	5	7	S						D																									
	ADD.L #xx:16, @aa:16	L	4	4	6	S							D																								
	ADD.L #xx:16, @aa:32	L	5	4	7	S							D																								
	ADD.L #xx:32, @ERd	L	4	4	6	S						D																									
	ADD.L #xx:32, @ERd+	L	4	4	6	S							D																								
	ADD.L #xx:32, @ERd-	L	4	4	6	S							D																								
	ADD.L #xx:32, @+ERd	L	4	5	6	S						D																									
	ADD.L #xx:32, @-ERd	L	4	5	6	S							D																								
	ADD.L #xx:32, @(d:2,ERd)	L	4	5	6	S							D																								
	ADD.L #xx:32, @(d:16,ERd)	L	5	5	7	S							D																								
	ADD.L #xx:32, @(d:32,ERd)	L	6	5	8	S							D																								
	ADD.L #xx:32, @(d:16,Rd,B)	L	5	5	7	S							D																								

Arithmetic operations (28)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Status ¹	Addressing Mode							Operation ⁵					Condition Codes ²																															
						#xx	Rn	@(d,ERn)	@(d,Rn,W,ERn,L)	@(d,ERn)/@ERn+/@ERn-/@+ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C																											
ADD	ADD.L @ERs+,ERd	L	2	3	3		D								ERd32	+	@ERs	→	ERd32	ERs32+4→ERs																													
	ADD.L @ERs-,ERd	L	2	3	3		D								ERd32	+	@ERs	→	ERd32	ERs32-4→ERs																													
	ADD.L @+ERs,ERd	L	2	4	4		D								ERs32+4→ERs32				ERd32	+	@ERs	→	ERd32																										
	ADD.L @-ERs,ERd	L	2	4	4		D								ERs32-4→ERs32				ERd32	+	@ERs	→	ERd32																										
	ADD.L @(d:2,ERs),ERd	L	2	4	4		D		S										ERd32	+	@(4/8/12+ERs)	→	ERd32																										
	ADD.L @(d:16,ERs),ERd	L	3	4	4		D		S										ERd32	+	@(d:16+ERs)	→	ERd32																										
	ADD.L @(d:32,ERs),ERd	L	4	4	5		D		S										ERd32	+	@(d:32+ERs)	→	ERd32																										
	ADD.L @(d:16,Rs,B),ERd	L	3	4	4		D		S										ERd32	+	@(d:16+Rs<<2)	→	ERd32																										
	ADD.L @(d:16,Rs,W),ERd	L	3	4	4		D		S										ERd32	+	@(d:16+Rs<<2)	→	ERd32																										
	ADD.L @(d:16,ERs,L),ERd	L	3	4	4		D		S										ERd32	+	@(d:16+ERs<<2)	→	ERd32																										
	ADD.L @(d:32,Rs,B),ERd	L	4	4	5		D		S										ERd32	+	@(d:32+Rs<<2)	→	ERd32																										
	ADD.L @(d:32,Rs,W),ERd	L	4	4	5		D		S										ERd32	+	@(d:32+Rs<<2)	→	ERd32																										
	ADD.L @(d:32,ERs,L),ERd	L	4	4	5		D		S										ERd32	+	@(d:32+ERs<<2)	→	ERd32																										
	ADD.L @aa:16,ERd	L	3	3	4		D					S							ERd32	+	@aa:16	→	ERd32																										
	ADD.L @aa:32,ERd	L	4	3	5		D					S							ERd32	+	@aa:32	→	ERd32																										
	ADD.L @ERs,@ERd	L	3	5	6			S											@ERd	+	@ERs	→	@ERd																										
	ADD.L @ERs,@ERd+	L	3	5	6			S		D									@ERd	+	@ERs	→	@ERd			ERd32+4→ERd32																							
	ADD.L @ERs,@ERd-	L	3	5	6			S		D									@ERd	+	@ERs	→	@ERd			ERd32-4→ERd32																							
	ADD.L @ERs,@+ERd	L	3	6	7			S		D						ERd32+4→ERd32			@ERd	+	@ERs	→	@ERd																										
	ADD.L @ERs,@-ERd	L	3	6	7			S		D						ERd32-4→ERd32			@ERd	+	@ERs	→	@ERd																										
	ADD.L @ERs,@(d:2,ERd)	L	3	6	7			S	D										@(4/8/12+ERd)	+	@ERs	→	@(4/8/12+ERd)																										
	ADD.L @ERs,@(d:16,ERd)	L	4	6	7			S	D										@(d:16+ERd)	+	@ERs	→	@(d:16+ERd)																										
	ADD.L @ERs,@(d:32,ERd)	L	5	6	8			S	D										@(d:32+ERd)	+	@ERs	→	@(d:32+ERd)																										
	ADD.L @ERs,@(d:16,Rd,B)	L	4	6	7			S	D										@(d:16+Rd<<2)	+	@ERs	→	@(d:16+Rd<<2)																										
	ADD.L @ERs,@(d:16,Rd,W)	L	4	6	7			S	D										@(d:16+Rd<<2)	+	@ERs	→	@(d:16+Rd<<2)																										

Arithmetic operations (30)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode								Operation ⁶					Condition Codes ⁵²															
				Min.	16-Bit Instruction Fetch Execution Status ⁶¹	Prx	Rn	@ERN	@(d, ERn)	@(d, ERn, W, ERn, L)	@(d, ERn, W, ERn, L)	@(d, ERn, W, ERn, L)	@(d, ERn, W, ERn, L)	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C								
ADD	ADD.L @ERs, @ERd-	L	3	5	6																											
	ADD.L @ERs, @+ERd	L	3	6	7																											
	ADD.L @ERs, @-ERd	L	3	6	7																											
	ADD.L @ERs, @(d:2, ERd)	L	3	6	7																											
	ADD.L @ERs, @(d:16, ERd)	L	4	6	7																											
	ADD.L @ERs, @(d:32, ERd)	L	5	6	8																											
	ADD.L @ERs, @(d:16, Rd.B)	L	4	6	7																											
	ADD.L @ERs, @(d:16, Rd.W)	L	4	6	7																											
	ADD.L @ERs, @(d:16, ERd.L)	L	4	6	7																											
	ADD.L @ERs, @(d:32, Rd.B)	L	5	6	8																											
	ADD.L @ERs, @(d:32, Rd.W)	L	5	6	8																											
	ADD.L @ERs, @(d:32, ERd.L)	L	5	6	8																											
	ADD.L @ERs, @aa:16	L	4	5	7																											
	ADD.L @ERs, @aa:32	L	5	5	8																											
	ADD.L @+ERs, @ERd	L	3	6	6																											
	ADD.L @+ERs, @ERd+	L	3	6	6																											
	ADD.L @+ERs, @ERd-	L	3	6	6																											
	ADD.L @+ERs, @+ERd	L	3	7	7																											
	ADD.L @+ERs, @-ERd	L	3	7	7																											
	ADD.L @+ERs, @(d:2, ERd)	L	3	7	7																											
	ADD.L @+ERs, @(d:16, ERd)	L	4	7	7																											
	ADD.L @+ERs, @(d:32, ERd)	L	5	7	8																											
	ADD.L @+ERs, @(d:16, Rd.B)	L	4	7	7																											
	ADD.L @+ERs, @(d:16, Rd.W)	L	4	7	7																											
	ADD.L @+ERs, @(d:16, ERd.L)	L	4	7	7																											

Arithmetic operations (31)

Instruction	Mnemonic	Size	Instruction Length	Min.	Addressing Mode									Operation ⁵					Condition Code ^{3,2}					
					16-Bit Instruction Fetch Execution Status ¹	px	Rn	@ERn	@(d,ERn)	@(d,Rn,ERn,WERn,L)	@ERn/@ERn+/@ERn+/@ERn+/@+ERn	@aa:8/@aa:16/@aa:32												
					Number of Execution Stages ¹										Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V
ADD	ADD.L @+ERs,@(d:32,Rd,B)	L	5 7 8							D	S				ERs32+4→ERs32		@(d:32+RdL<<2) + @ERs → @(d:32+RdL<<2)							
	ADD.L @+ERs,@(d:32,Rd,W)	L	5 7 8							D	S				ERs32+4→ERs32		@(d:32+Rd<<2) + @ERs → @(d:32+Rd<<2)							
	ADD.L @+ERs,@(d:32,ERd,L)	L	5 7 8							D	S				ERs32+4→ERs32		@(d:32+ERd<<2) + @ERs → @(d:32+ERd<<2)							
	ADD.L @+ERs,@aa:16	L	4 6 7								S	D			ERs32+4→ERs32		@aa:16 + @ERs → @aa:16							
	ADD.L @+ERs,@aa:32	L	5 6 8								S	D			ERs32+4→ERs32		@aa:32 + @ERs → @aa:32							
	ADD.L @-ERs,@ERd	L	3 6 6							D	S				ERs32-4→ERs32		@ERd + @ERs → @ERd							
	ADD.L @-ERs,@ERd+	L	3 6 6								S				ERs32-4→ERs32		@ERd + @ERs → @ERd					ERd32+4→ERd32		
	ADD.L @-ERs,@ERd-	L	3 6 6								S				ERs32-4→ERs32		@ERd + @ERs → @ERd					ERd32-4→ERd32		
	ADD.L @-ERs,@+ERd	L	3 7 7								S				ERs32-4→ERs32	ERd32+4→ERd32	@ERd + @ERs → @ERd							
	ADD.L @-ERs,@-ERd	L	3 7 7								S				ERs32-4→ERs32	ERd32-4→ERd32	@ERd + @ERs → @ERd							
	ADD.L @-ERs,@(d:2,ERd)	L	3 7 7							D	S				ERs32-4→ERs32		@(4/8/12+ERd) + @ERs → @(4/8/12+ERd)							
	ADD.L @-ERs,@(d:16,ERd)	L	4 7 7							D	S				ERs32-4→ERs32		@(d:16+ERd) + @ERs → @(d:16+ERd)							
	ADD.L @-ERs,@(d:32,ERd)	L	5 7 8							D	S				ERs32-4→ERs32		@(d:32+ERd) + @ERs → @(d:32+ERd)							
	ADD.L @-ERs,@(d:16,Rd,B)	L	4 7 7							D	S				ERs32-4→ERs32		@(d:16+RdL<<2) + @ERs → @(d:16+RdL<<2)							
	ADD.L @-ERs,@(d:16,Rd,W)	L	4 7 7							D	S				ERs32-4→ERs32		@(d:16+Rd<<2) + @ERs → @(d:16+Rd<<2)							
	ADD.L @-ERs,@(d:16,ERd,L)	L	4 7 7							D	S				ERs32-4→ERs32		@(d:16+ERd<<2) + @ERs → @(d:16+ERd<<2)							
	ADD.L @-ERs,@(d:32,Rd,B)	L	5 7 8							D	S				ERs32-4→ERs32		@(d:32+RdL<<2) + @ERs → @(d:32+RdL<<2)							
	ADD.L @-ERs,@(d:32,Rd,W)	L	5 7 8							D	S				ERs32-4→ERs32		@(d:32+Rd<<2) + @ERs → @(d:32+Rd<<2)							
	ADD.L @-ERs,@(d:32,ERd,L)	L	5 7 8							D	S				ERs32-4→ERs32		@(d:32+ERd<<2) + @ERs → @(d:32+ERd<<2)							
	ADD.L @-ERs,@aa:16	L	4 6 7								S	D			ERs32-4→ERs32		@aa:16 + @ERs → @aa:16							
	ADD.L @-ERs,@aa:32	L	5 6 8								S	D			ERs32-4→ERs32		@aa:32 + @ERs → @aa:32							
	ADD.L @(d:2,ERs),@ERd	L	3 6 6							D	S					@ERd + @ERs → @ERd								
	ADD.L @(d:2,ERs),@ERd+	L	3 6 6								S	D				@ERd + @ERs → @ERd						ERd32+4→ERd32		
	ADD.L @(d:2,ERs),@ERd-	L	3 6 6								S	D				@ERd + @ERs → @ERd						ERd32-4→ERd32		
	ADD.L @(d:2,ERs),@+ERd	L	3 7 7								S	D				ERd32+4→ERd32	@ERd + @ERs → @ERd							

Arithmetic operations (35)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode										Operation ⁵					Condition Codes ^{6,7}																					
				Min.	16-Bit Instruction Fetch Execution Status ⁸	16-Bit Instruction Fetch Execution Status ⁸	16-Bit Instruction Fetch Execution Status ⁸	16-Bit Instruction Fetch Execution Status ⁸	16-Bit Instruction Fetch Execution Status ⁸	16-Bit Instruction Fetch Execution Status ⁸	16-Bit Instruction Fetch Execution Status ⁸	16-Bit Instruction Fetch Execution Status ⁸	16-Bit Instruction Fetch Execution Status ⁸												16-Bit Instruction Fetch Execution Status ⁸															
ADD	ADD.L @(d:16, Rs.W), @aa:32	L	6	6	9					S	D																													
	ADD.L @(d:16, ERs.L), @ERd	L	4	6	7					D	S																													
	ADD.L @(d:16, ERs.L), @ERd+	L	4	6	7						S	D																												
	ADD.L @(d:16, ERs.L), @ERd-	L	4	6	7						S	D																												
	ADD.L @(d:16, ERs.L), @+ERd	L	4	7	8						S	D					ERd32+4	→	ERd32																					
	ADD.L @(d:16, ERs.L), @-ERd	L	4	7	8						S	D					ERd32-4	→	ERd32																					
	ADD.L @(d:16, ERs.L), @(d:2, ERd)	L	4	7	8						D	S																												
	ADD.L @(d:16, ERs.L), @(d:16, ERd)	L	5	7	8						D	S																												
	ADD.L @(d:16, ERs.L), @(d:32, ERd)	L	6	7	9						D	S																												
	ADD.L @(d:16, ERs.L), @(d:16, Rd.B)	L	5	7	8						S																													
	ADD.L @(d:16, ERs.L), @(d:16, Rd.W)	L	5	7	8						S																													
	ADD.L @(d:16, ERs.L), @(d:16, ERd.L)	L	5	7	8						S																													
	ADD.L @(d:16, ERs.L), @(d:32, Rd.B)	L	6	7	9						S																													
	ADD.L @(d:16, ERs.L), @(d:32, Rd.W)	L	6	7	9						S																													
	ADD.L @(d:16, ERs.L), @(d:32, ERd.L)	L	6	7	9						S																													
	ADD.L @(d:16, ERs.L), @aa:16	L	5	6	8						S	D																												
	ADD.L @(d:16, ERs.L), @aa:32	L	6	6	9						S	D																												
	ADD.L @(d:32, Rs.B), @ERd	L	5	6	8						D	S																												
	ADD.L @(d:32, Rs.B), @ERd+	L	5	6	8						S	D																												
	ADD.L @(d:32, Rs.B), @ERd-	L	5	6	8						S	D																												
ADD.L @(d:32, Rs.B), @+ERd	L	5	7	9						S	D					ERd32+4	→	ERd32																						
ADD.L @(d:32, Rs.B), @-ERd	L	5	7	9						S	D					ERd32-4	→	ERd32																						
ADD.L @(d:32, Rs.B), @(d:2, ERd)	L	5	7	9						D	S																													
ADD.L @(d:32, Rs.B), @(d:16, ERd)	L	6	7	9						D	S																													
ADD.L @(d:32, Rs.B), @(d:32, ERd)	L	7	7	1						D	S																													

Arithmetic operations (36)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages¹	Addressing Mode											Operation ⁵					Condition Codes ^{2,3}								
						Number of Execution Stages ⁴	#rx	Rn	@ERN	@(d,ERN)	@(d,Rn,ERN,WERN,L)	@ERN/@ERN+/@ERN+/@ERN+/@+ERN	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C						
ADD	ADD.L @(d:32,Rs,B),@(d:16,Rd,B)	L	6	7	9					S															—	↓	↓	↓	↓	↓
ADD.L	ADD.L @(d:32,Rs,B),@(d:16,Rd,W)	L	6	7	9					S															—	↓	↓	↓	↓	↓
ADD.L	ADD.L @(d:32,Rs,B),@(d:16,ERd,L)	L	6	7	9					S															—	↓	↓	↓	↓	↓
ADD.L	ADD.L @(d:32,Rs,B),@(d:32,Rd,B)	L	7	7	1					S															—	↓	↓	↓	↓	↓
ADD.L	ADD.L @(d:32,Rs,B),@(d:32,Rd,W)	L	7	7	1					S															—	↓	↓	↓	↓	↓
ADD.L	ADD.L @(d:32,Rs,B),@(d:32,ERd,L)	L	7	7	1					S															—	↓	↓	↓	↓	↓
ADD.L	ADD.L @(d:32,Rs,B),@aa:16	L	6	6	9					S	D														—	↓	↓	↓	↓	↓
ADD.L	ADD.L @(d:32,Rs,B),@aa:32	L	7	6	1					S	D														—	↓	↓	↓	↓	↓
ADD.L	ADD.L @(d:32,Rs,W),@ERN	L	5	6	8				D	S															—	↓	↓	↓	↓	↓
ADD.L	ADD.L @(d:32,Rs,W),@ERd+	L	5	6	8					S	D												ERd32+4→ERd32	—	↓	↓	↓	↓	↓	
ADD.L	ADD.L @(d:32,Rs,W),@ERd-	L	5	6	8					S	D												ERd32-4→ERd32	—	↓	↓	↓	↓	↓	
ADD.L	ADD.L @(d:32,Rs,W),@+ERd	L	5	7	9					S	D		ERd32+4→ERd32	@ERd	+	@(d:32+Rs<<2)	→	@ERd							—	↓	↓	↓	↓	↓
ADD.L	ADD.L @(d:32,Rs,W),@-ERd	L	5	7	9					S	D		ERd32-4→ERd32	@ERd	+	@(d:32+Rs<<2)	→	@ERd							—	↓	↓	↓	↓	↓
ADD.L	ADD.L @(d:32,Rs,W),@(d:2,ERd)	L	5	7	9					D	S							@(4/8/12+ERd)	+	@(d:32+Rs<<2)	→	@(4/8/12+ERd)				—	↓	↓	↓	↓
ADD.L	ADD.L @(d:32,Rs,W),@(d:16,ERd)	L	6	7	9					D	S							@(d:16+ERd)	+	@(d:32+Rs<<2)	→	@(d:16+ERd)				—	↓	↓	↓	↓
ADD.L	ADD.L @(d:32,Rs,W),@(d:32,ERd)	L	7	7	1					D	S							@(d:32+ERd)	+	@(d:32+Rs<<2)	→	@(d:32+ERd)				—	↓	↓	↓	↓
ADD.L	ADD.L @(d:32,Rs,W),@(d:16,Rd,B)	L	6	7	9					S								@(d:16+Rd<<2)	+	@(d:32+Rs<<2)	→	@(d:16+Rd<<2)				—	↓	↓	↓	↓
ADD.L	ADD.L @(d:32,Rs,W),@(d:16,Rd,W)	L	6	7	9					S								@(d:16+Rd<<2)	+	@(d:32+Rs<<2)	→	@(d:16+Rd<<2)				—	↓	↓	↓	↓
ADD.L	ADD.L @(d:32,Rs,W),@(d:16,ERd,L)	L	6	7	9					S								@(d:16+ERd<<2)	+	@(d:32+Rs<<2)	→	@(d:16+ERd<<2)				—	↓	↓	↓	↓
ADD.L	ADD.L @(d:32,Rs,W),@(d:32,Rd,B)	L	7	7	1					S								@(d:32+RdL<<2)	+	@(d:32+Rs<<2)	→	@(d:32+RdL<<2)				—	↓	↓	↓	↓
ADD.L	ADD.L @(d:32,Rs,W),@(d:32,Rd,W)	L	7	7	1					S								@(d:32+Rd<<2)	+	@(d:32+Rs<<2)	→	@(d:32+Rd<<2)				—	↓	↓	↓	↓
ADD.L	ADD.L @(d:32,Rs,W),@(d:32,ERd,L)	L	7	7	1					S								@(d:32+ERd<<2)	+	@(d:32+Rs<<2)	→	@(d:32+ERd<<2)				—	↓	↓	↓	↓
ADD.L	ADD.L @(d:32,Rs,W),@aa:16	L	6	6	9					S	D							@aa:16	+	@(d:32+Rs<<2)	→	@aa:16				—	↓	↓	↓	↓
ADD.L	ADD.L @(d:32,Rs,W),@aa:32	L	7	6	1					S	D							@aa:32	+	@(d:32+Rs<<2)	→	@aa:32				—	↓	↓	↓	↓



Arithmetic operations (38)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹	Addressing Mode								Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	Condition Codes ²									
						#xx	Rn	@ERN	@(d,ERN)	@(d,Rn,WRn,L)	@ERN/@ERN+/@ERN-/@+ERn	@aa:8/@aa:16/@aa:32	Operation 1						Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C
ADD	ADD.L @aa:16,@(d:16,Rd,W)	L	5	6	8						D	S			@(d:16+Rd<<2) + @aa:16 → @(d:16+Rd<<2)							—	↑	↓	↑	↓	↑	↓
	ADD.L @aa:16,@(d:16,ERd,L)	L	5	6	8						D	S			@(d:16+ERd<<2) + @aa:16 → @(d:16+ERd<<2)							—	↑	↓	↑	↓	↑	↓
	ADD.L @aa:16,@(d:32,Rd,B)	L	6	6	9						D	S			@(d:32+RdL<<2) + @aa:16 → @(d:32+RdL<<2)							—	↑	↓	↑	↓	↑	↓
	ADD.L @aa:16,@(d:32,Rd,W)	L	6	6	9						D	S			@(d:32+Rd<<2) + @aa:16 → @(d:32+Rd<<2)							—	↑	↓	↑	↓	↑	↓
	ADD.L @aa:16,@(d:32,ERd,L)	L	6	6	9						D	S			@(d:32+ERd<<2) + @aa:16 → @(d:32+ERd<<2)							—	↑	↓	↑	↓	↑	↓
	ADD.L @aa:16,@aa:16	L	5	5	8								S		@aa:16 + @aa:16 → @aa:16							—	↑	↓	↑	↓	↑	↓
	ADD.L @aa:16,@aa:32	L	6	5	9								S		@aa:32 + @aa:16 → @aa:32							—	↑	↓	↑	↓	↑	↓
	ADD.L @aa:32,@ERd	L	5	5	8							D	S		@ERd + @aa:32 → @ERd							—	↑	↓	↑	↓	↑	↓
	ADD.L @aa:32,@ERd+	L	5	5	8							D	S		@ERd + @aa:32 → @ERd					ERd32+4→ERd32		—	↑	↓	↑	↓	↑	↓
	ADD.L @aa:32,@ERd-	L	5	5	8							D	S		@ERd + @aa:32 → @ERd					ERd32-4→ERd32		—	↑	↓	↑	↓	↑	↓
	ADD.L @aa:32,@+ERd	L	5	6	9							D	S		ERd32+4→ERd32	@ERd + @aa:32 → @ERd						—	↑	↓	↑	↓	↑	↓
	ADD.L @aa:32,@-ERd	L	5	6	9							D	S		ERd32-4→ERd32	@ERd + @aa:32 → @ERd						—	↑	↓	↑	↓	↑	↓
	ADD.L @aa:32,@(d:2,ERd)	L	5	6	9							D	S			@(4/8/12+ERd) + @aa:32 → @(4/8/12+ERd)						—	↑	↓	↑	↓	↑	↓
	ADD.L @aa:32,@(d:16,ERd)	L	6	6	9							D	S			@(d:16+ERd) + @aa:32 → @(d:16+ERd)						—	↑	↓	↑	↓	↑	↓
	ADD.L @aa:32,@(d:32,ERd)	L	7	6	1							D	S			@(d:32+ERd) + @aa:32 → @(d:32+ERd)						—	↑	↓	↑	↓	↑	↓
	ADD.L @aa:32,@(d:16,Rd,B)	L	6	6	9							D	S			@(d:16+RdL<<2) + @aa:32 → @(d:16+RdL<<2)						—	↑	↓	↑	↓	↑	↓
	ADD.L @aa:32,@(d:16,Rd,W)	L	6	6	9							D	S			@(d:16+Rd<<2) + @aa:32 → @(d:16+Rd<<2)						—	↑	↓	↑	↓	↑	↓
	ADD.L @aa:32,@(d:16,ERd,L)	L	6	6	9							D	S			@(d:16+ERd<<2) + @aa:32 → @(d:16+ERd<<2)						—	↑	↓	↑	↓	↑	↓
	ADD.L @aa:32,@(d:32,Rd,B)	L	7	6	1							D	S			@(d:32+RdL<<2) + @aa:32 → @(d:32+RdL<<2)						—	↑	↓	↑	↓	↑	↓
	ADD.L @aa:32,@(d:32,Rd,W)	L	7	6	1							D	S			@(d:32+Rd<<2) + @aa:32 → @(d:32+Rd<<2)						—	↑	↓	↑	↓	↑	↓
	ADD.L @aa:32,@(d:32,ERd,L)	L	7	6	1							D	S			@(d:32+ERd<<2) + @aa:32 → @(d:32+ERd<<2)						—	↑	↓	↑	↓	↑	↓
ADD.L @aa:32,@aa:16	L	6	5	9								S		@aa:16 + @aa:32 → @aa:16							—	↑	↓	↑	↓	↑	↓	
ADD.L @aa:32,@aa:32	L	7	5	1								S		@aa:32 + @aa:32 → @aa:32							—	↑	↓	↑	↓	↑	↓	
CMP	CMP.B #xx:8,Rd	B	1	1	1	S	D							Rd8 - #xx							—	↑	↓	↑	↓	↑	↓	
	CMP.B #xx:8,@ERd	B	2	3	3	S	D							@ERd - #xx							—	↑	↓	↑	↓	↑	↓	

Arithmetic operations (39)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	16-Bit Instruction Fetch Execution Status ¹	Addressing Mode							Operation ⁵					Condition Code ^{2,3}								
							16-Bit Instruction Fetch Execution Status ¹	#xx	Rn	@(d, ERn)	@(d, Rn.L, ERn.W, ERn.L)	@(ERn)/@ERn+/@ERn-/@ERn+/@+ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C			
CMP	CMP.B #xx:8, @ERd+	B	3	4	4	S					D				@ERd	-	#xx				ERd32+1→ERd32	—	↓	↓	↓	↓	↓
	CMP.B #xx:8, @ERd-	B	3	4	4	S					D				@ERd	-	#xx				ERd32-1→ERd32	—	↓	↓	↓	↓	↓
	CMP.B #xx:8, @+ERd	B	3	5	5	S					D			ERd32+1→ERd32	@ERd	-	#xx					—	↓	↓	↓	↓	↓
	CMP.B #xx:8, @-ERd	B	3	5	5	S					D			ERd32-1→ERd32	@ERd	-	#xx					—	↓	↓	↓	↓	↓
	CMP.B #xx:8, @(d:2, ERd)	B	3	5	5	S					D				@(1/2/3+ERd)	-	#xx					—	↓	↓	↓	↓	↓
	CMP.B #xx:8, @(d:16, ERd)	B	4	5	6	S					D				@(d:16+ERd)	-	#xx					—	↓	↓	↓	↓	↓
	CMP.B #xx:8, @(d:32, ERd)	B	5	5	7	S					D				@(d:32+ERd)	-	#xx					—	↓	↓	↓	↓	↓
	CMP.B #xx:8, @(d:16, Rd.B)	B	4	5	6	S					D				@(d:16+RdL)	-	#xx					—	↓	↓	↓	↓	↓
	CMP.B #xx:8, @(d:16, Rd.W)	B	4	5	6	S					D				@(d:16+Rd)	-	#xx					—	↓	↓	↓	↓	↓
	CMP.B #xx:8, @(d:16, ERd.L)	B	4	5	6	S					D				@(d:16+ERd)	-	#xx					—	↓	↓	↓	↓	↓
	CMP.B #xx:8, @(d:32, Rd.B)	B	5	5	7	S					D				@(d:32+RdL)	-	#xx					—	↓	↓	↓	↓	↓
	CMP.B #xx:8, @(d:32, Rd.W)	B	5	5	7	S					D				@(d:32+Rd)	-	#xx					—	↓	↓	↓	↓	↓
	CMP.B #xx:8, @(d:32, ERd.L)	B	5	5	7	S					D				@(d:32+ERd)	-	#xx					—	↓	↓	↓	↓	↓
	CMP.B #xx:8, @aa:8	B	2	3	3	S						D			@aa:8	-	#xx					—	↓	↓	↓	↓	↓
	CMP.B #xx:8, @aa:16	B	3	3	4	S						D			@aa:16	-	#xx					—	↓	↓	↓	↓	↓
	CMP.B #xx:8, @aa:32	B	4	3	5	S						D			@aa:32	-	#xx					—	↓	↓	↓	↓	↓
	CMP.B Rs, Rd	B	1	1	1	S									Rd8	-	Rs8					—	↓	↓	↓	↓	↓
	CMP.B Rs, @ERd	B	2	3	3	S					D				@ERd	-	Rs8					—	↓	↓	↓	↓	↓
	CMP.B Rs, @ERd+	B	2	3	3	S						D			@ERd	-	Rs8				ERd32+1→ERd32	—	↓	↓	↓	↓	↓
	CMP.B Rs, @ERd-	B	2	3	3	S						D			@ERd	-	Rs8				ERd32-1→ERd32	—	↓	↓	↓	↓	↓
CMP.B Rs, @+ERd	B	2	4	4	S						D			ERd32+1→ERd32	@ERd	-	Rs8				—	↓	↓	↓	↓	↓	
CMP.B Rs, @-ERd	B	2	4	4	S						D			ERd32-1→ERd32	@ERd	-	Rs8				—	↓	↓	↓	↓	↓	
CMP.B Rs, @(d:2, ERd)	B	2	4	4	S						D			@(1/2/3+ERd)	-	Rs8					—	↓	↓	↓	↓	↓	
CMP.B Rs, @(d:16, ERd)	B	3	4	4	S						D			@(d:16+ERd)	-	Rs8					—	↓	↓	↓	↓	↓	
CMP.B Rs, @(d:32, ERd)	B	4	4	5	S						D			@(d:32+ERd)	-	Rs8					—	↓	↓	↓	↓	↓	

Arithmetic operations (42)

Instruction n	Mnemonic	Size	Instruction Length	Addressing Mode								Operation ⁶					Condition Codes ^{8,9}						
				Min.	16-Bit Instruction Fetch	Number of Execution Stages ⁷ #xx	Rn	Rm	@ERn	@ERn @ERn.W/ERn.L	@ERn/ @ERn/ERn+/@ERn+/@+ERn	@aa:16/ @aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C
CMP	CMP.B @ERs+, @(d:16,Rd.B)	B	4	5	6				D	S				@(d:16+RdL) - @ERs	ERs32+1→ERs32		—	↓	↓	↓	↓	↓	
	CMP.B @ERs+, @(d:16,Rd.W)	B	4	5	6				D	S				@(d:16+Rd) - @ERs	ERs32+1→ERs32		—	↓	↓	↓	↓	↓	
	CMP.B @ERs+, @(d:16,ERd.L)	B	4	5	6				D	S				@(d:16+ERd) - @ERs	ERs32+1→ERs32		—	↓	↓	↓	↓	↓	
	CMP.B @ERs+, @(d:32,Rd.B)	B	5	5	7				D	S				@(d:32+RdL) - @ERs	ERs32+1→ERs32		—	↓	↓	↓	↓	↓	
	CMP.B @ERs+, @(d:32,Rd.W)	B	5	5	7				D	S				@(d:32+Rd) - @ERs	ERs32+1→ERs32		—	↓	↓	↓	↓	↓	
	CMP.B @ERs+, @(d:32,ERd.L)	B	5	5	7				D	S				@(d:32+ERd) - @ERs	ERs32+1→ERs32		—	↓	↓	↓	↓	↓	
	CMP.B @ERs+, @aa:16	B	4	4	6					S	D			@aa:16 - @ERs	ERs32+1→ERs32		—	↓	↓	↓	↓	↓	
	CMP.B @ERs+, @aa:32	B	5	4	7					S	D			@aa:32 - @ERs	ERs32+1→ERs32		—	↓	↓	↓	↓	↓	
	CMP.B @ERs-, @ERd	B	3	4	5				D		S			@ERd - @ERs	ERs32-1→ERs32		—	↓	↓	↓	↓	↓	
	CMP.B @ERs-, @ERd+	B	3	4	5						S			@ERd - @ERs	ERs32-1→ERs32	ERd32+1→ERd32	—	↓	↓	↓	↓	↓	
	CMP.B @ERs-, @ERd-	B	3	4	5						S			@ERd - @ERs	ERs32-1→ERs32	ERd32-1→ERd32	—	↓	↓	↓	↓	↓	
	CMP.B @ERs-, @+ERd	B	3	5	6						S		ERd32+1→ERd32	@ERd - @ERs	ERs32-1→ERs32		—	↓	↓	↓	↓	↓	
	CMP.B @ERs-, @-ERd	B	3	5	6						S		ERd32-1→ERd32	@ERd - @ERs	ERs32-1→ERs32		—	↓	↓	↓	↓	↓	
	CMP.B @ERs-, @(d:2,ERd)	B	3	5	6				D	S				@(1/2/3+ERd) - @ERs	ERs32-1→ERs32		—	↓	↓	↓	↓	↓	
	CMP.B @ERs-, @(d:16,ERd)	B	4	5	6				D	S				@(d:16+ERd) - @ERs	ERs32-1→ERs32		—	↓	↓	↓	↓	↓	
	CMP.B @ERs-, @(d:32,ERd)	B	5	5	7				D	S				@(d:32+ERd) - @ERs	ERs32-1→ERs32		—	↓	↓	↓	↓	↓	
	CMP.B @ERs-, @(d:16,Rd.B)	B	4	5	6					D	S			@(d:16+RdL) - @ERs	ERs32-1→ERs32		—	↓	↓	↓	↓	↓	
	CMP.B @ERs-, @(d:16,Rd.W)	B	4	5	6					D	S			@(d:16+Rd) - @ERs	ERs32-1→ERs32		—	↓	↓	↓	↓	↓	
	CMP.B @ERs-, @(d:16,ERd.L)	B	4	5	6					D	S			@(d:16+ERd) - @ERs	ERs32-1→ERs32		—	↓	↓	↓	↓	↓	
	CMP.B @ERs-, @(d:32,Rd.B)	B	5	5	7					D	S			@(d:32+RdL) - @ERs	ERs32-1→ERs32		—	↓	↓	↓	↓	↓	
	CMP.B @ERs-, @(d:32,Rd.W)	B	5	5	7					D	S			@(d:32+Rd) - @ERs	ERs32-1→ERs32		—	↓	↓	↓	↓	↓	
	CMP.B @ERs-, @(d:32,ERd.L)	B	5	5	7					D	S			@(d:32+ERd) - @ERs	ERs32-1→ERs32		—	↓	↓	↓	↓	↓	
	CMP.B @ERs-, @aa:16	B	4	4	6						S	D		@aa:16 - @ERs	ERs32-1→ERs32		—	↓	↓	↓	↓	↓	
CMP.B @ERs-, @aa:32	B	5	4	7						S	D		@aa:32 - @ERs	ERs32-1→ERs32		—	↓	↓	↓	↓	↓		
CMP.B @+ERs, @ERd	B	3	5	5				D		S		ERs32+1→ERs32	@ERd - @ERs			—	↓	↓	↓	↓	↓		

Arithmetic operations (43)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹ #xx	Addressing Mode						Operation ⁵					Condition Codes ²										
						Rn	@ERn	@ERn	@ERn.W/ERn.L	@ERn/@ERn+/@ERn+/@ERn	@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C					
																							Number of Execution Stages ¹	Operation 1	Operation 2	Operation 3	Operation 4
CMP	CMP.B @+ERs,@ERd+	B	3	5	5						S	ERs32+1→ERs32		@ERd	-	@ERs			ERd32+1→ERd32	—	↑	↓	↑	↓	↑	↓	
	CMP.B @+ERs,@ERd-	B	3	5	5						S	ERs32+1→ERs32		@ERd	-	@ERs			ERd32-1→ERd32	—	↑	↓	↑	↓	↑	↓	
	CMP.B @+ERs,@+ERd	B	3	6	6							ERs32+1→ERs32	ERd32+1→ERd32	@ERd	-	@ERs				—	↑	↓	↑	↓	↑	↓	
	CMP.B @+ERs,@-ERd	B	3	6	6							ERs32+1→ERs32	ERd32-1→ERd32	@ERd	-	@ERs				—	↑	↓	↑	↓	↑	↓	
	CMP.B @+ERs,@(d:2,ERd)	B	3	6	6					D	S	ERs32+1→ERs32		@(1/2/3+ERd)	-	@ERs				—	↑	↓	↑	↓	↑	↓	
	CMP.B @+ERs,@(d:16,ERd)	B	4	6	6					D	S	ERs32+1→ERs32		@(d:16+ERd)	-	@ERs				—	↑	↓	↑	↓	↑	↓	
	CMP.B @+ERs,@(d:32,ERd)	B	5	6	7					D	S	ERs32+1→ERs32		@(d:32+ERd)	-	@ERs				—	↑	↓	↑	↓	↑	↓	
	CMP.B @+ERs,@(d:16,Rd,B)	B	4	6	6					D	S	ERs32+1→ERs32		@(d:16+RdL)	-	@ERs				—	↑	↓	↑	↓	↑	↓	
	CMP.B @+ERs,@(d:16,Rd,W)	B	4	6	6					D	S	ERs32+1→ERs32		@(d:16+Rd)	-	@ERs				—	↑	↓	↑	↓	↑	↓	
	CMP.B @+ERs,@(d:16,ERd,L)	B	4	6	6					D	S	ERs32+1→ERs32		@(d:16+ERd)	-	@ERs				—	↑	↓	↑	↓	↑	↓	
	CMP.B @+ERs,@(d:32,Rd,B)	B	5	6	7					D	S	ERs32+1→ERs32		@(d:32+RdL)	-	@ERs				—	↑	↓	↑	↓	↑	↓	
	CMP.B @+ERs,@(d:32,Rd,W)	B	5	6	7					D	S	ERs32+1→ERs32		@(d:32+Rd)	-	@ERs				—	↑	↓	↑	↓	↑	↓	
	CMP.B @+ERs,@(d:32,ERd,L)	B	5	6	7					D	S	ERs32+1→ERs32		@(d:32+ERd)	-	@ERs				—	↑	↓	↑	↓	↑	↓	
	CMP.B @+ERs,@aa:16	B	4	5	6						S	D	ERs32+1→ERs32		@aa:16	-	@ERs				—	↑	↓	↑	↓	↑	↓
	CMP.B @+ERs,@aa:32	B	5	5	7						S	D	ERs32+1→ERs32		@aa:32	-	@ERs				—	↑	↓	↑	↓	↑	↓
	CMP.B @-ERs,@ERd	B	3	5	5				D		S	ERs32-1→ERs32		@ERd	-	@ERs				—	↑	↓	↑	↓	↑	↓	
	CMP.B @-ERs,@ERd+	B	3	5	5						S		ERs32-1→ERs32		@ERd	-	@ERs			ERd32+1→ERd32	—	↑	↓	↑	↓	↑	↓
	CMP.B @-ERs,@ERd-	B	3	5	5						S		ERs32-1→ERs32		@ERd	-	@ERs			ERd32-1→ERd32	—	↑	↓	↑	↓	↑	↓
	CMP.B @-ERs,@+ERd	B	3	6	6							S	ERs32-1→ERs32	ERd32+1→ERd32	@ERd	-	@ERs				—	↑	↓	↑	↓	↑	↓
	CMP.B @-ERs,@-ERd	B	3	6	6							S	ERs32-1→ERs32	ERd32-1→ERd32	@ERd	-	@ERs				—	↑	↓	↑	↓	↑	↓
CMP.B @-ERs,@(d:2,ERd)	B	3	6	6					D	S	ERs32-1→ERs32		@(1/2/3+ERd)	-	@ERs				—	↑	↓	↑	↓	↑	↓		
CMP.B @-ERs,@(d:16,ERd)	B	4	6	6					D	S	ERs32-1→ERs32		@(d:16+ERd)	-	@ERs				—	↑	↓	↑	↓	↑	↓		
CMP.B @-ERs,@(d:32,ERd)	B	5	6	7					D	S	ERs32-1→ERs32		@(d:32+ERd)	-	@ERs				—	↑	↓	↑	↓	↑	↓		
CMP.B @-ERs,@(d:16,Rd,B)	B	4	6	6					D	S	ERs32-1→ERs32		@(d:16+RdL)	-	@ERs				—	↑	↓	↑	↓	↑	↓		

Arithmetic operations (45)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	Addressing Mode							Operation ⁵					Condition Code ^{6,7}											
						Number of 16-Bit Instruction Fetch Execution Stages ¹	Fetch # ²	Rn	@ERn	@(d,ERn)	@(d,Rn,WRn,L)	@ERn/@ERn+/@ERn+/@ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C					
CMP	CMP.B @(d:16,ERs),@ERd-	B	4	5	6				S	D					@ERd	-	@(d:16+ERs)			ERd32-1→ERd32	---	↑	↑	↑	↑	↑	↑		
	CMP.B @(d:16,ERs),@+ERd	B	4	6	7				S	D					ERd32+1→ERd32	@ERd	-	@(d:16+ERs)				---	↑	↑	↑	↑	↑	↑	
	CMP.B @(d:16,ERs),@-ERd	B	4	6	7				S	D					ERd32-1→ERd32	@ERd	-	@(d:16+ERs)				---	↑	↑	↑	↑	↑	↑	
	CMP.B @(d:16,ERs),@(d:2,ERd)	B	4	6	7				S							@(1/2/3+ERd)	-	@(d:16+ERs)				---	↑	↑	↑	↑	↑	↑	
	CMP.B @(d:16,ERs),@(d:16,ERd)	B	5	6	7				S							@(d:16+ERd)	-	@(d:16+ERs)				---	↑	↑	↑	↑	↑	↑	
	CMP.B @(d:16,ERs),@(d:32,ERd)	B	6	6	8				S							@(d:32+ERd)	-	@(d:16+ERs)				---	↑	↑	↑	↑	↑	↑	
	CMP.B @(d:16,ERs),@(d:16,Rd,B)	B	5	6	7				S	D						@(d:16+RdL)	-	@(d:16+ERs)				---	↑	↑	↑	↑	↑	↑	
	CMP.B @(d:16,ERs),@(d:16,Rd,W)	B	5	6	7				S	D						@(d:16+Rd)	-	@(d:16+ERs)				---	↑	↑	↑	↑	↑	↑	
	CMP.B @(d:16,ERs),@(d:16,ERdL)	B	5	6	7				S	D						@(d:16+ERd)	-	@(d:16+ERs)				---	↑	↑	↑	↑	↑	↑	
	CMP.B @(d:16,ERs),@(d:32,Rd,B)	B	6	6	8				S	D						@(d:32+RdL)	-	@(d:16+ERs)				---	↑	↑	↑	↑	↑	↑	
	CMP.B @(d:16,ERs),@(d:32,Rd,W)	B	6	6	8				S	D						@(d:32+Rd)	-	@(d:16+ERs)				---	↑	↑	↑	↑	↑	↑	
	CMP.B @(d:16,ERs),@(d:32,ERdL)	B	6	6	8				S	D						@(d:32+ERd)	-	@(d:16+ERs)				---	↑	↑	↑	↑	↑	↑	
	CMP.B @(d:16,ERs),@aa:16	B	5	5	7				S		D					@aa:16	-	@(d:16+ERs)				---	↑	↑	↑	↑	↑	↑	
	CMP.B @(d:16,ERs),@aa:32	B	6	5	8				S		D					@aa:32	-	@(d:16+ERs)				---	↑	↑	↑	↑	↑	↑	
	CMP.B @(d:32,ERs),@ERd	B	5	5	7			D	S							@ERd	-	@(d:32+ERs)				---	↑	↑	↑	↑	↑	↑	
	CMP.B @(d:32,ERs),@ERd+	B	5	5	7				S	D						@ERd	-	@(d:32+ERs)			ERd32+1→ERd32	---	↑	↑	↑	↑	↑	↑	
	CMP.B @(d:32,ERs),@ERd-	B	5	5	7				S	D						@ERd	-	@(d:32+ERs)			ERd32-1→ERd32	---	↑	↑	↑	↑	↑	↑	
	CMP.B @(d:32,ERs),@+ERd	B	5	6	8				S	D						ERd32+1→ERd32	@ERd	-	@(d:32+ERs)				---	↑	↑	↑	↑	↑	↑
	CMP.B @(d:32,ERs),@-ERd	B	5	6	8				S	D						ERd32-1→ERd32	@ERd	-	@(d:32+ERs)				---	↑	↑	↑	↑	↑	↑
	CMP.B @(d:32,ERs),@(d:2,ERd)	B	5	6	8				S							@(1/2/3+ERd)	-	@(d:32+ERs)				---	↑	↑	↑	↑	↑	↑	
CMP.B @(d:32,ERs),@(d:16,ERd)	B	6	6	8				S							@(d:16+ERd)	-	@(d:32+ERs)				---	↑	↑	↑	↑	↑	↑		
CMP.B @(d:32,ERs),@(d:32,ERd)	B	7	6	9				S							@(d:32+ERd)	-	@(d:32+ERs)				---	↑	↑	↑	↑	↑	↑		
CMP.B @(d:32,ERs),@(d:16,Rd,B)	B	6	6	8				S	D						@(d:16+RdL)	-	@(d:32+ERs)				---	↑	↑	↑	↑	↑	↑		
CMP.B @(d:32,ERs),@(d:16,Rd,W)	B	6	6	8				S	D						@(d:16+Rd)	-	@(d:32+ERs)				---	↑	↑	↑	↑	↑	↑		
CMP.B @(d:32,ERs),@(d:16,ERdL)	B	6	6	8				S	D						@(d:16+ERd)	-	@(d:32+ERs)				---	↑	↑	↑	↑	↑	↑		

Arithmetic operations (47)

Instruction	Mnemonic	Size	Instruction Length	Min.	Addressing Mode												Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	Condition Codes ²								
					16-Bit Instruction Fetch Execution Status ¹ #xx	Rn	@ERn	@ERn	@Rn.L(Rn.W ERn.L)	@ERn/@ERn+/@ERn+/@ERn	@aa:R/@aa:16/@aa:32	S	D	Operation ⁵								I	H	N	Z	V	C			
														Number of	16-Bit Instruction Fetch Execution Status ¹	Execution Status ¹												Execution Status ¹	Execution Status ¹	Execution Status ¹
CMP	CMP.B @(d:16,Rs,W),@-ERd	B	4	6	7						S	D			ERd32-1→ERd32	@ERd	-	@(d:16+Rs)						—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,Rs,W),@(d:2,ERd)	B	4	6	7						D	S			@(1/2/3+ERd)	-	@(d:16+Rs)							—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,Rs,W),@(d:16,ERd)	B	5	6	7						D	S			@(d:16+ERd)	-	@(d:16+Rs)							—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,Rs,W),@(d:32,ERd)	B	6	6	8						D	S			@(d:32+ERd)	-	@(d:16+Rs)							—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,Rs,W),@(d:16,Rd,B)	B	5	6	7						S				@(d:16+RdL)	-	@(d:16+Rs)							—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,Rs,W),@(d:16,Rd,W)	B	5	6	7						S				@(d:16+Rd)	-	@(d:16+Rs)							—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,Rs,W),@(d:16,ERd,L)	B	5	6	7						S				@(d:16+ERd)	-	@(d:16+Rs)							—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,Rs,W),@(d:32,Rd,B)	B	6	6	8						S				@(d:32+RdL)	-	@(d:16+Rs)							—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,Rs,W),@(d:32,Rd,W)	B	6	6	8						S				@(d:32+Rd)	-	@(d:16+Rs)							—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,Rs,W),@(d:32,ERd,L)	B	6	6	8						S				@(d:32+ERd)	-	@(d:16+Rs)							—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,Rs,W),@aa:16	B	5	5	7						S	D			@aa:16	-	@(d:16+Rs)							—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,Rs,W),@aa:32	B	6	5	8						S	D			@aa:32	-	@(d:16+Rs)							—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,ERs,L),@ERd	B	4	5	6							D	S		@ERd	-	@(d:16+ERs)							—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,ERs,L),@ERd+	B	4	5	6							S	D		@ERd	-	@(d:16+ERs)					ERd32+1→ERd32		—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,ERs,L),@ERd-	B	4	5	6							S	D		@ERd	-	@(d:16+ERs)					ERd32-1→ERd32		—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,ERs,L),@+ERd	B	4	6	7							S	D		ERd32+1→ERd32	@ERd	-	@(d:16+ERs)						—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,ERs,L),@-ERd	B	4	6	7							S	D		ERd32-1→ERd32	@ERd	-	@(d:16+ERs)						—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,ERs,L),@(d:2,ERd)	B	4	6	7							D	S		@(1/2/3+ERd)	-	@(d:16+ERs)							—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,ERs,L),@(d:16,ERd)	B	5	6	7							D	S		@(d:16+ERd)	-	@(d:16+ERs)							—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,ERs,L),@(d:32,ERd)	B	6	6	8							D	S		@(d:32+ERd)	-	@(d:16+ERs)							—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,ERs,L),@(d:16,Rd,B)	B	5	6	7							S			@(d:16+RdL)	-	@(d:16+ERs)							—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,ERs,L),@(d:16,Rd,W)	B	5	6	7							S			@(d:16+Rd)	-	@(d:16+ERs)							—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,ERs,L),@(d:16,ERd,L)	B	5	6	7							S			@(d:16+ERd)	-	@(d:16+ERs)							—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,ERs,L),@(d:32,Rd,B)	B	6	6	8							S			@(d:32+RdL)	-	@(d:16+ERs)							—	↑	↓	↑	↓	↑	↓
	CMP.B @(d:16,ERs,L),@(d:32,Rd,W)	B	6	6	8							S			@(d:32+Rd)	-	@(d:16+ERs)							—	↑	↓	↑	↓	↑	↓

Arithmetic operations (48)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹	Addressing Mode							Operation ⁵					Condition Codes ^{6,7}											
						Number of	Execution Stages ¹	Rn	@ERn	@(d,ERn)	@(d,Rn,WRn,LRn)	@ERn/ERn+/ERn-/@ERn												@aa:8/@aa:16/@aa:32					
n					pxx																	I	H	N	Z	V	C		
CMP	CMP.B @(d:16,ERs.L),@(d:32,ERd.L)	B	6	6	8					S					@(d:32+ERd)	-	@(d:16+ERs)					---	↑	↓	↑	↓	↑	↓	
	CMP.B @(d:16,ERs.L),@aa:16	B	5	5	7					S	D				@aa:16	-	@(d:16+ERs)					---	↑	↓	↑	↓	↑	↓	
	CMP.B @(d:16,ERs.L),@aa:32	B	6	5	8					S	D				@aa:32	-	@(d:16+ERs)					---	↑	↓	↑	↓	↑	↓	
	CMP.B @(d:32,Rs.B),@ERd	B	5	5	7					D	S				@ERd	-	@(d:32+RsL)					---	↑	↓	↑	↓	↑	↓	
	CMP.B @(d:32,Rs.B),@ERd+	B	5	5	7					S	D				@ERd	-	@(d:32+RsL)					ERd32+1→ERd32	---	↑	↓	↑	↓	↑	↓
	CMP.B @(d:32,Rs.B),@ERd-	B	5	5	7					S	D				@ERd	-	@(d:32+RsL)					ERd32-1→ERd32	---	↑	↓	↑	↓	↑	↓
	CMP.B @(d:32,Rs.B),@+ERd	B	5	6	8					S	D				ERd32+1→ERd32	@ERd	-	@(d:32+RsL)					---	↑	↓	↑	↓	↑	↓
	CMP.B @(d:32,Rs.B),@-ERd	B	5	6	8					S	D				ERd32-1→ERd32	@ERd	-	@(d:32+RsL)					---	↑	↓	↑	↓	↑	↓
	CMP.B @(d:32,Rs.B),@(d:2,ERd)	B	5	6	8					D	S				@(1/2/3+ERd)	-	@(d:32+RsL)					---	↑	↓	↑	↓	↑	↓	
	CMP.B @(d:32,Rs.B),@(d:16,ERd)	B	6	6	8					D	S				@(d:16+ERd)	-	@(d:32+RsL)					---	↑	↓	↑	↓	↑	↓	
	CMP.B @(d:32,Rs.B),@(d:32,ERd)	B	7	6	9					D	S				@(d:32+ERd)	-	@(d:32+RsL)					---	↑	↓	↑	↓	↑	↓	
	CMP.B @(d:32,Rs.B),@(d:16,Rd.B)	B	6	6	8					S					@(d:16+RdL)	-	@(d:32+RsL)					---	↑	↓	↑	↓	↑	↓	
	CMP.B @(d:32,Rs.B),@(d:16,Rd.W)	B	6	6	8					S					@(d:16+Rd)	-	@(d:32+RsL)					---	↑	↓	↑	↓	↑	↓	
	CMP.B @(d:32,Rs.B),@(d:16,ERdL)	B	6	6	8					S					@(d:16+ERd)	-	@(d:32+RsL)					---	↑	↓	↑	↓	↑	↓	
	CMP.B @(d:32,Rs.B),@(d:32,Rd.B)	B	7	6	9					S					@(d:32+RdL)	-	@(d:32+RsL)					---	↑	↓	↑	↓	↑	↓	
	CMP.B @(d:32,Rs.B),@(d:32,Rd.W)	B	7	6	9					S					@(d:32+Rd)	-	@(d:32+RsL)					---	↑	↓	↑	↓	↑	↓	
	CMP.B @(d:32,Rs.B),@(d:32,ERdL)	B	7	6	9					S					@(d:32+ERd)	-	@(d:32+RsL)					---	↑	↓	↑	↓	↑	↓	
	CMP.B @(d:32,Rs.B),@aa:16	B	6	5	8					S	D				@aa:16	-	@(d:32+RsL)					---	↑	↓	↑	↓	↑	↓	
	CMP.B @(d:32,Rs.B),@aa:32	B	7	5	9					S	D				@aa:32	-	@(d:32+RsL)					---	↑	↓	↑	↓	↑	↓	
	CMP.B @(d:32,Rs.W),@ERd	B	5	5	7					D	S				@ERd	-	@(d:32+Rs)					---	↑	↓	↑	↓	↑	↓	
	CMP.B @(d:32,Rs.W),@ERd+	B	5	5	7					S	D				@ERd	-	@(d:32+Rs)					ERd32+1→ERd32	---	↑	↓	↑	↓	↑	↓
	CMP.B @(d:32,Rs.W),@ERd-	B	5	5	7					S	D				@ERd	-	@(d:32+Rs)					ERd32-1→ERd32	---	↑	↓	↑	↓	↑	↓
	CMP.B @(d:32,Rs.W),@+ERd	B	5	6	8					S	D				ERd32+1→ERd32	@ERd	-	@(d:32+Rs)					---	↑	↓	↑	↓	↑	↓
	CMP.B @(d:32,Rs.W),@-ERd	B	5	6	8					S	D				ERd32-1→ERd32	@ERd	-	@(d:32+Rs)					---	↑	↓	↑	↓	↑	↓
	CMP.B @(d:32,Rs.W),@(d:2,ERd)	B	5	6	8					D	S				@(1/2/3+ERd)	-	@(d:32+Rs)					---	↑	↓	↑	↓	↑	↓	

Arithmetic operations (49)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages* #cyc	Addressing Mode								Operation ⁵					Condition Codes ⁶									
						Rn	@ERn	@(d, ERn)	@(d.Rn, ERn.WERn.L)	@.ERn/ERn+/ERn-/@ERn	@.aa:R/ @.aa:16/ @.aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C						
CMP	CMP.B @(d:32.Rs.W), @(d:16, ERd)	B	6	6	8			D	S			@(d:16+ERd)	-	@(d:32+Rs)														
	CMP.B @(d:32.Rs.W), @(d:32, ERd)	B	7	6	9			D	S			@(d:32+ERd)	-	@(d:32+Rs)														
	CMP.B @(d:32.Rs.W), @(d:16, Rd.B)	B	6	6	8				S			@(d:16+RdL)	-	@(d:32+Rs)														
	CMP.B @(d:32.Rs.W), @(d:16, Rd.W)	B	6	6	8				S			@(d:16+Rd)	-	@(d:32+Rs)														
	CMP.B @(d:32.Rs.W), @(d:16, ERd.L)	B	6	6	8				S			@(d:16+ERd)	-	@(d:32+Rs)														
	CMP.B @(d:32.Rs.W), @(d:32, Rd.B)	B	7	6	9				S			@(d:32+RdL)	-	@(d:32+Rs)														
	CMP.B @(d:32.Rs.W), @(d:32, Rd.W)	B	7	6	9				S			@(d:32+Rd)	-	@(d:32+Rs)														
	CMP.B @(d:32.Rs.W), @(d:32, ERd.L)	B	7	6	9				S			@(d:32+ERd)	-	@(d:32+Rs)														
	CMP.B @(d:32.Rs.W), @.aa:16	B	6	5	8				S	D		@.aa:16	-	@(d:32+Rs)														
	CMP.B @(d:32.Rs.W), @.aa:32	B	7	5	9				S	D		@.aa:32	-	@(d:32+Rs)														
	CMP.B @(d:32, ERs.L), @ERd	B	5	5	7			D	S			@ERd	-	@(d:32+ERs)														
	CMP.B @(d:32, ERs.L), @ERd+	B	5	5	7				S	D		@ERd	-	@(d:32+ERs)						ERd32+1 → ERd32								
	CMP.B @(d:32, ERs.L), @ERd-	B	5	5	7				S	D		@ERd	-	@(d:32+ERs)						ERd32-1 → ERd32								
	CMP.B @(d:32, ERs.L), @+ERd	B	5	6	8				S	D		ERd32+1 → ERd32	@ERd	-	@(d:32+ERs)													
	CMP.B @(d:32, ERs.L), @-ERd	B	5	6	8				S	D		ERd32-1 → ERd32	@ERd	-	@(d:32+ERs)													
	CMP.B @(d:32, ERs.L), @(d:2, ERd)	B	5	6	8			D	S			@(1/2/3+ERd)	-	@(d:32+ERs)														
	CMP.B @(d:32, ERs.L), @(d:16, ERd)	B	6	6	8				D	S		@(d:16+ERd)	-	@(d:32+ERs)														
	CMP.B @(d:32, ERs.L), @(d:32, ERd)	B	7	6	9				D	S		@(d:32+ERd)	-	@(d:32+ERs)														
	CMP.B @(d:32, ERs.L), @(d:16, Rd.B)	B	6	6	8				S			@(d:16+RdL)	-	@(d:32+ERs)														
	CMP.B @(d:32, ERs.L), @(d:16, Rd.W)	B	6	6	8				S			@(d:16+Rd)	-	@(d:32+ERs)														
	CMP.B @(d:32, ERs.L), @(d:16, ERd.L)	B	6	6	8				S			@(d:16+ERd)	-	@(d:32+ERs)														
	CMP.B @(d:32, ERs.L), @(d:32, Rd.B)	B	7	6	9				S			@(d:32+RdL)	-	@(d:32+ERs)														
	CMP.B @(d:32, ERs.L), @(d:32, Rd.W)	B	7	6	9				S			@(d:32+Rd)	-	@(d:32+ERs)														
	CMP.B @(d:32, ERs.L), @(d:32, ERd.L)	B	7	6	9				S			@(d:32+ERd)	-	@(d:32+ERs)														
CMP.B @(d:32, ERs.L), @.aa:16	B	6	5	8				S	D		@.aa:16	-	@(d:32+ERs)															

Arithmetic operations (50)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Status ¹ #px	Addressing Mode						Operation ⁵					Condition Codes ^{6,2}					
						Rn	@ERn	@d.ERn	@d.Rn.L,ERn,W,ERn.L	@.ERn/@.ERn+/@.ERn+/@.ERn	@aa:R/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C
CMP	CMP.B @(d:32,ERs.L),@aa:32	B	7	5	9			S	D			@aa:32 - @d:32+ERs			—	↑	↑	↑	↑	↑	↑	
	CMP.B @aa:16,@ERd	B	3	3	5		D		S			@ERd - @aa:16			—	↑	↑	↑	↑	↑	↑	
	CMP.B @aa:16,@ERd+	B	3	3	5			D	S			@ERd - @aa:16		ERd32+1→ERd32	—	↑	↑	↑	↑	↑	↑	
	CMP.B @aa:16,@ERd-	B	3	3	5			D	S			@ERd - @aa:16		ERd32-1→ERd32	—	↑	↑	↑	↑	↑	↑	
	CMP.B @aa:16,@+ERd	B	3	4	6			D	S		ERd32+1→ERd32	@ERd - @aa:16			—	↑	↑	↑	↑	↑	↑	
	CMP.B @aa:16,@-ERd	B	3	4	6			D	S		ERd32-1→ERd32	@ERd - @aa:16			—	↑	↑	↑	↑	↑	↑	
	CMP.B @aa:16,@(d:2,ERd)	B	3	4	6			D	S			@(1/2/3+ERd) - @aa:16			—	↑	↑	↑	↑	↑	↑	
	CMP.B @aa:16,@(d:16,ERd)	B	4	4	6			D	S			@(d:16+ERd) - @aa:16			—	↑	↑	↑	↑	↑	↑	
	CMP.B @aa:16,@(d:32,ERd)	B	5	4	7			D	S			@(d:32+ERd) - @aa:16			—	↑	↑	↑	↑	↑	↑	
	CMP.B @aa:16,@(d:16,Rd.B)	B	4	4	6			D	S			@(d:16+RdL) - @aa:16			—	↑	↑	↑	↑	↑	↑	
	CMP.B @aa:16,@(d:16,Rd.W)	B	4	4	6			D	S			@(d:16+Rd) - @aa:16			—	↑	↑	↑	↑	↑	↑	
	CMP.B @aa:16,@(d:16,ERd.L)	B	4	4	6			D	S			@(d:16+ERd) - @aa:16			—	↑	↑	↑	↑	↑	↑	
	CMP.B @aa:16,@(d:32,Rd.B)	B	5	4	7			D	S			@(d:32+RdL) - @aa:16			—	↑	↑	↑	↑	↑	↑	
	CMP.B @aa:16,@(d:32,Rd.W)	B	5	4	7			D	S			@(d:32+Rd) - @aa:16			—	↑	↑	↑	↑	↑	↑	
	CMP.B @aa:16,@(d:32,ERd.L)	B	5	4	7			D	S			@(d:32+ERd) - @aa:16			—	↑	↑	↑	↑	↑	↑	
	CMP.B @aa:16,@aa:16	B	4	3	6				S			@aa:16 - @aa:16			—	↑	↑	↑	↑	↑	↑	
	CMP.B @aa:16,@aa:32	B	5	3	7				S			@aa:32 - @aa:16			—	↑	↑	↑	↑	↑	↑	
	CMP.B @aa:32,@ERd	B	4	3	6			D	S			@ERd - @aa:32			—	↑	↑	↑	↑	↑	↑	
	CMP.B @aa:32,@ERd+	B	4	3	6			D	S			@ERd - @aa:32		ERd32+1→ERd32	—	↑	↑	↑	↑	↑	↑	
	CMP.B @aa:32,@ERd-	B	4	3	6			D	S			@ERd - @aa:32		ERd32-1→ERd32	—	↑	↑	↑	↑	↑	↑	
CMP.B @aa:32,@+ERd	B	4	4	7			D	S		ERd32+1→ERd32	@ERd - @aa:32			—	↑	↑	↑	↑	↑	↑		
CMP.B @aa:32,@-ERd	B	4	4	7			D	S		ERd32-1→ERd32	@ERd - @aa:32			—	↑	↑	↑	↑	↑	↑		
CMP.B @aa:32,@(d:2,ERd)	B	4	4	7			D	S			@(1/2/3+ERd) - @aa:32			—	↑	↑	↑	↑	↑	↑		
CMP.B @aa:32,@(d:16,ERd)	B	5	4	7			D	S			@(d:16+ERd) - @aa:32			—	↑	↑	↑	↑	↑	↑		

Arithmetic operations (60)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode										Operation ⁶					Condition Codes ⁸							
				Number of		16-Bit Instruction Fetch Execution Stages ¹	Pxx	Rn	@ERN	@ (d, ERn)	@ (d, Rn, ERn, WERn, L)	@ ERn / @ ERn+ / @ ERn+ / @ ERn	@ aa:9 / @ aa:16 / @ aa:32											Operation 1	Operation 2	Operation 3
				Min.	Max.																					
CMP	CMP.W @(d:16,ERs.L),@ERd+	W	4	5	6							S	D			@ERd - @(d:16+ERs<<1)			ERd32+2→ERd32	—	↓	↑	↓	↑	↓	
	CMP.W @(d:16,ERs.L),@ERd-	W	4	5	6							S	D			@ERd - @(d:16+ERs<<1)			ERd32-2→ERd32	—	↑	↓	↑	↓	↑	
	CMP.W @(d:16,ERs.L),@+ERd	W	4	6	7								S	D		ERd32+2→ERd32	@ERd - @(d:16+ERs<<1)				—	↑	↓	↑	↓	↑
	CMP.W @(d:16,ERs.L),@-ERd	W	4	6	7								S	D		ERd32-2→ERd32	@ERd - @(d:16+ERs<<1)				—	↓	↑	↓	↑	↓
	CMP.W @(d:16,ERs.L),@(d:2,ERd)	W	4	6	7							D	S			@(2/4+ERd) - @(d:16+ERs<<1)				—	↑	↓	↑	↓	↑	
	CMP.W @(d:16,ERs.L),@(d:16,ERd)	W	5	6	7							D	S			@(d:16+ERd) - @(d:16+ERs<<1)				—	↓	↑	↓	↑	↓	
	CMP.W @(d:16,ERs.L),@(d:32,ERd)	W	6	6	8							D	S			@(d:32+ERd) - @(d:16+ERs<<1)				—	↓	↑	↓	↑	↓	
	CMP.W @(d:16,ERs.L),@(d:16,Rd,B)	W	5	6	7								S			@(d:16+Rd<<1) - @(d:16+ERs<<1)				—	↑	↓	↑	↓	↑	
	CMP.W @(d:16,ERs.L),@(d:16,Rd,W)	W	5	6	7								S			@(d:16+Rd<<1) - @(d:16+ERs<<1)				—	↑	↓	↑	↓	↑	
	CMP.W @(d:16,ERs.L),@(d:16,ERd,L)	W	5	6	7								S			@(d:16+ERd<<1) - @(d:16+ERs<<1)				—	↑	↓	↑	↓	↑	
	CMP.W @(d:16,ERs.L),@(d:32,Rd,B)	W	6	6	8								S			@(d:32+Rd<<1) - @(d:16+ERs<<1)				—	↑	↓	↑	↓	↑	
	CMP.W @(d:16,ERs.L),@(d:32,Rd,W)	W	6	6	8								S			@(d:32+Rd<<1) - @(d:16+ERs<<1)				—	↑	↓	↑	↓	↑	
	CMP.W @(d:16,ERs.L),@(d:32,ERd,L)	W	6	6	8								S			@(d:32+ERd<<1) - @(d:16+ERs<<1)				—	↑	↓	↑	↓	↑	
	CMP.W @(d:16,ERs.L),@aa:16	W	5	5	7								S	D		@aa:16 - @(d:16+ERs<<1)				—	↑	↓	↑	↓	↑	
	CMP.W @(d:16,ERs.L),@aa:32	W	6	5	8								S	D		@aa:32 - @(d:16+ERs<<1)				—	↑	↓	↑	↓	↑	
	CMP.W @(d:32,Rs,B),@ERd	W	5	5	7							D	S			@ERd - @(d:32+RsL<<1)				—	↑	↓	↑	↓	↑	
	CMP.W @(d:32,Rs,B),@ERd+	W	5	5	7								S	D		@ERd - @(d:32+RsL<<1)			ERd32+2→ERd32	—	↑	↓	↑	↓	↑	
	CMP.W @(d:32,Rs,B),@ERd-	W	5	5	7								S	D		@ERd - @(d:32+RsL<<1)			ERd32-2→ERd32	—	↓	↑	↓	↑	↓	
	CMP.W @(d:32,Rs,B),@+ERd	W	5	6	8								S	D		ERd32+2→ERd32	@ERd - @(d:32+RsL<<1)				—	↑	↓	↑	↓	↑
	CMP.W @(d:32,Rs,B),@-ERd	W	5	6	8								S	D		ERd32-2→ERd32	@ERd - @(d:32+RsL<<1)				—	↓	↑	↓	↑	↓
	CMP.W @(d:32,Rs,B),@(d:2,ERd)	W	6	6	8							D	S			@(2/4+ERd) - @(d:32+RsL<<1)				—	↑	↓	↑	↓	↑	
	CMP.W @(d:32,Rs,B),@(d:16,ERd)	W	6	6	8								D	S		@(d:16+ERd) - @(d:32+RsL<<1)				—	↑	↓	↑	↓	↑	
	CMP.W @(d:32,Rs,B),@(d:32,ERd)	W	7	6	9								D	S		@(d:32+ERd) - @(d:32+RsL<<1)				—	↑	↓	↑	↓	↑	
	CMP.W @(d:32,Rs,B),@(d:16,Rd,B)	W	6	6	8								S			@(d:16+Rd<<1) - @(d:32+RsL<<1)				—	↑	↓	↑	↓	↑	
	CMP.W @(d:32,Rs,B),@(d:16,Rd,W)	W	6	6	8								S			@(d:16+Rd<<1) - @(d:32+RsL<<1)				—	↑	↓	↑	↓	↑	



Arithmetic operations (64)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	16-Bit Instruction Fetch Execution Status ¹	Addressing Mode								Operation ⁵					Condition Codes ²											
							Number of Execution Status ¹	Rn	@ERn	@(d,ERn)	@(d,Rn,WRn,L)	@ERn/ERn+/@ERn/ERn-/@+ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C							
CMP	CMP.L #xx:16,@ERd-	L	3	3	4	S											@ERd	-	#xx:16			ERd32-4→ERd32	---	↑	↑	↑	↑	↑	↑		
	CMP.L #xx:16,@+ERd	L	3	4	4	S											ERd32+4→ERd32	@ERd	-	#xx:16				---	↑	↑	↑	↑	↑	↑	
	CMP.L #xx:16,@-ERd	L	3	4	4	S											ERd32-4→ERd32	@ERd	-	#xx:16				---	↑	↑	↑	↑	↑	↑	
	CMP.L #xx:16,@(d-2,ERd)	L	3	4	4	S												@(4/8/12+ERd)	-	#xx:16				---	↑	↑	↑	↑	↑	↑	
	CMP.L #xx:16,@(d:16,ERd)	L	4	4	5	S												@(d:16+ERd)	-	#xx:16				---	↑	↑	↑	↑	↑	↑	
	CMP.L #xx:16,@(d:32,ERd)	L	5	4	6	S												@(d:32+ERd)	-	#xx:16				---	↑	↑	↑	↑	↑	↑	
	CMP.L #xx:16,@(d:16,Rd.B)	L	4	4	5	S												@(d:16+RdL<<2)	-	#xx:16				---	↑	↑	↑	↑	↑	↑	
	CMP.L #xx:16,@(d:16,Rd.W)	L	4	4	5	S												@(d:16+Rd<<2)	-	#xx:16				---	↑	↑	↑	↑	↑	↑	
	CMP.L #xx:16,@(d:16,ERd.L)	L	4	4	5	S												@(d:16+ERdL<<2)	-	#xx:16				---	↑	↑	↑	↑	↑	↑	
	CMP.L #xx:16,@(d:32,Rd.B)	L	5	4	6	S												@(d:32+RdL<<2)	-	#xx:16				---	↑	↑	↑	↑	↑	↑	
	CMP.L #xx:16,@(d:32,Rd.W)	L	5	4	6	S												@(d:32+Rd<<2)	-	#xx:16				---	↑	↑	↑	↑	↑	↑	
	CMP.L #xx:16,@(d:32,ERd.L)	L	5	4	6	S												@(d:32+ERdL<<2)	-	#xx:16				---	↑	↑	↑	↑	↑	↑	
	CMP.L #xx:16,@aa:16	L	4	3	5	S												@aa:16	-	#xx:16				---	↑	↑	↑	↑	↑	↑	
	CMP.L #xx:16,@aa:32	L	5	3	6	S												@aa:32	-	#xx:16				---	↑	↑	↑	↑	↑	↑	
	CMP.L #xx:32,@ERd	L	4	3	5	S												@ERd	-	#xx				---	↑	↑	↑	↑	↑	↑	
	CMP.L #xx:32,@ERd+	L	4	3	5	S												@ERd	-	#xx				ERd32+4→ERd32	---	↑	↑	↑	↑	↑	↑
	CMP.L #xx:32,@ERd-	L	4	3	5	S												@ERd	-	#xx				ERd32-4→ERd32	---	↑	↑	↑	↑	↑	↑
	CMP.L #xx:32,@+ERd	L	4	4	5	S												ERd32+4→ERd32	@ERd	-	#xx				---	↑	↑	↑	↑	↑	↑
	CMP.L #xx:32,@-ERd	L	4	4	5	S												ERd32-4→ERd32	@ERd	-	#xx				---	↑	↑	↑	↑	↑	↑
	CMP.L #xx:32,@(d-2,ERd)	L	4	4	5	S												@(4/8/12+ERd)	-	#xx					---	↑	↑	↑	↑	↑	↑
	CMP.L #xx:32,@(d:16,ERd)	L	5	4	6	S												@(d:16+ERd)	-	#xx					---	↑	↑	↑	↑	↑	↑
	CMP.L #xx:32,@(d:32,ERd)	L	6	4	7	S												@(d:32+ERd)	-	#xx					---	↑	↑	↑	↑	↑	↑
	CMP.L #xx:32,@(d:16,Rd.B)	L	5	4	6	S												@(d:16+RdL<<2)	-	#xx					---	↑	↑	↑	↑	↑	↑
CMP.L #xx:32,@(d:16,Rd.W)	L	5	4	6	S												@(d:16+Rd<<2)	-	#xx					---	↑	↑	↑	↑	↑	↑	



Arithmetic operations (66)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode							Operation ⁵					Condition Codes ^{6,7}									
				Min.	16-Bit Instruction Fetch Execution Status ⁸	Prx	Rn	@ERn	@(d,ERn)	@(d,Rn,ERn,WERn,L)	@ERn/@ERn+/@ERn+/@ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C		
CMP	CMP.L @ERs, ERd	L	2	3	3	D								ERd - @ERs	ERs32-4→ERs32										
	CMP.L @+ERs, ERd	L	2	4	4	D								ERs32+4→ERs32											
	CMP.L @-ERs, ERd	L	2	4	4	D								ERs32-4→ERs32											
	CMP.L @(d:2,ERs), ERd	L	2	4	4	D	S							ERd - @(4/8/12+ERs)											
	CMP.L @(d:16,ERs), ERd	L	3	4	4	D	S							ERd - @(d:16+ERs)											
	CMP.L @(d:32,ERs), ERd	L	4	4	5	D	S							ERd - @(d:32+ERs)											
	CMP.L @(d:16,Rs,B), ERd	L	3	4	4	D	S							ERd - @(d:16+RsL<<2)											
	CMP.L @(d:16,Rs,W), ERd	L	3	4	4	D	S							ERd - @(d:16+Rs<<2)											
	CMP.L @(d:16,ERs,L), ERd	L	3	4	4	D	S							ERd - @(d:16+ERs<<2)											
	CMP.L @(d:32,Rs,B), ERd	L	4	4	5	D	S							ERd - @(d:32+RsL<<2)											
	CMP.L @(d:32,Rs,W), ERd	L	4	4	5	D	S							ERd - @(d:32+Rs<<2)											
	CMP.L @(d:32,ERs,L), ERd	L	4	4	5	D	S							ERd - @(d:32+ERs<<2)											
	CMP.L @aa:16, ERd	L	3	3	4	D				S				ERd - @aa:16											
	CMP.L @aa:32, ERd	L	4	3	5	D				S				ERd - @aa:32											
	CMP.L @ERs, @ERd	L	3	4	5		S							@ERd - @ERs											
	CMP.L @ERs, @ERd+	L	3	4	5		S		D					@ERd - @ERs	ERd32+4→ERd32										
	CMP.L @ERs, @ERd-	L	3	4	5		S		D					@ERd - @ERs	ERd32-4→ERd32										
	CMP.L @ERs, @+ERd	L	3	5	6		S		D					ERd32+4→ERd32	@ERd - @ERs										
	CMP.L @ERs, @-ERd	L	3	5	6		S		D					ERd32-4→ERd32	@ERd - @ERs										
	CMP.L @ERs, @(d:2,ERd)	L	3	5	6		S	D						@(4/8/12+ERd) - @ERs											
	CMP.L @ERs, @(d:16,ERd)	L	4	5	6		S	D						@(d:16+ERd) - @ERs											
	CMP.L @ERs, @(d:32,ERd)	L	5	5	7		S	D						@(d:32+ERd) - @ERs											
	CMP.L @ERs, @(d:16,Rd,B)	L	4	5	6		S	D						@(d:16+RdL<<2) - @ERs											
CMP.L @ERs, @(d:16,Rd,W)	L	4	5	6		S	D						@(d:16+Rd<<2) - @ERs												
CMP.L @ERs, @(d:16,ERd,L)	L	4	5	6		S	D						@(d:16+ERd<<2) - @ERs												



Arithmetic operations (71)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode							Operation ⁶					Condition Codes ^{5,2}																	
				Min.	16-Bit Instruction Fetch Execution Status ¹ #xx	Rn	@ERn	@ERn	@ERn	@ERn+@ERn											@ERn+@ERn+@ERn	@ERn+@ERn+@ERn+@ERn	@ERn+@ERn+@ERn+@ERn										
											Number of Execution Stages ¹	Execution Stages ¹	Execution Stages ¹	Execution Stages ¹	Execution Stages ¹	Execution Stages ¹	Execution Stages ¹	Execution Stages ¹															
n				Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C																			
CMP	CMP.L @(d:16,ERs), @aa:32	L	6	5	8									S	D																		
	CMP.L @(d:32,ERs), @ERd	L	5	5	7									D	S																		
	CMP.L @(d:32,ERs), @ERd+	L	5	5	7										S	D																	
	CMP.L @(d:32,ERs), @ERd-	L	5	5	7										S	D																	
	CMP.L @(d:32,ERs), @+ERd	L	5	6	8										S	D																	
	CMP.L @(d:32,ERs), @-ERd	L	5	6	8										S	D																	
	CMP.L @(d:32,ERs), @(d:2,ERd)	L	5	6	8										S																		
	CMP.L @(d:32,ERs), @(d:16,ERd)	L	6	6	8										S																		
	CMP.L @(d:32,ERs), @(d:32,ERd)	L	7	6	9										S																		
	CMP.L @(d:32,ERs), @(d:16,Rd.B)	L	6	6	8										S	D																	
	CMP.L @(d:32,ERs), @(d:16,Rd.W)	L	6	6	8										S	D																	
	CMP.L @(d:32,ERs), @(d:16,ERd.L)	L	6	6	8										S	D																	
	CMP.L @(d:32,ERs), @(d:32,Rd.B)	L	7	6	9										S	D																	
	CMP.L @(d:32,ERs), @(d:32,Rd.W)	L	7	6	9										S	D																	
	CMP.L @(d:32,ERs), @(d:32,ERd.L)	L	7	6	9										S	D																	
	CMP.L @(d:32,ERs), @aa:16	L	6	5	8										S																		
	CMP.L @(d:32,ERs), @aa:32	L	7	5	9										S																		
	CMP.L @(d:16,Rs.B), @ERd	L	4	5	6											D	S																
	CMP.L @(d:16,Rs.B), @ERd+	L	4	5	6											S	D																
	CMP.L @(d:16,Rs.B), @ERd-	L	4	5	6											S	D																
CMP.L @(d:16,Rs.B), @+ERd	L	4	6	7											S	D																	
CMP.L @(d:16,Rs.B), @-ERd	L	4	6	7											S	D																	
CMP.L @(d:16,Rs.B), @(d:2,ERd)	L	4	6	7											D	S																	
CMP.L @(d:16,Rs.B), @(d:16,ERd)	L	5	6	7											D	S																	



Arithmetic operations (73)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode								Operation ⁵					Condition Code ^{6,2}																					
				Min	Max	16-Bit Instruction Fetch Execution Status ¹	Prx	Rn	@ERN	@(d,ERN)	@(d,Rn,LRn,WERN,LRn)											@ERN/@ERN+/@ERN+/@ERN+/@ERN	@aa:9/@aa:16/@aa:32															
n																																						
CMP	CMP.L @(d:16,ERs.L),@ERd	L	4 5 6				D	S							@ERd	-	@(d:16+ERs<2)																					
	CMP.L @(d:16,ERs.L),@ERd+	L	4 5 6					S	D						@ERd	-	@(d:16+ERs<2)		ERd32+4→ERd32																			
	CMP.L @(d:16,ERs.L),@ERd-	L	4 5 6					S	D						@ERd	-	@(d:16+ERs<2)		ERd32-4→ERd32																			
	CMP.L @(d:16,ERs.L),@+ERd	L	4 6 7					S	D					ERd32+4→ERd32	@ERd	-	@(d:16+ERs<2)																					
	CMP.L @(d:16,ERs.L),@-ERd	L	4 6 7					S	D					ERd32-4→ERd32	@ERd	-	@(d:16+ERs<2)																					
	CMP.L @(d:16,ERs.L),@(d:2,ERd)	L	4 6 7					D	S						@(4/8/12+ERd)	-	@(d:16+ERs<2)																					
	CMP.L @(d:16,ERs.L),@(d:16,ERd)	L	5 6 7					D	S						@(d:16+ERd)	-	@(d:16+ERs<2)																					
	CMP.L @(d:16,ERs.L),@(d:32,ERd)	L	6 6 8					D	S						@(d:32+ERd)	-	@(d:16+ERs<2)																					
	CMP.L @(d:16,ERs.L),@(d:16,Rd,B)	L	5 6 7						S						@(d:16+RdL<2)	-	@(d:16+ERs<2)																					
	CMP.L @(d:16,ERs.L),@(d:16,Rd,W)	L	5 6 7						S						@(d:16+Rd<2)	-	@(d:16+ERs<2)																					
	CMP.L @(d:16,ERs.L),@(d:16,ERd,L)	L	5 6 7						S						@(d:16+ERd<2)	-	@(d:16+ERs<2)																					
	CMP.L @(d:16,ERs.L),@(d:32,Rd,B)	L	6 6 8						S						@(d:32+RdL<2)	-	@(d:16+ERs<2)																					
	CMP.L @(d:16,ERs.L),@(d:32,Rd,W)	L	6 6 8						S						@(d:32+Rd<2)	-	@(d:16+ERs<2)																					
	CMP.L @(d:16,ERs.L),@(d:32,ERd,L)	L	6 6 8						S						@(d:32+ERd<2)	-	@(d:16+ERs<2)																					
	CMP.L @(d:16,ERs.L),@aa:16	L	5 5 7						S	D					@aa:16	-	@(d:16+ERs<2)																					
	CMP.L @(d:16,ERs.L),@aa:32	L	6 5 8						S	D					@aa:32	-	@(d:16+ERs<2)																					
	CMP.L @(d:32,Rs,B),@ERd	L	5 5 7					D	S						@ERd	-	@(d:32+RSL<2)																					
	CMP.L @(d:32,Rs,B),@ERd+	L	5 5 7						S	D					@ERd	-	@(d:32+RSL<2)		ERd32+4→ERd32																			
	CMP.L @(d:32,Rs,B),@ERd-	L	5 5 7						S	D					@ERd	-	@(d:32+RSL<2)		ERd32-4→ERd32																			
	CMP.L @(d:32,Rs,B),@+ERd	L	5 6 8						S	D				ERd32+4→ERd32	@ERd	-	@(d:32+RSL<2)																					
	CMP.L @(d:32,Rs,B),@-ERd	L	5 6 8						S	D				ERd32-4→ERd32	@ERd	-	@(d:32+RSL<2)																					
	CMP.L @(d:32,Rs,B),@(d:2,ERd)	L	5 6 8					D	S						@(4/8/12+ERd)	-	@(d:32+RSL<2)																					
	CMP.L @(d:32,Rs,B),@(d:16,ERd)	L	6 6 8						D	S					@(d:16+ERd)	-	@(d:32+RSL<2)																					
	CMP.L @(d:32,Rs,B),@(d:32,ERd)	L	7 6 9						D	S					@(d:32+ERd)	-	@(d:32+RSL<2)																					
	CMP.L @(d:32,Rs,B),@(d:16,Rd,B)	L	6 6 8						S						@(d:16+RdL<2)	-	@(d:32+RSL<2)																					

Arithmetic operations (74)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode										Operation ⁶					Condition Codes ^{8,9}					
				Min.	16-Bit Instruction Fetch Execution Status ¹	16-Bit Instruction Fetch Execution Status ²	#xx	Rn	@ERN	@(d,ERN)	@(d,Rn,WRn,L)	@ERN/@ERN+/@ERNr/@+ERN	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C
CMP	CMP.L @(d:32,Rs.B),@(d:16,Rd.W)	L	6 6 8									S					@(d:16+Rd<<2) - @(d:32+RsL<<2)							
CMP.L @(d:32,Rs.B),@(d:16,ERd.L)		L	6 6 8									S					@(d:16+ERd<<2) - @(d:32+RsL<<2)							
CMP.L @(d:32,Rs.B),@(d:32,Rd.B)		L	7 6 9									S					@(d:32+RdL<<2) - @(d:32+RsL<<2)							
CMP.L @(d:32,Rs.B),@(d:32,Rd.W)		L	7 6 9									S					@(d:32+Rd<<2) - @(d:32+RsL<<2)							
CMP.L @(d:32,Rs.B),@(d:32,ERd.L)		L	7 6 9									S					@(d:32+ERd<<2) - @(d:32+RsL<<2)							
CMP.L @(d:32,Rs.B),@aa:16		L	6 5 8									S	D				@aa:16 - @(d:32+RsL<<2)							
CMP.L @(d:32,Rs.B),@aa:32		L	7 5 9									S	D				@aa:32 - @(d:32+RsL<<2)							
CMP.L @(d:32,Rs.W),@ERd		L	5 5 7							D		S					@ERd - @(d:32+Rs<<2)							
CMP.L @(d:32,Rs.W),@ERd+		L	5 5 7									S	D				@ERd - @(d:32+Rs<<2)					ERd32+4→ERd32		
CMP.L @(d:32,Rs.W),@ERd-		L	5 5 7									S	D				@ERd - @(d:32+Rs<<2)					ERd32-4→ERd32		
CMP.L @(d:32,Rs.W),@+ERd		L	5 6 8									S	D			ERd32+4→ERd32	@ERd - @(d:32+Rs<<2)							
CMP.L @(d:32,Rs.W),@-ERd		L	5 6 8									S	D			ERd32-4→ERd32	@ERd - @(d:32+Rs<<2)							
CMP.L @(d:32,Rs.W),@(d:2,ERd)		L	5 6 8							D	S						@(4/8/12+ERd) - @(d:32+Rs<<2)							
CMP.L @(d:32,Rs.W),@(d:16,ERd)		L	6 6 8									D	S				@(d:16+ERd) - @(d:32+Rs<<2)							
CMP.L @(d:32,Rs.W),@(d:32,ERd)		L	7 6 9									D	S				@(d:32+ERd) - @(d:32+Rs<<2)							
CMP.L @(d:32,Rs.W),@(d:16,Rd.B)		L	6 6 8									S					@(d:16+RdL<<2) - @(d:32+Rs<<2)							
CMP.L @(d:32,Rs.W),@(d:16,Rd.W)		L	6 6 8									S					@(d:16+Rd<<2) - @(d:32+Rs<<2)							
CMP.L @(d:32,Rs.W),@(d:16,ERd.L)		L	6 6 8									S					@(d:16+ERd<<2) - @(d:32+Rs<<2)							
CMP.L @(d:32,Rs.W),@(d:32,Rd.B)		L	7 6 9									S					@(d:32+RdL<<2) - @(d:32+Rs<<2)							
CMP.L @(d:32,Rs.W),@(d:32,Rd.W)		L	7 6 9									S					@(d:32+Rd<<2) - @(d:32+Rs<<2)							
CMP.L @(d:32,Rs.W),@(d:32,ERd.L)		L	7 6 9									S					@(d:32+ERd<<2) - @(d:32+Rs<<2)							
CMP.L @(d:32,Rs.W),@aa:16		L	6 5 8									S	D				@aa:16 - @(d:32+Rs<<2)							
CMP.L @(d:32,Rs.W),@aa:32		L	7 5 9									S	D				@aa:32 - @(d:32+Rs<<2)							
CMP.L @(d:32,ERs.L),@ERd		L	5 5 7							D		S					@ERd - @(d:32+ERs<<2)							
CMP.L @(d:32,ERs.L),@ERd+		L	5 5 7									S	D				@ERd - @(d:32+ERs<<2)					ERd32+4→ERd32		

Arithmetic operations (75)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode										Operation ⁵					Condition Codes ^{6,7}										
				Number of		16-Bit Instruction Fetch Execution Status ¹	Pxx	Rn	@ERn	@ERn	@ERn	@ERn	@ERn											@ERn	@ERn	@ERn	@ERn	@ERn	
				Min.	Max.																								
CMP	CMP.L @(d:32,ERs.L),@ERd-	L	5	5	7							S	D			@ERd	-	@(d:32+ERs<<2)			ERd32-4→ERd32	---	↑	↑	↑	↑	↑		
	CMP.L @(d:32,ERs.L),@+ERd	L	5	6	8							S	D			ERd32+4→ERd32	@ERd	-	@(d:32+ERs<<2)				---	↑	↑	↑	↑	↑	
	CMP.L @(d:32,ERs.L),@-ERd	L	5	6	8							S	D			ERd32-4→ERd32	@ERd	-	@(d:32+ERs<<2)				---	↑	↑	↑	↑	↑	
	CMP.L @(d:32,ERs.L),@(d:2,ERd)	L	5	6	8							D	S				@(4/8/12+ERd)	-	@(d:32+ERs<<2)				---	↑	↑	↑	↑	↑	
	CMP.L @(d:32,ERs.L),@(d:16,ERd)	L	6	6	8							D	S				@(d:16+ERd)	-	@(d:32+ERs<<2)				---	↑	↑	↑	↑	↑	
	CMP.L @(d:32,ERs.L),@(d:32,ERd)	L	7	6	9							D	S				@(d:32+ERd)	-	@(d:32+ERs<<2)				---	↑	↑	↑	↑	↑	
	CMP.L @(d:32,ERs.L),@(d:16,Rd.B)	L	6	6	8								S				@(d:16+Rd<<2)	-	@(d:32+ERs<<2)				---	↑	↑	↑	↑	↑	
	CMP.L @(d:32,ERs.L),@(d:16,Rd.W)	L	6	6	8								S				@(d:16+Rd<<2)	-	@(d:32+ERs<<2)				---	↑	↑	↑	↑	↑	
	CMP.L @(d:32,ERs.L),@(d:16,ERd.L)	L	6	6	8								S				@(d:16+ERd<<2)	-	@(d:32+ERs<<2)				---	↑	↑	↑	↑	↑	
	CMP.L @(d:32,ERs.L),@(d:32,Rd.B)	L	7	6	9								S				@(d:32+Rd<<2)	-	@(d:32+ERs<<2)				---	↑	↑	↑	↑	↑	
	CMP.L @(d:32,ERs.L),@(d:32,Rd.W)	L	7	6	9								S				@(d:32+Rd<<2)	-	@(d:32+ERs<<2)				---	↑	↑	↑	↑	↑	
	CMP.L @(d:32,ERs.L),@(d:32,ERd.L)	L	7	6	9								S				@(d:32+ERd<<2)	-	@(d:32+ERs<<2)				---	↑	↑	↑	↑	↑	
	CMP.L @(d:32,ERs.L),@aa:16	L	6	5	8								S	D			@aa:16	-	@(d:32+ERs<<2)				---	↑	↑	↑	↑	↑	
	CMP.L @(d:32,ERs.L),@aa:32	L	7	5	9								S	D			@aa:32	-	@(d:32+ERs<<2)				---	↑	↑	↑	↑	↑	
	CMP.L @aa:16,@ERd	L	4	4	6							D		S			@ERd	-	@aa:16				---	↑	↑	↑	↑	↑	
	CMP.L @aa:16,@ERd+	L	4	4	6								D	S			@ERd	-	@aa:16			ERd32+4→ERd32	---	↑	↑	↑	↑	↑	
	CMP.L @aa:16,@ERd-	L	4	4	6								D	S			@ERd	-	@aa:16			ERd32-4→ERd32	---	↑	↑	↑	↑	↑	
	CMP.L @aa:16,@+ERd	L	4	5	7								D	S			ERd32+4→ERd32	@ERd	-	@aa:16				---	↑	↑	↑	↑	↑
	CMP.L @aa:16,@-ERd	L	4	5	7								D	S			ERd32-4→ERd32	@ERd	-	@aa:16				---	↑	↑	↑	↑	↑
	CMP.L @aa:16,@(d:2,ERd)	L	4	5	7								D	S				@(4/8/12+ERd)	-	@aa:16				---	↑	↑	↑	↑	↑
	CMP.L @aa:16,@(d:16,ERd)	L	5	5	7								D	S				@(d:16+ERd)	-	@aa:16				---	↑	↑	↑	↑	↑
	CMP.L @aa:16,@(d:32,ERd)	L	6	5	8								D	S				@(d:32+ERd)	-	@aa:16				---	↑	↑	↑	↑	↑
	CMP.L @aa:16,@(d:16,Rd.B)	L	5	5	7								D	S				@(d:16+Rd<<2)	-	@aa:16				---	↑	↑	↑	↑	↑
	CMP.L @aa:16,@(d:16,Rd.W)	L	5	5	7								D	S				@(d:16+Rd<<2)	-	@aa:16				---	↑	↑	↑	↑	↑
	CMP.L @aa:16,@(d:16,ERd.L)	L	5	5	7								D	S				@(d:16+ERd<<2)	-	@aa:16				---	↑	↑	↑	↑	↑



Arithmetic operations (76)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode								Operation ^a					Condition Codes ^b											
				Min.	16-Bit Instruction Fetch Execution Status ^c	#xx	Rn	@ERn	@(d, ERn)	@ (d, Rn, L, ERn, W, ERn, L)	@ ERn / @ ERn+ / @ ERn+ / @ ERn	@ aa:8 / @ aa:16 / @ aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C					
				Number of	Execution Status ^c																							
CMP	CMP.L @aa:16, @(d:32, Rd, B)	L	6	5	8				D	S				@(d:32+RdL<<2) - @aa:16							—	↓	↓	↓	↓	↓	↓	
	CMP.L @aa:16, @(d:32, Rd, W)	L	6	5	8				D	S				@(d:32+Rd<<2) - @aa:16							—	↓	↓	↓	↓	↓	↓	
	CMP.L @aa:16, @(d:32, ERd, L)	L	6	5	8				D	S				@(d:32+ERd<<2) - @aa:16							—	↓	↓	↓	↓	↓	↓	
	CMP.L @aa:16, @aa:16	L	5	4	7						S				@aa:16 - @aa:16							—	↓	↓	↓	↓	↓	↓
	CMP.L @aa:16, @aa:32	L	6	4	8						S				@aa:32 - @aa:16							—	↓	↓	↓	↓	↓	↓
	CMP.L @aa:32, @ERd	L	5	4	7				D		S				@ERd - @aa:32							—	↓	↓	↓	↓	↓	↓
	CMP.L @aa:32, @ERd+	L	5	4	7					D	S				@ERd - @aa:32						ERd32+4→ERd32	—	↓	↓	↓	↓	↓	↓
	CMP.L @aa:32, @ERd-	L	5	4	7					D	S				@ERd - @aa:32						ERd32-4→ERd32	—	↓	↓	↓	↓	↓	↓
	CMP.L @aa:32, @+ERd	L	5	5	8						D	S		ERd32+4→ERd32	@ERd - @aa:32							—	↓	↓	↓	↓	↓	↓
	CMP.L @aa:32, @-ERd	L	5	5	8						D	S		ERd32-4→ERd32	@ERd - @aa:32							—	↓	↓	↓	↓	↓	↓
	CMP.L @aa:32, @(d:2, ERd)	L	5	5	8				D		S				@(4/8/12+ERd) - @aa:32							—	↓	↓	↓	↓	↓	↓
	CMP.L @aa:32, @(d:16, ERd)	L	6	5	8				D		S				@(d:16+ERd) - @aa:32							—	↓	↓	↓	↓	↓	↓
	CMP.L @aa:32, @(d:32, ERd)	L	7	5	9				D		S				@(d:32+ERd) - @aa:32							—	↓	↓	↓	↓	↓	↓
	CMP.L @aa:32, @(d:16, Rd, B)	L	6	5	8				D	S					@(d:16+RdL<<2) - @aa:32							—	↓	↓	↓	↓	↓	↓
	CMP.L @aa:32, @(d:16, Rd, W)	L	6	5	8					D	S				@(d:16+Rd<<2) - @aa:32							—	↓	↓	↓	↓	↓	↓
	CMP.L @aa:32, @(d:16, ERd, L)	L	6	5	8				D	S					@(d:16+ERd<<2) - @aa:32							—	↓	↓	↓	↓	↓	↓
	CMP.L @aa:32, @(d:32, Rd, B)	L	7	5	9				D	S					@(d:32+RdL<<2) - @aa:32							—	↓	↓	↓	↓	↓	↓
	CMP.L @aa:32, @(d:32, Rd, W)	L	7	5	9					D	S				@(d:32+Rd<<2) - @aa:32							—	↓	↓	↓	↓	↓	↓
	CMP.L @aa:32, @(d:32, ERd, L)	L	7	5	9				D	S					@(d:32+ERd<<2) - @aa:32							—	↓	↓	↓	↓	↓	↓
	CMP.L @aa:32, @aa:16	L	6	4	8						S				@aa:16 - @aa:32							—	↓	↓	↓	↓	↓	↓
	CMP.L @aa:32, @aa:32	L	7	4	9						S				@aa:32 - @aa:32							—	↓	↓	↓	↓	↓	↓
	SUB	SUB.B #xx:8, @ERd	B	2	3	4	S			D					@ERd - #xx → @ERd							—	—	↓	↓	↓	↓	↓
		SUB.B #xx:8, @ERd+	B	3	4	5	S				D				@ERd - #xx → @ERd						ERd32+1→ERd32	—	↓	↓	↓	↓	↓	↓
		SUB.B #xx:8, @ERd-	B	3	4	5	S				D				@ERd - #xx → @ERd						ERd32-1→ERd32	—	↓	↓	↓	↓	↓	↓
SUB.B #xx:8, @+ERd		B	3	5	5	S					D			ERd32+1→ERd32	@ERd - #xx → @ERd							—	↓	↓	↓	↓	↓	

Arithmetic operations (77)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode										Operation ⁵					Condition Codes ^{6,7}											
				Min.	16-Bit Instruction Fetch Execution Stages ¹			Execution Stages ¹																						
				5	5	5	#xx	Rn	@ERn	@(d,ERn)	@(d,Rn,LRn,WRn,L)	@ERn/@ERn+/@ERn+/@ERn+/@ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C						
SUB	SUB.B #xx:8,@-ERd	B	3	5	5	S					D						ERd32-1→ERd32	@ERd	- #xx	→ @ERd				—	↕	↕	↕	↕	↕	
	SUB.B #xx:8,@(d:2,ERd)	B	3	5	5	S				D								@(1/2/3+ERd)	- #xx	→ @(1/2/3+ERd)				—	↕	↕	↕	↕	↕	
	SUB.B #xx:8,@(d:16,ERd)	B	4	5	6	S				D								@(d:16+ERd)	- #xx	→ @(d:16+ERd)				—	↕	↕	↕	↕	↕	
	SUB.B #xx:8,@(d:32,ERd)	B	5	5	7	S				D								@(d:32+ERd)	- #xx	→ @(d:32+ERd)				—	↕	↕	↕	↕	↕	
	SUB.B #xx:8,@(d:16,Rd,B)	B	4	5	6	S				D								@(d:16+RdL)	- #xx	→ @(d:16+RdL)				—	↕	↕	↕	↕	↕	
	SUB.B #xx:8,@(d:16,Rd,W)	B	4	5	6	S				D								@(d:16+Rd)	- #xx	→ @(d:16+Rd)				—	↕	↕	↕	↕	↕	
	SUB.B #xx:8,@(d:16,ERd,L)	B	4	5	6	S				D								@(d:16+ERd)	- #xx	→ @(d:16+ERd)				—	↕	↕	↕	↕	↕	
	SUB.B #xx:8,@(d:32,Rd,B)	B	5	5	7	S				D								@(d:32+RdL)	- #xx	→ @(d:32+RdL)				—	↕	↕	↕	↕	↕	
	SUB.B #xx:8,@(d:32,Rd,W)	B	5	5	7	S				D								@(d:32+Rd)	- #xx	→ @(d:32+Rd)				—	↕	↕	↕	↕	↕	
	SUB.B #xx:8,@(d:32,ERd,L)	B	5	5	7	S				D								@(d:32+ERd)	- #xx	→ @(d:32+ERd)				—	↕	↕	↕	↕	↕	
	SUB.B #xx:8,@aa:8	B	2	3	4	S							D					@aa:8	- #xx	→ @aa:8				—	↕	↕	↕	↕	↕	
	SUB.B #xx:8,@aa:16	B	3	3	5	S							D					@aa:16	- #xx	→ @aa:16				—	↕	↕	↕	↕	↕	
	SUB.B #xx:8,@aa:32	B	4	3	6	S							D					@aa:32	- #xx	→ @aa:32				—	↕	↕	↕	↕	↕	
	SUB.B Rs,Rd	B	1	1	1	S												Rd8	- Rs8	→ Rd8				—	↕	↕	↕	↕	↕	
	SUB.B Rs,@ERd	B	2	3	4	S	D											@ERd	- Rs8	→ @ERd				—	↕	↕	↕	↕	↕	
	SUB.B Rs,@ERd+	B	2	4	4	S					D							@ERd	- Rs8	→ @ERd				ERd32+1→ERd32	—	↕	↕	↕	↕	↕
	SUB.B Rs,@ERd-	B	2	4	4	S					D							@ERd	- Rs8	→ @ERd				ERd32-1→ERd32	—	↕	↕	↕	↕	↕
	SUB.B Rs,@+ERd	B	2	5	5	S					D							ERd32+1→ERd32	@ERd	- Rs8	→ @ERd				—	↕	↕	↕	↕	↕
	SUB.B Rs,@-ERd	B	2	5	5	S					D							ERd32-1→ERd32	@ERd	- Rs8	→ @ERd				—	↕	↕	↕	↕	↕
	SUB.B Rs,@(d:2,ERd)	B	2	5	5	S				D								@(1/2/3+ERd)	- Rs8	→ @(1/2/3+ERd)				—	↕	↕	↕	↕	↕	
	SUB.B Rs,@(d:16,ERd)	B	3	5	5	S	D											@(d:16+ERd)	- Rs8	→ @(d:16+ERd)				—	↕	↕	↕	↕	↕	
	SUB.B Rs,@(d:32,ERd)	B	4	5	6	S				D								@(d:32+ERd)	- Rs8	→ @(d:32+ERd)				—	↕	↕	↕	↕	↕	
	SUB.B Rs,@(d:16,Rd,B)	B	3	5	5	S				D								@(d:16+RdL)	- Rs8	→ @(d:16+RdL)				—	↕	↕	↕	↕	↕	
	SUB.B Rs,@(d:16,Rd,W)	B	3	5	5	S				D								@(d:16+Rd)	- Rs8	→ @(d:16+Rd)				—	↕	↕	↕	↕	↕	
	SUB.B Rs,@(d:16,ERd,L)	B	3	5	5	S				D								@(d:16+ERd)	- Rs8	→ @(d:16+ERd)				—	↕	↕	↕	↕	↕	

Arithmetic operations (78)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode									Operation ^s					Condition Codes ^{s2}							
				Min.	Max.	16-Bit Instruction Fetch Execution Status ^{s1}	#xxx	Rn	@ERn	@(d,ERn)	@(d,Rn,W,ERn,L)	@.ERn/@.ERn+/@.ERn+/@.ERn+/@.ERn	@.aa:8/@.aa:16/@.aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C	
				4	5	6	S		D	D	D	D	D						—	↑	↓	↑	↓	↑	↓
SUB	SUB.B Rs, @(d:32,Rd,B)	B	4	5	6	S		D							@(d:32+RdL) - Rs8 → @(d:32+RdL)										
	SUB.B Rs, @(d:32,Rd,W)	B	4	5	6	S		D							@(d:32+Rd) - Rs8 → @(d:32+Rd)										
	SUB.B Rs, @(d:32,ERd,L)	B	4	5	6	S		D							@(d:32+ERd) - Rs8 → @(d:32+ERd)										
	SUB.B Rs, @aa:8	B	2	3	4	S					D				@aa:8 - Rs8 → @aa:8										
	SUB.B Rs, @aa:16	B	3	3	5	S					D				@aa:16 - Rs8 → @aa:16										
	SUB.B Rs, @aa:32	B	4	3	6	S					D				@aa:32 - Rs8 → @aa:32										
	SUB.B @ERs,Rd	B	2	3	3	D	S								Rd8 - @ERs → Rd8										
	SUB.B @ERs+,Rd	B	2	3	3	D			S						Rd8 - @ERs → Rd8	ERs32+1→ERs32									
	SUB.B @ERs-,Rd	B	2	3	3	D			S						Rd8 - @ERs → Rd8	ERs32-1→ERs32									
	SUB.B @+ERs,Rd	B	2	4	4	D			S					ERs32+1→ERs32	Rd8 - @ERs → Rd8										
	SUB.B @-ERs,Rd	B	2	4	4	D			S					ERs32-1→ERs32	Rd8 - @ERs → Rd8										
	SUB.B @(d:2,ERs),Rd	B	2	4	4	D	S								Rd8 - @(1/2/3+ERs) → Rd8										
	SUB.B @(d:16,ERs),Rd	B	3	4	4	D	S								Rd8 - @(d:16+ERs) → Rd8										
	SUB.B @(d:32,ERs),Rd	B	4	4	5	D	S								Rd8 - @(d:32+ERs) → Rd8										
	SUB.B @(d:16,Rs,B),Rd	B	3	4	4	D		S							Rd8 - @(d:16+Rs) → Rd8										
	SUB.B @(d:16,Rs,W),Rd	B	3	4	4	D		S							Rd8 - @(d:16+Rs) → Rd8										
	SUB.B @(d:16,ERs,L),Rd	B	3	4	4	D		S							Rd8 - @(d:16+ERs) → Rd8										
	SUB.B @(d:32,Rs,B),Rd	B	4	4	5	D		S							Rd8 - @(d:32+RsL) → Rd8										
	SUB.B @(d:32,Rs,W),Rd	B	4	4	5	D		S							Rd8 - @(d:32+Rs) → Rd8										
	SUB.B @(d:32,ERs,L),Rd	B	4	4	5	D		S							Rd8 - @(d:32+ERs) → Rd8										
	SUB.B @aa:8,Rd	B	2	3	3	D				S					Rd8 - @aa:8 → Rd8										
	SUB.B @aa:16,Rd	B	3	3	4	D				S					Rd8 - @aa:16 → Rd8										
	SUB.B @aa:32,Rd	B	4	3	5	D				S					Rd8 - @aa:32 → Rd8										
	SUB.B @ERs,@ERd	B	2	4	5		S								@ERd - @ERs → @ERd										
	SUB.B @ERs,@ERd+	B	2	4	5		S			D					@ERd - @ERs → @ERd	ERd32+1→ERd32									

Arithmetic operations (79)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode							Operation ⁵					Condition Code ^{6,7}											
				Min.	Max.	16-Bit Instruction Fetch Execution Stages ¹	Prx	Rn	@(d,ERn)	@(d,Rn,LRn,WERn,L)	@(d,ERn)/@ERn+/@ERn+/@ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C				
SUB	SUB.B @ERs,@ERd-	B	2	4	5			S		D						@ERd - @ERs → @ERd			ERd32-1→ERd32	---	↑	↑	↑	↑	↑		
	SUB.B @ERs,@+ERd	B	2	5	6			S		D						ERd32+1→ERd32	@ERd - @ERs → @ERd				---	↑	↑	↑	↑	↑	
	SUB.B @ERs,@-ERd	B	2	5	6			S		D						ERd32-1→ERd32	@ERd - @ERs → @ERd				---	↑	↑	↑	↑	↑	
	SUB.B @ERs,@(d:2,ERd)	B	2	5	6			S	D							@(1/2/3+ERd) - @ERs → @(1/2/3+ERd)				---	↑	↑	↑	↑	↑		
	SUB.B @ERs,@(d:16,ERd)	B	3	5	6			S	D							@(d:16+ERd) - @ERs → @(d:16+ERd)				---	↑	↑	↑	↑	↑		
	SUB.B @ERs,@(d:32,ERd)	B	4	5	7			S	D							@(d:32+ERd) - @ERs → @(d:32+ERd)				---	↑	↑	↑	↑	↑		
	SUB.B @ERs,@(d:16,Rd.B)	B	3	5	6			S		D						@(d:16+RdL) - @ERs → @(d:16+RdL)				---	↑	↑	↑	↑	↑		
	SUB.B @ERs,@(d:16,Rd.W)	B	3	5	6			S	D							@(d:16+Rd) - @ERs → @(d:16+Rd)				---	↑	↑	↑	↑	↑		
	SUB.B @ERs,@(d:16,ERd.L)	B	3	5	6			S	D							@(d:16+ERdL) - @ERs → @(d:16+ERdL)				---	↑	↑	↑	↑	↑		
	SUB.B @ERs,@(d:32,Rd.B)	B	4	5	7			S		D						@(d:32+RdL) - @ERs → @(d:32+RdL)				---	↑	↑	↑	↑	↑		
	SUB.B @ERs,@(d:32,Rd.W)	B	4	5	7			S	D							@(d:32+Rd) - @ERs → @(d:32+Rd)				---	↑	↑	↑	↑	↑		
	SUB.B @ERs,@(d:32,ERd.L)	B	4	5	7			S	D							@(d:32+ERdL) - @ERs → @(d:32+ERdL)				---	↑	↑	↑	↑	↑		
	SUB.B @ERs,@aa:16	B	3	4	6			S			D					@aa:16 - @ERs → @aa:16				---	↑	↑	↑	↑	↑		
	SUB.B @ERs,@aa:32	B	4	4	7			S			D					@aa:32 - @ERs → @aa:32				---	↑	↑	↑	↑	↑		
	SUB.B @ERs+,@ERd	B	3	5	6				D		S					@ERd - @ERs → @ERd	ERs32+1→ERs32				---	↑	↑	↑	↑	↑	
	SUB.B @ERs+,@ERd+	B	3	5	6						S					@ERd - @ERs → @ERd	ERs32+1→ERs32	ERd32+1→ERd32				---	↑	↑	↑	↑	↑
	SUB.B @ERs+,@ERd-	B	3	5	6						S					@ERd - @ERs → @ERd	ERs32+1→ERs32	ERd32-1→ERd32				---	↑	↑	↑	↑	↑
	SUB.B @ERs+,@+ERd	B	3	6	7						S					ERd32+1→ERd32	@ERd - @ERs → @ERd				---	↑	↑	↑	↑	↑	
	SUB.B @ERs+,@-ERd	B	3	6	7						S					ERd32-1→ERd32	@ERd - @ERs → @ERd				---	↑	↑	↑	↑	↑	
	SUB.B @ERs+,@(d:2,ERd)	B	3	6	7				D		S					@(1/2/3+ERd) - @ERs → @(1/2/3+ERd)	ERs32+1→ERs32				---	↑	↑	↑	↑	↑	
SUB.B @ERs+,@(d:16,ERd)	B	4	6	7				D		S					@(d:16+ERd) - @ERs → @(d:16+ERd)	ERs32+1→ERs32				---	↑	↑	↑	↑	↑		
SUB.B @ERs+,@(d:32,ERd)	B	5	6	8				D		S					@(d:32+ERd) - @ERs → @(d:32+ERd)	ERs32+1→ERs32				---	↑	↑	↑	↑	↑		
SUB.B @ERs+,@(d:16,Rd.B)	B	4	6	7					D	S					@(d:16+RdL) - @ERs → @(d:16+RdL)	ERs32+1→ERs32				---	↑	↑	↑	↑	↑		
SUB.B @ERs+,@(d:16,Rd.W)	B	4	6	7						S					@(d:16+Rd) - @ERs → @(d:16+Rd)	ERs32+1→ERs32				---	↑	↑	↑	↑	↑		
SUB.B @ERs+,@(d:16,ERd.L)	B	4	6	7						S					@(d:16+ERdL) - @ERs → @(d:16+ERdL)	ERs32+1→ERs32				---	↑	↑	↑	↑	↑		

Arithmetic operations (80)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹	Addressing Mode						Operation ⁵					Condition Codes ²						
						#px	Rn	@ERn	@(d,ERn)	@(d,Rn,ERn,W,ERn,L)	@.ERn/@.ERn+/@.ERn+/@.ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C
SUB	SUB.B @ERs+,@(d:32,Rd,B)	B	5	6	8				D	S				@(d:32+RdL) - @ERs → @(d:32+RdL)	ERs32+1→ERs32		—	↓	↑	↓	↑	↓	↑
	SUB.B @ERs+,@(d:32,Rd,W)	B	5	6	8				D	S				@(d:32+Rd) - @ERs → @(d:32+Rd)	ERs32+1→ERs32		—	↓	↑	↓	↑	↓	↑
	SUB.B @ERs+,@(d:32,ERd,L)	B	5	6	8				D	S				@(d:32+ERd) - @ERs → @(d:32+ERd)	ERs32+1→ERs32		—	↓	↑	↓	↑	↓	↑
	SUB.B @ERs+,@aa:16	B	4	5	7						S	D		@aa:16 - @ERs → @aa:16	ERs32+1→ERs32		—	↓	↑	↓	↑	↓	↑
	SUB.B @ERs+,@aa:32	B	5	5	8						S	D		@aa:32 - @ERs → @aa:32	ERs32+1→ERs32		—	↓	↑	↓	↑	↓	↑
	SUB.B @ERs-,@ERd	B	3	5	6				D		S			@ERd - @ERs → @ERd	ERs32-1→ERs32		—	↓	↑	↓	↑	↓	↑
	SUB.B @ERs-,@ERd+	B	3	5	6						S			@ERd - @ERs → @ERd	ERs32-1→ERs32	ERd32+1→ERd32	—	↓	↑	↓	↑	↓	↑
	SUB.B @ERs-,@ERd-	B	3	5	6						S			@ERd - @ERs → @ERd	ERs32-1→ERs32	ERd32-1→ERd32	—	↓	↑	↓	↑	↓	↑
	SUB.B @ERs-,@+ERd	B	3	6	7						S		ERd32+1→ERd32	@ERd - @ERs → @ERd	ERs32-1→ERs32		—	↓	↑	↓	↑	↓	↑
	SUB.B @ERs-,@-ERd	B	3	6	7						S		ERd32-1→ERd32	@ERd - @ERs → @ERd	ERs32-1→ERs32		—	↓	↑	↓	↑	↓	↑
	SUB.B @ERs-,@(d:2,ERd)	B	3	6	7				D	S				@(1/2/3+ERd) - @ERs → @(1/2/3+ERd)	ERs32-1→ERs32		—	↓	↑	↓	↑	↓	↑
	SUB.B @ERs-,@(d:16,ERd)	B	4	6	7				D	S				@(d:16+ERd) - @ERs → @(d:16+ERd)	ERs32-1→ERs32		—	↓	↑	↓	↑	↓	↑
	SUB.B @ERs-,@(d:32,ERd)	B	5	6	8				D	S				@(d:32+ERd) - @ERs → @(d:32+ERd)	ERs32-1→ERs32		—	↓	↑	↓	↑	↓	↑
	SUB.B @ERs-,@(d:16,Rd,B)	B	4	6	7				D	S				@(d:16+RdL) - @ERs → @(d:16+RdL)	ERs32-1→ERs32		—	↓	↑	↓	↑	↓	↑
	SUB.B @ERs-,@(d:16,Rd,W)	B	4	6	7				D	S				@(d:16+Rd) - @ERs → @(d:16+Rd)	ERs32-1→ERs32		—	↓	↑	↓	↑	↓	↑
	SUB.B @ERs-,@(d:16,ERd,L)	B	4	6	7				D	S				@(d:16+ERd) - @ERs → @(d:16+ERd)	ERs32-1→ERs32		—	↓	↑	↓	↑	↓	↑
	SUB.B @ERs-,@(d:32,Rd,B)	B	5	6	8				D	S				@(d:32+RdL) - @ERs → @(d:32+RdL)	ERs32-1→ERs32		—	↓	↑	↓	↑	↓	↑
	SUB.B @ERs-,@(d:32,Rd,W)	B	5	6	8				D	S				@(d:32+Rd) - @ERs → @(d:32+Rd)	ERs32-1→ERs32		—	↓	↑	↓	↑	↓	↑
	SUB.B @ERs-,@(d:32,ERd,L)	B	5	6	8				D	S				@(d:32+ERd) - @ERs → @(d:32+ERd)	ERs32-1→ERs32		—	↓	↑	↓	↑	↓	↑
	SUB.B @ERs-,@aa:16	B	4	5	7						S	D		@aa:16 - @ERs → @aa:16	ERs32-1→ERs32		—	↓	↑	↓	↑	↓	↑
	SUB.B @ERs-,@aa:32	B	5	5	8						S	D		@aa:32 - @ERs → @aa:32	ERs32-1→ERs32		—	↓	↑	↓	↑	↓	↑
	SUB.B @+ERs,@ERd	B	3	6	6				D		S		ERs32+1→ERs32	@ERd - @ERs → @ERd			—	↓	↑	↓	↑	↓	↑
	SUB.B @+ERs,@ERd+	B	3	6	6						S		ERs32+1→ERs32	@ERd - @ERs → @ERd		ERd32+1→ERd32	—	↓	↑	↓	↑	↓	↑
	SUB.B @+ERs,@ERd-	B	3	6	6						S		ERs32+1→ERs32	@ERd - @ERs → @ERd		ERd32-1→ERd32	—	↓	↑	↓	↑	↓	↑
	SUB.B @+ERs,@+ERd	B	3	7	7						S		ERs32+1→ERs32	@ERd - @ERs → @ERd		ERd32+1→ERd32	—	↓	↑	↓	↑	↓	↑



Arithmetic operations (82)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode										Operation ⁶					Condition Codes ²																					
				Min.	16-Bit Instruction Fetch Execution Status ¹ #xx	Rn	@ERN	@ (d, ERn)	@ (d, Rn, L, W, ERn, L)	@ ERn/ @ ERn+/ @ ERn+/ @ ERn+/ @ ERn	@ aa:9/ @ aa:16/ @ aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C																		
																							Number of	Execution Status ¹																
SUB	SUB.B @-ERs, @ (d:32, ERd, L)	B	5	7	8				D	S				ERs32-1→ERs32		@ (d:32+ERd) - @ERs → @ (d:32+ERd)																								
	SUB.B @-ERs, @aa:16	B	4	6	7						S	D		ERs32-1→ERs32		@aa:16 - @ERs → @aa:16																								
	SUB.B @-ERs, @aa:32	B	5	6	8								S	D	ERs32-1→ERs32		@aa:32 - @ERs → @aa:32																							
	SUB.B @ (d:2, ERs), @ERd	B	3	6	6				D	S						@ERd - @ (1/2/3+ERs) → @ERd																								
	SUB.B @ (d:2, ERs), @ERd+	B	3	6	6					S	D					@ERd - @ (1/2/3+ERs) → @ERd					ERd32+1→ERd32																			
	SUB.B @ (d:2, ERs), @ERd-	B	3	6	6					S	D					@ERd - @ (1/2/3+ERs) → @ERd					ERd32-1→ERd32																			
	SUB.B @ (d:2, ERs), @+ERd	B	3	7	7					S	D			ERd32+1→ERd32	@ERd - @ (1/2/3+ERs) → @ERd																									
	SUB.B @ (d:2, ERs), @-ERd	B	3	7	7					S	D			ERd32-1→ERd32	@ERd - @ (1/2/3+ERs) → @ERd																									
	SUB.B @ (d:2, ERs), @ (d:2, ERd)	B	3	7	7					S					@ (1/2/3+ERd) - @ (1/2/3+ERs) → @ (1/2/3+ERd)																									
	SUB.B @ (d:2, ERs), @ (d:16, ERd)	B	4	7	7					S					@ (d:16+ERd) - @ (1/2/3+ERs) → @ (d:16+ERd)																									
	SUB.B @ (d:2, ERs), @ (d:32, ERd)	B	5	7	8					S					@ (d:32+ERd) - @ (1/2/3+ERs) → @ (d:32+ERd)																									
	SUB.B @ (d:2, ERs), @ (d:16, Rd, B)	B	4	7	7					S	D				@ (d:16+RdL) - @ (1/2/3+ERs) → @ (d:16+RdL)																									
	SUB.B @ (d:2, ERs), @ (d:16, Rd, W)	B	4	7	7					S	D				@ (d:16+Rd) - @ (1/2/3+ERs) → @ (d:16+Rd)																									
	SUB.B @ (d:2, ERs), @ (d:16, ERd, L)	B	4	7	7					S	D				@ (d:16+ERd) - @ (1/2/3+ERs) → @ (d:16+ERd)																									
	SUB.B @ (d:2, ERs), @ (d:32, Rd, B)	B	5	7	8					S	D				@ (d:32+RdL) - @ (1/2/3+ERs) → @ (d:32+RdL)																									
	SUB.B @ (d:2, ERs), @ (d:32, Rd, W)	B	5	7	8					S	D				@ (d:32+Rd) - @ (1/2/3+ERs) → @ (d:32+Rd)																									
	SUB.B @ (d:2, ERs), @ (d:32, ERd, L)	B	5	7	8					S	D				@ (d:32+ERd) - @ (1/2/3+ERs) → @ (d:32+ERd)																									
	SUB.B @ (d:2, ERs), @aa:16	B	4	6	7					S		D			@aa:16 - @ (1/2/3+ERs) → @aa:16																									
	SUB.B @ (d:2, ERs), @aa:32	B	5	6	8					S		D			@aa:32 - @ (1/2/3+ERs) → @aa:32																									
	SUB.B @ (d:16, ERs), @ERd	B	4	6	7				D	S					@ERd - @ (d:16+ERs) → @ERd																									
	SUB.B @ (d:16, ERs), @ERd+	B	4	6	7					S	D				@ERd - @ (d:16+ERs) → @ERd						ERd32+1→ERd32																			
	SUB.B @ (d:16, ERs), @ERd-	B	4	6	7					S	D				@ERd - @ (d:16+ERs) → @ERd						ERd32-1→ERd32																			
	SUB.B @ (d:16, ERs), @+ERd	B	4	7	8					S	D			ERd32+1→ERd32	@ERd - @ (d:16+ERs) → @ERd																									
	SUB.B @ (d:16, ERs), @-ERd	B	4	7	8					S	D			ERd32-1→ERd32	@ERd - @ (d:16+ERs) → @ERd																									
	SUB.B @ (d:16, ERs), @ (d:2, ERd)	B	4	7	8					S					@ (1/2/3+ERd) - @ (d:16+ERs) → @ (1/2/3+ERd)																									



Arithmetic operations (104)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹	Addressing Mode										Operation ⁵					Condition Codes ²																				
						16-Bit Instruction Fetch Execution Stages ¹	#xx	Rn	@ERn	@(d,ERn)	@(d,Rn,ERn,WERn,L)	@ERn/@ERn+/@ERn+/@+ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2												Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C						
SUB	SUB.L @(d:16,ERs),ERd	L	3	4	4		D	S										ERd32	-	@(d:16+ERs)	→	ERd32																			
	SUB.L @(d:32,ERs),ERd	L	4	4	5		D	S										ERd32	-	@(d:32+ERs)	→	ERd32																			
	SUB.L @(d:16,Rs,B),ERd	L	3	4	4		D	S										ERd32	-	@(d:16+RsL<<2)	→	ERd32																			
	SUB.L @(d:16,Rs,W),ERd	L	3	4	4		D	S										ERd32	-	@(d:16+Rs<<2)	→	ERd32																			
	SUB.L @(d:16,ERs,L),ERd	L	3	4	4		D	S										ERd32	-	@(d:16+ERs<<2)	→	ERd32																			
	SUB.L @(d:32,Rs,B),ERd	L	4	4	5		D	S										ERd32	-	@(d:32+RsL<<2)	→	ERd32																			
	SUB.L @(d:32,Rs,W),ERd	L	4	4	5		D	S										ERd32	-	@(d:32+Rs<<2)	→	ERd32																			
	SUB.L @(d:32,ERs,L),ERd	L	4	4	5		D	S										ERd32	-	@(d:32+ERs<<2)	→	ERd32																			
	SUB.L @aa:16,ERd	L	3	3	4		D						S					ERd32	-	@aa:16	→	ERd32																			
	SUB.L @aa:32,ERd	L	4	3	5		D						S					ERd32	-	@aa:32	→	ERd32																			
	SUB.L @ERs,@ERd	L	3	5	6			S										@ERd	-	@ERs	→	@ERd																			
	SUB.L @ERs,@ERd+	L	3	5	6			S			D							@ERd	-	@ERs	→	@ERd		ERd32+4	→	ERd32															
	SUB.L @ERs,@ERd-	L	3	5	6			S			D							@ERd	-	@ERs	→	@ERd		ERd32-4	→	ERd32															
	SUB.L @ERs,@+ERd	L	3	6	7			S			D							ERd32+4	→	ERd32	@ERd	-	@ERs	→	@ERd																
	SUB.L @ERs,@-ERd	L	3	6	7			S			D							ERd32-4	→	ERd32	@ERd	-	@ERs	→	@ERd																
	SUB.L @ERs,@(d:2,ERd)	L	3	6	7			S	D									@(4/8/12+ERd)	-	@ERs	→	@(4/8/12+ERd)																			
	SUB.L @ERs,@(d:16,ERd)	L	4	6	7			S	D									@(d:16+ERd)	-	@ERs	→	@(d:16+ERd)																			
	SUB.L @ERs,@(d:32,ERd)	L	5	6	8			S	D									@(d:32+ERd)	-	@ERs	→	@(d:32+ERd)																			
	SUB.L @ERs,@(d:16,Rd,B)	L	4	6	7			S	D									@(d:16+RdL<<2)	-	@ERs	→	@(d:16+RdL<<2)																			
	SUB.L @ERs,@(d:16,Rd,W)	L	4	6	7			S	D									@(d:16+Rd<<2)	-	@ERs	→	@(d:16+Rd<<2)																			
	SUB.L @ERs,@(d:16,ERd,L)	L	4	6	7			S	D									@(d:16+ERd<<2)	-	@ERs	→	@(d:16+ERd<<2)																			
	SUB.L @ERs,@(d:32,Rd,B)	L	5	6	8			S	D									@(d:32+RdL<<2)	-	@ERs	→	@(d:32+RdL<<2)																			
	SUB.L @ERs,@(d:32,Rd,W)	L	5	6	8			S	D									@(d:32+Rd<<2)	-	@ERs	→	@(d:32+Rd<<2)																			
	SUB.L @ERs,@(d:32,ERd,L)	L	5	6	8			S	D									@(d:32+ERd<<2)	-	@ERs	→	@(d:32+ERd<<2)																			
	SUB.L @ERs,@aa:16	L	4	5	7			S					D					@aa:16	-	@ERs	→	@aa:16																			



Arithmetic operations (106)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode										Operation ⁵					Condition Codes ^{6,7}								
				Min.	16-Bit Instruction Fetch Execution Status ¹	Instruction Fetch	16-Bit Instruction Fetch Execution Status ¹	Rn	@(d,ERn)	@(d,Rn,WRn,LL)	@(d,ERn)/@ERn+/@ERn+/@ERn	@aa:9/@aa:16/@aa:32	Operation 1												Operation 2	Operation 3	Operation 4
SUB	SUB.L @ERs, @(d:16,Rd,B)	L	4	6	7							D	S				@(d:16+RdL<<2) - @ERs → @(d:16+RdL<<2)	ERs32-4→ERs32			—	↓	↑	↓	↑	↓	↑
	SUB.L @ERs, @(d:16,Rd,W)	L	4	6	7							D	S				@(d:16+Rd<<2) - @ERs → @(d:16+Rd<<2)	ERs32-4→ERs32			—	↓	↑	↓	↑	↓	
	SUB.L @ERs, @(d:16,ERd,L)	L	4	6	7							D	S				@(d:16+ERd<<2) - @ERs → @(d:16+ERd<<2)	ERs32-4→ERs32			—	↓	↑	↓	↑	↓	
	SUB.L @ERs, @(d:32,Rd,B)	L	5	6	8							D	S				@(d:32+RdL<<2) - @ERs → @(d:32+RdL<<2)	ERs32-4→ERs32			—	↓	↑	↓	↑	↓	
	SUB.L @ERs, @(d:32,Rd,W)	L	5	6	8							D	S				@(d:32+Rd<<2) - @ERs → @(d:32+Rd<<2)	ERs32-4→ERs32			—	↓	↑	↓	↑	↓	
	SUB.L @ERs, @(d:32,ERd,L)	L	5	6	8							D	S				@(d:32+ERd<<2) - @ERs → @(d:32+ERd<<2)	ERs32-4→ERs32			—	↓	↑	↓	↑	↓	
	SUB.L @ERs, @aa:16	L	4	5	7								S	D				@aa:16 - @ERs → @aa:16	ERs32-4→ERs32			—	↓	↑	↓	↑	↓
	SUB.L @ERs, @aa:32	L	5	5	8								S	D				@aa:32 - @ERs → @aa:32	ERs32-4→ERs32			—	↓	↑	↓	↑	↓
	SUB.L @+ERs, @ERd	L	3	6	6							D	S		ERs32+4→ERs32	@ERd	- @ERs → @ERd				—	↓	↑	↓	↑	↓	↑
	SUB.L @+ERs, @ERd+	L	3	6	6								S		ERs32+4→ERs32	@ERd	- @ERs → @ERd		ERd32+4→ERd32			—	↓	↑	↓	↑	↓
	SUB.L @+ERs, @ERd-	L	3	6	6								S		ERs32+4→ERs32	@ERd	- @ERs → @ERd		ERd32-4→ERd32			—	↓	↑	↓	↑	↓
	SUB.L @+ERs, @+ERd	L	3	7	7								S		ERs32+4→ERs32	ERd32+4→ERd32	@ERd	- @ERs → @ERd				—	↓	↑	↓	↑	↓
	SUB.L @+ERs, @-ERd	L	3	7	7								S		ERs32+4→ERs32	ERd32-4→ERd32	@ERd	- @ERs → @ERd				—	↓	↑	↓	↑	↓
	SUB.L @+ERs, @(d:2,ERd)	L	3	7	7							D	S		ERs32+4→ERs32	@(4/8/12+ERd)	- @ERs → @(4/8/12+ERd)				—	↓	↑	↓	↑	↓	
	SUB.L @+ERs, @(d:16,ERd)	L	4	7	7							D	S		ERs32+4→ERs32	@(d:16+ERd)	- @ERs → @(d:16+ERd)				—	↓	↑	↓	↑	↓	
	SUB.L @+ERs, @(d:32,ERd)	L	5	7	8							D	S		ERs32+4→ERs32	@(d:32+ERd)	- @ERs → @(d:32+ERd)				—	↓	↑	↓	↑	↓	
	SUB.L @+ERs, @(d:16,Rd,B)	L	4	7	7							D	S		ERs32+4→ERs32	@(d:16+RdL<<2) - @ERs → @(d:16+RdL<<2)				—	↓	↑	↓	↑	↓		
	SUB.L @+ERs, @(d:16,Rd,W)	L	4	7	7							D	S		ERs32+4→ERs32	@(d:16+Rd<<2) - @ERs → @(d:16+Rd<<2)				—	↓	↑	↓	↑	↓		
	SUB.L @+ERs, @(d:16,ERd,L)	L	4	7	7							D	S		ERs32+4→ERs32	@(d:16+ERd<<2) - @ERs → @(d:16+ERd<<2)				—	↓	↑	↓	↑	↓		
	SUB.L @+ERs, @(d:32,Rd,B)	L	5	7	8							D	S		ERs32+4→ERs32	@(d:32+RdL<<2) - @ERs → @(d:32+RdL<<2)				—	↓	↑	↓	↑	↓		
SUB.L @+ERs, @(d:32,Rd,W)	L	5	7	8							D	S		ERs32+4→ERs32	@(d:32+Rd<<2) - @ERs → @(d:32+Rd<<2)				—	↓	↑	↓	↑	↓			
SUB.L @+ERs, @(d:32,ERd,L)	L	5	7	8							D	S		ERs32+4→ERs32	@(d:32+ERd<<2) - @ERs → @(d:32+ERd<<2)				—	↓	↑	↓	↑	↓			
SUB.L @+ERs, @aa:16	L	4	6	7								S	D	ERs32+4→ERs32	@aa:16	- @ERs → @aa:16				—	↓	↑	↓	↑	↓		
SUB.L @+ERs, @aa:32	L	5	6	8								S	D	ERs32+4→ERs32	@aa:32	- @ERs → @aa:32				—	↓	↑	↓	↑	↓		



Arithmetic operations (107)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	Number of 16-Bit Instruction Fetch Execution Stages ¹	Addressing Mode						Operation ⁵					Condition Codes ^{6,7}																							
							Rn	@(d, ERn)	@(d, Rn.L, ERn.W, ERn.L)	@.ERn/@.ERn+/@.ERn+/@.ERn+/@.ERn	@aa:9/@.aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C																			
SUB	SUB.L @-ERs, @ERd	L	3	6	6			D							ERs32-4→ERs32		@ERd - @ERs → @ERd																								
	SUB.L @-ERs, @ERd+	L	3	6	6										ERs32-4→ERs32		@ERd - @ERs → @ERd			ERd32+4→ERd32																					
	SUB.L @-ERs, @ERd-	L	3	6	6										ERs32-4→ERs32		@ERd - @ERs → @ERd			ERd32-4→ERd32																					
	SUB.L @-ERs, @+ERd	L	3	7	7										ERs32-4→ERs32	ERd32+4→ERd32	@ERd - @ERs → @ERd																								
	SUB.L @-ERs, @-ERd	L	3	7	7										ERs32-4→ERs32	ERd32-4→ERd32	@ERd - @ERs → @ERd																								
	SUB.L @-ERs, @(d:2,ERd)	L	3	7	7					D	S				ERs32-4→ERs32		@(4/8/12+ERd) - @ERs → @(4/8/12+ERd)																								
	SUB.L @-ERs, @(d:16,ERd)	L	4	7	7					D	S				ERs32-4→ERs32		@(d:16+ERd) - @ERs → @(d:16+ERd)																								
	SUB.L @-ERs, @(d:32,ERd)	L	5	7	8					D	S				ERs32-4→ERs32		@(d:32+ERd) - @ERs → @(d:32+ERd)																								
	SUB.L @-ERs, @(d:16,Rd.B)	L	4	7	7					D	S				ERs32-4→ERs32		@(d:16+RdL<<2) - @ERs → @(d:16+RdL<<2)																								
	SUB.L @-ERs, @(d:16,Rd.W)	L	4	7	7					D	S				ERs32-4→ERs32		@(d:16+RdL<<2) - @ERs → @(d:16+RdL<<2)																								
	SUB.L @-ERs, @(d:16,ERd.L)	L	4	7	7					D	S				ERs32-4→ERs32		@(d:16+ERdL<<2) - @ERs → @(d:16+ERdL<<2)																								
	SUB.L @-ERs, @(d:32,Rd.B)	L	5	7	8					D	S				ERs32-4→ERs32		@(d:32+RdL<<2) - @ERs → @(d:32+RdL<<2)																								
	SUB.L @-ERs, @(d:32,Rd.W)	L	5	7	8					D	S				ERs32-4→ERs32		@(d:32+RdL<<2) - @ERs → @(d:32+RdL<<2)																								
	SUB.L @-ERs, @(d:32,ERd.L)	L	5	7	8					D	S				ERs32-4→ERs32		@(d:32+ERdL<<2) - @ERs → @(d:32+ERdL<<2)																								
	SUB.L @-ERs, @aa:16	L	4	6	7						S	D			ERs32-4→ERs32		@aa:16 - @ERs → @aa:16																								
	SUB.L @-ERs, @aa:32	L	5	6	8						S	D			ERs32-4→ERs32		@aa:32 - @ERs → @aa:32																								
	SUB.L @(d:2,ERs), @ERd	L	3	6	6					D	S						@ERd - @(4/8/12+ERs) → @ERd																								
	SUB.L @(d:2,ERs), @ERd+	L	3	6	6						S	D					@ERd - @(4/8/12+ERs) → @ERd				ERd32+4→ERd32																				
	SUB.L @(d:2,ERs), @ERd-	L	3	6	6						S	D					@ERd - @(4/8/12+ERs) → @ERd				ERd32-4→ERd32																				
	SUB.L @(d:2,ERs), @+ERd	L	3	7	7						S	D				ERd32+4→ERd32	@ERd - @(4/8/12+ERs) → @ERd																								
	SUB.L @(d:2,ERs), @-ERd	L	3	7	7						S	D				ERd32-4→ERd32	@ERd - @(4/8/12+ERs) → @ERd																								
	SUB.L @(d:2,ERs), @(d:2,ERd)	L	3	7	7						S						@(4/8/12+ERd) - @(4/8/12+ERs) → @(4/8/12+ERd)																								
	SUB.L @(d:2,ERs), @(d:16,ERd)	L	4	7	7						S						@(d:16+ERd) - @(4/8/12+ERs) → @(d:16+ERd)																								
	SUB.L @(d:2,ERs), @(d:32,ERd)	L	5	7	8						S						@(d:32+ERd) - @(4/8/12+ERs) → @(d:32+ERd)																								
	SUB.L @(d:2,ERs), @(d:16,Rd.B)	L	4	7	7						S	D					@(d:16+RdL<<2) - @(4/8/12+ERs) → @(d:16+RdL<<2)																								

Arithmetic operations (109)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	Addressing Mode										Operation ⁵					Condition Codes ^{6,7}							
						Number of Execution Stages ¹																						
						16-Bit Instruction Fetch	Fetch	pxx	Rn	@ERn	@(d,ERn)	@(d,Rn,LRn,WERn,L)	@ERn/@ERn+/@ERn+/@ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C			
SUB	SUB.L @(d:32,ERs),@ERd-	L	5	6	8						S	D					@ERd - @(d:32+ERs) → @ERd			ERd32-4→ERd32	---	↑	↑	↑	↑	↑		
	SUB.L @(d:32,ERs),@+ERd	L	5	7	9						S	D					ERd32+4→ERd32 @ERd - @(d:32+ERs) → @ERd				---	↑	↑	↑	↑	↑		
	SUB.L @(d:32,ERs),@-ERd	L	5	7	9						S	D					ERd32-4→ERd32 @ERd - @(d:32+ERs) → @ERd				---	↑	↑	↑	↑	↑		
	SUB.L @(d:32,ERs),@(d:2,ERd)	L	5	7	9						S							@(4/8/12+ERd) - @(d:32+ERs) → @(4/8/12+ERd)				---	↑	↑	↑	↑	↑	
	SUB.L @(d:32,ERs),@(d:16,ERd)	L	6	7	9						S							@(d:16+ERd) - @(d:32+ERs) → @(d:16+ERd)				---	↑	↑	↑	↑	↑	
	SUB.L @(d:32,ERs),@(d:32,ERd)	L	7	7	1						S							@(d:32+ERd) - @(d:32+ERs) → @(d:32+ERd)				---	↑	↑	↑	↑	↑	
	SUB.L @(d:32,ERs),@(d:16,Rd,B)	L	6	7	9						S	D						@(d:16+RdL<<2) - @(d:32+ERs) → @(d:16+RdL<<2)				---	↑	↑	↑	↑	↑	
	SUB.L @(d:32,ERs),@(d:16,Rd,W)	L	6	7	9						S	D						@(d:16+RdL<<2) - @(d:32+ERs) → @(d:16+RdL<<2)				---	↑	↑	↑	↑	↑	
	SUB.L @(d:32,ERs),@(d:16,ERd,L)	L	6	7	9						S	D						@(d:16+ERdL<<2) - @(d:32+ERs) → @(d:16+ERdL<<2)				---	↑	↑	↑	↑	↑	
	SUB.L @(d:32,ERs),@(d:32,Rd,B)	L	7	7	1						S	D						@(d:32+RdL<<2) - @(d:32+ERs) → @(d:32+RdL<<2)				---	↑	↑	↑	↑	↑	
	SUB.L @(d:32,ERs),@(d:32,Rd,W)	L	7	7	1						S	D						@(d:32+RdL<<2) - @(d:32+ERs) → @(d:32+RdL<<2)				---	↑	↑	↑	↑	↑	
	SUB.L @(d:32,ERs),@(d:32,ERd,L)	L	7	7	1						S	D						@(d:32+ERdL<<2) - @(d:32+ERs) → @(d:32+ERdL<<2)				---	↑	↑	↑	↑	↑	
	SUB.L @(d:32,ERs),@aa:16	L	6	6	9						S		D					@aa:16 - @(d:32+ERs) → @aa:16				---	↑	↑	↑	↑	↑	
	SUB.L @(d:32,ERs),@aa:32	L	7	6	1						S		D					@aa:32 - @(d:32+ERs) → @aa:32				---	↑	↑	↑	↑	↑	
	SUB.L @(d:16,Rs,B),@ERd	L	4	6	7				D		S							@ERd - @(d:16+RsL<<2) → @ERd				---	↑	↑	↑	↑	↑	
	SUB.L @(d:16,Rs,B),@ERd+	L	4	6	7						S	D						@ERd - @(d:16+RsL<<2) → @ERd				ERd32+4→ERd32	---	↑	↑	↑	↑	↑
	SUB.L @(d:16,Rs,B),@ERd-	L	4	6	7						S	D						@ERd - @(d:16+RsL<<2) → @ERd				ERd32-4→ERd32	---	↑	↑	↑	↑	↑
	SUB.L @(d:16,Rs,B),@+ERd	L	4	7	8						S	D						ERd32+4→ERd32 @ERd - @(d:16+RsL<<2) → @ERd					---	↑	↑	↑	↑	↑
	SUB.L @(d:16,Rs,B),@-ERd	L	4	7	8						S	D						ERd32-4→ERd32 @ERd - @(d:16+RsL<<2) → @ERd					---	↑	↑	↑	↑	↑
	SUB.L @(d:16,Rs,B),@(d:2,ERd)	L	4	7	8					D	S							@(4/8/12+ERd) - @(d:16+RsL<<2) → @(4/8/12+ERd)				---	↑	↑	↑	↑	↑	
	SUB.L @(d:16,Rs,B),@(d:16,ERd)	L	5	7	8					D	S							@(d:16+ERd) - @(d:16+RsL<<2) → @(d:16+ERd)				---	↑	↑	↑	↑	↑	
	SUB.L @(d:16,Rs,B),@(d:32,ERd)	L	6	7	9					D	S							@(d:32+ERd) - @(d:16+RsL<<2) → @(d:32+ERd)				---	↑	↑	↑	↑	↑	
	SUB.L @(d:16,Rs,B),@(d:16,Rd,B)	L	5	7	8						S							@(d:16+RdL<<2) - @(d:16+RsL<<2) → @(d:16+RdL<<2)				---	↑	↑	↑	↑	↑	
	SUB.L @(d:16,Rs,B),@(d:16,Rd,W)	L	5	7	8						S							@(d:16+RdL<<2) - @(d:16+RsL<<2) → @(d:16+RdL<<2)				---	↑	↑	↑	↑	↑	
SUB.L @(d:16,Rs,B),@(d:16,ERd,L)	L	5	7	8						S							@(d:16+ERdL<<2) - @(d:16+RsL<<2) → @(d:16+ERdL<<2)				---	↑	↑	↑	↑	↑		

Arithmetic operations (110)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	Addressing Mode								Operation ⁵					Condition Codes ⁶																				
						16-Bit Instruction Fetch Execution Status ¹	Prx	Rn	@ERn	@ERn	@Rn.L/Rn.W/Rn.L	@ERn/@ERn+/@ERn+/@ERn	@aa:8/@aa:16/@aa:32																										
																									Number of	Execution Status ²	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C		
SUB	SUB.L @(d:16, Rs.B), @(d:32, Rd.B)	L	6	7	9					S																													
	SUB.L @(d:16, Rs.B), @(d:32, Rd.W)	L	6	7	9					S																													
	SUB.L @(d:16, Rs.B), @(d:32, ERd.L)	L	6	7	9					S																													
	SUB.L @(d:16, Rs.B), @aa:16	L	5	6	8					S	D																												
	SUB.L @(d:16, Rs.B), @aa:32	L	6	6	9					S	D																												
	SUB.L @(d:16, Rs.W), @ERd	L	4	6	7				D	S																													
	SUB.L @(d:16, Rs.W), @ERd+	L	4	6	7					S	D																												
	SUB.L @(d:16, Rs.W), @ERd-	L	4	6	7					S	D																												
	SUB.L @(d:16, Rs.W), @+ERd	L	4	7	8					S	D		ERd32+4 → ERd32																										
	SUB.L @(d:16, Rs.W), @-ERd	L	4	7	8					S	D		ERd32-4 → ERd32																										
	SUB.L @(d:16, Rs.W), @(d:2, ERd)	L	4	7	8					D	S																												
	SUB.L @(d:16, Rs.W), @(d:16, ERd)	L	5	7	8					D	S																												
	SUB.L @(d:16, Rs.W), @(d:32, ERd)	L	6	7	9					D	S																												
	SUB.L @(d:16, Rs.W), @(d:16, Rd.B)	L	5	7	8						S																												
	SUB.L @(d:16, Rs.W), @(d:16, Rd.W)	L	5	7	8						S																												
	SUB.L @(d:16, Rs.W), @(d:16, ERd.L)	L	5	7	8						S																												
	SUB.L @(d:16, Rs.W), @(d:32, Rd.B)	L	6	7	9						S																												
	SUB.L @(d:16, Rs.W), @(d:32, Rd.W)	L	6	7	9						S																												
	SUB.L @(d:16, Rs.W), @(d:32, ERd.L)	L	6	7	9						S																												
	SUB.L @(d:16, Rs.W), @aa:16	L	5	6	8						S	D																											
	SUB.L @(d:16, Rs.W), @aa:32	L	6	6	9						S	D																											
	SUB.L @(d:16, ERs.L), @ERd	L	4	6	7				D	S																													
	SUB.L @(d:16, ERs.L), @ERd+	L	4	6	7					S	D																												
	SUB.L @(d:16, ERs.L), @ERd-	L	4	6	7					S	D																												
	SUB.L @(d:16, ERs.L), @+ERd	L	4	7	8					S	D		ERd32+4 → ERd32																										

Arithmetic operations (112)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode								Operation ^s					Condition Codes ^{s2}															
				Min	16-Bit Instruction Fetch Execution Stages ¹	FXZ	Rn	@(d,ERn)	@(d,ERn)	@(d,Rn,LRn,W,ERn,L)	@(ERn)/@ERn+/@ERn//@ERn+/@+ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C									
SUB	SUB.L @(d:32,Rs,B),@(d:32,ERd,L)	L	7	7	1							S				@(d:32+ERd<<2) - @(d:32+RsL<<2) → @(d:32+ERd<<2)									—	↑	↓	↑	↓	↑		
	SUB.L @(d:32,Rs,B),@aa:16	L	6	6	9							S	D			@aa:16 - @(d:32+RsL<<2) → @aa:16									—	↑	↓	↑	↓	↑		
	SUB.L @(d:32,Rs,B),@aa:32	L	7	6	1							S	D			@aa:32 - @(d:32+RsL<<2) → @aa:32									—	↑	↓	↑	↓	↑		
	SUB.L @(d:32,Rs,W),@ERd	L	5	6	8								D	S			@ERd - @(d:32+Rs<<2) → @ERd									—	↑	↓	↑	↓	↑	
	SUB.L @(d:32,Rs,W),@ERd+	L	5	6	8								S	D			@ERd - @(d:32+Rs<<2) → @ERd					ERd32+4→ERd32					—	↑	↓	↑	↓	↑
	SUB.L @(d:32,Rs,W),@ERd-	L	5	6	8								S	D			@ERd - @(d:32+Rs<<2) → @ERd					ERd32-4→ERd32					—	↑	↓	↑	↓	↑
	SUB.L @(d:32,Rs,W),@+ERd	L	5	7	9								S	D		ERd32+4→ERd32	@ERd - @(d:32+Rs<<2) → @ERd									—	↑	↓	↑	↓	↑	
	SUB.L @(d:32,Rs,W),@-ERd	L	5	7	9								S	D		ERd32-4→ERd32	@ERd - @(d:32+Rs<<2) → @ERd									—	↑	↓	↑	↓	↑	
	SUB.L @(d:32,Rs,W),@(d:2,ERd)	L	5	7	9								D	S			@(4/8/12+ERd) - @(d:32+Rs<<2) → @(4/8/12+ERd)									—	↑	↓	↑	↓	↑	
	SUB.L @(d:32,Rs,W),@(d:16,ERd)	L	6	7	9								D	S			@(d:16+ERd) - @(d:32+Rs<<2) → @(d:16+ERd)									—	↑	↓	↑	↓	↑	
	SUB.L @(d:32,Rs,W),@(d:32,ERd)	L	7	7	1								D	S			@(d:32+ERd) - @(d:32+Rs<<2) → @(d:32+ERd)									—	↑	↓	↑	↓	↑	
	SUB.L @(d:32,Rs,W),@(d:16,Rd,B)	L	6	7	9								S				@(d:16+RdL<<2) - @(d:32+Rs<<2) → @(d:16+RdL<<2)									—	↑	↓	↑	↓	↑	
	SUB.L @(d:32,Rs,W),@(d:16,Rd,W)	L	6	7	9								S				@(d:16+Rd<<2) - @(d:32+Rs<<2) → @(d:16+Rd<<2)									—	↑	↓	↑	↓	↑	
	SUB.L @(d:32,Rs,W),@(d:16,ERdL)	L	6	7	9								S				@(d:16+ERd<<2) - @(d:32+Rs<<2) → @(d:16+ERd<<2)									—	↑	↓	↑	↓	↑	
	SUB.L @(d:32,Rs,W),@(d:32,Rd,B)	L	7	7	1								S				@(d:32+RdL<<2) - @(d:32+Rs<<2) → @(d:32+RdL<<2)									—	↑	↓	↑	↓	↑	
	SUB.L @(d:32,Rs,W),@(d:32,Rd,W)	L	7	7	1								S				@(d:32+Rd<<2) - @(d:32+Rs<<2) → @(d:32+Rd<<2)									—	↑	↓	↑	↓	↑	
	SUB.L @(d:32,Rs,W),@(d:32,ERdL)	L	7	7	1								S				@(d:32+ERd<<2) - @(d:32+Rs<<2) → @(d:32+ERd<<2)									—	↑	↓	↑	↓	↑	
	SUB.L @(d:32,Rs,W),@aa:16	L	6	6	9								S	D			@aa:16 - @(d:32+Rs<<2) → @aa:16									—	↑	↓	↑	↓	↑	
	SUB.L @(d:32,Rs,W),@aa:32	L	7	6	1								S	D			@aa:32 - @(d:32+Rs<<2) → @aa:32									—	↑	↓	↑	↓	↑	
	SUB.L @(d:32,ERs,L),@ERd	L	5	6	8									D	S			@ERd - @(d:32+ERs<<2) → @ERd									—	↑	↓	↑	↓	↑
	SUB.L @(d:32,ERs,L),@ERd+	L	5	6	8								S	D			@ERd - @(d:32+ERs<<2) → @ERd					ERd32+4→ERd32					—	↑	↓	↑	↓	↑
	SUB.L @(d:32,ERs,L),@ERd-	L	5	6	8								S	D			@ERd - @(d:32+ERs<<2) → @ERd					ERd32-4→ERd32					—	↑	↓	↑	↓	↑
	SUB.L @(d:32,ERs,L),@+ERd	L	5	7	9								S	D		ERd32+4→ERd32	@ERd - @(d:32+ERs<<2) → @ERd										—	↑	↓	↑	↓	↑
	SUB.L @(d:32,ERs,L),@-ERd	L	5	7	9								S	D		ERd32-4→ERd32	@ERd - @(d:32+ERs<<2) → @ERd										—	↑	↓	↑	↓	↑
	SUB.L @(d:32,ERs,L),@(d:2,ERd)	L	5	7	9								D	S			@(4/8/12+ERd) - @(d:32+ERs<<2) → @(4/8/12+ERd)										—	↑	↓	↑	↓	↑



Arithmetic operations (113)



Instruction	Mnemonic	Size	Instruction Length	Addressing Mode											Operation ⁶					Condition Codes ²																		
				Min.	16-Bit Instruction Fetch Execution Status ¹	Number of Fetch Execution Status ¹	16-Bit Instruction Fetch Execution Status ¹	16-Bit Instruction Fetch Execution Status ¹	16-Bit Instruction Fetch Execution Status ¹	16-Bit Instruction Fetch Execution Status ¹	16-Bit Instruction Fetch Execution Status ¹	16-Bit Instruction Fetch Execution Status ¹	16-Bit Instruction Fetch Execution Status ¹	16-Bit Instruction Fetch Execution Status ¹											16-Bit Instruction Fetch Execution Status ¹	16-Bit Instruction Fetch Execution Status ¹												
SUB	SUB.L @(d:32,ERs.L),@(d:16,ERd)	L	6 7 9										D	S																								
	SUB.L @(d:32,ERs.L),@(d:32,ERd)	L	7 7 1										D	S																								
	SUB.L @(d:32,ERs.L),@(d:16,Rd,B)	L	6 7 9											S																								
	SUB.L @(d:32,ERs.L),@(d:16,Rd,W)	L	6 7 9											S																								
	SUB.L @(d:32,ERs.L),@(d:16,ERd,L)	L	6 7 9											S																								
	SUB.L @(d:32,ERs.L),@(d:32,Rd,B)	L	7 7 1											S																								
	SUB.L @(d:32,ERs.L),@(d:32,Rd,W)	L	7 7 1											S																								
	SUB.L @(d:32,ERs.L),@(d:32,ERd,L)	L	7 7 1											S																								
	SUB.L @(d:32,ERs.L),@aa:16	L	6 6 9										S	D																								
	SUB.L @(d:32,ERs.L),@aa:32	L	7 6 1										S	D																								
	SUB.L @aa:16,@ERd	L	4 5 7										D	S																								
	SUB.L @aa:16,@ERd+	L	4 5 7											D S																								
	SUB.L @aa:16,@ERd-	L	4 5 7											D S																								
	SUB.L @aa:16,@+ERd	L	4 6 8											D S		ERd32+4 → ERd32	@ ERd	-	@aa:16	→	@ ERd																	
	SUB.L @aa:16,@-ERd	L	4 6 8											D S		ERd32-4 → ERd32	@ ERd	-	@aa:16	→	@ ERd																	
	SUB.L @aa:16,@(d:2,ERd)	L	4 6 8										D	S			@(4/8/12+ERd)	-	@aa:16	→	@(4/8/12+ERd)																	
	SUB.L @aa:16,@(d:16,ERd)	L	5 6 8										D	S			@(d:16+ERd)	-	@aa:16	→	@(d:16+ERd)																	
	SUB.L @aa:16,@(d:32,ERd)	L	6 6 9										D	S			@(d:32+ERd)	-	@aa:16	→	@(d:32+ERd)																	
	SUB.L @aa:16,@(d:16,Rd,B)	L	5 6 8										D	S			@(d:16+RdL<<2)	-	@aa:16	→	@(d:16+RdL<<2)																	
	SUB.L @aa:16,@(d:16,Rd,W)	L	5 6 8										D	S			@(d:16+RdL<<2)	-	@aa:16	→	@(d:16+RdL<<2)																	
	SUB.L @aa:16,@(d:16,ERd,L)	L	5 6 8										D	S			@(d:16+ERdL<<2)	-	@aa:16	→	@(d:16+ERdL<<2)																	
	SUB.L @aa:16,@(d:32,Rd,B)	L	6 6 9										D	S			@(d:32+RdL<<2)	-	@aa:16	→	@(d:32+RdL<<2)																	
	SUB.L @aa:16,@(d:32,Rd,W)	L	6 6 9										D	S			@(d:32+RdL<<2)	-	@aa:16	→	@(d:32+RdL<<2)																	
	SUB.L @aa:16,@(d:32,ERd,L)	L	6 6 9										D	S			@(d:32+ERdL<<2)	-	@aa:16	→	@(d:32+ERdL<<2)																	

Arithmetic operations (114)

Instruction	Mnemonic	Size	Instruction Length	Min.	Addressing Mode										Operation ⁵					Condition Code ^{s2}											
					Number of Execution Stages ¹																										
					16-Bit Instruction Fetch	xx	Rn	@ERN	@(d,ERN)	@(d,Rn,W,ERN,L)	@ERN/@ERN+/@ERN+/@ERN	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C								
SUB	SUB.L @aa:16,@aa:16	L	5	5	8									S			@aa:16 - @aa:16 → @aa:16							—	↓	↑	↓	↑	↓		
	SUB.L @aa:16,@aa:32	L	6	5	9									S			@aa:32 - @aa:16 → @aa:32								—	↓	↑	↓	↑	↓	
	SUB.L @aa:32,@ERd	L	5	5	8			D						S			@ERd - @aa:32 → @ERd								—	↓	↑	↓	↑	↓	
	SUB.L @aa:32,@ERd+	L	5	5	8							D		S			@ERd - @aa:32 → @ERd					ERd32+4→ERd32				—	↓	↑	↓	↑	↓
	SUB.L @aa:32,@ERd-	L	5	5	8							D		S			@ERd - @aa:32 → @ERd					ERd32-4→ERd32				—	↓	↑	↓	↑	↓
	SUB.L @aa:32,@+ERd	L	5	6	9								D	S		ERd32+4→ERd32	@ERd - @aa:32 → @ERd								—	↓	↑	↓	↑	↓	
	SUB.L @aa:32,@-ERd	L	5	6	9								D	S		ERd32-4→ERd32	@ERd - @aa:32 → @ERd								—	↓	↑	↓	↑	↓	
	SUB.L @aa:32,@(d,2,ERd)	L	5	6	9							D		S			@(4/8/12+ERd) - @aa:32 → @(4/8/12+ERd)								—	↓	↑	↓	↑	↓	
	SUB.L @aa:32,@(d,16,ERd)	L	6	6	9							D		S			@(d:16+ERd) - @aa:32 → @(d:16+ERd)								—	↓	↑	↓	↑	↓	
	SUB.L @aa:32,@(d:32,ERd)	L	7	6	1							D		S			@(d:32+ERd) - @aa:32 → @(d:32+ERd)								—	↓	↑	↓	↑	↓	
	SUB.L @aa:32,@(d:16,Rd,B)	L	6	6	9							D		S			@(d:16+Rd<<2) - @aa:32 → @(d:16+Rd<<2)								—	↓	↑	↓	↑	↓	
	SUB.L @aa:32,@(d:16,Rd,W)	L	6	6	9							D		S			@(d:16+Rd<<2) - @aa:32 → @(d:16+Rd<<2)								—	↓	↑	↓	↑	↓	
	SUB.L @aa:32,@(d:16,ERd,L)	L	6	6	9							D		S			@(d:16+ERd<<2) - @aa:32 → @(d:16+ERd<<2)								—	↓	↑	↓	↑	↓	
	SUB.L @aa:32,@(d:32,Rd,B)	L	7	6	1							D		S			@(d:32+Rd<<2) - @aa:32 → @(d:32+Rd<<2)								—	↓	↑	↓	↑	↓	
	SUB.L @aa:32,@(d:32,Rd,W)	L	7	6	1							D		S			@(d:32+Rd<<2) - @aa:32 → @(d:32+Rd<<2)								—	↓	↑	↓	↑	↓	
	SUB.L @aa:32,@(d:32,ERd,L)	L	7	6	1							D		S			@(d:32+ERd<<2) - @aa:32 → @(d:32+ERd<<2)								—	↓	↑	↓	↑	↓	
SUB.L @aa:32,@aa:16	L	6	5	9										S		@aa:16 - @aa:32 → @aa:16									—	↓	↑	↓	↑	↓	
SUB.L @aa:32,@aa:32	L	7	5	1										S		@aa:32 - @aa:32 → @aa:32									—	↓	↑	↓	↑	↓	
ADDX	ADDX.B #xx:8,Rd	B	1	1	1	S	D									Rd8 + #xx + C → Rd8									—	↓	↑	↓	↑	↓	
	ADDX.B #xx:8,@ERd	B	2	4	4	S	D									@ERd + #xx + C → @ERd										—	↓	↑	↓	↑	↓
	ADDX.B #xx:8,@ERd-	B	3	5	5	S						D				@ERd + #xx + C → @ERd					ERd32-1→ERd32					—	↓	↑	↓	↑	↓
	ADDX.B Rs,Rd	B	1	1	1	S										Rd8 + Rs8 + C → Rd8										—	↓	↑	↓	↑	↓
	ADDX.B Rs,@ERd	B	2	4	4	S	D									@ERd + Rs8 + C → @ERd										—	↓	↑	↓	↑	↓
	ADDX.B Rs,@ERd-	B	3	5	5	S						D				@ERd + Rs8 + C → @ERd					ERd32-1→ERd32					—	↓	↑	↓	↑	↓
ADDX.B @ERs,Rd	B	2	3	3	D	S									Rd8 + @ERd + C → Rd8										—	↓	↑	↓	↑	↓	



Arithmetic operations (115)

Instruction	Mnemonic	Size	Instruction Length	Min.	Addressing Mode							Operation ⁵					Condition Codes ^{6,7}																					
					Number of Execution Stages ¹	16-Bit Instruction Fetch #xx	Rn	@ERn	@(d,ERn)	@(d,Rn.L,Rn.W,ERn.L)	@.ERn/@.ERn+/@.ERn+/@.ERn	@aa:R/@.aa:16/@.aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C															
ADDD	ADDD.B @ERs, Rd	B	3	4	4		D																															
	ADDD.B @ERs, @ERd	B	3	5	6			S																														
	ADDD.B @ERs-, @ERd-	B	3	5	6																																	
	ADDD.W #xx:16, Rd	W	3	2	3	S	D																															
	ADDD.W #xx:16, @ERd	W	3	4	5	S		D																														
	ADDD.W #xx:16, @ERd-	W	4	5	6	S																																
	ADDD.W Rn, Rd	W	2	2	2		S																															
	ADDD.W Rn, @ERd	W	2	4	4		S	D																														
	ADDD.W Rn, @ERd-	W	3	5	5	S																																
	ADDD.W @ERs, Rd	W	2	3	3		D	S																														
	ADDD.W @ERs-, Rd	W	3	4	4		D																															
	ADDD.W @ERs, @ERd	W	3	5	6			S																														
	ADDD.W @ERs-, @ERd-	W	3	5	6																																	
	ADDD.L #xx:32, ERd	L	4	2	4	S	D																															
	ADDD.L #xx:32, @ERd	L	5	5	7	S		D																														
	ADDD.L #xx:32, @ERd-	L	5	5	7	S																																
	ADDD.L ERs, ERd	L	2	2	2		S																															
	ADDD.L ERs, @ERd	L	3	5	5		S	D																														
	ADDD.L ERs, @ERd-	L	3	5	5		S																															
	ADDD.L @ERs, ERd	L	3	4	4		D	S																														
	ADDD.L @ERs-, ERd	L	3	4	4			D																														
	ADDD.L @ERs, @ERd	L	3	5	6			S																														

Arithmetic operations (117)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹	Addressing Mode												Operation ⁵					Condition Codes ²																			
						Number of Execution Stages ¹	Rn	@ERn	@ERn	@ERn	@ERn	@ERn	@ERn	@ERn	@ERn	@ERn	@ERn												Operation 1	Operation 2	Operation 3	Operation 4	Operation 5									
																		16-Bit Instruction Fetch Execution Stages ¹	Rn	@ERn	@ERn	@ERn	@ERn	@ERn	@ERn	@ERn	@ERn	@ERn						@ERn								
SUBX	SUBX.L ERs, @ERd	L	3	5	5	S	D									@ERd	-	ERs-C	→	@ERd																						
	SUBX.L ERs, @ERd-	L	3	5	5	S	D									@ERd	-	ERs-C	→	@ERd			ERd32-4→ERd32																			
	SUBX.L @ERs, ERd	L	3	4	4	D	S									ERd32	-	@ERd-C	→	ERd32																						
	SUBX.L @ERs-, ERd	L	3	4	4	D	S									ERd32	-	@ERd-C	→	ERd32	ERs32-4→ERs32																					
	SUBX.L @ERs, @ERd	L	3	5	6		S									@ERd	-	@ERs-C	→	@ERd																						
	SUBX.L @ERs-, @ERd-	L	3	5	6		S									@ERd	-	@ERs-C	→	@ERd	ERs32-4→ERs32	ERd32-4→ERd32																				
INC	INC.B Rd	B	1	1	1	D										Rd8	+	1	→	Rd8																						
	INC.W #1, Rd	W	1	1	1	D										Rd16	+	1	→	Rd16																						
	INC.W #2, Rd	W	1	1	1	D										Rd16	+	2	→	Rd16																						
	INC.L #1, ERd	L	1	1	1	D										ERd	+	1	→	ERd																						
	INC.L #2, ERd	L	1	1	1	D										ERd	+	2	→	ERd																						
DEC	DEC.B Rd	B	1	1	1	D										Rd8	-	1	→	Rd8																						
	DEC.W #1, Rd	W	1	1	1	D										Rd16	-	1	→	Rd16																						
	DEC.W #2, Rd	W	1	1	1	D										Rd16	-	2	→	Rd16																						
	DEC.L #1, ERd	L	1	1	1	D										ERd32	-	1	→	ERd32																						
	DEC.L #2, ERd	L	1	1	1	D										ERd32	-	2	→	ERd32																						
	ADDS	ADDS #1, ERd	L	1	1	1	D										ERd32	+	1	→	ERd32																					
	ADDS #2, ERd	L	1	1	1	D										ERd32	+	2	→	ERd32																						
	ADDS #4, ERd	L	1	1	1	D										ERd32	+	4	→	ERd32																						
SUBS	SUBS #1, ERd	L	1	1	1	D										ERd32	-	1	→	ERd32																						
	SUBS #2, ERd	L	1	1	1	D										ERd32	-	2	→	ERd32																						
	SUBS #4, ERd	L	1	1	1	D										ERd32	-	4	→	ERd32																						
DAA	DAA Rd	B	1	1	1	D																																				
DAS	DAS Rd	B	1	1	1	D																																				



Arithmetic operations (118)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	Number of Execution Stages ¹	Addressing Mode							Operation ⁵					Condition Codes ²																							
							16-Bit Instruction Fetch	Fetch	Execute	Write Back	Read	Write Back	Execute	Write Back	Execute	Write Back	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C															
MULXU	MULXU.B #xx:4,Rd	B	2	2	2	2	S	D									Rd8	×	#xx(Unsigned multiplication)	→	Rd16																					
	MULXU.B Rs,Rd	B	1	1	1	1	S	D									Rd8	×	Rs8(Unsigned multiplication)	→	Rd16																					
	MULXU.W #xx:4,ERd	W	2	2	2	2	S	D									ERd	×	#xx(Unsigned multiplication)	→	ERd																					
	MULXU.W Rs,ERd	W	1	1	1	1	S	D										Rd16	×	Rs16(Unsigned multiplication)	→	ERd																				
DIVXU	DIVXU.B #xx:4,Rd	B	2	11	11	S	D										Rd16	÷	#xx(Unsigned division)	→	Rd16 (RdH:remainder, RdL:quotient)																					
	DIVXU.B Rs,Rd	B	1	10	10	S	D										Rd16	÷	Rs8(Unsigned division)	→	Rd16 (RdH:remainder, RdL:quotient)																					
	DIVXU.W #xx:4,ERd	W	2	19	19	S	D										ERd	÷	#xx(Unsigned division)	→	ERd(Ed:remainder, Rd:quotient)																					
	DIVXU.W Rs,ERd	W	1	18	18	S	D											ERd	÷	Rs16 (Unsigned division)	→	ERd(Ed:remainder, Rd:quotient)																				
MULU	MULU.W #xx:4,Rd	W	2	2	2	S	D										Rd16	×	#xx(Unsigned multiplication)	→	Rd16																					
	MULU.W Rs,Rd	W	2	2	2	S	D										Rd16	×	Rs16(Unsigned multiplication)	→	ERd																					
	MULU.L #xx:4,ERd	L	2	5	5	S	D										ERd	×	#xx(Unsigned multiplication)	→	ERd																					
	MULU.L ERs,ERd	L	2	5	5	S	D											ERd	×	ERs(Unsigned multiplication)	→	ERd																				
MULU/U	MULU/U.L #xx:4,ERd	L	2	6	6	S	D										ERd	×	#xx(Unsigned multiplication)	→	ERd(upper 32 bits)																					
	MULU/U.L ERs,ERd	L	2	6	6	S	D											ERd	×	ERs(Unsigned multiplication)	→	ERd(upper 32 bits)																				
DIVU	DIVU.W #xx:4,Rd	W	2	10	10	S	D										Rd16	÷	#xx(Unsigned division)	→	Rd16(quotient)																					
	DIVU.W Rs,Rd	W	2	10	10	S	D										Rd16	÷	Rs16(Unsigned division)	→	Rd16(quotient)																					
	DIVU.L #xx:4,ERd	L	2	18	18	S	D											ERd	÷	#xx(Unsigned division)	→	ERd(quotient)																				



Arithmetic operations (119)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	Number of Execution Stages ¹	Addressing Mode								Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	Condition Codes ²							
							16-Bit Instruction Fetch	Fetch	Rn	@ERN	@(d,ERn)	@(d,Rn,L,Rn,W,ERn,L)	@.ERn/@.ERn+/@.ERn+/@.ERn	@.aa:8/@.aa:16/@.aa:32						I	H	N	Z	V	C		
DIVU	DIVU.L ERs,ERd	L	2	18	18			S	D						ERd	÷ ERs(Unsigned division)	→ ERd(quotient)					—	—	↓	↓	—	—
MULXS	MULXS.B #xx:4,Rd	B	2	2	2	S	D								Rd8	× #xx(Signed multiplication)	→ Rd16					—	—	↓	↓	—	—
	MULXS.B Rs,Rd	B	2	2	2	S	D								Rd8	× Rs8(Signed multiplication)	→ Rd16					—	—	↓	↓	—	—
	MULXS.W #xx:4,ERd	W	2	2	2	S	D								ERd	× #xx(Signed multiplication)	→ ERd					—	—	↓	↓	—	—
	MULXS.W Rs,ERd	W	2	2	2	S	D								Rd16	× Rs16(Signed multiplication)	→ ERd					—	—	↓	↓	—	—
DIVXS	DIVXS.B #xx:4,Rd	B	2	12	12	S	D								Rd16	÷ #xx (Signed division)	→ Rd16 (RdH:remainder, RdL:quotient)					—	—	↓	↓	—	—
	DIVXS.B Rs,Rd	B	2	12	12	S	D								Rd16	÷ Rs8 (Signed division)	→ Rd16 (RdH:remainder, RdL:quotient)					—	—	↓	↓	—	—
	DIVXS.W #xx:4,ERd	W	2	20	20	S	D								ERd	÷ #xx (Signed division)	→ ERd(Ed:remainder, Rd:quotient)					—	—	↓	↓	—	—
	DIVXS.W Rs,ERd	W	2	20	20	S	D								ERd	÷ Rs16 (Signed division)	→ ERd(Ed:remainder, Rd:quotient)					—	—	↓	↓	—	—
MULS	MULS.W #xx:4,Rd	W	2	2	2	S	D								Rd16	× #xx(Signed multiplication)	→ Rd16					—	—	↓	↓	—	—
	MULS.W Rs,Rd	W	2	2	2	S	D								Rd16	× Rs16(Signed multiplication)	→ ERd					—	—	↓	↓	—	—
	MULS.L #xx:4,ERd	L	2	5	5	S	D								ERd	× #xx(Signed multiplication)	→ ERd					—	—	↓	↓	—	—
	MULS.L ERs,ERd	L	2	5	5	S	D								ERd	× ERs(Signed multiplication)	→ ERd					—	—	↓	↓	—	—
MULS/U	MULS/U.L #xx:4,ERd	L	2	6	6	S	D								ERd	× #xx(Signed multiplication)	→ ERd(upper 32 bits)					—	—	↓	↓	—	—
	MULS/U.L ERs,ERd	L	2	6	6	S	D								ERd	× ERs(Signed multiplication)	→ ERd(upper 32 bits)					—	—	↓	↓	—	—

Arithmetic operations (120)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode							Operation ⁵					Condition Codes ^{6,7}								
				Min.	Max.	16-Bit Instruction Fetch Execution Status ¹	Prx	Rn	@ERn	@(d,ERn)	@(d,Rn),ERn,WERn,L	@ERn/@ERn+/@ERn-/@ERn-/@+ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C
DIVS	DIVS.W #xx:4,Rd	W	2	11	11	S	D							Rd16 ÷ #xx(Signed division) → Rd16(quotient)				↑	↓	↑	↓			
	DIVS.W Rs,Rd	W	2	11	11	S	D							Rd16 ÷ Rs16 (Signed division) → Rd16(quotient)				↑	↓	↑	↓			
DIVS	DIVS.L #xx:4,ERd	L	2	19	19	S	D							ERd ÷ #xx(Signed division) → ERd(quotient)				↑	↓	↑	↓			
	DIVS.L ERs,ERd	L	2	19	19	S	D							ERd ÷ ERs (Signed division) → ERd(quotient)				↑	↓	↑	↓			
NEG	NEG.B Rd	B	1	1	1		D							0 - Rd8 → Rd8				↑	↓	↑	↓	↑	↓	
	NEG.B @ERd	B	2	4	4			D						0 - @ERd → @ERd				↑	↓	↑	↓	↑	↓	
	NEG.B @ERd+	B	3	5	5				D					0 - @ERd → @ERd				↑	↓	↑	↓	↑	↓	
	NEG.B @ERd-	B	3	5	5				D					0 - @ERd → @ERd				↑	↓	↑	↓	↑	↓	
	NEG.B @+ERd	B	3	5	5				D				ERd32+1→ERd32	0 - @ERd → @ERd				↑	↓	↑	↓	↑	↓	
	NEG.B @-ERd	B	3	5	5				D				ERd32-1→ERd32	0 - @ERd → @ERd				↑	↓	↑	↓	↑	↓	
	NEG.B @(d:2,ERd)	B	3	5	5				D					0 - @(1/2/3+ERd) → @(1/2/3+ERd)				↑	↓	↑	↓	↑	↓	
	NEG.B @(d:16,ERd)	B	4	5	6				D					0 - @(d:16+ERd) → @(d:16+ERd)				↑	↓	↑	↓	↑	↓	
	NEG.B @(d:32,ERd)	B	5	5	7				D					0 - @(d:32+ERd) → @(d:32+ERd)				↑	↓	↑	↓	↑	↓	
	NEG.B @(d:16,Rd.B)	B	4	5	6				D					0 - @(d:16+RdL) → @(d:16+RdL)				↑	↓	↑	↓	↑	↓	
	NEG.B @(d:16,Rd.W)	B	4	5	6				D					0 - @(d:16+Rd) → @(d:16+Rd)				↑	↓	↑	↓	↑	↓	
	NEG.B @(d:16,ERd.L)	B	4	5	6				D					0 - @(d:16+ERd) → @(d:16+ERd)				↑	↓	↑	↓	↑	↓	
	NEG.B @(d:32,Rd.B)	B	5	5	7				D					0 - @(d:32+RdL) → @(d:32+RdL)				↑	↓	↑	↓	↑	↓	
	NEG.B @(d:32,Rd.W)	B	5	5	7				D					0 - @(d:32+Rd) → @(d:32+Rd)				↑	↓	↑	↓	↑	↓	
	NEG.B @(d:32,ERd.L)	B	5	5	7				D					0 - @(d:32+ERd) → @(d:32+ERd)				↑	↓	↑	↓	↑	↓	
	NEG.B @aa:8	B	2	4	4					D					0 - @aa:8 → @aa:8				↑	↓	↑	↓	↑	↓
	NEG.B @aa:16	B	3	4	5					D					0 - @aa:16 → @aa:16				↑	↓	↑	↓	↑	↓
NEG.B @aa:32	B	4	4	6					D					0 - @aa:32 → @aa:32				↑	↓	↑	↓	↑	↓	



Arithmetic operations (121)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	Number of Execution Stages ¹	Addressing Mode											Operation ⁵					Condition Code ²								
							16-Bit Instruction Fetch	Branch	Rn	@ERn	@(d,ERn)	@(d,Rn),W(Rn),L	@ERn/@ERn+/@ERn-/@+ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C						
NEG	NEG.W Rd	W	1	1	1	1	D											0	-	Rd16	→	Rd16				—	↓	↑	↑	↓	↓
	NEG.W @ERd	W	2	4	4		D											0	-	@ERd	→	@ERd				—	↓	↑	↑	↓	↓
	NEG.W @ERd+	W	3	5	5													0	-	@ERd	→	@ERd	ERd32+2→ERd32			—	↓	↑	↑	↓	↓
	NEG.W @ERd-	W	3	5	5													0	-	@ERd	→	@ERd	ERd32-2→ERd32			—	↓	↑	↑	↓	↓
	NEG.W @+ERd	W	3	5	5								D				ERd32+2→ERd32	0	-	@ERd	→	@ERd				—	↓	↑	↑	↓	↓
	NEG.W @-ERd	W	3	5	5								D			ERd32-2→ERd32	0	-	@ERd	→	@ERd				—	↓	↑	↑	↓	↓	
	NEG.W @(d:2,ERd)	W	3	5	5								D				0	-	@(2/4/6+ERd)	→	@(2/4/6+ERd)				—	↓	↑	↑	↓	↓	
	NEG.W @(d:16,ERd)	W	4	5	6								D				0	-	@(d:16+ERd)	→	@(d:16+ERd)				—	↓	↑	↑	↓	↓	
	NEG.W @(d:32,ERd)	W	5	5	7								D				0	-	@(d:32+ERd)	→	@(d:32+ERd)				—	↓	↑	↑	↓	↓	
	NEG.W @(d:16,Rd.B)	W	4	5	6								D				0	-	@(d:16+RdL)	→	@(d:16+RdL)				—	↓	↑	↑	↓	↓	
	NEG.W @(d:16,Rd.W)	W	4	5	6								D				0	-	@(d:16+Rd)	→	@(d:16+Rd)				—	↓	↑	↑	↓	↓	
	NEG.W @(d:16,ERd.L)	W	4	5	6								D				0	-	@(d:16+ERd)	→	@(d:16+ERd)				—	↓	↑	↑	↓	↓	
	NEG.W @(d:32,Rd.B)	W	5	5	7								D				0	-	@(d:32+RdL)	→	@(d:32+RdL)				—	↓	↑	↑	↓	↓	
	NEG.W @(d:32,Rd.W)	W	5	5	7								D				0	-	@(d:32+Rd)	→	@(d:32+Rd)				—	↓	↑	↑	↓	↓	
	NEG.W @(d:32,ERd.L)	W	5	5	7								D				0	-	@(d:32+ERd)	→	@(d:32+ERd)				—	↓	↑	↑	↓	↓	
	NEG.W @aa:16	W	3	4	5									D			0	-	@aa:16	→	@aa:16				—	↓	↑	↑	↓	↓	
	NEG.W @aa:32	W	4	4	6									D			0	-	@aa:32	→	@aa:32				—	↓	↑	↑	↓	↓	
	NEG.L ERd	L	1	1	1	1	D										0	-	ERd	→	ERd				—	↓	↑	↑	↓	↓	
	NEG.L @ERd	L	3	5	5								D				0	-	@ERd	→	@ERd				—	↓	↑	↑	↓	↓	
	NEG.L @ERd+	L	3	5	5								D				0	-	@ERd	→	@ERd	ERd32+4→ERd32			—	↓	↑	↑	↓	↓	
	NEG.L @ERd-	L	3	5	5								D				0	-	@ERd	→	@ERd	ERd32-4→ERd32			—	↓	↑	↑	↓	↓	
	NEG.L @+ERd	L	3	5	5								D			ERd32+4→ERd32	0	-	@ERd	→	@ERd				—	↓	↑	↑	↓	↓	
	NEG.L @-ERd	L	3	5	5								D			ERd32-4→ERd32	0	-	@ERd	→	@ERd				—	↓	↑	↑	↓	↓	
	NEG.L @(d:2,ERd)	L	3	5	5								D				0	-	@(4/8/12+ERd)	→	@(4/8/12+ERd)				—	↓	↑	↑	↓	↓	
	NEG.L @(d:16,ERd)	L	4	5	6								D				0	-	@(d:16+ERd)	→	@(d:16+ERd)				—	↓	↑	↑	↓	↓	

Arithmetic operations (122)

Instruction	Mnemonic	Size	Instruction Length	Number of		Addressing Mode									Operation ⁵					Condition Codes ^{6,2}																								
				Min.	Max.	16-Bit Instruction Fetch Execution Status ¹	PRx	Rn	@(d,ERn)	@(d,Rn,WRn,L)	@(Rn)/@ERn+/@ERn-/@+ERn	@aa:8/ @aa:16/ @aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C																					
NEG	NEG.L @(d:32,ERd)	L	5	5	7						D						0	-	@(d:32+ERd)	→	@(d:32+ERd)																							
	NEG.L @(d:16,Rd.B)	L	4	5	6												0	-	@(d:16+RdL)	→	@(d:16+RdL)																							
	NEG.L @(d:16,Rd.W)	L	4	5	6												0	-	@(d:16+Rd)	→	@(d:16+Rd)																							
	NEG.L @(d:16,ERd.L)	L	4	5	6												0	-	@(d:16+ERd)	→	@(d:16+ERd)																							
	NEG.L @(d:32,Rd.B)	L	5	5	7												0	-	@(d:32+RdL)	→	@(d:32+RdL)																							
	NEG.L @(d:32,Rd.W)	L	5	5	7												0	-	@(d:32+Rd)	→	@(d:32+Rd)																							
	NEG.L @(d:32,ERd.L)	L	5	5	7												0	-	@(d:32+ERd)	→	@(d:32+ERd)																							
	NEG.L @aa:16	L	4	5	6											D		0	-	@aa:16	→	@aa:16																						
NEG.L @aa:32	L	5	5	7											D		0	-	@aa:32	→	@aa:32																							
EXTU	EXTU.W Rd	W	1	1	1						D						0																											
	EXTU.W @ERd	W	2	4	4							D					0																											
	EXTU.W @ERd+	W	3	5	5											D		0								ERd32+2 → ERd32																		
	EXTU.W @ERd-	W	3	5	5											D		0								ERd32-2 → ERd32																		
	EXTU.W @+ERd	W	3	5	5											D		0								ERd32+2 → ERd32																		
	EXTU.W @-ERd	W	3	5	5											D		0								ERd32-2 → ERd32																		

Arithmetic operations (131)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode										Operation ⁵					Condition Codes ²					
				Number of Min.	16-Bit Instruction Fetch	Fetch	Execution Status ¹	Rn	@ERn	@(d,ERn)	@(d,Rn,WRn,L)	@ERn/@ERn+/@ERn+/@+ERn	@aa:8/@aa:16/@aa:32											
n				Min.	16-Bit Instruction Fetch	Fetch	Execution Status ¹	Rn	@ERn	@(d,ERn)	@(d,Rn,WRn,L)	@ERn/@ERn+/@ERn+/@+ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C
EXTS	EXTS.L @aa:16	L	4	5	6								D			(<Bit 15> of @aa:16) → (<Bits 31 to 16> of @aa:16)			—	—	↑	↓	0	—
	EXTS.L @aa:32	L	5	5	7								D			(<Bit 15> of @aa:32) → (<Bits 31 to 16> of @aa:32)			—	—	↑	↓	0	—
	EXTS.L #2,ERd	L	1	1	1			D								(<Bit 7> of ERd) → (<Bits 31 to 8> of ERd)			—	—	↑	↓	0	—
	EXTS.L #2,@ERd	L	3	5	5			D								(<Bit 7> of @ERd) → (<Bits 31 to 8> of @ERd)			—	—	↑	↓	0	—
	EXTS.L #2,@ERd+	L	3	5	5							D				(<Bit 7> of @ERd) → (<Bits 31 to 8> of @ERd)	ERd32+4→ERd32		—	—	↑	↓	0	—
	EXTS.L #2,@ERd-	L	3	5	5							D				(<Bit 7> of @ERd) → (<Bits 31 to 8> of @ERd)	ERd32-4→ERd32		—	—	↑	↓	0	—
	EXTS.L #2,@+ERd	L	3	5	5							D		ERd32+4→ERd32		(<Bit 7> of @ERd) → (<Bits 31 to 8> of @ERd)			—	—	↑	↓	0	—
	EXTS.L #2,@-ERd	L	3	5	5							D		ERd32-4→ERd32		(<Bit 7> of @ERd) → (<Bits 31 to 8> of @ERd)			—	—	↑	↓	0	—
	EXTS.L #2,@(d:2,ERd)	L	3	5	5				D							(<Bit 7> of @(d:2,ERd)) → (<Bits 31 to 8> of @(d:2,ERd))			—	—	↑	↓	0	—
	EXTS.L #2,@(d:16,ERd)	L	4	5	6				D							(<Bit 7> of @(d:16,ERd)) → (<Bits 31 to 8> of @(d:16,ERd))			—	—	↑	↓	0	—
	EXTS.L #2,@(d:32,ERd)	L	5	5	7				D							(<Bit 7> of @(d:32,ERd)) → (<Bits 31 to 8> of @(d:32,ERd))			—	—	↑	↓	0	—

Logic operations (7)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹	Addressing Mode							Operation ⁵					Condition Codes ²											
						Number of Execution Stages ¹	Fetch	Rn	@ERn	@(d,ERn)	@(d,Rn),W,ERn,L	@ERn/@ERn+/@ERn+/@ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C					
AND	AND.B @(d:16,ERs),@ERd	B	4	6	7			D	S							@ERd	^	@(d:16+ERs)	→	@ERd				—	—	↑	↓	0	—
	AND.B @(d:16,ERs),@ERd+	B	4	6	7				S	D						@ERd	^	@(d:16+ERs)	→	@ERd		ERd32+1→ERd32		—	—	↑	↓	0	—
	AND.B @(d:16,ERs),@ERd-	B	4	6	7				S	D						@ERd	^	@(d:16+ERs)	→	@ERd		ERd32-1→ERd32		—	—	↑	↓	0	—
	AND.B @(d:16,ERs),@+ERd	B	4	7	8				S	D				ERd32+1→ERd32		@ERd	^	@(d:16+ERs)	→	@ERd				—	—	↑	↓	0	—
	AND.B @(d:16,ERs),@-ERd	B	4	7	8				S	D				ERd32-1→ERd32		@ERd	^	@(d:16+ERs)	→	@ERd				—	—	↑	↓	0	—
	AND.B @(d:16,ERs),@(d:2,ERd)	B	4	7	8				S							@(1/2/3+ERd)	^	@(d:16+ERs)	→	@(1/2/3+ERd)				—	—	↑	↓	0	—
	AND.B @(d:16,ERs),@(d:16,ERd)	B	5	7	8				S							@(d:16+ERd)	^	@(d:16+ERs)	→	@(d:16+ERd)				—	—	↑	↓	0	—
	AND.B @(d:16,ERs),@(d:32,ERd)	B	6	7	9				S							@(d:32+ERd)	^	@(d:16+ERs)	→	@(d:32+ERd)				—	—	↑	↓	0	—
	AND.B @(d:16,ERs),@(d:16,Rd.B)	B	5	7	8				S	D						@(d:16+RdL)	^	@(d:16+ERs)	→	@(d:16+RdL)				—	—	↑	↓	0	—
	AND.B @(d:16,ERs),@(d:16,Rd.W)	B	5	7	8				S	D						@(d:16+Rd)	^	@(d:16+ERs)	→	@(d:16+Rd)				—	—	↑	↓	0	—
	AND.B @(d:16,ERs),@(d:16,ERdL)	B	5	7	8				S	D						@(d:16+ERd)	^	@(d:16+ERs)	→	@(d:16+ERd)				—	—	↑	↓	0	—
	AND.B @(d:16,ERs),@(d:32,Rd.B)	B	6	7	9				S	D						@(d:32+RdL)	^	@(d:16+ERs)	→	@(d:32+RdL)				—	—	↑	↓	0	—
	AND.B @(d:16,ERs),@(d:32,Rd.W)	B	6	7	9				S	D						@(d:32+Rd)	^	@(d:16+ERs)	→	@(d:32+Rd)				—	—	↑	↓	0	—
	AND.B @(d:16,ERs),@(d:32,ERdL)	B	6	7	9				S	D						@(d:32+ERd)	^	@(d:16+ERs)	→	@(d:32+ERd)				—	—	↑	↓	0	—
	AND.B @(d:16,ERs),@aa:16	B	5	6	8				S		D					@aa:16	^	@(d:16+ERs)	→	@aa:16				—	—	↑	↓	0	—
	AND.B @(d:16,ERs),@aa:32	B	6	6	9				S		D					@aa:32	^	@(d:16+ERs)	→	@aa:32				—	—	↑	↓	0	—
	AND.B @(d:32,ERs),@ERd	B	5	6	8			D	S							@ERd	^	@(d:32+ERs)	→	@ERd				—	—	↑	↓	0	—
	AND.B @(d:32,ERs),@ERd+	B	5	6	8				S	D						@ERd	^	@(d:32+ERs)	→	@ERd		ERd32+1→ERd32		—	—	↑	↓	0	—
	AND.B @(d:32,ERs),@ERd-	B	5	6	8				S	D						@ERd	^	@(d:32+ERs)	→	@ERd		ERd32-1→ERd32		—	—	↑	↓	0	—
	AND.B @(d:32,ERs),@+ERd	B	5	7	9				S	D				ERd32+1→ERd32		@ERd	^	@(d:32+ERs)	→	@ERd				—	—	↑	↓	0	—
	AND.B @(d:32,ERs),@-ERd	B	5	7	9				S	D				ERd32-1→ERd32		@ERd	^	@(d:32+ERs)	→	@ERd				—	—	↑	↓	0	—
	AND.B @(d:32,ERs),@(d:2,ERd)	B	5	7	9				S							@(1/2/3+ERd)	^	@(d:32+ERs)	→	@(1/2/3+ERd)				—	—	↑	↓	0	—
	AND.B @(d:32,ERs),@(d:16,ERd)	B	6	7	9				S							@(d:16+ERd)	^	@(d:32+ERs)	→	@(d:16+ERd)				—	—	↑	↓	0	—
	AND.B @(d:32,ERs),@(d:32,ERd)	B	7	7	1				S							@(d:32+ERd)	^	@(d:32+ERs)	→	@(d:32+ERd)				—	—	↑	↓	0	—
	AND.B @(d:32,ERs),@(d:16,Rd.B)	B	6	7	9				S	D						@(d:16+RdL)	^	@(d:32+ERs)	→	@(d:16+RdL)				—	—	↑	↓	0	—

Logic operations (9)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	Number of 16-Bit Instruction Fetches ¹	Addressing Mode								Operation ⁵					Condition Codes ²											
							Rn	@ERn	@(d,ERn)	@(d,Rn,WRn,LRn)	@ERn/@ERn+/@ERn+/@ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C								
AND	AND.B @(d:16,Rs,W),@ERd	B	4	6	7					S	D					@ERd	^	@(d:16+Rs)	→	@ERd			ERd32-1→ERd32	—	—	↑	↓	0	—		
	AND.B @(d:16,Rs,W),@+ERd	B	4	7	8					S	D					ERd32+1→ERd32	@ERd	^	@(d:16+Rs)	→	@ERd				—	—	↑	↓	0	—	
	AND.B @(d:16,Rs,W),@-ERd	B	4	7	8					S	D					ERd32-1→ERd32	@ERd	^	@(d:16+Rs)	→	@ERd				—	—	↑	↓	0	—	
	AND.B @(d:16,Rs,W),@(d:2,ERd)	B	4	7	8					D	S						@(1/2/3+ERd)	^	@(d:16+Rs)	→	@(1/2/3+ERd)				—	—	↑	↓	0	—	
	AND.B @(d:16,Rs,W),@(d:16,ERd)	B	5	7	8					D	S						@(d:16+ERd)	^	@(d:16+Rs)	→	@(d:16+ERd)				—	—	↑	↓	0	—	
	AND.B @(d:16,Rs,W),@(d:32,ERd)	B	6	7	9					D	S						@(d:32+ERd)	^	@(d:16+Rs)	→	@(d:32+ERd)				—	—	↑	↓	0	—	
	AND.B @(d:16,Rs,W),@(d:16,Rd,B)	B	5	7	8						S						@(d:16+RdL)	^	@(d:16+Rs)	→	@(d:16+RdL)				—	—	↑	↓	0	—	
	AND.B @(d:16,Rs,W),@(d:16,Rd,W)	B	5	7	8						S						@(d:16+Rd)	^	@(d:16+Rs)	→	@(d:16+Rd)				—	—	↑	↓	0	—	
	AND.B @(d:16,Rs,W),@(d:16,ERdL)	B	5	7	8						S						@(d:16+ERd)	^	@(d:16+Rs)	→	@(d:16+ERd)				—	—	↑	↓	0	—	
	AND.B @(d:16,Rs,W),@(d:32,Rd,B)	B	6	7	9						S						@(d:32+RdL)	^	@(d:16+Rs)	→	@(d:32+RdL)				—	—	↑	↓	0	—	
	AND.B @(d:16,Rs,W),@(d:32,Rd,W)	B	6	7	9						S						@(d:32+Rd)	^	@(d:16+Rs)	→	@(d:32+Rd)				—	—	↑	↓	0	—	
	AND.B @(d:16,Rs,W),@(d:32,ERdL)	B	6	7	9						S						@(d:32+ERd)	^	@(d:16+Rs)	→	@(d:32+ERd)				—	—	↑	↓	0	—	
	AND.B @(d:16,Rs,W),@aa:16	B	5	6	8						S	D					@aa:16	^	@(d:16+Rs)	→	@aa:16				—	—	↑	↓	0	—	
	AND.B @(d:16,Rs,W),@aa:32	B	6	6	9						S	D					@aa:32	^	@(d:16+Rs)	→	@aa:32				—	—	↑	↓	0	—	
	AND.B @(d:16,ERs,L),@ERd	B	4	6	7				D		S						@ERd	^	@(d:16+ERs)	→	@ERd				—	—	↑	↓	0	—	
	AND.B @(d:16,ERs,L),@ERd+	B	4	6	7						S	D					@ERd	^	@(d:16+ERs)	→	@ERd			ERd32+1→ERd32	—	—	↑	↓	0	—	
	AND.B @(d:16,ERs,L),@ERd-	B	4	6	7						S	D					@ERd	^	@(d:16+ERs)	→	@ERd			ERd32-1→ERd32	—	—	↑	↓	0	—	
	AND.B @(d:16,ERs,L),@+ERd	B	4	7	8						S	D						@(d:16+ERs)	^	@(d:16+ERs)	→	@ERd				—	—	↑	↓	0	—
	AND.B @(d:16,ERs,L),@-ERd	B	4	7	8						S	D						@(d:16+ERs)	^	@(d:16+ERs)	→	@ERd				—	—	↑	↓	0	—
	AND.B @(d:16,ERs,L),@(d:2,ERd)	B	4	7	8					D	S							@(1/2/3+ERd)	^	@(d:16+ERs)	→	@(1/2/3+ERd)				—	—	↑	↓	0	—
	AND.B @(d:16,ERs,L),@(d:16,ERd)	B	5	7	8					D	S							@(d:16+ERd)	^	@(d:16+ERs)	→	@(d:16+ERd)				—	—	↑	↓	0	—
	AND.B @(d:16,ERs,L),@(d:32,ERd)	B	6	7	9					D	S							@(d:32+ERd)	^	@(d:16+ERs)	→	@(d:32+ERd)				—	—	↑	↓	0	—
	AND.B @(d:16,ERs,L),@(d:16,Rd,B)	B	5	7	8						S							@(d:16+RdL)	^	@(d:16+ERs)	→	@(d:16+RdL)				—	—	↑	↓	0	—
	AND.B @(d:16,ERs,L),@(d:16,Rd,W)	B	5	7	8						S							@(d:16+Rd)	^	@(d:16+ERs)	→	@(d:16+Rd)				—	—	↑	↓	0	—
	AND.B @(d:16,ERs,L),@(d:16,ERdL)	B	5	7	8						S							@(d:16+ERd)	^	@(d:16+ERs)	→	@(d:16+ERd)				—	—	↑	↓	0	—

Logic operations (11)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	Addressing Mode								Operation ⁵					Condition Codes ^{6,7}											
						16-Bit Instruction Fetch Execution Status ¹	Prx	Rn	@ERn	@(d,ERn)	@(d,Rn,WRn,L)	@ERn/@ERn+/@ERn+/@+ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C						
AND	AND.B @(d:32,Rs.W),@ERd-	B	5	6	8					S	D					@ERd	^	@(d:32+Rs)	→	@ERd			ERd32-1→ERd32	—	—	↑	↓	0	—	
	AND.B @(d:32,Rs.W),@+ERd	B	5	7	9					S	D				ERd32+1→ERd32	@ERd	^	@(d:32+Rs)	→	@ERd				—	—	↑	↓	0	—	
	AND.B @(d:32,Rs.W),@-ERd	B	5	7	9					S	D				ERd32-1→ERd32	@ERd	^	@(d:32+Rs)	→	@ERd				—	—	↑	↓	0	—	
	AND.B @(d:32,Rs.W),@(d:2,ERd)	B	5	7	9					D	S					@(1/2/3+ERd)	^	@(d:32+Rs)	→	@(1/2/3+ERd)				—	—	↑	↓	0	—	
	AND.B @(d:32,Rs.W),@(d:16,ERd)	B	6	7	9					D	S					@(d:16+ERd)	^	@(d:32+Rs)	→	@(d:16+ERd)				—	—	↑	↓	0	—	
	AND.B @(d:32,Rs.W),@(d:32,ERd)	B	7	7	1					D	S					@(d:32+ERd)	^	@(d:32+Rs)	→	@(d:32+ERd)				—	—	↑	↓	0	—	
	AND.B @(d:32,Rs.W),@(d:16,Rd.B)	B	6	7	9						S					@(d:16+RdL)	^	@(d:32+Rs)	→	@(d:16+RdL)				—	—	↑	↓	0	—	
	AND.B @(d:32,Rs.W),@(d:16,Rd.W)	B	6	7	9						S					@(d:16+Rd)	^	@(d:32+Rs)	→	@(d:16+Rd)				—	—	↑	↓	0	—	
	AND.B @(d:32,Rs.W),@(d:16,ERdL)	B	6	7	9						S					@(d:16+ERd)	^	@(d:32+Rs)	→	@(d:16+ERd)				—	—	↑	↓	0	—	
	AND.B @(d:32,Rs.W),@(d:32,Rd.B)	B	7	7	1						S					@(d:32+RdL)	^	@(d:32+Rs)	→	@(d:32+RdL)				—	—	↑	↓	0	—	
	AND.B @(d:32,Rs.W),@(d:32,Rd.W)	B	7	7	1						S					@(d:32+Rd)	^	@(d:32+Rs)	→	@(d:32+Rd)				—	—	↑	↓	0	—	
	AND.B @(d:32,Rs.W),@(d:32,ERdL)	B	7	7	1						S					@(d:32+ERd)	^	@(d:32+Rs)	→	@(d:32+ERd)				—	—	↑	↓	0	—	
	AND.B @(d:32,Rs.W),@aa:16	B	6	6	9						S	D				@aa:16	^	@(d:32+Rs)	→	@aa:16				—	—	↑	↓	0	—	
	AND.B @(d:32,Rs.W),@aa:32	B	7	6	1						S	D				@aa:32	^	@(d:32+Rs)	→	@aa:32				—	—	↑	↓	0	—	
	AND.B @(d:32,ERs.L),@ERd	B	5	6	8				D		S					@ERd	^	@(d:32+ERs)	→	@ERd				—	—	↑	↓	0	—	
	AND.B @(d:32,ERs.L),@ERd+	B	5	6	8						S	D				@ERd	^	@(d:32+ERs)	→	@ERd			ERd32+1→ERd32	—	—	↑	↓	0	—	
	AND.B @(d:32,ERs.L),@ERd-	B	5	6	8						S	D				@ERd	^	@(d:32+ERs)	→	@ERd			ERd32-1→ERd32	—	—	↑	↓	0	—	
	AND.B @(d:32,ERs.L),@+ERd	B	5	7	9						S	D				ERd32+1→ERd32	@ERd	^	@(d:32+ERs)	→	@ERd				—	—	↑	↓	0	—
	AND.B @(d:32,ERs.L),@-ERd	B	5	7	9						S	D				ERd32-1→ERd32	@ERd	^	@(d:32+ERs)	→	@ERd				—	—	↑	↓	0	—
	AND.B @(d:32,ERs.L),@(d:2,ERd)	B	5	7	9					D	S					@(1/2/3+ERd)	^	@(d:32+ERs)	→	@(1/2/3+ERd)				—	—	↑	↓	0	—	
	AND.B @(d:32,ERs.L),@(d:16,ERd)	B	6	7	9					D	S					@(d:16+ERd)	^	@(d:32+ERs)	→	@(d:16+ERd)				—	—	↑	↓	0	—	
	AND.B @(d:32,ERs.L),@(d:32,ERd)	B	7	7	1					D	S					@(d:32+ERd)	^	@(d:32+ERs)	→	@(d:32+ERd)				—	—	↑	↓	0	—	
	AND.B @(d:32,ERs.L),@(d:16,Rd.B)	B	6	7	9						S					@(d:16+RdL)	^	@(d:32+ERs)	→	@(d:16+RdL)				—	—	↑	↓	0	—	
	AND.B @(d:32,ERs.L),@(d:16,Rd.W)	B	6	7	9						S					@(d:16+Rd)	^	@(d:32+ERs)	→	@(d:16+Rd)				—	—	↑	↓	0	—	

Logic operations (13)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	Addressing Mode							Operation ⁵					Condition Codes ^{6,7}											
						Number of 16-Bit Instruction Fetch Execution Stages ¹	Rn	@ERn	@(d,ERn)	@(d,Rn,WRn,LRn)	@ERn/@ERn+/@ERn+/@ERn+/@ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C						
AND	AND.B @aa:32,@ERd-	B	4	4	7							D	S			@ERd	^	@aa:32	→	@ERd			ERd32-1→ERd32	—	—	↑	↓	0	—
	AND.B @aa:32,@+ERd	B	4	5	8							D	S		ERd32+1→ERd32	@ERd	^	@aa:32	→	@ERd				—	—	↑	↓	0	—
	AND.B @aa:32,@-ERd	B	4	5	8							D	S		ERd32-1→ERd32	@ERd	^	@aa:32	→	@ERd				—	—	↑	↓	0	—
	AND.B @aa:32,@(d:2,ERd)	B	4	5	8							D	S			@(1/2/3+ERd)	^	@aa:32	→	@(1/2/3+ERd)				—	—	↑	↓	0	—
	AND.B @aa:32,@(d:16,ERd)	B	5	5	8							D	S			@(d:16+ERd)	^	@aa:32	→	@(d:16+ERd)				—	—	↑	↓	0	—
	AND.B @aa:32,@(d:32,ERd)	B	6	5	9							D	S			@(d:32+ERd)	^	@aa:32	→	@(d:32+ERd)				—	—	↑	↓	0	—
	AND.B @aa:32,@(d:16,Rd.B)	B	5	5	8							D	S			@(d:16+RdL)	^	@aa:32	→	@(d:16+RdL)				—	—	↑	↓	0	—
	AND.B @aa:32,@(d:16,Rd.W)	B	5	5	8							D	S			@(d:16+Rd)	^	@aa:32	→	@(d:16+Rd)				—	—	↑	↓	0	—
	AND.B @aa:32,@(d:16,ERd.L)	B	5	5	8							D	S			@(d:16+ERd)	^	@aa:32	→	@(d:16+ERd)				—	—	↑	↓	0	—
	AND.B @aa:32,@(d:32,Rd.B)	B	6	5	9							D	S			@(d:32+RdL)	^	@aa:32	→	@(d:32+RdL)				—	—	↑	↓	0	—
	AND.B @aa:32,@(d:32,Rd.W)	B	6	5	9							D	S			@(d:32+Rd)	^	@aa:32	→	@(d:32+Rd)				—	—	↑	↓	0	—
	AND.B @aa:32,@(d:32,ERd.L)	B	6	5	9							D	S			@(d:32+ERd)	^	@aa:32	→	@(d:32+ERd)				—	—	↑	↓	0	—
	AND.B @aa:32,@aa:16	B	5	4	8								S			@aa:16	^	@aa:32	→	@aa:16				—	—	↑	↓	0	—
	AND.B @aa:32,@aa:32	B	6	4	9								S			@aa:32	^	@aa:32	→	@aa:32				—	—	↑	↓	0	—
	AND.W #xx:16,Rd	W	2	1	2	S	D								Rd16	^	#xx	→	Rd16					—	—	↑	↓	0	—
	AND.W #xx:16,@ERd	W	3	4	5	S		D							@ERd	^	#xx	→	@ERd					—	—	↑	↓	0	—
	AND.W #xx:16,@ERd+	W	3	4	5	S						D			@ERd	^	#xx	→	@ERd			ERd32+2→ERd32	—	—	↑	↓	0	—	
	AND.W #xx:16,@ERd-	W	3	4	5	S						D			@ERd	^	#xx	→	@ERd			ERd32-2→ERd32	—	—	↑	↓	0	—	
	AND.W #xx:16,@+ERd	W	3	5	5	S							D		ERd32+2→ERd32	@ERd	^	#xx	→	@ERd				—	—	↑	↓	0	—
	AND.W #xx:16,@-ERd	W	3	5	5	S							D		ERd32-2→ERd32	@ERd	^	#xx	→	@ERd				—	—	↑	↓	0	—
	AND.W #xx:16,@(d:2,ERd)	W	3	5	5	S							D			@(2/4/6+ERd)	^	#xx	→	@(2/4/6+ERd)				—	—	↑	↓	0	—
	AND.W #xx:16,@(d:16,ERd)	W	4	5	6	S							D			@(d:16+ERd)	^	#xx	→	@(d:16+ERd)				—	—	↑	↓	0	—
	AND.W #xx:16,@(d:32,ERd)	W	5	5	7	S							D			@(d:32+ERd)	^	#xx	→	@(d:32+ERd)				—	—	↑	↓	0	—
	AND.W #xx:16,@(d:16,Rd.B)	W	4	5	6	S							D			@(d:16+RdL<<1)	^	#xx	→	@(d:16+RdL<<1)				—	—	↑	↓	0	—

Logic operations (15)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode							Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	Condition Codes ^{s2}												
				Min.	Max.	16-Bit Instruction Fetch Execution Status ¹	Prx	Rn	@(d,ERn)	@(d,Rn,WRn,L)						@(d,ERn)/@ERn+/@ERn+/@ERn+/@ERn	@aa:8/@aa:16/@aa:32	I	H	N	Z	V	C					
AND	AND.W @ERs+,Rd	W	2	3	3		D					Rd16	^ @ERs	→ Rd16	ERs32+2→ERs32													
	AND.W @ERs-,Rd	W	2	3	3		D					Rd16	^ @ERs	→ Rd16	ERs32-2→ERs32													
	AND.W @+ERs,Rd	W	2	4	4		D					ERs32+2→ERs32	Rd16	^ @ERs	→ Rd16													
	AND.W @-ERs,Rd	W	2	4	4		D					ERs32-2→ERs32	Rd16	^ @ERs	→ Rd16													
	AND.W @(d:2,ERs),Rd	W	2	4	4		D		S			Rd16	^ @(2/4/6+ERs)	→ Rd16														
	AND.W @(d:16,ERs),Rd	W	3	4	4		D		S			Rd16	^ @(d:16+ERs)	→ Rd16														
	AND.W @(d:32,ERs),Rd	W	4	4	5		D		S			Rd16	^ @(d:32+ERs)	→ Rd16														
	AND.W @(d:16,Rs,B),Rd	W	3	4	4		D		S			Rd16	^ @(d:16+Rs<<1)	→ Rd16														
	AND.W @(d:16,Rs,W),Rd	W	3	4	4		D		S			Rd16	^ @(d:16+Rs<<1)	→ Rd16														
	AND.W @(d:16,ERs,L),Rd	W	3	4	4		D		S			Rd16	^ @(d:16+ERs<<1)	→ Rd16														
	AND.W @(d:32,Rs,B),Rd	W	4	4	5		D		S			Rd16	^ @(d:32+Rs<<1)	→ Rd16														
	AND.W @(d:32,Rs,W),Rd	W	4	4	5		D		S			Rd16	^ @(d:32+Rs<<1)	→ Rd16														
	AND.W @(d:32,ERs,L),Rd	W	4	4	5		D		S			Rd16	^ @(d:32+ERs<<1)	→ Rd16														
	AND.W @aa:16,Rd	W	3	3	4		D					Rd16	^ @aa:16	→ Rd16														
	AND.W @aa:32,Rd	W	4	3	5		D					Rd16	^ @aa:32	→ Rd16														
	AND.W @ERs,@ERd	W	2	4	5				S			@ERd	^ @ERs	→ @ERd														
	AND.W @ERs,@ERd+	W	2	4	5				S			@ERd	^ @ERs	→ @ERd														
	AND.W @ERs,@ERd-	W	2	4	5				S			@ERd	^ @ERs	→ @ERd														
	AND.W @ERs,@+ERd	W	2	5	6				S			ERd32+2→ERd32	@ERd	^ @ERs	→ @ERd													
	AND.W @ERs,@-ERd	W	2	5	6				S			ERd32-2→ERd32	@ERd	^ @ERs	→ @ERd													
	AND.W @ERs,@(d:2,ERd)	W	2	5	6				S	D		@(2/4/6+ERd)	^ @ERs	→ @(2/4/6+ERd)														
	AND.W @ERs,@(d:16,ERd)	W	3	5	6				S	D		@(d:16+ERd)	^ @ERs	→ @(d:16+ERd)														
	AND.W @ERs,@(d:32,ERd)	W	4	5	7				S	D		@(d:32+ERd)	^ @ERs	→ @(d:32+ERd)														
	AND.W @ERs,@(d:16,Rd,B)	W	3	5	6				S	D		@(d:16+Rd<<1)	^ @ERs	→ @(d:16+Rd<<1)														
	AND.W @ERs,@(d:16,Rd,W)	W	3	5	6				S	D		@(d:16+Rd<<1)	^ @ERs	→ @(d:16+Rd<<1)														

Logic operations (18)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode										Operation ⁵					Condition Codes ^{6,7}											
				Min.	16-Bit Instruction Fetch Execution Status ¹	Prx	Rn	@ERn	@(d,ERn)	@(d,Rn,ERn,WERn,L)	@.ERn/@.ERn+/@.ERn+/@.ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C							
AND	AND.W @+ERs,@(d:32,Rd,W)	W	5	7	8						D	S					ERs32-2→ERs32		@(d:32+Rd<<1) ^ @ERs	→	@(d:32+Rd<<1)				—	—	↑	↓	0	—
	AND.W @+ERs,@(d:32,ERd,L)	W	5	7	8						D	S					ERs32-2→ERs32		@(d:32+ERd<<1) ^ @ERs	→	@(d:32+ERd<<1)				—	—	↑	↓	0	—
	AND.W @+ERs,@aa:16	W	4	6	7							S	D				ERs32+2→ERs32		@aa:16 ^ @ERs	→	@aa:16				—	—	↑	↓	0	—
	AND.W @+ERs,@aa:32	W	5	6	8							S	D				ERs32+2→ERs32		@aa:32 ^ @ERs	→	@aa:32				—	—	↑	↓	0	—
	AND.W @-ERs,@ERd	W	3	6	6						D	S					ERs32-2→ERs32		@ERd ^ @ERs	→	@ERd				—	—	↑	↓	0	—
	AND.W @-ERs,@ERd+	W	3	6	6							S					ERs32-2→ERs32		@ERd ^ @ERs	→	@ERd		ERd32+2→ERd32		—	—	↑	↓	0	—
	AND.W @-ERs,@ERd-	W	3	6	6							S					ERs32-2→ERs32		@ERd ^ @ERs	→	@ERd		ERd32-2→ERd32		—	—	↑	↓	0	—
	AND.W @-ERs,@+ERd	W	3	7	7							S					ERs32-2→ERs32	ERd32+2→ERd32	@ERd ^ @ERs	→	@ERd				—	—	↑	↓	0	—
	AND.W @-ERs,@-ERd	W	3	7	7							S					ERs32-2→ERs32	ERd32-2→ERd32	@ERd ^ @ERs	→	@ERd				—	—	↑	↓	0	—
	AND.W @-ERs,@(d:2,ERd)	W	3	7	7						D	S					ERs32-2→ERs32		@(2/4/6+ERd) ^ @ERs	→	@(2/4/6+ERd)				—	—	↑	↓	0	—
	AND.W @-ERs,@(d:16,ERd)	W	4	7	7						D	S					ERs32-2→ERs32		@(d:16+ERd) ^ @ERs	→	@(d:16+ERd)				—	—	↑	↓	0	—
	AND.W @-ERs,@(d:32,ERd)	W	5	7	8						D	S					ERs32-2→ERs32		@(d:32+ERd) ^ @ERs	→	@(d:32+ERd)				—	—	↑	↓	0	—
	AND.W @-ERs,@(d:16,Rd,B)	W	4	7	7						D	S					ERs32-2→ERs32		@(d:16+RdL<<1) ^ @ERs	→	@(d:16+RdL<<1)				—	—	↑	↓	0	—
	AND.W @-ERs,@(d:16,Rd,W)	W	4	7	7						D	S					ERs32-2→ERs32		@(d:16+Rd<<1) ^ @ERs	→	@(d:16+Rd<<1)				—	—	↑	↓	0	—
	AND.W @-ERs,@(d:16,ERd,L)	W	4	7	7						D	S					ERs32-2→ERs32		@(d:16+ERd<<1) ^ @ERs	→	@(d:16+ERd<<1)				—	—	↑	↓	0	—
	AND.W @-ERs,@(d:32,Rd,B)	W	5	7	8						D	S					ERs32-2→ERs32		@(d:32+RdL<<1) ^ @ERs	→	@(d:32+RdL<<1)				—	—	↑	↓	0	—
	AND.W @-ERs,@(d:32,Rd,W)	W	5	7	8						D	S					ERs32-2→ERs32		@(d:32+Rd<<1) ^ @ERs	→	@(d:32+Rd<<1)				—	—	↑	↓	0	—
	AND.W @-ERs,@(d:32,ERd,L)	W	5	7	8						D	S					ERs32-2→ERs32		@(d:32+ERd<<1) ^ @ERs	→	@(d:32+ERd<<1)				—	—	↑	↓	0	—
	AND.W @-ERs,@aa:16	W	4	6	7							S	D				ERs32-2→ERs32		@aa:16 ^ @ERs	→	@aa:16				—	—	↑	↓	0	—
	AND.W @-ERs,@aa:32	W	5	6	8							S	D				ERs32-2→ERs32		@aa:32 ^ @ERs	→	@aa:32				—	—	↑	↓	0	—
	AND.W @(d:2,ERs),@ERd	W	3	6	6						D	S							@ERd ^ @(2/4/6+ERs)	→	@ERd				—	—	↑	↓	0	—
	AND.W @(d:2,ERs),@ERd+	W	3	6	6							S	D						@ERd ^ @(2/4/6+ERs)	→	@ERd		ERd32+2→ERd32		—	—	↑	↓	0	—
	AND.W @(d:2,ERs),@ERd-	W	3	6	6							S	D						@ERd ^ @(2/4/6+ERs)	→	@ERd		ERd32-2→ERd32		—	—	↑	↓	0	—
	AND.W @(d:2,ERs),@+ERd	W	3	7	7							S	D					ERd32+2→ERd32	@ERd ^ @(2/4/6+ERs)	→	@ERd				—	—	↑	↓	0	—
	AND.W @(d:2,ERs),@-ERd	W	3	7	7							S	D					ERd32-2→ERd32	@ERd ^ @(2/4/6+ERs)	→	@ERd				—	—	↑	↓	0	—



Logic operations (25)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode											Operation ⁶					Condition Code ^{8,2}													
				Min.	16-Bit Instruction Fetch Execution Status ¹	FXZ	Rn	@(d,ERn)	@(d,Rn,WRn,L)	@(d,ERn)/@ERn+/@ERn-/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4												Operation 5	I	H	N	Z	V	C	
AND	AND.W @aa:16, @(d:16,Rd.B)	W	4	5	7					D	S					@(d:16+RdL<<1) ^ @aa:16 → @(d:16+RdL<<1)																	
	AND.W @aa:16, @(d:16,Rd.W)	W	4	5	7					D	S					@(d:16+Rd<<1) ^ @aa:16 → @(d:16+Rd<<1)																	
	AND.W @aa:16, @(d:16,ERd.L)	W	4	5	7					D	S					@(d:16+ERd<<1) ^ @aa:16 → @(d:16+ERd<<1)																	
	AND.W @aa:16, @(d:32,Rd.B)	W	5	5	8					D	S					@(d:32+RdL<<1) ^ @aa:16 → @(d:32+RdL<<1)																	
	AND.W @aa:16, @(d:32,Rd.W)	W	5	5	8					D	S					@(d:32+Rd<<1) ^ @aa:16 → @(d:32+Rd<<1)																	
	AND.W @aa:16, @(d:32,ERd.L)	W	5	5	8					D	S					@(d:32+ERd<<1) ^ @aa:16 → @(d:32+ERd<<1)																	
	AND.W @aa:16, @aa:16	W	4	4	7						S					@aa:16 ^ @aa:16 → @aa:16																	
	AND.W @aa:16, @aa:32	W	5	4	8						S					@aa:32 ^ @aa:16 → @aa:32																	
	AND.W @aa:32, @ERd	W	4	4	7				D		S					@ERd ^ @aa:32 → @ERd																	
	AND.W @aa:32, @ERd+	W	4	4	7					D	S					@ERd ^ @aa:32 → @ERd				ERd32+2 → ERd32													
	AND.W @aa:32, @ERd-	W	4	4	7					D	S					@ERd ^ @aa:32 → @ERd				ERd32-2 → ERd32													
	AND.W @aa:32, @+ERd	W	4	5	8					D	S			ERd32+2 → ERd32		@ERd ^ @aa:32 → @ERd																	
	AND.W @aa:32, @-ERd	W	4	5	8					D	S			ERd32-2 → ERd32		@ERd ^ @aa:32 → @ERd																	
	AND.W @aa:32, @(d:2,ERd)	W	4	5	8				D		S					@(2/4/6+ERd) ^ @aa:32 → @(2/4/6+ERd)																	
	AND.W @aa:32, @(d:16,ERd)	W	5	5	8				D		S					@(d:16+ERd) ^ @aa:32 → @(d:16+ERd)																	
	AND.W @aa:32, @(d:32,ERd)	W	6	5	9				D		S					@(d:32+ERd) ^ @aa:32 → @(d:32+ERd)																	
	AND.W @aa:32, @(d:16,Rd.B)	W	5	5	8					D	S					@(d:16+RdL<<1) ^ @aa:32 → @(d:16+RdL<<1)																	
	AND.W @aa:32, @(d:16,Rd.W)	W	5	5	8					D	S					@(d:16+Rd<<1) ^ @aa:32 → @(d:16+Rd<<1)																	
	AND.W @aa:32, @(d:16,ERd.L)	W	5	5	8					D	S					@(d:16+ERd<<1) ^ @aa:32 → @(d:16+ERd<<1)																	
	AND.W @aa:32, @(d:32,Rd.B)	W	6	5	9					D	S					@(d:32+RdL<<1) ^ @aa:32 → @(d:32+RdL<<1)																	
	AND.W @aa:32, @(d:32,Rd.W)	W	6	5	9					D	S					@(d:32+Rd<<1) ^ @aa:32 → @(d:32+Rd<<1)																	
	AND.W @aa:32, @(d:32,ERd.L)	W	6	5	9					D	S					@(d:32+ERd<<1) ^ @aa:32 → @(d:32+ERd<<1)																	
	AND.W @aa:32, @aa:16	W	5	4	8						S					@aa:16 ^ @aa:32 → @aa:16																	
	AND.W @aa:32, @aa:32	W	6	4	9						S					@aa:32 ^ @aa:32 → @aa:32																	

Logic operations (26)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode											Operation ⁵					Condition Codes ^{6,7}															
				Number of		16-Bit Instruction Fetch		Execution Status ⁸		Rn		@ERn		@ERn/@ERn.W/ERn.L												@ERn/@ERn+/@ERn+/@ERn		@aa:8/@aa:16/@aa:32							
				Min.	Max.	16-Bit	32-Bit	Fetch	Fetch	Execution	Execution	Rn	@ERn	@ERn	@ERn	@ERn	@ERn	@ERn	@ERn	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C					
AND	AND.L #xx:16, ERd	L	2	1	2	S	D												ERd32	^	#xx:16	→	ERd32				—	—	↑	↓	0	—			
	AND.L #xx:32, ERd	L	3	1	3	S	D												ERd32	^	#xx	→	ERd32				—	—	↑	↓	0	—			
	AND.L #xx:16, @ERd	L	3	4	5	S	D												@ERd	^	#xx:16	→	@ERd				—	—	↑	↓	0	—			
	AND.L #xx:16, @ERd+	L	3	4	5	S													@ERd	^	#xx:16	→	@ERd		ERd32+4	→	ERd32			—	—	↑	↓	0	—
	AND.L #xx:16, @ERd-	L	3	4	5	S							D						@ERd	^	#xx:16	→	@ERd		ERd32-4	→	ERd32			—	—	↑	↓	0	—
	AND.L #xx:16, @+ERd	L	3	5	5	S							D				ERd32+4	→	ERd32	@ERd	^	#xx:16	→	@ERd						—	—	↑	↓	0	—
	AND.L #xx:16, @-ERd	L	3	5	5	S							D				ERd32-4	→	ERd32	@ERd	^	#xx:16	→	@ERd						—	—	↑	↓	0	—
	AND.L #xx:16, @(d:2, ERd)	L	3	5	5	S							D						@(4/8/12+ERd)	^	#xx:16	→	@(4/8/12+ERd)						—	—	↑	↓	0	—	
	AND.L #xx:16, @(d:16, ERd)	L	4	5	6	S							D						@(d:16+ERd)	^	#xx:16	→	@(d:16+ERd)						—	—	↑	↓	0	—	
	AND.L #xx:16, @(d:32, ERd)	L	5	5	7	S							D						@(d:32+ERd)	^	#xx:16	→	@(d:32+ERd)						—	—	↑	↓	0	—	
	AND.L #xx:16, @(d:16, Rd.B)	L	4	5	6	S							D						@(d:16+RdL<<2)	^	#xx:16	→	@(d:16+RdL<<2)						—	—	↑	↓	0	—	
	AND.L #xx:16, @(d:16, Rd.W)	L	4	5	6	S							D						@(d:16+Rd<<2)	^	#xx:16	→	@(d:16+Rd<<2)						—	—	↑	↓	0	—	
	AND.L #xx:16, @(d:16, ERd.L)	L	4	5	6	S							D						@(d:16+ERd<<2)	^	#xx:16	→	@(d:16+ERd<<2)						—	—	↑	↓	0	—	
	AND.L #xx:16, @(d:32, Rd.B)	L	5	5	7	S							D						@(d:32+RdL<<2)	^	#xx:16	→	@(d:32+RdL<<2)						—	—	↑	↓	0	—	
	AND.L #xx:16, @(d:32, Rd.W)	L	5	5	7	S							D						@(d:32+Rd<<2)	^	#xx:16	→	@(d:32+Rd<<2)						—	—	↑	↓	0	—	
	AND.L #xx:16, @(d:32, ERd.L)	L	5	5	7	S							D						@(d:32+ERd<<2)	^	#xx:16	→	@(d:32+ERd<<2)						—	—	↑	↓	0	—	
	AND.L #xx:16, @aa:16	L	4	4	6	S								D					@aa:16	^	#xx:16	→	@aa:16						—	—	↑	↓	0	—	
	AND.L #xx:16, @aa:32	L	5	4	7	S									D				@aa:32	^	#xx:16	→	@aa:32						—	—	↑	↓	0	—	
	AND.L #xx:32, @ERd	L	4	4	6	S							D						@ERd	^	#xx:16	→	@ERd						—	—	↑	↓	0	—	
	AND.L #xx:32, @ERd+	L	4	4	6	S								D					@ERd	^	#xx:16	→	@ERd		ERd32+4	→	ERd32			—	—	↑	↓	0	—
	AND.L #xx:32, @ERd-	L	4	4	6	S								D					@ERd	^	#xx:16	→	@ERd		ERd32-4	→	ERd32			—	—	↑	↓	0	—
	AND.L #xx:32, @+ERd	L	4	5	6	S								D			ERd32+4	→	ERd32	@ERd	^	#xx:16	→	@ERd						—	—	↑	↓	0	—
	AND.L #xx:32, @-ERd	L	4	5	6	S								D			ERd32-4	→	ERd32	@ERd	^	#xx:16	→	@ERd						—	—	↑	↓	0	—
	AND.L #xx:32, @(d:2, ERd)	L	4	5	6	S								D					@(4/8/12+ERd)	^	#xx:16	→	@(4/8/12+ERd)						—	—	↑	↓	0	—	
	AND.L #xx:32, @(d:16, ERd)	L	5	5	7	S								D					@(d:16+ERd)	^	#xx:16	→	@(d:16+ERd)						—	—	↑	↓	0	—	



Logic operations (29)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode								Operation ⁶					Condition Codes ^{5,2}					
				Min.	Max.	16-Bit Instruction Fetch Execution Status ¹	#rx	Rn	@(d, ERn)	@(d, ERn, W, ERn, L)	@(d, ERn)/@ERn+/@ERn-/@+ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V
AND	AND.L @ERs, @(d:16, Rd, B)	L	4	6	7		S	D				@(d:16+RdL<<2) ^ @ERs → @(d:16+RdL<<2)				—	—	↑	↓	0	—	
	AND.L @ERs, @(d:16, Rd, W)	L	4	6	7		S	D				@(d:16+Rd<<2) ^ @ERs → @(d:16+Rd<<2)				—	—	↑	↓	0	—	
	AND.L @ERs, @(d:16, ERd, L)	L	4	6	7		S	D				@(d:16+ERd<<2) ^ @ERs → @(d:16+ERd<<2)				—	—	↑	↓	0	—	
	AND.L @ERs, @(d:32, Rd, B)	L	5	6	8		S	D				@(d:32+RdL<<2) ^ @ERs → @(d:32+RdL<<2)				—	—	↑	↓	0	—	
	AND.L @ERs, @(d:32, Rd, W)	L	5	6	8		S	D				@(d:32+Rd<<2) ^ @ERs → @(d:32+Rd<<2)				—	—	↑	↓	0	—	
	AND.L @ERs, @(d:32, ERd, L)	L	5	6	8		S	D				@(d:32+ERd<<2) ^ @ERs → @(d:32+ERd<<2)				—	—	↑	↓	0	—	
	AND.L @ERs, @aa:16	L	4	5	7		S			D		@aa:16 ^ @ERs → @aa:16				—	—	↑	↓	0	—	
	AND.L @ERs, @aa:32	L	5	5	8		S			D		@aa:32 ^ @ERs → @aa:32				—	—	↑	↓	0	—	
	AND.L @ERs+, @ERd	L	3	5	6		D			S		@ERd ^ @ERs → @ERd	ERs32+4→ERs32			—	—	↑	↓	0	—	
	AND.L @ERs+, @ERd+	L	3	5	6					S		@ERd ^ @ERs → @ERd	ERs32+4→ERs32	ERd32+4→ERd32		—	—	↑	↓	0	—	
	AND.L @ERs+, @ERd-	L	3	5	6					S		@ERd ^ @ERs → @ERd	ERs32+4→ERs32	ERd32-4→ERd32		—	—	↑	↓	0	—	
	AND.L @ERs+, @+ERd	L	3	6	7					S		@ERd ^ @ERs → @ERd	ERd32+4→ERd32		ERs32+4→ERs32		—	—	↑	↓	0	—
	AND.L @ERs+, @-ERd	L	3	6	7					S		@ERd ^ @ERs → @ERd	ERd32-4→ERd32		ERs32+4→ERs32		—	—	↑	↓	0	—
	AND.L @ERs+, @(d:2, ERd)	L	3	6	7			D	S			@(4/8/12+ERd) ^ @ERs → @(4/8/12+ERd)	ERs32+4→ERs32			—	—	↑	↓	0	—	
	AND.L @ERs+, @(d:16, ERd)	L	4	6	7			D	S			@(d:16+ERd) ^ @ERs → @(d:16+ERd)	ERs32+4→ERs32			—	—	↑	↓	0	—	
	AND.L @ERs+, @(d:32, ERd)	L	5	6	8			D	S			@(d:32+ERd) ^ @ERs → @(d:32+ERd)	ERs32+4→ERs32			—	—	↑	↓	0	—	
	AND.L @ERs+, @(d:16, Rd, B)	L	4	6	7			D	S			@(d:16+RdL<<2) ^ @ERs → @(d:16+RdL<<2)	ERs32+4→ERs32			—	—	↑	↓	0	—	
	AND.L @ERs+, @(d:16, Rd, W)	L	4	6	7			D	S			@(d:16+Rd<<2) ^ @ERs → @(d:16+Rd<<2)	ERs32+4→ERs32			—	—	↑	↓	0	—	
	AND.L @ERs+, @(d:16, ERd, L)	L	4	6	7			D	S			@(d:16+ERd<<2) ^ @ERs → @(d:16+ERd<<2)	ERs32+4→ERs32			—	—	↑	↓	0	—	
	AND.L @ERs+, @(d:32, Rd, B)	L	5	6	8			D	S			@(d:32+RdL<<2) ^ @ERs → @(d:32+RdL<<2)	ERs32+4→ERs32			—	—	↑	↓	0	—	
	AND.L @ERs+, @(d:32, Rd, W)	L	5	6	8			D	S			@(d:32+Rd<<2) ^ @ERs → @(d:32+Rd<<2)	ERs32+4→ERs32			—	—	↑	↓	0	—	
	AND.L @ERs+, @(d:32, ERd, L)	L	5	6	8			D	S			@(d:32+ERd<<2) ^ @ERs → @(d:32+ERd<<2)	ERs32+4→ERs32			—	—	↑	↓	0	—	
	AND.L @ERs+, @aa:16	L	4	5	7			S	D			@aa:16 ^ @ERs → @aa:16	ERs32+4→ERs32			—	—	↑	↓	0	—	
	AND.L @ERs+, @aa:32	L	5	5	8			S	D			@aa:32 ^ @ERs → @aa:32	ERs32+4→ERs32			—	—	↑	↓	0	—	
	AND.L @ERs-, @ERd	L	3	5	6		D			S		@ERd ^ @ERs → @ERd	ERs32-4→ERs32			—	—	↑	↓	0	—	

Logic operations (30)

Instruction n	Mnemonic	Size	Instruction Length		Number of 16-bit Instruction Fetch Execution Stages ¹ #xx	Addressing Mode						Operation ⁵					Condition Codes ^{2,3}													
			Min.	Max.		Rn	@ERn	@(d,ERn)	@(d,Rn,ERn,WERn,L)	@ERn/@ERn+/@ERn-/@+ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C								
			AND	ANDL @ERs, @ERd+		L	3	5	6									@ERd	∧	@ERs	→	@ERd	ERs32-4→ERs32	ERd32+4→ERd32	—	—	↑	↓	0	—
	ANDL @ERs, @ERd-	L	3	5	6									@ERd	∧	@ERs	→	@ERd	ERs32-4→ERs32	ERd32-4→ERd32	—	—	↑	↓	0	—				
	ANDL @ERs, @+ERd	L	3	6	7									@ERd	∧	@ERs	→	@ERd	ERs32-4→ERs32		—	—	↑	↓	0	—				
	ANDL @ERs, @-ERd	L	3	6	7									@ERd	∧	@ERs	→	@ERd	ERs32-4→ERs32		—	—	↑	↓	0	—				
	ANDL @ERs, @(d:2,ERd)	L	3	6	7							D	S	@(4/8/12+ERd)	∧	@ERs	→	@(4/8/12+ERd)	ERs32-4→ERs32		—	—	↑	↓	0	—				
	ANDL @ERs, @(d:16,ERd)	L	4	6	7							D	S	@(d:16+ERd)	∧	@ERs	→	@(d:16+ERd)	ERs32-4→ERs32		—	—	↑	↓	0	—				
	ANDL @ERs, @(d:32,ERd)	L	5	6	8							D	S	@(d:32+ERd)	∧	@ERs	→	@(d:32+ERd)	ERs32-4→ERs32		—	—	↑	↓	0	—				
	ANDL @ERs, @(d:16,Rd,B)	L	4	6	7							D	S	@(d:16+Rd<<2)	∧	@ERs	→	@(d:16+Rd<<2)	ERs32-4→ERs32		—	—	↑	↓	0	—				
	ANDL @ERs, @(d:16,Rd,W)	L	4	6	7							D	S	@(d:16+Rd<<2)	∧	@ERs	→	@(d:16+Rd<<2)	ERs32-4→ERs32		—	—	↑	↓	0	—				
	ANDL @ERs, @(d:16,ERd,L)	L	4	6	7							D	S	@(d:16+ERd<<2)	∧	@ERs	→	@(d:16+ERd<<2)	ERs32-4→ERs32		—	—	↑	↓	0	—				
	ANDL @ERs, @(d:32,Rd,B)	L	5	6	8							D	S	@(d:32+Rd<<2)	∧	@ERs	→	@(d:32+Rd<<2)	ERs32-4→ERs32		—	—	↑	↓	0	—				
	ANDL @ERs, @(d:32,Rd,W)	L	5	6	8							D	S	@(d:32+Rd<<2)	∧	@ERs	→	@(d:32+Rd<<2)	ERs32-4→ERs32		—	—	↑	↓	0	—				
	ANDL @ERs, @(d:32,ERd,L)	L	5	6	8							D	S	@(d:32+ERd<<2)	∧	@ERs	→	@(d:32+ERd<<2)	ERs32-4→ERs32		—	—	↑	↓	0	—				
	ANDL @ERs, @aa:16	L	4	5	7									S	D	@aa:16	∧	@ERs	→	@aa:16	ERs32-4→ERs32		—	—	↑	↓	0	—		
	ANDL @ERs, @aa:32	L	5	5	8									S	D	@aa:32	∧	@ERs	→	@aa:32	ERs32-4→ERs32		—	—	↑	↓	0	—		
	ANDL @+ERs, @ERd	L	3	6	6								D	S	ERs32+4→ERs32		@ERd	∧	@ERs	→	@ERd		—	—	↑	↓	0	—		
	ANDL @+ERs, @ERd+	L	3	6	6									S		ERs32+4→ERs32		@ERd	∧	@ERs	→	@ERd		ERd32+4→ERd32	—	—	↑	↓	0	—
	ANDL @+ERs, @ERd-	L	3	6	6									S		ERs32+4→ERs32		@ERd	∧	@ERs	→	@ERd		ERd32-4→ERd32	—	—	↑	↓	0	—
	ANDL @+ERs, @+ERd	L	3	7	7									S		ERs32+4→ERs32	ERd32+4→ERd32	@ERd	∧	@ERs	→	@ERd			—	—	↑	↓	0	—
	ANDL @+ERs, @-ERd	L	3	7	7									S		ERs32+4→ERs32	ERd32-4→ERd32	@ERd	∧	@ERs	→	@ERd			—	—	↑	↓	0	—
	ANDL @+ERs, @(d:2,ERd)	L	3	7	7							D	S	ERs32+4→ERs32		@(4/8/12+ERd)	∧	@ERs	→	@(4/8/12+ERd)		—	—	↑	↓	0	—			
	ANDL @+ERs, @(d:16,ERd)	L	4	7	7							D	S	ERs32+4→ERs32		@(d:16+ERd)	∧	@ERs	→	@(d:16+ERd)		—	—	↑	↓	0	—			
	ANDL @+ERs, @(d:32,ERd)	L	5	7	8							D	S	ERs32+4→ERs32		@(d:32+ERd)	∧	@ERs	→	@(d:32+ERd)		—	—	↑	↓	0	—			
	ANDL @+ERs, @(d:16,Rd,B)	L	4	7	7							D	S	ERs32+4→ERs32		@(d:16+Rd<<2)	∧	@ERs	→	@(d:16+Rd<<2)		—	—	↑	↓	0	—			
	ANDL @+ERs, @(d:16,Rd,W)	L	4	7	7							D	S	ERs32+4→ERs32		@(d:16+Rd<<2)	∧	@ERs	→	@(d:16+Rd<<2)		—	—	↑	↓	0	—			



Logic operations (32)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode										Operation ⁵					Condition Codes ^{6,7}										
				Min.	16-Bit Instruction Fetch Execution Stages ¹	Prx	Rn	@ERn	@(d,ERn)	@(d,Rn,WRn,LRn)	@ERn/@ERn+/@ERn+/@ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C						
AND	AND.L @(d:2,ERs),@ERd-	L	3	6	6					S	D					@ERd	^	@(4/8/12+ERs)	→	@ERd			ERd32-4→ERd32	—	—	↑	↓	0	—
	AND.L @(d:2,ERs),@+ERd	L	3	7	7					S	D				ERd32+4→ERd32	@ERd	^	@(4/8/12+ERs)	→	@ERd				—	—	↑	↓	0	—
	AND.L @(d:2,ERs),@-ERd	L	3	7	7					S	D				ERd32-4→ERd32	@ERd	^	@(4/8/12+ERs)	→	@ERd				—	—	↑	↓	0	—
	AND.L @(d:2,ERs),@(d:2,ERd)	L	3	7	7					S						@(4/8/12+ERd)	^	@(4/8/12+ERs)	→	@(4/8/12+ERd)				—	—	↑	↓	0	—
	AND.L @(d:2,ERs),@(d:16,ERd)	L	4	7	7					S						@(d:16+ERd)	^	@(4/8/12+ERs)	→	@(d:16+ERd)				—	—	↑	↓	0	—
	AND.L @(d:2,ERs),@(d:32,ERd)	L	5	7	8					S						@(d:32+ERd)	^	@(4/8/12+ERs)	→	@(d:32+ERd)				—	—	↑	↓	0	—
	AND.L @(d:2,ERs),@(d:16,Rd,B)	L	4	7	7					S	D					@(d:16+RdL<<2)	^	@(4/8/12+ERs)	→	@(d:16+RdL<<2)				—	—	↑	↓	0	—
	AND.L @(d:2,ERs),@(d:16,Rd,W)	L	4	7	7					S	D					@(d:16+Rd<<2)	^	@(4/8/12+ERs)	→	@(d:16+Rd<<2)				—	—	↑	↓	0	—
	AND.L @(d:2,ERs),@(d:16,ERd,L)	L	4	7	7					S	D					@(d:16+ERdL<<2)	^	@(4/8/12+ERs)	→	@(d:16+ERdL<<2)				—	—	↑	↓	0	—
	AND.L @(d:2,ERs),@(d:32,Rd,B)	L	5	7	8					S	D					@(d:32+RdL<<2)	^	@(4/8/12+ERs)	→	@(d:32+RdL<<2)				—	—	↑	↓	0	—
	AND.L @(d:2,ERs),@(d:32,Rd,W)	L	5	7	8					S	D					@(d:32+Rd<<2)	^	@(4/8/12+ERs)	→	@(d:32+Rd<<2)				—	—	↑	↓	0	—
	AND.L @(d:2,ERs),@(d:32,ERd,L)	L	5	7	8					S	D					@(d:32+ERdL<<2)	^	@(4/8/12+ERs)	→	@(d:32+ERdL<<2)				—	—	↑	↓	0	—
	AND.L @(d:2,ERs),@aa:16	L	4	6	7					S		D				@aa:16	^	@(4/8/12+ERs)	→	@aa:16				—	—	↑	↓	0	—
	AND.L @(d:2,ERs),@aa:32	L	5	6	8					S		D				@aa:32	^	@(4/8/12+ERs)	→	@aa:32				—	—	↑	↓	0	—
	AND.L @(d:16,ERs),@ERd	L	4	6	7				D	S						@ERd	^	@(d:16+ERs)	→	@ERd				—	—	↑	↓	0	—
	AND.L @(d:16,ERs),@ERd+	L	4	6	7					S	D					@ERd	^	@(d:16+ERs)	→	@ERd			ERd32+4→ERd32	—	—	↑	↓	0	—
	AND.L @(d:16,ERs),@ERd-	L	4	6	7					S	D					@ERd	^	@(d:16+ERs)	→	@ERd			ERd32-4→ERd32	—	—	↑	↓	0	—
	AND.L @(d:16,ERs),@+ERd	L	4	7	8					S	D				ERd32+4→ERd32	@ERd	^	@(d:16+ERs)	→	@ERd				—	—	↑	↓	0	—
	AND.L @(d:16,ERs),@-ERd	L	4	7	8					S	D				ERd32-4→ERd32	@ERd	^	@(d:16+ERs)	→	@ERd				—	—	↑	↓	0	—
	AND.L @(d:16,ERs),@(d:2,ERd)	L	4	7	8					S						@(4/8/12+ERd)	^	@(d:16+ERs)	→	@(4/8/12+ERd)				—	—	↑	↓	0	—
	AND.L @(d:16,ERs),@(d:16,ERd)	L	5	7	8					S						@(d:16+ERd)	^	@(d:16+ERs)	→	@(d:16+ERd)				—	—	↑	↓	0	—
	AND.L @(d:16,ERs),@(d:32,ERd)	L	6	7	9					S						@(d:32+ERd)	^	@(d:16+ERs)	→	@(d:32+ERd)				—	—	↑	↓	0	—
	AND.L @(d:16,ERs),@(d:16,Rd,B)	L	5	7	8					S	D					@(d:16+RdL<<2)	^	@(d:16+ERs)	→	@(d:16+RdL<<2)				—	—	↑	↓	0	—
	AND.L @(d:16,ERs),@(d:16,Rd,W)	L	5	7	8					S	D					@(d:16+Rd<<2)	^	@(d:16+ERs)	→	@(d:16+Rd<<2)				—	—	↑	↓	0	—
	AND.L @(d:16,ERs),@(d:16,ERd,L)	L	5	7	8					S	D					@(d:16+ERdL<<2)	^	@(d:16+ERs)	→	@(d:16+ERdL<<2)				—	—	↑	↓	0	—



Logic operations (43)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	Addressing Mode										Operation ⁵					Condition Codes ^{6,7}									
						Number of Execution Stages ¹	16-Bit Instruction Fetch #xxx	Rn	@ERn	@(d,ERn)	@(d,Rn,ERn,WERn,L)	@ERn/@ERn+/@ERn+/@ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C						
OR	OR.B @ERs, @(d:16,Rd,W)	B	4	6	7							D	S				@(d:16+Rd)	∨	@ERs	→	@(d:16+Rd)	ERs32-1→ERs32				∧	∨	0	∞	
	OR.B @ERs, @(d:16,ERd,L)	B	4	6	7							D	S				@(d:16+ERd)	∨	@ERs	→	@(d:16+ERd)	ERs32-1→ERs32				∧	∨	0	∞	
	OR.B @ERs, @(d:32,Rd,B)	B	5	6	8							D	S				@(d:32+RdL)	∨	@ERs	→	@(d:32+RdL)	ERs32-1→ERs32				∧	∨	0	∞	
	OR.B @ERs, @(d:32,Rd,W)	B	5	6	8							D	S				@(d:32+Rd)	∨	@ERs	→	@(d:32+Rd)	ERs32-1→ERs32				∧	∨	0	∞	
	OR.B @ERs, @(d:32,ERd,L)	B	5	6	8							D	S				@(d:32+ERd)	∨	@ERs	→	@(d:32+ERd)	ERs32-1→ERs32				∧	∨	0	∞	
	OR.B @ERs, @aa:16	B	4	5	7								S	D				@aa:16	∨	@ERs	→	@aa:16	ERs32-1→ERs32				∧	∨	0	∞
	OR.B @ERs, @aa:32	B	5	5	8								S	D				@aa:32	∨	@ERs	→	@aa:32	ERs32-1→ERs32				∧	∨	0	∞
	OR.B @+ERs, @ERd	B	3	6	6							D	S				ERs32+1→ERs32	@ERd	∨	@ERs	→	@ERd				∧	∨	0	∞	
	OR.B @+ERs, @ERd+	B	3	6	6								S				ERs32+1→ERs32	@ERd	∨	@ERs	→	@ERd	ERd32+1→ERd32				∧	∨	0	∞
	OR.B @+ERs, @ERd-	B	3	6	6								S				ERs32+1→ERs32	@ERd	∨	@ERs	→	@ERd	ERd32-1→ERd32				∧	∨	0	∞
	OR.B @+ERs, @+ERd	B	3	7	7								S				ERs32+1→ERs32	ERd32+1→ERd32	@ERd	∨	@ERs	→	@ERd				∧	∨	0	∞
	OR.B @+ERs, @-ERd	B	3	7	7								S				ERs32+1→ERs32	ERd32-1→ERd32	@ERd	∨	@ERs	→	@ERd				∧	∨	0	∞
	OR.B @+ERs, @(d:2,ERd)	B	3	7	7							D	S				ERs32+1→ERs32	@(1/2/3+ERd)	∨	@ERs	→	@(1/2/3+ERd)				∧	∨	0	∞	
	OR.B @+ERs, @(d:16,ERd)	B	4	7	7							D	S				ERs32+1→ERs32	@(d:16+ERd)	∨	@ERs	→	@(d:16+ERd)				∧	∨	0	∞	
	OR.B @+ERs, @(d:32,ERd)	B	5	7	8							D	S				ERs32+1→ERs32	@(d:32+ERd)	∨	@ERs	→	@(d:32+ERd)				∧	∨	0	∞	
	OR.B @+ERs, @(d:16,Rd,B)	B	4	7	7							D	S				ERs32+1→ERs32	@(d:16+RdL)	∨	@ERs	→	@(d:16+RdL)				∧	∨	0	∞	
	OR.B @+ERs, @(d:16,Rd,W)	B	4	7	7							D	S				ERs32+1→ERs32	@(d:16+Rd)	∨	@ERs	→	@(d:16+Rd)				∧	∨	0	∞	
	OR.B @+ERs, @(d:16,ERd,L)	B	4	7	7							D	S				ERs32+1→ERs32	@(d:16+ERd)	∨	@ERs	→	@(d:16+ERd)				∧	∨	0	∞	
	OR.B @+ERs, @(d:32,Rd,B)	B	5	7	8							D	S				ERs32+1→ERs32	@(d:32+RdL)	∨	@ERs	→	@(d:32+RdL)				∧	∨	0	∞	
	OR.B @+ERs, @(d:32,Rd,W)	B	5	7	8							D	S				ERs32+1→ERs32	@(d:32+Rd)	∨	@ERs	→	@(d:32+Rd)				∧	∨	0	∞	
OR.B @+ERs, @(d:32,ERd,L)	B	5	7	8							D	S				ERs32+1→ERs32	@(d:32+ERd)	∨	@ERs	→	@(d:32+ERd)				∧	∨	0	∞		
OR.B @+ERs, @aa:16	B	4	6	7							S	D				ERs32+1→ERs32	@aa:16	∨	@ERs	→	@aa:16				∧	∨	0	∞		
OR.B @+ERs, @aa:32	B	5	6	8							S	D				ERs32+1→ERs32	@aa:32	∨	@ERs	→	@aa:32				∧	∨	0	∞		
OR.B @ERs, @ERd	B	3	6	6							D	S				ERs32-1→ERs32	@ERd	∨	@ERs	→	@ERd				∧	∨	0	∞		
OR.B @ERs, @ERd+	B	3	6	6							S					ERs32-1→ERs32	@ERd	∨	@ERs	→	@ERd	ERd32+1→ERd32				∧	∨	0	∞	

Logic operations (44)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹	Addressing Mode						Operation ⁵					Condition Codes ²												
						Number of Execution Stages ¹	Rn	@ERn	@(d,ERn)	@(d,Rn,WRn,LL)	@.ERn/@.ERn+/@.ERn+/@.ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C						
OR	OR.B @-ERs, @ERd-	B	3	6	6						S		ERs32-1→ERs32		@ERd	∨	@ERs	→	@ERd			ERd32-1→ERd32	—	—	↑	↓	0	—	
	OR.B @-ERs, @+ERd	B	3	7	7						S		ERs32-1→ERs32	ERd32+1→ERd32	@ERd	∨	@ERs	→	@ERd				—	—	↑	↓	0	—	
	OR.B @-ERs, @-ERd	B	3	7	7						S		ERs32-1→ERs32	ERd32-1→ERd32	@ERd	∨	@ERs	→	@ERd				—	—	↑	↓	0	—	
	OR.B @-ERs, @(d:2,ERd)	B	3	7	7						D	S	ERs32-1→ERs32		@(1/2/3+ERd)	∨	@ERs	→	@(1/2/3+ERd)				—	—	↑	↓	0	—	
	OR.B @-ERs, @(d:16,ERd)	B	4	7	7						D	S	ERs32-1→ERs32		@(d:16+ERd)	∨	@ERs	→	@(d:16+ERd)				—	—	↑	↓	0	—	
	OR.B @-ERs, @(d:32,ERd)	B	5	7	8						D	S	ERs32-1→ERs32		@(d:32+ERd)	∨	@ERs	→	@(d:32+ERd)				—	—	↑	↓	0	—	
	OR.B @-ERs, @(d:16,Rd.B)	B	4	7	7						D	S	ERs32-1→ERs32		@(d:16+RdL)	∨	@ERs	→	@(d:16+RdL)				—	—	↑	↓	0	—	
	OR.B @-ERs, @(d:16,Rd.W)	B	4	7	7						D	S	ERs32-1→ERs32		@(d:16+Rd)	∨	@ERs	→	@(d:16+Rd)				—	—	↑	↓	0	—	
	OR.B @-ERs, @(d:16,ERd.L)	B	4	7	7						D	S	ERs32-1→ERs32		@(d:16+ERd)	∨	@ERs	→	@(d:16+ERd)				—	—	↑	↓	0	—	
	OR.B @-ERs, @(d:32,Rd.B)	B	5	7	8						D	S	ERs32-1→ERs32		@(d:32+RdL)	∨	@ERs	→	@(d:32+RdL)				—	—	↑	↓	0	—	
	OR.B @-ERs, @(d:32,Rd.W)	B	5	7	8						D	S	ERs32-1→ERs32		@(d:32+Rd)	∨	@ERs	→	@(d:32+Rd)				—	—	↑	↓	0	—	
	OR.B @-ERs, @(d:32,ERd.L)	B	5	7	8						D	S	ERs32-1→ERs32		@(d:32+ERd)	∨	@ERs	→	@(d:32+ERd)				—	—	↑	↓	0	—	
	OR.B @-ERs, @aa:16	B	4	6	7							S	D	ERs32-1→ERs32		@aa:16	∨	@ERs	→	@aa:16				—	—	↑	↓	0	—
	OR.B @-ERs, @aa:32	B	5	6	8							S	D	ERs32-1→ERs32		@aa:32	∨	@ERs	→	@aa:32				—	—	↑	↓	0	—
	OR.B @(d:2,ERs), @ERd	B	3	6	6						D	S			@ERd	∨	@(1/2/3+ERs)	→	@ERd				—	—	↑	↓	0	—	
	OR.B @(d:2,ERs), @ERd+	B	3	6	6						S	D			@ERd	∨	@(1/2/3+ERs)	→	@ERd			ERd32+1→ERd32	—	—	↑	↓	0	—	
	OR.B @(d:2,ERs), @ERd-	B	3	6	6						S	D			@ERd	∨	@(1/2/3+ERs)	→	@ERd			ERd32-1→ERd32	—	—	↑	↓	0	—	
	OR.B @(d:2,ERs), @+ERd	B	3	7	7						S	D		ERd32+1→ERd32	@ERd	∨	@(1/2/3+ERs)	→	@ERd				—	—	↑	↓	0	—	
	OR.B @(d:2,ERs), @-ERd	B	3	7	7						S	D		ERd32-1→ERd32	@ERd	∨	@(1/2/3+ERs)	→	@ERd				—	—	↑	↓	0	—	
	OR.B @(d:2,ERs), @(d:2,ERd)	B	3	7	7						S				@(1/2/3+ERd)	∨	@(1/2/3+ERs)	→	@(1/2/3+ERd)				—	—	↑	↓	0	—	
	OR.B @(d:2,ERs), @(d:16,ERd)	B	4	7	7						S				@(d:16+ERd)	∨	@(1/2/3+ERs)	→	@(d:16+ERd)				—	—	↑	↓	0	—	
	OR.B @(d:2,ERs), @(d:32,ERd)	B	5	7	8						S				@(d:32+ERd)	∨	@(1/2/3+ERs)	→	@(d:32+ERd)				—	—	↑	↓	0	—	
	OR.B @(d:2,ERs), @(d:16,Rd.B)	B	4	7	7						S	D			@(d:16+RdL)	∨	@(1/2/3+ERs)	→	@(d:16+RdL)				—	—	↑	↓	0	—	
	OR.B @(d:2,ERs), @(d:16,Rd.W)	B	4	7	7						S	D			@(d:16+Rd)	∨	@(1/2/3+ERs)	→	@(d:16+Rd)				—	—	↑	↓	0	—	
	OR.B @(d:2,ERs), @(d:16,ERd.L)	B	4	7	7						S	D			@(d:16+ERd)	∨	@(1/2/3+ERs)	→	@(d:16+ERd)				—	—	↑	↓	0	—	



Logic operations (51)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode							Operation ⁶					Condition Codes ²																		
				Min.	16-Bit Instruction Fetch #xxx	Fetch	Rn	@(d,ERn)	@(d,Rn,WRn,LRn)	@(Rn)/@ERn+/@ERn+/@ERn+/@ERn+/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C													
OR	OR.B @aa:32,@ERd	B	4	4	7				D						@ERd	∨	@aa:32	→	@ERd						—	—	↑	↓	0	—				
	OR.B @aa:32,@ERd+	B	4	4	7							D	S		@ERd	∨	@aa:32	→	@ERd				ERd32+1→ERd32	—	—	↑	↓	0	—					
	OR.B @aa:32,@ERd-	B	4	4	7							D	S		@ERd	∨	@aa:32	→	@ERd				ERd32-1→ERd32	—	—	↑	↓	0	—					
	OR.B @aa:32,@+ERd	B	4	5	8									D	S				ERd32+1→ERd32	@ERd	∨	@aa:32	→	@ERd				—	—	↑	↓	0	—	
	OR.B @aa:32,@-ERd	B	4	5	8									D	S				ERd32-1→ERd32	@ERd	∨	@aa:32	→	@ERd				—	—	↑	↓	0	—	
	OR.B @aa:32,@(d:16,ERd)	B	4	5	8									D	S					@(1/2/3+ERd)	∨	@aa:32	→	@(1/2/3+ERd)				—	—	↑	↓	0	—	
	OR.B @aa:32,@(d:16,ERd)	B	5	5	8									D	S					@(d:16+ERd)	∨	@aa:32	→	@(d:16+ERd)				—	—	↑	↓	0	—	
	OR.B @aa:32,@(d:32,ERd)	B	6	5	9									D	S					@(d:32+ERd)	∨	@aa:32	→	@(d:32+ERd)				—	—	↑	↓	0	—	
	OR.B @aa:32,@(d:16,Rd,B)	B	5	5	8									D	S					@(d:16+RdL)	∨	@aa:32	→	@(d:16+RdL)				—	—	↑	↓	0	—	
	OR.B @aa:32,@(d:16,Rd,W)	B	5	5	8									D	S					@(d:16+Rd)	∨	@aa:32	→	@(d:16+Rd)				—	—	↑	↓	0	—	
	OR.B @aa:32,@(d:16,ERd,L)	B	5	5	8									D	S					@(d:16+ERd)	∨	@aa:32	→	@(d:16+ERd)				—	—	↑	↓	0	—	
	OR.B @aa:32,@(d:32,Rd,B)	B	6	5	9									D	S					@(d:32+RdL)	∨	@aa:32	→	@(d:32+RdL)				—	—	↑	↓	0	—	
	OR.B @aa:32,@(d:32,Rd,W)	B	6	5	9									D	S					@(d:32+Rd)	∨	@aa:32	→	@(d:32+Rd)				—	—	↑	↓	0	—	
	OR.B @aa:32,@(d:32,ERd,L)	B	6	5	9									D	S					@(d:32+ERd)	∨	@aa:32	→	@(d:32+ERd)				—	—	↑	↓	0	—	
	OR.B @aa:32,@aa:16	B	5	4	8									S						@aa:16	∨	@aa:32	→	@aa:16				—	—	↑	↓	0	—	
	OR.B @aa:32,@aa:32	B	6	4	9									S						@aa:32	∨	@aa:32	→	@aa:32				—	—	↑	↓	0	—	
	OR.W #xx:16,Rd	W	2	1	2					D					S					Rd16	∨	#xx	→	Rd16				—	—	↑	↓	0	—	
	OR.W #xx:16,@ERd	W	3	4	5						D				S					@ERd	∨	#xx	→	@ERd				—	—	↑	↓	0	—	
	OR.W #xx:16,@ERd+	W	3	4	5										D	S				@ERd	∨	#xx	→	@ERd				ERd32+2→ERd32	—	—	↑	↓	0	—
	OR.W #xx:16,@ERd-	W	3	4	5										D	S				@ERd	∨	#xx	→	@ERd				ERd32-2→ERd32	—	—	↑	↓	0	—
	OR.W #xx:16,@+ERd	W	3	5	5										D	S				ERd32+2→ERd32	@ERd	∨	#xx	→	@ERd				—	—	↑	↓	0	—
	OR.W #xx:16,@-ERd	W	3	5	5										D	S				ERd32-2→ERd32	@ERd	∨	#xx	→	@ERd				—	—	↑	↓	0	—
OR.W #xx:16,@(d:2,ERd)	W	3	5	5										D	S				@(2/4/6+ERd)	∨	#xx	→	@(2/4/6+ERd)				—	—	↑	↓	0	—		
OR.W #xx:16,@(d:16,ERd)	W	4	5	6										D	S				@(d:16+ERd)	∨	#xx	→	@(d:16+ERd)				—	—	↑	↓	0	—		
OR.W #xx:16,@(d:32,ERd)	W	5	5	7										D	S				@(d:32+ERd)	∨	#xx	→	@(d:32+ERd)				—	—	↑	↓	0	—		

Logic operations (52)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode							Operation ⁵					Condition Codes ^{6,7}											
				Min.	16-Bit Instruction Fetch Execution Status ¹	Number of Execution Status ²	#xx	Rn	@ERn	@(d,ERn)	@(d,Rn,ERn,WERn,L)	@ERn/@ERn+/@ERn+/@ERn	@aa:0/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C			
OR	OR.W #xx:16,@(d:16,Rd,B)	W	4	5	6					D	S				@(d:16+RdL<<1) ∨ #xx → @(d:16+RdL<<1)						—	—	↑	↓	0	—	
	OR.W #xx:16,@(d:16,Rd,W)	W	4	5	6					D	S				@(d:16+Rd<<1) ∨ #xx → @(d:16+Rd<<1)						—	—	↑	↓	0	—	
	OR.W #xx:16,@(d:16,ERd,L)	W	4	5	6					D	S				@(d:16+ERd<<1) ∨ #xx → @(d:16+ERd<<1)						—	—	↑	↓	0	—	
	OR.W #xx:16,@(d:32,Rd,B)	W	5	5	7					D	S				@(d:32+RdL<<1) ∨ #xx → @(d:32+RdL<<1)						—	—	↑	↓	0	—	
	OR.W #xx:16,@(d:32,Rd,W)	W	5	5	7					D	S				@(d:32+Rd<<1) ∨ #xx → @(d:32+Rd<<1)						—	—	↑	↓	0	—	
	OR.W #xx:16,@(d:32,ERd,L)	W	5	5	7					D	S				@(d:32+ERd<<1) ∨ #xx → @(d:32+ERd<<1)						—	—	↑	↓	0	—	
	OR.W #xx:16,@aa:16	W	4	4	6							S				@aa:16 ∨ #xx → @aa:16						—	—	↑	↓	0	—
	OR.W #xx:16,@aa:32	W	5	4	7							S				@aa:32 ∨ #xx → @aa:32						—	—	↑	↓	0	—
	OR.W Rs,Rd	W	1	1	1			S								Rd16 ∨ Rs16 → Rd16						—	—	↑	↓	0	—
	OR.W Rs,@ERd	W	2	3	4			S	D							@ERd ∨ Rs16 → @ERd						—	—	↑	↓	0	—
	OR.W Rs,@ERd+	W	2	4	4			S			D					@ERd ∨ Rs16 → @ERd					ERd32+2→ERd32	—	—	↑	↓	0	—
	OR.W Rs,@ERd-	W	2	4	4			S			D					@ERd ∨ Rs16 → @ERd					ERd32-2→ERd32	—	—	↑	↓	0	—
	OR.W Rs,@+ERd	W	2	5	5			S			D				ERd32+2→ERd32	@ERd ∨ Rs16 → @ERd						—	—	↑	↓	0	—
	OR.W Rs,@-ERd	W	2	5	5			S			D				ERd32-2→ERd32	@ERd ∨ Rs16 → @ERd						—	—	↑	↓	0	—
	OR.W Rs,@(d:2,ERd)	W	2	5	5			S	D							@(2/4/6+ERd) ∨ Rs16 → @(2/4/6+ERd)						—	—	↑	↓	0	—
	OR.W Rs,@(d:16,ERd)	W	3	5	5			S	D							@(d:16+ERd) ∨ Rs16 → @(d:16+ERd)						—	—	↑	↓	0	—
	OR.W Rs,@(d:32,ERd)	W	4	5	6			S	D							@(d:32+ERd) ∨ Rs16 → @(d:32+ERd)						—	—	↑	↓	0	—
	OR.W Rs,@(d:16,Rd,B)	W	3	5	5			S			D					@(d:16+RdL<<1) ∨ Rs16 → @(d:16+RdL<<1)						—	—	↑	↓	0	—
	OR.W Rs,@(d:16,Rd,W)	W	3	5	5			S			D					@(d:16+Rd<<1) ∨ Rs16 → @(d:16+Rd<<1)						—	—	↑	↓	0	—
	OR.W Rs,@(d:16,ERd,L)	W	3	5	5			S			D					@(d:16+ERd<<1) ∨ Rs16 → @(d:16+ERd<<1)						—	—	↑	↓	0	—
	OR.W Rs,@(d:32,Rd,B)	W	4	5	6			S			D					@(d:32+RdL<<1) ∨ Rs16 → @(d:32+RdL<<1)						—	—	↑	↓	0	—
OR.W Rs,@(d:32,Rd,W)	W	4	5	6			S			D					@(d:32+Rd<<1) ∨ Rs16 → @(d:32+Rd<<1)						—	—	↑	↓	0	—	
OR.W Rs,@(d:32,ERd,L)	W	4	5	6			S			D					@(d:32+ERd<<1) ∨ Rs16 → @(d:32+ERd<<1)						—	—	↑	↓	0	—	
OR.W Rs,@aa:16	W	3	3	5			S				D				@aa:16 ∨ Rs16 → @aa:16						—	—	↑	↓	0	—	
OR.W Rs,@aa:32	W	4	3	6			S				D				@aa:32 ∨ Rs16 → @aa:32						—	—	↑	↓	0	—	



Logic operations (55)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode											Operation ⁵					Condition Codes ^{5,2}																																						
				Min.	16-Bit Instruction Fetch	Execution Status ¹	Execution Status ¹	Execution Status ¹	Execution Status ¹	Execution Status ¹	Execution Status ¹	Execution Status ¹	Execution Status ¹	Execution Status ¹											Execution Status ¹																																	
				Prx	Rn	@ERn	@(d,ERn)	@(d,Rn,LRn,WERn,L)	@(d,ERn)/@ERn+/@ERn+/@+ERn	@aa:9/@aa:16/@aa:32																																																
n																																																										
OR	OR.W @ERs-, @ERd+	W	3	5	6																																																					
	OR.W @ERs-, @ERd-	W	3	5	6																																																					
	OR.W @ERs-, @+ERd	W	3	6	7																																																					
	OR.W @ERs-, @-ERd	W	3	6	7																																																					
	OR.W @ERs-, @(d:2,ERd)	W	3	6	7																																																					
	OR.W @ERs-, @(d:16,ERd)	W	4	6	7																																																					
	OR.W @ERs-, @(d:32,ERd)	W	5	6	8																																																					
	OR.W @ERs-, @(d:16,Rd,B)	W	4	6	7																																																					
	OR.W @ERs-, @(d:16,Rd,W)	W	4	6	7																																																					
	OR.W @ERs-, @(d:16,ERd,L)	W	4	6	7																																																					
	OR.W @ERs-, @(d:32,Rd,B)	W	5	6	8																																																					
	OR.W @ERs-, @(d:32,Rd,W)	W	5	6	8																																																					
	OR.W @ERs-, @(d:32,ERd,L)	W	5	6	8																																																					
	OR.W @ERs-, @aa:16	W	4	5	7																																																					
	OR.W @ERs-, @aa:32	W	5	5	8																																																					
	OR.W @+ERs, @ERd	W	3	6	6																																																					
	OR.W @+ERs, @ERd+	W	3	6	6																																																					
	OR.W @+ERs, @ERd-	W	3	6	6																																																					
	OR.W @+ERs, @+ERd	W	3	7	7																																																					
	OR.W @+ERs, @-ERd	W	3	7	7																																																					
OR.W @+ERs, @(d:2,ERd)	W	3	7	7																																																						
OR.W @+ERs, @(d:16,ERd)	W	4	7	7																																																						
OR.W @+ERs, @(d:32,ERd)	W	5	7	8																																																						
OR.W @+ERs, @(d:16,Rd,B)	W	4	7	7																																																						
OR.W @+ERs, @(d:16,Rd,W)	W	4	7	7																																																						



Logic operations (56)

Instruction n	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Status ¹ #xx	Addressing Mode						Operation ⁵					Condition Codes ²						
						Rn	@ERn	@ (d, ERn)	@ (d, Rn, ERn, WERn, L)	@ ERn / @ ERn+ / @ ERn+ / @ ERn	@ aa:9 / @ aa:16 / @ aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C	
																							Rn
OR	OR.W @+ERs, @ (d:16, ERd.L)	W	4	7	7					D	S		ERs32+2→ERs32		@ (d:16+ERd<<1) ∨ @ERs → @ (d:16+ERd<<1)				↑	↓	0	—	
	OR.W @+ERs, @ (d:32, Rd.B)	W	5	7	8					D	S		ERs32+2→ERs32		@ (d:32+RdL<<1) ∨ @ERs → @ (d:32+RdL<<1)				↑	↓	0	—	
	OR.W @+ERs, @ (d:32, Rd.W)	W	5	7	8					D	S		ERs32+2→ERs32		@ (d:32+Rd<<1) ∨ @ERs → @ (d:32+Rd<<1)				↑	↓	0	—	
	OR.W @+ERs, @ (d:32, ERd.L)	W	5	7	8					D	S		ERs32+2→ERs32		@ (d:32+ERd<<1) ∨ @ERs → @ (d:32+ERd<<1)				↑	↓	0	—	
	OR.W @+ERs, @aa:16	W	4	6	7						S	D		ERs32+2→ERs32		@aa:16 ∨ @ERs → @aa:16				↑	↓	0	—
	OR.W @+ERs, @aa:32	W	5	6	8						S	D		ERs32+2→ERs32		@aa:32 ∨ @ERs → @aa:32				↑	↓	0	—
	OR.W @-ERs, @ERd	W	3	6	6						D	S		ERs32-2→ERs32		@ERd ∨ @ERs → @ERd				↑	↓	0	—
	OR.W @-ERs, @ERd+	W	3	6	6						S	S		ERs32-2→ERs32		@ERd ∨ @ERs → @ERd				↑	↓	0	—
	OR.W @-ERs, @ERd-	W	3	6	6						S	S		ERs32-2→ERs32		@ERd ∨ @ERs → @ERd				↑	↓	0	—
	OR.W @-ERs, @+ERd	W	3	7	7						S	S		ERs32-2→ERs32	ERd32+2→ERd32	@ERd ∨ @ERs → @ERd				↑	↓	0	—
	OR.W @-ERs, @-ERd	W	3	7	7						S	S		ERs32-2→ERs32	ERd32-2→ERd32	@ERd ∨ @ERs → @ERd				↑	↓	0	—
	OR.W @-ERs, @ (d:2, ERd)	W	3	7	7						D	S		ERs32-2→ERs32		@ (2/4/6+ERd) ∨ @ERs → @ (2/4/6+ERd)				↑	↓	0	—
	OR.W @-ERs, @ (d:16, ERd)	W	4	7	7						D	S		ERs32-2→ERs32		@ (d:16+ERd) ∨ @ERs → @ (d:16+ERd)				↑	↓	0	—
	OR.W @-ERs, @ (d:32, ERd)	W	5	7	8						D	S		ERs32-2→ERs32		@ (d:32+ERd) ∨ @ERs → @ (d:32+ERd)				↑	↓	0	—
	OR.W @-ERs, @ (d:16, Rd.B)	W	4	7	7						D	S		ERs32-2→ERs32		@ (d:16+RdL<<1) ∨ @ERs → @ (d:16+RdL<<1)				↑	↓	0	—
	OR.W @-ERs, @ (d:16, Rd.W)	W	4	7	7						D	S		ERs32-2→ERs32		@ (d:16+Rd<<1) ∨ @ERs → @ (d:16+Rd<<1)				↑	↓	0	—
	OR.W @-ERs, @ (d:16, ERd.L)	W	4	7	7						D	S		ERs32-2→ERs32		@ (d:16+ERd<<1) ∨ @ERs → @ (d:16+ERd<<1)				↑	↓	0	—
	OR.W @-ERs, @ (d:32, Rd.B)	W	5	7	8						D	S		ERs32-2→ERs32		@ (d:32+RdL<<1) ∨ @ERs → @ (d:32+RdL<<1)				↑	↓	0	—
	OR.W @-ERs, @ (d:32, Rd.W)	W	5	7	8						D	S		ERs32-2→ERs32		@ (d:32+Rd<<1) ∨ @ERs → @ (d:32+Rd<<1)				↑	↓	0	—
	OR.W @-ERs, @ (d:32, ERd.L)	W	5	7	8						D	S		ERs32-2→ERs32		@ (d:32+ERd<<1) ∨ @ERs → @ (d:32+ERd<<1)				↑	↓	0	—
	OR.W @-ERs, @aa:16	W	4	6	7						S	D		ERs32-2→ERs32		@aa:16 ∨ @ERs → @aa:16				↑	↓	0	—
	OR.W @-ERs, @aa:32	W	5	6	8						S	D		ERs32-2→ERs32		@aa:32 ∨ @ERs → @aa:32				↑	↓	0	—
	OR.W @ (d:2, ERs), @ERd	W	3	6	6						D	S				@ERd ∨ @ (2/4/6+ERs) → @ERd				↑	↓	0	—
	OR.W @ (d:2, ERs), @ERd+	W	3	6	6						S	D				@ERd ∨ @ (2/4/6+ERs) → @ERd				↑	↓	0	—
OR.W @ (d:2, ERs), @ERd-	W	3	6	6						S	D				@ERd ∨ @ (2/4/6+ERs) → @ERd				↑	↓	0	—	



Logic operations (57)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode										Operation ⁵					Condition Codes ⁶																									
				Min.	16-Bit Instruction Fetch Execution Stages ¹	OpX	Rn	@ERn	@(d,ERn)	@(d,Rn,ERn,WERn,L)	@ERn/@ERn+/@ERn+/@ERn+/@ERn	@aa:9/@aa:16/@aa:32																																
				Number of	16-Bit Instruction Fetch Execution Stages ¹	OpX	Rn	@ERn	@(d,ERn)	@(d,Rn,ERn,WERn,L)	@ERn/@ERn+/@ERn+/@ERn+/@ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C																					
OR	OR.W @(d:2,ERs),@+ERd	W	3	7	7						S		D				ERd32+2→ERd32	@ERd	∨	@(2/4/6+ERs)	→	@ERd																						
	OR.W @(d:2,ERs),@-ERd	W	3	7	7						S		D				ERd32-2→ERd32	@ERd	∨	@(2/4/6+ERs)	→	@ERd																						
	OR.W @(d:2,ERs),@(d:2,ERd)	W	3	7	7													@(2/4/6+ERd)	∨	@(2/4/6+ERs)	→	@(2/4/6+ERd)																						
	OR.W @(d:2,ERs),@(d:16,ERd)	W	4	7	7							S						@(d:16+ERd)	∨	@(2/4/6+ERs)	→	@(d:16+ERd)																						
	OR.W @(d:2,ERs),@(d:32,ERd)	W	5	7	8							S							@(d:32+ERd)	∨	@(2/4/6+ERs)	→	@(d:32+ERd)																					
	OR.W @(d:2,ERs),@(d:16,Rd.B)	W	4	7	7							S	D						@(d:16+RdL<<1)	∨	@(2/4/6+ERs)	→	@(d:16+RdL<<1)																					
	OR.W @(d:2,ERs),@(d:16,Rd.W)	W	4	7	7							S	D						@(d:16+Rd<<1)	∨	@(2/4/6+ERs)	→	@(d:16+Rd<<1)																					
	OR.W @(d:2,ERs),@(d:16,ERd.L)	W	4	7	7							S	D						@(d:16+ERd<<1)	∨	@(2/4/6+ERs)	→	@(d:16+ERd<<1)																					
	OR.W @(d:2,ERs),@(d:32,Rd.B)	W	5	7	8							S	D						@(d:32+RdL<<1)	∨	@(2/4/6+ERs)	→	@(d:32+RdL<<1)																					
	OR.W @(d:2,ERs),@(d:32,Rd.W)	W	5	7	8							S	D						@(d:32+Rd<<1)	∨	@(2/4/6+ERs)	→	@(d:32+Rd<<1)																					
	OR.W @(d:2,ERs),@(d:32,ERd.L)	W	5	7	8							S	D						@(d:32+ERd<<1)	∨	@(2/4/6+ERs)	→	@(d:32+ERd<<1)																					
	OR.W @(d:2,ERs),@aa:16	W	4	6	7							S			D				@aa:16	∨	@(2/4/6+ERs)	→	@aa:16																					
	OR.W @(d:2,ERs),@aa:32	W	5	6	8							S			D				@aa:32	∨	@(2/4/6+ERs)	→	@aa:32																					
	OR.W @(d:16,ERs),@ERd	W	4	6	7								D						@ERd	∨	@(d:16+ERs)	→	@ERd																					
	OR.W @(d:16,ERs),@ERd+	W	4	6	7							S		D					@ERd	∨	@(d:16+ERs)	→	@ERd				ERd32+2→ERd32																	
	OR.W @(d:16,ERs),@ERd-	W	4	6	7							S		D					@ERd	∨	@(d:16+ERs)	→	@ERd				ERd32-2→ERd32																	
	OR.W @(d:16,ERs),@+ERd	W	4	7	8							S		D					ERd32+2→ERd32	@ERd	∨	@(d:16+ERs)	→	@ERd																				
	OR.W @(d:16,ERs),@-ERd	W	4	7	8							S		D					ERd32-2→ERd32	@ERd	∨	@(d:16+ERs)	→	@ERd																				
	OR.W @(d:16,ERs),@(d:2,ERd)	W	4	7	8							S							@(2/4/6+ERd)	∨	@(d:16+ERs)	→	@(2/4/6+ERd)																					
	OR.W @(d:16,ERs),@(d:16,ERd)	W	5	7	8							S							@(d:16+ERd)	∨	@(d:16+ERs)	→	@(d:16+ERd)																					
	OR.W @(d:16,ERs),@(d:32,ERd)	W	6	7	9							S							@(d:32+ERd)	∨	@(d:16+ERs)	→	@(d:32+ERd)																					
	OR.W @(d:16,ERs),@(d:16,Rd.B)	W	5	7	8							S	D						@(d:16+RdL<<1)	∨	@(d:16+ERs)	→	@(d:16+RdL<<1)																					
	OR.W @(d:16,ERs),@(d:16,Rd.W)	W	5	7	8							S	D						@(d:16+Rd<<1)	∨	@(d:16+ERs)	→	@(d:16+Rd<<1)																					
OR.W @(d:16,ERs),@(d:16,ERd.L)	W	5	7	8							S	D						@(d:16+ERd<<1)	∨	@(d:16+ERs)	→	@(d:16+ERd<<1)																						
OR.W @(d:16,ERs),@(d:32,Rd.B)	W	6	7	9							S	D						@(d:32+RdL<<1)	∨	@(d:16+ERs)	→	@(d:32+RdL<<1)																						

Logic operations (58)

Instruction	Mnemonic	Size	Instruction Length	Number of		Addressing Mode							Operation ⁵					Condition Codes ^{6,2}											
				Min.	Max.	16-Bit Instruction Fetch Execution Stages ¹							Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C						
				16-Bit Instruction Fetch	Execution Stages ¹	PCx	Rn	@ERN	@(d,ERn)	@(d,Rn.L,ERn.W,ERn.L)	@ERN/@ERN+/@ERN+/@ERN+/@ERN	@aa:0/@aa:16/@aa:32												Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I
OR	OR.W @(d:16,ERs),@(d:32,Rd,W)	W	6 7 9							S	D					@(d:32+Rd<<1)	∨	@(d:16+ERs)	→	@(d:32+Rd<<1)				—	—	↑	↓	0	—
	OR.W @(d:16,ERs),@(d:32,ERd,L)	W	6 7 9							S	D					@(d:32+ERd<<1)	∨	@(d:16+ERs)	→	@(d:32+ERd<<1)				—	—	↑	↓	0	—
	OR.W @(d:16,ERs),@aa:16	W	5 6 8							S		D				@aa:16	∨	@(d:16+ERs)	→	@aa:16				—	—	↑	↓	0	—
	OR.W @(d:16,ERs),@aa:32	W	6 6 9							S		D				@aa:32	∨	@(d:16+ERs)	→	@aa:32				—	—	↑	↓	0	—
	OR.W @(d:32,ERs),@ERd	W	5 6 8							D	S					@ERd	∨	@(d:32+ERs)	→	@ERd				—	—	↑	↓	0	—
	OR.W @(d:32,ERs),@ERd+	W	5 6 8							S	D					@ERd	∨	@(d:32+ERs)	→	@ERd			ERd32+2→ERd32	—	—	↑	↓	0	—
	OR.W @(d:32,ERs),@ERd-	W	5 6 8							S	D					@ERd	∨	@(d:32+ERs)	→	@ERd			ERd32-2→ERd32	—	—	↑	↓	0	—
	OR.W @(d:32,ERs),@+ERd	W	5 7 9							S	D				ERd32+2→ERd32	@ERd	∨	@(d:32+ERs)	→	@ERd				—	—	↑	↓	0	—
	OR.W @(d:32,ERs),@-ERd	W	5 7 9							S	D				ERd32-2→ERd32	@ERd	∨	@(d:32+ERs)	→	@ERd				—	—	↑	↓	0	—
	OR.W @(d:32,ERs),@(d:2,ERd)	W	5 7 9							S						@(2/4/6+ERd)	∨	@(d:32+ERs)	→	@(2/4/6+ERd)				—	—	↑	↓	0	—
	OR.W @(d:32,ERs),@(d:16,ERd)	W	6 7 9							S						@(d:16+ERd)	∨	@(d:32+ERs)	→	@(d:16+ERd)				—	—	↑	↓	0	—
	OR.W @(d:32,ERs),@(d:32,ERd)	W	7 7 1							S						@(d:32+ERd)	∨	@(d:32+ERs)	→	@(d:32+ERd)				—	—	↑	↓	0	—
	OR.W @(d:32,ERs),@(d:16,Rd,B)	W	6 7 9							S	D					@(d:16+RdL<<1)	∨	@(d:32+ERs)	→	@(d:16+RdL<<1)				—	—	↑	↓	0	—
	OR.W @(d:32,ERs),@(d:16,Rd,W)	W	6 7 9							S	D					@(d:16+Rd<<1)	∨	@(d:32+ERs)	→	@(d:16+Rd<<1)				—	—	↑	↓	0	—
	OR.W @(d:32,ERs),@(d:16,ERd,L)	W	6 7 9							S	D					@(d:16+ERd<<1)	∨	@(d:32+ERs)	→	@(d:16+ERd<<1)				—	—	↑	↓	0	—
	OR.W @(d:32,ERs),@(d:32,Rd,B)	W	7 7 1							S	D					@(d:32+RdL<<1)	∨	@(d:32+ERs)	→	@(d:32+RdL<<1)				—	—	↑	↓	0	—
	OR.W @(d:32,ERs),@(d:32,Rd,W)	W	7 7 1							S	D					@(d:32+Rd<<1)	∨	@(d:32+ERs)	→	@(d:32+Rd<<1)				—	—	↑	↓	0	—
	OR.W @(d:32,ERs),@(d:32,ERd,L)	W	7 7 1							S	D					@(d:32+ERd<<1)	∨	@(d:32+ERs)	→	@(d:32+ERd<<1)				—	—	↑	↓	0	—
	OR.W @(d:32,ERs),@aa:16	W	6 6 9							S		D				@aa:16	∨	@(d:32+ERs)	→	@aa:16				—	—	↑	↓	0	—
	OR.W @(d:32,ERs),@aa:32	W	7 6 1							S		D				@aa:32	∨	@(d:32+ERs)	→	@aa:32				—	—	↑	↓	0	—
	OR.W @(d:16,Rs,B),@ERd	W	4 6 7							D	S					@ERd	∨	@(d:16+RsL<<1)	→	@ERd				—	—	↑	↓	0	—
	OR.W @(d:16,Rs,B),@ERd+	W	4 6 7							S	D					@ERd	∨	@(d:16+RsL<<1)	→	@ERd			ERd32+2→ERd32	—	—	↑	↓	0	—
	OR.W @(d:16,Rs,B),@ERd-	W	4 6 7							S	D					@ERd	∨	@(d:16+RsL<<1)	→	@ERd			ERd32-2→ERd32	—	—	↑	↓	0	—
	OR.W @(d:16,Rs,B),@+ERd	W	4 7 8							S	D				ERd32+2→ERd32	@ERd	∨	@(d:16+RsL<<1)	→	@ERd				—	—	↑	↓	0	—
	OR.W @(d:16,Rs,B),@-ERd	W	4 7 8							S	D				ERd32-2→ERd32	@ERd	∨	@(d:16+RsL<<1)	→	@ERd				—	—	↑	↓	0	—

Logic operations (59)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹	Addressing Mode								Operation ⁵					Condition Code ^{6,2}																			
						Number of		Rn	@ERN	@ERN	@RN.L/Rn.W/ERN.L	@ERN/@ERN+/@ERN+/@ERN	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C														
						Fetch	Execute																		Stages													
OR	OR.W @(d:16, Rs.B), @(d:2, ERd)	W	4	7	8					D	S																											
	OR.W @(d:16, Rs.B), @(d:16, ERd)	W	5	7	8					D	S																											
	OR.W @(d:16, Rs.B), @(d:32, ERd)	W	6	7	9					D	S																											
	OR.W @(d:16, Rs.B), @(d:16, Rd.B)	W	5	7	8						S																											
	OR.W @(d:16, Rs.B), @(d:16, Rd.W)	W	5	7	8						S																											
	OR.W @(d:16, Rs.B), @(d:16, ERd.L)	W	5	7	8						S																											
	OR.W @(d:16, Rs.B), @(d:32, Rd.B)	W	6	7	9						S																											
	OR.W @(d:16, Rs.B), @(d:32, Rd.W)	W	6	7	9						S																											
	OR.W @(d:16, Rs.B), @(d:32, ERd.L)	W	6	7	9						S																											
	OR.W @(d:16, Rs.B), @aa:16	W	5	6	8						S		D																									
	OR.W @(d:16, Rs.B), @aa:32	W	6	6	9						S		D																									
	OR.W @(d:16, Rs.W), @ERd	W	4	6	7					D	S																											
	OR.W @(d:16, Rs.W), @ERd+	W	4	6	7						S		D																									
	OR.W @(d:16, Rs.W), @ERd-	W	4	6	7						S		D																									
	OR.W @(d:16, Rs.W), @+ERd	W	4	7	8						S		D																									
	OR.W @(d:16, Rs.W), @-ERd	W	4	7	8						S		D																									
	OR.W @(d:16, Rs.W), @(d:2, ERd)	W	4	7	8						D	S																										
	OR.W @(d:16, Rs.W), @(d:16, ERd)	W	5	7	8						D	S																										
	OR.W @(d:16, Rs.W), @(d:32, ERd)	W	6	7	9						D	S																										
	OR.W @(d:16, Rs.W), @(d:16, Rd.B)	W	5	7	8						S																											
	OR.W @(d:16, Rs.W), @(d:16, Rd.W)	W	5	7	8						S																											
	OR.W @(d:16, Rs.W), @(d:16, ERd.L)	W	5	7	8						S																											
	OR.W @(d:16, Rs.W), @(d:32, Rd.B)	W	6	7	9						S																											
	OR.W @(d:16, Rs.W), @(d:32, Rd.W)	W	6	7	9						S																											
	OR.W @(d:16, Rs.W), @(d:32, ERd.L)	W	6	7	9						S																											

Logic operations (60)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode								Operation ⁶					Condition Codes ^{5,2}															
				Min.	16-Bit Instruction Fetch Execution Stages ¹	Number of Execution Stages ¹							Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C									
				16-Bit Instruction Fetch Execution Stages ¹	16-Bit Instruction Fetch Execution Stages ¹	16-Bit Instruction Fetch Execution Stages ¹	16-Bit Instruction Fetch Execution Stages ¹	16-Bit Instruction Fetch Execution Stages ¹	16-Bit Instruction Fetch Execution Stages ¹	16-Bit Instruction Fetch Execution Stages ¹	16-Bit Instruction Fetch Execution Stages ¹																					
OR	OR.W @(d:16,RS.W),@aa:16	W	5 6 8							S	D				@aa:16	∨	@(d:16+RS<<1)	→	@aa:16													
	OR.W @(d:16,RS.W),@aa:32	W	6 6 9							S	D				@aa:32	∨	@(d:16+RS<<1)	→	@aa:32													
	OR.W @(d:16,ERS.L),@ERd	W	4 6 7							D	S				@ERd	∨	@(d:16+ERS<<1)	→	@ERd													
	OR.W @(d:16,ERS.L),@ERd+	W	4 6 7								S	D				@ERd	∨	@(d:16+ERS<<1)	→	@ERd			ERd32+2→ERd32									
	OR.W @(d:16,ERS.L),@ERd-	W	4 6 7								S	D				@ERd	∨	@(d:16+ERS<<1)	→	@ERd			ERd32-2→ERd32									
	OR.W @(d:16,ERS.L),@+ERd	W	4 7 8									S	D			ERd32+2→ERd32	@ERd	∨	@(d:16+ERS<<1)	→	@ERd											
	OR.W @(d:16,ERS.L),@-ERd	W	4 7 8									S	D			ERd32-2→ERd32	@ERd	∨	@(d:16+ERS<<1)	→	@ERd											
	OR.W @(d:16,ERS.L),@(d:2,ERd)	W	4 7 8								D	S				@(2/4+ERd)	∨	@(d:16+ERS<<1)	→	@(2/4+ERd)												
	OR.W @(d:16,ERS.L),@(d:16,ERd)	W	5 7 8								D	S				@(d:16+ERd)	∨	@(d:16+ERS<<1)	→	@(d:16+ERd)												
	OR.W @(d:16,ERS.L),@(d:32,ERd)	W	6 7 9								D	S				@(d:32+ERd)	∨	@(d:16+ERS<<1)	→	@(d:32+ERd)												
	OR.W @(d:16,ERS.L),@(d:16,Rd,B)	W	5 7 8									S				@(d:16+RdL<<1)	∨	@(d:16+ERS<<1)	→	@(d:16+RdL<<1)												
	OR.W @(d:16,ERS.L),@(d:16,Rd,W)	W	5 7 8									S				@(d:16+Rd<<1)	∨	@(d:16+ERS<<1)	→	@(d:16+Rd<<1)												
	OR.W @(d:16,ERS.L),@(d:16,ERdL)	W	5 7 8									S				@(d:16+ERdL<<1)	∨	@(d:16+ERS<<1)	→	@(d:16+ERdL<<1)												
	OR.W @(d:16,ERS.L),@(d:32,Rd,B)	W	6 7 9									S				@(d:32+RdL<<1)	∨	@(d:16+ERS<<1)	→	@(d:32+RdL<<1)												
	OR.W @(d:16,ERS.L),@(d:32,Rd,W)	W	6 7 9									S				@(d:32+Rd<<1)	∨	@(d:16+ERS<<1)	→	@(d:32+Rd<<1)												
	OR.W @(d:16,ERS.L),@(d:32,ERdL)	W	6 7 9									S				@(d:32+ERdL<<1)	∨	@(d:16+ERS<<1)	→	@(d:32+ERdL<<1)												
	OR.W @(d:16,ERS.L),@aa:16	W	5 6 8									S	D			@aa:16	∨	@(d:16+ERS<<1)	→	@aa:16												
	OR.W @(d:16,ERS.L),@aa:32	W	6 6 9									S	D			@aa:32	∨	@(d:16+ERS<<1)	→	@aa:32												
	OR.W @(d:32,RS.B),@ERd	W	5 6 8								D	S				@ERd	∨	@(d:32+RSL<<1)	→	@ERd												
	OR.W @(d:32,RS.B),@ERd+	W	5 6 8									S	D			@ERd	∨	@(d:32+RSL<<1)	→	@ERd			ERd32+2→ERd32									
	OR.W @(d:32,RS.B),@ERd-	W	5 6 8									S	D			@ERd	∨	@(d:32+RSL<<1)	→	@ERd			ERd32-2→ERd32									
	OR.W @(d:32,RS.B),@+ERd	W	5 7 9									S	D			ERd32+2→ERd32	@ERd	∨	@(d:32+RSL<<1)	→	@ERd											
	OR.W @(d:32,RS.B),@-ERd	W	5 7 9									S	D			ERd32-2→ERd32	@ERd	∨	@(d:32+RSL<<1)	→	@ERd											
	OR.W @(d:32,RS.B),@(d:2,ERd)	W	5 7 9									D	S			@(2/4+ERd)	∨	@(d:32+RSL<<1)	→	@(2/4+ERd)												

Logic operations (61)

Instruction	Mnemonic	Size	Instruction Length					Addressing Mode						Operation ⁶					Condition Codes ^{6,2}																			
			Min.	Max.	16-Bit Instruction Fetch Execution Stage ¹ #rx	Rn	@ERn	@(d,ERn)	@(d,Rn),@ERn,W@ERn,L	@ERn/@ERn+/@ERn+/@+ERn	@aa:R/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C																
OR	OR.W @(d:32,Rs.B),@(d:16,ERd)	W	6	7	9				D	S																												
	OR.W @(d:32,Rs.B),@(d:32,ERd)	W	7	7	1				D	S																												
	OR.W @(d:32,Rs.B),@(d:16,Rd.B)	W	6	7	9					S																												
	OR.W @(d:32,Rs.B),@(d:16,Rd.W)	W	6	7	9					S																												
	OR.W @(d:32,Rs.B),@(d:16,ERd.L)	W	6	7	9					S																												
	OR.W @(d:32,Rs.B),@(d:32,Rd.B)	W	7	7	1					S																												
	OR.W @(d:32,Rs.B),@(d:32,Rd.W)	W	7	7	1					S																												
	OR.W @(d:32,Rs.B),@(d:32,ERd.L)	W	7	7	1					S																												
	OR.W @(d:32,Rs.B),@aa:16	W	6	6	9					S	D																											
	OR.W @(d:32,Rs.B),@aa:32	W	7	6	1					S	D																											
	OR.W @(d:32,Rs.W),@ERd	W	5	6	8				D	S																												
	OR.W @(d:32,Rs.W),@ERd+	W	5	6	8					S	D																											
	OR.W @(d:32,Rs.W),@ERd-	W	5	6	8					S	D																											
	OR.W @(d:32,Rs.W),@+ERd	W	5	7	9					S	D																											
	OR.W @(d:32,Rs.W),@-ERd	W	5	7	9					S	D																											
	OR.W @(d:32,Rs.W),@(d:2,ERd)	W	5	7	9				D	S																												
	OR.W @(d:32,Rs.W),@(d:16,ERd)	W	6	7	9					D	S																											
	OR.W @(d:32,Rs.W),@(d:32,ERd)	W	7	7	1					D	S																											
	OR.W @(d:32,Rs.W),@(d:16,Rd.B)	W	6	7	9						S																											
	OR.W @(d:32,Rs.W),@(d:16,Rd.W)	W	6	7	9						S																											
	OR.W @(d:32,Rs.W),@(d:16,ERd.L)	W	6	7	9						S																											
	OR.W @(d:32,Rs.W),@(d:32,Rd.B)	W	7	7	1						S																											
	OR.W @(d:32,Rs.W),@(d:32,Rd.W)	W	7	7	1						S																											
	OR.W @(d:32,Rs.W),@(d:32,ERd.L)	W	7	7	1						S																											
	OR.W @(d:32,Rs.W),@aa:16	W	6	6	9						S	D																										

Logic operations (62)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹ Fxx	Addressing Mode								Operation ⁵					Condition Codes ^{6,2}												
						Rn	@ERN	@ERN	@ERN	@RN.L,WRn.L	@ERN/@ERN+/@ERN+/@ERN	@aa:R/@aa:R/@aa:32	D	S	D	S	D	S	D	S	D	S	D	I	H	N	Z	V	C		
																														Number of Instruction Fetch Execution Stages ¹	Operation 1
OR	OR.W @(d:32,RS,W),@aa:32	W	7	6	1									S		D			@aa:32	∨	@(d:32+RS<<1)	→	@aa:32			—	—	↑	↓	0	—
	OR.W @(d:32,ERS,L),@ERd	W	5	6	8				D		S			S				@ERd	∨	@(d:16+ERS<<1)	→	@ERd			—	—	↑	↓	0	—	
	OR.W @(d:32,ERS,L),@ERd+	W	5	6	8						S	D		S		D		@ERd	∨	@(d:16+ERS<<1)	→	@ERd	ERd32+2→ERd32		—	—	↑	↓	0	—	
	OR.W @(d:32,ERS,L),@ERd-	W	5	6	8						S	D		S		D		@ERd	∨	@(d:16+ERS<<1)	→	@ERd	ERd32-2→ERd32		—	—	↑	↓	0	—	
	OR.W @(d:32,ERS,L),@+ERd	W	5	7	9						S	D		S		D		ERd32+2→ERd32	@ERd	∨	@(d:16+ERS<<1)	→	@ERd		—	—	↑	↓	0	—	
	OR.W @(d:32,ERS,L),@-ERd	W	5	7	9						S	D		S		D		ERd32-2→ERd32	@ERd	∨	@(d:16+ERS<<1)	→	@ERd		—	—	↑	↓	0	—	
	OR.W @(d:32,ERS,L),@(d:2,ERd)	W	5	7	9					D	S			S				@(2/4/6+ERd)	∨	@(d:16+ERS<<1)	→	@(2/4/6+ERd)		—	—	↑	↓	0	—		
	OR.W @(d:32,ERS,L),@(d:16,ERd)	W	6	7	9					D	S			S				@(d:16+ERd)	∨	@(d:16+ERS<<1)	→	@(d:16+ERd)		—	—	↑	↓	0	—		
	OR.W @(d:32,ERS,L),@(d:32,ERd)	W	7	7	1					D	S			S				@(d:32+ERd)	∨	@(d:16+ERS<<1)	→	@(d:32+ERd)		—	—	↑	↓	0	—		
	OR.W @(d:32,ERS,L),@(d:16,Rd,B)	W	6	7	9						S			S				@(d:16+RdL<<1)	∨	@(d:16+ERS<<1)	→	@(d:16+RdL<<1)		—	—	↑	↓	0	—		
	OR.W @(d:32,ERS,L),@(d:16,Rd,W)	W	6	7	9						S			S				@(d:16+Rd<<1)	∨	@(d:16+ERS<<1)	→	@(d:16+Rd<<1)		—	—	↑	↓	0	—		
	OR.W @(d:32,ERS,L),@(d:16,ERdL)	W	6	7	9						S			S				@(d:16+ERd<<1)	∨	@(d:16+ERS<<1)	→	@(d:16+ERd<<1)		—	—	↑	↓	0	—		
	OR.W @(d:32,ERS,L),@(d:32,Rd,B)	W	7	7	1						S			S				@(d:32+RdL<<1)	∨	@(d:16+ERS<<1)	→	@(d:32+RdL<<1)		—	—	↑	↓	0	—		
	OR.W @(d:32,ERS,L),@(d:32,Rd,W)	W	7	7	1						S			S				@(d:32+Rd<<1)	∨	@(d:16+ERS<<1)	→	@(d:32+Rd<<1)		—	—	↑	↓	0	—		
	OR.W @(d:32,ERS,L),@(d:32,ERdL)	W	7	7	1						S			S				@(d:32+ERd<<1)	∨	@(d:16+ERS<<1)	→	@(d:32+ERd<<1)		—	—	↑	↓	0	—		
	OR.W @(d:32,ERS,L),@aa:16	W	6	6	9						S	D		S		D		@aa:16	∨	@(d:16+ERS<<1)	→	@aa:16		—	—	↑	↓	0	—		
	OR.W @(d:32,ERS,L),@aa:32	W	7	6	1						S	D		S		D		@aa:32	∨	@(d:16+ERS<<1)	→	@aa:32		—	—	↑	↓	0	—		
	OR.W @aa:16,@ERd	W	3	4	6					D				S				@ERd	∨	@aa:16	→	@ERd		—	—	↑	↓	0	—		
	OR.W @aa:16,@ERd+	W	3	4	6						D	S		S				@ERd	∨	@aa:16	→	@ERd	ERd32+2→ERd32	—	—	↑	↓	0	—		
	OR.W @aa:16,@ERd-	W	3	4	6						D	S		S				@ERd	∨	@aa:16	→	@ERd	ERd32-2→ERd32	—	—	↑	↓	0	—		
	OR.W @aa:16,@+ERd	W	3	5	7						D	S		S				ERd32+2→ERd32	@ERd	∨	@aa:16	→	@ERd		—	—	↑	↓	0	—	
	OR.W @aa:16,@-ERd	W	3	5	7						D	S		S				ERd32-2→ERd32	@ERd	∨	@aa:16	→	@ERd		—	—	↑	↓	0	—	
	OR.W @aa:16,@(d:2,ERd)	W	3	5	7					D				S				@(2/4/6+ERd)	∨	@aa:16	→	@(2/4/6+ERd)		—	—	↑	↓	0	—		
	OR.W @aa:16,@(d:16,ERd)	W	4	5	7					D				S				@(d:16+ERd)	∨	@aa:16	→	@(d:16+ERd)		—	—	↑	↓	0	—		
	OR.W @aa:16,@(d:32,ERd)	W	5	5	8					D				S				@(d:32+ERd)	∨	@aa:16	→	@(d:32+ERd)		—	—	↑	↓	0	—		

Logic operations (63)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode										Operation ⁵					Condition Codes ^{6,7}					
				Number of		16-Bit Instruction Fetch Execution Status ⁸	Rn	Rn	@ERn	@(d,ERn)	@Rn.L/Rn.W/ERn.L	@ERn/@ERn+/@ERn+/@+ERn	@aa:0/@aa:16/@aa:32											
				Min.	Max.																			
n				Min.	Max.	16-Bit Instruction Fetch Execution Status ⁸	Rn	Rn	@ERn	@(d,ERn)	@Rn.L/Rn.W/ERn.L	@ERn/@ERn+/@ERn+/@+ERn	@aa:0/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C
OR	OR.W @aa:16,@(d:16,Rd.B)	W	4	5	7						D	S				@(d:16+RdL<<1) ∨ @aa:16 → @(d:16+RdL<<1)			—	—	↑	↓	0	—
	OR.W @aa:16,@(d:16,Rd.W)	W	4	5	7						D	S				@(d:16+Rd<<1) ∨ @aa:16 → @(d:16+Rd<<1)			—	—	↑	↓	0	—
	OR.W @aa:16,@(d:16,ERd.L)	W	4	5	7						D	S				@(d:16+ERd<<1) ∨ @aa:16 → @(d:16+ERd<<1)			—	—	↑	↓	0	—
	OR.W @aa:16,@(d:32,Rd.B)	W	5	5	8						D	S				@(d:32+RdL<<1) ∨ @aa:16 → @(d:32+RdL<<1)			—	—	↑	↓	0	—
	OR.W @aa:16,@(d:32,Rd.W)	W	5	5	8						D	S				@(d:32+Rd<<1) ∨ @aa:16 → @(d:32+Rd<<1)			—	—	↑	↓	0	—
	OR.W @aa:16,@(d:32,ERd.L)	W	5	5	8						D	S				@(d:32+ERd<<1) ∨ @aa:16 → @(d:32+ERd<<1)			—	—	↑	↓	0	—
	OR.W @aa:16,@aa:16	W	4	4	7								S			@aa:16 ∨ @aa:16 → @aa:16			—	—	↑	↓	0	—
	OR.W @aa:16,@aa:32	W	5	4	8								S			@aa:32 ∨ @aa:16 → @aa:32			—	—	↑	↓	0	—
	OR.W @aa:32,@ERd	W	4	4	7				D				S			@ERd ∨ @aa:32 → @ERd			—	—	↑	↓	0	—
	OR.W @aa:32,@ERd+	W	4	4	7					D	S		S			@ERd ∨ @aa:32 → @ERd		ERd32+2→ERd32	—	—	↑	↓	0	—
	OR.W @aa:32,@ERd-	W	4	4	7					D	S		S			@ERd ∨ @aa:32 → @ERd		ERd32-2→ERd32	—	—	↑	↓	0	—
	OR.W @aa:32,@+ERd	W	4	5	8					D	S		S		ERd32+2→ERd32	@ERd ∨ @aa:32 → @ERd			—	—	↑	↓	0	—
	OR.W @aa:32,@-ERd	W	4	5	8					D	S		S		ERd32-2→ERd32	@ERd ∨ @aa:32 → @ERd			—	—	↑	↓	0	—
	OR.W @aa:32,@(d:2,ERd)	W	4	5	8					D	S		S			@(2/4/6+ERd) ∨ @aa:32 → @(2/4/6+ERd)			—	—	↑	↓	0	—
	OR.W @aa:32,@(d:16,ERd)	W	5	5	8					D	S		S			@(d:16+ERd) ∨ @aa:32 → @(d:16+ERd)			—	—	↑	↓	0	—
	OR.W @aa:32,@(d:32,ERd)	W	6	5	9					D	S		S			@(d:32+ERd) ∨ @aa:32 → @(d:32+ERd)			—	—	↑	↓	0	—
	OR.W @aa:32,@(d:16,Rd.B)	W	5	5	8						D	S				@(d:16+RdL<<1) ∨ @aa:32 → @(d:16+RdL<<1)			—	—	↑	↓	0	—
	OR.W @aa:32,@(d:16,Rd.W)	W	5	5	8						D	S				@(d:16+Rd<<1) ∨ @aa:32 → @(d:16+Rd<<1)			—	—	↑	↓	0	—
	OR.W @aa:32,@(d:16,ERd.L)	W	5	5	8						D	S				@(d:16+ERd<<1) ∨ @aa:32 → @(d:16+ERd<<1)			—	—	↑	↓	0	—
	OR.W @aa:32,@(d:32,Rd.B)	W	6	5	9						D	S				@(d:32+RdL<<1) ∨ @aa:32 → @(d:32+RdL<<1)			—	—	↑	↓	0	—
OR.W @aa:32,@(d:32,Rd.W)	W	6	5	9						D	S				@(d:32+Rd<<1) ∨ @aa:32 → @(d:32+Rd<<1)			—	—	↑	↓	0	—	
OR.W @aa:32,@(d:32,ERd.L)	W	6	5	9						D	S				@(d:32+ERd<<1) ∨ @aa:32 → @(d:32+ERd<<1)			—	—	↑	↓	0	—	
OR.W @aa:32,@aa:16	W	5	4	8								S			@aa:16 ∨ @aa:32 → @aa:16			—	—	↑	↓	0	—	
OR.W @aa:32,@aa:32	W	6	4	9								S			@aa:32 ∨ @aa:32 → @aa:32			—	—	↑	↓	0	—	
OR.L #xx:16,ERd	L	2	1	2	S		D									ERd32 ∨ #xx:16 → ERd32			—	—	↑	↓	0	—

Logic operations (64)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stage ¹	Addressing Mode							Operation ⁵					Condition Codes ^{6,7}																		
						Number of	Rn	@(d,ERn)	@(d,Rn,WRn,LRn)	@(d,ERn)/@ERn+/@ERn/+/ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C														
OR	ORL #xx:32,ERd	L	3	1	3	S		D							ERd32	∨	#xx	→	ERd32																	
	ORL #xx:16,@ERd	L	3	4	5	S		D							@ERd	∨	#xx:16	→	@ERd																	
	ORL #xx:16,@ERd+	L	3	4	5	S				D					@ERd	∨	#xx:16	→	@ERd		ERd32+4→ERd32															
	ORL #xx:16,@ERd-	L	3	4	5	S				D					@ERd	∨	#xx:16	→	@ERd		ERd32-4→ERd32															
	ORL #xx:16,@+ERd	L	3	5	5	S					D				ERd32+4→ERd32	@ERd	∨	#xx:16	→	@ERd																
	ORL #xx:16,@-ERd	L	3	5	5	S					D				ERd32-4→ERd32	@ERd	∨	#xx:16	→	@ERd																
	ORL #xx:16,@(d:2,ERd)	L	3	5	5	S					D				@(4/8/12+ERd)	∨	#xx:16	→	@(4/8/12+ERd)																	
	ORL #xx:16,@(d:16,ERd)	L	4	5	6	S					D				@(d:16+ERd)	∨	#xx:16	→	@(d:16+ERd)																	
	ORL #xx:16,@(d:32,ERd)	L	5	5	7	S					D				@(d:32+ERd)	∨	#xx:16	→	@(d:32+ERd)																	
	ORL #xx:16,@(d:16,Rd,B)	L	4	5	6	S					D				@(d:16+Rd<<2)	∨	#xx:16	→	@(d:16+Rd<<2)																	
	ORL #xx:16,@(d:16,Rd,W)	L	4	5	6	S					D				@(d:16+Rd<<2)	∨	#xx:16	→	@(d:16+Rd<<2)																	
	ORL #xx:16,@(d:16,ERd,L)	L	4	5	6	S					D				@(d:16+ERd<<2)	∨	#xx:16	→	@(d:16+ERd<<2)																	
	ORL #xx:16,@(d:32,Rd,B)	L	5	5	7	S					D				@(d:32+Rd<<2)	∨	#xx:16	→	@(d:32+Rd<<2)																	
	ORL #xx:16,@(d:32,Rd,W)	L	5	5	7	S					D				@(d:32+Rd<<2)	∨	#xx:16	→	@(d:32+Rd<<2)																	
	ORL #xx:16,@(d:32,ERd,L)	L	5	5	7	S					D				@(d:32+ERd<<2)	∨	#xx:16	→	@(d:32+ERd<<2)																	
	ORL #xx:16,@aa:16	L	4	4	6	S							D		@aa:16	∨	#xx:16	→	@aa:16																	
	ORL #xx:16,@aa:32	L	5	4	7	S							D		@aa:32	∨	#xx:16	→	@aa:32																	
	ORL #xx:32,@ERd	L	4	4	6	S					D				@ERd	∨	#xx	→	@ERd																	
	ORL #xx:32,@ERd+	L	4	4	6	S							D		@ERd	∨	#xx	→	@ERd		ERd32+4→ERd32															
	ORL #xx:32,@ERd-	L	4	4	6	S							D		@ERd	∨	#xx	→	@ERd		ERd32-4→ERd32															
	ORL #xx:32,@+ERd	L	4	5	6	S								D	ERd32+4→ERd32	@ERd	∨	#xx	→	@ERd																
	ORL #xx:32,@-ERd	L	4	5	6	S								D	ERd32-4→ERd32	@ERd	∨	#xx	→	@ERd																
	ORL #xx:32,@(d:2,ERd)	L	4	5	6	S								D	@(4/8/12+ERd)	∨	#xx	→	@(4/8/12+ERd)																	
	ORL #xx:32,@(d:16,ERd)	L	5	5	7	S								D	@(d:16+ERd)	∨	#xx	→	@(d:16+ERd)																	
	ORL #xx:32,@(d:32,ERd)	L	6	5	8	S								D	@(d:32+ERd)	∨	#xx	→	@(d:32+ERd)																	

Logic operations (65)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	Number of 16-Bit Instruction Fetch Execution Stages ¹	Addressing Mode							Operation ⁵					Condition Codes ²											
							Rn	RRn	RRn	@(d, ERn)	@(d, Rn, L, RRn, W, ERn, L)	@(d, Rn, L, ERn, W, ERn, L)	@(ERn, @ERn, @ERn, @ERn, @+ERn)	@aa:9 / @aa:16 / @aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C					
OR	OR.L #xx:32, @(d:16, Rd, B)	L	5	5	7	S					D					@(d:16+RdL<<2) ∨ #xx → @(d:16+RdL<<2)							—	—	↑	↓	0	—		
	OR.L #xx:32, @(d:16, Rd, W)	L	5	5	7	S					D					@(d:16+Rd<<2) ∨ #xx → @(d:16+Rd<<2)							—	—	↑	↓	0	—		
	OR.L #xx:32, @(d:16, ERd, L)	L	5	5	7	S					D					@(d:16+ERd<<2) ∨ #xx → @(d:16+ERd<<2)							—	—	↑	↓	0	—		
	OR.L #xx:32, @(d:32, Rd, B)	L	6	5	8	S					D					@(d:32+RdL<<2) ∨ #xx → @(d:32+RdL<<2)							—	—	↑	↓	0	—		
	OR.L #xx:32, @(d:32, Rd, W)	L	6	5	8	S					D					@(d:32+Rd<<2) ∨ #xx → @(d:32+Rd<<2)							—	—	↑	↓	0	—		
	OR.L #xx:32, @(d:32, ERd, L)	L	6	5	8	S					D					@(d:32+ERd<<2) ∨ #xx → @(d:32+ERd<<2)							—	—	↑	↓	0	—		
	OR.L #xx:32, @aa:16	L	5	4	7	S						D				@aa:16 ∨ #xx → @aa:16							—	—	↑	↓	0	—		
	OR.L #xx:32, @aa:32	L	6	4	8	S						D				@aa:32 ∨ #xx → @aa:32							—	—	↑	↓	0	—		
	OR.L ERs, ERd	L	2	2	2			S								ERd32	∨	ERs							—	—	↑	↓	0	—
	OR.L ERs, @ERd	L	2	4	4			S	D							@ERd	∨	ERs							—	—	↑	↓	0	—
	OR.L ERs, @ERd+	L	2	4	4			S			D					@ERd	∨	ERs							—	—	↑	↓	0	—
	OR.L ERs, @ERd-	L	2	4	4			S			D					@ERd	∨	ERs							—	—	↑	↓	0	—
	OR.L ERs, @+ERd	L	2	5	5			S			D				ERd32+4 → ERd32	@ERd	∨	ERs							—	—	↑	↓	0	—
	OR.L ERs, @-ERd	L	2	5	5			S			D				ERd32-4 → ERd32	@ERd	∨	ERs							—	—	↑	↓	0	—
	OR.L ERs, @(d:2, ERd)	L	2	5	5			S	D							@(4/8/12+ERd) ∨ ERs → @(4/8/12+ERd)							—	—	↑	↓	0	—		
	OR.L ERs, @(d:16, ERd)	L	3	5	5			S	D							@(d:16+ERd) ∨ ERs → @(d:16+ERd)							—	—	↑	↓	0	—		
	OR.L ERs, @(d:32, ERd)	L	4	5	6			S	D							@(d:32+ERd) ∨ ERs → @(d:32+ERd)							—	—	↑	↓	0	—		
	OR.L ERs, @(d:16, Rd, B)	L	3	5	5			S			D					@(d:16+RdL<<2) ∨ ERs → @(d:16+RdL<<2)							—	—	↑	↓	0	—		
	OR.L ERs, @(d:16, Rd, W)	L	3	5	5			S			D					@(d:16+Rd<<2) ∨ ERs → @(d:16+Rd<<2)							—	—	↑	↓	0	—		
	OR.L ERs, @(d:16, ERd, L)	L	3	5	5			S			D					@(d:16+ERd<<2) ∨ ERs → @(d:16+ERd<<2)							—	—	↑	↓	0	—		
	OR.L ERs, @(d:32, Rd, B)	L	4	5	6			S			D					@(d:32+RdL<<2) ∨ ERs → @(d:32+RdL<<2)							—	—	↑	↓	0	—		
OR.L ERs, @(d:32, Rd, W)	L	4	5	6			S			D					@(d:32+Rd<<2) ∨ ERs → @(d:32+Rd<<2)							—	—	↑	↓	0	—			
OR.L ERs, @(d:32, ERd, L)	L	4	5	6			S			D					@(d:32+ERd<<2) ∨ ERs → @(d:32+ERd<<2)							—	—	↑	↓	0	—			
OR.L ERs, @aa:16	L	3	4	5			S				D				@aa:16 ∨ ERs → @aa:16							—	—	↑	↓	0	—			
OR.L ERs, @aa:32	L	4	4	6			S				D				@aa:32 ∨ ERs → @aa:32							—	—	↑	↓	0	—			



Logic operations (67)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode						Operation ⁵					Condition Codes ^{6,7}						
				Min.	16-Bit Instruction Fetch #xxx	Rn	@(d,ERn)	@(d,Rn,ERn,WERn.L)	@(ERn)/@ERn+/@ERn+/@+ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C
OR	ORL @ERs,@(d:16,Rd,W)	L	4 6 7			S	D				@(d:16+Rd<<2) ∨ @ERs → @(d:16+Rd<<2)										
	ORL @ERs,@(d:16,ERd,L)	L	4 6 7			S	D				@(d:16+ERd<<2) ∨ @ERs → @(d:16+ERd<<2)										
	ORL @ERs,@(d:32,Rd,B)	L	5 6 8			S	D				@(d:32+RdL<<2) ∨ @ERs → @(d:32+RdL<<2)										
	ORL @ERs,@(d:32,Rd,W)	L	5 6 8			S	D				@(d:32+Rd<<2) ∨ @ERs → @(d:32+Rd<<2)										
	ORL @ERs,@(d:32,ERd,L)	L	5 6 8			S	D				@(d:32+ERd<<2) ∨ @ERs → @(d:32+ERd<<2)										
	ORL @ERs,@aa:16	L	4 5 7			S			D		@aa:16 ∨ @ERs → @aa:16										
	ORL @ERs,@aa:32	L	5 5 8			S			D		@aa:32 ∨ @ERs → @aa:32										
	ORL @ERs+,@ERd	L	3 5 6			D	S		S		@ERd ∨ @ERs → @ERd	ERs32+4→ERs32									
	ORL @ERs+,@ERd+	L	3 5 6				S		S		@ERd ∨ @ERs → @ERd	ERs32+4→ERs32	ERd32+4→ERd32								
	ORL @ERs+,@ERd-	L	3 5 6				S		S		@ERd ∨ @ERs → @ERd	ERs32+4→ERs32	ERd32-4→ERd32								
	ORL @ERs+,@+ERd	L	3 6 7				S		S		ERd32+4→ERd32 @ERd ∨ @ERs → @ERd	ERs32+4→ERs32									
	ORL @ERs+,@-ERd	L	3 6 7				S		S		ERd32-4→ERd32 @ERd ∨ @ERs → @ERd	ERs32+4→ERs32									
	ORL @ERs+,@(d:2,ERd)	L	3 6 7				D	S			@(4/8/12+ERd) ∨ @ERs → @(4/8/12+ERd)	ERs32+4→ERs32									
	ORL @ERs+,@(d:16,ERd)	L	4 6 7				D	S			@(d:16+ERd) ∨ @ERs → @(d:16+ERd)	ERs32+4→ERs32									
	ORL @ERs+,@(d:32,ERd)	L	5 6 8				D	S			@(d:32+ERd) ∨ @ERs → @(d:32+ERd)	ERs32+4→ERs32									
	ORL @ERs+,@(d:16,Rd,B)	L	4 6 7				D	S			@(d:16+RdL<<2) ∨ @ERs → @(d:16+RdL<<2)	ERs32+4→ERs32									
	ORL @ERs+,@(d:16,Rd,W)	L	4 6 7				D	S			@(d:16+Rd<<2) ∨ @ERs → @(d:16+Rd<<2)	ERs32+4→ERs32									
	ORL @ERs+,@(d:16,ERd,L)	L	4 6 7				D	S			@(d:16+ERd<<2) ∨ @ERs → @(d:16+ERd<<2)	ERs32+4→ERs32									
	ORL @ERs+,@(d:32,Rd,B)	L	5 6 8				D	S			@(d:32+RdL<<2) ∨ @ERs → @(d:32+RdL<<2)	ERs32+4→ERs32									
	ORL @ERs+,@(d:32,Rd,W)	L	5 6 8				D	S			@(d:32+Rd<<2) ∨ @ERs → @(d:32+Rd<<2)	ERs32+4→ERs32									
	ORL @ERs+,@(d:32,ERd,L)	L	5 6 8				D	S			@(d:32+ERd<<2) ∨ @ERs → @(d:32+ERd<<2)	ERs32+4→ERs32									
	ORL @ERs+,@aa:16	L	4 5 7				S	D			@aa:16 ∨ @ERs → @aa:16	ERs32+4→ERs32									
	ORL @ERs+,@aa:32	L	5 5 8				S	D			@aa:32 ∨ @ERs → @aa:32	ERs32+4→ERs32									
	ORL @ERs+,@ERd	L	3 5 6			D	S		S		@ERd ∨ @ERs → @ERd	ERs32-4→ERs32									

Logic operations (68)

Instruction n	Mnemonic	Size	Instruction Length	Addressing Mode								Operation ⁵					Condition Codes ^{6,7}										
				Min.	16-bit Instruction #xx	Rn	@Ern	@ (d,Ern)	@ (d,Rn,Ern,WEm.L)	@ -Ern/@Ern/@Ern-/@+Ern	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C					
																							Number of Execution Stages ¹				
OR	ORL @Ers-, @Erd+	L	3	5	6									@Erd	∨	@Ers	→	@Erd	Ers32-4→Ers32	Erd32+4→Erd32	—	—	↑	↓	0	—	
	ORL @Ers-, @Erd-	L	3	5	6									@Erd	∨	@Ers	→	@Erd	Ers32-4→Ers32	Erd32-4→Erd32	—	—	↑	↓	0	—	
	ORL @Ers-, @+Erd	L	3	6	7									Erd32+4→Erd32	@Erd	∨	@Ers	→	@Erd	Ers32-4→Ers32		—	—	↑	↓	0	—
	ORL @Ers-, @-Erd	L	3	6	7									Erd32-4→Erd32	@Erd	∨	@Ers	→	@Erd	Ers32-4→Ers32		—	—	↑	↓	0	—
	ORL @Ers-, @ (d:2,Erd)	L	3	6	7				D	S				@ (4/8/12+Erd)	∨	@Ers	→	@ (4/8/12+Erd)	Ers32-4→Ers32		—	—	↑	↓	0	—	
	ORL @Ers-, @ (d:16,Erd)	L	4	6	7				D	S				@ (d:16+Erd)	∨	@Ers	→	@ (d:16+Erd)	Ers32-4→Ers32		—	—	↑	↓	0	—	
	ORL @Ers-, @ (d:32,Erd)	L	5	6	8				D	S				@ (d:32+Erd)	∨	@Ers	→	@ (d:32+Erd)	Ers32-4→Ers32		—	—	↑	↓	0	—	
	ORL @Ers-, @ (d:16,Rd,B)	L	4	6	7				D	S				@ (d:16+Rd<<2)	∨	@Ers	→	@ (d:16+Rd<<2)	Ers32-4→Ers32		—	—	↑	↓	0	—	
	ORL @Ers-, @ (d:16,Rd,W)	L	4	6	7				D	S				@ (d:16+Rd<<2)	∨	@Ers	→	@ (d:16+Rd<<2)	Ers32-4→Ers32		—	—	↑	↓	0	—	
	ORL @Ers-, @ (d:16,Erd,L)	L	4	6	7				D	S				@ (d:16+Erd<<2)	∨	@Ers	→	@ (d:16+Erd<<2)	Ers32-4→Ers32		—	—	↑	↓	0	—	
	ORL @Ers-, @ (d:32,Rd,B)	L	5	6	8				D	S				@ (d:32+Rd<<2)	∨	@Ers	→	@ (d:32+Rd<<2)	Ers32-4→Ers32		—	—	↑	↓	0	—	
	ORL @Ers-, @ (d:32,Rd,W)	L	5	6	8				D	S				@ (d:32+Rd<<2)	∨	@Ers	→	@ (d:32+Rd<<2)	Ers32-4→Ers32		—	—	↑	↓	0	—	
	ORL @Ers-, @ (d:32,Erd,L)	L	5	6	8				D	S				@ (d:32+Erd<<2)	∨	@Ers	→	@ (d:32+Erd<<2)	Ers32-4→Ers32		—	—	↑	↓	0	—	
	ORL @Ers-, @aa:16	L	4	5	7						S	D			@aa:16	∨	@Ers	→	@aa:16	Ers32-4→Ers32		—	—	↑	↓	0	—
	ORL @Ers-, @aa:32	L	5	5	8							S	D		@aa:32	∨	@Ers	→	@aa:32	Ers32-4→Ers32		—	—	↑	↓	0	—
	ORL @+Ers, @Erd	L	3	6	6				D		S			Ers32+4→Ers32	@Erd	∨	@Ers	→	@Erd			—	—	↑	↓	0	—
	ORL @+Ers, @Erd+	L	3	6	6						S			Ers32+4→Ers32	@Erd	∨	@Ers	→	@Erd		Erd32+4→Erd32	—	—	↑	↓	0	—
	ORL @+Ers, @Erd-	L	3	6	6						S			Ers32+4→Ers32	@Erd	∨	@Ers	→	@Erd		Erd32-4→Erd32	—	—	↑	↓	0	—
	ORL @+Ers, @+Erd	L	3	7	7						S			Ers32+4→Ers32	Erd32+4→Erd32	@Erd	∨	@Ers	→	@Erd		—	—	↑	↓	0	—
	ORL @+Ers, @-Erd	L	3	7	7						S			Ers32+4→Ers32	Erd32-4→Erd32	@Erd	∨	@Ers	→	@Erd		—	—	↑	↓	0	—
	ORL @+Ers, @ (d:2,Erd)	L	3	7	7				D	S				Ers32+4→Ers32	@ (4/8/12+Erd)	∨	@Ers	→	@ (4/8/12+Erd)			—	—	↑	↓	0	—
	ORL @+Ers, @ (d:16,Erd)	L	4	7	7				D	S				Ers32+4→Ers32	@ (d:16+Erd)	∨	@Ers	→	@ (d:16+Erd)		—	—	↑	↓	0	—	
	ORL @+Ers, @ (d:32,Erd)	L	5	7	8				D	S				Ers32+4→Ers32	@ (d:32+Erd)	∨	@Ers	→	@ (d:32+Erd)		—	—	↑	↓	0	—	
	ORL @+Ers, @ (d:16,Rd,B)	L	4	7	7				D	S				Ers32+4→Ers32	@ (d:16+Rd<<2)	∨	@Ers	→	@ (d:16+Rd<<2)		—	—	↑	↓	0	—	
ORL @+Ers, @ (d:16,Rd,W)	L	4	7	7				D	S				Ers32+4→Ers32	@ (d:16+Rd<<2)	∨	@Ers	→	@ (d:16+Rd<<2)		—	—	↑	↓	0	—		



Logic operations (69)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch	Addressing Mode						Operation ⁵					Condition Codes ^{6,7}										
						Number of Execution Stages ⁸	Frxx	Rn	@ERn	@(d,ERn)	@(d,Rn,ERn,WERn,L)	@(d,Rn,ERn+/@ERn+/@ERn+/@+ERn)	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C			
OR	ORL @+ERs,@(d:16,ERd,L)	L	4	7	7					D	S			ERs32+4→ERs32		@(d:16+ERd<<2) ∨ @ERs → @(d:16+ERd<<2)				—	—	↑	↓	0	—		
	ORL @+ERs,@(d:32,Rd,B)	L	5	7	8					D	S			ERs32+4→ERs32		@(d:32+RdL<<2) ∨ @ERs → @(d:32+RdL<<2)				—	—	↑	↓	0	—		
	ORL @+ERs,@(d:32,Rd,W)	L	5	7	8					D	S			ERs32+4→ERs32		@(d:32+Rd<<2) ∨ @ERs → @(d:32+Rd<<2)				—	—	↑	↓	0	—		
	ORL @+ERs,@(d:32,ERd,L)	L	5	7	8					D	S			ERs32+4→ERs32		@(d:32+ERd<<2) ∨ @ERs → @(d:32+ERd<<2)				—	—	↑	↓	0	—		
	ORL @+ERs,@aa:16	L	4	6	7							S	D	ERs32+4→ERs32		@aa:16 ∨ @ERs → @aa:16				—	—	↑	↓	0	—		
	ORL @+ERs,@aa:32	L	5	6	8								S	D	ERs32+4→ERs32		@aa:32 ∨ @ERs → @aa:32				—	—	↑	↓	0	—	
	ORL @-ERs,@ERd	L	3	6	6						D	S			ERs32-4→ERs32		@ERd ∨ @ERs → @ERd				—	—	↑	↓	0	—	
	ORL @-ERs,@ERd+	L	3	6	6							S			ERs32-4→ERs32		@ERd ∨ @ERs → @ERd	ERd32+4→ERd32				—	—	↑	↓	0	—
	ORL @-ERs,@ERd-	L	3	6	6							S			ERs32-4→ERs32		@ERd ∨ @ERs → @ERd	ERd32-4→ERd32				—	—	↑	↓	0	—
	ORL @-ERs,@+ERd	L	3	7	7							S			ERs32-4→ERs32	ERd32+4→ERd32	@ERd ∨ @ERs → @ERd				—	—	↑	↓	0	—	
	ORL @-ERs,@-ERd	L	3	7	7							S			ERs32-4→ERs32	ERd32-4→ERd32	@ERd ∨ @ERs → @ERd				—	—	↑	↓	0	—	
	ORL @-ERs,@(d:2,ERd)	L	3	7	7						D	S			ERs32-4→ERs32		@(4/8/12+ERd) ∨ @ERs → @(4/8/12+ERd)				—	—	↑	↓	0	—	
	ORL @-ERs,@(d:16,ERd)	L	4	7	7						D	S			ERs32-4→ERs32		@(d:16+ERd) ∨ @ERs → @(d:16+ERd)				—	—	↑	↓	0	—	
	ORL @-ERs,@(d:32,ERd)	L	5	7	8						D	S			ERs32-4→ERs32		@(d:32+ERd) ∨ @ERs → @(d:32+ERd)				—	—	↑	↓	0	—	
	ORL @-ERs,@(d:16,Rd,B)	L	4	7	7							D	S		ERs32-4→ERs32		@(d:16+RdL<<2) ∨ @ERs → @(d:16+RdL<<2)				—	—	↑	↓	0	—	
	ORL @-ERs,@(d:16,Rd,W)	L	4	7	7							D	S		ERs32-4→ERs32		@(d:16+Rd<<2) ∨ @ERs → @(d:16+Rd<<2)				—	—	↑	↓	0	—	
	ORL @-ERs,@(d:16,ERd,L)	L	4	7	7							D	S		ERs32-4→ERs32		@(d:16+ERd<<2) ∨ @ERs → @(d:16+ERd<<2)				—	—	↑	↓	0	—	
	ORL @-ERs,@(d:32,Rd,B)	L	5	7	8							D	S		ERs32-4→ERs32		@(d:32+RdL<<2) ∨ @ERs → @(d:32+RdL<<2)				—	—	↑	↓	0	—	
	ORL @-ERs,@(d:32,Rd,W)	L	5	7	8							D	S		ERs32-4→ERs32		@(d:32+Rd<<2) ∨ @ERs → @(d:32+Rd<<2)				—	—	↑	↓	0	—	
	ORL @-ERs,@(d:32,ERd,L)	L	5	7	8							D	S		ERs32-4→ERs32		@(d:32+ERd<<2) ∨ @ERs → @(d:32+ERd<<2)				—	—	↑	↓	0	—	
ORL @-ERs,@aa:16	L	4	6	7							S	D		ERs32-4→ERs32		@aa:16 ∨ @ERs → @aa:16				—	—	↑	↓	0	—		
ORL @-ERs,@aa:32	L	5	6	8							S	D		ERs32-4→ERs32		@aa:32 ∨ @ERs → @aa:32				—	—	↑	↓	0	—		
ORL @(d:2,ERs),@ERd	L	3	6	6						D	S					@ERd ∨ @(4/8/12+ERs) → @ERd				—	—	↑	↓	0	—		
ORL @(d:2,ERs),@ERd+	L	3	6	6							S	D				@ERd ∨ @(4/8/12+ERs) → @ERd	ERd32+4→ERd32				—	—	↑	↓	0	—	
ORL @(d:2,ERs),@ERd-	L	3	6	6							S	D				@ERd ∨ @(4/8/12+ERs) → @ERd	ERd32-4→ERd32				—	—	↑	↓	0	—	



Logic operations (70)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode								Operation ^s					Condition Codes ^{s2}										
				Min.	16-Bit Instruction Fetch Execution Stages ¹	#rx	Rn	@ERn	@(d,ERn)	@(d,Rn,LRn,WERn,L)	@ERn/@ERN+/@ERNr/@+ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C				
OR	ORL @(d:2,ERs),@+ERd	L	3 7 7					S		D				ERd32+4 → ERd32	@ERd	∨ @ (4/8/12+ERs) → @ERd					—	—	↑	↓	0	—	
	ORL @(d:2,ERs),@-ERd	L	3 7 7					S		D				ERd32-4 → ERd32	@ERd	∨ @ (4/8/12+ERs) → @ERd					—	—	↑	↓	0	—	
	ORL @(d:2,ERs),@(d:2,ERd)	L	3 7 7					S							@ (4/8/12+ERd)	∨ @ (4/8/12+ERs) → @ (4/8/12+ERd)					—	—	↑	↓	0	—	
	ORL @(d:2,ERs),@(d:16,ERd)	L	4 7 7					S							@ (d:16+ERd)	∨ @ (4/8/12+ERs) → @ (d:16+ERd)					—	—	↑	↓	0	—	
	ORL @(d:2,ERs),@(d:32,ERd)	L	5 7 8					S							@ (d:32+ERd)	∨ @ (4/8/12+ERs) → @ (d:32+ERd)					—	—	↑	↓	0	—	
	ORL @(d:2,ERs),@(d:16,Rd.B)	L	4 7 7					S		D					@ (d:16+RdL<<2)	∨ @ (4/8/12+ERs) → @ (d:16+RdL<<2)					—	—	↑	↓	0	—	
	ORL @(d:2,ERs),@(d:16,Rd.W)	L	4 7 7					S		D					@ (d:16+Rd<<2)	∨ @ (4/8/12+ERs) → @ (d:16+Rd<<2)					—	—	↑	↓	0	—	
	ORL @(d:2,ERs),@(d:16,ERd.L)	L	4 7 7					S		D					@ (d:16+ERd<<2)	∨ @ (4/8/12+ERs) → @ (d:16+ERd<<2)					—	—	↑	↓	0	—	
	ORL @(d:2,ERs),@(d:32,Rd.B)	L	5 7 8					S		D					@ (d:32+RdL<<2)	∨ @ (4/8/12+ERs) → @ (d:32+RdL<<2)					—	—	↑	↓	0	—	
	ORL @(d:2,ERs),@(d:32,Rd.W)	L	5 7 8					S		D					@ (d:32+Rd<<2)	∨ @ (4/8/12+ERs) → @ (d:32+Rd<<2)					—	—	↑	↓	0	—	
	ORL @(d:2,ERs),@(d:32,ERd.L)	L	5 7 8					S		D					@ (d:32+ERd<<2)	∨ @ (4/8/12+ERs) → @ (d:32+ERd<<2)					—	—	↑	↓	0	—	
	ORL @(d:2,ERs),@aa:16	L	4 6 7					S		D					@aa:16	∨ @ (4/8/12+ERs) → @aa:16					—	—	↑	↓	0	—	
	ORL @(d:2,ERs),@aa:32	L	5 6 8					S		D					@aa:32	∨ @ (4/8/12+ERs) → @aa:32					—	—	↑	↓	0	—	
	ORL @(d:16,ERs),@ERd	L	4 6 7					S			D				@ERd	∨ @ (d:16+ERs) → @ERd					—	—	↑	↓	0	—	
	ORL @(d:16,ERs),@ERd+	L	4 6 7					S		D					@ERd	∨ @ (d:16+ERs) → @ERd					ERd32+4 → ERd32	—	—	↑	↓	0	—
	ORL @(d:16,ERs),@ERd-	L	4 6 7					S		D					@ERd	∨ @ (d:16+ERs) → @ERd					ERd32-4 → ERd32	—	—	↑	↓	0	—
	ORL @(d:16,ERs),@+ERd	L	4 7 8					S		D				ERd32+4 → ERd32	@ERd	∨ @ (d:16+ERs) → @ERd					—	—	↑	↓	0	—	
	ORL @(d:16,ERs),@-ERd	L	4 7 8					S		D				ERd32-4 → ERd32	@ERd	∨ @ (d:16+ERs) → @ERd					—	—	↑	↓	0	—	
	ORL @(d:16,ERs),@(d:2,ERd)	L	4 7 8					S							@ (4/8/12+ERd)	∨ @ (d:16+ERs) → @ (4/8/12+ERd)					—	—	↑	↓	0	—	
	ORL @(d:16,ERs),@(d:16,ERd)	L	5 7 8					S							@ (d:16+ERd)	∨ @ (d:16+ERs) → @ (d:16+ERd)					—	—	↑	↓	0	—	
	ORL @(d:16,ERs),@(d:32,ERd)	L	6 7 9					S							@ (d:32+ERd)	∨ @ (d:16+ERs) → @ (d:32+ERd)					—	—	↑	↓	0	—	
	ORL @(d:16,ERs),@(d:16,Rd.B)	L	5 7 8					S		D					@ (d:16+RdL<<2)	∨ @ (d:16+ERs) → @ (d:16+RdL<<2)					—	—	↑	↓	0	—	
	ORL @(d:16,ERs),@(d:16,Rd.W)	L	5 7 8					S		D					@ (d:16+Rd<<2)	∨ @ (d:16+ERs) → @ (d:16+Rd<<2)					—	—	↑	↓	0	—	
	ORL @(d:16,ERs),@(d:16,ERd.L)	L	5 7 8					S		D					@ (d:16+ERd<<2)	∨ @ (d:16+ERs) → @ (d:16+ERd<<2)					—	—	↑	↓	0	—	
	ORL @(d:16,ERs),@(d:32,Rd.B)	L	6 7 9					S		D					@ (d:32+RdL<<2)	∨ @ (d:16+ERs) → @ (d:32+RdL<<2)					—	—	↑	↓	0	—	

Logic operations (71)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode								Operation ⁵					Condition Codes ^{6,2}											
				Min.	16-Bit Instruction Fetch Execution Stages ¹	Number of Fetches	Rn	ERn	@(d,ERn)	@(d,Rn,LRn,WERn,L)	@(d,Rn)/@ERn+/@ERn+/@ERn+/@ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C					
OR	ORL @(d:16,ERs),@(d:32,Rd,W)	L	6 7 9					S	D																			
	ORL @(d:16,ERs),@(d:32,ERd,L)	L	6 7 9					S	D																			
	ORL @(d:16,ERs),@aa:16	L	5 6 8					S	D																			
	ORL @(d:16,ERs),@aa:32	L	6 6 9					S	D																			
	ORL @(d:32,ERs),@ERd	L	5 6 8					D	S																			
	ORL @(d:32,ERs),@ERd+	L	5 6 8					S	D																			
	ORL @(d:32,ERs),@ERd-	L	5 6 8					S	D																			
	ORL @(d:32,ERs),@+ERd	L	5 7 9					S	D																			
	ORL @(d:32,ERs),@-ERd	L	5 7 9					S	D																			
	ORL @(d:32,ERs),@(d:2,ERd)	L	5 7 9					S																				
	ORL @(d:32,ERs),@(d:16,ERd)	L	6 7 9					S																				
	ORL @(d:32,ERs),@(d:32,ERd)	L	7 7 1					S																				
	ORL @(d:32,ERs),@(d:16,Rd,B)	L	6 7 9					S	D																			
	ORL @(d:32,ERs),@(d:16,Rd,W)	L	6 7 9					S	D																			
	ORL @(d:32,ERs),@(d:16,ERd,L)	L	6 7 9					S	D																			
	ORL @(d:32,ERs),@(d:32,Rd,B)	L	7 7 1					S	D																			
	ORL @(d:32,ERs),@(d:32,Rd,W)	L	7 7 1					S	D																			
	ORL @(d:32,ERs),@(d:32,ERd,L)	L	7 7 1					S	D																			
	ORL @(d:32,ERs),@aa:16	L	6 6 9					S	D																			
	ORL @(d:32,ERs),@aa:32	L	7 6 1					S	D																			
	ORL @(d:16,Rs,B),@ERd	L	4 6 7					D	S																			
	ORL @(d:16,Rs,B),@ERd+	L	4 6 7					S	D																			
	ORL @(d:16,Rs,B),@ERd-	L	4 6 7					S	D																			
	ORL @(d:16,Rs,B),@+ERd	L	4 7 8					S	D																			
	ORL @(d:16,Rs,B),@-ERd	L	4 7 8					S	D																			

Logic operations (72)



Instruction n	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch #pxx	Addressing Mode												Operation ⁶					Condition Codes ^{5,2}					
						Number of Execution Stages ¹	Rn	Rm	Rd	Op	Op	Op	Op	Op	Op	Op	Op											
OR	ORL @(d:16,Rs.B),@(d:2,ERd)	L	4	7	8				D	S							@(4/8/12+ERd) ∨ @(d:16+RsL<<2) → @(4/8/12+ERd)						—	—	↑	↓	0	—
	ORL @(d:16,Rs.B),@(d:16,ERd)	L	5	7	8				D	S							@(d:16+ERd) ∨ @(d:16+RsL<<2) → @(d:16+ERd)						—	—	↑	↓	0	—
	ORL @(d:16,Rs.B),@(d:32,ERd)	L	6	7	9				D	S							@(d:32+ERd) ∨ @(d:16+RsL<<2) → @(d:32+ERd)						—	—	↑	↓	0	—
	ORL @(d:16,Rs.B),@(d:16,Rd.B)	L	5	7	8					S							@(d:16+RdL<<2) ∨ @(d:16+RsL<<2) → @(d:16+RdL<<2)						—	—	↑	↓	0	—
	ORL @(d:16,Rs.B),@(d:16,Rd.W)	L	5	7	8					S							@(d:16+Rd<<2) ∨ @(d:16+RsL<<2) → @(d:16+Rd<<2)						—	—	↑	↓	0	—
	ORL @(d:16,Rs.B),@(d:16,ERd.L)	L	5	7	8					S							@(d:16+ERd<<2) ∨ @(d:16+RsL<<2) → @(d:16+ERd<<2)						—	—	↑	↓	0	—
	ORL @(d:16,Rs.B),@(d:32,Rd.B)	L	6	7	9					S							@(d:32+RdL<<2) ∨ @(d:16+RsL<<2) → @(d:32+RdL<<2)						—	—	↑	↓	0	—
	ORL @(d:16,Rs.B),@(d:32,Rd.W)	L	6	7	9					S							@(d:32+Rd<<2) ∨ @(d:16+RsL<<2) → @(d:32+Rd<<2)						—	—	↑	↓	0	—
	ORL @(d:16,Rs.B),@(d:32,ERd.L)	L	6	7	9					S							@(d:32+ERd<<2) ∨ @(d:16+RsL<<2) → @(d:32+ERd<<2)						—	—	↑	↓	0	—
	ORL @(d:16,Rs.B),@aa:16	L	5	6	8					S	D						@aa:16 ∨ @(d:16+RsL<<2) → @aa:16						—	—	↑	↓	0	—
	ORL @(d:16,Rs.B),@aa:32	L	6	6	9					S	D						@aa:32 ∨ @(d:16+RsL<<2) → @aa:32						—	—	↑	↓	0	—
	ORL @(d:16,Rs.W),@ERd	L	4	6	7				D	S							@ERd ∨ @(d:16+Rs<<2) → @ERd						—	—	↑	↓	0	—
	ORL @(d:16,Rs.W),@ERd+	L	4	6	7					S	D						@ERd ∨ @(d:16+Rs<<2) → @ERd					ERd32+4→ERd32	—	—	↑	↓	0	—
	ORL @(d:16,Rs.W),@ERd-	L	4	6	7					S	D						@ERd ∨ @(d:16+Rs<<2) → @ERd					ERd32-4→ERd32	—	—	↑	↓	0	—
	ORL @(d:16,Rs.W),@+ERd	L	4	7	8					S	D				ERd32+4→ERd32		@ERd ∨ @(d:16+Rs<<2) → @ERd						—	—	↑	↓	0	—
	ORL @(d:16,Rs.W),@-ERd	L	4	7	8					S	D				ERd32-4→ERd32		@ERd ∨ @(d:16+Rs<<2) → @ERd						—	—	↑	↓	0	—
	ORL @(d:16,Rs.W),@(d:2,ERd)	L	4	7	8				D	S							@(4/8/12+ERd) ∨ @(d:16+Rs<<2) → @(4/8/12+ERd)						—	—	↑	↓	0	—
	ORL @(d:16,Rs.W),@(d:16,ERd)	L	5	7	8				D	S							@(d:16+ERd) ∨ @(d:16+Rs<<2) → @(d:16+ERd)						—	—	↑	↓	0	—
	ORL @(d:16,Rs.W),@(d:32,ERd)	L	6	7	9				D	S							@(d:32+ERd) ∨ @(d:16+Rs<<2) → @(d:32+ERd)						—	—	↑	↓	0	—
	ORL @(d:16,Rs.W),@(d:16,Rd.B)	L	5	7	8					S							@(d:16+RdL<<2) ∨ @(d:16+Rs<<2) → @(d:16+RdL<<2)						—	—	↑	↓	0	—
	ORL @(d:16,Rs.W),@(d:16,Rd.W)	L	5	7	8					S							@(d:16+Rd<<2) ∨ @(d:16+Rs<<2) → @(d:16+Rd<<2)						—	—	↑	↓	0	—
	ORL @(d:16,Rs.W),@(d:16,ERd.L)	L	5	7	8					S							@(d:16+ERd<<2) ∨ @(d:16+Rs<<2) → @(d:16+ERd<<2)						—	—	↑	↓	0	—
	ORL @(d:16,Rs.W),@(d:32,Rd.B)	L	6	7	9					S							@(d:32+RdL<<2) ∨ @(d:16+Rs<<2) → @(d:32+RdL<<2)						—	—	↑	↓	0	—
	ORL @(d:16,Rs.W),@(d:32,Rd.W)	L	6	7	9					S							@(d:32+Rd<<2) ∨ @(d:16+Rs<<2) → @(d:32+Rd<<2)						—	—	↑	↓	0	—
	ORL @(d:16,Rs.W),@(d:32,ERd.L)	L	6	7	9					S							@(d:32+ERd<<2) ∨ @(d:16+Rs<<2) → @(d:32+ERd<<2)						—	—	↑	↓	0	—

Logic operations (75)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹	Addressing Mode										Operation ⁵					Condition Codes ²													
						Number of Execution Stages ¹	Rn	@ERn	@(d,ERn)	@(d,Rn,ERn,W,ERn,L)	@(d,ERn)/@ERn+/@ERn+/@ERn+/@ERn	@aa:R/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C											
OR	ORL @(d:32,Rs,W),@aa:32	L	7	6	1							S	D				@aa:32	∨	@(d:32+Rs<<2)	→	@aa:32				—	—	↑	↓	0	—				
	ORL @(d:32,ERs,L),@ERd	L	5	6	8							D	S				@ERd	∨	@(d:32+ERs<<2)	→	@ERd				—	—	↑	↓	0	—				
	ORL @(d:32,ERs,L),@ERd+	L	5	6	8								D	S				@ERd	∨	@(d:32+ERs<<2)	→	@ERd	ERd32+4→ERd32					—	—	↑	↓	0	—	
	ORL @(d:32,ERs,L),@ERd-	L	5	6	8								D	S				@ERd	∨	@(d:32+ERs<<2)	→	@ERd	ERd32-4→ERd32					—	—	↑	↓	0	—	
	ORL @(d:32,ERs,L),@+ERd	L	5	7	9									D	S				@ERd	∨	@(d:32+ERs<<2)	→	@ERd	ERd32+4→ERd32					—	—	↑	↓	0	—
	ORL @(d:32,ERs,L),@-ERd	L	5	7	9									D	S				@ERd	∨	@(d:32+ERs<<2)	→	@ERd	ERd32-4→ERd32					—	—	↑	↓	0	—
	ORL @(d:32,ERs,L),@(d:2,ERd)	L	5	7	9									D	S				@(4/8/12+ERd)	∨	@(d:32+ERs<<2)	→	@(4/8/12+ERd)					—	—	↑	↓	0	—	
	ORL @(d:32,ERs,L),@(d:16,ERd)	L	6	7	9									D	S				@(d:16+ERd)	∨	@(d:32+ERs<<2)	→	@(d:16+ERd)					—	—	↑	↓	0	—	
	ORL @(d:32,ERs,L),@(d:32,ERd)	L	7	7	1									D	S				@(d:32+ERd)	∨	@(d:32+ERs<<2)	→	@(d:32+ERd)					—	—	↑	↓	0	—	
	ORL @(d:32,ERs,L),@(d:16,Rd,B)	L	6	7	9									S					@(d:16+RdL<<2)	∨	@(d:32+ERs<<2)	→	@(d:16+RdL<<2)					—	—	↑	↓	0	—	
	ORL @(d:32,ERs,L),@(d:16,Rd,W)	L	6	7	9									S					@(d:16+Rd<<2)	∨	@(d:32+ERs<<2)	→	@(d:16+Rd<<2)					—	—	↑	↓	0	—	
	ORL @(d:32,ERs,L),@(d:16,ERdL)	L	6	7	9									S					@(d:16+ERd<<2)	∨	@(d:32+ERs<<2)	→	@(d:16+ERd<<2)					—	—	↑	↓	0	—	
	ORL @(d:32,ERs,L),@(d:32,Rd,B)	L	7	7	1									S					@(d:32+RdL<<2)	∨	@(d:32+ERs<<2)	→	@(d:32+RdL<<2)					—	—	↑	↓	0	—	
	ORL @(d:32,ERs,L),@(d:32,Rd,W)	L	7	7	1									S					@(d:32+Rd<<2)	∨	@(d:32+ERs<<2)	→	@(d:32+Rd<<2)					—	—	↑	↓	0	—	
	ORL @(d:32,ERs,L),@(d:32,ERdL)	L	7	7	1									S					@(d:32+ERd<<2)	∨	@(d:32+ERs<<2)	→	@(d:32+ERd<<2)					—	—	↑	↓	0	—	
	ORL @(d:32,ERs,L),@aa:16	L	6	6	9									S	D				@aa:16	∨	@(d:32+ERs<<2)	→	@aa:16					—	—	↑	↓	0	—	
	ORL @(d:32,ERs,L),@aa:32	L	7	6	1									S	D				@aa:32	∨	@(d:32+ERs<<2)	→	@aa:32					—	—	↑	↓	0	—	
	ORL @aa:16,@ERd	L	4	5	7										D	S			@ERd	∨	@aa:16	→	@ERd					—	—	↑	↓	0	—	
	ORL @aa:16,@ERd+	L	4	5	7											D	S		@ERd	∨	@aa:16	→	@ERd	ERd32+4→ERd32					—	—	↑	↓	0	—
	ORL @aa:16,@ERd-	L	4	5	7											D	S		@ERd	∨	@aa:16	→	@ERd	ERd32-4→ERd32					—	—	↑	↓	0	—
ORL @aa:16,@+ERd	L	4	6	8											D	S		@ERd	∨	@aa:16	→	@ERd	ERd32+4→ERd32					—	—	↑	↓	0	—	
ORL @aa:16,@-ERd	L	4	6	8											D	S		@ERd	∨	@aa:16	→	@ERd	ERd32-4→ERd32					—	—	↑	↓	0	—	
ORL @aa:16,@(d:2,ERd)	L	4	6	8											D	S		@(4/8/12+ERd)	∨	@aa:16	→	@(4/8/12+ERd)					—	—	↑	↓	0	—		
ORL @aa:16,@(d:16,ERd)	L	5	6	8											D	S		@(d:16+ERd)	∨	@aa:16	→	@(d:16+ERd)					—	—	↑	↓	0	—		
ORL @aa:16,@(d:32,ERd)	L	6	6	9											D	S		@(d:32+ERd)	∨	@aa:16	→	@(d:32+ERd)					—	—	↑	↓	0	—		

Logic operations (76)

Instruction	Mnemonic	Size	Instruction Length	Number of 16-bit Instruction Fetch Execution Stages ¹						Addressing Mode								Operation ⁵					Condition Codes ^{3,2}																																																														
				Min.	Max.	Branch	xx	Rn	ERn	(d, ERn)	(d, ERn, W, ERn, L)	ERn/ERn+/ERn-/ERn+/@ERn	ERn+/@ERn	ERn-/@ERn	ERn-/@ERn	ERn+/@ERn	ERn+/@ERn												ERn+/@ERn	ERn+/@ERn																																																							
				Operation 1																Operation 2																Operation 3																Operation 4																Operation 5																I	H
OR	ORL @aa:16, @(d:16, Rd, B)	L	5	6	8					D	S																																																																										
	ORL @aa:16, @(d:16, Rd, W)	L	5	6	8					D	S																																																																										
	ORL @aa:16, @(d:16, ERd, L)	L	5	6	8					D	S																																																																										
	ORL @aa:16, @(d:32, Rd, B)	L	6	6	9					D	S																																																																										
	ORL @aa:16, @(d:32, Rd, W)	L	6	6	9					D	S																																																																										
	ORL @aa:16, @(d:32, ERd, L)	L	6	6	9					D	S																																																																										
	ORL @aa:16, @aa:16	L	5	5	8						S																																																																										
	ORL @aa:16, @aa:32	L	6	5	9						S																																																																										
	ORL @aa:32, @ERd	L	5	5	8					D	S																																																																										
	ORL @aa:32, @ERd+	L	5	5	8					D	S																																																																										
	ORL @aa:32, @ERd-	L	5	5	8					D	S																																																																										
	ORL @aa:32, @+ERd	L	5	6	9					D	S																																																																										
	ORL @aa:32, @-ERd	L	5	6	9					D	S																																																																										
	ORL @aa:32, @(d:2, ERd)	L	5	6	9					D	S																																																																										
	ORL @aa:32, @(d:16, ERd)	L	6	6	9					D	S																																																																										
	ORL @aa:32, @(d:32, ERd)	L	7	6	1					D	S																																																																										
	ORL @aa:32, @(d:16, Rd, B)	L	6	6	9					D	S																																																																										
	ORL @aa:32, @(d:16, Rd, W)	L	6	6	9					D	S																																																																										
	ORL @aa:32, @(d:16, ERd, L)	L	6	6	9					D	S																																																																										
	ORL @aa:32, @(d:32, Rd, B)	L	7	6	1					D	S																																																																										
	ORL @aa:32, @(d:32, Rd, W)	L	7	6	1					D	S																																																																										
	ORL @aa:32, @(d:32, ERd, L)	L	7	6	1					D	S																																																																										
	ORL @aa:32, @aa:16	L	6	5	9						S																																																																										
	ORL @aa:32, @aa:32	L	7	5	1						S																																																																										

Logic operations (77)

Instruction	Mnemonic	Size	Instruction Length	Min.	Addressing Mode											Operation ⁵					Condition Codes ^{5,2}											
					Number of 16-Bit Instruction Fetch Execution Stages ¹	#xx	Rn	@(d,ERn)	@(d,Rn,WRn,L)	@(ERn)/@ERn+/@ERn-/@+ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C										
XOR	XOR.B #xx:8,Rd	B	1	1	1	S	D										Rd8	⊕	#xx	→	Rd8						—	—	↑	↓	0	—
	XOR.B #xx:8,@ERd	B	2	3	4	S	D										@ERd	⊕	#xx	→	@ERd						—	—	↑	↓	0	—
	XOR.B #xx:8,@ERd+	B	3	4	5	S			D								@ERd	⊕	#xx	→	@ERd		ERd32+1→ERd32			—	—	↑	↓	0	—	
	XOR.B #xx:8,@ERd-	B	3	4	5	S			D								@ERd	⊕	#xx	→	@ERd		ERd32-1→ERd32			—	—	↑	↓	0	—	
	XOR.B #xx:8,@+ERd	B	3	5	5	S			D							ERd32+1→ERd32	@ERd	⊕	#xx	→	@ERd					—	—	↑	↓	0	—	
	XOR.B #xx:8,@-ERd	B	3	5	5	S			D						ERd32-1→ERd32	@ERd	⊕	#xx	→	@ERd						—	—	↑	↓	0	—	
	XOR.B #xx:8,@(d:2,ERd)	B	3	5	5	S			D								@(1/2/3+ERd)	⊕	#xx	→	@(1/2/3+ERd)					—	—	↑	↓	0	—	
	XOR.B #xx:8,@(d:16,ERd)	B	4	5	6	S			D								@(d:16+ERd)	⊕	#xx	→	@(d:16+ERd)					—	—	↑	↓	0	—	
	XOR.B #xx:8,@(d:32,ERd)	B	5	5	7	S			D								@(d:32+ERd)	⊕	#xx	→	@(d:32+ERd)					—	—	↑	↓	0	—	
	XOR.B #xx:8,@(d:16,Rd,B)	B	4	5	6	S			D								@(d:16+RdL)	⊕	#xx	→	@(d:16+RdL)					—	—	↑	↓	0	—	
	XOR.B #xx:8,@(d:16,Rd,W)	B	4	5	6	S			D								@(d:16+Rd)	⊕	#xx	→	@(d:16+Rd)					—	—	↑	↓	0	—	
	XOR.B #xx:8,@(d:16,ERd,L)	B	4	5	6	S			D								@(d:16+ERd)	⊕	#xx	→	@(d:16+ERd)					—	—	↑	↓	0	—	
	XOR.B #xx:8,@(d:32,Rd,B)	B	5	5	7	S			D								@(d:32+RdL)	⊕	#xx	→	@(d:32+RdL)					—	—	↑	↓	0	—	
	XOR.B #xx:8,@(d:32,Rd,W)	B	5	5	7	S			D								@(d:32+Rd)	⊕	#xx	→	@(d:32+Rd)					—	—	↑	↓	0	—	
	XOR.B #xx:8,@(d:32,ERd,L)	B	5	5	7	S			D								@(d:32+ERd)	⊕	#xx	→	@(d:32+ERd)					—	—	↑	↓	0	—	
	XOR.B #xx:8,@aa:8	B	2	3	4	S					D						@aa:8	⊕	#xx	→	@aa:8					—	—	↑	↓	0	—	
	XOR.B #xx:8,@aa:16	B	3	3	5	S					D						@aa:16	⊕	#xx	→	@aa:16					—	—	↑	↓	0	—	
	XOR.B #xx:8,@aa:32	B	4	3	6	S					D						@aa:32	⊕	#xx	→	@aa:32					—	—	↑	↓	0	—	
	XOR.B Rs,Rd	B	1	1	1	S											Rd8	⊕	Rs8	→	Rd8					—	—	↑	↓	0	—	
	XOR.B Rs,@ERd	B	2	3	4	S	D										@ERd	⊕	Rs8	→	@ERd					—	—	↑	↓	0	—	
	XOR.B Rs,@ERd+	B	2	4	4	S				D							@ERd	⊕	Rs8	→	@ERd		ERd32+1→ERd32			—	—	↑	↓	0	—	
	XOR.B Rs,@ERd-	B	2	4	4	S				D							@ERd	⊕	Rs8	→	@ERd		ERd32-1→ERd32			—	—	↑	↓	0	—	
	XOR.B Rs,@+ERd	B	2	5	5	S				D						ERd32+1→ERd32	@ERd	⊕	Rs8	→	@ERd					—	—	↑	↓	0	—	
	XOR.B Rs,@-ERd	B	2	5	5	S				D					ERd32-1→ERd32	@ERd	⊕	Rs8	→	@ERd						—	—	↑	↓	0	—	
	XOR.B Rs,@(d:2,ERd)	B	2	5	5	S		D									@(1/2/3+ERd)	⊕	Rs8	→	@(1/2/3+ERd)					—	—	↑	↓	0	—	

Logic operations (78)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Status ¹	Addressing Mode								Operation ⁵					Condition Codes ²					
						Number of Execution Stages ³	Fetch	Execute	Write Back	Read	Write Back	Read	Write Back	Read	Write Back	Read	Write Back	Read	Write Back	Read	Write Back	I	H	N
n						Rn	@ERn	@(d,ERn)	@(d,Rn,WRn,L)	@ERn/@ERn+/@ERn+/@+ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5								
XOR	XOR.B Rs, @(d:16,ERd)	B	3	5	5	S		D						@(d:16+ERd) ⊕ Rs8 → @(d:16+ERd)			—	—	↑	↓	0	—		
	XOR.B Rs, @(d:32,ERd)	B	4	5	6	S		D						@(d:32+ERd) ⊕ Rs8 → @(d:32+ERd)			—	—	↑	↓	0	—		
	XOR.B Rs, @(d:16,Rd.B)	B	3	5	5	S			D					@(d:16+RdL) ⊕ Rs8 → @(d:16+RdL)			—	—	↑	↓	0	—		
	XOR.B Rs, @(d:16,Rd.W)	B	3	5	5	S			D					@(d:16+Rd) ⊕ Rs8 → @(d:16+Rd)			—	—	↑	↓	0	—		
	XOR.B Rs, @(d:16,ERd.L)	B	3	5	5	S			D					@(d:16+ERd) ⊕ Rs8 → @(d:16+ERd)			—	—	↑	↓	0	—		
	XOR.B Rs, @(d:32,Rd.B)	B	4	5	6	S			D					@(d:32+RdL) ⊕ Rs8 → @(d:32+RdL)			—	—	↑	↓	0	—		
	XOR.B Rs, @(d:32,Rd.W)	B	4	5	6	S			D					@(d:32+Rd) ⊕ Rs8 → @(d:32+Rd)			—	—	↑	↓	0	—		
	XOR.B Rs, @(d:32,ERd.L)	B	4	5	6	S			D					@(d:32+ERd) ⊕ Rs8 → @(d:32+ERd)			—	—	↑	↓	0	—		
	XOR.B Rs, @aa:8	B	2	3	4	S				D				@aa:8 ⊕ Rs8 → @aa:8			—	—	↑	↓	0	—		
	XOR.B Rs, @aa:16	B	3	3	5	S				D				@aa:16 ⊕ Rs8 → @aa:16			—	—	↑	↓	0	—		
	XOR.B Rs, @aa:32	B	4	3	6	S				D				@aa:32 ⊕ Rs8 → @aa:32			—	—	↑	↓	0	—		
	XOR.B @ERs,Rd	B	2	3	3	D	S							Rd8 ⊕ @ERs → Rd8			—	—	↑	↓	0	—		
	XOR.B @ERs+,Rd	B	2	3	3	D			S					Rd8 ⊕ @ERs → Rd8	ERs32+1→ERs32		—	—	↑	↓	0	—		
	XOR.B @ERs-,Rd	B	2	3	3	D			S					Rd8 ⊕ @ERs → Rd8	ERs32-1→ERs32		—	—	↑	↓	0	—		
	XOR.B @+ERs,Rd	B	2	4	4	D			S		ERs32+1→ERs32			Rd8 ⊕ @ERs → Rd8			—	—	↑	↓	0	—		
	XOR.B @-ERs,Rd	B	2	4	4	D			S		ERs32-1→ERs32			Rd8 ⊕ @ERs → Rd8			—	—	↑	↓	0	—		
	XOR.B @(d:2,ERs),Rd	B	2	4	4	D	S							Rd8 ⊕ @(1/2/3+ERs) → Rd8			—	—	↑	↓	0	—		
	XOR.B @(d:16,ERs),Rd	B	3	4	4	D	S							Rd8 ⊕ @(d:16+ERs) → Rd8			—	—	↑	↓	0	—		
	XOR.B @(d:32,ERs),Rd	B	4	4	5	D	S							Rd8 ⊕ @(d:32+ERs) → Rd8			—	—	↑	↓	0	—		
	XOR.B @(d:16,Rs.B),Rd	B	3	4	4	D		S						Rd8 ⊕ @(d:16+RsL) → Rd8			—	—	↑	↓	0	—		
	XOR.B @(d:16,Rs.W),Rd	B	3	4	4	D		S						Rd8 ⊕ @(d:16+Rs) → Rd8			—	—	↑	↓	0	—		
	XOR.B @(d:16,ERs.L),Rd	B	3	4	4	D		S						Rd8 ⊕ @(d:16+ERs) → Rd8			—	—	↑	↓	0	—		
	XOR.B @(d:32,Rs.B),Rd	B	4	4	5	D		S						Rd8 ⊕ @(d:32+RsL) → Rd8			—	—	↑	↓	0	—		
	XOR.B @(d:32,Rs.W),Rd	B	4	4	5	D		S						Rd8 ⊕ @(d:32+Rs) → Rd8			—	—	↑	↓	0	—		
	XOR.B @(d:32,ERs.L),Rd	B	4	4	5	D		S						Rd8 ⊕ @(d:32+ERs) → Rd8			—	—	↑	↓	0	—		



Logic operations (83)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	Addressing Mode										Operation ⁵					Condition Codes ^{6,7}					
						Number of 16-Bit Instruction Fetch Execution Stages ¹	Fetch	Execute	Write Back	Write Back	Write Back	Write Back	Write Back	Write Back	Write Back											
n							Rn	@ERn	@(d,ERn)	@(d,Rn,WRn,L)	@ERn/@ERn+/@ERn+/@ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C			
XOR	XOR.B @(d:16,ERs),@ERd+	B	4	6	7			S	D						@ERd @ @(d:16+ERs) → @ERd		ERd32+1→ERd32									
	XOR.B @(d:16,ERs),@ERd-	B	4	6	7			S	D						@ERd @ @(d:16+ERs) → @ERd		ERd32-1→ERd32									
	XOR.B @(d:16,ERs),@+ERd	B	4	7	8			S	D				ERd32+1→ERd32	@ERd @ @(d:16+ERs) → @ERd												
	XOR.B @(d:16,ERs),@-ERd	B	4	7	8			S	D				ERd32-1→ERd32	@ERd @ @(d:16+ERs) → @ERd												
	XOR.B @(d:16,ERs),@(d:2,ERd)	B	4	7	8			S						@(1/2/3+ERd) @ @(d:16+ERs) → @(1/2/3+ERd)												
	XOR.B @(d:16,ERs),@(d:16,ERd)	B	5	7	8			S						@(d:16+ERd) @ @(d:16+ERs) → @(d:16+ERd)												
	XOR.B @(d:16,ERs),@(d:32,ERd)	B	6	7	9			S						@(d:32+ERd) @ @(d:16+ERs) → @(d:32+ERd)												
	XOR.B @(d:16,ERs),@(d:16,Rd,B)	B	5	7	8			S	D					@(d:16+RdL) @ @(d:16+ERs) → @(d:16+RdL)												
	XOR.B @(d:16,ERs),@(d:16,Rd,W)	B	5	7	8			S	D					@(d:16+Rd) @ @(d:16+ERs) → @(d:16+Rd)												
	XOR.B @(d:16,ERs),@(d:16,ERdL)	B	5	7	8			S	D					@(d:16+ERd) @ @(d:16+ERs) → @(d:16+ERd)												
	XOR.B @(d:16,ERs),@(d:32,Rd,B)	B	6	7	9			S	D					@(d:32+RdL) @ @(d:16+ERs) → @(d:32+RdL)												
	XOR.B @(d:16,ERs),@(d:32,Rd,W)	B	6	7	9			S	D					@(d:32+Rd) @ @(d:16+ERs) → @(d:32+Rd)												
	XOR.B @(d:16,ERs),@(d:32,ERdL)	B	6	7	9			S	D					@(d:32+ERd) @ @(d:16+ERs) → @(d:32+ERd)												
	XOR.B @(d:16,ERs),@aa:16	B	5	6	8			S			D			@aa:16 @ @(d:16+ERs) → @aa:16												
	XOR.B @(d:16,ERs),@aa:32	B	6	6	9			S			D			@aa:32 @ @(d:16+ERs) → @aa:32												
	XOR.B @(d:32,ERs),@ERd	B	5	6	8			D	S					@ERd @ @(d:32+ERs) → @ERd												
	XOR.B @(d:32,ERs),@ERd+	B	5	6	8			S	D					@ERd @ @(d:32+ERs) → @ERd			ERd32+1→ERd32									
	XOR.B @(d:32,ERs),@ERd-	B	5	6	8			S	D					@ERd @ @(d:32+ERs) → @ERd			ERd32-1→ERd32									
	XOR.B @(d:32,ERs),@+ERd	B	5	7	9			S	D				ERd32+1→ERd32	@ERd @ @(d:32+ERs) → @ERd												
	XOR.B @(d:32,ERs),@-ERd	B	5	7	9			S	D				ERd32-1→ERd32	@ERd @ @(d:32+ERs) → @ERd												
	XOR.B @(d:32,ERs),@(d:2,ERd)	B	5	7	9			S						@(1/2/3+ERd) @ @(d:32+ERs) → @(1/2/3+ERd)												
	XOR.B @(d:32,ERs),@(d:16,ERd)	B	6	7	9			S						@(d:16+ERd) @ @(d:32+ERs) → @(d:16+ERd)												
	XOR.B @(d:32,ERs),@(d:32,ERd)	B	7	7	10			S						@(d:32+ERd) @ @(d:32+ERs) → @(d:32+ERd)												
	XOR.B @(d:32,ERs),@(d:16,Rd,B)	B	6	7	9			S	D					@(d:16+RdL) @ @(d:32+ERs) → @(d:16+RdL)												
	XOR.B @(d:32,ERs),@(d:16,Rd,W)	B	6	7	9			S	D					@(d:16+Rd) @ @(d:32+ERs) → @(d:16+Rd)												

Logic operations (92)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹ #Fxx	Addressing Mode								Operation ⁵					Condition Codes ²													
						Rn	@(d,ERn)	@(d,Rn,WRn,L)	@(d,ERn)/@ERn+/@ERn+/@ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C											
XOR	XOR.W @ERs+, @ERd	W	3	5	6		D							@ERd	⊕	@ERs	→	@ERd	ERs32+2→ERs32			—	—	↑	↓	0	—					
	XOR.W @ERs+, @ERd+	W	3	5	6									@ERd	⊕	@ERs	→	@ERd	ERs32+2→ERs32	ERd32+2→ERd32			—	—	↑	↓	0	—				
	XOR.W @ERs+, @ERd-	W	3	5	6									@ERd	⊕	@ERs	→	@ERd	ERs32+2→ERs32	ERd32+2→ERd32			—	—	↑	↓	0	—				
	XOR.W @ERs+, @+ERd	W	3	6	7										ERd32+2→ERd32	@ERd	⊕	@ERs	→	@ERd	ERs32+2→ERs32			—	—	↑	↓	0	—			
	XOR.W @ERs+, @-ERd	W	3	6	7										ERd32-2→ERd32	@ERd	⊕	@ERs	→	@ERd	ERs32+2→ERs32			—	—	↑	↓	0	—			
	XOR.W @ERs+, @(d:2,ERd)	W	3	6	7									D	S		@(2/4/6+ERd)	⊕	@ERs	→	@(2/4/6+ERd)	ERs32+2→ERs32			—	—	↑	↓	0	—		
	XOR.W @ERs+, @(d:16,ERd)	W	4	6	7									D	S		@(d:16+ERd)	⊕	@ERs	→	@(d:16+ERd)	ERs32+2→ERs32			—	—	↑	↓	0	—		
	XOR.W @ERs+, @(d:32,ERd)	W	5	6	8									D	S		@(d:32+ERd)	⊕	@ERs	→	@(d:32+ERd)	ERs32+2→ERs32			—	—	↑	↓	0	—		
	XOR.W @ERs+, @(d:16,Rd.B)	W	4	6	7									D	S		@(d:16+RdL<<1)	⊕	@ERs	→	@(d:16+RdL<<1)	ERs32+2→ERs32			—	—	↑	↓	0	—		
	XOR.W @ERs+, @(d:16,Rd.W)	W	4	6	7									D	S		@(d:16+Rd<<1)	⊕	@ERs	→	@(d:16+Rd<<1)	ERs32+2→ERs32			—	—	↑	↓	0	—		
	XOR.W @ERs+, @(d:16,ERd.L)	W	4	6	7									D	S		@(d:16+ERd<<1)	⊕	@ERs	→	@(d:16+ERd<<1)	ERs32+2→ERs32			—	—	↑	↓	0	—		
	XOR.W @ERs+, @(d:32,Rd.B)	W	5	6	8									D	S		@(d:32+RdL<<1)	⊕	@ERs	→	@(d:32+RdL<<1)	ERs32+2→ERs32			—	—	↑	↓	0	—		
	XOR.W @ERs+, @(d:32,Rd.W)	W	5	6	8									D	S		@(d:32+Rd<<1)	⊕	@ERs	→	@(d:32+Rd<<1)	ERs32+2→ERs32			—	—	↑	↓	0	—		
	XOR.W @ERs+, @(d:32,ERd.L)	W	5	6	8									D	S		@(d:32+ERd<<1)	⊕	@ERs	→	@(d:32+ERd<<1)	ERs32+2→ERs32			—	—	↑	↓	0	—		
	XOR.W @ERs+, @aa:16	W	4	5	7									S	D		@aa:16	⊕	@ERs	→	@aa:16	ERs32+2→ERs32			—	—	↑	↓	0	—		
	XOR.W @ERs+, @aa:32	W	5	5	8									S	D		@aa:32	⊕	@ERs	→	@aa:32	ERs32+2→ERs32			—	—	↑	↓	0	—		
	XOR.W @ERs-, @ERd	W	3	5	6									D	S		@ERd	⊕	@ERs	→	@ERd	ERs32-2→ERs32			—	—	↑	↓	0	—		
	XOR.W @ERs-, @ERd+	W	3	5	6												@ERd	⊕	@ERs	→	@ERd	ERs32-2→ERs32	ERd32+2→ERd32			—	—	↑	↓	0	—	
	XOR.W @ERs-, @ERd-	W	3	5	6												@ERd	⊕	@ERs	→	@ERd	ERs32-2→ERs32	ERd32-2→ERd32			—	—	↑	↓	0	—	
	XOR.W @ERs-, @+ERd	W	3	6	7													ERd32+2→ERd32	@ERd	⊕	@ERs	→	@ERd	ERs32-2→ERs32			—	—	↑	↓	0	—
	XOR.W @ERs-, @-ERd	W	3	6	7													ERd32-2→ERd32	@ERd	⊕	@ERs	→	@ERd	ERs32-2→ERs32			—	—	↑	↓	0	—
	XOR.W @ERs-, @(d:2,ERd)	W	3	6	7										D	S		@(2/4/6+ERd)	⊕	@ERs	→	@(2/4/6+ERd)	ERs32-2→ERs32			—	—	↑	↓	0	—	
	XOR.W @ERs-, @(d:16,ERd)	W	4	7	7									D	S		@(d:16+ERd)	⊕	@ERs	→	@(d:16+ERd)	ERs32-2→ERs32			—	—	↑	↓	0	—		
	XOR.W @ERs-, @(d:32,ERd)	W	5	7	8									D	S		@(d:32+ERd)	⊕	@ERs	→	@(d:32+ERd)	ERs32-2→ERs32			—	—	↑	↓	0	—		
XOR.W @ERs-, @(d:16,Rd.B)	W	4	7	7									D	S		@(d:16+RdL<<1)	⊕	@ERs	→	@(d:16+RdL<<1)	ERs32-2→ERs32			—	—	↑	↓	0	—			



Logic operations (93)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode										Operation ⁵					Condition Codes ^{6,7}							
				Min.	16-Bit Instruction Fetch	Fetch	Execution Status ⁸	Prx	Rn	@ERn	@(d,ERn)	@(d,Rn,ERn,WERn,L)	@ERn/@ERn+/@ERn+/@ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C	
XOR	XOR.W @ERs-, @(d:16,Rd,W)	W	4	7	7							D	S				@(d:16+Rd<<1) ⊕ @ERs → @(d:16+Rd<<1)	ERs32-2→ERs32				↑	↓	0	—	
	XOR.W @ERs-, @(d:16,ERd,L)	W	4	7	7							D	S				@(d:16+ERd<<1) ⊕ @ERs → @(d:16+ERd<<1)	ERs32-2→ERs32				↑	↓	0	—	
	XOR.W @ERs-, @(d:32,Rd,B)	W	5	7	8							D	S				@(d:32+RdL<<1) ⊕ @ERs → @(d:32+RdL<<1)	ERs32-2→ERs32				↑	↓	0	—	
	XOR.W @ERs-, @(d:32,Rd,W)	W	5	7	8							D	S				@(d:32+Rd<<1) ⊕ @ERs → @(d:32+Rd<<1)	ERs32-2→ERs32				↑	↓	0	—	
	XOR.W @ERs-, @(d:32,ERd,L)	W	5	7	8							D	S				@(d:32+ERd<<1) ⊕ @ERs → @(d:32+ERd<<1)	ERs32-2→ERs32				↑	↓	0	—	
	XOR.W @ERs-, @aa:16	W	4	5	7								S	D				@aa:16 ⊕ @ERs → @aa:16	ERs32-2→ERs32				↑	↓	0	—
	XOR.W @ERs-, @aa:32	W	5	5	8								S	D				@aa:32 ⊕ @ERs → @aa:32	ERs32-2→ERs32				↑	↓	0	—
	XOR.W @+ERs, @ERd	W	3	6	6							D	S		S	ERs32+2→ERs32	@ERd ⊕ @ERs → @ERd					↑	↓	0	—	
	XOR.W @+ERs, @ERd+	W	3	6	6								S		S	ERs32+2→ERs32	@ERd ⊕ @ERs → @ERd	ERd32+2→ERd32				↑	↓	0	—	
	XOR.W @+ERs, @ERd-	W	3	6	6								S		S	ERs32+2→ERs32	@ERd ⊕ @ERs → @ERd	ERd32-2→ERd32				↑	↓	0	—	
	XOR.W @+ERs, @+ERd	W	3	7	7								S		S	ERs32+2→ERs32	ERd32+2→ERd32	@ERd ⊕ @ERs → @ERd				↑	↓	0	—	
	XOR.W @+ERs, @-ERd	W	3	7	7								S		S	ERs32+2→ERs32	ERd32-2→ERd32	@ERd ⊕ @ERs → @ERd				↑	↓	0	—	
	XOR.W @+ERs, @(d:2,ERd)	W	3	7	7							D	S		S	ERs32+2→ERs32	@(2/4/6+ERd) ⊕ @ERs → @(2/4/6+ERd)				↑	↓	0	—		
	XOR.W @+ERs, @(d:16,ERd)	W	4	7	7							D	S		S	ERs32+2→ERs32	@(d:16+ERd) ⊕ @ERs → @(d:16+ERd)				↑	↓	0	—		
	XOR.W @+ERs, @(d:32,ERd)	W	5	7	8							D	S		S	ERs32+2→ERs32	@(d:32+ERd) ⊕ @ERs → @(d:32+ERd)				↑	↓	0	—		
	XOR.W @+ERs, @(d:16,Rd,B)	W	4	7	7							D	S		S	ERs32+2→ERs32	@(d:16+RdL<<1) ⊕ @ERs → @(d:16+RdL<<1)				↑	↓	0	—		
	XOR.W @+ERs, @(d:16,Rd,W)	W	4	7	7							D	S		S	ERs32+2→ERs32	@(d:16+Rd<<1) ⊕ @ERs → @(d:16+Rd<<1)				↑	↓	0	—		
	XOR.W @+ERs, @(d:16,ERd,L)	W	4	7	7							D	S		S	ERs32+2→ERs32	@(d:16+ERd<<1) ⊕ @ERs → @(d:16+ERd<<1)				↑	↓	0	—		
	XOR.W @+ERs, @(d:32,Rd,B)	W	5	7	8							D	S		S	ERs32+2→ERs32	@(d:32+RdL<<1) ⊕ @ERs → @(d:32+RdL<<1)				↑	↓	0	—		
	XOR.W @+ERs, @(d:32,Rd,W)	W	5	7	8							D	S		S	ERs32+2→ERs32	@(d:32+Rd<<1) ⊕ @ERs → @(d:32+Rd<<1)				↑	↓	0	—		
	XOR.W @+ERs, @(d:32,ERd,L)	W	5	7	8							D	S		S	ERs32+2→ERs32	@(d:32+ERd<<1) ⊕ @ERs → @(d:32+ERd<<1)				↑	↓	0	—		
	XOR.W @+ERs, @aa:16	W	4	6	7							S	D		S	ERs32+2→ERs32	@aa:16 ⊕ @ERs → @aa:16				↑	↓	0	—		
	XOR.W @+ERs, @aa:32	W	5	6	8							S	D		S	ERs32+2→ERs32	@aa:32 ⊕ @ERs → @aa:32				↑	↓	0	—		
XOR.W @-ERs, @ERd	W	3	6	6							D	S		S	ERs32-2→ERs32	@ERd ⊕ @ERs → @ERd				↑	↓	0	—			
XOR.W @-ERs, @ERd+	W	3	6	6							S		S	ERs32-2→ERs32	@ERd ⊕ @ERs → @ERd	ERd32+2→ERd32				↑	↓	0	—			

Logic operations (94)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹	Addressing Mode							Operation ⁵					Condition Codes ²									
						Number of Execution Stages ¹	Rn	@(d,ERn)	@(d,Rn,WRn,L)	@(Rn,ERn)/@ERn/ ² @+ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C					
XOR	XOR.W @-ERs,@ERd-	W	3	6	6						S			ERs32-2→ERs32		@ERd	@ @ERs	→ @ERd			ERd32-2→ERd32	—	—	↑	↓	0	—
	XOR.W @-ERs,@+ERd	W	3	7	7						S			ERs32-2→ERs32	ERd32+2→ERd32	@ERd	@ @ERs	→ @ERd				—	—	↑	↓	0	—
	XOR.W @-ERs,@-ERd	W	3	7	7						S			ERs32-2→ERs32	ERd32-2→ERd32	@ERd	@ @ERs	→ @ERd				—	—	↑	↓	0	—
	XOR.W @-ERs,@(d:2,ERd)	W	3	7	7					D	S			ERs32-2→ERs32		@(2/4/6+ERd)	@ @ERs	→ @(2/4/6+ERd)				—	—	↑	↓	0	—
	XOR.W @-ERs,@(d:16,ERd)	W	4	7	7					D	S			ERs32-2→ERs32		@(d:16+ERd)	@ @ERs	→ @(d:16+ERd)				—	—	↑	↓	0	—
	XOR.W @-ERs,@(d:32,ERd)	W	5	7	8					D	S			ERs32-2→ERs32		@(d:32+ERd)	@ @ERs	→ @(d:32+ERd)				—	—	↑	↓	0	—
	XOR.W @-ERs,@(d:16,Rd,B)	W	4	7	7					D	S			ERs32-2→ERs32		@(d:16+Rd<<1)	@ @ERs	→ @(d:16+Rd<<1)				—	—	↑	↓	0	—
	XOR.W @-ERs,@(d:16,Rd,W)	W	4	7	7					D	S			ERs32-2→ERs32		@(d:16+Rd<<1)	@ @ERs	→ @(d:16+Rd<<1)				—	—	↑	↓	0	—
	XOR.W @-ERs,@(d:16,ERd,L)	W	4	7	7					D	S			ERs32-2→ERs32		@(d:16+ERd<<1)	@ @ERs	→ @(d:16+ERd<<1)				—	—	↑	↓	0	—
	XOR.W @-ERs,@(d:32,Rd,B)	W	5	7	8					D	S			ERs32-2→ERs32		@(d:32+Rd<<1)	@ @ERs	→ @(d:32+Rd<<1)				—	—	↑	↓	0	—
	XOR.W @-ERs,@(d:32,Rd,W)	W	5	7	8					D	S			ERs32-2→ERs32		@(d:32+Rd<<1)	@ @ERs	→ @(d:32+Rd<<1)				—	—	↑	↓	0	—
	XOR.W @-ERs,@(d:32,ERd,L)	W	5	7	8					D	S			ERs32-2→ERs32		@(d:32+ERd<<1)	@ @ERs	→ @(d:32+ERd<<1)				—	—	↑	↓	0	—
	XOR.W @-ERs,@aa:16	W	4	6	7						S	D		ERs32-2→ERs32		@aa:16	@ @ERs	→ @aa:16				—	—	↑	↓	0	—
	XOR.W @-ERs,@aa:32	W	5	6	8						S	D		ERs32-2→ERs32		@aa:32	@ @ERs	→ @aa:32				—	—	↑	↓	0	—
	XOR.W @(d:2,ERs),@ERd	W	3	6	6					D	S					@ERd	@ @(2/4/6+ERs)	→ @ERd				—	—	↑	↓	0	—
	XOR.W @(d:2,ERs),@ERd+	W	3	6	6						S	D				@ERd	@ @(2/4/6+ERs)	→ @ERd			ERd32+2→ERd32	—	—	↑	↓	0	—
	XOR.W @(d:2,ERs),@ERd-	W	3	6	6						S	D				@ERd	@ @(2/4/6+ERs)	→ @ERd			ERd32-2→ERd32	—	—	↑	↓	0	—
	XOR.W @(d:2,ERs),@+ERd	W	3	7	7						S	D			ERd32+2→ERd32	@ERd	@ @(2/4/6+ERs)	→ @ERd				—	—	↑	↓	0	—
	XOR.W @(d:2,ERs),@-ERd	W	3	7	7						S	D			ERd32-2→ERd32	@ERd	@ @(2/4/6+ERs)	→ @ERd				—	—	↑	↓	0	—
	XOR.W @(d:2,ERs),@(d:2,ERd)	W	3	7	7						S					@(2/4/6+ERd)	@ @(2/4/6+ERs)	→ @(2/4/6+ERd)				—	—	↑	↓	0	—
	XOR.W @(d:2,ERs),@(d:16,ERd)	W	4	7	7						S					@(d:16+ERd)	@ @(2/4/6+ERs)	→ @(d:16+ERd)				—	—	↑	↓	0	—
	XOR.W @(d:2,ERs),@(d:32,ERd)	W	5	7	8						S					@(d:32+ERd)	@ @(2/4/6+ERs)	→ @(d:32+ERd)				—	—	↑	↓	0	—
	XOR.W @(d:2,ERs),@(d:16,Rd,B)	W	4	7	7						S	D				@(d:16+Rd<<1)	@ @(2/4/6+ERs)	→ @(d:16+Rd<<1)				—	—	↑	↓	0	—
	XOR.W @(d:2,ERs),@(d:16,Rd,W)	W	4	7	7						S	D				@(d:16+Rd<<1)	@ @(2/4/6+ERs)	→ @(d:16+Rd<<1)				—	—	↑	↓	0	—
	XOR.W @(d:2,ERs),@(d:16,ERd,L)	W	4	7	7						S	D				@(d:16+ERd<<1)	@ @(2/4/6+ERs)	→ @(d:16+ERd<<1)				—	—	↑	↓	0	—



Logic operations (98)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹	Addressing Mode										Operation ⁵					Condition Codes ^{2,3}					
						Number of	Fetch	Execution	Stages ¹	Op	Rn	@(d,ERn)	@(d,Rn,WRn,LRn)	@(d,ERn)/@ERn+/@ERn-/@+ERn	@aa:8/@aa:16/@aa:32											
n						Op	Rn	@(d,ERn)	@(d,Rn,WRn,LRn)	@(d,ERn)/@ERn+/@ERn-/@+ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C				
XOR	XOR.W @(d:16,ERs.L),@(d:2,ERd)	W	4	7	8			D	S					@(2/4/6+ERd) ⊕ @(d:16+ERs<<1) → @(2/4/6+ERd)			—	—	↑	↓	0	—				
	XOR.W @(d:16,ERs.L),@(d:16,ERd)	W	5	7	8			D	S					@(d:16+ERd) ⊕ @(d:16+ERs<<1) → @(d:16+ERd)			—	—	↑	↓	0	—				
	XOR.W @(d:16,ERs.L),@(d:32,ERd)	W	6	7	9			D	S					@(d:32+ERd) ⊕ @(d:16+ERs<<1) → @(d:32+ERd)			—	—	↑	↓	0	—				
	XOR.W @(d:16,ERs.L),@(d:16,Rd.B)	W	5	7	8				S					@(d:16+RdL<<1) ⊕ @(d:16+ERs<<1) → @(d:16+RdL<<1)			—	—	↑	↓	0	—				
	XOR.W @(d:16,ERs.L),@(d:16,Rd.W)	W	5	7	8				S					@(d:16+Rd<<1) ⊕ @(d:16+ERs<<1) → @(d:16+Rd<<1)			—	—	↑	↓	0	—				
	XOR.W @(d:16,ERs.L),@(d:16,ERd.L)	W	5	7	8				S					@(d:16+ERd<<1) ⊕ @(d:16+ERs<<1) → @(d:16+ERd<<1)			—	—	↑	↓	0	—				
	XOR.W @(d:16,ERs.L),@(d:32,Rd.B)	W	6	7	9				S					@(d:32+RdL<<1) ⊕ @(d:16+ERs<<1) → @(d:32+RdL<<1)			—	—	↑	↓	0	—				
	XOR.W @(d:16,ERs.L),@(d:32,Rd.W)	W	6	7	9				S					@(d:32+Rd<<1) ⊕ @(d:16+ERs<<1) → @(d:32+Rd<<1)			—	—	↑	↓	0	—				
	XOR.W @(d:16,ERs.L),@(d:32,ERd.L)	W	6	7	9				S					@(d:32+ERd<<1) ⊕ @(d:16+ERs<<1) → @(d:32+ERd<<1)			—	—	↑	↓	0	—				
	XOR.W @(d:16,ERs.L),@aa:16	W	5	6	8				S	D				@aa:16 ⊕ @(d:16+ERs<<1) → @aa:16			—	—	↑	↓	0	—				
	XOR.W @(d:16,ERs.L),@aa:32	W	6	6	9				S	D				@aa:32 ⊕ @(d:16+ERs<<1) → @aa:32			—	—	↑	↓	0	—				
	XOR.W @(d:32,Rs.B),@ERd	W	5	6	8			D	S					@ERd ⊕ @(d:16+RsL<<1) → @ERd			—	—	↑	↓	0	—				
	XOR.W @(d:32,Rs.B),@ERd+	W	5	6	8				S	D				@ERd ⊕ @(d:16+RsL<<1) → @ERd	ERd32+2→ERd32		—	—	↑	↓	0	—				
	XOR.W @(d:32,Rs.B),@ERd-	W	5	6	8				S	D				@ERd ⊕ @(d:16+RsL<<1) → @ERd	ERd32-2→ERd32		—	—	↑	↓	0	—				
	XOR.W @(d:32,Rs.B),@+ERd	W	5	7	9				S	D		ERd32+2→ERd32	@ERd	@(d:16+RsL<<1) → @ERd			—	—	↑	↓	0	—				
	XOR.W @(d:32,Rs.B),@-ERd	W	5	7	9				S	D		ERd32-2→ERd32	@ERd	@(d:16+RsL<<1) → @ERd			—	—	↑	↓	0	—				
	XOR.W @(d:32,Rs.B),@(d:2,ERd)	W	5	7	9			D	S					@(2/4/6+ERd) ⊕ @(d:16+RsL<<1) → @(2/4/6+ERd)			—	—	↑	↓	0	—				
	XOR.W @(d:32,Rs.B),@(d:16,ERd)	W	6	7	9			D	S					@(d:16+ERd) ⊕ @(d:16+RsL<<1) → @(d:16+ERd)			—	—	↑	↓	0	—				
	XOR.W @(d:32,Rs.B),@(d:32,ERd)	W	7	7	10			D	S					@(d:32+ERd) ⊕ @(d:16+RsL<<1) → @(d:32+ERd)			—	—	↑	↓	0	—				
	XOR.W @(d:32,Rs.B),@(d:16,Rd.B)	W	6	7	9				S					@(d:16+RdL<<1) ⊕ @(d:16+RsL<<1) → @(d:16+RdL<<1)			—	—	↑	↓	0	—				
	XOR.W @(d:32,Rs.B),@(d:16,Rd.W)	W	6	7	9				S					@(d:16+Rd<<1) ⊕ @(d:16+RsL<<1) → @(d:16+Rd<<1)			—	—	↑	↓	0	—				
	XOR.W @(d:32,Rs.B),@(d:16,ERd.L)	W	6	7	9				S					@(d:16+ERd<<1) ⊕ @(d:16+RsL<<1) → @(d:16+ERd<<1)			—	—	↑	↓	0	—				
	XOR.W @(d:32,Rs.B),@(d:32,Rd.B)	W	7	7	10				S					@(d:32+RdL<<1) ⊕ @(d:16+RsL<<1) → @(d:32+RdL<<1)			—	—	↑	↓	0	—				
	XOR.W @(d:32,Rs.B),@(d:32,Rd.W)	W	7	7	10				S					@(d:32+Rd<<1) ⊕ @(d:16+RsL<<1) → @(d:32+Rd<<1)			—	—	↑	↓	0	—				
	XOR.W @(d:32,Rs.B),@(d:32,ERd.L)	W	7	7	10				S					@(d:32+ERd<<1) ⊕ @(d:32+RsL<<1) → @(d:32+ERd<<1)			—	—	↑	↓	0	—				



Logic operations (101)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹	Addressing Mode							Operation ⁵					Condition Codes ²						
						Number of Execution Stages ¹	Rn	@(d,ERn)	@(d,Rn,ERn,WERn,L)	@(d,ERn)/@ERn+/@ERn+/@ERn+/@ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C		
XOR	XOR.W @aa:32,@ERd	W	4	4	7			D				S			@ERd @ aa:32 → @ERd				—	—	↑	↓	0	—
	XOR.W @aa:32,@ERd+	W	4	4	7						D	S			@ERd @ aa:32 → @ERd				—	—	↑	↓	0	—
	XOR.W @aa:32,@ERd-	W	4	4	7						D	S			@ERd @ aa:32 → @ERd				—	—	↑	↓	0	—
	XOR.W @aa:32,@+ERd	W	4	5	8						D	S			ERd32+2→ERd32 @ERd @ aa:32 → @ERd				—	—	↑	↓	0	—
	XOR.W @aa:32,@-ERd	W	4	5	8						D	S			ERd32-2→ERd32 @ERd @ aa:32 → @ERd				—	—	↑	↓	0	—
	XOR.W @aa:32,@(d:2,ERd)	W	4	5	8						D	S			@(2/4/6+ERd) @ aa:32 → @(2/4/6+ERd)				—	—	↑	↓	0	—
	XOR.W @aa:32,@(d:16,ERd)	W	5	5	8						D	S			@(d:16+ERd) @ aa:32 → @(d:16+ERd)				—	—	↑	↓	0	—
	XOR.W @aa:32,@(d:32,ERd)	W	6	5	9						D	S			@(d:32+ERd) @ aa:32 → @(d:32+ERd)				—	—	↑	↓	0	—
	XOR.W @aa:32,@(d:16,Rd.B)	W	5	5	8						D	S			@(d:16+RdL<<1) @ aa:32 → @(d:16+RdL<<1)				—	—	↑	↓	0	—
	XOR.W @aa:32,@(d:16,Rd.W)	W	5	5	8						D	S			@(d:16+Rd<<1) @ aa:32 → @(d:16+Rd<<1)				—	—	↑	↓	0	—
	XOR.W @aa:32,@(d:16,ERd.L)	W	5	5	8						D	S			@(d:16+ERd<<1) @ aa:32 → @(d:16+ERd<<1)				—	—	↑	↓	0	—
	XOR.W @aa:32,@(d:32,Rd.B)	W	6	5	9						D	S			@(d:32+RdL<<1) @ aa:32 → @(d:32+RdL<<1)				—	—	↑	↓	0	—
	XOR.W @aa:32,@(d:32,Rd.W)	W	6	5	9						D	S			@(d:32+Rd<<1) @ aa:32 → @(d:32+Rd<<1)				—	—	↑	↓	0	—
	XOR.W @aa:32,@(d:32,ERd.L)	W	6	5	9						D	S			@(d:32+ERd<<1) @ aa:32 → @(d:32+ERd<<1)				—	—	↑	↓	0	—
	XOR.W @aa:32,@aa:16	W	5	4	8							S			@aa:16 @ aa:32 → @aa:16				—	—	↑	↓	0	—
	XOR.W @aa:32,@aa:32	W	6	4	9							S			@aa:32 @ aa:32 → @aa:32				—	—	↑	↓	0	—
	XOR.L #xx:16,ERd	L	2	1	2	S	D								ERd32 @ #xx:16 → ERd32				—	—	↑	↓	0	—
	XOR.L #xx:32,ERd	L	3	1	3	S	D								ERd32 @ #xx → ERd32				—	—	↑	↓	0	—
	XOR.L #xx:16,@ERd	L	3	4	5	S		D							@ERd @ #xx:16 → @ERd				—	—	↑	↓	0	—
	XOR.L #xx:16,@ERd+	L	3	4	5	S			D						@ERd @ #xx:16 → @ERd				—	—	↑	↓	0	—
XOR.L #xx:16,@ERd-	L	3	4	5	S				D					@ERd @ #xx:16 → @ERd				—	—	↑	↓	0	—	
XOR.L #xx:16,@+ERd	L	3	5	5	S					D				ERd32+4→ERd32 @ERd @ #xx:16 → @ERd				—	—	↑	↓	0	—	
XOR.L #xx:16,@-ERd	L	3	5	5	S					D				ERd32-4→ERd32 @ERd @ #xx:16 → @ERd				—	—	↑	↓	0	—	
XOR.L #xx:16,@(d:2,ERd)	L	3	5	5	S					D				@(4/8/12+ERd) @ #xx:16 → @(4/8/12+ERd)				—	—	↑	↓	0	—	
XOR.L #xx:16,@(d:16,ERd)	L	4	5	6	S					D				@(d:16+ERd) @ #xx:16 → @(d:16+ERd)				—	—	↑	↓	0	—	

Logic operations (105)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹ #xx	Addressing Mode							Operation ⁵					Condition Codes ²											
						Rn	@ERn	@(d,ERn)	@(d,Rn,ERn,W,ERn,L)	@.ERn/@.ERn+/@.ERn-/@.ERn+/@.ERn	@aa:8/@.aa:16/@.aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C							
XOR	XOR.L @ERs+, @ERd	L	3	5	6		D								@ERd	@	@ERs	→	@ERd	ERs32+4→ERs32			—	—	↑	↓	0	—	
	XOR.L @ERs+, @ERd+	L	3	5	6										@ERd	@	@ERs	→	@ERd	ERs32+4→ERs32	ERd32+4→ERd32			—	—	↑	↓	0	—
	XOR.L @ERs+, @ERd-	L	3	5	6										@ERd	@	@ERs	→	@ERd	ERs32+4→ERs32	ERd32+4→ERd32			—	—	↑	↓	0	—
	XOR.L @ERs+, @+ERd	L	3	6	7										ERd32+4→ERd32	@	@ERs	→	@ERd	ERs32+4→ERs32			—	—	↑	↓	0	—	
	XOR.L @ERs+, @-ERd	L	3	6	7										ERd32-4→ERd32	@	@ERs	→	@ERd	ERs32+4→ERs32			—	—	↑	↓	0	—	
	XOR.L @ERs+, @(d:2,ERd)	L	3	6	7				D		S				@(4/8/12+ERd)	@	@ERs	→	@(4/8/12+ERd)	ERs32+4→ERs32			—	—	↑	↓	0	—	
	XOR.L @ERs+, @(d:16,ERd)	L	4	6	7				D		S				@(d:16+ERd)	@	@ERs	→	@(d:16+ERd)	ERs32+4→ERs32			—	—	↑	↓	0	—	
	XOR.L @ERs+, @(d:32,ERd)	L	5	6	8				D		S				@(d:32+ERd)	@	@ERs	→	@(d:32+ERd)	ERs32+4→ERs32			—	—	↑	↓	0	—	
	XOR.L @ERs+, @(d:16,Rd.B)	L	4	6	7				D		S				@(d:16+RdL<<2)	@	@ERs	→	@(d:16+RdL<<2)	ERs32+4→ERs32			—	—	↑	↓	0	—	
	XOR.L @ERs+, @(d:16,Rd.W)	L	4	6	7				D		S				@(d:16+RdL<<2)	@	@ERs	→	@(d:16+RdL<<2)	ERs32+4→ERs32			—	—	↑	↓	0	—	
	XOR.L @ERs+, @(d:16,ERd.L)	L	4	6	7				D		S				@(d:16+ERdL<<2)	@	@ERs	→	@(d:16+ERdL<<2)	ERs32+4→ERs32			—	—	↑	↓	0	—	
	XOR.L @ERs+, @(d:32,Rd.B)	L	5	6	8				D		S				@(d:32+RdL<<2)	@	@ERs	→	@(d:32+RdL<<2)	ERs32+4→ERs32			—	—	↑	↓	0	—	
	XOR.L @ERs+, @(d:32,Rd.W)	L	5	6	8				D		S				@(d:32+RdL<<2)	@	@ERs	→	@(d:32+RdL<<2)	ERs32+4→ERs32			—	—	↑	↓	0	—	
	XOR.L @ERs+, @(d:32,ERd.L)	L	5	6	8				D		S				@(d:32+ERdL<<2)	@	@ERs	→	@(d:32+ERdL<<2)	ERs32+4→ERs32			—	—	↑	↓	0	—	
	XOR.L @ERs+, @aa:16	L	4	5	7						S	D			@aa:16	@	@ERs	→	@aa:16	ERs32+4→ERs32			—	—	↑	↓	0	—	
	XOR.L @ERs+, @aa:32	L	5	5	8						S	D			@aa:32	@	@ERs	→	@aa:32	ERs32+4→ERs32			—	—	↑	↓	0	—	
	XOR.L @ERs-, @ERd	L	3	5	6				D		S				@ERd	@	@ERs	→	@ERd	ERs32-4→ERs32			—	—	↑	↓	0	—	
	XOR.L @ERs-, @ERd+	L	3	5	6						S				@ERd	@	@ERs	→	@ERd	ERs32-4→ERs32	ERd32+4→ERd32			—	—	↑	↓	0	—
	XOR.L @ERs-, @ERd-	L	3	5	6						S				@ERd	@	@ERs	→	@ERd	ERs32-4→ERs32	ERd32+4→ERd32			—	—	↑	↓	0	—
	XOR.L @ERs-, @+ERd	L	3	6	7						S				ERd32+4→ERd32	@	@ERs	→	@ERd	ERs32-4→ERs32			—	—	↑	↓	0	—	
XOR.L @ERs-, @-ERd	L	3	6	7						S				ERd32-4→ERd32	@	@ERs	→	@ERd	ERs32-4→ERs32			—	—	↑	↓	0	—		
XOR.L @ERs-, @(d:2,ERd)	L	3	6	7				D		S				@(4/8/12+ERd)	@	@ERs	→	@(4/8/12+ERd)	ERs32-4→ERs32			—	—	↑	↓	0	—		
XOR.L @ERs-, @(d:16,ERd)	L	4	6	7				D		S				@(d:16+ERd)	@	@ERs	→	@(d:16+ERd)	ERs32-4→ERs32			—	—	↑	↓	0	—		
XOR.L @ERs-, @(d:32,ERd)	L	5	6	8				D		S				@(d:32+ERd)	@	@ERs	→	@(d:32+ERd)	ERs32-4→ERs32			—	—	↑	↓	0	—		
XOR.L @ERs-, @(d:16,Rd.B)	L	4	6	7				D		S				@(d:16+RdL<<2)	@	@ERs	→	@(d:16+RdL<<2)	ERs32-4→ERs32			—	—	↑	↓	0	—		

Logic operations (106)

Instruction n	Mnemonic	Size	Instruction Length	Addressing Mode										Operation ⁵					Condition Codes ^{6,7}					
				Min.	16-Bit Instruction Fetch Execution Status ¹	Rn	@ERn	@ (d,ERn)	@ (d,Rn,ERn,WERn,L)	@ ERn/@ ERn+/@ ERn-/@ +ERn	@ aa:9/@ aa:16/@ aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C		
				Max.	16-Bit Instruction Fetch Execution Status ¹																			
XOR	XOR.L @ERs, @(d:16,Rd,W)	L	4	6	7				D	S				@ (d:16+Rd<<2) @ @ERs → @ (d:16+Rd<<2)	ERs32-4→ERs32				↑	↓	0	—		
	XOR.L @ERs, @(d:16,ERd,L)	L	4	6	7				D	S				@ (d:16+ERd<<2) @ @ERs → @ (d:16+ERd<<2)	ERs32-4→ERs32				↑	↓	0	—		
	XOR.L @ERs, @(d:32,Rd,B)	L	5	6	8				D	S				@ (d:32+RdL<<2) @ @ERs → @ (d:32+RdL<<2)	ERs32-4→ERs32				↑	↓	0	—		
	XOR.L @ERs, @(d:32,Rd,W)	L	5	6	8				D	S				@ (d:32+Rd<<2) @ @ERs → @ (d:32+Rd<<2)	ERs32-4→ERs32				↑	↓	0	—		
	XOR.L @ERs, @(d:32,ERd,L)	L	5	6	8				D	S				@ (d:32+ERd<<2) @ @ERs → @ (d:32+ERd<<2)	ERs32-4→ERs32				↑	↓	0	—		
	XOR.L @ERs, @aa:16	L	4	5	7					S	D				@ aa:16 @ @ERs → @ aa:16	ERs32-4→ERs32				↑	↓	0	—	
	XOR.L @ERs, @aa:32	L	5	5	8					S	D				@ aa:32 @ @ERs → @ aa:32	ERs32-4→ERs32				↑	↓	0	—	
	XOR.L @+ERs, @ERd	L	3	6	6				D	S		S	ERs32+4→ERs32	@ ERd @ @ERs → @ ERd					↑	↓	0	—		
	XOR.L @+ERs, @ERd+	L	3	6	6					S		S	ERs32+4→ERs32	@ ERd @ @ERs → @ ERd	ERd32+4→ERd32				↑	↓	0	—		
	XOR.L @+ERs, @ERd-	L	3	6	6					S		S	ERs32+4→ERs32	@ ERd @ @ERs → @ ERd	ERd32-4→ERd32				↑	↓	0	—		
	XOR.L @+ERs, @+ERd	L	3	7	7					S		S	ERs32+4→ERs32	ERd32+4→ERd32 @ ERd @ @ERs → @ ERd					↑	↓	0	—		
	XOR.L @+ERs, @-ERd	L	3	7	7					S		S	ERs32+4→ERs32	ERd32-4→ERd32 @ ERd @ @ERs → @ ERd					↑	↓	0	—		
	XOR.L @+ERs, @(d:2,ERd)	L	3	7	7				D	S		S	ERs32+4→ERs32	@ (4/8/12+ERd) @ @ERs → @ (4/8/12+ERd)					↑	↓	0	—		
	XOR.L @+ERs, @(d:16,ERd)	L	4	7	7				D	S		S	ERs32+4→ERs32	@ (d:16+ERd) @ @ERs → @ (d:16+ERd)					↑	↓	0	—		
	XOR.L @+ERs, @(d:32,ERd)	L	5	7	8				D	S		S	ERs32+4→ERs32	@ (d:32+ERd) @ @ERs → @ (d:32+ERd)					↑	↓	0	—		
	XOR.L @+ERs, @(d:16,Rd,B)	L	4	7	7				D	S		S	ERs32+4→ERs32	@ (d:16+RdL<<2) @ @ERs → @ (d:16+RdL<<2)					↑	↓	0	—		
	XOR.L @+ERs, @(d:16,Rd,W)	L	4	7	7				D	S		S	ERs32+4→ERs32	@ (d:16+Rd<<2) @ @ERs → @ (d:16+Rd<<2)					↑	↓	0	—		
	XOR.L @+ERs, @(d:16,ERd,L)	L	4	7	7				D	S		S	ERs32+4→ERs32	@ (d:16+ERd<<2) @ @ERs → @ (d:16+ERd<<2)					↑	↓	0	—		
	XOR.L @+ERs, @(d:32,Rd,B)	L	5	7	8				D	S		S	ERs32+4→ERs32	@ (d:32+RdL<<2) @ @ERs → @ (d:32+RdL<<2)					↑	↓	0	—		
	XOR.L @+ERs, @(d:32,Rd,W)	L	5	7	8				D	S		S	ERs32+4→ERs32	@ (d:32+Rd<<2) @ @ERs → @ (d:32+Rd<<2)					↑	↓	0	—		
	XOR.L @+ERs, @(d:32,ERd,L)	L	5	7	8				D	S		S	ERs32+4→ERs32	@ (d:32+ERd<<2) @ @ERs → @ (d:32+ERd<<2)					↑	↓	0	—		
	XOR.L @+ERs, @aa:16	L	4	6	7					S	D		S	ERs32+4→ERs32	@ aa:16 @ @ERs → @ aa:16					↑	↓	0	—	
	XOR.L @+ERs, @aa:32	L	5	6	8					S	D		S	ERs32+4→ERs32	@ aa:32 @ @ERs → @ aa:32					↑	↓	0	—	
	XOR.L @-ERs, @ERd	L	3	6	6				D	S		S	ERs32-4→ERs32	@ ERd @ @ERs → @ ERd					↑	↓	0	—		
	XOR.L @-ERs, @ERd+	L	3	6	6					S		S	ERs32-4→ERs32	@ ERd @ @ERs → @ ERd	ERd32+4→ERd32					↑	↓	0	—	



Logic operations (107)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹	Addressing Mode							Operation ⁵					Condition Codes ²									
						Number of Execution Stages ¹	Rn	@(d,ERn)	@(d,Rn,WRn,L)	@(d,ERn)/@ERn+/@ERn+/@ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C					
XOR	XOR.L @-ERs,@ERd-	L	3	6	6									ERs32-4→ERs32		@ ERd	@ @ERs	→ @ERd			ERd32-4→ERd32	—	—	↑	↓	0	—
	XOR.L @-ERs,@+ERd	L	3	7	7									ERs32-4→ERs32	ERd32+4→ERd32	@ ERd	@ @ERs	→ @ERd				—	—	↑	↓	0	—
	XOR.L @-ERs,@-ERd	L	3	7	7									ERs32-4→ERs32	ERd32-4→ERd32	@ ERd	@ @ERs	→ @ERd				—	—	↑	↓	0	—
	XOR.L @-ERs,@(d:2,ERd)	L	3	7	7							D	S	ERs32-4→ERs32		@ (4/8/12+ERd)	@ @ERs	→ @ (4/8/12+ERd)				—	—	↑	↓	0	—
	XOR.L @-ERs,@(d:16,ERd)	L	4	7	7							D	S	ERs32-4→ERs32		@ (d:16+ERd)	@ @ERs	→ @ (d:16+ERd)				—	—	↑	↓	0	—
	XOR.L @-ERs,@(d:32,ERd)	L	5	7	8							D	S	ERs32-4→ERs32		@ (d:32+ERd)	@ @ERs	→ @ (d:32+ERd)				—	—	↑	↓	0	—
	XOR.L @-ERs,@(d:16,Rd.B)	L	4	7	7							D	S	ERs32-4→ERs32		@ (d:16+Rd<<2)	@ @ERs	→ @ (d:16+Rd<<2)				—	—	↑	↓	0	—
	XOR.L @-ERs,@(d:16,Rd.W)	L	4	7	7							D	S	ERs32-4→ERs32		@ (d:16+Rd<<2)	@ @ERs	→ @ (d:16+Rd<<2)				—	—	↑	↓	0	—
	XOR.L @-ERs,@(d:16,ERd.L)	L	4	7	7							D	S	ERs32-4→ERs32		@ (d:16+ERd<<2)	@ @ERs	→ @ (d:16+ERd<<2)				—	—	↑	↓	0	—
	XOR.L @-ERs,@(d:32,Rd.B)	L	5	7	8							D	S	ERs32-4→ERs32		@ (d:32+Rd<<2)	@ @ERs	→ @ (d:32+Rd<<2)				—	—	↑	↓	0	—
	XOR.L @-ERs,@(d:32,Rd.W)	L	5	7	8							D	S	ERs32-4→ERs32		@ (d:32+Rd<<2)	@ @ERs	→ @ (d:32+Rd<<2)				—	—	↑	↓	0	—
	XOR.L @-ERs,@(d:32,ERd.L)	L	5	7	8							D	S	ERs32-4→ERs32		@ (d:32+ERd<<2)	@ @ERs	→ @ (d:32+ERd<<2)				—	—	↑	↓	0	—
	XOR.L @-ERs,@aa:16	L	4	6	7								S	D	ERs32-4→ERs32	@ aa:16	@ @ERs	→ @aa:16				—	—	↑	↓	0	—
	XOR.L @-ERs,@aa:32	L	5	6	8								S	D	ERs32-4→ERs32	@ aa:32	@ @ERs	→ @aa:32				—	—	↑	↓	0	—
	XOR.L @(d:2,ERs),@ERd	L	3	6	6							D	S			@ ERd	@ @ (4/8/12+ERs)	→ @ERd				—	—	↑	↓	0	—
	XOR.L @(d:2,ERs),@ERd+	L	3	6	6							S	D			@ ERd	@ @ (4/8/12+ERs)	→ @ERd			ERd32+4→ERd32	—	—	↑	↓	0	—
	XOR.L @(d:2,ERs),@ERd-	L	3	6	6							S	D			@ ERd	@ @ (4/8/12+ERs)	→ @ERd			ERd32-4→ERd32	—	—	↑	↓	0	—
	XOR.L @(d:2,ERs),@+ERd	L	3	7	7							S	D		ERd32+4→ERd32	@ ERd	@ @ (4/8/12+ERs)	→ @ERd				—	—	↑	↓	0	—
	XOR.L @(d:2,ERs),@-ERd	L	3	7	7							S	D		ERd32-4→ERd32	@ ERd	@ @ (4/8/12+ERs)	→ @ERd				—	—	↑	↓	0	—
	XOR.L @(d:2,ERs),@(d:2,ERd)	L	3	7	7							S	D			@ (4/8/12+ERd)	@ @ (4/8/12+ERs)	→ @ (4/8/12+ERd)				—	—	↑	↓	0	—
	XOR.L @(d:2,ERs),@(d:16,ERd)	L	4	7	7							S	D			@ (d:16+ERd)	@ @ (4/8/12+ERs)	→ @ (d:16+ERd)				—	—	↑	↓	0	—
	XOR.L @(d:2,ERs),@(d:32,ERd)	L	5	7	8							S	D			@ (d:32+ERd)	@ @ (4/8/12+ERs)	→ @ (d:32+ERd)				—	—	↑	↓	0	—
	XOR.L @(d:2,ERs),@(d:16,Rd.B)	L	4	7	7							S	D			@ (d:16+Rd<<2)	@ @ (4/8/12+ERs)	→ @ (d:16+Rd<<2)				—	—	↑	↓	0	—
	XOR.L @(d:2,ERs),@(d:16,Rd.W)	L	4	7	7							S	D			@ (d:16+Rd<<2)	@ @ (4/8/12+ERs)	→ @ (d:16+Rd<<2)				—	—	↑	↓	0	—
	XOR.L @(d:2,ERs),@(d:16,ERd.L)	L	4	7	7							S	D			@ (d:16+ERd<<2)	@ @ (4/8/12+ERs)	→ @ (d:16+ERd<<2)				—	—	↑	↓	0	—

Logic operations (109)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode										Operation ⁵					Condition Codes ^{6,7}							
				Number of		Execution Status ⁸		Rn	@ERn	@ (d, ERn)	@ (d, Rn, ERn, W, ERn, L)	@ ERn / @ ERn + / @ ERn + / @ ERn	@ aa:8 / @ aa:16 / @ aa:32													
				Min.	Max.	16-Bit Instruction Fetch	Execution Status ⁸																			
n				Min.	Max.	16-Bit Instruction Fetch	Execution Status ⁸	Rn	@ERn	@ (d, ERn)	@ (d, Rn, ERn, W, ERn, L)	@ ERn / @ ERn + / @ ERn + / @ ERn	@ aa:8 / @ aa:16 / @ aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C		
XOR	XOR.L @ (d:32, ERs), @ -ERd	L	5	7	9				S		D				ERd32-4 → ERd32	@ ERd @ @ (d:32+ERs) → @ ERd										
	XOR.L @ (d:32, ERs), @ (d:2, ERd)	L	5	7	9				S							@ (4/8/12+ERd) @ @ (d:32+ERs) → @ (4/8/12+ERd)										
	XOR.L @ (d:32, ERs), @ (d:16, ERd)	L	6	7	9				S							@ (d:16+ERd) @ @ (d:32+ERs) → @ (d:16+ERd)										
	XOR.L @ (d:32, ERs), @ (d:32, ERd)	L	7	7	10				S							@ (d:32+ERd) @ @ (d:32+ERs) → @ (d:32+ERd)										
	XOR.L @ (d:32, ERs), @ (d:16, Rd, B)	L	6	7	9				S	D						@ (d:16+RdL<<2) @ @ (d:32+ERs) → @ (d:16+RdL<<2)										
	XOR.L @ (d:32, ERs), @ (d:16, Rd, W)	L	6	7	9				S	D						@ (d:16+Rd<<2) @ @ (d:32+ERs) → @ (d:16+Rd<<2)										
	XOR.L @ (d:32, ERs), @ (d:16, ERd, L)	L	6	7	9				S	D						@ (d:16+ERd<<2) @ @ (d:32+ERs) → @ (d:16+ERd<<2)										
	XOR.L @ (d:32, ERs), @ (d:32, Rd, B)	L	7	7	10				S	D						@ (d:32+RdL<<2) @ @ (d:32+ERs) → @ (d:32+RdL<<2)										
	XOR.L @ (d:32, ERs), @ (d:32, Rd, W)	L	7	7	10				S	D						@ (d:32+Rd<<2) @ @ (d:32+ERs) → @ (d:32+Rd<<2)										
	XOR.L @ (d:32, ERs), @ (d:32, ERd, L)	L	7	7	10				S	D						@ (d:32+ERd<<2) @ @ (d:32+ERs) → @ (d:32+ERd<<2)										
	XOR.L @ (d:32, ERs), @ aa:16	L	6	6	9				S			D				@ aa:16 @ @ (d:32+ERs) → @ aa:16										
	XOR.L @ (d:32, ERs), @ aa:32	L	7	6	10				S			D				@ aa:32 @ @ (d:32+ERs) → @ aa:32										
	XOR.L @ (d:16, Rs, B), @ ERd	L	4	6	7				D	S						@ ERd @ @ (d:16+RsL<<2) → @ ERd										
	XOR.L @ (d:16, Rs, B), @ ERd+	L	4	6	7					S	D					@ ERd @ @ (d:16+RsL<<2) → @ ERd										
	XOR.L @ (d:16, Rs, B), @ ERd-	L	4	6	7					S	D					@ ERd @ @ (d:16+RsL<<2) → @ ERd										
	XOR.L @ (d:16, Rs, B), @ +ERd	L	4	7	8					S	D				ERd32+4 → ERd32	@ ERd @ @ (d:16+RsL<<2) → @ ERd										
	XOR.L @ (d:16, Rs, B), @ -ERd	L	4	7	8					S	D				ERd32-4 → ERd32	@ ERd @ @ (d:16+RsL<<2) → @ ERd										
	XOR.L @ (d:16, Rs, B), @ (d:2, ERd)	L	4	7	8				D	S						@ (4/8/12+ERd) @ @ (d:16+RsL<<2) → @ (4/8/12+ERd)										
	XOR.L @ (d:16, Rs, B), @ (d:16, ERd)	L	5	7	8				D	S						@ (d:16+ERd) @ @ (d:16+RsL<<2) → @ (d:16+ERd)										
	XOR.L @ (d:16, Rs, B), @ (d:32, ERd)	L	6	7	9				D	S						@ (d:32+ERd) @ @ (d:16+RsL<<2) → @ (d:32+ERd)										
	XOR.L @ (d:16, Rs, B), @ (d:16, Rd, B)	L	5	7	8					S						@ (d:16+RdL<<2) @ @ (d:16+RsL<<2) → @ (d:16+RdL<<2)										
	XOR.L @ (d:16, Rs, B), @ (d:16, Rd, W)	L	5	7	8					S						@ (d:16+Rd<<2) @ @ (d:16+RsL<<2) → @ (d:16+Rd<<2)										
	XOR.L @ (d:16, Rs, B), @ (d:16, ERd, L)	L	5	7	8					S						@ (d:16+ERd<<2) @ @ (d:16+RsL<<2) → @ (d:16+ERd<<2)										
	XOR.L @ (d:16, Rs, B), @ (d:32, Rd, B)	L	6	7	9					S						@ (d:32+RdL<<2) @ @ (d:16+RsL<<2) → @ (d:32+RdL<<2)										

Shift (11)

Table with columns for Instruction n, Mnemonic, Size, Instruction Length, Min. 16-bit Instruction Fetch Execution Stages, Addressing Mode, Operation 1-5, and Condition Codes. Rows include SHLR instructions with various addressing modes and shift operations.



Shift (12)

Instruction n	Mnemonic	Size	Instruction Length	Addressing Mode							Operation ⁵					Condition Codes ^{6,2}									
				Min.	Max.	16-Bit Instruction Fetch Execution Stages ¹ #xx	Rn	@ERn	@(d,ERn)	@(d,Rn,ERn,WERn,L)	@ERN/@ERN+/@ERN-/@+ERn	@aa:R/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C		
SHLR	SHLR.W @-ERd	W	3	5	5									ERd32-2→ERd32	@ERd	>>1 (Logical shift) → @ERd				—	—	↓	↓	0	↓
	SHLR.W @(d,2,ERd)	W	3	5	5				D					@(2/4/6+ERd)	>>1 (Logical shift) → @(2/4/6+ERd)				—	—	↓	↓	0	↓	
	SHLR.W @(d,16,ERd)	W	4	5	6				D					@(d:16+ERd)	>>1 (Logical shift) → @(d:16+ERd)				—	—	↓	↓	0	↓	
	SHLR.W @(d,32,ERd)	W	5	5	7				D					@(d:32+ERd)	>>1 (Logical shift) → @(d:32+ERd)				—	—	↓	↓	0	↓	
	SHLR.W @(d,16,Rd.B)	W	4	5	6					D				@(d:16+Rd<<1)	>>1 (Logical shift) → @(d:16+Rd<<1)				—	—	↓	↓	0	↓	
	SHLR.W @(d,16,Rd.W)	W	4	5	6						D			@(d:16+Rd<<1)	>>1 (Logical shift) → @(d:16+Rd<<1)				—	—	↓	↓	0	↓	
	SHLR.W @(d,16,ERd.L)	W	4	5	6					D				@(d:16+ERd<<1)	>>1 (Logical shift) → @(d:16+ERd<<1)				—	—	↓	↓	0	↓	
	SHLR.W @(d,32,Rd.B)	W	5	5	7						D			@(d:32+Rd<<1)	>>1 (Logical shift) → @(d:32+Rd<<1)				—	—	↓	↓	0	↓	
	SHLR.W @(d,32,Rd.W)	W	5	5	7							D		@(d:32+Rd<<1)	>>1 (Logical shift) → @(d:32+Rd<<1)				—	—	↓	↓	0	↓	
	SHLR.W @(d,32,ERd.L)	W	5	5	7								D	@(d:32+ERd<<1)	>>1 (Logical shift) → @(d:32+ERd<<1)				—	—	↓	↓	0	↓	
	SHLR.W @aa:16	W	3	4	5								D	@aa:16	>>1 (Logical shift) → @aa:16				—	—	↓	↓	0	↓	
	SHLR.W @aa:32	W	4	4	6								D	@aa:32	>>1 (Logical shift) → @aa:32				—	—	↓	↓	0	↓	
	SHLR.W #2,Rd	W	1	1	1							D		Rd16	>>2 (Logical shift) → Rd16				—	—	↓	↓	0	↓	
	SHLR.W #2,@ERd	W	2	4	4								D	@ERd	>>2 (Logical shift) → @ERd				—	—	↓	↓	0	↓	
	SHLR.W #2,@ERd+	W	3	5	5									@ERd	>>2 (Logical shift) → @ERd				—	—	↓	↓	0	↓	
	SHLR.W #2,@ERd-	W	3	5	5									@ERd	>>2 (Logical shift) → @ERd				—	—	↓	↓	0	↓	
	SHLR.W #2,@+ERd	W	3	5	5								D	ERd32+2→ERd32	@ERd	>>2 (Logical shift) → @ERd				—	—	↓	↓	0	↓
	SHLR.W #2,@-ERd	W	3	5	5								D	ERd32-2→ERd32	@ERd	>>2 (Logical shift) → @ERd				—	—	↓	↓	0	↓
	SHLR.W #2,@(d,2,ERd)	W	3	5	5									@(2/4/6+ERd)	>>2 (Logical shift) → @(2/4/6+ERd)				—	—	↓	↓	0	↓	
	SHLR.W #2,@(d,16,ERd)	W	4	5	6									@(d:16+ERd)	>>2 (Logical shift) → @(d:16+ERd)				—	—	↓	↓	0	↓	
	SHLR.W #2,@(d,32,ERd)	W	5	5	7									@(d:32+ERd)	>>2 (Logical shift) → @(d:32+ERd)				—	—	↓	↓	0	↓	
	SHLR.W #2,@(d,16,Rd.B)	W	4	5	6									@(d:16+Rd<<1)	>>2 (Logical shift) → @(d:16+Rd<<1)				—	—	↓	↓	0	↓	
	SHLR.W #2,@(d,16,Rd.W)	W	4	5	6									@(d:16+Rd<<1)	>>2 (Logical shift) → @(d:16+Rd<<1)				—	—	↓	↓	0	↓	
	SHLR.W #2,@(d,16,ERd.L)	W	4	5	6									@(d:16+ERd<<1)	>>2 (Logical shift) → @(d:16+ERd<<1)				—	—	↓	↓	0	↓	
	SHLR.W #2,@(d,32,Rd.B)	W	5	5	7									@(d:32+Rd<<1)	>>2 (Logical shift) → @(d:32+Rd<<1)				—	—	↓	↓	0	↓	

Shift (13)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹ #xx	Addressing Mode						Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	Condition Codes ²					
						Rn	@ERn	@(d,ERn)	@(d,Rn,ERn,WERn,L)	@(d,Rn/ERn/ERn+/ERn+/ERn	@aa:R/aa:16/aa:32						I	H	N	Z	V	C
SHLR	SHLR.W #2, @(d:32,Rd.W)	W	5	5	7					D							—	—	↓	↓	0	↓
	SHLR.W #2, @(d:32,ERd.L)	W	5	5	7					D							—	—	↓	↓	0	↓
	SHLR.W #2, @aa:16	W	3	4	5											D	—	—	↓	↓	0	↓
	SHLR.W #2, @aa:32	W	4	4	6											D	—	—	↓	↓	0	↓
	SHLR.W #4, Rd	W	1	1	1					D							—	—	↓	↓	0	↓
	SHLR.W #4, @ERd	W	2	4	4					D							—	—	↓	↓	0	↓
	SHLR.W #4, @ERd+	W	3	5	5						D						—	—	↓	↓	0	↓
	SHLR.W #4, @ERd-	W	3	5	5						D						—	—	↓	↓	0	↓
	SHLR.W #4, @+ERd	W	3	5	5						D		ERd32+2→ERd32				—	—	↓	↓	0	↓
	SHLR.W #4, @-ERd	W	3	5	5						D		ERd32-2→ERd32				—	—	↓	↓	0	↓
	SHLR.W #4, @(d:2,ERd)	W	3	5	5					D							—	—	↓	↓	0	↓
	SHLR.W #4, @(d:16,ERd)	W	4	5	6					D							—	—	↓	↓	0	↓
	SHLR.W #4, @(d:32,ERd)	W	5	5	7					D							—	—	↓	↓	0	↓
	SHLR.W #4, @(d:16,Rd.B)	W	4	5	6						D						—	—	↓	↓	0	↓
	SHLR.W #4, @(d:16,Rd.W)	W	4	5	6						D						—	—	↓	↓	0	↓
	SHLR.W #4, @(d:16,ERd.L)	W	4	5	6						D						—	—	↓	↓	0	↓
	SHLR.W #4, @(d:32,Rd.B)	W	5	5	7						D						—	—	↓	↓	0	↓
	SHLR.W #4, @(d:32,Rd.W)	W	5	5	7						D						—	—	↓	↓	0	↓
	SHLR.W #4, @(d:32,ERd.L)	W	5	5	7						D						—	—	↓	↓	0	↓
	SHLR.W #4, @aa:16	W	3	4	5											D	—	—	↓	↓	0	↓
	SHLR.W #4, @aa:32	W	4	4	6											D	—	—	↓	↓	0	↓
	SHLR.W #8, Rd	W	1	1	1					D							—	—	↓	↓	0	↓
	SHLR.W #8, @ERd	W	2	4	4					D							—	—	↓	↓	0	↓
	SHLR.W #8, @ERd+	W	3	5	5						D						—	—	↓	↓	0	↓
	SHLR.W #8, @ERd-	W	3	5	5						D						—	—	↓	↓	0	↓

Shift (15)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode										Operation ⁵					Condition Codes ^{3,2}										
				Number of 16-Bit Instruction Fetch Execution Stages ¹																									
				Min.	Max.	Rn	Rn	@(d,ERn)	@(d,Rn,ERn,WERn,L)	@(d,Rn)/@ERn+/@ERn+/@ERn+/@ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C							
SHLR	SHLR.L @(d:16,Rd,W)	L	4	5	6							D					@(d:16+Rd<<2) >>1 (Logical shift) → @(d:16+Rd<<2)						—	—	↕	↕	0	↕	
	SHLR.L @(d:16,ERd,L)	L	4	5	6							D					@(d:16+ERd<<2) >>1 (Logical shift) → @(d:16+ERd<<2)						—	—	↕	↕	0	↕	
	SHLR.L @(d:32,Rd,B)	L	5	5	7								D				@(d:32+RdL<<2) >>1 (Logical shift) → @(d:32+RdL<<2)						—	—	↕	↕	0	↕	
	SHLR.L @(d:32,Rd,W)	L	5	5	7								D				@(d:32+Rd<<2) >>1 (Logical shift) → @(d:32+Rd<<2)						—	—	↕	↕	0	↕	
	SHLR.L @(d:32,ERd,L)	L	5	5	7								D				@(d:32+ERd<<2) >>1 (Logical shift) → @(d:32+ERd<<2)						—	—	↕	↕	0	↕	
	SHLR.L @aa:16	L	4	5	6									D				@aa:16 >>1 (Logical shift) → @aa:16						—	—	↕	↕	0	↕
	SHLR.L @aa:32	L	5	5	7									D				@aa:32 >>1 (Logical shift) → @aa:32						—	—	↕	↕	0	↕
	SHLR.L #2,ERd	L	1	1	1			D										Rd32 >>2 (Logical shift) → Rd32						—	—	↕	↕	0	↕
	SHLR.L #2,@ERd	L	3	5	5				D									@ERd >>2 (Logical shift) → @ERd						—	—	↕	↕	0	↕
	SHLR.L #2,@ERd+	L	3	5	5					D								@ERd >>2 (Logical shift) → @ERd					ERd32+4→ERd32	—	—	↕	↕	0	↕
	SHLR.L #2,@ERd-	L	3	5	5						D							@ERd >>2 (Logical shift) → @ERd					ERd32-4→ERd32	—	—	↕	↕	0	↕
	SHLR.L #2,@+ERd	L	3	5	5							D				ERd32+4→ERd32	@ERd >>2 (Logical shift) → @ERd						—	—	↕	↕	0	↕	
	SHLR.L #2,@-ERd	L	3	5	5								D			ERd32-4→ERd32	@ERd >>2 (Logical shift) → @ERd						—	—	↕	↕	0	↕	
	SHLR.L #2,@(d:2,ERd)	L	3	5	5								D					@(4/8/12+ERd) >>2 (Logical shift) → @(4/8/12+ERd)						—	—	↕	↕	0	↕
	SHLR.L #2,@(d:16,ERd)	L	4	5	6									D				@(d:16+ERd) >>2 (Logical shift) → @(d:16+ERd)						—	—	↕	↕	0	↕
	SHLR.L #2,@(d:32,ERd)	L	5	5	7										D			@(d:32+ERd) >>2 (Logical shift) → @(d:32+ERd)						—	—	↕	↕	0	↕
	SHLR.L #2,@(d:16,Rd,B)	L	4	5	6													@(d:16+RdL<<2) >>2 (Logical shift) → @(d:16+RdL<<2)						—	—	↕	↕	0	↕
	SHLR.L #2,@(d:16,Rd,W)	L	4	5	6													@(d:16+Rd<<2) >>2 (Logical shift) → @(d:16+Rd<<2)						—	—	↕	↕	0	↕
	SHLR.L #2,@(d:16,ERd,L)	L	4	5	6													@(d:16+ERd<<2) >>2 (Logical shift) → @(d:16+ERd<<2)						—	—	↕	↕	0	↕
	SHLR.L #2,@(d:32,Rd,B)	L	5	5	7													@(d:32+RdL<<2) >>2 (Logical shift) → @(d:32+RdL<<2)						—	—	↕	↕	0	↕
	SHLR.L #2,@(d:32,Rd,W)	L	5	5	7													@(d:32+Rd<<2) >>2 (Logical shift) → @(d:32+Rd<<2)						—	—	↕	↕	0	↕
	SHLR.L #2,@(d:32,ERd,L)	L	5	5	7													@(d:32+ERd<<2) >>2 (Logical shift) → @(d:32+ERd<<2)						—	—	↕	↕	0	↕
	SHLR.L #2,@aa:16	L	4	5	6													@aa:16 >>2 (Logical shift) → @aa:16						—	—	↕	↕	0	↕
	SHLR.L #2,@aa:32	L	5	5	7													@aa:32 >>2 (Logical shift) → @aa:32						—	—	↕	↕	0	↕
	SHLR.L #4,ERd	L	1	1	1			D										Rd32 >>4 (Logical shift) → Rd32						—	—	↕	↕	0	↕

Shift (16)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	Addressing Mode										Operation ⁵					Condition Codes ²												
						16-Bit Instruction Fetch Execution Status ¹	Prx	Rn	@(d,ERn)	@(d,Rn,WRn,WRn,L)	@(d,Rn)/@ERn+/@ERn+/@ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3											Operation 4	Operation 5	I	H	N	Z	V	C
SHLR	SHLR.L #4, @ERd	L	3	5	5					D							@ERd	>>4 (Logical shift)	→	@ERd				—	—	↓	↓	0	↓				
	SHLR.L #4, @ERd+	L	3	5	5									D			@ERd	>>4 (Logical shift)	→	@ERd						ERd32+4→ERd32	—	—	↓	↓	0	↓	
	SHLR.L #4, @ERd-	L	3	5	5										D		@ERd	>>4 (Logical shift)	→	@ERd						ERd32-4→ERd32	—	—	↓	↓	0	↓	
	SHLR.L #4, @+ERd	L	3	5	5												ERd32+4→ERd32	@ERd	>>4 (Logical shift)	→	@ERd							—	—	↓	↓	0	↓
	SHLR.L #4, @-ERd	L	3	5	5												ERd32-4→ERd32	@ERd	>>4 (Logical shift)	→	@ERd							—	—	↓	↓	0	↓
	SHLR.L #4, @(d:2,ERd)	L	3	5	5									D				@(4/8/12+ERd)	>>4 (Logical shift)	→	@(4/8/12+ERd)							—	—	↓	↓	0	↓
	SHLR.L #4, @(d:16,ERd)	L	4	5	6									D				@(d:16+ERd)	>>4 (Logical shift)	→	@(d:16+ERd)							—	—	↓	↓	0	↓
	SHLR.L #4, @(d:32,ERd)	L	5	5	7									D				@(d:32+ERd)	>>4 (Logical shift)	→	@(d:32+ERd)							—	—	↓	↓	0	↓
	SHLR.L #4, @(d:16,Rd,B)	L	4	5	6										D			@(d:16+RdL<<2)	>>4 (Logical shift)	→	@(d:16+RdL<<2)							—	—	↓	↓	0	↓
	SHLR.L #4, @(d:16,Rd,W)	L	4	5	6										D			@(d:16+RdL<<2)	>>4 (Logical shift)	→	@(d:16+RdL<<2)							—	—	↓	↓	0	↓
	SHLR.L #4, @(d:16,ERd,L)	L	4	5	6										D			@(d:16+ERdL<<2)	>>4 (Logical shift)	→	@(d:16+ERdL<<2)							—	—	↓	↓	0	↓
	SHLR.L #4, @(d:32,Rd,B)	L	5	5	7										D			@(d:32+RdL<<2)	>>4 (Logical shift)	→	@(d:32+RdL<<2)							—	—	↓	↓	0	↓
	SHLR.L #4, @(d:32,Rd,W)	L	5	5	7										D			@(d:32+RdL<<2)	>>4 (Logical shift)	→	@(d:32+RdL<<2)							—	—	↓	↓	0	↓
	SHLR.L #4, @(d:32,ERd,L)	L	5	5	7										D			@(d:32+ERdL<<2)	>>4 (Logical shift)	→	@(d:32+ERdL<<2)							—	—	↓	↓	0	↓
	SHLR.L #4, @aa:16	L	4	5	6											D		@aa:16	>>4 (Logical shift)	→	@aa:16							—	—	↓	↓	0	↓
	SHLR.L #4, @aa:32	L	5	5	7											D		@aa:32	>>4 (Logical shift)	→	@aa:32							—	—	↓	↓	0	↓
	SHLR.L #8, ERd	L	1	1	1						D							Rd32	>>8 (Logical shift)	→	Rd32							—	—	↓	↓	0	↓
	SHLR.L #8, @ERd	L	3	5	5							D						@ERd	>>8 (Logical shift)	→	@ERd							—	—	↓	↓	0	↓
	SHLR.L #8, @ERd+	L	3	5	5													@ERd	>>8 (Logical shift)	→	@ERd						ERd32+4→ERd32	—	—	↓	↓	0	↓
	SHLR.L #8, @ERd-	L	3	5	5													@ERd	>>8 (Logical shift)	→	@ERd						ERd32-4→ERd32	—	—	↓	↓	0	↓
	SHLR.L #8, @+ERd	L	3	5	5													ERd32+4→ERd32	@ERd	>>8 (Logical shift)	→	@ERd							—	—	↓	↓	0
SHLR.L #8, @-ERd	L	3	5	5													ERd32-4→ERd32	@ERd	>>8 (Logical shift)	→	@ERd							—	—	↓	↓	0	↓
SHLR.L #8, @(d:2,ERd)	L	3	5	5										D			@(4/8/12+ERd)	>>8 (Logical shift)	→	@(4/8/12+ERd)							—	—	↓	↓	0	↓	
SHLR.L #8, @(d:16,ERd)	L	4	5	6										D			@(d:16+ERd)	>>8 (Logical shift)	→	@(d:16+ERd)							—	—	↓	↓	0	↓	
SHLR.L #8, @(d:32,ERd)	L	5	5	7										D			@(d:32+ERd)	>>8 (Logical shift)	→	@(d:32+ERd)							—	—	↓	↓	0	↓	

Shift (17)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode								Operation ⁵					Condition Codes ^{6,7}							
				Min.	16-Bit Instruction Fetch Execution Status ¹	Rn	@ERn	@ (d, ERn)	@ (d, Rn, ERn, W, ERn, L)	@ ERn / @ ERn+ / @ ERn+ / @ ERn+ / @ ERn	@ aa:9 / @ aa:16 / @ aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C		
																							Number of 16-Bit Instruction Fetch Execution Status ¹	pxx
SHLR	SHLR.L #8, @(d:16, Rd.B)	L	4	5	6				D			@(d:16+RdL<<2) >>8 (Logical shift) → @(d:16+RdL<<2)						—	—	↑	↓	0	↓	
	SHLR.L #8, @(d:16, Rd.W)	L	4	5	6				D			@(d:16+Rd<<2) >>8 (Logical shift) → @(d:16+Rd<<2)						—	—	↑	↓	0	↓	
	SHLR.L #8, @(d:16, ERd.L)	L	4	5	6				D			@(d:16+ERd<<2) >>8 (Logical shift) → @(d:16+ERd<<2)						—	—	↑	↓	0	↓	
	SHLR.L #8, @(d:32, Rd.B)	L	5	5	7				D			@(d:32+RdL<<2) >>8 (Logical shift) → @(d:32+RdL<<2)						—	—	↑	↓	0	↓	
	SHLR.L #8, @(d:32, Rd.W)	L	5	5	7				D			@(d:32+Rd<<2) >>8 (Logical shift) → @(d:32+Rd<<2)						—	—	↑	↓	0	↓	
	SHLR.L #8, @(d:32, ERd.L)	L	5	5	7				D			@(d:32+ERd<<2) >>8 (Logical shift) → @(d:32+ERd<<2)						—	—	↑	↓	0	↓	
	SHLR.L #8, @aa:16	L	4	5	6					D			@aa:16 >>8 (Logical shift) → @aa:16						—	—	↑	↓	0	↓
	SHLR.L #8, @aa:32	L	5	5	7					D			@aa:32 >>8 (Logical shift) → @aa:32						—	—	↑	↓	0	↓
	SHLR.L #16, ERd	L	1	1	1			D					Rd32 >>16 (Logical shift) → Rd32						—	—	↑	↓	0	↓
	SHLR.L #16, @ERd	L	3	5	5				D				@ERd >>16 (Logical shift) → @ERd						—	—	↑	↓	0	↓
	SHLR.L #16, @ERd+	L	3	5	5					D			@ERd >>16 (Logical shift) → @ERd		ERd32+4→ERd32				—	—	↑	↓	0	↓
	SHLR.L #16, @ERd-	L	3	5	5					D			@ERd >>16 (Logical shift) → @ERd		ERd32-4→ERd32				—	—	↑	↓	0	↓
	SHLR.L #16, @+ERd	L	3	5	5					D	ERd32+4→ERd32		@ERd >>16 (Logical shift) → @ERd						—	—	↑	↓	0	↓
	SHLR.L #16, @-ERd	L	3	5	5					D	ERd32-4→ERd32		@ERd >>16 (Logical shift) → @ERd						—	—	↑	↓	0	↓
	SHLR.L #16, @(d:2, ERd)	L	3	5	5				D				@(4/8/12+ERd) >>16 (Logical shift) → @(4/8/12+ERd)						—	—	↑	↓	0	↓
	SHLR.L #16, @(d:16, ERd)	L	4	5	6				D				@(d:16+ERd) >>16 (Logical shift) → @(d:16+ERd)						—	—	↑	↓	0	↓
	SHLR.L #16, @(d:32, ERd)	L	5	5	7				D				@(d:32+ERd) >>16 (Logical shift) → @(d:32+ERd)						—	—	↑	↓	0	↓
	SHLR.L #16, @(d:16, Rd.B)	L	4	5	6					D			@(d:16+RdL<<2) >>16 (Logical shift) → @(d:16+RdL<<2)						—	—	↑	↓	0	↓
	SHLR.L #16, @(d:16, Rd.W)	L	4	5	6					D			@(d:16+Rd<<2) >>16 (Logical shift) → @(d:16+Rd<<2)						—	—	↑	↓	0	↓
	SHLR.L #16, @(d:16, ERd.L)	L	4	5	6					D			@(d:16+ERd<<2) >>16 (Logical shift) → @(d:16+ERd<<2)						—	—	↑	↓	0	↓
SHLR.L #16, @(d:32, Rd.B)	L	5	5	7					D			@(d:32+RdL<<2) >>16 (Logical shift) → @(d:32+RdL<<2)						—	—	↑	↓	0	↓	
SHLR.L #16, @(d:32, Rd.W)	L	5	5	7					D			@(d:32+Rd<<2) >>16 (Logical shift) → @(d:32+Rd<<2)						—	—	↑	↓	0	↓	
SHLR.L #16, @(d:32, ERd.L)	L	5	5	7					D			@(d:32+ERd<<2) >>16 (Logical shift) → @(d:32+ERd<<2)						—	—	↑	↓	0	↓	
SHLR.L #16, @aa:16	L	4	5	6						D		@aa:16 >>16 (Logical shift) → @aa:16						—	—	↑	↓	0	↓	
SHLR.L #16, @aa:32	L	5	5	7						D		@aa:32 >>16 (Logical shift) → @aa:32						—	—	↑	↓	0	↓	

Shift (18)

Instruction	Mnemonic	Size	Instruction Length			Addressing Mode										Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	Condition Codes ^{s2}											
			Min.	Max.	16-Bit Instruction Fetch Execution Stages ^{s1}	#xxx	Rn	@ERn	@(d,ERn)	@(d,Rn,LRn,WERn,L)	@ERn/@ERn+/@ERn-/@+ERn	@aa:8/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3						Operation 4	Operation 5	I	H	N	Z	V	C				
SHAL	SHAL_B Rd	B	1	1	1		D										Rd8 <<1 (Arithmetic shift) → Rd8								—	—	—	—	—	—		
	SHAL_B @ERd	B	2	4	4			D									@ERd <<1 (Arithmetic shift) → @ERd								—	—	—	—	—	—		
	SHAL_B @ERd+	B	3	5	5								D				@ERd <<1 (Arithmetic shift) → @ERd					ERd32+1→ERd32				—	—	—	—	—	—	
	SHAL_B @ERd-	B	3	5	5								D				@ERd <<1 (Arithmetic shift) → @ERd					ERd32-1→ERd32				—	—	—	—	—	—	
	SHAL_B @+ERd	B	3	5	5									D			ERd32+1→ERd32	@ERd <<1 (Arithmetic shift) → @ERd								—	—	—	—	—	—	
	SHAL_B @-ERd	B	3	5	5									D			ERd32-1→ERd32	@ERd <<1 (Arithmetic shift) → @ERd								—	—	—	—	—	—	
	SHAL_B @(d:2,ERd)	B	3	5	5									D				@(1/2/3+ERd) <<1 (Arithmetic shift) → @(1/2/3+ERd)								—	—	—	—	—	—	
	SHAL_B @(d:16,ERd)	B	4	5	6									D				@(d:16+ERd) <<1 (Arithmetic shift) → @(d:16+ERd)								—	—	—	—	—	—	
	SHAL_B @(d:32,ERd)	B	5	5	7									D				@(d:32+ERd) <<1 (Arithmetic shift) → @(d:32+ERd)								—	—	—	—	—	—	
	SHAL_B @(d:16,Rd,B)	B	4	5	6									D				@(d:16+RdL) <<1 (Arithmetic shift) → @(d:16+RdL)								—	—	—	—	—	—	
	SHAL_B @(d:16,Rd,W)	B	4	5	6									D				@(d:16+Rd) <<1 (Arithmetic shift) → @(d:16+Rd)								—	—	—	—	—	—	
	SHAL_B @(d:16,ERd,L)	B	4	5	6									D				@(d:16+ERd) <<1 (Arithmetic shift) → @(d:16+ERd)								—	—	—	—	—	—	
	SHAL_B @(d:32,Rd,B)	B	5	5	7									D				@(d:32+RdL) <<1 (Arithmetic shift) → @(d:32+RdL)								—	—	—	—	—	—	
	SHAL_B @(d:32,Rd,W)	B	5	5	7									D				@(d:32+Rd) <<1 (Arithmetic shift) → @(d:32+Rd)								—	—	—	—	—	—	
	SHAL_B @(d:32,ERd,L)	B	5	5	7									D				@(d:32+ERd) <<1 (Arithmetic shift) → @(d:32+ERd)								—	—	—	—	—	—	
	SHAL_B @aa:8	B	2	4	4										D			@aa:8 <<1 (Arithmetic shift) → @aa:8								—	—	—	—	—	—	
	SHAL_B @aa:16	B	3	4	5										D			@aa:16 <<1 (Arithmetic shift) → @aa:16								—	—	—	—	—	—	
	SHAL_B @aa:32	B	4	4	6										D			@aa:32 <<1 (Arithmetic shift) → @aa:32								—	—	—	—	—	—	
	SHAL_B #2,Rd	B	1	1	1									D				Rd8 <<2 (Arithmetic shift) → Rd8								—	—	—	—	—	—	
	SHAL_B #2,@ERd	B	2	4	4									D				@ERd <<2 (Arithmetic shift) → @ERd								—	—	—	—	—	—	
	SHAL_B #2,@ERd+	B	3	5	5									D				@ERd <<2 (Arithmetic shift) → @ERd					ERd32+1→ERd32				—	—	—	—	—	—
	SHAL_B #2,@ERd-	B	3	5	5									D				@ERd <<2 (Arithmetic shift) → @ERd					ERd32-1→ERd32				—	—	—	—	—	—
	SHAL_B #2,@+ERd	B	3	5	5									D				ERd32+1→ERd32	@ERd <<2 (Arithmetic shift) → @ERd								—	—	—	—	—	
	SHAL_B #2,@-ERd	B	3	5	5									D				ERd32-1→ERd32	@ERd <<2 (Arithmetic shift) → @ERd								—	—	—	—	—	
	SHAL_B #2,@(d:2,ERd)	B	3	5	5									D				@(1/2/3+ERd) <<2 (Arithmetic shift) → @(1/2/3+ERd)								—	—	—	—	—	—	



Shift (26)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode								Operation ⁶					Condition Codes ^{5,2}						
				Min.	16-Bit Instruction Fetch Execution Status ¹	#px	Rn	@ERn	@ (d,ERn)	@ (d,Rn,ERn,WERn,L)	@ ERn/ @ ERn+/ @ ERn+/ @ ERn	@ aa:0/ @ aa:16/ @ aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C
SHAR	SHAR.L #2, @(d:16,Rd.B)	L	4	5	6					D			@ (d:16+RdL<<2) >>2 (Arithmetic shift) → @ (d:16+RdL<<2)			—	—	↑	↓	0	↑		
	SHAR.L #2, @(d:16,Rd.W)	L	4	5	6					D			@ (d:16+Rd<<2) >>2 (Arithmetic shift) → @ (d:16+Rd<<2)			—	—	↑	↓	0	↑		
	SHAR.L #2, @(d:16,ERd.L)	L	4	5	6					D			@ (d:16+ERd<<2) >>2 (Arithmetic shift) → @ (d:16+ERd<<2)			—	—	↑	↓	0	↑		
	SHAR.L #2, @(d:32,Rd.B)	L	5	5	7					D			@ (d:32+RdL<<2) >>2 (Arithmetic shift) → @ (d:32+RdL<<2)			—	—	↑	↓	0	↑		
	SHAR.L #2, @(d:32,Rd.W)	L	5	5	7					D			@ (d:32+Rd<<2) >>2 (Arithmetic shift) → @ (d:32+Rd<<2)			—	—	↑	↓	0	↑		
	SHAR.L #2, @(d:32,ERd.L)	L	5	5	7					D			@ (d:32+ERd<<2) >>2 (Arithmetic shift) → @ (d:32+ERd<<2)			—	—	↑	↓	0	↑		
	SHAR.L #2, @aa:16	L	4	5	6					D			@aa:16 >>2 (Arithmetic shift) → @aa:16			—	—	↑	↓	0	↑		
	SHAR.L #2, @aa:32	L	5	5	7					D			@aa:32 >>2 (Arithmetic shift) → @aa:32			—	—	↑	↓	0	↑		
ROTL	ROTL.B Rd	B	1	1	1				D				Rd8 <<1 (rotate) → Rd8			—	—	↑	↓	0	↑		
	ROTL.B @ERd	B	2	4	4				D				@ERd <<1 (rotate) → @ERd			—	—	↑	↓	0	↑		
	ROTL.B @ERd+	B	3	5	5					D			@ERd <<1 (rotate) → @ERd	ERd32+1→ERd32		—	—	↑	↓	0	↑		
	ROTL.B @ERd-	B	3	5	5					D			@ERd <<1 (rotate) → @ERd	ERd32-1→ERd32		—	—	↑	↓	0	↑		
	ROTL.B @+ERd	B	3	5	5					D	ERd32+1→ERd32	@ERd	<<1 (rotate) → @ERd			—	—	↑	↓	0	↑		
	ROTL.B @-ERd	B	3	5	5					D	ERd32-1→ERd32	@ERd	<<1 (rotate) → @ERd			—	—	↑	↓	0	↑		
	ROTL.B @(d:2,ERd)	B	3	5	5					D			@ (1/2/3+ERd) <<1 (rotate) → @ (1/2/3+ERd)			—	—	↑	↓	0	↑		
	ROTL.B @(d:16,ERd)	B	4	5	6					D			@ (d:16+ERd) <<1 (rotate) → @ (d:16+ERd)			—	—	↑	↓	0	↑		
	ROTL.B @(d:32,ERd)	B	5	5	7					D			@ (d:32+ERd) <<1 (rotate) → @ (d:32+ERd)			—	—	↑	↓	0	↑		
	ROTL.B @(d:16,Rd.B)	B	4	5	6					D			@ (d:16+RdL) <<1 (rotate) → @ (d:16+RdL)			—	—	↑	↓	0	↑		
	ROTL.B @(d:16,Rd.W)	B	4	5	6					D			@ (d:16+Rd) <<1 (rotate) → @ (d:16+Rd)			—	—	↑	↓	0	↑		
	ROTL.B @(d:16,ERd.L)	B	4	5	6					D			@ (d:16+ERd) <<1 (rotate) → @ (d:16+ERd)			—	—	↑	↓	0	↑		
	ROTL.B @(d:32,Rd.B)	B	5	5	7					D			@ (d:32+RdL) <<1 (rotate) → @ (d:32+RdL)			—	—	↑	↓	0	↑		
	ROTL.B @(d:32,Rd.W)	B	5	5	7					D			@ (d:32+Rd) <<1 (rotate) → @ (d:32+Rd)			—	—	↑	↓	0	↑		
	ROTL.B @(d:32,ERd.L)	B	5	5	7					D			@ (d:32+ERd) <<1 (rotate) → @ (d:32+ERd)			—	—	↑	↓	0	↑		
	ROTL.B @aa:8	B	2	4	4					D			@aa:8 <<1 (rotate) → @aa:8			—	—	↑	↓	0	↑		
	ROTL.B @aa:16	B	3	4	5					D			@aa:16 <<1 (rotate) → @aa:16			—	—	↑	↓	0	↑		



Shift (27)

Instruction	Mnemonic	Size	Instruction Length						Addressing Mode								Operation ⁵					Condition Code ^{6,2}				
			Min.	16-Bit	Instruction	Fetch	Execution	Stages ⁴	Rn	@ERN	@(d,ERN)	@(d,Rn,ERN,W,ERN,L)	@ERN/ERN+/ERN-/ERn/ERn+	@aa:8/aa:16/aa:32												
			4	4	4	6																				
n			Min.	16-Bit	Instruction	Fetch	Execution	Stages ⁴	Rn	@ERN	@(d,ERN)	@(d,Rn,ERN,W,ERN,L)	@ERN/ERN+/ERN-/ERn/ERn+	@aa:8/aa:16/aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C	
ROTL	ROTLB @aa:32	B	4	4	6								D				@aa:32 <<1 (rotate) → @aa:32			—	—	↑	↓	0	↓	
	ROTLB #2,Rd	B	1	1	1	D											Rd8 <<2 (rotate) → Rd8			—	—	↑	↓	0	↓	
	ROTLB #2,@ERd	B	2	4	4			D									@ERd <<2 (rotate) → @ERd			—	—	↑	↓	0	↓	
	ROTLB #2,@ERd+	B	3	5	5							D					@ERd <<2 (rotate) → @ERd		ERd32+1→ERd32	—	—	↑	↓	0	↓	
	ROTLB #2,@ERd-	B	3	5	5							D					@ERd <<2 (rotate) → @ERd		ERd32-1→ERd32	—	—	↑	↓	0	↓	
	ROTLB #2,@+ERd	B	3	5	5								D		ERd32+1→ERd32		@ERd <<2 (rotate) → @ERd			—	—	↑	↓	0	↓	
	ROTLB #2,@-ERd	B	3	5	5								D		ERd32-1→ERd32		@ERd <<2 (rotate) → @ERd			—	—	↑	↓	0	↓	
	ROTLB #2,@(d:2,ERd)	B	3	5	5						D						@(1/2/3+ERd) <<2 (rotate) → @(1/2/3+ERd)			—	—	↑	↓	0	↓	
	ROTLB #2,@(d:16,ERd)	B	4	5	6						D						@(d:16+ERd) <<2 (rotate) → @(d:16+ERd)			—	—	↑	↓	0	↓	
	ROTLB #2,@(d:32,ERd)	B	5	5	7						D						@(d:32+ERd) <<2 (rotate) → @(d:32+ERd)			—	—	↑	↓	0	↓	
	ROTLB #2,@(d:16,Rd,B)	B	4	5	6						D						@(d:16+RdL) <<2 (rotate) → @(d:16+RdL)			—	—	↑	↓	0	↓	
	ROTLB #2,@(d:16,Rd,W)	B	4	5	6						D						@(d:16+Rd) <<2 (rotate) → @(d:16+Rd)			—	—	↑	↓	0	↓	
	ROTLB #2,@(d:16,ERd,L)	B	4	5	6						D						@(d:16+ERd) <<2 (rotate) → @(d:16+ERd)			—	—	↑	↓	0	↓	
	ROTLB #2,@(d:32,Rd,B)	B	5	5	7						D						@(d:32+RdL) <<2 (rotate) → @(d:32+RdL)			—	—	↑	↓	0	↓	
	ROTLB #2,@(d:32,Rd,W)	B	5	5	7						D						@(d:32+Rd) <<2 (rotate) → @(d:32+Rd)			—	—	↑	↓	0	↓	
	ROTLB #2,@(d:32,ERd,L)	B	5	5	7						D						@(d:32+ERd) <<2 (rotate) → @(d:32+ERd)			—	—	↑	↓	0	↓	
	ROTLB #2,@aa:8	B	2	4	4								D				@aa:8 <<2 (rotate) → @aa:8			—	—	↑	↓	0	↓	
	ROTLB #2,@aa:16	B	3	4	5								D				@aa:16 <<2 (rotate) → @aa:16			—	—	↑	↓	0	↓	
	ROTLB #2,@aa:32	B	4	4	6								D				@aa:32 <<2 (rotate) → @aa:32			—	—	↑	↓	0	↓	
	ROTLW Rd	W	1	1	1	D											Rd16 <<1 (rotate) → Rd16			—	—	↑	↓	0	↓	
	ROTLW @ERd	W	2	4	4						D						@ERd <<1 (rotate) → @ERd			—	—	↑	↓	0	↓	
	ROTLW @ERd+	W	3	5	5							D					@ERd <<1 (rotate) → @ERd		ERd32+2→ERd32	—	—	↑	↓	0	↓	
	ROTLW @ERd-	W	3	5	5							D					@ERd <<1 (rotate) → @ERd		ERd32-2→ERd32	—	—	↑	↓	0	↓	
	ROTLW @+ERd	W	3	5	5							D			ERd32+2→ERd32		@ERd <<1 (rotate) → @ERd			—	—	↑	↓	0	↓	
	ROTLW @-ERd	W	3	5	5							D			ERd32-2→ERd32		@ERd <<1 (rotate) → @ERd			—	—	↑	↓	0	↓	

Shift (28)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode										Operation 1	Operation 2	Operation 3 ⁶			Operation 4	Operation 5	Condition Codes ^{3,2}										
				Min.	16-Bit Instruction Fetch Execution Stages ¹	16-Bit Instruction Fetch Execution Stages ¹	16-Bit Instruction Fetch Execution Stages ¹	16-Bit Instruction Fetch Execution Stages ¹	16-Bit Instruction Fetch Execution Stages ¹	16-Bit Instruction Fetch Execution Stages ¹	16-Bit Instruction Fetch Execution Stages ¹	16-Bit Instruction Fetch Execution Stages ¹	16-Bit Instruction Fetch Execution Stages ¹			16-Bit Instruction Fetch Execution Stages ¹	Operation 3	Operation 3			Operation 3	I	H	N	Z	V	C				
ROTL	ROTLW @(d:2,ERd)	W	3	5	5							D						@(2/4/6+ERd) <<1 (rotate) → @(2/4/6+ERd)						—	—	—	—	—	—	—	—
	ROTLW @(d:16,ERd)	W	4	5	6							D						@(d:16+ERd) <<1 (rotate) → @(d:16+ERd)						—	—	—	—	—	—	—	
	ROTLW @(d:32,ERd)	W	5	5	7							D						@(d:32+ERd) <<1 (rotate) → @(d:32+ERd)						—	—	—	—	—	—	—	
	ROTLW @(d:16,Rd,B)	W	4	5	6							D						@(d:16+Rd<<1) <<1 (rotate) → @(d:16+Rd<<1)						—	—	—	—	—	—	—	
	ROTLW @(d:16,Rd,W)	W	4	5	6							D						@(d:16+Rd<<1) <<1 (rotate) → @(d:16+Rd<<1)						—	—	—	—	—	—	—	
	ROTLW @(d:16,ERd,L)	W	4	5	6							D						@(d:16+ERd<<1) <<1 (rotate) → @(d:16+ERd<<1)						—	—	—	—	—	—	—	
	ROTLW @(d:32,Rd,B)	W	5	5	7							D						@(d:32+Rd<<1) <<1 (rotate) → @(d:32+Rd<<1)						—	—	—	—	—	—	—	
	ROTLW @(d:32,Rd,W)	W	5	5	7							D						@(d:32+Rd<<1) <<1 (rotate) → @(d:32+Rd<<1)						—	—	—	—	—	—	—	
	ROTLW @(d:32,ERd,L)	W	5	5	7							D						@(d:32+ERd<<1) <<1 (rotate) → @(d:32+ERd<<1)						—	—	—	—	—	—	—	
	ROTLW @aa:16	W	3	4	5								D					@aa:16 <<1 (rotate) → @aa:16						—	—	—	—	—	—	—	
	ROTLW @aa:32	W	4	4	6								D					@aa:32 <<1 (rotate) → @aa:32						—	—	—	—	—	—	—	
	ROTLW #2,Rd	W	1	1	1						D							Rd16 <<2 (rotate) → Rd16						—	—	—	—	—	—	—	
	ROTLW #2,@ERd	W	2	4	4							D						@ERd <<2 (rotate) → @ERd						—	—	—	—	—	—	—	
	ROTLW #2,@ERd+	W	3	5	5							D						@ERd <<2 (rotate) → @ERd				ERd32+2→ERd32		—	—	—	—	—	—	—	
	ROTLW #2,@ERd-	W	3	5	5							D						@ERd <<2 (rotate) → @ERd				ERd32-2→ERd32		—	—	—	—	—	—	—	
	ROTLW #2,@+ERd	W	3	5	5							D		ERd32+2→ERd32	@ERd	<<2 (rotate)	→ @ERd							—	—	—	—	—	—	—	
	ROTLW #2,@-ERd	W	3	5	5							D		ERd32-2→ERd32	@ERd	<<2 (rotate)	→ @ERd							—	—	—	—	—	—	—	
	ROTLW #2,@(d:2,ERd)	W	3	5	5							D						@(2/4/6+ERd) <<2 (rotate) → @(2/4/6+ERd)						—	—	—	—	—	—	—	
	ROTLW #2,@(d:16,ERd)	W	4	5	6							D						@(d:16+ERd) <<2 (rotate) → @(d:16+ERd)						—	—	—	—	—	—	—	
	ROTLW #2,@(d:32,ERd)	W	5	5	7							D						@(d:32+ERd) <<2 (rotate) → @(d:32+ERd)						—	—	—	—	—	—	—	
	ROTLW #2,@(d:16,Rd,B)	W	4	5	6							D						@(d:16+Rd<<1) <<2 (rotate) → @(d:16+Rd<<1)						—	—	—	—	—	—	—	
	ROTLW #2,@(d:16,Rd,W)	W	4	5	6							D						@(d:16+Rd<<1) <<2 (rotate) → @(d:16+Rd<<1)						—	—	—	—	—	—	—	
	ROTLW #2,@(d:16,ERd,L)	W	4	5	6							D						@(d:16+ERd<<1) <<2 (rotate) → @(d:16+ERd<<1)						—	—	—	—	—	—	—	
	ROTLW #2,@(d:32,Rd,B)	W	5	5	7							D						@(d:32+Rd<<1) <<2 (rotate) → @(d:32+Rd<<1)						—	—	—	—	—	—	—	
	ROTLW #2,@(d:32,Rd,W)	W	5	5	7							D						@(d:32+Rd<<1) <<2 (rotate) → @(d:32+Rd<<1)						—	—	—	—	—	—	—	

Shift (29)

Instruction	Mnemonic	Size	Instruction Length	Min.	Max.	Number of 16-Bit Instruction Fetch Execution Stages ¹ #xx	Addressing Mode					Operation ⁵					Condition Codes ²																			
							Rn	@ERn	@(d,ERn)	@(d,Rn,ERn,WERn,L)	@.ERn/@.ERn+/@.ERn+/@.ERn+/@.ERn	@.aa:0/@.aa:16/@.aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C													
ROTL	ROTLW #2,@(d:32,ERd,L)	W	5	5	7					D				@(d:32+ERd<<1) <<2 (rotate) → @(d:32+ERd<<1)																						
	ROTLW #2,@aa:16	W	3	4	5									@aa:16 <<2 (rotate) → @aa:16																						
	ROTLW #2,@aa:32	W	4	4	6									@aa:32 <<2 (rotate) → @aa:32																						
	ROTLLErd	L	1	1	1			D						ERd <<1 (rotate) → ERd																						
	ROTLLErd	L	3	5	5				D					@ERd <<1 (rotate) → @ERd																						
	ROTLLErd+	L	3	5	5					D				@ERd <<1 (rotate) → @ERd					ERd32+4 → ERd32																	
	ROTLLErd-	L	3	5	5					D				@ERd <<1 (rotate) → @ERd					ERd32-4 → ERd32																	
	ROTLLErd	L	3	5	5					D				ERd32+4 → ERd32 @ERd <<1 (rotate) → @ERd																						
	ROTLLErd-	L	3	5	5					D				@ERd <<1 (rotate) → @ERd					ERd32-4 → ERd32																	
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						
	ROTLLErd	L	3	5	5									@ERd <<1 (rotate) → @ERd																						



Shift (32)

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch Execution Stages ¹	Addressing Mode							Operation ⁵					Condition Codes ²									
						Number of	Fetch	Rn	@(d, ERn)	@(d, Rn, ERn, WERn, L)	@(Rn)/@ ERn+/@ ERn-/@ ERn	@aa:9/@ aa:16/@ aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z	V	C				
ROTR	ROTR.W @ERd+	W	3	5	5					D						@ERd	>>1 (rotate)	→ @ERd			ERd32+2→ERd32	—	—	↓	↓	0	↓
	ROTR.W @ERd-	W	3	5	5					D						@ERd	>>1 (rotate)	→ @ERd			ERd32-2→ERd32	—	—	↓	↓	0	↓
	ROTR.W @+ERd	W	3	5	5					D				ERd32+2→ERd32		@ERd	>>1 (rotate)	→ @ERd				—	—	↓	↓	0	↓
	ROTR.W @-ERd	W	3	5	5					D				ERd32-2→ERd32		@ERd	>>1 (rotate)	→ @ERd				—	—	↓	↓	0	↓
	ROTR.W @(d.2,ERd)	W	3	5	5					D						@(2/4/6+ERd)	>>1 (rotate)	→ @(2/4/6+ERd)				—	—	↓	↓	0	↓
	ROTR.W @(d.16,ERd)	W	4	5	6					D						@(d.16+ERd)	>>1 (rotate)	→ @(d.16+ERd)				—	—	↓	↓	0	↓
	ROTR.W @(d.32,ERd)	W	5	5	7					D						@(d.32+ERd)	>>1 (rotate)	→ @(d.32+ERd)				—	—	↓	↓	0	↓
	ROTR.W @(d.16,Rd.B)	W	4	5	6					D						@(d.16+Rd<<1)	>>1 (rotate)	→ @(d.16+Rd<<1)				—	—	↓	↓	0	↓
	ROTR.W @(d.16,Rd.W)	W	4	5	6					D						@(d.16+Rd<<1)	>>1 (rotate)	→ @(d.16+Rd<<1)				—	—	↓	↓	0	↓
	ROTR.W @(d.16,ERd.L)	W	4	5	6					D						@(d.16+ERd<<1)	>>1 (rotate)	→ @(d.16+ERd<<1)				—	—	↓	↓	0	↓
	ROTR.W @(d.32,Rd.B)	W	5	5	7					D						@(d.32+Rd<<1)	>>1 (rotate)	→ @(d.32+Rd<<1)				—	—	↓	↓	0	↓
	ROTR.W @(d.32,Rd.W)	W	5	5	7					D						@(d.32+Rd<<1)	>>1 (rotate)	→ @(d.32+Rd<<1)				—	—	↓	↓	0	↓
	ROTR.W @(d.32,ERd.L)	W	5	5	7					D						@(d.32+ERd<<1)	>>1 (rotate)	→ @(d.32+ERd<<1)				—	—	↓	↓	0	↓
	ROTR.W @aa:16	W	3	4	5						D					@aa:16	>>1 (rotate)	→ @aa:16				—	—	↓	↓	0	↓
	ROTR.W @aa:32	W	4	4	6							D				@aa:32	>>1 (rotate)	→ @aa:32				—	—	↓	↓	0	↓
	ROTR.W #2,Rd	W	1	1	1				D							Rd16	>>2 (rotate)	→ Rd16				—	—	↓	↓	0	↓
	ROTR.W #2,@ERd	W	2	4	4					D						@ERd	>>2 (rotate)	→ @ERd				—	—	↓	↓	0	↓
	ROTR.W #2,@ERd+	W	3	5	5					D						@ERd	>>2 (rotate)	→ @ERd			ERd32+2→ERd32	—	—	↓	↓	0	↓
	ROTR.W #2,@ERd-	W	3	5	5					D						@ERd	>>2 (rotate)	→ @ERd			ERd32-2→ERd32	—	—	↓	↓	0	↓
	ROTR.W #2,@+ERd	W	3	5	5					D				ERd32+2→ERd32		@ERd	>>2 (rotate)	→ @ERd				—	—	↓	↓	0	↓
	ROTR.W #2,@-ERd	W	3	5	5					D				ERd32-2→ERd32		@ERd	>>2 (rotate)	→ @ERd				—	—	↓	↓	0	↓
	ROTR.W #2,@(d.2,ERd)	W	3	5	5					D						@(2/4/6+ERd)	>>2 (rotate)	→ @(2/4/6+ERd)				—	—	↓	↓	0	↓
	ROTR.W #2,@(d.16,ERd)	W	4	5	6					D						@(d.16+ERd)	>>2 (rotate)	→ @(d.16+ERd)				—	—	↓	↓	0	↓
	ROTR.W #2,@(d.32,ERd)	W	5	5	7					D						@(d.32+ERd)	>>2 (rotate)	→ @(d.32+ERd)				—	—	↓	↓	0	↓
	ROTR.W #2,@(d.16,Rd.B)	W	4	5	6					D						@(d.16+Rd<<1)	>>2 (rotate)	→ @(d.16+Rd<<1)				—	—	↓	↓	0	↓



Shift (34)

Instruction	Mnemonic	Size	Instruction Length	Addressing Mode							Operation ⁵					Condition Codes ^{6,7}					
				Min.	16-Bit Instruction Fetch Execution Status ¹	FXZ	Rn	@(d,ERn)	@(d,ERn)	@(d,Rn,ERn,WERn,L)	@(d,ERn)/@ERn+/@ERn+/@ERn	@aa:9/@aa:16/@aa:32	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	I	H	N	Z
ROTR	ROTR.L #2,@ERd	L	3	5	5				D			@ERd >>2 (rotate) → @ERd			—	—	↓	↓	0	↓	
	ROTR.L #2,@ERd+	L	3	5	5					D		@ERd >>2 (rotate) → @ERd		ERd32+4→ERd32	—	—	↓	↓	0	↓	
	ROTR.L #2,@ERd-	L	3	5	5					D		@ERd >>2 (rotate) → @ERd		ERd32-4→ERd32	—	—	↓	↓	0	↓	
	ROTR.L #2,@+ERd	L	3	5	5					D		ERd32+4→ERd32 @ERd >>2 (rotate) → @ERd			—	—	↓	↓	0	↓	
	ROTR.L #2,@-ERd	L	3	5	5					D		ERd32-4→ERd32 @ERd >>2 (rotate) → @ERd			—	—	↓	↓	0	↓	
	ROTR.L #2,@(d:2,ERd)	L	3	5	5				D			@(4/8/12+ERd) >>2 (rotate) → @(4/8/12+ERd)			—	—	↓	↓	0	↓	
	ROTR.L #2,@(d:16,ERd)	L	4	5	6				D			@(d:16+ERd) >>2 (rotate) → @(d:16+ERd)			—	—	↓	↓	0	↓	
	ROTR.L #2,@(d:32,ERd)	L	5	5	7				D			@(d:32+ERd) >>2 (rotate) → @(d:32+ERd)			—	—	↓	↓	0	↓	
	ROTR.L #2,@(d:16,Rd,B)	L	4	5	6				D			@(d:16+RdL<<2) >>2 (rotate) → @(d:16+RdL<<2)			—	—	↓	↓	0	↓	
	ROTR.L #2,@(d:16,Rd,W)	L	4	5	6				D			@(d:16+Rd<<2) >>2 (rotate) → @(d:16+Pd<<2)			—	—	↓	↓	0	↓	
	ROTR.L #2,@(d:16,ERd,L)	L	4	5	6				D			@(d:16+ERd<<2) >>2 (rotate) → @(d:16+ERd<<2)			—	—	↓	↓	0	↓	
	ROTR.L #2,@(d:32,Rd,B)	L	5	5	7				D			@(d:32+RdL<<2) >>2 (rotate) → @(d:32+RdL<<2)			—	—	↓	↓	0	↓	
	ROTR.L #2,@(d:32,Rd,W)	L	5	5	7				D			@(d:32+Rd<<2) >>2 (rotate) → @(d:32+Pd<<2)			—	—	↓	↓	0	↓	
ROTR.L #2,@(d:32,ERd,L)	L	5	5	7				D			@(d:32+ERd<<2) >>2 (rotate) → @(d:32+ERd<<2)			—	—	↓	↓	0	↓		
ROTR.L #2,@aa:16	L	4	5	6					D		@aa:16 >>2 (rotate) → @aa:16			—	—	↓	↓	0	↓		
ROTR.L #2,@aa:32	L	5	5	7					D		@aa:32 >>2 (rotate) → @aa:32			—	—	↓	↓	0	↓		
ROTXL	ROTXL.B Rd	B	1	1	1				D			Rd8 <<1 (rotate with carry) → Rd8			—	—	↓	↓	0	↓	
	ROTXL.B @ERd	B	2	4	4				D			@ERd <<1 (rotate with carry) → @ERd			—	—	↓	↓	0	↓	
	ROTXL.B @ERd+	B	3	5	5					D		@ERd <<1 (rotate with carry) → @ERd		ERd32+1→ERd32	—	—	↓	↓	0	↓	
	ROTXL.B @ERd-	B	3	5	5					D		@ERd <<1 (rotate with carry) → @ERd		ERd32-1→ERd32	—	—	↓	↓	0	↓	
	ROTXL.B @+ERd	B	3	5	5					D		ERd32+1→ERd32 @ERd <<1 (rotate with carry) → @ERd			—	—	↓	↓	0	↓	
	ROTXL.B @-ERd	B	3	5	5					D		ERd32-1→ERd32 @ERd <<1 (rotate with carry) → @ERd			—	—	↓	↓	0	↓	
	ROTXL.B @(d:2,ERd)	B	3	5	5				D			@(1/2/3+ERd) <<1 (rotate with carry) → @(1/2/3+ERd)			—	—	↓	↓	0	↓	
	ROTXL.B @(d:16,ERd)	B	4	5	6				D			@(d:16+ERd) <<1 (rotate with carry) → @(d:16+ERd)			—	—	↓	↓	0	↓	
ROTXL.B @(d:32,ERd)	B	5	5	7				D			@(d:32+ERd) <<1 (rotate with carry) → @(d:32+ERd)			—	—	↓	↓	0	↓		



2.4 List of Instruction Codes

Table 2.2 shows the list of the instruction codes.

Table 2.2 List of Instruction Codes

Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension															
		15	8	7	0	15	8	7	0		15	8	7	0																
ADD	ADD.B #xx:8,Rd										8	rd	x	x	x	x	x	x	x	x										
	ADD.B #xx:8,@ERd					7	D	0	rd	0	0	0	0	8	0	0	0	0	x	x	x	x	x	x	x					
	ADD.B #xx:8,@ERd+	0	1	7	0	1	0	0	6	C	0	rd	1	0	0	0	0	8	0	0	0	0	x	x	x	x	x			
	ADD.B #xx:8,@ERd-	0	1	7	0	1	1	0	6	C	0	rd	1	0	0	0	0	8	0	0	0	0	x	x	x	x	x			
	ADD.B #xx:8,@+ERd	0	1	7	0	1	0	1	6	C	0	rd	1	0	0	0	0	8	0	0	0	0	x	x	x	x	x			
	ADD.B #xx:8,@-ERd	0	1	7	0	1	1	1	6	C	0	rd	1	0	0	0	0	8	0	0	0	0	x	x	x	x	x			
	ADD.B #xx:8,@(d:2,ERd)	0	1	7	0	1	d	d	6	8	0	rd	1	0	0	0	0	8	0	0	0	0	x	x	x	x	x			
	ADD.B #xx:8,@(d:16,ERd)	0	1	7	0	1	0	0	6	E	0	rd	1	0	0	0	d	8	0	0	0	0	x	x	x	x	x			
	ADD.B #xx:8,@(d:32,ERd)	7	8	0	rd	0	1	0	0	6	A	0	0	1	0	1	0	d	8	0	0	0	0	x	x	x	x	x		
	ADD.B #xx:8,@(d:16,Rd,B)	0	1	7	0	1	0	1	6	E	0	rd	1	0	0	0	d	8	0	0	0	0	x	x	x	x	x			
	ADD.B #xx:8,@(d:16,Rd,W)	0	1	7	0	1	1	0	6	E	0	rd	1	0	0	0	d	8	0	0	0	0	x	x	x	x	x			
	ADD.B #xx:8,@(d:16,ERd,L)	0	1	7	0	1	1	1	6	E	0	rd	1	0	0	0	d	8	0	0	0	0	x	x	x	x	x			
	ADD.B #xx:8,@(d:32,Rd,B)	7	8	0	rd	0	1	0	1	6	A	0	0	1	0	1	0	d	8	0	0	0	0	x	x	x	x	x		
	ADD.B #xx:8,@(d:32,Rd,W)	7	8	0	rd	0	1	1	0	6	A	0	0	1	0	1	0	d	8	0	0	0	0	x	x	x	x	x		
	ADD.B #xx:8,@(d:32,ERd,L)	7	8	0	rd	0	1	1	1	6	A	0	0	1	0	1	0	d	8	0	0	0	0	x	x	x	x	x		
	ADD.B #xx:8,@aa:8								7	F	a	a	a	a	a	a	a	8	0	0	0	0	x	x	x	x	x			
	ADD.B #xx:8,@aa:16								6	A	0	0	0	1	1	0	0	a	8	0	0	0	0	x	x	x	x	x		
	ADD.B #xx:8,@aa:32								6	A	0	0	1	1	1	0	0	a	a	8	0	0	0	0	x	x	x	x	x	
	ADD.B Rs,Rd																		0	8	rs	rd								
	ADD.B Rs,@ERd								7	D	0	rd	0	0	0	0	0			0	8	rs	0	0	0	0	0	0		
ADD.B Rs,@ERd+								0	1	7	1	0	0	1			1	0	0	0	0	rd	0	0	0	1	rs			
ADD.B Rs,@ERd-								0	1	7	1	0	0	1			1	0	1	0	0	rd	0	0	0	1	rs			
ADD.B Rs,@+ERd								0	1	7	1	0	0	1			1	0	0	1	0	rd	0	0	0	1	rs			
ADD.B Rs,@-ERd								0	1	7	1	0	0	1			1	0	1	1	0	rd	0	0	0	1	rs			
ADD.B Rs,@(d:2,ERd)								0	1	7	1	0	0	1			0	0	d	d	0	rd	0	0	0	1	rs			

Instruction	Mnemonic	Opcode				EA Ex- tension	Opcode				EA Extension																			
		15	8	7	0		15	8	7	0																				
ADD	ADD.B Rs, @(d:16,ERd)				0				1				1	1	1	0	0	0	1	rd					rs	d				
	ADD.B Rs, @(d:32,ERd)				0				1				1	1	1	0	0	1	rd					rs	d	d				
	ADD.B Rs, @(d:16,Rd.B)				0				1				1	1	0	1	0	rd					rs	d						
	ADD.B Rs, @(d:16,Rd.W)				0				1				1	1	1	0	0	rd					rs	d						
	ADD.B Rs, @(d:16,ERd.L)				0				1				1	1	1	1	0	rd					rs	d						
	ADD.B Rs, @(d:32,Rd.B)				0				1				1	1	0	1	1	rd					rs	d	d					
	ADD.B Rs, @(d:32,Rd.W)				0				1				1	1	1	1	0	rd					rs	d	d					
	ADD.B Rs, @(d:32,ERd.L)				0				1				1	1	1	1	1	rd					rs	d	d					
	ADD.B Rs, @aa:8				7		F	a	a	a	a	a	a	a	a	a	0		8	rs					0	0	0	0		
	ADD.B Rs, @aa:16				6		A	0	0	0	1			1	0	0	0	a		8	rs					0	0	0	0	
	ADD.B Rs, @aa:32				6		A	0	0	1	1			1	0	0	0	a	a		8	rs					0	0	0	0
	ADD.B @ERs,Rd				7		C	0	1	rs							0		8						0	0	0	0		
	ADD.B @ERs+,Rd				0				1				7				1	0	0	1	0						1	0	0	0
	ADD.B @ERs-,Rd				0				1				7				1	0	1	0	0						1	0	1	0
	ADD.B @+ERs,Rd				0				1				7				1	0	0	1	0						1	0	0	1
	ADD.B @-ERs,Rd				0				1				7				1	0	1	1	0						1	0	1	1
	ADD.B @(d:2,ERs),Rd				0				1				7				1	0	1	0	0						0	0	0	d
	ADD.B @(d:16,ERs),Rd				0				1				7				1	0	1	0	0						1	1	0	0
	ADD.B @(d:32,ERs),Rd				0				1				7				1	0	1	0	0						1	1	0	0
	ADD.B @(d:16,Rs.B),Rd				0				1				7				1	0	1	0	0						1	1	0	1
	ADD.B @(d:16,Rs.W),Rd				0				1				7				1	0	1	0	0						1	1	1	0
	ADD.B @(d:16,ERs.L),Rd				0				1				7				1	0	1	1	0						1	1	1	1
	ADD.B @(d:32,Rs.B),Rd				0				1				7				1	0	1	0	0						1	1	0	1
	ADD.B @(d:32,Rs.W),Rd				0				1				7				1	0	1	0	0						1	1	1	0
	ADD.B @(d:32,ERs.L),Rd				0				1				7				1	0	1	1	0						1	1	1	1
	ADD.B @aa:8,Rd				7		E	a	a	a	a	a	a	a	a	a	0		8						0	0	0	0		
	ADD.B @aa:16,Rd				6		A	0	0	0	1			0	0	0	0	a		8					0	0	0	0		
	ADD.B @aa:32,Rd				6		A	0	0	1	1			0	0	0	0	a	a		8					0	0	0	0	
	ADD.B @ERs, @ERd				7		C	0	1	rs							0	1	0	1	0					0	0	0	0	
	ADD.B @ERs, @ERd+				7		C	0	1	rs							0	1	0	1	0					1	0	0	0	
	ADD.B @ERs, @ERd-				7		C	0	1	rs							0	1	0	1	0					1	0	1	0	
	ADD.B @ERs, @+ERd				7		C	0	1	rs							0	1	0	1	0					1	0	0	1	



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
ADD	ADD.B @ERs, @-ERd				7	C	0	rs	0 1 0 1		1 0 1 1	0	rd	0 0 0 1	0 0 0 0		
	ADD.B @ERs, @(d:2,ERd)				7	C	0	rs	0 1 0 1		0 0 d d	0	rd	0 0 0 1	0 0 0 0		
	ADD.B @ERs, @(d:16,ERd)				7	C	0	rs	0 1 0 1		1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d	
	ADD.B @ERs, @(d:32,ERd)				7	C	0	rs	0 1 0 1		1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d d	
	ADD.B @ERs, @(d:16,Rd.B)				7	C	0	rs	0 1 0 1		1 1 0 1	0	rd	0 0 0 1	0 0 0 0	d	
	ADD.B @ERs, @(d:16,Rd.W)				7	C	0	rs	0 1 0 1		1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d	
	ADD.B @ERs, @(d:16,ERd.L)				7	C	0	rs	0 1 0 1		1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d	
	ADD.B @ERs, @(d:32,Rd.B)				7	C	0	rs	0 1 0 1		1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d d	
	ADD.B @ERs, @(d:32,Rd.W)				7	C	0	rs	0 1 0 1		1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d d	
	ADD.B @ERs, @(d:32,ERd.L)				7	C	0	rs	0 1 0 1		1 1 1 1	1	rd	0 0 0 1	0 0 0 0	d d	
	ADD.B @ERs, @aa:16				7	C	0	rs	0 1 0 1		0 1 0 0	0 0 0 0	0 0 0 1	0 0 0 0	a		
	ADD.B @ERs, @aa:32				7	C	0	rs	0 1 0 1		0 1 0 0	1 0 0 0	0 0 0 1	0 0 0 0	a a		
	ADD.B @ERs+, @ERd	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		0 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @ERs+, @ERd+	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @ERs+, @ERd-	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 0 1 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @ERs+, @+ERd	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @ERs+, @-ERd	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 0 1 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @ERs+, @(d:2,ERd)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		0 0 d d	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @ERs+, @(d:16,ERd)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @ERs+, @(d:32,ERd)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @ERs+, @(d:16,Rd.B)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 1 0 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @ERs+, @(d:16,Rd.W)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @ERs+, @(d:16,ERd.L)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @ERs+, @(d:32,Rd.B)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @ERs+, @(d:32,Rd.W)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @ERs+, @(d:32,ERd.L)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 1 1 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @ERs+, @aa:16	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		0 1 0 0	0 0 0 0	0 0 0 1	0 0 0 0	a	
	ADD.B @ERs+, @aa:32	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		0 1 0 0	1 0 0 0	0 0 0 1	0 0 0 0	a a	
	ADD.B @ERs-, @ERd	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		0 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @ERs-, @ERd+	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @ERs-, @ERd-	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 0 1 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @ERs-, @+ERd	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 0 1	0 0 0 0	

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
ADD	ADD.B @ERs, @-ERd	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @ERs, @(d:2,ERd)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		0 0 d d	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @ERs, @(d:16,ERd)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @ERs, @(d:32,ERd)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @ERs, @(d:16,Rd.B)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 1 0 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @ERs, @(d:16,Rd.W)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @ERs, @(d:16,ERd.L)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @ERs, @(d:32,Rd.B)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @ERs, @(d:32,Rd.W)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @ERs, @(d:32,ERd.L)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 1 1 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @ERs, @aa:16	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		0 1 0 0	0	0 0 0 0	0 0 0 1	0 0 0 0	a
	ADD.B @ERs, @aa:32	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		0 1 0 0	1	0 0 0 0	0 0 0 1	0 0 0 0	a a
	ADD.B @+ERs, @ERd	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		0 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @+ERs, @ERd+	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		1 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @+ERs, @ERd-	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		1 0 1 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @+ERs, @+ERd	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @+ERs, @-ERd	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		1 0 1 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @+ERs, @(d:2,ERd)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		0 0 d d	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @+ERs, @(d:16,ERd)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @+ERs, @(d:32,ERd)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @+ERs, @(d:16,Rd.B)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		1 1 0 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @+ERs, @(d:16,Rd.W)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @+ERs, @(d:16,ERd.L)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @+ERs, @(d:32,Rd.B)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @+ERs, @(d:32,Rd.W)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @+ERs, @(d:32,ERd.L)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		1 1 1 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @+ERs, @aa:16	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		0 1 0 0	0	0 0 0 0	0 0 0 1	0 0 0 0	a
	ADD.B @+ERs, @aa:32	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		0 1 0 0	1	0 0 0 0	0 0 0 1	0 0 0 0	a a
	ADD.B @-ERs, @ERd	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0		0 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @-ERs, @ERd+	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0		1 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @-ERs, @ERd-	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0		1 0 1 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @-ERs, @+ERd	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 0 1	0 0 0 0	



Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
ADD	ADD.B @-ERs, @-ERd	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @-ERs, @(d:2,ERd)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0		0 0 d d	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @-ERs, @(d:16,ERd)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0		1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @-ERs, @(d:32,ERd)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0		1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @-ERs, @(d:16,Rd.B)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0		1 1 0 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @-ERs, @(d:16,Rd.W)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0		1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @-ERs, @(d:16,ERd.L)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0		1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @-ERs, @(d:32,Rd.B)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0		1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @-ERs, @(d:32,Rd.W)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0		1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @-ERs, @(d:32,ERd.L)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0		1 1 1 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @-ERs, @aa:16	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0		0 1 0 0	0	0 0 0 0	0 0 0 1	0 0 0 0	a
	ADD.B @-ERs, @aa:32	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0		0 1 0 0	1	0 0 0 0	0 0 0 1	0 0 0 0	a a
	ADD.B @(d:2,ERs), @ERd	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		0 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:2,ERs), @ERd+	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:2,ERs), @ERd-	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 0 1 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:2,ERs), @+ERd	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:2,ERs), @-ERd	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 0 1 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:2,ERs), @(d:2,ERd)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		0 0 d d	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:2,ERs), @(d:16,ERd)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:2,ERs), @(d:32,ERd)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:2,ERs), @(d:16,Rd.B)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 1 0 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:2,ERs), @(d:16,Rd.W)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:2,ERs), @(d:16,ERd.L)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:2,ERs), @(d:32,Rd.B)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:2,ERs), @(d:32,Rd.W)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:2,ERs), @(d:32,ERd.L)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 1 1 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:2,ERs), @aa:16	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		0 1 0 0	0	0 0 0 0	0 0 0 1	0 0 0 0	a
	ADD.B @(d:2,ERs), @aa:32	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		0 1 0 0	1	0 0 0 0	0 0 0 1	0 0 0 0	a a
	ADD.B @(d:16,ERs), @ERd	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:16,ERs), @ERd+	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:16,ERs), @ERd-	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:16,ERs), @+ERd	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 0 0 1	0 0 0 0	

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension			
		15	8	7	0	15	8	7	0		15	8	7	0				
ADD	ADD.B @(d:16,ERs),@-ERd	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1	0 0 1 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:16,ERs),@(d:2,ERd)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	0	0 0 d d	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:16,ERs),@(d:16,ERd)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1	1 0 0 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:16,ERs),@(d:32,ERd)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1	1 0 0 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:16,ERs),@(d:16,Rd.B)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1	1 0 0 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:16,ERs),@(d:16,Rd.W)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1	1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:16,ERs),@(d:16,ERd.L)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1	1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:16,ERs),@(d:32,Rd.B)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1	1 0 0 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:16,ERs),@(d:32,Rd.W)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1	1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:16,ERs),@(d:32,ERd.L)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1	1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:16,ERs),@aa:16	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	0	1 0 0 0	0	0 0 0 0	0 0 0 1	0 0 0 0	a
	ADD.B @(d:16,ERs),@aa:32	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	0	1 0 0 0	1	0 0 0 0	0 0 0 1	0 0 0 0	a a
	ADD.B @(d:32,ERs),@ERd	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	0	0 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:32,ERs),@ERd+	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1	0 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:32,ERs),@ERd-	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1	0 1 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:32,ERs),@+ERd	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1	0 0 1 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:32,ERs),@-ERd	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1	0 1 1 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:32,ERs),@(d:2,ERd)	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	0	0 d d	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:32,ERs),@(d:16,ERd)	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1	1 0 0 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:32,ERs),@(d:32,ERd)	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1	1 0 0 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:32,ERs),@(d:16,Rd.B)	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1	1 0 0 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:32,ERs),@(d:16,Rd.W)	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1	1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:32,ERs),@(d:32,ERd.L)	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1	1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:32,ERs),@(d:32,Rd.B)	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1	1 0 0 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:32,ERs),@(d:32,Rd.W)	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1	1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:32,ERs),@(d:32,ERd.L)	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1	1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:32,ERs),@aa:16	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	0	1 0 0 0	0	0 0 0 0	0 0 0 1	0 0 0 0	a
	ADD.B @(d:32,ERs),@aa:32	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	0	1 0 0 0	1	0 0 0 0	0 0 0 1	0 0 0 0	a a
	ADD.B @(d:16,Rs.B),@ERd	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	0	0 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:16,Rs.B),@ERd+	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1	0 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:16,Rs.B),@ERd-	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1	0 1 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:16,Rs.B),@+ERd	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1	0 0 0 1	0	rd	0 0 0 1	0 0 0 0	



Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
ADD	ADD.B @(d:16,Rs.B),@-ERd	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:16,Rs.B),@(d:2,ERd)	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:16,Rs.B),@(d:16,ERd)	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:16,Rs.B),@(d:32,ERd)	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:16,Rs.B),@(d:16,Rd.B)	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:16,Rs.B),@(d:16,Rd.W)	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:16,Rs.B),@(d:16,ERd.L)	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:16,Rs.B),@(d:32,Rd.B)	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:16,Rs.B),@(d:32,Rd.W)	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:16,Rs.B),@(d:32,ERd.L)	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:16,Rs.B),@aa:16	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	0 1 0 0	0	0 0 0 0	0 0 0 1	0 0 0 0	a
	ADD.B @(d:16,Rs.B),@aa:32	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	0 1 0 0	1	0 0 0 0	0 0 0 1	0 0 0 0	a a
	ADD.B @(d:16,Rs.W),@ERd	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:16,Rs.W),@ERd+	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:16,Rs.W),@ERd-	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:16,Rs.W),@+ERd	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:16,Rs.W),@-ERd	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:16,Rs.W),@(d:2,ERd)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:16,Rs.W),@(d:16,ERd)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:16,Rs.W),@(d:32,ERd)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d d
ADD.B @(d:16,Rs.W),@(d:16,Rd.B)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 0 0 1	0 0 0 0	d	
ADD.B @(d:16,Rs.W),@(d:16,Rd.W)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d	
ADD.B @(d:16,Rs.W),@(d:16,ERd.L)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d	
ADD.B @(d:16,Rs.W),@(d:32,Rd.B)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d d	
ADD.B @(d:16,Rs.W),@(d:32,Rd.W)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d d	
ADD.B @(d:16,Rs.W),@(d:32,ERd.L)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 0 0 1	0 0 0 0	d d	
ADD.B @(d:16,Rs.W),@aa:16	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	0 1 0 0	0	0 0 0 0	0 0 0 1	0 0 0 0	a	
ADD.B @(d:16,Rs.W),@aa:32	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	0 1 0 0	1	0 0 0 0	0 0 0 1	0 0 0 0	a a	
ADD.B @(d:16,ERs.L),@ERd	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 0 0 1	0 0 0 0		
ADD.B @(d:16,ERs.L),@ERd+	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 0 0 1	0 0 0 0		
ADD.B @(d:16,ERs.L),@ERd-	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 0 0 1	0 0 0 0		

Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension			
		15	8	7	0	15	8	7	0		15	8	7	0				
ADD	ADD.B @(d:16,ERs.L),@+ERd	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1	0 0 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:16,ERs.L),@-ERd	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1	0 1 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:16,ERs.L),@(d:2,ERd)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	0	0 d d	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:16,ERs.L),@(d:16,ERd)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1	1 0 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:16,ERs.L),@(d:32,ERd)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1	1 0 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:16,ERs.L),@(d:16,Rd.B)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1	1 0 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:16,ERs.L),@(d:16,Rd.W)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1	1 1 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:16,ERs.L),@(d:16,ERd.L)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1	1 1 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:16,ERs.L),@(d:32,Rd.B)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1	1 0 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:16,ERs.L),@(d:32,Rd.W)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1	1 1 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:16,ERs.L),@(d:32,ERd.L)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1	1 1 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:16,ERs.L),@aa:16	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	0	1 0 0	0	0 0 0 0	0 0 0 1	0 0 0 0	a
	ADD.B @(d:16,ERs.L),@aa:32	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	0	1 0 0	1	0 0 0 0	0 0 0 1	0 0 0 0	a a
	ADD.B @(d:32,Rs.B),@ERd	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	0	0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:32,Rs.B),@ERd+	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:32,Rs.B),@ERd-	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	0 1 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:32,Rs.B),@+ERd	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	0 0 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:32,Rs.B),@-ERd	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	0 1 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:32,Rs.B),@(d:2,ERd)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	0	0 d d	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:32,Rs.B),@(d:16,ERd)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	1 0 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:32,Rs.B),@(d:32,ERd)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	1 0 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:32,Rs.B),@(d:16,Rd.B)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	1 0 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:32,Rs.B),@(d:16,Rd.W)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	1 1 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:32,Rs.B),@(d:16,ERd.L)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	1 1 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:32,Rs.B),@(d:32,Rd.B)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	1 0 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:32,Rs.B),@(d:32,Rd.W)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	1 1 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:32,Rs.B),@(d:32,ERd.L)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	1 1 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:32,Rs.B),@aa:16	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	0	1 0 0	0	0 0 0 0	0 0 0 1	0 0 0 0	a
	ADD.B @(d:32,Rs.B),@aa:32	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	0	1 0 0	1	0 0 0 0	0 0 0 1	0 0 0 0	a a
	ADD.B @(d:32,Rs.W),@ERd	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	0	0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:32,Rs.W),@ERd+	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	0	0 0 0	0	rd	0 0 0 1	0 0 0 0	



Instruction	Mnemonic	Opcode						EA Ex- tension	Opcode						EA Extension		
		15	8	7	0	15	8		7	0	15	8	7	0			
ADD	ADD.B @(d:32.Rs.W),@ERd+	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:32.Rs.W),@+ERd	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:32.Rs.W),@-ERd	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:32.Rs.W),@(d:2,ERd)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:32.Rs.W),@(d:16,ERd)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:32.Rs.W),@(d:32,ERd)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:32.Rs.W),@(d:16,Rd.B)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:32.Rs.W),@(d:16,Rd.W)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:32.Rs.W),@(d:16,ERd.L)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:32.Rs.W),@(d:32,Rd.B)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:32.Rs.W),@(d:32,Rd.W)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:32.Rs.W),@(d:32,ERd.L)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:32.Rs.W),@aa:16	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 0 0 1	0 0 0 0	a
	ADD.B @(d:32.Rs.W),@aa:32	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 0 0 1	0 0 0 0	a a
	ADD.B @(d:32,ERs.L),@ERd+	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:32,ERs.L),@ERd+	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:32,ERs.L),@ERd-	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:32,ERs.L),@+ERd	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:32,ERs.L),@-ERd	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:32,ERs.L),@(d:2,ERd)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 0 0 1	0 0 0 0	
	ADD.B @(d:32,ERs.L),@(d:16,ERd)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:32,ERs.L),@(d:32,ERd)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:32,ERs.L),@(d:16,Rd.B)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:32,ERs.L),@(d:16,Rd.W)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:32,ERs.L),@(d:16,ERd.L)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.B @(d:32,ERs.L),@(d:32,Rd.B)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:32,ERs.L),@(d:32,Rd.W)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.B @(d:32,ERs.L),@(d:32,ERd.L)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 0 0 1	0 0 0 0	d d
ADD.B @(d:32,ERs.L),@aa:16	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 0 0 1	0 0 0 0	a	
ADD.B @(d:32,ERs.L),@aa:32	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 0 0 1	0 0 0 0	a a	
ADD.B @aa:16,@ERd							6	A	0 0 0 1	0 1 0 1	a	0 0 0 0	0	rd	0 0 0 1	0 0 0 0	

Instruction	Mnemonic	Opcode				EA Ex- tension	Opcode				EA Extension																								
		15	8	7	0		15	8	7	0																									
ADD	ADD.B @aa:16,@ERd+				6	A	0	0	0	0	1	0	1	0	1	a	1	0	0	0	0	rd	0	0	0	0	1	0	0	0	0	0			
	ADD.B @aa:16,@ERd-				6	A	0	0	0	0	1	0	1	0	1	a	1	0	1	0	0	rd	0	0	0	0	1	0	0	0	0	0			
	ADD.B @aa:16,@+ERd				6	A	0	0	0	0	1	0	1	0	1	a	1	0	0	1	0	rd	0	0	0	0	1	0	0	0	0	0			
	ADD.B @aa:16,@-ERd				6	A	0	0	0	0	1	0	1	0	1	a	1	0	1	1	0	rd	0	0	0	0	1	0	0	0	0	0			
	ADD.B @aa:16,@(d:2,ERd)				6	A	0	0	0	0	1	0	1	0	1	a	0	0	d	d	0	rd	0	0	0	1	0	0	0	0	0				
	ADD.B @aa:16,@(d:16,ERd)				6	A	0	0	0	0	1	0	1	0	1	a	1	1	0	0	0	rd	0	0	0	1	0	0	0	0	0	d			
	ADD.B @aa:16,@(d:32,ERd)				6	A	0	0	0	0	1	0	1	0	1	a	1	1	0	0	1	rd	0	0	0	1	0	0	0	0	0	d d			
	ADD.B @aa:16,@(d:16,Rd,B)				6	A	0	0	0	0	1	0	1	0	1	a	1	1	0	1	0	rd	0	0	0	1	0	0	0	0	0	d			
	ADD.B @aa:16,@(d:16,Rd,W)				6	A	0	0	0	0	1	0	1	0	1	a	1	1	1	0	0	rd	0	0	0	1	0	0	0	0	0	d			
	ADD.B @aa:16,@(d:16,ERd,L)				6	A	0	0	0	0	1	0	1	0	1	a	1	1	1	1	0	rd	0	0	0	1	0	0	0	0	0	d			
	ADD.B @aa:16,@(d:32,Rd,B)				6	A	0	0	0	0	1	0	1	0	1	a	1	1	0	1	1	rd	0	0	0	1	0	0	0	0	0	d d			
	ADD.B @aa:16,@(d:32,Rd,W)				6	A	0	0	0	0	1	0	1	0	1	a	1	1	1	0	1	rd	0	0	0	1	0	0	0	0	0	d d			
	ADD.B @aa:16,@(d:32,ERd,L)				6	A	0	0	0	0	1	0	1	0	1	a	1	1	1	1	1	rd	0	0	0	1	0	0	0	0	0	d d			
	ADD.B @aa:16,@aa:16				6	A	0	0	0	0	1	0	1	0	1	a	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	a
	ADD.B @aa:16,@aa:32				6	A	0	0	0	0	1	0	1	0	1	a	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	a a
	ADD.B @aa:32,@ERd				6	A	0	0	1	1	0	1	0	1	a	a	0	0	0	0	0	rd	0	0	0	1	0	0	0	0	0				
	ADD.B @aa:32,@ERd+				6	A	0	0	1	1	0	1	0	1	a	a	1	0	0	0	0	rd	0	0	0	1	0	0	0	0	0				
	ADD.B @aa:32,@ERd-				6	A	0	0	1	1	0	1	0	1	a	a	1	0	1	0	0	rd	0	0	0	1	0	0	0	0	0				
	ADD.B @aa:32,@+ERd				6	A	0	0	1	1	0	1	0	1	a	a	1	0	0	1	0	rd	0	0	0	1	0	0	0	0	0				
	ADD.B @aa:32,@-ERd				6	A	0	0	1	1	0	1	0	1	a	a	1	0	1	1	0	rd	0	0	0	1	0	0	0	0	0				
	ADD.B @aa:32,@(d:2,ERd)				6	A	0	0	1	1	0	1	0	1	a	a	0	0	d	d	0	rd	0	0	0	1	0	0	0	0	0				
	ADD.B @aa:32,@(d:16,ERd)				6	A	0	0	1	1	0	1	0	1	a	a	1	1	0	0	0	rd	0	0	0	1	0	0	0	0	0	d			
	ADD.B @aa:32,@(d:32,ERd)				6	A	0	0	1	1	0	1	0	1	a	a	1	1	0	0	1	rd	0	0	0	1	0	0	0	0	0	d d			
	ADD.B @aa:32,@(d:16,Rd,B)				6	A	0	0	1	1	0	1	0	1	a	a	1	1	0	1	0	rd	0	0	0	1	0	0	0	0	0	d			
	ADD.B @aa:32,@(d:16,Rd,W)				6	A	0	0	1	1	0	1	0	1	a	a	1	1	1	0	0	rd	0	0	0	1	0	0	0	0	0	d			
	ADD.B @aa:32,@(d:16,ERd,L)				6	A	0	0	1	1	0	1	0	1	a	a	1	1	1	1	0	rd	0	0	0	1	0	0	0	0	0	d			
	ADD.B @aa:32,@(d:32,Rd,B)				6	A	0	0	1	1	0	1	0	1	a	a	1	1	0	1	1	rd	0	0	0	1	0	0	0	0	0	d d			
	ADD.B @aa:32,@(d:32,Rd,W)				6	A	0	0	1	1	0	1	0	1	a	a	1	1	1	0	1	rd	0	0	0	1	0	0	0	0	0	d d			
	ADD.B @aa:32,@(d:32,ERd,L)				6	A	0	0	1	1	0	1	0	1	a	a	1	1	1	1	1	rd	0	0	0	1	0	0	0	0	0	d d			
	ADD.B @aa:32,@aa:16				6	A	0	0	1	1	0	1	0	1	a	a	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	a
	ADD.B @aa:32,@aa:32				6	A	0	0	1	1	0	1	0	1	a	a	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	a a



Instruction	Mnemonic	Opcode			Opcode			EA Extension	Opcode			EA Extension							
		15	8	7	0	15	8		7	0	15		8	7	0				
ADD	ADD.W #xx:3,Rd								0	A	0	x	x	x	rd				
	ADD.W #xx:16,Rd								7	9	0	0	0	1	rd		x		
	ADD.W #xx:3,@ERd			7	D	1	rd	0	0	0	0	0	0	0	0	0	0	0	0
	ADD.W #xx:3,@aa:16			6	B	0	0	0	1	1	0	0	0	0	a	0	0	0	0
	ADD.W #xx:3,@aa:32			6	B	0	0	1	1	1	0	0	0	0	a	a	0	0	0
	ADD.W #xx:16,@ERd			0	1	5	1	1	1	0	0	0	0	0	rd	0	0	0	0
	ADD.W #xx:16,@ERd+			0	1	5	1	1	1	0	1	0	0	0	rd	0	0	0	0
	ADD.W #xx:16,@ERd-			0	1	5	1	1	1	0	1	0	1	0	rd	0	0	0	0
	ADD.W #xx:16,@+ERd			0	1	5	1	1	1	0	1	0	0	1	rd	0	0	0	0
	ADD.W #xx:16,@-ERd			0	1	5	1	1	1	0	1	0	1	0	rd	0	0	0	0
	ADD.W #xx:16,@(d:2,ERd)			0	1	5	1	1	1	0	0	d	d	0	rd	0	0	0	0
	ADD.W #xx:16,@(d:16,ERd)			0	1	5	1	1	1	0	1	1	0	0	rd	0	0	0	0
	ADD.W #xx:16,@(d:32,ERd)			0	1	5	1	1	1	0	1	1	0	0	rd	0	0	0	0
	ADD.W #xx:16,@(d:16,Rd,B)			0	1	5	1	1	1	0	1	1	0	1	rd	0	0	0	0
	ADD.W #xx:16,@(d:16,Rd,W)			0	1	5	1	1	1	0	1	1	1	0	rd	0	0	0	0
	ADD.W #xx:16,@(d:16,ERd,L)			0	1	5	1	1	1	0	1	1	1	0	rd	0	0	0	0
	ADD.W #xx:16,@(d:32,Rd,B)			0	1	5	1	1	1	0	1	1	0	1	rd	0	0	0	0
	ADD.W #xx:16,@(d:32,Rd,W)			0	1	5	1	1	1	0	1	1	1	0	rd	0	0	0	0
	ADD.W #xx:16,@(d:32,ERd,L)			0	1	5	1	1	1	0	1	1	1	0	rd	0	0	0	0
	ADD.W #xx:16,@aa:16			0	1	5	1	1	1	0	0	1	0	0	0	0	0	0	0
	ADD.W #xx:16,@aa:32			0	1	5	1	1	1	0	0	1	0	0	0	0	0	0	0
	ADD.W Rs,Rd									0	9				rs				
	ADD.W Rs,@ERd			7	D	1	rd	0	0	0	0	0	0	0					
	ADD.W Rs,@ERd+			0	1	5	1	0	0	1	1	0	0	0	rd	0	0	0	1
	ADD.W Rs,@ERd-			0	1	5	1	0	0	1	1	0	1	0	rd	0	0	0	1
	ADD.W Rs,@+ERd			0	1	5	1	0	0	1	1	0	0	1	rd	0	0	0	1
	ADD.W Rs,@-ERd			0	1	5	1	0	0	1	1	0	1	0	rd	0	0	0	1
ADD.W Rs,@(d:2,ERd)			0	1	5	1	0	0	1	0	0	d	d	rd	0	0	0	1	
ADD.W Rs,@(d:16,ERd)			0	1	5	1	0	0	1	1	1	0	0	rd	0	0	0	1	
ADD.W Rs,@(d:32,ERd)			0	1	5	1	0	0	1	1	1	0	0	rd	0	0	0	1	
ADD.W Rs,@(d:16,Rd,B)			0	1	5	1	0	0	1	1	1	0	1	rd	0	0	0	1	

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0		15	8	7	0			
ADD	ADD.W Rs, @(d:16,Rd,W)	0	1	5	1 0 0 1		1 1 1 0	0	rd	0 0 0 1	rs	d	
	ADD.W Rs, @(d:16,ERd,L)	0	1	5	1 0 0 1		1 1 1 1	0	rd	0 0 0 1	rs	d	
	ADD.W Rs, @(d:32,Rd,B)	0	1	5	1 0 0 1		1 1 0 1	1	rd	0 0 0 1	rs	d d	
	ADD.W Rs, @(d:32,Rd,W)	0	1	5	1 0 0 1		1 1 1 0	1	rd	0 0 0 1	rs	d d	
	ADD.W Rs, @(d:32,ERd,L)	0	1	5	1 0 0 1		1 1 1 1	1	rd	0 0 0 1	rs	d d	
	ADD.W Rs, @aa:16	6	B	0 0 0 1	1 0 0 0	a	0	9	rs	0 0 0 0			
	ADD.W Rs, @aa:32	6	B	0 0 1 1	1 0 0 0	a a	0	9	rs	0 0 0 0			
	ADD.W @ERs,Rd	7	C	1 rs	0 0 0 0		0	9	0 0 0 0	rd			
	ADD.W @ERs+,Rd	0	1	5	1 0 1 0		1 0 0 0	0	rs	0 0 0 1	rd		
	ADD.W @ERs-,Rd	0	1	5	1 0 1 0		1 0 1 0	0	rs	0 0 0 1	rd		
	ADD.W @+ERs,Rd	0	1	5	1 0 1 0		1 0 0 1	0	rs	0 0 0 1	rd		
	ADD.W @-ERs,Rd	0	1	5	1 0 1 0		1 0 1 1	0	rs	0 0 0 1	rd		
	ADD.W @(d:2,ERs),Rd	0	1	5	1 0 1 0		0 0 d d	0	rs	0 0 0 1	rd		
	ADD.W @(d:16,ERs),Rd	0	1	5	1 0 1 0		1 1 0 0	0	rs	0 0 0 1	rd	d	
	ADD.W @(d:32,ERs),Rd	0	1	5	1 0 1 0		1 1 0 0	1	rs	0 0 0 1	rd	d d	
	ADD.W @(d:16,Rs,B),Rd	0	1	5	1 0 1 0		1 1 0 1	0	rs	0 0 0 1	rd	d	
	ADD.W @(d:16,Rs,W),Rd	0	1	5	1 0 1 0		1 1 1 0	0	rs	0 0 0 1	rd	d	
	ADD.W @(d:16,ERs,L),Rd	0	1	5	1 0 1 0		1 1 1 1	0	rs	0 0 0 1	rd	d	
	ADD.W @(d:32,Rs,B),Rd	0	1	5	1 0 1 0		1 1 0 1	1	rs	0 0 0 1	rd	d d	
	ADD.W @(d:32,Rs,W),Rd	0	1	5	1 0 1 0		1 1 1 0	1	rs	0 0 0 1	rd	d d	
	ADD.W @(d:32,ERs,L),Rd	0	1	5	1 0 1 0		1 1 1 1	1	rs	0 0 0 1	rd	d d	
	ADD.W @aa:16,Rd	6	B	0 0 0 1	0 0 0 0	a	0	9	0 0 0 0	rd			
	ADD.W @aa:32,Rd	6	B	0 0 1 1	0 0 0 0	a a	0	9	0 0 0 0	rd			
	ADD.W @ERs, @ERd	7	C	1 rs	0 1 0 1		0 0 0 0	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @ERs, @ERd+	7	C	1 rs	0 1 0 1		1 0 0 0	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @ERs, @ERd-	7	C	1 rs	0 1 0 1		1 0 1 0	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @ERs, @+ERd	7	C	1 rs	0 1 0 1		1 0 0 1	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @ERs, @-ERd	7	C	1 rs	0 1 0 1		1 0 1 1	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @ERs, @(d:2,ERd)	7	C	1 rs	0 1 0 1		0 0 d d	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @ERs, @(d:16,ERd)	7	C	1 rs	0 1 0 1		1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d	
	ADD.W @ERs, @(d:32,ERd)	7	C	1 rs	0 1 0 1		1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d d	



Instruction	Mnemonic	Opcode					Opcode					EA Extension	Opcode					EA Extension											
		15	8	7	0	15	8	7	0	15	8		7	0	EA Extension														
ADD	ADD.W @ERs,@(d:16,Rd,B)					7	C	1	rs	0	1	0	1	1	1	0	1	0	rd	0	0	0	1	0	0	0	0	d	
	ADD.W @ERs,@(d:16,Rd,W)					7	C	1	rs	0	1	0	1	1	1	1	0	0	rd	0	0	0	1	0	0	0	0	d	
	ADD.W @ERs,@(d:16,ERd,L)					7	C	1	rs	0	1	0	1	1	1	1	0	0	rd	0	0	0	1	0	0	0	0	d	
	ADD.W @ERs,@(d:32,Rd,B)					7	C	1	rs	0	1	0	1	1	1	0	1	1	rd	0	0	0	1	0	0	0	0	d d	
	ADD.W @ERs,@(d:32,Rd,W)					7	C	1	rs	0	1	0	1	1	1	0	1	1	rd	0	0	0	1	0	0	0	0	d d	
	ADD.W @ERs,@(d:32,ERd,L)					7	C	1	rs	0	1	0	1	1	1	1	1	1	rd	0	0	0	1	0	0	0	0	d d	
	ADD.W @ERs,@aa:16					7	C	1	rs	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	a
	ADD.W @ERs,@aa:32					7	C	1	rs	0	1	0	1	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	a a
	ADD.W @ERs+,@ERd	0		1		5			0	1	0	0	6	D	0	rs	1	1	0	0	0	0	0	0	0	0	0	0	
	ADD.W @ERs+,@ERd+	0		1		5			0	1	0	0	6	D	0	rs	1	1	0	0	1	0	0	0	0	0	0	0	
	ADD.W @ERs+,@ERd-	0		1		5			0	1	0	0	6	D	0	rs	1	1	0	0	1	0	0	0	0	0	0	0	
	ADD.W @ERs+,@+ERd	0		1		5			0	1	0	0	6	D	0	rs	1	1	0	0	1	0	0	0	0	0	0	0	
	ADD.W @ERs+,@-ERd	0		1		5			0	1	0	0	6	D	0	rs	1	1	0	0	1	0	0	0	0	0	0	0	
	ADD.W @ERs+,@(d:2,ERd)	0		1		5			0	1	0	0	6	D	0	rs	1	1	0	0	0	0	d	d	0	0	0	0	
	ADD.W @ERs+,@(d:16,ERd)	0		1		5			0	1	0	0	6	D	0	rs	1	1	0	0	1	0	0	0	0	0	0	0	d
	ADD.W @ERs+,@(d:32,ERd)	0		1		5			0	1	0	0	6	D	0	rs	1	1	0	0	1	0	0	0	0	0	0	0	d d
	ADD.W @ERs+,@(d:16,Rd,B)	0		1		5			0	1	0	0	6	D	0	rs	1	1	0	0	1	0	0	1	0	0	0	0	
	ADD.W @ERs+,@(d:16,Rd,W)	0		1		5			0	1	0	0	6	D	0	rs	1	1	0	0	1	0	0	1	0	0	0	0	d
	ADD.W @ERs+,@(d:16,ERd,L)	0		1		5			0	1	0	0	6	D	0	rs	1	1	0	0	1	0	0	1	0	0	0	0	d
	ADD.W @ERs+,@(d:32,Rd,B)	0		1		5			0	1	0	0	6	D	0	rs	1	1	0	0	1	0	1	1	0	0	0	0	d d
	ADD.W @ERs+,@(d:32,Rd,W)	0		1		5			0	1	0	0	6	D	0	rs	1	1	0	0	1	0	1	1	0	0	0	0	d d
	ADD.W @ERs+,@(d:32,ERd,L)	0		1		5			0	1	0	0	6	D	0	rs	1	1	0	0	1	0	1	1	1	0	0	0	d d
	ADD.W @ERs+,@aa:16	0		1		5			0	1	0	0	6	D	0	rs	1	1	0	0	0	1	0	0	0	0	0	0	a
	ADD.W @ERs+,@aa:32	0		1		5			0	1	0	0	6	D	0	rs	1	1	0	0	0	1	0	0	0	0	0	0	a a
	ADD.W @ERs-,@ERd	0		1		5			0	1	1	0	6	D	0	rs	1	1	0	0	0	0	0	0	1	0	0	0	
	ADD.W @ERs-,@ERd+	0		1		5			0	1	1	0	6	D	0	rs	1	1	0	0	0	0	0	1	0	0	0	0	
	ADD.W @ERs-,@ERd-	0		1		5			0	1	1	0	6	D	0	rs	1	1	0	0	0	0	1	0	0	0	0	0	
	ADD.W @ERs-,@+ERd	0		1		5			0	1	1	0	6	D	0	rs	1	1	0	0	0	0	1	0	0	0	0	0	
	ADD.W @ERs-,@-ERd	0		1		5			0	1	1	0	6	D	0	rs	1	1	0	0	0	0	1	0	0	0	0	0	
	ADD.W @ERs-,@(d:2,ERd)	0		1		5			0	1	1	0	6	D	0	rs	1	1	0	0	0	0	d	d	0	0	0	0	
ADD.W @ERs-,@(d:16,ERd)	0		1		5			0	1	1	0	6	D	0	rs	1	1	0	0	0	0	1	0	0	0	0	0	d	
ADD.W @ERs-,@(d:32,ERd)	0		1		5			0	1	1	0	6	D	0	rs	1	1	0	0	0	0	1	0	0	0	0	0	d d	
ADD.W @ERs-,@(d:16,Rd,B)	0		1		5			0	1	1	0	6	D	0	rs	1	1	0	0	0	0	1	0	0	0	0	0	d	
ADD.W @ERs-,@(d:16,Rd,W)	0		1		5			0	1	1	0	6	D	0	rs	1	1	0	0	0	0	1	0	0	0	0	0	d	

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
ADD	ADD.W @ERs, @(d:16,RdL)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.W @ERs, @(d:32,Rd.B)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.W @ERs, @(d:32,Rd.W)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.W @ERs, @(d:32,ERd.L)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 1 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.W @ERs, @aa:16	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		0 1 0 0	0	0 0 0 0	0 0 0 1	0 0 0 0	a
	ADD.W @ERs, @aa:32	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		0 1 0 0	1	0 0 0 0	0 0 0 1	0 0 0 0	a a
	ADD.W @+ERs,@ERd	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		0 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.W @+ERs,@ERd+	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.W @+ERs,@ERd-	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 0 1 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.W @+ERs,@+ERd	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.W @+ERs,@-ERd	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 0 1 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.W @+ERs,@(d:2,ERd)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		0 0 d d	0	rd	0 0 0 1	0 0 0 0	
	ADD.W @+ERs,@(d:16,ERd)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.W @+ERs,@(d:32,ERd)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.W @+ERs,@(d:16,Rd.B)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 0 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.W @+ERs,@(d:16,Rd.W)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.W @+ERs,@(d:16,ERd.L)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.W @+ERs,@(d:32,Rd.B)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.W @+ERs,@(d:32,Rd.W)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.W @+ERs,@(d:32,ERd.L)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.W @+ERs,@aa:16	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		0 1 0 0	0	0 0 0 0	0 0 0 1	0 0 0 0	a
	ADD.W @+ERs,@aa:32	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		0 1 0 0	1	0 0 0 0	0 0 0 1	0 0 0 0	a a
	ADD.W @-ERs,@ERd	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		0 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.W @-ERs,@ERd+	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.W @-ERs,@ERd-	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 1 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.W @-ERs,@+ERd	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.W @-ERs,@-ERd	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 1 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.W @-ERs,@(d:2,ERd)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		0 0 d d	0	rd	0 0 0 1	0 0 0 0	
	ADD.W @-ERs,@(d:16,ERd)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.W @-ERs,@(d:32,ERd)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.W @-ERs,@(d:16,Rd.B)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.W @-ERs,@(d:16,Rd.W)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d
ADD.W @-ERs,@(d:16,ERd.L)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d	
ADD.W @-ERs,@(d:32,Rd.B)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d d	



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension			
		15	8	7	0	15	8	7	0		15	8	7	0				
ADD	ADD.W @-ERs,@(d:32,Rd,W)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d	d
	ADD.W @-ERs,@(d:32,ERd,L)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 1	1	rd	0 0 0 1	0 0 0 0	d	d
	ADD.W @-ERs,@aa:16	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		0 1 0 0	0	0 0 0 0	0 0 0 1	0 0 0 0	a	
	ADD.W @-ERs,@aa:32	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		0 1 0 0	1	0 0 0 0	0 0 0 1	0 0 0 0	a	a
	ADD.W @(d:2,ERs),@ERd	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		0 0 0 0	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @(d:2,ERs),@ERd+	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 0 0 0	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @(d:2,ERs),@ERd-	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 0 1 0	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @(d:2,ERs),@+ERd	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @(d:2,ERs),@-ERd	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 0 1 1	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @(d:2,ERs),@(d:2,ERd)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		0 0 d d	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @(d:2,ERs),@(d:16,ERd)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d	
	ADD.W @(d:2,ERs),@(d:32,ERd)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d	d
	ADD.W @(d:2,ERs),@(d:16,Rd,B)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 1	0	rd	0 0 0 1	0 0 0 0	d	
	ADD.W @(d:2,ERs),@(d:16,Rd,W)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d	
	ADD.W @(d:2,ERs),@(d:16,ERd,L)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d	
	ADD.W @(d:2,ERs),@(d:32,Rd,B)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d	d
	ADD.W @(d:2,ERs),@(d:32,Rd,W)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d	d
	ADD.W @(d:2,ERs),@(d:32,ERd,L)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 1 1	1	rd	0 0 0 1	0 0 0 0	d	d
	ADD.W @(d:2,ERs),@aa:16	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		0 1 0 0	0	0 0 0 0	0 0 0 1	0 0 0 0	a	
	ADD.W @(d:2,ERs),@aa:32	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		0 1 0 0	1	0 0 0 0	0 0 0 1	0 0 0 0	a	a
	ADD.W @(d:16,ERs),@ERd	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @(d:16,ERs),@ERd+	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @(d:16,ERs),@ERd-	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @(d:16,ERs),@+ERd	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @(d:16,ERs),@-ERd	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @(d:16,ERs),@(d:2,ERd)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @(d:16,ERs),@(d:16,ERd)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d	
	ADD.W @(d:16,ERs),@(d:32,ERd)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d	d
ADD.W @(d:16,ERs),@(d:16,Rd,B)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 0 0 1	0 0 0 0	d		
ADD.W @(d:16,ERs),@(d:16,Rd,W)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d		
ADD.W @(d:16,ERs),@(d:16,ERd,L)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d		
ADD.W @(d:16,ERs),@(d:32,Rd,B)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d	d	
ADD.W @(d:16,ERs),@(d:32,Rd,W)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d	d	
ADD.W @(d:16,ERs),@(d:32,ERd,L)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 0 0 1	0 0 0 0	d	d	

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension
		15	8	7	0	15	8	7	0		15	8	7	0	
ADD	ADD.W @(d:16,ERs),@aa:16	0	1	5	0 1 0 0	6	F	0 rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 0 0 1	0 0 0 0	a
	ADD.W @(d:16,ERs),@aa:32	0	1	5	0 1 0 0	6	F	0 rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 0 0 1	0 0 0 0	a a
	ADD.W @(d:32,ERs),@ERd	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0 rd	0 0 0 1	0 0 0 0
	ADD.W @(d:32,ERs),@ERd+	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0 rd	0 0 0 1	0 0 0 0
	ADD.W @(d:32,ERs),@ERd-	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0 rd	0 0 0 1	0 0 0 0
	ADD.W @(d:32,ERs),@+ERd	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0 rd	0 0 0 1	0 0 0 0
	ADD.W @(d:32,ERs),@-ERd	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0 rd	0 0 0 1	0 0 0 0
	ADD.W @(d:32,ERs),@(d:2,ERd)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0 rd	0 0 0 1	0 0 0 0
	ADD.W @(d:32,ERs),@(d:16,ERd)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0 rd	0 0 0 1	0 0 0 0
	ADD.W @(d:32,ERs),@(d:32,ERd)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1 rd	0 0 0 1	0 0 0 0
	ADD.W @(d:32,ERs),@(d:16,Rd,B)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0 rd	0 0 0 1	0 0 0 0
	ADD.W @(d:32,ERs),@(d:16,Rd,W)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0 rd	0 0 0 1	0 0 0 0
	ADD.W @(d:32,ERs),@(d:16,ERd,L)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0 rd	0 0 0 1	0 0 0 0
	ADD.W @(d:32,ERs),@(d:32,Rd,B)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1 rd	0 0 0 1	0 0 0 0
	ADD.W @(d:32,ERs),@(d:32,Rd,W)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1 rd	0 0 0 1	0 0 0 0
	ADD.W @(d:32,ERs),@(d:32,ERd,L)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1 rd	0 0 0 1	0 0 0 0
	ADD.W @(d:32,ERs),@aa:16	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0 0 0 0	0 0 0 1	0 0 0 0
	ADD.W @(d:32,ERs),@aa:32	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1 0 0 0	0 0 0 1	0 0 0 0
	ADD.W @(d:16,Rs,B),@ERd	0	1	5	0 1 0 1	6	F	0 rs	1 1 0 0	d	0 0 0 0	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:16,Rs,B),@ERd+	0	1	5	0 1 0 1	6	F	0 rs	1 1 0 0	d	1 0 0 0	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:16,Rs,B),@ERd-	0	1	5	0 1 0 1	6	F	0 rs	1 1 0 0	d	1 0 1 0	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:16,Rs,B),@+ERd	0	1	5	0 1 0 1	6	F	0 rs	1 1 0 0	d	1 0 0 1	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:16,Rs,B),@-ERd	0	1	5	0 1 0 1	6	F	0 rs	1 1 0 0	d	1 0 1 1	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:16,Rs,B),@(d:2,ERd)	0	1	5	0 1 0 1	6	F	0 rs	1 1 0 0	d	0 0 d d	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:16,Rs,B),@(d:16,ERd)	0	1	5	0 1 0 1	6	F	0 rs	1 1 0 0	d	1 1 0 0	0 rd	0 0 0 1	0 0 0 0	d
	ADD.W @(d:16,Rs,B),@(d:32,ERd)	0	1	5	0 1 0 1	6	F	0 rs	1 1 0 0	d	1 1 0 0	1 rd	0 0 0 1	0 0 0 0	d d
	ADD.W @(d:16,Rs,B),@(d:16,Rd,B)	0	1	5	0 1 0 1	6	F	0 rs	1 1 0 0	d	1 1 0 1	0 rd	0 0 0 1	0 0 0 0	d
	ADD.W @(d:16,Rs,B),@(d:16,Rd,W)	0	1	5	0 1 0 1	6	F	0 rs	1 1 0 0	d	1 1 1 0	0 rd	0 0 0 1	0 0 0 0	d
	ADD.W @(d:16,Rs,B),@(d:16,ERd,L)	0	1	5	0 1 0 1	6	F	0 rs	1 1 0 0	d	1 1 1 1	0 rd	0 0 0 1	0 0 0 0	d
	ADD.W @(d:16,Rs,B),@(d:32,Rd,B)	0	1	5	0 1 0 1	6	F	0 rs	1 1 0 0	d	1 1 0 1	1 rd	0 0 0 1	0 0 0 0	d d
ADD.W @(d:16,Rs,B),@(d:32,Rd,W)	0	1	5	0 1 0 1	6	F	0 rs	1 1 0 0	d	1 1 1 0	1 rd	0 0 0 1	0 0 0 0	d d	
ADD.W @(d:16,Rs,B),@(d:32,ERd,L)	0	1	5	0 1 0 1	6	F	0 rs	1 1 0 0	d	1 1 1 1	1 rd	0 0 0 1	0 0 0 0	d d	
ADD.W @(d:16,Rs,B),@aa:16	0	1	5	0 1 0 1	6	F	0 rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 0 0 1	0 0 0 0	a	



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
ADD	ADD.W @(d:16,Rs.B),@aa:32	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 0 0 1	0 0 0 0	a a
	ADD.W @(d:16,Rs.W),@ERd	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 0 0 1	0 0 0 00
	ADD.W @(d:16,Rs.W),@ERd+	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 0 0 1	0 0 0 0 0
	ADD.W @(d:16,Rs.W),@ERd-	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 0 0 1	0 0 0 0 0
	ADD.W @(d:16,Rs.W),@+ERd	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 0 0 1	0 0 0 0 0
	ADD.W @(d:16,Rs.W),@-ERd	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 0 0 1	0 0 0 0 0
	ADD.W @(d:16,Rs.W),@(d:2,ERd)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 0 0 1	0 0 0 0 0
	ADD.W @(d:16,Rs.W),@(d:16,ERd)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 0 0 1	0 0 0 0 0 d
	ADD.W @(d:16,Rs.W),@(d:32,ERd)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 0 0 1	0 0 0 0 0 d d
	ADD.W @(d:16,Rs.W),@(d:16,Rd.B)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 0 0 1	0 0 0 0 0 d
	ADD.W @(d:16,Rs.W),@(d:16,Rd.W)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 0 0 1	0 0 0 0 0 d
	ADD.W @(d:16,Rs.W),@(d:16,ERd.L)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 0 0 1	0 0 0 0 0 d
	ADD.W @(d:16,Rs.W),@(d:32,Rd.B)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 0 0 1	0 0 0 0 0 d d
	ADD.W @(d:16,Rs.W),@(d:32,Rd.W)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 0 0 1	0 0 0 0 0 d d
	ADD.W @(d:16,Rs.W),@(d:32,ERd.L)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 0 0 1	0 0 0 0 0 d d
	ADD.W @(d:16,Rs.W),@aa:16	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 0 0 1	0 0 0 0 0 a	
	ADD.W @(d:16,Rs.W),@aa:32	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 0 0 1	0 0 0 0 0 a a	
	ADD.W @(d:16,ERs.L),@ERd	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 0 0 1	0 0 0 0 0
	ADD.W @(d:16,ERs.L),@ERd+	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 0 0 1	0 0 0 0 0
	ADD.W @(d:16,ERs.L),@ERd-	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 0 0 1	0 0 0 0 0
	ADD.W @(d:16,ERs.L),@-ERd	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 0 0 1	0 0 0 0 0
	ADD.W @(d:16,ERs.L),@+ERd	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 0 0 1	0 0 0 0 0
	ADD.W @(d:16,ERs.L),@(d:2,ERd)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 0 0 1	0 0 0 0 0
	ADD.W @(d:16,ERs.L),@(d:16,ERd)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 0 0 1	0 0 0 0 0 d
	ADD.W @(d:16,ERs.L),@(d:32,ERd)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 0 0 1	0 0 0 0 0 d d
	ADD.W @(d:16,ERs.L),@(d:16,Rd.B)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 0 0 1	0 0 0 0 0 d
	ADD.W @(d:16,ERs.L),@(d:16,Rd.W)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 0 0 1	0 0 0 0 0 d
	ADD.W @(d:16,ERs.L),@(d:16,ERd.L)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 0 0 1	0 0 0 0 0 d
	ADD.W @(d:16,ERs.L),@(d:32,Rd.B)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 0 0 1	0 0 0 0 0 d d
	ADD.W @(d:16,ERs.L),@(d:32,Rd.W)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 0 0 1	0 0 0 0 0 d d
	ADD.W @(d:16,ERs.L),@(d:32,ERd.L)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 0 0 1	0 0 0 0 0 d d
	ADD.W @(d:16,ERs.L),@aa:16	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 0 0 1	0 0 0 0 0 a	

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
ADD	ADD.W @(d:16,ERs.L),@aa:32	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 0 0 1	0 0 0 0	a a
	ADD.W @(d:32,Rs.B),@ERd	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.B),@ERd+	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.B),@ERd-	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.B),@+ERd	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.B),@-ERd	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.B),@(d:2,ERd)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.B),@(d:16,ERd)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.B),@(d:32,ERd)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.B),@(d:16,Rd.B)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.B),@(d:16,Rd.W)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.B),@(d:16,ERd.L)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.B),@(d:32,Rd.B)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.B),@(d:32,Rd.W)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.B),@(d:32,ERd.L)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.B),@aa:16	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0 0 0 0	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.B),@aa:32	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1 0 0 0	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.W),@ERd	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.W),@ERd+	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.W),@ERd-	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.W),@+ERd	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.W),@-ERd	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.W),@(d:2,ERd)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.W),@(d:16,ERd)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.W),@(d:32,ERd)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.W),@(d:16,Rd.B)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.W),@(d:16,Rd.W)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.W),@(d:16,ERd.L)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.W),@(d:32,Rd.B)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.W),@(d:32,Rd.W)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.W),@(d:32,ERd.L)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1 rd	0 0 0 1	0 0 0 0	
	ADD.W @(d:32,Rs.W),@aa:16	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0 0 0 0	0 0 0 1	0 0 0 0	
ADD.W @(d:32,Rs.W),@aa:32	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1 0 0 0	0 0 0 1	0 0 0 0		
ADD.W @(d:32,ERs.L),@ERd	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0 rd	0 0 0 1	0 0 0 0		



Instruction	Mnemonic	Opcode					EA Extension	Opcode					EA Extension					
		15	8	7	0	15		8	7	0	15	8		7	0			
ADD	ADD.W @(d:32.ERs.L),@ERd+	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @(d:32.ERs.L),@ERd-	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @(d:32.ERs.L),@+ERd	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @(d:32.ERs.L),@-ERd	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @(d:32.ERs.L),@(d:2.ERd)	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @(d:32.ERs.L),@(d:16.ERd)	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d	
	ADD.W @(d:32.ERs.L),@(d:32.ERd)	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d d	
	ADD.W @(d:32.ERs.L),@(d:16.Rd.B)	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 0 0 1	0 0 0 0	d	
	ADD.W @(d:32.ERs.L),@(d:16.Rd.W)	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d	
	ADD.W @(d:32.ERs.L),@(d:16.ERd.L)	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d	
	ADD.W @(d:32.ERs.L),@(d:32.Rd.B)	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d d	
	ADD.W @(d:32.ERs.L),@(d:32.Rd.W)	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d d	
	ADD.W @(d:32.ERs.L),@(d:32.ERd.L)	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 0 0 1	0 0 0 0	d d	
	ADD.W @(d:32.ERs.L),@aa:16	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0 0 0 0	0 0 0 1	0 0 0 0	a		
	ADD.W @(d:32.ERs.L),@aa:32	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1 0 0 0	0 0 0 1	0 0 0 0	a a		
	ADD.W @aa:16,@ERd						6	B	0 0 0 1	0 1 0 1	a		0 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.W @aa:16,@ERd+						6	B	0 0 0 1	0 1 0 1	a		1 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.W @aa:16,@ERd-						6	B	0 0 0 1	0 1 0 1	a		1 0 1 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.W @aa:16,@+ERd						6	B	0 0 0 1	0 1 0 1	a		1 0 0 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.W @aa:16,@-ERd						6	B	0 0 0 1	0 1 0 1	a		1 0 1 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.W @aa:16,@(d:2.ERd)						6	B	0 0 0 1	0 1 0 1	a		0 0 d d	0	rd	0 0 0 1	0 0 0 0	
	ADD.W @aa:16,@(d:16.ERd)						6	B	0 0 0 1	0 1 0 1	a		1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.W @aa:16,@(d:32.ERd)						6	B	0 0 0 1	0 1 0 1	a		1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.W @aa:16,@(d:16.Rd.B)						6	B	0 0 0 1	0 1 0 1	a		1 1 0 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.W @aa:16,@(d:16.Rd.W)						6	B	0 0 0 1	0 1 0 1	a		1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.W @aa:16,@(d:16.ERd.L)						6	B	0 0 0 1	0 1 0 1	a		1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.W @aa:16,@(d:32.Rd.B)						6	B	0 0 0 1	0 1 0 1	a		1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.W @aa:16,@(d:32.Rd.W)						6	B	0 0 0 1	0 1 0 1	a		1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.W @aa:16,@(d:32.ERd.L)						6	B	0 0 0 1	0 1 0 1	a		1 1 1 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.W @aa:16,@aa:16						6	B	0 0 0 1	0 1 0 1	a		0 1 0 0	0 0 0 0	0 0 0 1	0 0 0 0	a	
ADD.W @aa:16,@aa:32						6	B	0 0 0 1	0 1 0 1	a		0 1 0 0	1 0 0 0	0 0 0 1	0 0 0 0	a a		
ADD.W @aa:32,@ERd						6	B	0 0 1 1	0 1 0 1	a a		0 0 0 0	0	rd	0 0 0 1	0 0 0 0		

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0		15	8	7	0			
ADD	ADD.W @aa:32,@ERd+	6	B	0 0 1 1	0 1 0 1	a a	1 0 0 0	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @aa:32,@ERd-	6	B	0 0 1 1	0 1 0 1	a a	1 0 1 0	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @aa:32,@+ERd	6	B	0 0 1 1	0 1 0 1	a a	1 0 0 1	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @aa:32,@-ERd	6	B	0 0 1 1	0 1 0 1	a a	1 0 1 1	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @aa:32,@(d:2,ERd)	6	B	0 0 1 1	0 1 0 1	a a	0 0 d d	0	rd	0 0 0 1	0 0 0 0		
	ADD.W @aa:32,@(d:16,ERd)	6	B	0 0 1 1	0 1 0 1	a a	1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d	
	ADD.W @aa:32,@(d:32,ERd)	6	B	0 0 1 1	0 1 0 1	a a	1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d d	
	ADD.W @aa:32,@(d:16,Rd,B)	6	B	0 0 1 1	0 1 0 1	a a	1 1 0 1	0	rd	0 0 0 1	0 0 0 0	d	
	ADD.W @aa:32,@(d:16,Rd,W)	6	B	0 0 1 1	0 1 0 1	a a	1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d	
	ADD.W @aa:32,@(d:16,ERd,L)	6	B	0 0 1 1	0 1 0 1	a a	1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d	
	ADD.W @aa:32,@(r:32,Rd,B)	6	B	0 0 1 1	0 1 0 1	a a	1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d d	
	ADD.W @aa:32,@(d:32,Rd,W)	6	B	0 0 1 1	0 1 0 1	a a	1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d d	
	ADD.W @aa:32,@(d:32,ERd,L)	6	B	0 0 1 1	0 1 0 1	a a	1 1 1 1	1	rd	0 0 0 1	0 0 0 0	d d	
	ADD.W @aa:32,@aa:16	6	B	0 0 1 1	0 1 0 1	a a	0 1 0 0	0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 0	a	
	ADD.W @aa:32,@aa:32	6	B	0 0 1 1	0 1 0 1	a a	0 1 0 0	1 0 0 0	0 0 0 0	0 0 0 1	0 0 0 0	a a	
	ADD.L #xx:3,ERd							0	A	1 x x x	1	rd	
	ADD.L #xx:16,ERd							7	A	0 0 0 1	1	rd	x
	ADD.L #xx:32,ERd							7	A	0 0 0 1	0	rd	x x
	ADD.L #xx:16,@ERd		0	1	0	1 1 1 0		0 0 0 0	0	rd	0 0 0 1	0 0 0 0	x
	ADD.L #xx:16,@ERd+		0	1	0	1 1 1 0		1 0 0 0	0	rd	0 0 0 1	0 0 0 0	x
ADD.L #xx:16,@ERd-		0	1	0	1 1 1 0		1 0 1 0	0	rd	0 0 0 1	0 0 0 0	x	
ADD.L #xx:16,@+ERd		0	1	0	1 1 1 0		1 0 0 1	0	rd	0 0 0 1	0 0 0 0	x	
ADD.L #xx:16,@-ERd		0	1	0	1 1 1 0		1 0 1 1	0	rd	0 0 0 1	0 0 0 0	x	
ADD.L #xx:16,@(d:2,ERd)		0	1	0	1 1 1 0		0 0 d d	0	rd	0 0 0 1	0 0 0 0	x	
ADD.L #xx:16,@(d:16,ERd)		0	1	0	1 1 1 0		1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d x	
ADD.L #xx:16,@(d:32,ERd)		0	1	0	1 1 1 0		1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d d x	
ADD.L #xx:16,@(d:16,Rd,B)		0	1	0	1 1 1 0		1 1 0 1	0	rd	0 0 0 1	0 0 0 0	d x	
ADD.L #xx:16,@(d:16,Rd,W)		0	1	0	1 1 1 0		1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d x	
ADD.L #xx:16,@(d:16,ERd,L)		0	1	0	1 1 1 0		1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d x	
ADD.L #xx:16,@(d:32,Rd,B)		0	1	0	1 1 1 0		1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d d x	
ADD.L #xx:16,@(d:32,Rd,W)		0	1	0	1 1 1 0		1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d d x	
ADD.L #xx:16,@(d:32,ERd,L)		0	1	0	1 1 1 0		1 1 1 1	1	rd	0 0 0 1	0 0 0 0	d d x	
ADD.L #xx:16,@aa:16		0	1	0	1 1 1 0		0 1 0 0	0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 0	a x	
ADD.L #xx:16,@aa:32		0	1	0	1 1 1 0		0 1 0 0	1 0 0 0	0 0 0 0	0 0 0 1	0 0 0 0	a a x	



Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension																					
		15	8	7	0		15	8	7	0																						
ADD	ADD.L #xx:32,@ERd				0	1		0	1	1	1	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0			x	x		
	ADD.L #xx:32,@ERd+				0	1		0	1	1	1	0	1	0	0	0	0	rd	0	0	0	1	1	0	0	0			x	x		
	ADD.L #xx:32,@ERd-				0	1		0	1	1	1	0	1	0	1	0	0	rd	0	0	0	1	1	0	0	0			x	x		
	ADD.L #xx:32,@+ERd				0	1		0	1	1	1	0	1	0	0	1	0	rd	0	0	0	1	1	0	0	0			x	x		
	ADD.L #xx:32,@-ERd				0	1		0	1	1	1	0	1	0	1	1	0	rd	0	0	0	1	1	0	0	0			x	x		
	ADD.L #xx:32,@(d:2,ERd)				0	1		0	1	1	1	0	0	0	d	d	0	rd	0	0	0	1	1	0	0	0			x	x		
	ADD.L #xx:32,@(d:16,ERd)				0	1		0	1	1	1	0	1	1	0	0	0	rd	0	0	0	1	1	0	0	0	d		x	x		
	ADD.L #xx:32,@(d:32,ERd)				0	1		0	1	1	1	0	1	1	0	0	1	rd	0	0	0	1	1	0	0	0	d	d	x	x		
	ADD.L #xx:32,@(d:16,Rd,B)				0	1		0	1	1	1	0	1	1	0	1	0	rd	0	0	0	1	1	0	0	0	d		x	x		
	ADD.L #xx:32,@(d:16,Rd,W)				0	1		0	1	1	1	0	1	1	1	0	0	rd	0	0	0	1	1	0	0	0	d		x	x		
	ADD.L #xx:32,@(d:16,ERd,L)				0	1		0	1	1	1	0	1	1	1	1	0	rd	0	0	0	1	1	0	0	0	d		x	x		
	ADD.L #xx:32,@(d:32,Rd,B)				0	1		0	1	1	1	0	1	1	0	1	1	rd	0	0	0	1	1	0	0	0	d	d	x	x		
	ADD.L #xx:32,@(d:32,Rd,W)				0	1		0	1	1	1	0	1	1	1	0	1	rd	0	0	0	1	1	0	0	0	d	d	x	x		
	ADD.L #xx:32,@(d:32,ERd,L)				0	1		0	1	1	1	0	1	1	1	1	1	rd	0	0	0	1	1	0	0	0	d	d	x	x		
	ADD.L #xx:32,@aa:16				0	1		0	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	a		x	x
	ADD.L #xx:32,@aa:32				0	1		0	1	1	1	0	0	1	0	0	1	0	0	0	0	0	0	1	1	0	0	0	a	a	x	x
	ADD.L ERs,ERd																0	A	1	rs	0	rd										
	ADD.L ERs,@ERd				0	1		0	1	0	0	1	0	0	0	0	0	rd	0	0	0	1	0	rs								
	ADD.L ERs,@ERd+				0	1		0	1	0	0	1	1	0	0	0	0	rd	0	0	0	1	0	rs								
	ADD.L ERs,@ERd-				0	1		0	1	0	0	1	1	0	1	0	0	rd	0	0	0	1	0	rs								
	ADD.L ERs,@+ERd				0	1		0	1	0	0	1	1	0	0	1	0	rd	0	0	0	1	0	rs								
	ADD.L ERs,@-ERd				0	1		0	1	0	0	1	1	0	1	1	0	rd	0	0	0	1	0	rs								
	ADD.L ERs,@(d:2,ERd)				0	1		0	1	0	0	1	0	0	d	d	0	rd	0	0	0	1	0	rs								
ADD.L ERs,@(d:16,ERd)				0	1		0	1	0	0	1	1	1	0	0	0	rd	0	0	0	1	0	rs	d								
ADD.L ERs,@(d:32,ERd)				0	1		0	1	0	0	1	1	1	0	0	1	rd	0	0	0	1	0	rs	d	d							
ADD.L ERs,@(d:16,Rd,B)				0	1		0	1	0	0	1	1	1	0	1	0	rd	0	0	0	1	0	rs	d								
ADD.L ERs,@(d:16,Rd,W)				0	1		0	1	0	0	1	1	1	1	0	0	rd	0	0	0	1	0	rs	d								
ADD.L ERs,@(d:16,ERd,L)				0	1		0	1	0	0	1	1	1	1	1	0	rd	0	0	0	1	0	rs	d								
ADD.L ERs,@(d:32,Rd,B)				0	1		0	1	0	0	1	1	1	0	1	1	rd	0	0	0	1	0	rs	d	d							
ADD.L ERs,@(d:32,Rd,W)				0	1		0	1	0	0	1	1	1	1	0	1	rd	0	0	0	1	0	rs	d	d							
ADD.L ERs,@(d:32,ERd,L)				0	1		0	1	0	0	1	1	1	1	1	1	rd	0	0	0	1	0	rs	d	d							
ADD.L ERs,@aa:16				0	1		0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	rs	a						
ADD.L ERs,@aa:32				0	1		0	1	0	0	1	0	1	0	0	1	0	0	0	0	0	0	1	0	rs	a	a					
ADD.L @ERs,ERd				0	1		0	1	0	1	0	0	0	0	0	0	rs	0	0	0	1	0	rd									

Instruction	Mnemonic	Opcode				EA Ex- tension	Opcode				EA Extension																					
		15	8	7	0		15	8	7	0																						
ADD	ADD.L @ERs+,ERd				0			1		0	1	0	1	0	1	0	0	0	0	rs	0	0	0	1	0	rd						
	ADD.L @ERs-,ERd				0			1		0	1	0	1	0	1	0	1	0	0	rs	0	0	0	1	0	rd						
	ADD.L @+ERs,ERd				0			1		0	1	0	1	0	1	0	0	1	0	rs	0	0	0	1	0	rd						
	ADD.L @-ERs,ERd				0			1		0	1	0	1	0	1	0	1	1	0	rs	0	0	0	1	0	rd						
	ADD.L @(d:2,ERs),ERd				0			1		0	1	0	1	0	0	0	d	d	0	rs	0	0	0	1	0	rd						
	ADD.L @(d:16,ERs),ERd				0			1		0	1	0	1	0	1	1	0	0	0	rs	0	0	0	1	0	rd	d					
	ADD.L @(d:32,ERs),ERd				0			1		0	1	0	1	0	1	1	0	0	1	rs	0	0	0	1	0	rd	d d					
	ADD.L @(d:16,Rs,B),ERd				0			1		0	1	0	1	0	1	1	0	1	0	rs	0	0	0	1	0	rd	d					
	ADD.L @(d:16,Rs,W),ERd				0			1		0	1	0	1	0	1	1	1	0	0	rs	0	0	0	1	0	rd	d					
	ADD.L @(d:16,ERs,L),ERd				0			1		0	1	0	1	0	1	1	1	1	0	rs	0	0	0	1	0	rd	d					
	ADD.L @(d:32,Rs,B),ERd				0			1		0	1	0	1	0	1	1	0	1	1	rs	0	0	0	1	0	rd	d d					
	ADD.L @(d:32,Rs,W),ERd				0			1		0	1	0	1	0	1	1	1	0	1	rs	0	0	0	1	0	rd	d d					
	ADD.L @(d:32,ERs,L),ERd				0			1		0	1	0	1	0	1	1	1	1	1	rs	0	0	0	1	0	rd	d d					
	ADD.L @aa:16,ERd				0			1		0	1	0	1	0	0	1	0	0	0	rs	0	0	0	1	0	rd	a					
	ADD.L @aa:32,ERd				0			1		0	1	0	1	0	0	1	0	0	1	rs	0	0	0	1	0	rd	a a					
	ADD.L @ERs,@ERd		0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	0	0	0	0	rd	0	0	0	1	0	0	0	0	0
	ADD.L @ERs,@ERd+		0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	0	0	0	rd	0	0	0	1	0	0	0	0	0
	ADD.L @ERs,@ERd-		0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	0	1	0	rd	0	0	0	1	0	0	0	0	0
	ADD.L @ERs,@+ERd		0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	0	0	1	rd	0	0	0	1	0	0	0	0	0
	ADD.L @ERs,@-ERd		0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	0	1	1	rd	0	0	0	1	0	0	0	0	0
	ADD.L @ERs,@(d:2,ERd)		0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	0	0	d	d	rd	0	0	0	1	0	0	0	0	0
	ADD.L @ERs,@(d:16,ERd)		0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	1	0	0	rd	0	0	0	1	0	0	0	0	0
	ADD.L @ERs,@(d:32,ERd)		0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	1	0	0	rd	0	0	0	1	0	0	0	0	0
	ADD.L @ERs,@(d:16,Rd,B)		0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	1	0	1	rd	0	0	0	1	0	0	0	0	0
	ADD.L @ERs,@(d:16,Rd,W)		0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	1	1	0	rd	0	0	0	1	0	0	0	0	0
	ADD.L @ERs,@(d:16,ERd,L)		0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	1	1	1	rd	0	0	0	1	0	0	0	0	0
	ADD.L @ERs,@(d:32,Rd,B)		0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	1	0	1	rd	0	0	0	1	0	0	0	0	0
	ADD.L @ERs,@(d:32,Rd,W)		0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	1	1	0	rd	0	0	0	1	0	0	0	0	0
	ADD.L @ERs,@(d:32,ERd,L)		0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	1	1	1	rd	0	0	0	1	0	0	0	0	0
	ADD.L @ERs,@aa:16		0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	0	1	0	0	rd	0	0	0	1	0	0	0	0	0
	ADD.L @ERs,@aa:32		0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	0	1	0	0	rd	0	0	0	1	0	0	0	0	0
	ADD.L @ERs+,@ERd		0	1	0	0	0	1	0	0	6	D	0	rs	1	1	0	0	0	0	0	0	rd	0	0	0	1	0	0	0	0	0
ADD.L @ERs+,@ERd+		0	1	0	0	0	1	0	0	6	D	0	rs	1	1	0	0	1	0	0	0	rd	0	0	0	1	0	0	0	0	0	
ADD.L @ERs+,@ERd-		0	1	0	0	0	1	0	0	6	D	0	rs	1	1	0	0	1	0	1	0	rd	0	0	0	1	0	0	0	0	0	



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
ADD	ADD.L @ERs+, @+ERd	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0	1	0 0 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @ERs+, @-ERd	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0	1	0 1 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @ERs+, @(d:2,ERd)	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0	0	0 d d	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @ERs+, @(d:16,ERd)	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0	1	1 0 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.L @ERs+, @(d:32,ERd)	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0	1	1 0 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.L @ERs+, @(d:16,Rd,B)	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0	1	1 0 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.L @ERs+, @(d:16,Rd,W)	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0	1	1 1 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.L @ERs+, @(d:16,ERd,L)	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0	1	1 1 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.L @ERs+, @(d:32,Rd,B)	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0	1	1 0 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.L @ERs+, @(d:32,Rd,W)	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0	1	1 1 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.L @ERs+, @(d:32,ERd,L)	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0	1	1 1 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.L @ERs+, @aa:16	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0	0	1 0 0	0	0 0 0 0	0 0 0 1	0 0 0 0	a
	ADD.L @ERs+, @aa:32	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0	0	1 0 0	1	0 0 0 0	0 0 0 1	0 0 0 0	a a
	ADD.L @ERs-, @ERd	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	0	0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @ERs-, @ERd+	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @ERs-, @ERd-	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	0 1 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @ERs-, @+ERd	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	0 0 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @ERs-, @-ERd	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	0 1 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @ERs-, @(d:2,ERd)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	0	0 d d	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @ERs-, @(d:16,ERd)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	1 0 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.L @ERs-, @(d:32,ERd)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	1 0 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.L @ERs-, @(d:16,Rd,B)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	1 0 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.L @ERs-, @(d:16,Rd,W)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	1 1 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.L @ERs-, @(d:16,ERd,L)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	1 1 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.L @ERs-, @(d:32,Rd,B)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	1 0 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.L @ERs-, @(d:32,Rd,W)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	1 1 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.L @ERs-, @(d:32,ERd,L)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	1 1 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.L @ERs-, @aa:16	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	0	1 0 0	0	0 0 0 0	0 0 0 1	0 0 0 0	a
	ADD.L @ERs-, @aa:32	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	0	1 0 0	1	0 0 0 0	0 0 0 1	0 0 0 0	a a
	ADD.L @+ERs, @ERd	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	0	0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @+ERs, @ERd+	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1	0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @+ERs, @ERd-	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1	0 1 0	0	rd	0 0 0 1	0 0 0 0	
ADD.L @+ERs, @+ERd	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1	0 0 1	0	rd	0 0 0 1	0 0 0 0		
ADD.L @+ERs, @-ERd	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1	0 1 1	0	rd	0 0 0 1	0 0 0 0		

Instruction	Mnemonic	Opcode						EA Ex- tension	Opcode						EA Extension																					
		15	8	7	0	15	8		7	0	15	8	7	0																						
ADD	ADD.L @+ERs, @(d:2,ERd)	0	1	0	0	1	0	1	0	0	D	0	rs	1	1	0	0	0	0	d	d	0	rd	0	0	0	1	0	0	0	0	0	0			
	ADD.L @+ERs, @(d:16,ERd)	0	1	0	0	1	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	0	0	0	rd	0	0	0	1	0	0	0	0	0	0	d	
	ADD.L @+ERs, @(d:32,ERd)	0	1	0	0	1	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	0	0	1	rd	0	0	0	1	0	0	0	0	0	0	d	d
	ADD.L @+ERs, @(d:16,Rd,B)	0	1	0	0	1	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	0	0	0	rd	0	0	0	1	0	0	0	0	0	0	d	
	ADD.L @+ERs, @(d:16,Rd,W)	0	1	0	0	1	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	1	0	0	rd	0	0	0	1	0	0	0	0	0	0	d	
	ADD.L @+ERs, @(d:16,ERd,L)	0	1	0	0	1	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	1	0	0	rd	0	0	0	1	0	0	0	0	0	0	d	
	ADD.L @+ERs, @(d:32,Rd,B)	0	1	0	0	1	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	0	1	rd	0	0	0	1	0	0	0	0	0	0	d	d	
	ADD.L @+ERs, @(d:32,Rd,W)	0	1	0	0	1	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	1	0	1	rd	0	0	0	1	0	0	0	0	0	0	d	d
	ADD.L @+ERs, @(d:32,ERd,L)	0	1	0	0	1	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	1	1	rd	0	0	0	1	0	0	0	0	0	0	d	d	
	ADD.L @+ERs, @aa:16	0	1	0	0	1	0	1	0	1	6	D	0	rs	1	1	0	0	0	1	0	0	0	rd	0	0	0	1	0	0	0	0	0	0	a	
	ADD.L @+ERs, @aa:32	0	1	0	0	1	0	1	0	1	6	D	0	rs	1	1	0	0	0	1	0	0	0	rd	0	0	0	1	0	0	0	0	0	0	a	a
	ADD.L @-ERs, @ERd	0	1	0	0	1	1	1	1	1	6	D	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	0	0	0	0	0	0		
	ADD.L @-ERs, @ERd+	0	1	0	0	1	1	1	1	1	6	D	0	rs	1	1	0	0	0	0	0	0	1	rd	0	0	0	1	0	0	0	0	0			
	ADD.L @-ERs, @ERd-	0	1	0	0	1	1	1	1	1	6	D	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	0	0	0	0	0			
	ADD.L @-ERs, @+ERd	0	1	0	0	1	1	1	1	1	6	D	0	rs	1	1	0	0	0	0	1	0	rd	0	0	0	1	0	0	0	0	0				
	ADD.L @-ERs, @-ERd	0	1	0	0	1	1	1	1	1	6	D	0	rs	1	1	0	0	0	0	1	1	rd	0	0	0	1	0	0	0	0	0				
	ADD.L @-ERs, @(d:2,ERd)	0	1	0	0	1	1	1	1	1	6	D	0	rs	1	1	0	0	0	0	1	0	rd	0	0	0	1	0	0	0	0	0				
	ADD.L @-ERs, @(d:16,ERd)	0	1	0	0	1	1	1	1	1	6	D	0	rs	1	1	0	0	0	0	1	1	rd	0	0	0	1	0	0	0	0	0	d			
	ADD.L @-ERs, @(d:32,ERd)	0	1	0	0	1	1	1	1	1	6	D	0	rs	1	1	0	0	0	0	1	1	rd	0	0	0	1	0	0	0	0	0	d	d		
	ADD.L @-ERs, @(d:16,Rd,B)	0	1	0	0	1	1	1	1	1	6	D	0	rs	1	1	0	0	0	0	1	1	rd	0	0	0	1	0	0	0	0	0	d			
	ADD.L @-ERs, @(d:16,Rd,W)	0	1	0	0	1	1	1	1	1	6	D	0	rs	1	1	0	0	0	0	1	1	rd	0	0	0	1	0	0	0	0	0	d			
	ADD.L @-ERs, @(d:16,ERd,L)	0	1	0	0	1	1	1	1	1	6	D	0	rs	1	1	0	0	0	0	1	1	rd	0	0	0	1	0	0	0	0	0	d			
	ADD.L @-ERs, @(d:32,Rd,B)	0	1	0	0	1	1	1	1	1	6	D	0	rs	1	1	0	0	0	0	1	1	rd	0	0	0	1	0	0	0	0	0	d	d		
	ADD.L @-ERs, @(d:32,Rd,W)	0	1	0	0	1	1	1	1	1	6	D	0	rs	1	1	0	0	0	0	1	1	rd	0	0	0	1	0	0	0	0	0	d	d		
	ADD.L @-ERs, @(d:32,ERd,L)	0	1	0	0	1	1	1	1	1	6	D	0	rs	1	1	0	0	0	0	1	1	rd	0	0	0	1	0	0	0	0	0	d	d		
	ADD.L @-ERs, @aa:16	0	1	0	0	1	1	1	1	1	6	D	0	rs	1	1	0	0	0	0	1	0	rd	0	0	0	1	0	0	0	0	0	a			
	ADD.L @-ERs, @aa:32	0	1	0	0	1	1	1	1	1	6	D	0	rs	1	1	0	0	0	0	1	0	rd	0	0	0	1	0	0	0	0	0	a	a		
	ADD.L @(d:2,ERs), @ERd	0	1	0	0	1	d	d	d	d	6	9	0	rs	1	1	0	0	0	0	0	0	rd	0	0	0	1	0	0	0	0	0				
	ADD.L @(d:2,ERs), @ERd+	0	1	0	0	1	d	d	d	d	6	9	0	rs	1	1	0	0	0	0	0	0	rd	0	0	0	1	0	0	0	0	0				
	ADD.L @(d:2,ERs), @ERd-	0	1	0	0	1	d	d	d	d	6	9	0	rs	1	1	0	0	0	0	0	0	rd	0	0	0	1	0	0	0	0	0				
	ADD.L @(d:2,ERs), @+ERd	0	1	0	0	1	d	d	d	d	6	9	0	rs	1	1	0	0	0	0	0	1	rd	0	0	0	1	0	0	0	0	0				
	ADD.L @(d:2,ERs), @-ERd	0	1	0	0	1	d	d	d	d	6	9	0	rs	1	1	0	0	0	0	0	1	rd	0	0	0	1	0	0	0	0	0				
ADD.L @(d:2,ERs), @(d:2,ERd)	0	1	0	0	1	d	d	d	d	6	9	0	rs	1	1	0	0	0	0	d	d	rd	0	0	0	1	0	0	0	0	0					
ADD.L @(d:2,ERs), @(d:16,ERd)	0	1	0	0	1	d	d	d	d	6	9	0	rs	1	1	0	0	0	0	d	d	rd	0	0	0	1	0	0	0	0	0	d				



Instruction	Mnemonic	Opcode						EA Ex- tension	Opcode						EA Extension																						
		15	8	7	0	15	8		7	0	15	8	7	0																							
ADD	ADD.L @(d:2,ERs),@(d:32,ERd)	0	1	0	0	1	d	d	6	9	0	rs	1	1	0	0		1	1	0	0	1	rd	0	0	0	1	0	0	0	0	0	d	d			
	ADD.L @(d:2,ERs),@(d:16,Rd,B)	0	1	0	0	0	1	d	d	6	9	0	rs	1	1	0	0		1	1	0	1	0	rd	0	0	0	1	0	0	0	0	0	d			
	ADD.L @(d:2,ERs),@(d:16,Rd,W)	0	1	0	0	0	1	d	d	6	9	0	rs	1	1	0	0		1	1	1	0	0	rd	0	0	0	1	0	0	0	0	0	d			
	ADD.L @(d:2,ERs),@(d:16,ERd,L)	0	1	0	0	0	1	d	d	6	9	0	rs	1	1	0	0		1	1	1	1	0	rd	0	0	0	1	0	0	0	0	0	d			
	ADD.L @(d:2,ERs),@(d:32,Rd,B)	0	1	0	0	0	1	d	d	6	9	0	rs	1	1	0	0		1	1	0	1	1	rd	0	0	0	1	0	0	0	0	0	d	d		
	ADD.L @(d:2,ERs),@(d:32,Rd,W)	0	1	0	0	0	1	d	d	6	9	0	rs	1	1	0	0		1	1	1	0	1	rd	0	0	0	1	0	0	0	0	0	d	d		
	ADD.L @(d:2,ERs),@(d:32,ERd,L)	0	1	0	0	0	1	d	d	6	9	0	rs	1	1	0	0		1	1	1	1	1	rd	0	0	0	1	0	0	0	0	0	d	d		
	ADD.L @(d:2,ERs),@aa:16	0	1	0	0	0	1	d	d	6	9	0	rs	1	1	0	0		0	1	0	0	0	rd	0	0	0	0	0	0	0	1	0	0	0	a	
	ADD.L @(d:2,ERs),@aa:32	0	1	0	0	0	1	d	d	6	9	0	rs	1	1	0	0		0	1	0	0	1	rd	0	0	0	0	0	0	0	1	0	0	0	a	a
	ADD.L @(d:16,ERs),@ERd	0	1	0	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	0	0	0	0	0	rd	0	0	0	1	0	0	0	0					
	ADD.L @(d:16,ERs),@ERd+	0	1	0	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	0	0	0	0	rd	0	0	0	1	0	0	0	0					
	ADD.L @(d:16,ERs),@ERd-	0	1	0	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	0	1	0	0	rd	0	0	0	1	0	0	0	0					
	ADD.L @(d:16,ERs),@+ERd	0	1	0	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	0	0	1	0	rd	0	0	0	1	0	0	0	0					
	ADD.L @(d:16,ERs),@-ERd	0	1	0	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	0	1	1	0	rd	0	0	0	1	0	0	0	0					
	ADD.L @(d:16,ERs),@(d:2,ERd)	0	1	0	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	0	0	d	d	0	rd	0	0	0	1	0	0	0	0					
	ADD.L @(d:16,ERs),@(d:16,ERd)	0	1	0	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	1	0	0	0	rd	0	0	0	1	0	0	0	0	d				
	ADD.L @(d:16,ERs),@(d:32,ERd)	0	1	0	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	1	0	0	1	rd	0	0	0	1	0	0	0	0	d	d			
	ADD.L @(d:16,ERs),@(d:16,Rd,B)	0	1	0	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	1	0	1	0	rd	0	0	0	1	0	0	0	0	d				
	ADD.L @(d:16,ERs),@(d:16,Rd,W)	0	1	0	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	1	1	0	0	rd	0	0	0	1	0	0	0	0	d				
	ADD.L @(d:16,ERs),@(d:16,ERd,L)	0	1	0	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	1	1	1	0	rd	0	0	0	1	0	0	0	0	d				
	ADD.L @(d:16,ERs),@(d:32,Rd,B)	0	1	0	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	1	0	1	1	rd	0	0	0	1	0	0	0	0	d	d			
	ADD.L @(d:16,ERs),@(d:32,Rd,W)	0	1	0	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	1	1	0	1	rd	0	0	0	1	0	0	0	0	d	d			
	ADD.L @(d:16,ERs),@(d:32,ERd,L)	0	1	0	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	1	1	1	1	rd	0	0	0	1	0	0	0	0	d	d			
	ADD.L @(d:16,ERs),@aa:16	0	1	0	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	0	1	0	0	0	rd	0	0	0	1	0	0	0	0	a				
	ADD.L @(d:16,ERs),@aa:32	0	1	0	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	0	1	0	0	1	rd	0	0	0	1	0	0	0	0	a	a			
	ADD.L @(d:32,ERs),@ERd	7	8	1	rs	0	1	0	0	6	B	0	0	1	0	1	1	1	0	0	d	d	0	0	0	0	0	0	rd	0	0	0	1	0	0	0	
	ADD.L @(d:32,ERs),@ERd+	7	8	1	rs	0	1	0	0	6	B	0	0	1	0	1	1	1	0	0	d	d	1	0	0	0	0	rd	0	0	0	1	0	0	0	0	
	ADD.L @(d:32,ERs),@ERd-	7	8	1	rs	0	1	0	0	6	B	0	0	1	0	1	1	1	0	0	d	d	1	0	1	0	0	rd	0	0	0	1	0	0	0	0	
ADD.L @(d:32,ERs),@+ERd	7	8	1	rs	0	1	0	0	6	B	0	0	1	0	1	1	1	0	0	d	d	1	0	0	1	0	rd	0	0	0	1	0	0	0	0		
ADD.L @(d:32,ERs),@-ERd	7	8	1	rs	0	1	0	0	6	B	0	0	1	0	1	1	1	0	0	d	d	1	0	1	1	0	rd	0	0	0	1	0	0	0	0		
ADD.L @(d:32,ERs),@(d:2,ERd)	7	8	1	rs	0	1	0	0	6	B	0	0	1	0	1	1	1	0	0	d	d	0	0	d	d	0	rd	0	0	0	1	0	0	0	0		
ADD.L @(d:32,ERs),@(d:16,ERd)	7	8	1	rs	0	1	0	0	6	B	0	0	1	0	1	1	1	0	0	d	d	1	1	0	0	0	rd	0	0	0	1	0	0	0	0	d	
ADD.L @(d:32,ERs),@(d:32,ERd)	7	8	1	rs	0	1	0	0	6	B	0	0	1	0	1	1	1	0	0	d	d	1	1	0	0	1	rd	0	0	0	1	0	0	0	0	d	d
ADD.L @(d:32,ERs),@(d:16,Rd,B)	7	8	1	rs	0	1	0	0	6	B	0	0	1	0	1	1	1	0	0	d	d	1	1	0	1	0	rd	0	0	0	1	0	0	0	0	d	

Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension				
		15	8	7	0	15	8		7	0	15	8	7	0					
ADD	ADD.L @(d:32,ERs),@(d:16,Rd,W)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d		
	ADD.L @(d:32,ERs),@(d:16,ERdL)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d		
	ADD.L @(d:32,ERs),@(d:32,Rd,B)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d d		
	ADD.L @(d:32,ERs),@(d:32,Rd,W)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d d		
	ADD.L @(d:32,ERs),@(d:32,ERdL)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 0 0 1	0 0 0 0	d d		
	ADD.L @(d:32,ERs),@aa:16	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 0 0 1	0 0 0 0	a		
	ADD.L @(d:32,ERs),@aa:32	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 0 0 1	0 0 0 0	a a		
	ADD.L @(d:16,Rs,B),@ERd	0	1			0 0 1 0 1	6	F	0	rs	1 1 0 0	d		0 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:16,Rs,B),@ERd+	0	1			0 0 1 0 1	6	F	0	rs	1 1 0 0	d		1 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:16,Rs,B),@ERd-	0	1			0 0 1 0 1	6	F	0	rs	1 1 0 0	d		1 0 1 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:16,Rs,B),@+ERd	0	1			0 0 1 0 1	6	F	0	rs	1 1 0 0	d		1 0 0 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:16,Rs,B),@-ERd	0	1			0 0 1 0 1	6	F	0	rs	1 1 0 0	d		1 0 1 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:16,Rs,B),@(d:2,ERd)	0	1			0 0 1 0 1	6	F	0	rs	1 1 0 0	d		0 0 d d	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:16,Rs,B),@(d:16,ERd)	0	1			0 0 1 0 1	6	F	0	rs	1 1 0 0	d		1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.L @(d:16,Rs,B),@(d:32,ERd)	0	1			0 0 1 0 1	6	F	0	rs	1 1 0 0	d		1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.L @(d:16,Rs,B),@(d:16,Rd,B)	0	1			0 0 1 0 1	6	F	0	rs	1 1 0 0	d		1 1 0 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.L @(d:16,Rs,B),@(d:16,Rd,W)	0	1			0 0 1 0 1	6	F	0	rs	1 1 0 0	d		1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.L @(d:16,Rs,B),@(d:16,ERdL)	0	1			0 0 1 0 1	6	F	0	rs	1 1 0 0	d		1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.L @(d:16,Rs,B),@(d:32,Rd,B)	0	1			0 0 1 0 1	6	F	0	rs	1 1 0 0	d		1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.L @(d:16,Rs,B),@(d:32,Rd,W)	0	1			0 0 1 0 1	6	F	0	rs	1 1 0 0	d		1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.L @(d:16,Rs,B),@(d:32,ERdL)	0	1			0 0 1 0 1	6	F	0	rs	1 1 0 0	d		1 1 1 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.L @(d:16,Rs,B),@aa:16	0	1			0 0 1 0 1	6	F	0	rs	1 1 0 0	d		0 1 0 0	0	0 0 0 0	0 0 0 1	0 0 0 0	a
	ADD.L @(d:16,Rs,B),@aa:32	0	1			0 0 1 0 1	6	F	0	rs	1 1 0 0	d		0 1 0 0	1	0 0 0 0	0 0 0 1	0 0 0 0	a a
	ADD.L @(d:16,Rs,W),@ERd	0	1			0 0 1 1 0	6	F	0	rs	1 1 0 0	d		0 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:16,Rs,W),@ERd+	0	1			0 0 1 1 0	6	F	0	rs	1 1 0 0	d		1 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:16,Rs,W),@ERd-	0	1			0 0 1 1 0	6	F	0	rs	1 1 0 0	d		1 0 1 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:16,Rs,W),@+ERd	0	1			0 0 1 1 0	6	F	0	rs	1 1 0 0	d		1 0 0 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:16,Rs,W),@-ERd	0	1			0 0 1 1 0	6	F	0	rs	1 1 0 0	d		1 0 1 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:16,Rs,W),@(d:2,ERd)	0	1			0 0 1 1 0	6	F	0	rs	1 1 0 0	d		0 0 d d	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:16,Rs,W),@(d:16,ERd)	0	1			0 0 1 1 0	6	F	0	rs	1 1 0 0	d		1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d
ADD.L @(d:16,Rs,W),@(d:32,ERd)	0	1			0 0 1 1 0	6	F	0	rs	1 1 0 0	d		1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d d	
ADD.L @(d:16,Rs,W),@(d:16,Rd,B)	0	1			0 0 1 1 0	6	F	0	rs	1 1 0 0	d		1 1 0 1	0	rd	0 0 0 1	0 0 0 0	d	
ADD.L @(d:16,Rs,W),@(d:16,Rd,W)	0	1			0 0 1 1 0	6	F	0	rs	1 1 0 0	d		1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d	
ADD.L @(d:16,Rs,W),@(d:16,ERdL)	0	1			0 0 1 1 0	6	F	0	rs	1 1 0 0	d		1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d	



Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
ADD	ADD.L @(d:16,Rs,W),@(d:32,Rd,B)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.L @(d:16,Rs,W),@(d:32,Rd,W)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.L @(d:16,Rs,W),@(d:32,ERd,L)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.L @(d:16,Rs,W),@aa:16	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	0	0 0 0 1	0 0 0 0	a	
	ADD.L @(d:16,Rs,W),@aa:32	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	1	0 0 0 0	0 0 0 1	0 0 0 0	a a
	ADD.L @(d:16,ERs,L),@ERd	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:16,ERs,L),@ERd+	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:16,ERs,L),@ERd-	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:16,ERs,L),@+ERd	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:16,ERs,L),@-ERd	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:16,ERs,L),@(d:2,ERd)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:16,ERs,L),@(d:16,ERd)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.L @(d:16,ERs,L),@(d:32,ERd)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.L @(d:16,ERs,L),@(d:16,Rd,B)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.L @(d:16,ERs,L),@(d:16,Rd,W)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.L @(d:16,ERs,L),@(d:16,ERd,L)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.L @(d:16,ERs,L),@(d:32,Rd,B)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.L @(d:16,ERs,L),@(d:32,Rd,W)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.L @(d:16,ERs,L),@(d:32,ERd,L)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.L @(d:16,ERs,L),@aa:16	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	0	0 0 0 0	0 0 0 1	0 0 0 0	a
	ADD.L @(d:16,ERs,L),@aa:32	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	1	0 0 0 0	0 0 0 1	0 0 0 0	a a
	ADD.L @(d:32,Rs,B),@ERd	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:32,Rs,B),@ERd+	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:32,Rs,B),@ERd-	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:32,Rs,B),@+ERd	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:32,Rs,B),@-ERd	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:32,Rs,B),@(d:2,ERd)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 0 0 1	0 0 0 0	
	ADD.L @(d:32,Rs,B),@(d:16,ERd)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.L @(d:32,Rs,B),@(d:32,ERd)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 0 0 1	0 0 0 0	d d
	ADD.L @(d:32,Rs,B),@(d:16,Rd,B)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 0 0 1	0 0 0 0	d
	ADD.L @(d:32,Rs,B),@(d:16,Rd,W)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 0 0 1	0 0 0 0	d
	ADD.L @(d:32,Rs,B),@(d:16,ERd,L)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 0 0 1	0 0 0 0	d
ADD.L @(d:32,Rs,B),@(d:32,Rd,B)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 0 0 1	0 0 0 0	d d	
ADD.L @(d:32,Rs,B),@(d:32,Rd,W)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 0 0 1	0 0 0 0	d d	

Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension		
		15	8	7	0	15	8		7	0	15	8	7	0			
ADD	ADD.L @(d:32.Rs.B),@(d:32,ERd.L)	7	8	1	rs	0 1 0 0 1	6	B	0 0 1 0 0	1 1 0 0 0	d d	1 1 1 1 1	1	rd	0 0 0 0 1	0 0 0 0 0	d d
	ADD.L @(d:32.Rs.B),@aa:16	7	8	1	rs	0 1 0 0 1	6	B	0 0 1 0 0	1 1 0 0 0	d d	0 1 0 0 0	0	0 0 0 0 0	0 0 0 0 1	0 0 0 0 0	a
	ADD.L @(d:32.Rs.B),@aa:32	7	8	1	rs	0 1 0 0 1	6	B	0 0 1 0 0	1 1 0 0 0	d d	0 1 0 0 0	1	0 0 0 0 0	0 0 0 0 1	0 0 0 0 0	a a
	ADD.L @(d:32.Rs.W),@ERd	7	8	1	rs	0 1 1 1 0	6	B	0 0 1 0 0	1 1 0 0 0	d d	0 0 0 0 0	0	rd	0 0 0 0 1	0 0 0 0 0	
	ADD.L @(d:32.Rs.W),@ERd+	7	8	1	rs	0 1 1 1 0	6	B	0 0 1 0 0	1 1 0 0 0	d d	1 0 0 0 0	0	rd	0 0 0 0 1	0 0 0 0 0	
	ADD.L @(d:32.Rs.W),@ERd-	7	8	1	rs	0 1 1 1 0	6	B	0 0 1 0 0	1 1 0 0 0	d d	1 0 1 0 0	0	rd	0 0 0 0 1	0 0 0 0 0	
	ADD.L @(d:32.Rs.W),@+ERd	7	8	1	rs	0 1 1 1 0	6	B	0 0 1 0 0	1 1 0 0 0	d d	1 0 0 0 1	0	rd	0 0 0 0 1	0 0 0 0 0	
	ADD.L @(d:32.Rs.W),@-ERd	7	8	1	rs	0 1 1 1 0	6	B	0 0 1 0 0	1 1 0 0 0	d d	1 0 1 1 0	0	rd	0 0 0 0 1	0 0 0 0 0	
	ADD.L @(d:32.Rs.W),@(d:2,ERd)	7	8	1	rs	0 1 1 1 0	6	B	0 0 1 0 0	1 1 0 0 0	d d	0 0 d d 0	0	rd	0 0 0 0 1	0 0 0 0 0	
	ADD.L @(d:32.Rs.W),@(d:16,ERd)	7	8	1	rs	0 1 1 1 0	6	B	0 0 1 0 0	1 1 0 0 0	d d	1 1 0 0 0	0	rd	0 0 0 0 1	0 0 0 0 0	d
	ADD.L @(d:32.Rs.W),@(d:32,ERd)	7	8	1	rs	0 1 1 1 0	6	B	0 0 1 0 0	1 1 0 0 0	d d	1 1 0 0 0	1	rd	0 0 0 0 1	0 0 0 0 0	d d
	ADD.L @(d:32.Rs.W),@(d:16,Rd.B)	7	8	1	rs	0 1 1 1 0	6	B	0 0 1 0 0	1 1 0 0 0	d d	1 1 0 1 0	0	rd	0 0 0 0 1	0 0 0 0 0	d
	ADD.L @(d:32.Rs.W),@(d:16,Rd.W)	7	8	1	rs	0 1 1 1 0	6	B	0 0 1 0 0	1 1 0 0 0	d d	1 1 1 1 0	0	rd	0 0 0 0 1	0 0 0 0 0	d
	ADD.L @(d:32.Rs.W),@(d:16,ERd.L)	7	8	1	rs	0 1 1 1 0	6	B	0 0 1 0 0	1 1 0 0 0	d d	1 1 1 1 1	0	rd	0 0 0 0 1	0 0 0 0 0	d
	ADD.L @(d:32.Rs.W),@(d:32,Rd.B)	7	8	1	rs	0 1 1 1 0	6	B	0 0 1 0 0	1 1 0 0 0	d d	1 1 0 1 1	1	rd	0 0 0 0 1	0 0 0 0 0	d d
	ADD.L @(d:32.Rs.W),@(d:32,Rd.W)	7	8	1	rs	0 1 1 1 0	6	B	0 0 1 0 0	1 1 0 0 0	d d	1 1 1 1 1	1	rd	0 0 0 0 1	0 0 0 0 0	d d
	ADD.L @(d:32.Rs.W),@(d:32,ERd.L)	7	8	1	rs	0 1 1 1 0	6	B	0 0 1 0 0	1 1 0 0 0	d d	1 1 1 1 1	1	rd	0 0 0 0 1	0 0 0 0 0	d d
	ADD.L @(d:32.Rs.W),@aa:16	7	8	1	rs	0 1 1 1 0	6	B	0 0 1 0 0	1 1 0 0 0	d d	0 1 0 0 0	0	0 0 0 0 0	0 0 0 0 1	0 0 0 0 0	a
	ADD.L @(d:32.Rs.W),@aa:32	7	8	1	rs	0 1 1 1 0	6	B	0 0 1 0 0	1 1 0 0 0	d d	0 1 0 0 0	1	0 0 0 0 0	0 0 0 0 1	0 0 0 0 0	a a
	ADD.L @(d:32,ERs.L),@ERd	7	8	1	rs	0 1 1 1 1	6	B	0 0 0 1 0	1 1 0 0 0	d d	0 0 0 0 0	0	rd	0 0 0 0 1	0 0 0 0 0	
	ADD.L @(d:32,ERs.L),@ERd+	7	8	1	rs	0 1 1 1 1	6	B	0 0 0 1 0	1 1 0 0 0	d d	1 0 0 0 0	0	rd	0 0 0 0 1	0 0 0 0 0	
	ADD.L @(d:32,ERs.L),@ERd-	7	8	1	rs	0 1 1 1 1	6	B	0 0 0 1 0	1 1 0 0 0	d d	1 0 1 0 0	0	rd	0 0 0 0 1	0 0 0 0 0	
	ADD.L @(d:32,ERs.L),@+ERd	7	8	1	rs	0 1 1 1 1	6	B	0 0 0 1 0	1 1 0 0 0	d d	1 0 0 0 1	0	rd	0 0 0 0 1	0 0 0 0 0	
	ADD.L @(d:32,ERs.L),@-ERd	7	8	1	rs	0 1 1 1 1	6	B	0 0 0 1 0	1 1 0 0 0	d d	1 0 1 1 0	0	rd	0 0 0 0 1	0 0 0 0 0	
	ADD.L @(d:32,ERs.L),@(d:2,ERd)	7	8	1	rs	0 1 1 1 1	6	B	0 0 0 1 0	1 1 0 0 0	d d	0 0 d d 0	0	rd	0 0 0 0 1	0 0 0 0 0	
	ADD.L @(d:32,ERs.L),@(d:16,ERd)	7	8	1	rs	0 1 1 1 1	6	B	0 0 0 1 0	1 1 0 0 0	d d	1 1 0 0 0	0	rd	0 0 0 0 1	0 0 0 0 0	d
	ADD.L @(d:32,ERs.L),@(d:32,ERd)	7	8	1	rs	0 1 1 1 1	6	B	0 0 0 1 0	1 1 0 0 0	d d	1 1 0 0 0	1	rd	0 0 0 0 1	0 0 0 0 0	d d
	ADD.L @(d:32,ERs.L),@(d:16,Rd.B)	7	8	1	rs	0 1 1 1 1	6	B	0 0 0 1 0	1 1 0 0 0	d d	1 1 0 1 0	0	rd	0 0 0 0 1	0 0 0 0 0	d
	ADD.L @(d:32,ERs.L),@(d:16,Rd.W)	7	8	1	rs	0 1 1 1 1	6	B	0 0 0 1 0	1 1 0 0 0	d d	1 1 1 1 0	0	rd	0 0 0 0 1	0 0 0 0 0	d
	ADD.L @(d:32,ERs.L),@(d:16,ERd.L)	7	8	1	rs	0 1 1 1 1	6	B	0 0 0 1 0	1 1 0 0 0	d d	1 1 1 1 1	0	rd	0 0 0 0 1	0 0 0 0 0	d
ADD.L @(d:32,ERs.L),@(d:32,Rd.B)	7	8	1	rs	0 1 1 1 1	6	B	0 0 0 1 0	1 1 0 0 0	d d	1 1 0 1 1	1	rd	0 0 0 0 1	0 0 0 0 0	d d	
ADD.L @(d:32,ERs.L),@(d:32,Rd.W)	7	8	1	rs	0 1 1 1 1	6	B	0 0 0 1 0	1 1 0 0 0	d d	1 1 1 1 0	1	rd	0 0 0 0 1	0 0 0 0 0	d d	
ADD.L @(d:32,ERs.L),@(d:32,ERd.L)	7	8	1	rs	0 1 1 1 1	6	B	0 0 0 1 0	1 1 0 0 0	d d	1 1 1 1 1	1	rd	0 0 0 0 1	0 0 0 0 0	d d	
ADD.L @(d:32,ERs.L),@aa:16	7	8	1	rs	0 1 1 1 1	6	B	0 0 0 1 0	1 1 0 0 0	d d	0 1 0 0 0	0	0 0 0 0 0	0 0 0 0 1	0 0 0 0 0	a	



Instruction	Mnemonic	Opcode					EA Extension	Opcode					EA Extension			
		15	8	7	0	15		8	7	0	15	8		7	0	
ADD	ADD.L @(d:32,ERsL),@aa:32	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1 0 0 0	0 0 0 1	0 0 0 0	a a
	ADD.L @aa:16,@ERd	0	1	0		0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 0 0 0	0 rd	0 0 0 1	0 0 0 0	
	ADD.L @aa:16,@ERd+	0	1	0		0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 0 0	0 rd	0 0 0 1	0 0 0 0	
	ADD.L @aa:16,@ERd-	0	1	0		0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 1 0	0 rd	0 0 0 1	0 0 0 0	
	ADD.L @aa:16,@+ERd	0	1	0		0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 0 1	0 rd	0 0 0 1	0 0 0 0	
	ADD.L @aa:16,@-ERd	0	1	0		0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 1 1	0 rd	0 0 0 1	0 0 0 0	
	ADD.L @aa:16,@(d:2,ERd)	0	1	0		0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 0 d d	0 rd	0 0 0 1	0 0 0 0	
	ADD.L @aa:16,@(d:16,ERd)	0	1	0		0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 0	0 rd	0 0 0 1	0 0 0 0	d
	ADD.L @aa:16,@(d:32,ERd)	0	1	0		0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 0	1 rd	0 0 0 1	0 0 0 0	d d
	ADD.L @aa:16,@(d:16,Rd,B)	0	1	0		0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 1	0 rd	0 0 0 1	0 0 0 0	d
	ADD.L @aa:16,@(d:16,Rd,W)	0	1	0		0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 0	0 rd	0 0 0 1	0 0 0 0	d
	ADD.L @aa:16,@(d:16,ERd,L)	0	1	0		0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 1	0 rd	0 0 0 1	0 0 0 0	d
	ADD.L @aa:16,@(d:32,Rd,B)	0	1	0		0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 1	1 rd	0 0 0 1	0 0 0 0	d d
	ADD.L @aa:16,@(d:32,Rd,W)	0	1	0		0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 0	1 rd	0 0 0 1	0 0 0 0	d d
	ADD.L @aa:16,@(d:32,ERd,L)	0	1	0		0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 1	1 rd	0 0 0 1	0 0 0 0	d d
	ADD.L @aa:16,@aa:16	0	1	0		0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 1 0 0	0 0 0 0	0 0 0 1	0 0 0 0	a
	ADD.L @aa:16,@aa:32	0	1	0		0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 1 0 0	1 0 0 0	0 0 0 1	0 0 0 0	a a
	ADD.L @aa:32,@ERd	0	1	0		0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 0 0 0	0 rd	0 0 0 1	0 0 0 0	
	ADD.L @aa:32,@ERd+	0	1	0		0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 0 0	0 rd	0 0 0 1	0 0 0 0	
	ADD.L @aa:32,@ERd-	0	1	0		0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 1 0	0 rd	0 0 0 1	0 0 0 0	
ADD.L @aa:32,@+ERd	0	1	0		0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 0 1	0 rd	0 0 0 1	0 0 0 0		
ADD.L @aa:32,@-ERd	0	1	0		0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 1 1	0 rd	0 0 0 1	0 0 0 0		
ADD.L @aa:32,@(d:2,ERd)	0	1	0		0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 0 d d	0 rd	0 0 0 1	0 0 0 0		
ADD.L @aa:32,@(d:16,ERd)	0	1	0		0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 0	0 rd	0 0 0 1	0 0 0 0	d	
ADD.L @aa:32,@(d:32,ERd)	0	1	0		0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 0	1 rd	0 0 0 1	0 0 0 0	d d	
ADD.L @aa:32,@(d:16,Rd,B)	0	1	0		0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 1	0 rd	0 0 0 1	0 0 0 0	d	
ADD.L @aa:32,@(d:16,Rd,W)	0	1	0		0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 0	0 rd	0 0 0 1	0 0 0 0	d	
ADD.L @aa:32,@(d:16,ERd,L)	0	1	0		0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 1	0 rd	0 0 0 1	0 0 0 0	d	
ADD.L @aa:32,@(d:32,Rd,B)	0	1	0		0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 1	1 rd	0 0 0 1	0 0 0 0	d d	
ADD.L @aa:32,@(d:32,Rd,W)	0	1	0		0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 0	1 rd	0 0 0 1	0 0 0 0	d d	
ADD.L @aa:32,@(d:32,ERd,L)	0	1	0		0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 1	1 rd	0 0 0 1	0 0 0 0	d d	
ADD.L @aa:32,@aa:16	0	1	0		0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 1 0 0	0 0 0 0	0 0 0 1	0 0 0 0	a	
ADD.L @aa:32,@aa:32	0	1	0		0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 1 0 0	1 0 0 0	0 0 0 1	0 0 0 0	a a	
ADDS	ADDS #1,ERd										0	B	0 0 0 0	0 rd		

Instruction	Mnemonic	Opcode			Opcode			EA Ex- tension	Opcode			EA Extension															
		15	8	7	0	15	8		7	0	15		8	7	0												
ADDS	ADDS #2,ERd								0	B	1	0	0	0	0	rd											
	ADDS #4,ERd								0	B	1	0	0	1	0	rd											
ADDX	ADDX.B #xx:8,Rd								9	rd	x	x	x	x	x	x	x										
	ADDX.B #xx:8,@ERd					7	D	0	rd	0	0	0	0	0	x	x	x	x	x	x	x						
	ADDX.B #xx:8,@ERd-	0	1	7	0	1	1	0	6	C	0	rd	1	0	0	0	0	x	x	x	x	x	x				
	ADDX.B Rs,Rd								0	E	rs	rd															
	ADDX.B Rs,@ERd					7	D	0	rd	0	0	0	0	0	0	E	rs	0	0	0	0	0					
	ADDX.B Rs,@ERd-	0	1	7	0	1	1	0	6	C	0	rd	1	0	0	0	0	E	rs	0	0	0	0	0			
	ADDX.B @ERs,Rd					7	C	0	rs	0	0	0	0	0	0	0	E	0	0	0	0	rd					
	ADDX.B @ERs-,Rd	0	1	7	0	1	1	0	6	C	0	rs	0	0	0	0	0	E	0	0	0	0	rd				
	ADDX.B @ERs,@ERd	0	1	7	0	1	0	0	6	8	0	rs	1	1	0	1	0	0	0	0	0	0	0	0			
	ADDX.B @ERs-,@ERd-	0	1	7	0	1	1	0	6	C	0	rs	1	1	0	1	1	0	1	0	0	0	0	0			
	ADDX.W #xx:16,Rd					0	1	5	0	0	0	1	7	9	0	0	0	1	rd	x							
	ADDX.W #xx:16,@ERd					7	D	1	rd	0	0	0	1	7	9	0	0	0	1	0	0	0	0	x			
	ADDX.W #xx:16,@ERd-	0	1	5	0	1	1	0	6	D	0	rd	1	0	0	1	7	9	0	0	0	1	0	0	0	x	
	ADDX.W Rs,Rd					0	1	5	0	0	0	1	0	9	rs	rd											
	ADDX.W Rs,@ERd					7	D	1	rd	0	0	0	1	0	9	rs	0	0	0	0	0						
	ADDX.W Rs,@ERd-	0	1	5	0	1	1	0	6	D	0	rd	1	0	0	1	0	9	rs	0	0	0	0				
	ADDX.W @ERs,Rd					7	C	1	rs	0	0	0	1	0	9	0	0	0	0	rd							
	ADDX.W @ERs-,Rd	0	1	5	0	1	1	0	6	D	0	rs	0	0	0	1	0	9	0	0	0	0	rd				
ADDX.W @ERs,@ERd	0	1	5	0	1	0	0	6	9	0	rs	1	1	0	1	0	0	0	0	0	0	0	0				
ADDX.W @ERs-,@ERd-	0	1	5	0	1	1	0	6	D	0	rs	1	1	0	1	1	0	1	0	0	0	0	0	0			
ADDX.L #xx:32,ERd					0	1	0	0	0	0	1	7	A	0	0	0	1	0	rd	x	x						
ADDX.L #xx:32,@ERd	0	1	0	0	1	0	0	6	9	0	rd	1	0	0	1	7	A	0	0	0	1	0	0	0	0	x	x
ADDX.L #xx:32,@ERd-	0	1	0	0	1	1	0	6	D	0	rd	1	0	0	1	7	A	0	0	0	1	0	0	0	0	x	x
ADDX.L ERs,ERd					0	1	0	0	0	0	1	0	A	rs	rd												
ADDX.L ERs,@ERd	0	1	0	0	1	0	0	6	9	0	rd	1	0	0	1	0	A	1	rs	0	0	0	0	0			
ADDX.L ERs,@ERd-	0	1	0	0	1	1	0	6	D	0	rd	1	0	0	1	0	A	1	rs	0	0	0	0	0			
ADDX.L @ERs,ERd	0	1	0	0	1	0	0	6	9	0	rs	0	0	0	1	0	A	1	0	0	0	0	0	rd			
ADDX.L @ERs-,ERd	0	1	0	0	1	1	0	6	D	0	rs	0	0	0	1	0	A	1	0	0	0	0	0	rd			
ADDX.L @ERs,@ERd	0	1	0	0	1	0	0	6	9	0	rs	1	1	0	1	0	0	0	0	0	1	0	0	0	0		
ADDX.L @ERs-,@ERd-	0	1	0	0	1	1	0	6	D	0	rs	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	
AND	AND.B #xx:8,Rd								E	rd	x	x	x	x	x	x	x										
	AND.B #xx:8,@ERd					7	D	0	rd	0	0	0	0	E	0	0	0	0	x	x	x	x	x	x	x		



Instruction	Mnemonic	Opcode			Opcode			EA Ex- tension	Opcode			EA Extension					
		15	8	7	0	15	8		7	0	15		8	7	0		
AND	AND.B #xx:8,@ERd+	0	1	7	0 1 0 0	6	C	0	rd	1 0 0 0		E	0 0 0 0	x x x x	x x x x		
	AND.B #xx:8,@ERd-	0	1	7	0 1 1 0	6	C	0	rd	1 0 0 0		E	0 0 0 0	x x x x	x x x x		
	AND.B #xx:8,@+ERd	0	1	7	0 1 0 1	6	C	0	rd	1 0 0 0		E	0 0 0 0	x x x x	x x x x		
	AND.B #xx:8,@-ERd	0	1	7	0 1 1 1	6	C	0	rd	1 0 0 0		E	0 0 0 0	x x x x	x x x x		
	AND.B #xx:8,(d:2,ERd)	0	1	7	0 1 d d	6	8	0	rd	1 0 0 0		E	0 0 0 0	x x x x	x x x x		
	AND.B #xx:8,(d:16,ERd)	0	1	7	0 1 0 0	6	E	0	rd	1 0 0 0	d	E	0 0 0 0	x x x x	x x x x		
	AND.B #xx:8,(d:32,ERd)	7	8	0	rd	0 1 0 0	6	A	0 0 1 0	1 0 0 0	d d	E	0 0 0 0	x x x x	x x x x		
	AND.B #xx:8,(d:16,Rd.B)	0	1	7	0 1 0 1	6	E	0	rd	1 0 0 0	d	E	0 0 0 0	x x x x	x x x x		
	AND.B #xx:8,(d:16,Rd.W)	0	1	7	0 1 1 0	6	E	0	rd	1 0 0 0	d	E	0 0 0 0	x x x x	x x x x		
	AND.B #xx:8,(d:16,ERd.L)	0	1	7	0 1 1 1	6	E	0	rd	1 0 0 0	d	E	0 0 0 0	x x x x	x x x x		
	AND.B #xx:8,(d:32,Rd.B)	7	8	0	rd	0 1 0 1	6	A	0 0 1 0	1 0 0 0	d d	E	0 0 0 0	x x x x	x x x x		
	AND.B #xx:8,(d:32,Rd.W)	7	8	0	rd	0 1 1 0	6	A	0 0 1 0	1 0 0 0	d d	E	0 0 0 0	x x x x	x x x x		
	AND.B #xx:8,(d:32,ERd.L)	7	8	0	rd	0 1 1 1	6	A	0 0 1 0	1 0 0 0	d d	E	0 0 0 0	x x x x	x x x x		
	AND.B #xx:8,@aa:8					7	F		a a a a a a a a			E	0 0 0 0	x x x x	x x x x		
	AND.B #xx:8,@aa:16					6	A		0 0 0 1 1 0 0 0	a		E	0 0 0 0	x x x x	x x x x		
	AND.B #xx:8,@aa:32					6	A		0 0 1 1 1 0 0 0	a a		E	0 0 0 0	x x x x	x x x x		
	AND.B Rs,Rd												1	6	rs	rd	
	AND.B Rs,@ERd					7	D	0	rd	0 0 0 0			1	6	rs	0 0 0 0	
	AND.B Rs,@ERd+					0	1	7	1 0 0 1		1 0 0 0	0	rd	0 1 1 0	rs		
	AND.B Rs,@ERd-					0	1	7	1 0 0 1		1 0 1 0	0	rd	0 1 1 0	rs		
	AND.B Rs,@+ERd					0	1	7	1 0 0 1		1 0 0 1	0	rd	0 1 1 0	rs		
	AND.B Rs,@-ERd					0	1	7	1 0 0 1		1 0 1 1	0	rd	0 1 1 0	rs		
	AND.B Rs,(d:2,ERd)					0	1	7	1 0 0 1		0 0 d d	0	rd	0 1 1 0	rs		
	AND.B Rs,(d:16,ERd)					0	1	7	1 0 0 1		1 1 0 0	0	rd	0 1 1 0	rs	d	
	AND.B Rs,(d:32,ERd)					0	1	7	1 0 0 1		1 1 0 0	1	rd	0 1 1 0	rs	d d	
	AND.B Rs,(d:16,Rd.B)					0	1	7	1 0 0 1		1 1 0 1	0	rd	0 1 1 0	rs	d	
	AND.B Rs,(d:16,Rd.W)					0	1	7	1 0 0 1		1 1 1 0	0	rd	0 1 1 0	rs	d	
	AND.B Rs,(d:16,ERd.L)					0	1	7	1 0 0 1		1 1 1 1	0	rd	0 1 1 0	rs	d	
	AND.B Rs,(d:32,Rd.B)					0	1	7	1 0 0 1		1 1 0 1	1	rd	0 1 1 0	rs	d d	
	AND.B Rs,(d:32,Rd.W)					0	1	7	1 0 0 1		1 1 1 0	1	rd	0 1 1 0	rs	d d	
	AND.B Rs,(d:32,ERd.L)					0	1	7	1 0 0 1		1 1 1 1	1	rd	0 1 1 0	rs	d d	
	AND.B Rs,@aa:8					7	F		a a a a a a a a			1	6	rs	0 0 0 0		
AND.B Rs,@aa:16					6	A		0 0 0 1 1 0 0 0	a		1	6	rs	0 0 0 0			
AND.B Rs,@aa:32					6	A		0 0 1 1 1 0 0 0	a a		1	6	rs	0 0 0 0			

Instruction	Mnemonic	Opcode			Opcode			EA Ex- tension	Opcode			EA Extension															
		15	8	7	0	15	8		7	0	15		8	7	0												
AND	AND.B @ERs,Rd				7	C	0	rs	0	0	0	0	0	1	6	0	0	0	0	rd							
	AND.B @ERs+,Rd				0	1		7	1	0	1	0		1	0	0	0	0	rs	0	1	1	0	rd			
	AND.B @ERs-,Rd				0	1		7	1	0	1	0		1	0	1	0	0	rs	0	1	1	0	rd			
	AND.B @+ERs,Rd				0	1		7	1	0	1	0		1	0	0	1	0	rs	0	1	1	0	rd			
	AND.B @-ERs,Rd				0	1		7	1	0	1	0		1	0	1	1	0	rs	0	1	1	0	rd			
	AND.B @(d:2,ERs),Rd				0	1		7	1	0	1	0		0	0	d	d	0	rs	0	1	1	0	rd			
	AND.B @(d:16,ERs),Rd				0	1		7	1	0	1	0		1	1	0	0	0	rs	0	1	1	0	rd			
	AND.B @(d:32,ERs),Rd				0	1		7	1	0	1	0		1	1	0	0	1	rs	0	1	1	0	rd			
	AND.B @(d:16,Rs.B),Rd				0	1		7	1	0	1	0		1	1	0	1	0	rs	0	1	1	0	rd			
	AND.B @(d:16,Rs.W),Rd				0	1		7	1	0	1	0		1	1	1	0	0	rs	0	1	1	0	rd			
	AND.B @(d:16,ERs.L),Rd				0	1		7	1	0	1	0		1	1	1	1	0	rs	0	1	1	0	rd			
	AND.B @(d:32,Rs.B),Rd				0	1		7	1	0	1	0		1	1	0	1	1	rs	0	1	1	0	rd			
	AND.B @(d:32,Rs.W),Rd				0	1		7	1	0	1	0		1	1	1	0	1	rs	0	1	1	0	rd			
	AND.B @(d:32,ERs.L),Rd				0	1		7	1	0	1	0		1	1	1	1	1	rs	0	1	1	0	rd			
	AND.B @aa:8,Rd				7	E	a	a	a	a	a	a	a	a	1	6	0	0	0	0	rd						
	AND.B @aa:16,Rd				6	A	0	0	0	1	0	0	0	0	a	1	6	0	0	0	0	rd					
	AND.B @aa:32,Rd				6	A	0	0	1	1	0	0	0	0	a	a	1	6	0	0	0	0	rd				
	AND.B @ERs,@ERd				7	C	0	rs	0	1	0	1		0	0	0	0	0	rd	0	1	1	0	0	0	0	
	AND.B @ERs,@ERd+				7	C	0	rs	0	1	0	1		1	0	0	0	0	rd	0	1	1	0	0	0	0	
	AND.B @ERs,@ERd-				7	C	0	rs	0	1	0	1		1	0	1	0	0	rd	0	1	1	0	0	0	0	
	AND.B @ERs,@+ERd				7	C	0	rs	0	1	0	1		1	0	0	1	0	rd	0	1	1	0	0	0	0	
	AND.B @ERs,@-ERd				7	C	0	rs	0	1	0	1		1	0	1	1	0	rd	0	1	1	0	0	0	0	
	AND.B @ERs,@(d:2,ERd)				7	C	0	rs	0	1	0	1		0	0	d	d	0	rd	0	1	1	0	0	0	0	
	AND.B @ERs,@(d:16,ERd)				7	C	0	rs	0	1	0	1		1	1	0	0	0	rd	0	1	1	0	0	0	0	
	AND.B @ERs,@(d:32,ERd)				7	C	0	rs	0	1	0	1		1	1	0	0	1	rd	0	1	1	0	0	0	0	
	AND.B @ERs,@(d:16,Rd.B)				7	C	0	rs	0	1	0	1		1	1	0	1	0	rd	0	1	1	0	0	0	0	
	AND.B @ERs,@(d:16,Rd.W)				7	C	0	rs	0	1	0	1		1	1	1	0	0	rd	0	1	1	0	0	0	0	
	AND.B @ERs,@(d:16,ERd.L)				7	C	0	rs	0	1	0	1		1	1	1	1	0	rd	0	1	1	0	0	0	0	
	AND.B @ERs,@(d:32,Rd.B)				7	C	0	rs	0	1	0	1		1	1	0	1	1	rd	0	1	1	0	0	0	0	
	AND.B @ERs,@(d:32,Rd.W)				7	C	0	rs	0	1	0	1		1	1	1	0	1	rd	0	1	1	0	0	0	0	
	AND.B @ERs,@(d:32,ERd.L)				7	C	0	rs	0	1	0	1		1	1	1	1	1	rd	0	1	1	0	0	0	0	
	AND.B @ERs,@aa:16				7	C	0	rs	0	1	0	1		0	1	0	0	0	0	rd	0	1	1	0	0	0	0
	AND.B @ERs,@aa:32				7	C	0	rs	0	1	0	1		0	1	0	0	1	0	rd	0	1	1	0	0	0	0
	AND.B @ERs+,@ERd				0	1		7	0	1	0	0		6	C	0	rs	1	1	0	0						



Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension																		
		15	8	7	0	15	8		7	0	15	8	7	0																			
AND	AND.B @ERs+,@ERd+	0	1	7	0	1	0	0	6	C	0	rs	1	1	0	0	1	0	0	0	0	rd	0	1	1	0	0	0	0	0			
	AND.B @ERs+,@ERd-	0	1	7	0	1	0	0	6	C	0	rs	1	1	0	0	1	0	1	0	0	rd	0	1	1	0	0	0	0	0			
	AND.B @ERs+,@+ERd	0	1	7	0	1	0	0	6	C	0	rs	1	1	0	0	1	0	0	1	0	rd	0	1	1	0	0	0	0	0			
	AND.B @ERs+,@-ERd	0	1	7	0	1	0	0	6	C	0	rs	1	1	0	0	1	0	1	1	0	rd	0	1	1	0	0	0	0	0			
	AND.B @ERs+,@(d:2,ERd)	0	1	7	0	1	0	0	6	C	0	rs	1	1	0	0	0	0	d	d	0	rd	0	1	1	0	0	0	0	0			
	AND.B @ERs+,@(d:16,ERd)	0	1	7	0	1	0	0	6	C	0	rs	1	1	0	0	1	1	0	0	0	rd	0	1	1	0	0	0	0	0	d		
	AND.B @ERs+,@(d:32,ERd)	0	1	7	0	1	0	0	6	C	0	rs	1	1	0	0	1	1	0	0	1	rd	0	1	1	0	0	0	0	0	d	d	
	AND.B @ERs+,@(d:16,Rd,B)	0	1	7	0	1	0	0	6	C	0	rs	1	1	0	0	1	1	0	1	0	rd	0	1	1	0	0	0	0	0	d		
	AND.B @ERs+,@(d:16,Rd,W)	0	1	7	0	1	0	0	6	C	0	rs	1	1	0	0	1	1	1	0	0	rd	0	1	1	0	0	0	0	0	d		
	AND.B @ERs+,@(d:16,ERd,L)	0	1	7	0	1	0	0	6	C	0	rs	1	1	0	0	1	1	1	1	0	rd	0	1	1	0	0	0	0	0	d		
	AND.B @ERs+,@(d:32,Rd,B)	0	1	7	0	1	0	0	6	C	0	rs	1	1	0	0	1	1	0	1	1	rd	0	1	1	0	0	0	0	0	d	d	
	AND.B @ERs+,@(d:32,Rd,W)	0	1	7	0	1	0	0	6	C	0	rs	1	1	0	0	1	1	1	0	1	rd	0	1	1	0	0	0	0	0	d	d	
	AND.B @ERs+,@(d:32,ERd,L)	0	1	7	0	1	0	0	6	C	0	rs	1	1	0	0	1	1	1	1	1	rd	0	1	1	0	0	0	0	0	d	d	
	AND.B @ERs+,@aa:16	0	1	7	0	1	0	0	6	C	0	rs	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	a	
	AND.B @ERs+,@aa:32	0	1	7	0	1	0	0	6	C	0	rs	1	1	0	0	0	1	0	0	1	0	0	0	0	0	1	1	0	0	a	a	
	AND.B @ERs,@ERd	0	1	7	0	1	1	0	6	C	0	rs	1	1	0	0	0	0	0	0	0	rd	0	1	1	0	0	0	0	0			
	AND.B @ERs,@ERd+	0	1	7	0	1	1	0	6	C	0	rs	1	1	0	0	1	0	0	0	0	rd	0	1	1	0	0	0	0	0			
	AND.B @ERs,@ERd-	0	1	7	0	1	1	0	6	C	0	rs	1	1	0	0	1	0	1	0	0	rd	0	1	1	0	0	0	0	0			
	AND.B @ERs,@+ERd	0	1	7	0	1	1	0	6	C	0	rs	1	1	0	0	1	0	0	1	0	rd	0	1	1	0	0	0	0	0			
	AND.B @ERs,@-ERd	0	1	7	0	1	1	0	6	C	0	rs	1	1	0	0	1	0	1	1	0	rd	0	1	1	0	0	0	0	0			
	AND.B @ERs,@(d:2,ERd)	0	1	7	0	1	1	0	6	C	0	rs	1	1	0	0	0	0	d	d	0	rd	0	1	1	0	0	0	0	0			
	AND.B @ERs,@(d:16,ERd)	0	1	7	0	1	1	0	6	C	0	rs	1	1	0	0	1	1	0	0	0	rd	0	1	1	0	0	0	0	0	d		
	AND.B @ERs,@(d:32,ERd)	0	1	7	0	1	1	0	6	C	0	rs	1	1	0	0	1	1	0	0	1	rd	0	1	1	0	0	0	0	0	d	d	
	AND.B @ERs,@(d:16,Rd,B)	0	1	7	0	1	1	0	6	C	0	rs	1	1	0	0	1	1	0	1	0	rd	0	1	1	0	0	0	0	0	d		
	AND.B @ERs,@(d:16,Rd,W)	0	1	7	0	1	1	0	6	C	0	rs	1	1	0	0	1	1	1	0	0	rd	0	1	1	0	0	0	0	0	d		
	AND.B @ERs,@(d:16,ERd,L)	0	1	7	0	1	1	0	6	C	0	rs	1	1	0	0	1	1	1	1	0	rd	0	1	1	0	0	0	0	0	d		
	AND.B @ERs,@(d:32,Rd,B)	0	1	7	0	1	1	0	6	C	0	rs	1	1	0	0	1	1	0	1	1	rd	0	1	1	0	0	0	0	0	d	d	
	AND.B @ERs,@(d:32,Rd,W)	0	1	7	0	1	1	0	6	C	0	rs	1	1	0	0	1	1	1	0	1	rd	0	1	1	0	0	0	0	0	d	d	
	AND.B @ERs,@(d:32,ERd,L)	0	1	7	0	1	1	0	6	C	0	rs	1	1	0	0	1	1	1	1	1	rd	0	1	1	0	0	0	0	0	d	d	
AND.B @ERs,@aa:16	0	1	7	0	1	1	0	6	C	0	rs	1	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	a			
AND.B @ERs,@aa:32	0	1	7	0	1	1	0	6	C	0	rs	1	1	0	0	0	1	0	0	1	0	0	0	0	0	1	1	0	0	a	a		
AND.B @+ERs,@ERd	0	1	7	0	1	0	1	6	C	0	rs	1	1	0	0	0	0	0	0	0	rd	0	1	1	0	0	0	0	0				
AND.B @+ERs,@ERd+	0	1	7	0	1	0	1	6	C	0	rs	1	1	0	0	1	0	0	0	0	rd	0	1	1	0	0	0	0	0				
AND.B @+ERs,@ERd-	0	1	7	0	1	0	1	6	C	0	rs	1	1	0	0	1	0	0	0	0	rd	0	1	1	0	0	0	0	0				

Instruction	Mnemonic	Opcode						EA Ex- tension	Opcode						EA Extension																			
		15	8	7	0	15	8		7	0	15	8	7	0																				
AND	AND.B @+ERs,@+ERd	0	1	7	0	1	0	1	6	C	0	rs	1	1	0	0	1	0	0	1	0	rd	0	0	1	1	0	0	0	0	0	0	0	
	AND.B @+ERs,@-ERd	0	1	7	0	1	0	1	6	C	0	rs	1	1	0	0	1	0	1	1	0	rd	0	0	1	1	0	0	0	0	0	0	0	
	AND.B @+ERs,@(d:2,ERd)	0	1	7	0	1	0	1	6	C	0	rs	1	1	0	0	0	0	d	d	0	rd	0	0	1	1	0	0	0	0	0	0	0	
	AND.B @+ERs,@(d:16,ERd)	0	1	7	0	1	0	1	6	C	0	rs	1	1	0	0	1	1	0	0	0	rd	0	0	1	1	0	0	0	0	0	0	0	0
	AND.B @+ERs,@(d:32,ERd)	0	1	7	0	1	0	1	6	C	0	rs	1	1	0	0	1	1	0	0	1	rd	0	0	1	1	0	0	0	0	0	0	0	0
	AND.B @+ERs,@(d:16,Rd,B)	0	1	7	0	1	0	1	6	C	0	rs	1	1	0	0	1	1	0	1	0	rd	0	0	1	1	0	0	0	0	0	0	0	d
	AND.B @+ERs,@(d:16,Rd,W)	0	1	7	0	1	0	1	6	C	0	rs	1	1	0	0	1	1	1	0	0	rd	0	0	1	1	0	0	0	0	0	0	0	d
	AND.B @+ERs,@(d:16,ERd,L)	0	1	7	0	1	0	1	6	C	0	rs	1	1	0	0	1	1	1	1	0	rd	0	0	1	1	0	0	0	0	0	0	0	d
	AND.B @+ERs,@(d:32,Rd,B)	0	1	7	0	1	0	1	6	C	0	rs	1	1	0	0	1	1	0	1	1	rd	0	0	1	1	0	0	0	0	0	0	d	d
	AND.B @+ERs,@(d:32,Rd,W)	0	1	7	0	1	0	1	6	C	0	rs	1	1	0	0	1	1	1	0	1	rd	0	0	1	1	0	0	0	0	0	0	d	d
	AND.B @+ERs,@(d:32,ERd,L)	0	1	7	0	1	0	1	6	C	0	rs	1	1	0	0	1	1	1	1	1	rd	0	0	1	1	0	0	0	0	0	0	d	d
	AND.B @+ERs,@aa:16	0	1	7	0	1	0	1	6	C	0	rs	1	1	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	a
	AND.B @+ERs,@aa:32	0	1	7	0	1	0	1	6	C	0	rs	1	1	0	0	0	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	a	a
	AND.B @-ERs,@ERd	0	1	7	0	1	1	1	6	C	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	1	1	0	0	0	0	0	0	0	0
	AND.B @-ERs,@ERd+	0	1	7	0	1	1	1	6	C	0	rs	1	1	0	0	1	0	0	0	0	rd	0	0	1	1	0	0	0	0	0	0	0	0
	AND.B @-ERs,@ERd-	0	1	7	0	1	1	1	6	C	0	rs	1	1	0	0	1	0	1	0	0	rd	0	0	1	1	0	0	0	0	0	0	0	0
	AND.B @-ERs,@+ERd	0	1	7	0	1	1	1	6	C	0	rs	1	1	0	0	1	0	0	1	0	rd	0	0	1	1	0	0	0	0	0	0	0	0
	AND.B @-ERs,@-ERd	0	1	7	0	1	1	1	6	C	0	rs	1	1	0	0	1	0	1	1	0	rd	0	0	1	1	0	0	0	0	0	0	0	0
	AND.B @-ERs,@(d:2,ERd)	0	1	7	0	1	1	1	6	C	0	rs	1	1	0	0	0	0	d	d	0	rd	0	0	1	1	0	0	0	0	0	0	0	0
	AND.B @-ERs,@(d:16,ERd)	0	1	7	0	1	1	1	6	C	0	rs	1	1	0	0	1	1	0	0	0	rd	0	0	1	1	0	0	0	0	0	0	d	
	AND.B @-ERs,@(d:32,ERd)	0	1	7	0	1	1	1	6	C	0	rs	1	1	0	0	1	1	0	0	1	rd	0	0	1	1	0	0	0	0	0	d	d	
	AND.B @-ERs,@(d:16,Rd,B)	0	1	7	0	1	1	1	6	C	0	rs	1	1	0	0	1	1	0	1	0	rd	0	0	1	1	0	0	0	0	0	d		
	AND.B @-ERs,@(d:16,Rd,W)	0	1	7	0	1	1	1	6	C	0	rs	1	1	0	0	1	1	1	0	0	rd	0	0	1	1	0	0	0	0	0	d		
	AND.B @-ERs,@(d:16,ERd,L)	0	1	7	0	1	1	1	6	C	0	rs	1	1	0	0	1	1	1	1	0	rd	0	0	1	1	0	0	0	0	0	d		
	AND.B @-ERs,@(d:32,Rd,B)	0	1	7	0	1	1	1	6	C	0	rs	1	1	0	0	1	1	0	1	1	rd	0	0	1	1	0	0	0	0	d	d		
	AND.B @-ERs,@(d:32,Rd,W)	0	1	7	0	1	1	1	6	C	0	rs	1	1	0	0	1	1	1	0	1	rd	0	0	1	1	0	0	0	0	d	d		
	AND.B @-ERs,@(d:32,ERd,L)	0	1	7	0	1	1	1	6	C	0	rs	1	1	0	0	1	1	1	1	1	rd	0	0	1	1	0	0	0	0	d	d		
	AND.B @-ERs,@aa:16	0	1	7	0	1	1	1	6	C	0	rs	1	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	a	
	AND.B @-ERs,@aa:32	0	1	7	0	1	1	1	6	C	0	rs	1	1	0	0	0	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	a	a
	AND.B @(d:2,ERs),@ERd	0	1	7	0	1	d	d	6	8	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	1	1	0	0	0	0	0	0	0	0
	AND.B @(d:2,ERs),@ERd+	0	1	7	0	1	d	d	6	8	0	rs	1	1	0	0	1	0	0	0	0	rd	0	0	1	1	0	0	0	0	0	0	0	0
	AND.B @(d:2,ERs),@ERd-	0	1	7	0	1	d	d	6	8	0	rs	1	1	0	0	1	0	1	0	0	rd	0	0	1	1	0	0	0	0	0	0	0	0
AND.B @(d:2,ERs),@+ERd	0	1	7	0	1	d	d	6	8	0	rs	1	1	0	0	1	0	0	1	0	rd	0	0	1	1	0	0	0	0	0	0	0	0	
AND.B @(d:2,ERs),@-ERd	0	1	7	0	1	d	d	6	8	0	rs	1	1	0	0	1	0	1	1	0	rd	0	0	1	1	0	0	0	0	0	0	0	0	



Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension																			
		15	8	7	0	15	8		7	0	15	8	7	0																				
AND	AND.B @(d2,ERs),@(d2,ERd)	0	1	7	0	1	d	d	6	8	0	rs	1	1	0	0	0	0	d	0	rd	0	1	1	0	0	0	0	0	0	0	d		
	AND.B @(d2,ERs),@(d:16,ERd)	0	1	7	0	1	d	d	6	8	0	rs	1	1	0	0	1	1	0	0	0	rd	0	1	1	0	0	0	0	0	0	0	d	
	AND.B @(d2,ERs),@(d:32,ERd)	0	1	7	0	1	d	d	6	8	0	rs	1	1	0	0	1	1	0	0	1	rd	0	1	1	0	0	0	0	0	0	0	d	
	AND.B @(d2,ERs),@(d:16,Rd.B)	0	1	7	0	1	d	d	6	8	0	rs	1	1	0	0	1	1	0	1	0	rd	0	1	1	0	0	0	0	0	0	0	d	
	AND.B @(d2,ERs),@(d:16,Rd.W)	0	1	7	0	1	d	d	6	8	0	rs	1	1	0	0	1	1	1	0	0	rd	0	1	1	0	0	0	0	0	0	0	0	d
	AND.B @(d2,ERs),@(d:16,ERd.L)	0	1	7	0	1	d	d	6	8	0	rs	1	1	0	0	1	1	1	0	0	rd	0	1	1	0	0	0	0	0	0	0	0	d
	AND.B @(d2,ERs),@(d:32,Rd.B)	0	1	7	0	1	d	d	6	8	0	rs	1	1	0	0	1	1	0	1	1	rd	0	1	1	0	0	0	0	0	0	0	d	
	AND.B @(d2,ERs),@(d:32,Rd.W)	0	1	7	0	1	d	d	6	8	0	rs	1	1	0	0	1	1	1	0	1	rd	0	1	1	0	0	0	0	0	0	0	d	
	AND.B @(d2,ERs),@(d:32,ERd.L)	0	1	7	0	1	d	d	6	8	0	rs	1	1	0	0	1	1	1	1	0	rd	0	1	1	0	0	0	0	0	0	0	d	
	AND.B @(d2,ERs),@aa:16	0	1	7	0	1	d	d	6	8	0	rs	1	1	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	a	
	AND.B @(d2,ERs),@aa:32	0	1	7	0	1	d	d	6	8	0	rs	1	1	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	a	
	AND.B @(d:16,ERs),@ERd	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	0	0	0	0	rd	0	1	1	0	0	0	0	0	0	0		
	AND.B @(d:16,ERs),@ERd+	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	0	0	0	rd	0	1	1	0	0	0	0	0	0	0		
	AND.B @(d:16,ERs),@ERd-	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	0	1	0	rd	0	1	1	0	0	0	0	0	0	0		
	AND.B @(d:16,ERs),@+ERd	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	0	0	1	0	rd	0	1	1	0	0	0	0	0	0		
	AND.B @(d:16,ERs),@-ERd	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	0	1	1	0	rd	0	1	1	0	0	0	0	0	0		
	AND.B @(d:16,ERs),@(d2,ERd)	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	0	0	d	0	rd	0	1	1	0	0	0	0	0	0	0		
	AND.B @(d:16,ERs),@(d:16,ERd)	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	1	0	0	rd	0	1	1	0	0	0	0	0	0	0	d	
	AND.B @(d:16,ERs),@(d:32,ERd)	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	1	0	0	1	rd	0	1	1	0	0	0	0	0	0	d	
	AND.B @(d:16,ERs),@(d:16,Rd.B)	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	1	0	1	0	rd	0	1	1	0	0	0	0	0	0	d	
	AND.B @(d:16,ERs),@(d:16,Rd.W)	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	1	1	0	0	rd	0	1	1	0	0	0	0	0	0	d	
	AND.B @(d:16,ERs),@(d:16,ERd.L)	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	1	1	1	0	rd	0	1	1	0	0	0	0	0	0	d	
	AND.B @(d:16,ERs),@(d:32,Rd.B)	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	1	0	1	1	rd	0	1	1	0	0	0	0	0	0	d	
	AND.B @(d:16,ERs),@(d:32,Rd.W)	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	1	1	0	1	rd	0	1	1	0	0	0	0	0	0	d	
	AND.B @(d:16,ERs),@(d:32,ERd.L)	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	1	1	1	1	rd	0	1	1	0	0	0	0	0	0	d	
	AND.B @(d:16,ERs),@aa:16	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	a	
	AND.B @(d:16,ERs),@aa:32	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	a	
	AND.B @(d:32,ERs),@ERd	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	1	1	1	0	0	0	0	rd	0	1	1	0	0	0	0	0	0		
	AND.B @(d:32,ERs),@ERd+	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	1	1	1	0	0	0	0	rd	0	1	1	0	0	0	0	0	0		
	AND.B @(d:32,ERs),@ERd-	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	1	1	1	0	0	0	0	rd	0	1	1	0	0	0	0	0	0		
AND.B @(d:32,ERs),@+ERd	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	1	1	1	0	0	0	1	rd	0	1	1	0	0	0	0	0	0			
AND.B @(d:32,ERs),@-ERd	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	1	1	1	0	0	0	1	rd	0	1	1	0	0	0	0	0	0			
AND.B @(d:32,ERs),@(d2,ERd)	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	1	1	1	0	0	0	d	0	rd	0	1	1	0	0	0	0	0			
AND.B @(d:32,ERs),@(d:16,ERd)	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	1	1	1	0	0	0	d	0	rd	0	1	1	0	0	0	0	0	d		

Instruction	Mnemonic	Opcode						Opcode						EA Extension																					
		15	8	7	0	15	8	7	0	EA Extension	15	8	7		0	EA Extension																			
AND	AND.B @(d:32,ERs),@(d:32,ERd)	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	d	d	1	1	0	0	1	rd	0	0	1	1	0	0	0	0	0	d	d	
	AND.B @(d:32,ERs),@(d:16,Rd,B)	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	d	d	1	1	0	1	0	rd	0	0	1	1	0	0	0	0	0	d		
	AND.B @(d:32,ERs),@(d:16,Rd,W)	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	d	d	1	1	1	0	0	rd	0	0	1	1	0	0	0	0	0	d		
	AND.B @(d:32,ERs),@(d:16,ERd,L)	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	d	d	1	1	1	1	0	rd	0	0	1	1	0	0	0	0	0	d		
	AND.B @(d:32,ERs),@(d:32,Rd,B)	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	d	d	1	1	0	1	1	rd	0	0	1	1	0	0	0	0	0	d	d	
	AND.B @(d:32,ERs),@(d:32,Rd,W)	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	d	d	1	1	1	0	1	rd	0	0	1	1	0	0	0	0	0	d	d	
	AND.B @(d:32,ERs),@(d:32,ERd,L)	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	d	d	1	1	1	1	1	rd	0	0	1	1	0	0	0	0	0	d	d	
	AND.B @(d:32,ERs),@aa:16	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	d	d	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	a		
	AND.B @(d:32,ERs),@aa:32	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	d	d	0	1	0	0	0	1	0	0	0	0	1	1	0	0	0	0	a	a
	AND.B @(d:16,Rs,B),@ERd	0	1	7	0	1	0	1	6	E	0	rs	1	1	0	d		0	0	0	0	0	rd	0	0	1	1	0	0	0	0	0			
	AND.B @(d:16,Rs,B),@ERd+	0	1	7	0	1	0	1	6	E	0	rs	1	1	0	d		1	0	0	0	0	rd	0	0	1	1	0	0	0	0				
	AND.B @(d:16,Rs,B),@ERd-	0	1	7	0	1	0	1	6	E	0	rs	1	1	0	d		1	0	1	0	0	rd	0	0	1	1	0	0	0	0				
	AND.B @(d:16,Rs,B),@+ERd	0	1	7	0	1	0	1	6	E	0	rs	1	1	0	d		1	0	0	1	0	rd	0	0	1	1	0	0	0	0				
	AND.B @(d:16,Rs,B),@-ERd	0	1	7	0	1	0	1	6	E	0	rs	1	1	0	d		1	0	1	1	0	rd	0	0	1	1	0	0	0	0				
	AND.B @(d:16,Rs,B),@(d:2,ERd)	0	1	7	0	1	0	1	6	E	0	rs	1	1	0	d		0	0	d	d	0	rd	0	0	1	1	0	0	0	0				
	AND.B @(d:16,Rs,B),@(d:16,ERd)	0	1	7	0	1	0	1	6	E	0	rs	1	1	0	d		1	1	0	0	0	rd	0	0	1	1	0	0	0	0	d			
	AND.B @(d:16,Rs,B),@(d:32,ERd)	0	1	7	0	1	0	1	6	E	0	rs	1	1	0	d		1	1	0	0	1	rd	0	0	1	1	0	0	0	0	d	d		
	AND.B @(d:16,Rs,B),@(d:16,Rd,B)	0	1	7	0	1	0	1	6	E	0	rs	1	1	0	d		1	1	0	1	0	rd	0	0	1	1	0	0	0	0	d			
	AND.B @(d:16,Rs,B),@(d:16,Rd,W)	0	1	7	0	1	0	1	6	E	0	rs	1	1	0	d		1	1	1	0	0	rd	0	0	1	1	0	0	0	0	d			
	AND.B @(d:16,Rs,B),@(d:16,ERd,L)	0	1	7	0	1	0	1	6	E	0	rs	1	1	0	d		1	1	1	1	0	rd	0	0	1	1	0	0	0	0	d			
	AND.B @(d:16,Rs,B),@(d:32,Rd,B)	0	1	7	0	1	0	1	6	E	0	rs	1	1	0	d		1	1	0	1	1	rd	0	0	1	1	0	0	0	0	d	d		
	AND.B @(d:16,Rs,B),@(d:32,Rd,W)	0	1	7	0	1	0	1	6	E	0	rs	1	1	0	d		1	1	1	0	1	rd	0	0	1	1	0	0	0	0	d	d		
	AND.B @(d:16,Rs,B),@(d:32,ERd,L)	0	1	7	0	1	0	1	6	E	0	rs	1	1	0	d		1	1	1	1	1	rd	0	0	1	1	0	0	0	0	d	d		
	AND.B @(d:16,Rs,B),@aa:16	0	1	7	0	1	0	1	6	E	0	rs	1	1	0	d		0	1	0	0	0	0	0	0	0	1	1	0	0	0	a			
	AND.B @(d:16,Rs,B),@aa:32	0	1	7	0	1	0	1	6	E	0	rs	1	1	0	d		0	1	0	0	1	0	0	0	0	1	1	0	0	0	a	a		
	AND.B @(d:16,Rs,W),@ERd	0	1	7	0	1	1	0	6	E	0	rs	1	1	0	d		0	0	0	0	0	rd	0	0	1	1	0	0	0	0				
	AND.B @(d:16,Rs,W),@ERd+	0	1	7	0	1	1	0	6	E	0	rs	1	1	0	d		1	0	0	0	0	rd	0	0	1	1	0	0	0	0				
	AND.B @(d:16,Rs,W),@ERd-	0	1	7	0	1	1	0	6	E	0	rs	1	1	0	d		1	0	1	0	0	rd	0	0	1	1	0	0	0	0				
	AND.B @(d:16,Rs,W),@+ERd	0	1	7	0	1	1	0	6	E	0	rs	1	1	0	d		1	0	0	1	0	rd	0	0	1	1	0	0	0	0				
	AND.B @(d:16,Rs,W),@-ERd	0	1	7	0	1	1	0	6	E	0	rs	1	1	0	d		1	0	1	1	0	rd	0	0	1	1	0	0	0	0				
	AND.B @(d:16,Rs,W),@(d:2,ERd)	0	1	7	0	1	1	0	6	E	0	rs	1	1	0	d		0	0	d	d	0	rd	0	0	1	1	0	0	0	0				
	AND.B @(d:16,Rs,W),@(d:16,ERd)	0	1	7	0	1	1	0	6	E	0	rs	1	1	0	d		1	1	0	0	0	rd	0	0	1	1	0	0	0	0	d			
	AND.B @(d:16,Rs,W),@(d:32,ERd)	0	1	7	0	1	1	0	6	E	0	rs	1	1	0	d		1	1	0	0	1	rd	0	0	1	1	0	0	0	0	d	d		
	AND.B @(d:16,Rs,W),@(d:16,Rd,B)	0	1	7	0	1	1	0	6	E	0	rs	1	1	0	d		1	1	0	1	0	rd	0	0	1	1	0	0	0	0	d			



Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension		
		15	8	7	0	15	8		7	0	15	8	7	0		EA Extension	
AND	AND.B @(d:16,Rs,W),@(d:16,Rd,W)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.B @(d:16,Rs,W),@(d:16,ERdL)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.B @(d:16,Rs,W),@(d:32,Rd,B)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.B @(d:16,Rs,W),@(d:32,Rd,W)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.B @(d:16,Rs,W),@(d:32,ERdL)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.B @(d:16,Rs,W),@aa:16	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 1 1 0	0 0 0 0	a	
	AND.B @(d:16,Rs,W),@aa:32	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 1 0	0 0 0 0	a a	
	AND.B @(d:16,ERs,L),@ERd	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.B @(d:16,ERs,L),@ERd+	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.B @(d:16,ERs,L),@ERd-	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 1 0	0 0 0 0	
	AND.B @(d:16,ERs,L),@+ERd	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 1 0	0 0 0 0	
	AND.B @(d:16,ERs,L),@-ERd	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 1 1 0	0 0 0 0	
	AND.B @(d:16,ERs,L),@(d:2,ERd)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 1 1 0	0 0 0 0	
	AND.B @(d:16,ERs,L),@(d:16,ERd)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.B @(d:16,ERs,L),@(d:32,ERd)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.B @(d:16,ERs,L),@(d:16,Rd,B)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.B @(d:16,ERs,L),@(d:16,Rd,W)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.B @(d:16,ERs,L),@(d:16,ERdL)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.B @(d:16,ERs,L),@(d:32,Rd,B)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.B @(d:16,ERs,L),@(d:32,Rd,W)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.B @(d:16,ERs,L),@(d:32,ERdL)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.B @(d:16,ERs,L),@aa:16	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 1 1 0	0 0 0 0	a	
	AND.B @(d:16,ERs,L),@aa:32	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 1 0	0 0 0 0	a a	
	AND.B @(d:32,Rs,B),@ERd	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.B @(d:32,Rs,B),@ERd+	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.B @(d:32,Rs,B),@ERd-	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 1 1 0	0 0 0 0	
	AND.B @(d:32,Rs,B),@+ERd	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 1 1 0	0 0 0 0	
	AND.B @(d:32,Rs,B),@-ERd	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 1 1 0	0 0 0 0	
	AND.B @(d:32,Rs,B),@(d:2,ERd)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 1 1 0	0 0 0 0	
	AND.B @(d:32,Rs,B),@(d:16,ERd)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.B @(d:32,Rs,B),@(d:32,ERd)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.B @(d:32,Rs,B),@(d:16,Rd,B)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 1 1 0	0 0 0 0	d
AND.B @(d:32,Rs,B),@(d:16,Rd,W)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 1 1 0	0 0 0 0	d	
AND.B @(d:32,Rs,B),@(d:16,ERdL)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 1 1 0	0 0 0 0	d	

Instruction	Mnemonic	Opcode						EA Ex- tension	Opcode						EA Extension																
		15	8	7	0				15	8	7	0																			
AND	AND.B @(d:32.Rs.B),@(d:32.Rd.B)	7	8	0	rs	0	1	0	1	6	A	0	0	1	0	1	1	1	0	0	d	d									
	AND.B @(d:32.Rs.B),@(d:32.Rd.W)	7	8	0	rs	0	1	0	1	6	A	0	0	1	0	1	1	1	0	1	rd	0	0	0	0	0	d	d			
	AND.B @(d:32.Rs.B),@(d:32.ERd.L)	7	8	0	rs	0	1	0	1	6	A	0	0	1	0	1	1	1	1	1	rd	0	0	1	1	0	0	d	d		
	AND.B @(d:32.Rs.B),@aa:16	7	8	0	rs	0	1	0	1	6	A	0	0	1	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	a
	AND.B @(d:32.Rs.B),@aa:32	7	8	0	rs	0	1	0	1	6	A	0	0	1	0	1	1	0	0	0	0	1	0	0	0	0	1	1	0	0	a
	AND.B @(d:32.Rs.W),@ERd	7	8	0	rs	0	1	1	0	6	A	0	0	1	0	1	1	0	0	0	0	rd	0	0	1	1	0	0	0	0	
	AND.B @(d:32.Rs.W),@ERd+	7	8	0	rs	0	1	1	0	6	A	0	0	1	0	1	1	0	0	0	0	rd	0	0	1	1	0	0	0	0	
	AND.B @(d:32.Rs.W),@ERd-	7	8	0	rs	0	1	1	0	6	A	0	0	1	0	1	1	0	0	0	0	rd	0	0	1	1	0	0	0	0	
	AND.B @(d:32.Rs.W),@+ERd	7	8	0	rs	0	1	1	0	6	A	0	0	1	0	1	1	0	0	0	0	rd	0	0	1	1	0	0	0	0	
	AND.B @(d:32.Rs.W),@-ERd	7	8	0	rs	0	1	1	0	6	A	0	0	1	0	1	1	0	0	0	0	rd	0	0	1	1	0	0	0	0	
	AND.B @(d:32.Rs.W),@(d:2.ERd)	7	8	0	rs	0	1	1	0	6	A	0	0	1	0	1	1	0	0	0	d	0	rd	0	1	1	0	0	0	0	
	AND.B @(d:32.Rs.W),@(d:16.ERd)	7	8	0	rs	0	1	1	0	6	A	0	0	1	0	1	1	0	0	0	rd	0	0	1	1	0	0	0	0	d	
	AND.B @(d:32.Rs.W),@(d:32.ERd)	7	8	0	rs	0	1	1	0	6	A	0	0	1	0	1	1	0	0	0	1	rd	0	0	1	1	0	0	0	d	d
	AND.B @(d:32.Rs.W),@(d:16.Rd.B)	7	8	0	rs	0	1	1	0	6	A	0	0	1	0	1	1	0	0	0	rd	0	0	1	1	0	0	0	0	d	
	AND.B @(d:32.Rs.W),@(d:16.Rd.W)	7	8	0	rs	0	1	1	0	6	A	0	0	1	0	1	1	0	0	0	rd	0	0	1	1	0	0	0	0	d	
	AND.B @(d:32.Rs.W),@(d:16.ERd.L)	7	8	0	rs	0	1	1	0	6	A	0	0	1	0	1	1	0	0	0	rd	0	0	1	1	0	0	0	0	d	
	AND.B @(d:32.Rs.W),@(d:32.Rd.B)	7	8	0	rs	0	1	1	0	6	A	0	0	1	0	1	1	0	0	0	rd	0	0	1	1	0	0	0	0	d	d
	AND.B @(d:32.Rs.W),@(d:32.Rd.W)	7	8	0	rs	0	1	1	0	6	A	0	0	1	0	1	1	0	0	0	rd	0	0	1	1	0	0	0	0	d	d
	AND.B @(d:32.Rs.W),@(d:32.ERd.L)	7	8	0	rs	0	1	1	0	6	A	0	0	1	0	1	1	0	0	0	rd	0	0	1	1	1	1	0	0	d	d
	AND.B @(d:32.Rs.W),@aa:16	7	8	0	rs	0	1	1	0	6	A	0	0	1	0	1	1	0	0	0	0	rd	0	0	1	1	0	0	0	a	
	AND.B @(d:32.Rs.W),@aa:32	7	8	0	rs	0	1	1	0	6	A	0	0	1	0	1	1	0	0	0	1	rd	0	0	1	1	0	0	0	a	a
	AND.B @(d:32.ERs.L),@ERd	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	0	0	rd	0	0	1	1	0	0	0	0	
	AND.B @(d:32.ERs.L),@ERd+	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	0	0	rd	0	0	1	1	0	0	0	0	
	AND.B @(d:32.ERs.L),@ERd-	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	0	0	rd	0	0	1	1	0	0	0	0	
	AND.B @(d:32.ERs.L),@+ERd	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	0	1	rd	0	0	1	1	0	0	0	0	
	AND.B @(d:32.ERs.L),@-ERd	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	0	0	rd	0	0	1	1	0	0	0	0	
	AND.B @(d:32.ERs.L),@(d:2.ERd)	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	0	d	0	rd	0	0	1	1	0	0	0	
	AND.B @(d:32.ERs.L),@(d:16.ERd)	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	0	rd	0	0	1	1	0	0	0	0	d	
	AND.B @(d:32.ERs.L),@(d:32.ERd)	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	0	1	rd	0	0	1	1	0	0	0	d	d
	AND.B @(d:32.ERs.L),@(d:16.Rd.B)	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	0	rd	0	0	1	1	0	0	0	0	d	
	AND.B @(d:32.ERs.L),@(d:16.Rd.W)	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	0	rd	0	0	1	1	0	0	0	0	d	
	AND.B @(d:32.ERs.L),@(d:16.ERd.L)	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	0	rd	0	0	1	1	0	0	0	0	d	
AND.B @(d:32.ERs.L),@(d:32.Rd.B)	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	0	1	rd	0	0	1	1	0	0	0	d	d	
AND.B @(d:32.ERs.L),@(d:32.Rd.W)	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	0	1	rd	0	0	1	1	0	0	0	d	d	



Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension			
		15	8	7	0	15	8		7	0	15	8	7	0				
AND	AND.B	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 1 1 0	0 0 0 0	d d	
	AND.B @(d:32,ERs.L),@aa:16	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	0 0 0 0	0 1 1 0	0 0 0 0	a		
	AND.B @(d:32,ERs.L),@aa:32	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	1 0 0 0	0 1 1 0	0 0 0 0	a a		
	AND.B @aa:16,@ERd						6	A	0 0 0 1	0 1 0 1	a	0 0 0 0	0	rd	0 1 1 0	0 0 0 0		
	AND.B @aa:16,@ERd+						6	A	0 0 0 1	0 1 0 1	a	1 0 0 0	0	rd	0 1 1 0	0 0 0 0		
	AND.B @aa:16,@ERd-						6	A	0 0 0 1	0 1 0 1	a	1 0 1 0	0	rd	0 1 1 0	0 0 0 0		
	AND.B @aa:16,@+ERd						6	A	0 0 0 1	0 1 0 1	a	1 0 0 1	0	rd	0 1 1 0	0 0 0 0		
	AND.B @aa:16,@-ERd						6	A	0 0 0 1	0 1 0 1	a	1 0 1 1	0	rd	0 1 1 0	0 0 0 0		
	AND.B @aa:16,@(d:2,ERd)						6	A	0 0 0 1	0 1 0 1	a	0 0 d d	0	rd	0 1 1 0	0 0 0 0		
	AND.B @aa:16,@(d:16,ERd)						6	A	0 0 0 1	0 1 0 1	a	1 1 0 0	0	rd	0 1 1 0	0 0 0 0	d	
	AND.B @aa:16,@(d:32,ERd)						6	A	0 0 0 1	0 1 0 1	a	1 1 0 0	1	rd	0 1 1 0	0 0 0 0	d d	
	AND.B @aa:16,@(d:16,Rd,B)						6	A	0 0 0 1	0 1 0 1	a	1 1 0 1	0	rd	0 1 1 0	0 0 0 0	d	
	AND.B @aa:16,@(d:16,Rd,W)						6	A	0 0 0 1	0 1 0 1	a	1 1 1 0	0	rd	0 1 1 0	0 0 0 0	d	
	AND.B @aa:16,@(d:16,ERd,L)						6	A	0 0 0 1	0 1 0 1	a	1 1 1 1	0	rd	0 1 1 0	0 0 0 0	d	
	AND.B @aa:16,@(d:32,Rd,B)						6	A	0 0 0 1	0 1 0 1	a	1 1 0 1	1	rd	0 1 1 0	0 0 0 0	d d	
	AND.B @aa:16,@(d:32,Rd,W)						6	A	0 0 0 1	0 1 0 1	a	1 1 1 0	1	rd	0 1 1 0	0 0 0 0	d d	
	AND.B @aa:16,@(d:32,ERd,L)						6	A	0 0 0 1	0 1 0 1	a	1 1 1 1	1	rd	0 1 1 0	0 0 0 0	d d	
	AND.B @aa:16,@aa:16						6	A	0 0 0 1	0 1 0 1	a	0 1 0 0	0 0 0 0	0 1 1 0	0 0 0 0	a		
	AND.B @aa:16,@aa:32						6	A	0 0 0 1	0 1 0 1	a	0 1 0 0	1 0 0 0	0 1 1 0	0 0 0 0	a a		
	AND.B @aa:32,@ERd						6	A	0 0 1 1	0 1 0 1	a a	0 0 0 0	0	rd	0 1 1 0	0 0 0 0		
	AND.B @aa:32,@ERd+						6	A	0 0 1 1	0 1 0 1	a a	1 0 0 0	0	rd	0 1 1 0	0 0 0 0		
	AND.B @aa:32,@ERd-						6	A	0 0 1 1	0 1 0 1	a a	1 0 1 0	0	rd	0 1 1 0	0 0 0 0		
	AND.B @aa:32,@+ERd						6	A	0 0 1 1	0 1 0 1	a a	1 0 0 1	0	rd	0 1 1 0	0 0 0 0		
	AND.B @aa:32,@-ERd						6	A	0 0 1 1	0 1 0 1	a a	1 0 1 1	0	rd	0 1 1 0	0 0 0 0		
	AND.B @aa:32,@(d:2,ERd)						6	A	0 0 1 1	0 1 0 1	a a	0 0 d d	0	rd	0 1 1 0	0 0 0 0		
	AND.B @aa:32,@(d:16,ERd)						6	A	0 0 1 1	0 1 0 1	a a	1 1 0 0	0	rd	0 1 1 0	0 0 0 0	d	
	AND.B @aa:32,@(d:32,ERd)						6	A	0 0 1 1	0 1 0 1	a a	1 1 0 0	1	rd	0 1 1 0	0 0 0 0	d d	
	AND.B @aa:32,@(d:16,Rd,B)						6	A	0 0 1 1	0 1 0 1	a a	1 1 0 1	0	rd	0 1 1 0	0 0 0 0	d	
	AND.B @aa:32,@(d:16,Rd,W)						6	A	0 0 1 1	0 1 0 1	a a	1 1 1 0	0	rd	0 1 1 0	0 0 0 0	d	
	AND.B @aa:32,@(d:16,ERd,L)						6	A	0 0 1 1	0 1 0 1	a a	1 1 1 1	0	rd	0 1 1 0	0 0 0 0	d	
	AND.B @aa:32,@(d:32,Rd,B)						6	A	0 0 1 1	0 1 0 1	a a	1 1 0 1	1	rd	0 1 1 0	0 0 0 0	d d	
	AND.B @aa:32,@(d:32,Rd,W)						6	A	0 0 1 1	0 1 0 1	a a	1 1 1 0	1	rd	0 1 1 0	0 0 0 0	d d	
AND.B @aa:32,@(d:32,ERd,L)						6	A	0 0 1 1	0 1 0 1	a a	1 1 1 1	1	rd	0 1 1 0	0 0 0 0	d d		
AND.B @aa:32,@aa:16						6	A	0 0 1 1	0 1 0 1	a a	0 1 0 0	0 0 0 0	0 1 1 0	0 0 0 0	a			

Instruction	Mnemonic	Opcode			Opcode			EA Ex- tension	Opcode			EA Extension		
		15	8	7	0	15	8		7	0	15		8	7
AND	AND.B @aa:32,@aa:32				6	A	0 0 1 1	0 1 0 1	a a	0 1 0 0	1 0 0 0	0 0 1 1	0 0 0 0	a a
	AND.W #hoc:16,Rd									7	9	0 1 1 0	rd	x
	AND.W #hoc:16,@ERd				0	1	5	1 1 1 0		0 0 0 0	0 rd	0 1 1 0	0 0 0 0	x
	AND.W #hoc:16,@ERd+				0	1	5	1 1 1 0		1 0 0 0	0 rd	0 1 1 0	0 0 0 0	x
	AND.W #hoc:16,@ERd-				0	1	5	1 1 1 0		1 0 1 0	0 rd	0 1 1 0	0 0 0 0	x
	AND.W #hoc:16,@+ERd				0	1	5	1 1 1 0		1 0 0 1	0 rd	0 1 1 0	0 0 0 0	x
	AND.W #hoc:16,@-ERd				0	1	5	1 1 1 0		1 0 1 1	0 rd	0 1 1 0	0 0 0 0	x
	AND.W #hoc:16,@(d:2,ERd)				0	1	5	1 1 1 0		0 0 d d	0 rd	0 1 1 0	0 0 0 0	x
	AND.W #hoc:16,@(d:16,ERd)				0	1	5	1 1 1 0		1 1 0 0	0 rd	0 1 1 0	0 0 0 0	d x
	AND.W #hoc:16,@(d:32,ERd)				0	1	5	1 1 1 0		1 1 0 0	1 rd	0 1 1 0	0 0 0 0	d d x
	AND.W #hoc:16,@(d:16,Rd,B)				0	1	5	1 1 1 0		1 1 0 1	0 rd	0 1 1 0	0 0 0 0	d x
	AND.W #hoc:16,@(d:16,Rd,W)				0	1	5	1 1 1 0		1 1 1 0	0 rd	0 1 1 0	0 0 0 0	d x
	AND.W #hoc:16,@(d:16,ERd,L)				0	1	5	1 1 1 0		1 1 1 1	0 rd	0 1 1 0	0 0 0 0	d x
	AND.W #hoc:16,@(d:32,Rd,B)				0	1	5	1 1 1 0		1 1 0 1	1 rd	0 1 1 0	0 0 0 0	d d x
	AND.W #hoc:16,@(d:32,Rd,W)				0	1	5	1 1 1 0		1 1 1 0	1 rd	0 1 1 0	0 0 0 0	d d x
	AND.W #hoc:16,@(d:32,ERd,L)				0	1	5	1 1 1 0		1 1 1 1	1 rd	0 1 1 0	0 0 0 0	d d x
	AND.W #hoc:16,@aa:16				0	1	5	1 1 1 0		0 1 0 0	0 0 0 0	0 1 1 0	0 0 0 0	a x
	AND.W #hoc:16,@aa:32				0	1	5	1 1 1 0		0 1 0 0	1 0 0 0	0 1 1 0	0 0 0 0	a a x
	AND.W Rs,Rd									6	6	rs	rd	
	AND.W Rs,@ERd				7	D	1	rd	0 0 0 0	6	6	rs	0 0 0 0	
	AND.W Rs,@ERd+				0	1	5	1 0 0 1		1 0 0 0	0 rd	0 1 1 0	rs	
	AND.W Rs,@ERd-				0	1	5	1 0 0 1		1 0 1 0	0 rd	0 1 1 0	rs	
	AND.W Rs,@+ERd				0	1	5	1 0 0 1		1 0 0 1	0 rd	0 1 1 0	rs	
	AND.W Rs,@-ERd				0	1	5	1 0 0 1		1 0 1 1	0 rd	0 1 1 0	rs	
	AND.W Rs,@(d:2,ERd)				0	1	5	1 0 0 1		0 0 d d	0 rd	0 1 1 0	rs	
	AND.W Rs,@(d:16,ERd)				0	1	5	1 0 0 1		1 1 0 0	0 rd	0 1 1 0	rs	d
	AND.W Rs,@(d:32,ERd)				0	1	5	1 0 0 1		1 1 0 0	1 rd	0 1 1 0	rs	d d
	AND.W Rs,@(d:16,Rd,B)				0	1	5	1 0 0 1		1 1 0 1	0 rd	0 1 1 0	rs	d
	AND.W Rs,@(d:16,Rd,W)				0	1	5	1 0 0 1		1 1 1 0	0 rd	0 1 1 0	rs	d
	AND.W Rs,@(d:16,ERd,L)				0	1	5	1 0 0 1		1 1 1 1	0 rd	0 1 1 0	rs	d
	AND.W Rs,@(d:32,Rd,B)				0	1	5	1 0 0 1		1 1 0 1	1 rd	0 1 1 0	rs	d d
	AND.W Rs,@(d:32,Rd,W)				0	1	5	1 0 0 1		1 1 1 0	1 rd	0 1 1 0	rs	d d
AND.W Rs,@(d:32,ERd,L)				0	1	5	1 0 0 1		1 1 1 1	1 rd	0 1 1 0	rs	d d	
AND.W Rs,@aa:16				6	B	0 0 0 1	1 0 0 0	a	6	6	rs	0 0 0 0		



Instruction	Mnemonic	Opcode			Opcode			EA Ex- tension	Opcode			EA Extension			
		15	8	7	0	15	8		7	0	15		8	7	0
AND	AND.W Rs,@aa:32				6	B	0 0 1 1	1 0 0 0 0	a a	6	6	rs	0 0 0 0		
	AND.W @ERs,Rd				7	C	1 rs	0 0 0 0 0		6	6	0 0 0 0 0	rd		
	AND.W @ERs+,Rd				0	1	5	1 0 1 0		1 0 0 0 0	0	rs	0 1 1 0	rd	
	AND.W @ERs-,Rd				0	1	5	1 0 1 0		1 0 1 0 0	0	rs	0 1 1 0	rd	
	AND.W @+ERs,Rd				0	1	5	1 0 1 0		1 0 0 1 0	0	rs	0 1 1 0	rd	
	AND.W @-ERs,Rd				0	1	5	1 0 1 0		1 0 1 1 0	0	rs	0 1 1 0	rd	
	AND.W @(d:2,ERs),Rd				0	1	5	1 0 1 0		0 0 d d 0	0	rs	0 1 1 0	rd	
	AND.W @(d:16,ERs),Rd				0	1	5	1 0 1 0		1 1 0 0 0	0	rs	0 1 1 0	rd	d
	AND.W @(d:32,ERs),Rd				0	1	5	1 0 1 0		1 1 0 0 1	rs	0 1 1 0	rd	d d	
	AND.W @(d:16,Rs,B),Rd				0	1	5	1 0 1 0		1 1 0 1 0	0	rs	0 1 1 0	rd	d
	AND.W @(d:16,Rs,W),Rd				0	1	5	1 0 1 0		1 1 1 0 0	0	rs	0 1 1 0	rd	d
	AND.W @(d:16,ERs,L),Rd				0	1	5	1 0 1 0		1 1 1 1 0	0	rs	0 1 1 0	rd	d
	AND.W @(d:32,Rs,B),Rd				0	1	5	1 0 1 0		1 1 0 1 1	rs	0 1 1 0	rd	d d	
	AND.W @(d:32,Rs,W),Rd				0	1	5	1 0 1 0		1 1 1 0 1	rs	0 1 1 0	rd	d d	
	AND.W @(d:32,ERs,L),Rd				0	1	5	1 0 1 0		1 1 1 1 1	rs	0 1 1 0	rd	d d	
	AND.W @aa:16,Rd				6	B	0 0 0 1	0 0 0 0 0	a	6	6	0 0 0 0	rd		
	AND.W @aa:32,Rd				6	B	0 0 1 1	0 0 0 0 0	a a	6	6	0 0 0 0 0	rd		
	AND.W @ERs,@ERd				7	C	1 rs	0 1 0 1		0 0 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @ERs,@ERd+				7	C	1 rs	0 1 0 1		1 0 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @ERs,@ERd-				7	C	1 rs	0 1 0 1		1 0 1 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @ERs,@+ERd				7	C	1 rs	0 1 0 1		1 0 0 1 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @ERs,@-ERd				7	C	1 rs	0 1 0 1		1 0 1 1 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @ERs,@(d:2,ERd)				7	C	1 rs	0 1 0 1		0 0 d d 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @ERs,@(d:16,ERd)				7	C	1 rs	0 1 0 1		1 1 0 0 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @ERs,@(d:32,ERd)				7	C	1 rs	0 1 0 1		1 1 0 0 1	rd	0 1 1 0	0 0 0 0	d d	
	AND.W @ERs,@(d:16,Rd,B)				7	C	1 rs	0 1 0 1		1 1 0 1 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @ERs,@(d:16,Rd,W)				7	C	1 rs	0 1 0 1		1 1 1 0 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @ERs,@(d:16,ERd,L)				7	C	1 rs	0 1 0 1		1 1 1 1 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @ERs,@(d:32,Rd,B)				7	C	1 rs	0 1 0 1		1 1 0 1 1	rd	0 1 1 0	0 0 0 0	d d	
	AND.W @ERs,@(d:32,Rd,W)				7	C	1 rs	0 1 0 1		1 1 1 0 1	rd	0 1 1 0	0 0 0 0	d d	
	AND.W @ERs,@(d:32,ERd,L)				7	C	1 rs	0 1 0 1		1 1 1 1 1	rd	0 1 1 0	0 0 0 0	d d	
	AND.W @ERs,@aa:16				7	C	1 rs	0 1 0 1		0 1 0 0 0	0 0 0 0 0	0 1 1 0	0 0 0 0	a	
AND.W @ERs,@aa:32				7	C	1 rs	0 1 0 1		0 1 0 0 1	0 0 0 0 0	0 1 1 0	0 0 0 0	a a		
AND.W @ERs+,@ERd				0	1	5	0 1 0 0		0 0 0 0 0	0	rd	0 1 1 0	0 0 0 0		

Instruction	Mnemonic	Opcode						EA Ex- tension	Opcode						EA Extension																		
		15	8	7	0	15	8		7	0	15	8	7	0																			
AND	AND.W @ERs+, @ERd+	0	1	5	0	1	0	0	6	D	0	rs	1	1	0	0	1	0	0	0	0	rd	0	1	1	0	0	0	0	0	0		
	AND.W @ERs+, @ERd-	0	1	5	0	1	0	0	6	D	0	rs	1	1	0	0	1	0	1	0	0	rd	0	1	1	0	0	0	0	0	0		
	AND.W @ERs+, @+ERd	0	1	5	0	1	0	0	6	D	0	rs	1	1	0	0	1	0	0	1	0	rd	0	1	1	0	0	0	0	0	0		
	AND.W @ERs+, @-ERd	0	1	5	0	1	0	0	6	D	0	rs	1	1	0	0	1	0	1	1	0	rd	0	1	1	0	0	0	0	0	0		
	AND.W @ERs+, @(d:2,ERd)	0	1	5	0	1	0	0	6	D	0	rs	1	1	0	0	0	0	d	d	0	rd	0	1	1	0	0	0	0	0	0		
	AND.W @ERs+, @(d:16,ERd)	0	1	5	0	1	0	0	6	D	0	rs	1	1	0	0	1	1	0	0	0	rd	0	1	1	0	0	0	0	0	0	d	
	AND.W @ERs+, @(d:32,ERd)	0	1	5	0	1	0	0	6	D	0	rs	1	1	0	0	1	1	0	0	1	rd	0	1	1	0	0	0	0	0	0	d	d
	AND.W @ERs+, @(d:16,Rd,B)	0	1	5	0	1	0	0	6	D	0	rs	1	1	0	0	1	1	0	1	0	rd	0	1	1	0	0	0	0	0	0	d	
	AND.W @ERs+, @(d:16,Rd,W)	0	1	5	0	1	0	0	6	D	0	rs	1	1	0	0	1	1	1	0	0	rd	0	1	1	0	0	0	0	0	0	d	
	AND.W @ERs+, @(d:16,ERd,L)	0	1	5	0	1	0	0	6	D	0	rs	1	1	0	0	1	1	1	1	0	rd	0	1	1	0	0	0	0	0	0	d	
	AND.W @ERs+, @(d:32,Rd,B)	0	1	5	0	1	0	0	6	D	0	rs	1	1	0	0	1	1	0	1	1	rd	0	1	1	0	0	0	0	0	0	d	d
	AND.W @ERs+, @(d:32,Rd,W)	0	1	5	0	1	0	0	6	D	0	rs	1	1	0	0	1	1	1	0	1	rd	0	1	1	0	0	0	0	0	0	d	d
	AND.W @ERs+, @(d:32,ERd,L)	0	1	5	0	1	0	0	6	D	0	rs	1	1	0	0	1	1	1	1	1	rd	0	1	1	0	0	0	0	0	0	d	d
	AND.W @ERs+, @aa:16	0	1	5	0	1	0	0	6	D	0	rs	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	a
	AND.W @ERs+, @aa:32	0	1	5	0	1	0	0	6	D	0	rs	1	1	0	0	0	1	0	0	1	0	0	0	0	0	1	1	0	0	0	a	a
	AND.W @ERs-, @ERd	0	1	5	0	1	1	0	6	D	0	rs	1	1	0	0	0	0	0	0	0	rd	0	1	1	0	0	0	0	0	0	0	
	AND.W @ERs-, @ERd+	0	1	5	0	1	1	0	6	D	0	rs	1	1	0	0	1	0	0	0	0	rd	0	1	1	0	0	0	0	0	0	0	
	AND.W @ERs-, @ERd-	0	1	5	0	1	1	0	6	D	0	rs	1	1	0	0	1	0	1	0	0	rd	0	1	1	0	0	0	0	0	0	0	
	AND.W @ERs-, @+ERd	0	1	5	0	1	1	0	6	D	0	rs	1	1	0	0	1	0	0	1	0	rd	0	1	1	0	0	0	0	0	0	0	
	AND.W @ERs-, @-ERd	0	1	5	0	1	1	0	6	D	0	rs	1	1	0	0	1	0	1	1	0	rd	0	1	1	0	0	0	0	0	0	0	
	AND.W @ERs-, @(d:2,ERd)	0	1	5	0	1	1	0	6	D	0	rs	1	1	0	0	0	0	d	d	0	rd	0	1	1	0	0	0	0	0	0	0	
	AND.W @ERs-, @(d:16,ERd)	0	1	5	0	1	1	0	6	D	0	rs	1	1	0	0	1	1	0	0	0	rd	0	1	1	0	0	0	0	0	0	d	
	AND.W @ERs-, @(d:32,ERd)	0	1	5	0	1	1	0	6	D	0	rs	1	1	0	0	1	1	0	0	1	rd	0	1	1	0	0	0	0	0	0	d	d
	AND.W @ERs-, @(d:16,Rd,B)	0	1	5	0	1	1	0	6	D	0	rs	1	1	0	0	1	1	0	1	0	rd	0	1	1	0	0	0	0	0	0	d	
	AND.W @ERs-, @(d:16,Rd,W)	0	1	5	0	1	1	0	6	D	0	rs	1	1	0	0	1	1	1	0	0	rd	0	1	1	0	0	0	0	0	0	d	
	AND.W @ERs-, @(d:16,ERd,L)	0	1	5	0	1	1	0	6	D	0	rs	1	1	0	0	1	1	1	1	0	rd	0	1	1	0	0	0	0	0	0	d	
	AND.W @ERs-, @(d:32,Rd,B)	0	1	5	0	1	1	0	6	D	0	rs	1	1	0	0	1	1	0	1	1	rd	0	1	1	0	0	0	0	0	0	d	d
	AND.W @ERs-, @(d:32,Rd,W)	0	1	5	0	1	1	0	6	D	0	rs	1	1	0	0	1	1	1	0	1	rd	0	1	1	0	0	0	0	0	0	d	d
	AND.W @ERs-, @(d:32,ERd,L)	0	1	5	0	1	1	0	6	D	0	rs	1	1	0	0	1	1	1	1	1	rd	0	1	1	0	0	0	0	0	0	d	d
	AND.W @ERs-, @aa:16	0	1	5	0	1	1	0	6	D	0	rs	1	1	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	a	
	AND.W @ERs-, @aa:32	0	1	5	0	1	1	0	6	D	0	rs	1	1	0	0	0	1	0	0	1	0	0	0	0	1	1	0	0	0	0	a	a
	AND.W @+ERs, @ERd	0	1	5	0	1	0	1	6	D	0	rs	1	1	0	0	0	0	0	0	0	rd	0	1	1	0	0	0	0	0	0	0	
AND.W @+ERs, @ERd+	0	1	5	0	1	0	1	6	D	0	rs	1	1	0	0	1	0	0	0	0	rd	0	1	1	0	0	0	0	0	0	0		
AND.W @+ERs, @ERd-	0	1	5	0	1	0	1	6	D	0	rs	1	1	0	0	1	0	1	0	0	rd	0	1	1	0	0	0	0	0	0	0		



Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension																				
		15	8	7	0	15	8		7	0	15	8	7	0		EA Extension																			
AND	AND.W @+ERs,@+ERd	0	1	5	0	1	0	1	6	D	0	rs	1	1	0	0	1	0	0	1	0	rd	0	1	1	0	0	0	0	0					
	AND.W @+ERs,@-ERd	0	1	5	0	1	0	1	6	D	0	rs	1	1	0	0	1	0	1	1	0	rd	0	1	1	0	0	0	0	0					
	AND.W @+ERs,@(d:2,ERd)	0	1	5	0	1	0	1	6	D	0	rs	1	1	0	0	0	0	d	d	0	rd	0	1	1	0	0	0	0	0					
	AND.W @+ERs,@(d:16,ERd)	0	1	5	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	0	0	0	rd	0	1	1	0	0	0	0	0	d				
	AND.W @+ERs,@(d:32,ERd)	0	1	5	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	0	0	1	rd	0	1	1	0	0	0	0	0	d d				
	AND.W @+ERs,@(d:16,Rd,B)	0	1	5	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	0	1	0	rd	0	1	1	0	0	0	0	0	d				
	AND.W @+ERs,@(d:16,Rd,W)	0	1	5	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	1	0	0	rd	0	1	1	0	0	0	0	0	d				
	AND.W @+ERs,@(d:16,ERd,L)	0	1	5	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	1	1	0	rd	0	1	1	0	0	0	0	0	d				
	AND.W @+ERs,@(d:32,Rd,B)	0	1	5	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	0	1	1	rd	0	1	1	0	0	0	0	0	d d				
	AND.W @+ERs,@(d:32,Rd,W)	0	1	5	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	1	0	1	rd	0	1	1	0	0	0	0	0	d d				
	AND.W @+ERs,@(d:32,ERd,L)	0	1	5	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	1	1	1	rd	0	1	1	0	0	0	0	0	d d				
	AND.W @+ERs,@aa:16	0	1	5	0	1	0	1	6	D	0	rs	1	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	a
	AND.W @+ERs,@aa:32	0	1	5	0	1	0	1	6	D	0	rs	1	1	0	0	0	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	a a
	AND.W @-ERs,@ERd	0	1	5	0	1	1	1	6	D	0	rs	1	1	0	0	0	0	0	0	0	rd	0	1	1	0	0	0	0	0					
	AND.W @-ERs,@ERd+	0	1	5	0	1	1	1	6	D	0	rs	1	1	0	0	1	0	0	0	0	rd	0	1	1	0	0	0	0	0					
	AND.W @-ERs,@ERd-	0	1	5	0	1	1	1	6	D	0	rs	1	1	0	0	1	0	1	0	0	rd	0	1	1	0	0	0	0	0					
	AND.W @-ERs,@+ERd	0	1	5	0	1	1	1	6	D	0	rs	1	1	0	0	1	0	0	1	0	rd	0	1	1	0	0	0	0	0					
	AND.W @-ERs,@-ERd	0	1	5	0	1	1	1	6	D	0	rs	1	1	0	0	1	0	1	1	0	rd	0	1	1	0	0	0	0	0					
	AND.W @-ERs,@(d:2,ERd)	0	1	5	0	1	1	1	6	D	0	rs	1	1	0	0	0	0	d	d	0	rd	0	1	1	0	0	0	0	0					
	AND.W @-ERs,@(d:16,ERd)	0	1	5	0	1	1	1	6	D	0	rs	1	1	0	0	1	1	0	0	0	rd	0	1	1	0	0	0	0	0	d				
	AND.W @-ERs,@(d:32,ERd)	0	1	5	0	1	1	1	6	D	0	rs	1	1	0	0	1	1	0	0	1	rd	0	1	1	0	0	0	0	0	d d				
	AND.W @-ERs,@(d:16,Rd,B)	0	1	5	0	1	1	1	6	D	0	rs	1	1	0	0	1	1	0	1	0	rd	0	1	1	0	0	0	0	0	d				
	AND.W @-ERs,@(d:16,Rd,W)	0	1	5	0	1	1	1	6	D	0	rs	1	1	0	0	1	1	1	0	0	rd	0	1	1	0	0	0	0	0	d				
	AND.W @-ERs,@(d:16,ERd,L)	0	1	5	0	1	1	1	6	D	0	rs	1	1	0	0	1	1	1	1	0	rd	0	1	1	0	0	0	0	0	d				
	AND.W @-ERs,@(d:32,Rd,B)	0	1	5	0	1	1	1	6	D	0	rs	1	1	0	0	1	1	0	1	1	rd	0	1	1	0	0	0	0	0	d d				
	AND.W @-ERs,@(d:32,Rd,W)	0	1	5	0	1	1	1	6	D	0	rs	1	1	0	0	1	1	1	0	1	rd	0	1	1	0	0	0	0	0	d d				
	AND.W @-ERs,@(d:32,ERd,L)	0	1	5	0	1	1	1	6	D	0	rs	1	1	0	0	1	1	1	1	1	rd	0	1	1	0	0	0	0	0	d d				
	AND.W @-ERs,@aa:16	0	1	5	0	1	1	1	6	D	0	rs	1	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	a
AND.W @-ERs,@aa:32	0	1	5	0	1	1	1	6	D	0	rs	1	1	0	0	0	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	a a	
AND.W @(d:2,ERs),@ERd	0	1	5	0	1	d	d	6	9	0	rs	1	1	0	0	0	0	0	0	0	rd	0	1	1	0	0	0	0	0						
AND.W @(d:2,ERs),@ERd+	0	1	5	0	1	d	d	6	9	0	rs	1	1	0	0	1	0	0	0	0	rd	0	1	1	0	0	0	0	0						
AND.W @(d:2,ERs),@ERd-	0	1	5	0	1	d	d	6	9	0	rs	1	1	0	0	1	0	1	0	0	rd	0	1	1	0	0	0	0	0						
AND.W @(d:2,ERs),@+ERd	0	1	5	0	1	d	d	6	9	0	rs	1	1	0	0	1	0	0	1	0	rd	0	1	1	0	0	0	0	0						
AND.W @(d:2,ERs),@-ERd	0	1	5	0	1	d	d	6	9	0	rs	1	1	0	0	1	0	1	1	0	rd	0	1	1	0	0	0	0	0						

Instruction	Mnemonic	Opcode						EA Ex- tension	Opcode						EA Extension									
		15	8	7	0				15	8	7	0												
AND	AND.W @(d:2,ERs),@(d:2,ERd)	0	1	5	0	1	d	d	6	9	0	rs	1	1	0	0	0	0	d	d				
	AND.W @(d:2,ERs),@(d:16,ERd)	0	1	5	0	1	d	d	6	9	0	rs	1	1	0	0	1	1	0	0	d			
	AND.W @(d:2,ERs),@(d:32,ERd)	0	1	5	0	1	d	d	6	9	0	rs	1	1	0	0	1	1	0	0	d	d		
	AND.W @(d:2,ERs),@(d:16,Rd.B)	0	1	5	0	1	d	d	6	9	0	rs	1	1	0	0	1	1	0	0	0	d	d	
	AND.W @(d:2,ERs),@(d:16,Rd.W)	0	1	5	0	1	d	d	6	9	0	rs	1	1	0	0	1	1	0	0	0	0	d	d
	AND.W @(d:2,ERs),@(d:16,ERd.L)	0	1	5	0	1	d	d	6	9	0	rs	1	1	0	0	1	1	1	0	0	0	d	d
	AND.W @(d:2,ERs),@(d:32,Rd.B)	0	1	5	0	1	d	d	6	9	0	rs	1	1	0	0	1	1	0	1	0	0	d	d
	AND.W @(d:2,ERs),@(d:32,Rd.W)	0	1	5	0	1	d	d	6	9	0	rs	1	1	0	0	1	1	0	1	0	0	d	d
	AND.W @(d:2,ERs),@(d:32,ERd.L)	0	1	5	0	1	d	d	6	9	0	rs	1	1	0	0	1	1	1	1	0	0	d	d
	AND.W @(d:2,ERs),@aa:16	0	1	5	0	1	d	d	6	9	0	rs	1	1	0	0	0	1	0	0	0	0	a	a
	AND.W @(d:2,ERs),@aa:32	0	1	5	0	1	d	d	6	9	0	rs	1	1	0	0	0	1	0	0	0	0	a	a
	AND.W @(d:16,ERs),@ERd	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	0	0	0	0	0	0		
	AND.W @(d:16,ERs),@ERd+	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	0	0	0	0	0	0		
	AND.W @(d:16,ERs),@ERd-	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	0	0	0	0	0	0		
	AND.W @(d:16,ERs),@+ERd	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	0	1	0	0	0	0		
	AND.W @(d:16,ERs),@-ERd	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	0	1	1	0	0	0		
	AND.W @(d:16,ERs),@(d:2,ERd)	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	0	0	d	0	0	0		
	AND.W @(d:16,ERs),@(d:16,ERd)	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	0	1	1	0	0	0	d	
	AND.W @(d:16,ERs),@(d:32,ERd)	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	0	1	1	0	1	0	d	d
	AND.W @(d:16,ERs),@(d:16,Rd.B)	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	0	1	1	0	0	0	d	
	AND.W @(d:16,ERs),@(d:16,Rd.W)	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	0	1	1	0	0	0	d	
	AND.W @(d:16,ERs),@(d:16,ERd.L)	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	0	1	1	1	0	0	d	
	AND.W @(d:16,ERs),@(d:32,Rd.B)	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	0	1	1	1	0	0	d	d
	AND.W @(d:16,ERs),@(d:32,Rd.W)	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	0	1	1	1	0	0	d	d
	AND.W @(d:16,ERs),@(d:32,ERd.L)	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	0	1	1	1	1	0	d	d
	AND.W @(d:16,ERs),@aa:16	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	0	0	1	0	0	0	a	
	AND.W @(d:16,ERs),@aa:32	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	0	1	0	0	0	0	a	a
	AND.W @(d:32,ERs),@ERd	7	8	0	rs	0	1	0	0	6	B	0	0	1	0	1	1	0	0	0	0	0		
	AND.W @(d:32,ERs),@ERd+	7	8	0	rs	0	1	0	0	6	B	0	0	1	0	1	1	0	0	0	0	0		
	AND.W @(d:32,ERs),@ERd-	7	8	0	rs	0	1	0	0	6	B	0	0	1	0	1	1	0	0	0	0	0		
	AND.W @(d:32,ERs),@+ERd	7	8	0	rs	0	1	0	0	6	B	0	0	1	0	1	1	0	0	1	0	0		
	AND.W @(d:32,ERs),@-ERd	7	8	0	rs	0	1	0	0	6	B	0	0	1	0	1	1	0	0	1	0	0		
	AND.W @(d:32,ERs),@(d:2,ERd)	7	8	0	rs	0	1	0	0	6	B	0	0	1	0	1	1	0	0	d	0	0		
AND.W @(d:32,ERs),@(d:16,ERd)	7	8	0	rs	0	1	0	0	6	B	0	0	1	0	1	1	0	0	0	0	0	d		

Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension		
		15	8	7	0	15	8		7	0	15	8	7	0			
AND	AND.W @(d:32.ERs),@(d:32.ERd)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.W @(d:32.ERs),@(d:16.Rd.B)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @(d:32.ERs),@(d:16.Rd.W)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @(d:32.ERs),@(d:16.ERd.L)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @(d:32.ERs),@(d:32.Rd.B)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.W @(d:32.ERs),@(d:32.Rd.W)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.W @(d:32.ERs),@(d:32.ERd.L)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.W @(d:32.ERs),@aa:16	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 1 1 0	0 0 0 0	a
	AND.W @(d:32.ERs),@aa:32	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 1 1 0	0 0 0 0	a a
	AND.W @(d:16.Rs.B),@ERd	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @(d:16.Rs.B),@ERd+	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @(d:16.Rs.B),@ERd-	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @(d:16.Rs.B),@+ERd	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 1 0	0 0 0 0	
	AND.W @(d:16.Rs.B),@-ERd	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 1 1 0	0 0 0 0	
	AND.W @(d:16.Rs.B),@(d:2.ERd)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 1 1 0	0 0 0 0	
	AND.W @(d:16.Rs.B),@(d:16.ERd)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @(d:16.Rs.B),@(d:32.ERd)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.W @(d:16.Rs.B),@(d:16.Rd.B)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @(d:16.Rs.B),@(d:16.Rd.W)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @(d:16.Rs.B),@(d:16.ERd.L)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @(d:16.Rs.B),@(d:32.Rd.B)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.W @(d:16.Rs.B),@(d:32.Rd.W)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.W @(d:16.Rs.B),@aa:16	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	0	0 0 0 0	0 1 1 0	0 0 0 0	a
	AND.W @(d:16.Rs.B),@aa:32	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	1	0 0 0 0	0 1 1 0	0 0 0 0	a a
	AND.W @(d:16.Rs.W),@ERd	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @(d:16.Rs.W),@ERd+	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @(d:16.Rs.W),@ERd-	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @(d:16.Rs.W),@+ERd	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 1 0	0 0 0 0	
	AND.W @(d:16.Rs.W),@-ERd	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 1 1 0	0 0 0 0	
	AND.W @(d:16.Rs.W),@(d:2.ERd)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 1 1 0	0 0 0 0	
AND.W @(d:16.Rs.W),@(d:16.ERd)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 1 1 0	0 0 0 0	d	
AND.W @(d:16.Rs.W),@(d:32.ERd)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 1 1 0	0 0 0 0	d d	
AND.W @(d:16.Rs.W),@(d:16.Rd.B)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 1 1 0	0 0 0 0	d	

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
AND	AND.W @(d:16,Rs.W),@(d:16,Rd.W)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @(d:16,Rs.W),@(d:16,ERd.L)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @(d:16,Rs.W),@(d:32,Rd.B)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.W @(d:16,Rs.W),@(d:32,Rd.W)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.W @(d:16,Rs.W),@(d:32,ERd.L)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.W @(d:16,Rs.W),@aa:16	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 1 1 0	0 0 0 0	a	
	AND.W @(d:16,Rs.W),@aa:32	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 1 0	0 0 0 0	a a	
	AND.W @(d:16,ERs.L),@ERd	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @(d:16,ERs.L),@ERd+	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @(d:16,ERs.L),@ERd-	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @(d:16,ERs.L),@+ERd	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 1 0	0 0 0 0	
	AND.W @(d:16,ERs.L),@-ERd	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 1 1 0	0 0 0 0	
	AND.W @(d:16,ERs.L),@(d:2,ERd)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 1 1 0	0 0 0 0	
	AND.W @(d:16,ERs.L),@(d:16,ERd)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @(d:16,ERs.L),@(d:32,ERd)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.W @(d:16,ERs.L),@(d:16,Rd.B)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @(d:16,ERs.L),@(d:16,Rd.W)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @(d:16,ERs.L),@(d:16,ERd.L)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @(d:16,ERs.L),@(d:32,Rd.B)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.W @(d:16,ERs.L),@(d:32,Rd.W)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.W @(d:16,ERs.L),@(d:32,ERd.L)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.W @(d:16,ERs.L),@aa:16	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 1 1 0	0 0 0 0	a	
	AND.W @(d:16,ERs.L),@aa:32	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 1 0	0 0 0 0	a a	
	AND.W @(d:32,Rs.B),@ERd	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @(d:32,Rs.B),@ERd+	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @(d:32,Rs.B),@ERd-	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @(d:32,Rs.B),@+ERd	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 1 1 0	0 0 0 0	
	AND.W @(d:32,Rs.B),@-ERd	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 1 1 0	0 0 0 0	
	AND.W @(d:32,Rs.B),@(d:2,ERd)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 1 1 0	0 0 0 0	
	AND.W @(d:32,Rs.B),@(d:16,ERd)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @(d:32,Rs.B),@(d:32,ERd)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.W @(d:32,Rs.B),@(d:16,Rd.B)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 1 1 0	0 0 0 0	d



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension																							
		15	8	7	0	15	8	7	0		15	8	7	0																								
AND	AND.W @(d:32,Rs.B),@(d:16,Rd,W)	7	8	0	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	0	0	rd	0	1	1	0	0	0	0	0	0	d	
	AND.W @(d:32,Rs.B),@(d:16,ERd,L)	7	8	0	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	1	0	rd	0	1	1	0	0	0	0	0	0	d	
	AND.W @(d:32,Rs.B),@(d:32,Rd.B)	7	8	0	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	1	1	rd	0	1	1	0	0	0	0	0	0	d	d
	AND.W @(d:32,Rs.B),@(d:32,Rd,W)	7	8	0	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	0	1	rd	0	1	1	0	0	0	0	0	0	d	d
	AND.W @(d:32,Rs.B),@(d:32,ERd,L)	7	8	0	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	1	1	rd	0	1	1	0	0	0	0	0	0	d	d
	AND.W @(d:32,Rs.B),@aa:16	7	8	0	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	d	d	0	1	0	0	0	0	rd	0	1	1	0	0	0	0	0	0	a
	AND.W @(d:32,Rs.B),@aa:32	7	8	0	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	d	d	0	1	0	0	0	1	rd	0	1	1	0	0	0	0	0	a	a
	AND.W @(d:32,Rs.W),@ERd	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	0	0	0	0	0	0	rd	0	1	1	0	0	0	0	0		
	AND.W @(d:32,Rs.W),@ERd+	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	0	0	0	0	0	rd	0	1	1	0	0	0	0	0		
	AND.W @(d:32,Rs.W),@ERd-	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	0	1	0	0	0	rd	0	1	1	0	0	0	0	0		
	AND.W @(d:32,Rs.W),@+ERd	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	0	0	1	0	0	rd	0	1	1	0	0	0	0	0		
	AND.W @(d:32,Rs.W),@-ERd	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	0	1	1	0	0	rd	0	1	1	0	0	0	0	0		
	AND.W @(d:32,Rs.W),@(d:2,ERd)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	0	0	d	d	0	0	rd	0	1	1	0	0	0	0	0		
	AND.W @(d:32,Rs.W),@(d:16,ERd)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	0	0	0	rd	0	1	1	0	0	0	0	0	d	
	AND.W @(d:32,Rs.W),@(d:32,ERd)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	0	1	rd	0	1	1	0	0	0	0	0	d	d	
	AND.W @(d:32,Rs.W),@(d:16,Rd.B)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	1	0	rd	0	1	1	0	0	0	0	0	d		
	AND.W @(d:32,Rs.W),@(d:16,Rd,W)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	0	0	rd	0	1	1	0	0	0	0	0	d		
	AND.W @(d:32,Rs.W),@(d:16,ERd,L)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	1	0	rd	0	1	1	0	0	0	0	0	d		
	AND.W @(d:32,Rs.W),@(d:32,Rd.B)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	1	1	rd	0	1	1	0	0	0	0	0	d	d	
	AND.W @(d:32,Rs.W),@(d:32,Rd,W)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	0	1	rd	0	1	1	0	0	0	0	0	d	d	
	AND.W @(d:32,Rs.W),@(d:32,ERd,L)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	1	1	rd	0	1	1	0	0	0	0	0	d	d	
	AND.W @(d:32,Rs.W),@aa:16	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	0	1	0	0	0	0	rd	0	1	1	0	0	0	0	0	a	
	AND.W @(d:32,Rs.W),@aa:32	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	0	1	0	0	0	1	rd	0	1	1	0	0	0	0	0	a	a
	AND.W @(d:32,ERs.L),@ERd	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	0	0	0	0	0	0	rd	0	1	1	0	0	0	0	0		
	AND.W @(d:32,ERs.L),@ERd+	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	0	0	0	0	0	rd	0	1	1	0	0	0	0	0		
	AND.W @(d:32,ERs.L),@ERd-	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	0	1	0	0	0	rd	0	1	1	0	0	0	0	0		
	AND.W @(d:32,ERs.L),@+ERd	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	0	0	1	0	0	rd	0	1	1	0	0	0	0	0		
	AND.W @(d:32,ERs.L),@-ERd	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	0	1	1	0	0	rd	0	1	1	0	0	0	0	0		
	AND.W @(d:32,ERs.L),@(d:2,ERd)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	0	0	d	d	0	0	rd	0	1	1	0	0	0	0	0		
	AND.W @(d:32,ERs.L),@(d:16,ERd)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	0	0	0	rd	0	1	1	0	0	0	0	0	d	
AND.W @(d:32,ERs.L),@(d:32,ERd)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	0	1	rd	0	1	1	0	0	0	0	0	d	d		
AND.W @(d:32,ERs.L),@(d:16,Rd.B)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	1	0	rd	0	1	1	0	0	0	0	0	d			
AND.W @(d:32,ERs.L),@(d:16,Rd,W)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	0	0	rd	0	1	1	0	0	0	0	0	d			

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension						
		15	8	7	0		15	8	7	0							
AND	AND.W @(d:32,ERs.L),@(d:16,ERd.L)	7	8	0	rs 0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 1 1 0	0 0 0 0	d	
	AND.W @(d:32,ERs.L),@(d:32,Rd.B)	7	8	0	rs 0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 1 1 0	0 0 0 0	d d	
	AND.W @(d:32,ERs.L),@(d:32,Rd.W)	7	8	0	rs 0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 1 1 0	0 0 0 0	d d	
	AND.W @(d:32,ERs.L),@(d:32,ERd.L)	7	8	0	rs 0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 1 1 0	0 0 0 0	d d	
	AND.W @(d:32,ERs.L),@aa:16	7	8	0	rs 0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 1 1 0	0 0 0 0	a	
	AND.W @(d:32,ERs.L),@aa:32	7	8	0	rs 0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 1 1 0	0 0 0 0	a a	
	AND.W @aa:16,@ERd					6	B	0 0 0 1	0 1 0 1	a		0 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @aa:16,@ERd+					6	B	0 0 0 1	0 1 0 1	a		1 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @aa:16,@ERd-					6	B	0 0 0 1	0 1 0 1	a		1 0 1 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @aa:16,@+ERd					6	B	0 0 0 1	0 1 0 1	a		1 0 0 1	0	rd	0 1 1 0	0 0 0 0	
	AND.W @aa:16,@-ERd					6	B	0 0 0 1	0 1 0 1	a		1 0 1 1	0	rd	0 1 1 0	0 0 0 0	
	AND.W @aa:16,@(d:2,ERd)					6	B	0 0 0 1	0 1 0 1	a		0 0 d d	0	rd	0 1 1 0	0 0 0 0	
	AND.W @aa:16,@(d:16,ERd)					6	B	0 0 0 1	0 1 0 1	a		1 1 0 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @aa:16,@(d:32,ERd)					6	B	0 0 0 1	0 1 0 1	a		1 1 0 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.W @aa:16,@(d:16,Rd.B)					6	B	0 0 0 1	0 1 0 1	a		1 1 0 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @aa:16,@(d:16,Rd.W)					6	B	0 0 0 1	0 1 0 1	a		1 1 1 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @aa:16,@(d:16,ERd.L)					6	B	0 0 0 1	0 1 0 1	a		1 1 1 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @aa:16,@(d:32,Rd.B)					6	B	0 0 0 1	0 1 0 1	a		1 1 0 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.W @aa:16,@(d:32,Rd.W)					6	B	0 0 0 1	0 1 0 1	a		1 1 1 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.W @aa:16,@(d:32,ERd.L)					6	B	0 0 0 1	0 1 0 1	a		1 1 1 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.W @aa:16,@aa:16					6	B	0 0 0 1	0 1 0 1	a		0 1 0 0	0	0 0 0 0	0 1 1 0	0 0 0 0	a
	AND.W @aa:16,@aa:32					6	B	0 0 0 1	0 1 0 1	a		0 1 0 0	1	0 0 0 0	0 1 1 0	0 0 0 0	a a
	AND.W @aa:32,@ERd					6	B	0 0 1 1	0 1 0 1	a a		0 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @aa:32,@ERd+					6	B	0 0 1 1	0 1 0 1	a a		1 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @aa:32,@ERd-					6	B	0 0 1 1	0 1 0 1	a a		1 0 1 0	0	rd	0 1 1 0	0 0 0 0	
	AND.W @aa:32,@+ERd					6	B	0 0 1 1	0 1 0 1	a a		1 0 0 1	0	rd	0 1 1 0	0 0 0 0	
	AND.W @aa:32,@-ERd					6	B	0 0 1 1	0 1 0 1	a a		1 0 1 1	0	rd	0 1 1 0	0 0 0 0	
	AND.W @aa:32,@(d:2,ERd)					6	B	0 0 1 1	0 1 0 1	a a		0 0 d d	0	rd	0 1 1 0	0 0 0 0	
	AND.W @aa:32,@(d:16,ERd)					6	B	0 0 1 1	0 1 0 1	a a		1 1 0 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @aa:32,@(d:32,ERd)					6	B	0 0 1 1	0 1 0 1	a a		1 1 0 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.W @aa:32,@(d:16,Rd.B)					6	B	0 0 1 1	0 1 0 1	a a		1 1 0 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.W @aa:32,@(d:16,Rd.W)					6	B	0 0 1 1	0 1 0 1	a a		1 1 1 0	0	rd	0 1 1 0	0 0 0 0	d



Instruction	Mnemonic	Opcode				EA Ex- tension	Opcode				EA Extension				
		15	8	7	0		15	8	7	0					
AND	AND.W @aa:32,@(d:16,ERd.L)			6	B	0 0 1 1	0 1 0 1	a a	1 1 1 1	0 rd	0 1 1 0	0 0 0 0	d		
	AND.W @aa:32,@(d:32,Rd.B)			6	B	0 0 1 1	0 1 0 1	a a	1 1 0 1	1 rd	0 1 1 0	0 0 0 0	d d		
	AND.W @aa:32,@(d:32,Rd.W)			6	B	0 0 1 1	0 1 0 1	a a	1 1 1 0	1 rd	0 1 1 0	0 0 0 0	d d		
	AND.W @aa:32,@(d:32,ERd.L)			6	B	0 0 1 1	0 1 0 1	a a	1 1 1 1	1 rd	0 1 1 0	0 0 0 0	d d		
	AND.W @aa:32,@aa:16			6	B	0 0 1 1	0 1 0 1	a a	0 1 0 0	0 0 0 0	0 1 1 0	0 0 0 0	a		
	AND.W @aa:32,@aa:32			6	B	0 0 1 1	0 1 0 1	a a	0 1 0 0	1 0 0 0	0 1 1 0	0 0 0 0	a a		
	AND.L #xx:16,ERd									7	A	0 1 1 0	1 rd	x	
	AND.L #xx:32,ERd									7	A	0 1 1 0	0 rd	x x	
	AND.L #xx:16,@ERd			0	1	0	1 1 1 0		0 0 0 0	0 rd	0 1 1 0	0 0 0 0		x	
	AND.L #xx:16,@ERd+			0	1	0	1 1 1 0		1 0 0 0	0 rd	0 1 1 0	0 0 0 0		x	
	AND.L #xx:16,@ERd-			0	1	0	1 1 1 0		1 0 1 0	0 rd	0 1 1 0	0 0 0 0		x	
	AND.L #xx:16,@+ERd			0	1	0	1 1 1 0		1 0 0 1	0 rd	0 1 1 0	0 0 0 0		x	
	AND.L #xx:16,@-ERd			0	1	0	1 1 1 0		1 0 1 1	0 rd	0 1 1 0	0 0 0 0		x	
	AND.L #xx:16,@(d:2,ERd)			0	1	0	1 1 1 0		0 0 d d	0 rd	0 1 1 0	0 0 0 0		x	
	AND.L #xx:16,@(d:16,ERd)			0	1	0	1 1 1 0		1 1 0 0	0 rd	0 1 1 0	0 0 0 0	d	x	
	AND.L #xx:16,@(d:32,ERd)			0	1	0	1 1 1 0		1 1 0 0	1 rd	0 1 1 0	0 0 0 0	d d	x	
	AND.L #xx:16,@(d:16,Rd.B)			0	1	0	1 1 1 0		1 1 0 1	0 rd	0 1 1 0	0 0 0 0	d	x	
	AND.L #xx:16,@(d:16,Rd.W)			0	1	0	1 1 1 0		1 1 1 0	0 rd	0 1 1 0	0 0 0 0	d	x	
	AND.L #xx:16,@(d:16,ERd.L)			0	1	0	1 1 1 0		1 1 1 1	0 rd	0 1 1 0	0 0 0 0	d	x	
	AND.L #xx:16,@(d:32,Rd.B)			0	1	0	1 1 1 0		1 1 0 1	1 rd	0 1 1 0	0 0 0 0	d d	x	
	AND.L #xx:16,@(d:32,Rd.W)			0	1	0	1 1 1 0		1 1 1 0	1 rd	0 1 1 0	0 0 0 0	d d	x	
	AND.L #xx:16,@(d:32,ERd.L)			0	1	0	1 1 1 0		1 1 1 1	1 rd	0 1 1 0	0 0 0 0	d d	x	
	AND.L #xx:16,@aa:16			0	1	0	1 1 1 0		0 1 0 0	0 0 0 0	0 1 1 0	0 0 0 0	a	x	
	AND.L #xx:16,@aa:32			0	1	0	1 1 1 0		0 1 0 0	1 0 0 0	0 1 1 0	0 0 0 0	a a	x	
	AND.L #xx:32,@ERd			0	1	0	1 1 1 0		0 0 0 0	0 rd	0 1 1 0	1 0 0 0		x x	
	AND.L #xx:32,@ERd+			0	1	0	1 1 1 0		1 0 0 0	0 rd	0 1 1 0	1 0 0 0		x x	
	AND.L #xx:32,@ERd-			0	1	0	1 1 1 0		1 0 1 0	0 rd	0 1 1 0	1 0 0 0		x x	
	AND.L #xx:32,@+ERd			0	1	0	1 1 1 0		1 0 0 1	0 rd	0 1 1 0	1 0 0 0		x x	
	AND.L #xx:32,@-ERd			0	1	0	1 1 1 0		1 0 1 1	0 rd	0 1 1 0	1 0 0 0		x x	
	AND.L #xx:32,@(d:2,ERd)			0	1	0	1 1 1 0		0 0 d d	0 rd	0 1 1 0	1 0 0 0		x x	
	AND.L #xx:32,@(d:16,ERd)			0	1	0	1 1 1 0		1 1 0 0	0 rd	0 1 1 0	1 0 0 0	d	x x	
	AND.L #xx:32,@(d:32,ERd)			0	1	0	1 1 1 0		1 1 0 0	1 rd	0 1 1 0	1 0 0 0	d d	x x	
AND.L #xx:32,@(d:16,Rd.B)			0	1	0	1 1 1 0		1 1 0 1	0 rd	0 1 1 0	1 0 0 0	d	x x		
AND.L #xx:32,@(d:16,Rd.W)			0	1	0	1 1 1 0		1 1 1 0	0 rd	0 1 1 0	1 0 0 0	d	x x		

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension																			
		15	8	7	0		15	8	7	0																				
AND	AND.L #xx:32,@(d:16,ERd.L)				0	1	0	1	1	1	0	1	1	1	0	rd	0	1	1	0	1	0	0	0	d			x	x	
	AND.L #xx:32,@(d:32,Rd.B)				0	1	0	1	1	1	0	1	1	0	1	rd	0	1	1	0	1	0	0	0	d	d		x	x	
	AND.L #xx:32,@(d:32,Rd.W)				0	1	0	1	1	1	0	1	1	0	1	rd	0	1	1	0	1	0	0	0	d	d		x	x	
	AND.L #xx:32,@(d:32,ERd.L)				0	1	0	1	1	1	0	1	1	1	1	rd	0	1	1	0	1	0	0	0	d	d		x	x	
	AND.L #xx:32,@aa:16				0	1	0	1	1	1	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	a			x	x
	AND.L #xx:32,@aa:32				0	1	0	1	1	1	0	0	1	0	0	0	0	0	0	1	1	0	0	0	a	a		x	x	
	AND.L ERs,ERd				0	1	F	0	0	0	0	6	6	0	rs	0	rd													
	AND.L ERs,@ERd				0	1	0	1	0	0	1	0	0	0	0	rd	0	1	1	0	0	0	rs							
	AND.L ERs,@ERd+				0	1	0	1	0	0	1	1	0	0	0	rd	0	1	1	0	0	0	rs							
	AND.L ERs,@ERd-				0	1	0	1	0	0	1	1	0	1	0	rd	0	1	1	0	0	0	rs							
	AND.L ERs,@+ERd				0	1	0	1	0	0	1	1	0	0	1	rd	0	1	1	0	0	0	rs							
	AND.L ERs,@-ERd				0	1	0	1	0	0	1	1	0	1	0	rd	0	1	1	0	0	0	rs							
	AND.L ERs,@(d:2,ERd)				0	1	0	1	0	0	1	0	0	d	d	rd	0	1	1	0	0	0	rs							
	AND.L ERs,@(d:16,ERd)				0	1	0	1	0	0	1	1	1	0	0	rd	0	1	1	0	0	0	rs	d						
	AND.L ERs,@(d:32,ERd)				0	1	0	1	0	0	1	1	1	0	1	rd	0	1	1	0	0	0	rs	d	d					
	AND.L ERs,@(d:16,Rd.B)				0	1	0	1	0	0	1	1	1	0	1	rd	0	1	1	0	0	0	rs	d						
	AND.L ERs,@(d:16,Rd.W)				0	1	0	1	0	0	1	1	1	1	0	rd	0	1	1	0	0	0	rs	d						
	AND.L ERs,@(d:16,ERd.L)				0	1	0	1	0	0	1	1	1	1	0	rd	0	1	1	0	0	0	rs	d						
	AND.L ERs,@(d:32,Rd.B)				0	1	0	1	0	0	1	1	1	0	1	rd	0	1	1	0	0	0	rs	d	d					
	AND.L ERs,@(d:32,Rd.W)				0	1	0	1	0	0	1	1	1	1	0	rd	0	1	1	0	0	0	rs	d	d					
	AND.L ERs,@(d:32,ERd.L)				0	1	0	1	0	0	1	1	1	1	1	rd	0	1	1	0	0	0	rs	d	d					
	AND.L ERs,@aa:16				0	1	0	1	0	0	1	0	1	0	0	0	0	0	0	1	1	0	0	rs	a					
	AND.L ERs,@aa:32				0	1	0	1	0	0	1	0	1	0	0	1	0	0	0	0	1	1	0	0	rs	a	a			
	AND.L @ERs,ERd				0	1	0	1	0	1	0	0	0	0	0	rs	0	1	1	0	0	0	rd							
	AND.L @ERs+,ERd				0	1	0	1	0	1	0	1	0	0	0	rs	0	1	1	0	0	0	rd							
	AND.L @ERs-,ERd				0	1	0	1	0	1	0	1	0	1	0	rs	0	1	1	0	0	0	rd							
	AND.L @+ERs,ERd				0	1	0	1	0	1	0	1	0	1	0	rs	0	1	1	0	0	0	rd							
	AND.L @-ERs,ERd				0	1	0	1	0	1	0	1	0	1	0	rs	0	1	1	0	0	0	rd							
	AND.L @(d:2,ERs),ERd				0	1	0	1	0	1	0	0	0	d	d	rs	0	1	1	0	0	0	rd							
	AND.L @(d:16,ERs),ERd				0	1	0	1	0	0	1	1	1	0	0	rs	0	1	1	0	0	0	rd	d						
	AND.L @(d:32,ERs),ERd				0	1	0	1	0	0	1	1	1	0	1	rs	0	1	1	0	0	0	rd	d	d					
	AND.L @(d:16,Rs.B),ERd				0	1	0	1	0	0	1	1	1	0	1	rs	0	1	1	0	0	0	rd	d						
	AND.L @(d:16,Rs.W),ERd				0	1	0	1	0	0	1	1	1	1	0	rs	0	1	1	0	0	0	rd	d						
	AND.L @(d:16,ERs.L),ERd				0	1	0	1	0	0	1	1	1	1	0	rs	0	1	1	0	0	0	rd	d						



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension																			
		15	8	7	0	15	8	7	0		15	8	7	0																				
AND	AND.L @(d:32.Rs.B),ERd				0		1		0	1	0	1	0	1	1	0	1	1	rs	0	1	1	0	0	rd	d	d							
	AND.L @(d:32.Rs.W),ERd				0		1		0	1	0	1	0	1	1	1	0	1	rs	0	1	1	0	0	rd	d	d							
	AND.L @(d:32.ERs.L),ERd				0		1		0	1	0	1	0	1	1	1	1	1	rs	0	1	1	0	0	rd	d	d							
	AND.L @aa:16,ERd				0		1		0	1	0	1	0	0	1	0	0	0	0	0	0	1	1	0	0	rd	a							
	AND.L @aa:32,ERd				0		1		0	1	0	1	0	0	1	0	0	1	0	0	0	1	1	0	0	rd	a a							
	AND.L @ERs, @ERd	0		1		0		0	1	0	0	0	6	9	0	rs	1	1	0	0	0	0	0	0	0	rd	0	1	1	0	0	0	0	0
	AND.L @ERs, @ERd+	0		1		0		0	1	0	0	6	9	0	rs	1	1	0	0	1	0	0	0	0	rd	0	1	1	0	0	0	0	0	
	AND.L @ERs, @ERd-	0		1		0		0	1	0	0	6	9	0	rs	1	1	0	0	1	0	1	0	0	rd	0	1	1	0	0	0	0	0	
	AND.L @ERs, @+ERd	0		1		0		0	1	0	0	6	9	0	rs	1	1	0	0	1	0	0	1	0	rd	0	1	1	0	0	0	0	0	
	AND.L @ERs, @-ERd	0		1		0		0	1	0	0	6	9	0	rs	1	1	0	0	1	0	1	1	0	rd	0	1	1	0	0	0	0	0	
	AND.L @ERs, @(d:2,ERd)	0		1		0		0	1	0	0	6	9	0	rs	1	1	0	0	0	0	d	d	0	rd	0	1	1	0	0	0	0	0	
	AND.L @ERs, @(d:16,ERd)	0		1		0		0	1	0	0	6	9	0	rs	1	1	0	0	1	1	0	0	0	rd	0	1	1	0	0	0	0	0	
	AND.L @ERs, @(d:32,ERd)	0		1		0		0	1	0	0	6	9	0	rs	1	1	0	0	1	1	0	0	1	rd	0	1	1	0	0	0	0	0	
	AND.L @ERs, @(d:16,Rd.B)	0		1		0		0	1	0	0	6	9	0	rs	1	1	0	0	1	1	0	1	0	rd	0	1	1	0	0	0	0	0	
	AND.L @ERs, @(d:16,Rd.W)	0		1		0		0	1	0	0	6	9	0	rs	1	1	0	0	1	1	1	0	0	rd	0	1	1	0	0	0	0	0	
	AND.L @ERs, @(d:16,ERd.L)	0		1		0		0	1	0	0	6	9	0	rs	1	1	0	0	1	1	1	1	0	rd	0	1	1	0	0	0	0	0	
	AND.L @ERs, @(d:32,Rd.B)	0		1		0		0	1	0	0	6	9	0	rs	1	1	0	0	1	1	0	1	1	rd	0	1	1	0	0	0	0	0	
	AND.L @ERs, @(d:32,Rd.W)	0		1		0		0	1	0	0	6	9	0	rs	1	1	0	0	1	1	1	1	1	rd	0	1	1	0	0	0	0	0	
	AND.L @ERs, @(d:32,ERd.L)	0		1		0		0	1	0	0	6	9	0	rs	1	1	0	0	1	1	1	1	1	rd	0	1	1	0	0	0	0	0	
	AND.L @ERs, @aa:16	0		1		0		0	1	0	0	6	9	0	rs	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	
	AND.L @ERs, @aa:32	0		1		0		0	1	0	0	6	9	0	rs	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	
	AND.L @ERs+, @ERd	0		1		0		0	1	0	0	6	D	0	rs	1	1	0	0	0	0	0	0	0	rd	0	1	1	0	0	0	0	0	
	AND.L @ERs+, @ERd+	0		1		0		0	1	0	0	6	D	0	rs	1	1	0	0	1	0	0	0	0	rd	0	1	1	0	0	0	0	0	
	AND.L @ERs+, @ERd-	0		1		0		0	1	0	0	6	D	0	rs	1	1	0	0	1	0	1	0	0	rd	0	1	1	0	0	0	0	0	
	AND.L @ERs+, @+ERd	0		1		0		0	1	0	0	6	D	0	rs	1	1	0	0	1	0	0	1	0	rd	0	1	1	0	0	0	0	0	
	AND.L @ERs+, @-ERd	0		1		0		0	1	0	0	6	D	0	rs	1	1	0	0	1	0	1	1	0	rd	0	1	1	0	0	0	0	0	
	AND.L @ERs+, @(d:2,ERd)	0		1		0		0	1	0	0	6	D	0	rs	1	1	0	0	0	0	d	d	0	rd	0	1	1	0	0	0	0	0	
	AND.L @ERs+, @(d:16,ERd)	0		1		0		0	1	0	0	6	D	0	rs	1	1	0	0	1	1	0	0	0	rd	0	1	1	0	0	0	0	0	
	AND.L @ERs+, @(d:32,ERd)	0		1		0		0	1	0	0	6	D	0	rs	1	1	0	0	1	1	0	1	1	rd	0	1	1	0	0	0	0	0	
	AND.L @ERs+, @(d:16,Rd.B)	0		1		0		0	1	0	0	6	D	0	rs	1	1	0	0	1	1	0	1	0	rd	0	1	1	0	0	0	0	0	
AND.L @ERs+, @(d:16,Rd.W)	0		1		0		0	1	0	0	6	D	0	rs	1	1	0	0	1	1	1	0	0	rd	0	1	1	0	0	0	0	0		
AND.L @ERs+, @(d:16,ERd.L)	0		1		0		0	1	0	0	6	D	0	rs	1	1	0	0	1	1	1	1	0	rd	0	1	1	0	0	0	0	0		
AND.L @ERs+, @(d:32,Rd.B)	0		1		0		0	1	0	0	6	D	0	rs	1	1	0	0	1	1	0	1	1	rd	0	1	1	0	0	0	0	0		
AND.L @ERs+, @(d:32,Rd.W)	0		1		0		0	1	0	0	6	D	0	rs	1	1	0	0	1	1	1	1	1	rd	0	1	1	0	0	0	0	0		

Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension																		
		15	8	7	0	15	8		7	0	15	8	7	0																			
AND	AND.L @ERs+, @(d:32,ERd.L)	0	1	0	0	1	0	0	6	D	0	rs	1	1	0	0	1	1	1	1	rd	0	1	1	0	0	0	0	0	0	0	d	d
	AND.L @ERs+, @aa:16	0	1	0	0	1	0	0	6	D	0	rs	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	a	
	AND.L @ERs+, @aa:32	0	1	0	0	1	0	0	6	D	0	rs	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	a	a	
	AND.L @ERs-, @ERd	0	1	0	0	1	1	0	6	D	0	rs	1	1	0	0	0	0	0	0	rd	0	1	1	0	0	0	0	0	0			
	AND.L @ERs-, @ERd+	0	1	0	0	1	1	0	6	D	0	rs	1	1	0	0	1	0	0	0	rd	0	1	1	0	0	0	0	0	0			
	AND.L @ERs-, @ERd-	0	1	0	0	1	1	0	6	D	0	rs	1	1	0	0	1	0	1	0	rd	0	1	1	0	0	0	0	0	0			
	AND.L @ERs-, @+ERd	0	1	0	0	1	1	0	6	D	0	rs	1	1	0	0	1	0	0	1	rd	0	1	1	0	0	0	0	0	0			
	AND.L @ERs-, @-ERd	0	1	0	0	1	1	0	6	D	0	rs	1	1	0	0	1	0	1	0	rd	0	1	1	0	0	0	0	0	0			
	AND.L @ERs-, @(d:2,ERd)	0	1	0	0	1	1	0	6	D	0	rs	1	1	0	0	0	d	d	0	rd	0	1	1	0	0	0	0	0				
	AND.L @ERs-, @(d:16,ERd)	0	1	0	0	1	1	0	6	D	0	rs	1	1	0	0	1	1	0	0	rd	0	1	1	0	0	0	0	0	0	d		
	AND.L @ERs-, @(d:32,ERd)	0	1	0	0	1	1	0	6	D	0	rs	1	1	0	0	1	1	0	0	rd	0	1	1	0	0	0	0	0	0	d	d	
	AND.L @ERs-, @(d:16,Rd.B)	0	1	0	0	1	1	0	6	D	0	rs	1	1	0	0	1	1	0	1	rd	0	1	1	0	0	0	0	0	0	d		
	AND.L @ERs-, @(d:16,Rd.W)	0	1	0	0	1	1	0	6	D	0	rs	1	1	0	0	1	1	0	0	rd	0	1	1	0	0	0	0	0	0	d		
	AND.L @ERs-, @(d:16,ERd.L)	0	1	0	0	1	1	0	6	D	0	rs	1	1	0	0	1	1	1	0	rd	0	1	1	0	0	0	0	0	0	d		
	AND.L @ERs-, @(d:32,Rd.B)	0	1	0	0	1	1	0	6	D	0	rs	1	1	0	0	1	1	0	1	rd	0	1	1	0	0	0	0	0	0	d	d	
	AND.L @ERs-, @(d:32,Rd.W)	0	1	0	0	1	1	0	6	D	0	rs	1	1	0	0	1	1	0	1	rd	0	1	1	0	0	0	0	0	0	d	d	
	AND.L @ERs-, @(d:32,ERd.L)	0	1	0	0	1	1	0	6	D	0	rs	1	1	0	0	1	1	1	1	rd	0	1	1	0	0	0	0	0	0	d	d	
	AND.L @ERs-, @aa:16	0	1	0	0	1	1	0	6	D	0	rs	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	a		
	AND.L @ERs-, @aa:32	0	1	0	0	1	1	0	6	D	0	rs	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	a	a	
	AND.L @+ERs-, @ERd	0	1	0	0	1	0	1	6	D	0	rs	1	1	0	0	0	0	0	0	rd	0	1	1	0	0	0	0	0	0			
	AND.L @+ERs-, @ERd+	0	1	0	0	1	0	1	6	D	0	rs	1	1	0	0	1	0	0	0	rd	0	1	1	0	0	0	0	0	0			
	AND.L @+ERs-, @ERd-	0	1	0	0	1	0	1	6	D	0	rs	1	1	0	0	1	0	1	0	rd	0	1	1	0	0	0	0	0	0			
	AND.L @+ERs-, @+ERd	0	1	0	0	1	0	1	6	D	0	rs	1	1	0	0	1	0	0	1	rd	0	1	1	0	0	0	0	0	0			
	AND.L @+ERs-, @-ERd	0	1	0	0	1	0	1	6	D	0	rs	1	1	0	0	1	0	1	0	rd	0	1	1	0	0	0	0	0	0			
	AND.L @+ERs-, @(d:2,ERd)	0	1	0	0	1	0	1	6	D	0	rs	1	1	0	0	0	d	d	0	rd	0	1	1	0	0	0	0	0	0			
	AND.L @+ERs-, @(d:16,ERd)	0	1	0	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	0	0	rd	0	1	1	0	0	0	0	0	0	d		
	AND.L @+ERs-, @(d:32,ERd)	0	1	0	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	0	0	rd	0	1	1	0	0	0	0	0	0	d	d	
	AND.L @+ERs-, @(d:16,Rd.B)	0	1	0	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	0	1	rd	0	1	1	0	0	0	0	0	0	d		
	AND.L @+ERs-, @(d:16,Rd.W)	0	1	0	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	0	0	rd	0	1	1	0	0	0	0	0	0			
	AND.L @+ERs-, @(d:16,ERd.L)	0	1	0	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	1	0	rd	0	1	1	0	0	0	0	0	0	d		
	AND.L @+ERs-, @(d:32,Rd.B)	0	1	0	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	0	1	rd	0	1	1	0	0	0	0	0	0	d	d	
	AND.L @+ERs-, @(d:32,Rd.W)	0	1	0	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	0	1	rd	0	1	1	0	0	0	0	0	0	d	d	
AND.L @+ERs-, @(d:32,ERd.L)	0	1	0	0	1	0	1	6	D	0	rs	1	1	0	0	1	1	1	1	rd	0	1	1	0	0	0	0	0	0	d	d		
AND.L @+ERs-, @aa:16	0	1	0	0	1	0	1	6	D	0	rs	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	a			



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
AND	AND.L @+ERs, @aa:32	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0 0		0 1 0 0 0	1 0 0 0 0	0 0 1 1 0	0 0 0 0 0	a a
	AND.L @-ERs, @ERd	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		0 0 0 0 0	0 rd	0 1 1 1 0	0 0 0 0 0	
	AND.L @-ERs, @ERd+	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 0 0 0 0	0 rd	0 1 1 1 0	0 0 0 0 0	
	AND.L @-ERs, @ERd-	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 0 1 0 0	0 rd	0 1 1 1 0	0 0 0 0 0	
	AND.L @-ERs, @+ERd	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 0 0 1 0	0 rd	0 1 1 1 0	0 0 0 0 0	
	AND.L @-ERs, @-ERd	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 0 1 1 0	0 rd	0 1 1 1 0	0 0 0 0 0	
	AND.L @-ERs, @(d:2,ERd)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		0 0 d d	0 rd	0 1 1 1 0	0 0 0 0 0	
	AND.L @-ERs, @(d:16,ERd)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 1 0 0 0	0 rd	0 1 1 1 0	0 0 0 0 0	d
	AND.L @-ERs, @(d:32,ERd)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 1 0 0 1	rd	0 1 1 1 0	0 0 0 0 0	d d
	AND.L @-ERs, @(d:16,Rd,B)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 1 0 1 0	0 rd	0 1 1 1 0	0 0 0 0 0	d
	AND.L @-ERs, @(d:16,Rd,W)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 1 1 1 0	0 rd	0 1 1 1 0	0 0 0 0 0	d
	AND.L @-ERs, @(d:16,ERd,L)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 1 1 1 0	0 rd	0 1 1 1 0	0 0 0 0 0	d
	AND.L @-ERs, @(d:32,Rd,B)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 1 0 1 1	rd	0 1 1 1 0	0 0 0 0 0	d d
	AND.L @-ERs, @(d:32,Rd,W)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 1 1 1 1	rd	0 1 1 1 0	0 0 0 0 0	d d
	AND.L @-ERs, @(d:32,ERd,L)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 1 1 1 1	rd	0 1 1 1 0	0 0 0 0 0	d d
	AND.L @-ERs, @aa:16	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		0 1 0 0 0	0 0 0 0 0	0 0 1 1 0	0 0 0 0 0	a
	AND.L @-ERs, @aa:32	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		0 1 0 0 1	0 0 0 0 0	0 0 1 1 0	0 0 0 0 0	a a
	AND.L @(d:2,ERs), @ERd	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		0 0 0 0 0	0 rd	0 1 1 1 0	0 0 0 0 0	
	AND.L @(d:2,ERs), @ERd+	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		1 0 0 0 0	0 rd	0 1 1 1 0	0 0 0 0 0	
	AND.L @(d:2,ERs), @ERd-	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		1 0 1 1 0	0 rd	0 1 1 1 0	0 0 0 0 0	
	AND.L @(d:2,ERs), @+ERd	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		1 0 0 1 0	0 rd	0 1 1 1 0	0 0 0 0 0	
	AND.L @(d:2,ERs), @-ERd	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		1 0 1 1 1	0 rd	0 1 1 1 0	0 0 0 0 0	
	AND.L @(d:2,ERs), @(d:2,ERd)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		0 0 d d	0 rd	0 1 1 1 0	0 0 0 0 0	
	AND.L @(d:2,ERs), @(d:16,ERd)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		1 1 0 0 0	0 rd	0 1 1 1 0	0 0 0 0 0	d
	AND.L @(d:2,ERs), @(d:32,ERd)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		1 1 0 0 1	rd	0 1 1 1 0	0 0 0 0 0	d d
	AND.L @(d:2,ERs), @(d:16,Rd,B)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		1 1 0 1 0	0 rd	0 1 1 1 0	0 0 0 0 0	d
	AND.L @(d:2,ERs), @(d:16,Rd,W)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		1 1 1 1 0	0 rd	0 1 1 1 0	0 0 0 0 0	d
	AND.L @(d:2,ERs), @(d:16,ERd,L)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		1 1 1 1 1	0 rd	0 1 1 1 0	0 0 0 0 0	d
	AND.L @(d:2,ERs), @(d:32,Rd,B)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		1 1 0 1 1	rd	0 1 1 1 0	0 0 0 0 0	d d
	AND.L @(d:2,ERs), @(d:32,Rd,W)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		1 1 1 1 1	rd	0 1 1 1 0	0 0 0 0 0	d d
	AND.L @(d:2,ERs), @(d:32,ERd,L)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		1 1 1 1 1	rd	0 1 1 1 0	0 0 0 0 0	d d
	AND.L @(d:2,ERs), @aa:16	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		0 1 0 0 0	0 0 0 0 0	0 0 1 1 0	0 0 0 0 0	a
AND.L @(d:2,ERs), @aa:32	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		0 1 0 0 1	0 0 0 0 0	0 0 1 1 0	0 0 0 0 0	a a	
AND.L @(d:16,ERs), @ERd	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0 0	d	0 0 0 0 0	0 rd	0 1 1 1 0	0 0 0 0 0		

Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
AND	AND.L @(d:16,ERs),@ERd+	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:16,ERs),@ERd-	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:16,ERs),@+ERd	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:16,ERs),@-ERd	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:16,ERs),@(d:2,ERd)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:16,ERs),@(d:16,ERd)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:16,ERs),@(d:32,ERd)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:16,ERs),@(d:16,Rd.B)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:16,ERs),@(d:16,Rd.W)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:16,ERs),@(d:16,ERdL)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:16,ERs),@(d:32,Rd.B)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:16,ERs),@(d:32,Rd.W)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:16,ERs),@(d:32,ERdL)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:16,ERs),@aa:16	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 1 1 0	0 0 0 0	a	
	AND.L @(d:16,ERs),@aa:32	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 1 0	0 0 0 0	a a	
	AND.L @(d:32,ERs),@ERd	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:32,ERs),@ERd+	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:32,ERs),@ERd-	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:32,ERs),@+ERd	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:32,ERs),@-ERd	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:32,ERs),@(d:2,ERd)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:32,ERs),@(d:16,ERd)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:32,ERs),@(d:32,ERd)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:32,ERs),@(d:16,Rd.B)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:32,ERs),@(d:16,Rd.W)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:32,ERs),@(d:16,ERdL)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:32,ERs),@(d:32,Rd.B)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:32,ERs),@(d:32,Rd.W)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:32,ERs),@(d:32,ERdL)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:32,ERs),@aa:16	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0 0 0 0	0 1 1 0	0 0 0 0	a	
	AND.L @(d:32,ERs),@aa:32	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1 0 0 0	0 1 1 0	0 0 0 0	a a	
	AND.L @(d:16,Rs.B),@ERd	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 1 1 0	0 0 0 0	
AND.L @(d:16,Rs.B),@ERd+	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 1 1 0	0 0 0 0		
AND.L @(d:16,Rs.B),@ERd-	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 1 0	0 0 0 0		



Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
AND	AND.L @(d:16,Rs.B),@+ERd	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:16,Rs.B),@-ERd	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:16,Rs.B),@(d:2,ERd)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:16,Rs.B),@(d:16,ERd)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:16,Rs.B),@(d:32,ERd)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:16,Rs.B),@(d:16,Rd.B)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:16,Rs.B),@(d:16,Rd.W)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:16,Rs.B),@(d:16,ERd.L)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:16,Rs.B),@(d:32,Rd.B)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:16,Rs.B),@(d:32,Rd.W)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:16,Rs.B),@(d:32,ERd.L)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:16,Rs.B),@aa:16	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 1 1 0	0 0 0 0	a	
	AND.L @(d:16,Rs.B),@aa:32	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 1 0	0 0 0 0	a a	
	AND.L @(d:16,Rs.W),@ERd	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:16,Rs.W),@ERd+	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:16,Rs.W),@ERd-	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:16,Rs.W),@+ERd	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:16,Rs.W),@-ERd	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:16,Rs.W),@(d:2,ERd)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:16,Rs.W),@(d:16,ERd)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:16,Rs.W),@(d:32,ERd)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:16,Rs.W),@(d:16,Rd.B)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:16,Rs.W),@(d:16,Rd.W)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:16,Rs.W),@(d:16,ERd.L)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:16,Rs.W),@(d:32,Rd.B)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:16,Rs.W),@(d:32,Rd.W)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:16,Rs.W),@(d:32,ERd.L)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:16,Rs.W),@aa:16	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 1 1 0	0 0 0 0	a	
AND.L @(d:16,Rs.W),@aa:32	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 1 0	0 0 0 0	a a		
AND.L @(d:16,ERs.L),@ERd	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 1 1 0	0 0 0 0		
AND.L @(d:16,ERs.L),@ERd+	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 1 1 0	0 0 0 0		
AND.L @(d:16,ERs.L),@ERd-	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 1 0	0 0 0 0		
AND.L @(d:16,ERs.L),@+ERd	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 1 0	0 0 0 0		
AND.L @(d:16,ERs.L),@-ERd	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 1 1 0	0 0 0 0		

Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension			
		15	8	7	0	15	8	7	0		15	8	7	0				
AND	AND.L @(d:16,ERs.L),@(d:2,ERd)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 0	d d	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:16,ERs.L),@(d:16,ERd)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1	0 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:16,ERs.L),@(d:32,ERd)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1	0 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:16,ERs.L),@(d:16,Rd.B)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1	0 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:16,ERs.L),@(d:16,Rd.W)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1	1 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:16,ERs.L),@(d:16,ERd.L)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1	1 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:16,ERs.L),@(d:32,Rd.B)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1	0 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:16,ERs.L),@(d:32,Rd.W)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1	1 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:16,ERs.L),@(d:32,ERd.L)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1	1 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:16,ERs.L),@aa:16	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 1	0 0	0	0 0 0 0	0 1 1 0	0 0 0 0	a
	AND.L @(d:16,ERs.L),@aa:32	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 1	0 0	1	0 0 0 0	0 1 1 0	0 0 0 0	a a
	AND.L @(d:32,Rs.B),@ERd	7	8	1	rs	0 1 0 1	6	B	0 0	1 0	1 1 0 0	d d	0 0	0 0	0	rd	0 1 1 0	0 0 0 0
	AND.L @(d:32,Rs.B),@ERd+	7	8	1	rs	0 1 0 1	6	B	0 0	1 0	1 1 0 0	d d	1 0	0 0	0	rd	0 1 1 0	0 0 0 0
	AND.L @(d:32,Rs.B),@ERd-	7	8	1	rs	0 1 0 1	6	B	0 0	1 0	1 1 0 0	d d	1 0	1 0	0	rd	0 1 1 0	0 0 0 0
	AND.L @(d:32,Rs.B),@+ERd	7	8	1	rs	0 1 0 1	6	B	0 0	1 0	1 1 0 0	d d	1 0	0 1	0	rd	0 1 1 0	0 0 0 0
	AND.L @(d:32,Rs.B),@-ERd	7	8	1	rs	0 1 0 1	6	B	0 0	1 0	1 1 0 0	d d	1 0	1 1	0	rd	0 1 1 0	0 0 0 0
	AND.L @(d:32,Rs.B),@(d:2,ERd)	7	8	1	rs	0 1 0 1	6	B	0 0	1 0	1 1 0 0	d d	0 0	0 d	0	rd	0 1 1 0	0 0 0 0
	AND.L @(d:32,Rs.B),@(d:16,ERd)	7	8	1	rs	0 1 0 1	6	B	0 0	1 0	1 1 0 0	d d	1 1	0 0	0	rd	0 1 1 0	0 0 0 0
	AND.L @(d:32,Rs.B),@(d:32,ERd)	7	8	1	rs	0 1 0 1	6	B	0 0	1 0	1 1 0 0	d d	1 1	0 0	1	rd	0 1 1 0	0 0 0 0
	AND.L @(d:32,Rs.B),@(d:16,Rd.B)	7	8	1	rs	0 1 0 1	6	B	0 0	1 0	1 1 0 0	d d	1 1	0 1	0	rd	0 1 1 0	0 0 0 0
	AND.L @(d:32,Rs.B),@(d:16,Rd.W)	7	8	1	rs	0 1 0 1	6	B	0 0	1 0	1 1 0 0	d d	1 1	1 0	0	rd	0 1 1 0	0 0 0 0
	AND.L @(d:32,Rs.B),@(d:16,ERd.L)	7	8	1	rs	0 1 0 1	6	B	0 0	1 0	1 1 0 0	d d	1 1	1 1	0	rd	0 1 1 0	0 0 0 0
	AND.L @(d:32,Rs.B),@(d:32,Rd.B)	7	8	1	rs	0 1 0 1	6	B	0 0	1 0	1 1 0 0	d d	1 1	0 1	1	rd	0 1 1 0	0 0 0 0
	AND.L @(d:32,Rs.B),@(d:32,Rd.W)	7	8	1	rs	0 1 0 1	6	B	0 0	1 0	1 1 0 0	d d	1 1	1 0	1	rd	0 1 1 0	0 0 0 0
	AND.L @(d:32,Rs.B),@(d:32,ERd.L)	7	8	1	rs	0 1 0 1	6	B	0 0	1 0	1 1 0 0	d d	1 1	1 1	1	rd	0 1 1 0	0 0 0 0
	AND.L @(d:32,Rs.B),@aa:16	7	8	1	rs	0 1 0 1	6	B	0 0	1 0	1 1 0 0	d d	0 1	0 0	0	0 0 0 0	0 1 1 0	0 0 0 0
	AND.L @(d:32,Rs.B),@aa:32	7	8	1	rs	0 1 0 1	6	B	0 0	1 0	1 1 0 0	d d	0 1	0 0	1	0 0 0 0	0 1 1 0	0 0 0 0
	AND.L @(d:32,Rs.W),@ERd	7	8	1	rs	0 1 1 0	6	B	0 0	1 0	1 1 0 0	d d	0 0	0 0	0	rd	0 1 1 0	0 0 0 0
	AND.L @(d:32,Rs.W),@ERd+	7	8	1	rs	0 1 1 0	6	B	0 0	1 0	1 1 0 0	d d	1 0	0 0	0	rd	0 1 1 0	0 0 0 0
	AND.L @(d:32,Rs.W),@ERd-	7	8	1	rs	0 1 1 0	6	B	0 0	1 0	1 1 0 0	d d	1 0	1 0	0	rd	0 1 1 0	0 0 0 0
	AND.L @(d:32,Rs.W),@+ERd	7	8	1	rs	0 1 1 0	6	B	0 0	1 0	1 1 0 0	d d	1 0	0 1	0	rd	0 1 1 0	0 0 0 0
	AND.L @(d:32,Rs.W),@-ERd	7	8	1	rs	0 1 1 0	6	B	0 0	1 0	1 1 0 0	d d	1 0	1 1	0	rd	0 1 1 0	0 0 0 0
AND.L @(d:32,Rs.W),@(d:2,ERd)	7	8	1	rs	0 1 1 0	6	B	0 0	1 0	1 1 0 0	d d	0 0	0 d	0	rd	0 1 1 0	0 0 0 0	
AND.L @(d:32,Rs.W),@(d:16,ERd)	7	8	1	rs	0 1 1 0	6	B	0 0	1 0	1 1 0 0	d d	1 1	0 0	0	rd	0 1 1 0	0 0 0 0	



Instruction	Mnemonic	Opcode						EA Ex- tension	Opcode						EA Extension		
		15	8	7	0	15	8		7	0	15	8	7	0			
AND	AND.L @(d:32.Rs.W),@(d:32.ERd)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:32.Rs.W),@(d:16.Rd.B)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:32.Rs.W),@(d:16.Rd.W)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:32.Rs.W),@(d:16.ERd.L)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:32.Rs.W),@(d:32.Rd.B)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:32.Rs.W),@(d:32.Rd.W)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:32.Rs.W),@(d:32.ERd.L)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:32.Rs.W),@aa:16	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 1 1 0	0 0 0 0	a
	AND.L @(d:32.Rs.W),@aa:32	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 1 1 0	0 0 0 0	a a
	AND.L @(d:32.ERs.L),@ERd	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:32.ERs.L),@ERd+	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:32.ERs.L),@ERd-	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:32.ERs.L),@+ERd	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:32.ERs.L),@-ERd	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:32.ERs.L),@(d:2.ERd)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 1 1 0	0 0 0 0	
	AND.L @(d:32.ERs.L),@(d:16.ERd)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:32.ERs.L),@(d:32.ERd)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:32.ERs.L),@(d:16.Rd.B)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:32.ERs.L),@(d:16.Rd.W)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:32.ERs.L),@(d:16.ERd.L)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @(d:32.ERs.L),@(d:32.Rd.B)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:32.ERs.L),@(d:32.Rd.W)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:32.ERs.L),@(d:32.ERd.L)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @(d:32.ERs.L),@aa:16	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 1 1 0	0 0 0 0	a
	AND.L @(d:32.ERs.L),@aa:32	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 1 1 0	0 0 0 0	a a
	AND.L @aa:16,@ERd	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.L @aa:16,@ERd+	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.L @aa:16,@ERd-	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 1 0	0	rd	0 1 1 0	0 0 0 0	
	AND.L @aa:16,@+ERd	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 0 1	0	rd	0 1 1 0	0 0 0 0	
	AND.L @aa:16,@-ERd	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 1 1	0	rd	0 1 1 0	0 0 0 0	
AND.L @aa:16,@(d:2.ERd)	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 0 d d	0	rd	0 1 1 0	0 0 0 0		
AND.L @aa:16,@(d:16.ERd)	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 0	0	rd	0 1 1 0	0 0 0 0	d	
AND.L @aa:16,@(d:32.ERd)	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 0	1	rd	0 1 1 0	0 0 0 0	d d	
AND.L @aa:16,@(d:16.Rd.B)	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 1	0	rd	0 1 1 0	0 0 0 0	d	

Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
AND	AND.L @aa:16,@(d:16,Rd,W)	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @aa:16,@(d:16,ERd,L)	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @aa:16,@(d:32,Rd,B)	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @aa:16,@(d:32,Rd,W)	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @aa:16,@(d:32,ERd,L)	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 1	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @aa:16,@aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 1 0 0	0 0 0 0	0 1 1 0	0 0 0 0	a	
	AND.L @aa:16,@aa:32	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 1 0 0	1 0 0 0	0 1 1 0	0 0 0 0	a a	
	AND.L @aa:32,@ERd	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.L @aa:32,@ERd+	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 0 0	0	rd	0 1 1 0	0 0 0 0	
	AND.L @aa:32,@ERd-	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 1 0	0	rd	0 1 1 0	0 0 0 0	
	AND.L @aa:32,@+ERd	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 0 1	0	rd	0 1 1 0	0 0 0 0	
	AND.L @aa:32,@-ERd	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 0 1	0	rd	0 1 1 0	0 0 0 0	
	AND.L @aa:32,@ERd	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 1 1	0	rd	0 1 1 0	0 0 0 0	
	AND.L @aa:32,@(d:2,ERd)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 0 d d	0	rd	0 1 1 0	0 0 0 0	
	AND.L @aa:32,@(d:16,ERd)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @aa:32,@(d:32,ERd)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 0	1	rd	0 1 1 0	0 0 0 0	d d
	AND.L @aa:32,@(d:16,Rd,B)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @aa:32,@(d:16,Rd,W)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 0	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @aa:32,@(d:16,ERd,L)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 1	0	rd	0 1 1 0	0 0 0 0	d
	AND.L @aa:32,@(d:32,Rd,B)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 1	1	rd	0 1 1 0	0 0 0 0	d d
AND.L @aa:32,@(d:32,Rd,W)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 0	1	rd	0 1 1 0	0 0 0 0	d d	
AND.L @aa:32,@(d:32,ERd,L)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 1	1	rd	0 1 1 0	0 0 0 0	d d	
AND.L @aa:32,@aa:16	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 1 0 0	0 0 0 0	0 1 1 0	0 0 0 0	a		
AND.L @aa:32,@aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 1 0 0	1 0 0 0	0 1 1 0	0 0 0 0	a a		
ANDC	ANDC #xx:8,CCR										0	6	x x x x	x x x x		
	ANDC #xx:8,EXR					0	1	4	0 0 0 1		0	6	x x x x	x x x x		
BAND	BAND #xx:3,Rd										7	6	0 x x x	rd		
	BAND #xx:3,@ERd					7	C	0	rd	0 0 0 0		7	6	0 x x x	0 0 0 0	
	BAND #xx:3,@aa:8					7	E	a a a a	a a a a		7	6	0 x x x	0 0 0 0		
	BAND #xx:3,@aa:16					6	A	0 0 0 1	0 0 0 0	a	7	6	0 x x x	0 0 0 0		
	BAND #xx:3,@aa:32					6	A	0 0 1 1	0 0 0 0	a a	7	6	0 x x x	0 0 0 0		
Bcc	BRA d:8(BT d:8)										4	0 0 0 0	d d d d	d d d d		
	BRA d:16(BT d:16)										5	8	0 0 0 0	0 0 0 0	d	
	BRN d:8(BF d:8)										4	0 0 0 1	d d d d	d d d d		
	BRN d:16(BF d:16)										5	8	0 0 0 1	0 0 0 0	d	



Instruction	Mnemonic	Opcode			Opcode			EA Extension	Opcode			EA Extension									
		15	8	7	0	15	8		7	0	15		8	7	0						
Bcc	BHI d:8								4	0	0	1	0	d	d	d	d	d	d	0	
	BHI d:16								5		8	0	0	1	0	0	0	0	0	0	d
	BLS d:8								4	0	0	1	1	d	d	d	d	d	d	0	
	BLS d:16								5		8	0	0	1	1	0	0	0	0	0	d
	BCC d:8(BHS d:8)								4	0	1	0	0	d	d	d	d	d	d	0	
	BCC d:16(BHS d:16)								5		8	0	1	0	0	0	0	0	0	0	d
	BCS d:8(BLO d:8)								4	0	1	0	1	d	d	d	d	d	d	0	
	BCS d:16(BLO d:16)								5		8	0	1	0	1	0	0	0	0	0	d
	BNE d:8								4	0	1	1	0	d	d	d	d	d	d	0	
	BNE d:16								5		8	0	1	1	0	0	0	0	0	0	d
	BEQ d:8								4	0	1	1	1	d	d	d	d	d	d	0	
	BEQ d:16								5		8	0	1	1	1	0	0	0	0	0	d
	BVC d:8								4	1	0	0	0	d	d	d	d	d	d	0	
	BVC d:16								5		8	1	0	0	0	0	0	0	0	0	d
	BVS d:8								4	1	0	0	1	d	d	d	d	d	d	0	
	BVS d:16								5		8	1	0	0	1	0	0	0	0	0	d
	BPL d:8								4	1	0	1	0	d	d	d	d	d	d	0	
	BPL d:16								5		8	1	0	1	0	0	0	0	0	0	d
	BMI d:8								4	1	0	1	1	d	d	d	d	d	d	0	
	BMI d:16								5		8	1	0	1	1	0	0	0	0	0	d
	BGE d:8								4	1	1	0	0	d	d	d	d	d	d	0	
	BGE d:16								5		8	1	1	0	0	0	0	0	0	0	d
	BLT d:8								4	1	1	0	1	d	d	d	d	d	d	0	
	BLT d:16								5		8	1	1	0	1	0	0	0	0	0	d
	BGT d:8								4	1	1	1	0	d	d	d	d	d	d	0	
	BGT d:16								5		8	1	1	1	0	0	0	0	0	0	d
	BLE d:8								4	1	1	1	1	d	d	d	d	d	d	0	
	BLE d:16								5		8	1	1	1	1	0	0	0	0	0	d
BCLR	BCLR #xx:3,Rd								7	2	0	x	x	x	rd						
	BCLR #xx:3,@ERd				7	D	0	rd	0	0	0	0									
	BCLR #xx:3,@aa:8				7	F	a	a	a	a	a	a	a	a							
	BCLR #xx:3,@aa:16				6	A	0	0	0	1	1	0	0	0	a						
	BCLR #xx:3,@aa:32				6	A	0	0	1	1	1	0	0	0	a	a					
	BCLR Rn,Rd									6	2	m		rd							
	BCLR Rn,@ERd				7	D	0	rd	0	0	0	0									

Instruction	Mnemonic	Opcode			Opcode			EA Extension	Opcode			EA Extension	
		15	8	7	0	15	8		7	0	15		8
BCLR	BCLR Rn, @aa:8				7	F	a a a a a	a a a a a		6	2	m	0 0 0 0
	BCLR Rn, @aa:16				6	A	0 0 0 0 1	1 0 0 0 0	a	6	2	m	0 0 0 0
	BCLR Rn, @aa:32				6	A	0 0 0 1 1	1 0 0 0 0	a a	6	2	m	0 0 0 0
BCLR/EQ	BCLR/EQ #xx:3, @ERd				7	D	0 rd	0 0 0 0 0		7	2	0 x x x	0 1 1 1
	BCLR/EQ #xx:3, @aa:8				7	F	a a a a a	a a a a a		7	2	0 x x x	0 1 1 1
	BCLR/EQ #xx:3, @aa:16				6	A	0 0 0 0 1	1 0 0 0 0	a	7	2	0 x x x	0 1 1 1
	BCLR/EQ #xx:3, @aa:32				6	A	0 0 0 1 1	1 0 0 0 0	a a	7	2	0 x x x	0 1 1 1
	BCLR/EQ Rn, @ERd				7	D	0 rd	0 0 0 0 0		6	2	m	0 1 1 1
	BCLR/EQ Rn, @aa:8				7	F	a a a a a	a a a a a		6	2	m	0 1 1 1
	BCLR/EQ Rn, @aa:16				6	A	0 0 0 0 1	1 0 0 0 0	a	6	2	m	0 1 1 1
	BCLR/EQ Rn, @aa:32				6	A	0 0 0 1 1	1 0 0 0 0	a a	6	2	m	0 1 1 1
BCLR/NE	BCLR/NE #xx:3, @ERd				7	D	0 rd	0 0 0 0 0		7	2	0 x x x	0 1 1 0
	BCLR/NE #xx:3, @aa:8				7	F	a a a a a	a a a a a		7	2	0 x x x	0 1 1 0
	BCLR/NE #xx:3, @aa:16				6	A	0 0 0 0 1	1 0 0 0 0	a	7	2	0 x x x	0 1 1 0
	BCLR/NE #xx:3, @aa:32				6	A	0 0 0 1 1	1 0 0 0 0	a a	7	2	0 x x x	0 1 1 0
	BCLR/NE Rn, @ERd				7	D	0 rd	0 0 0 0 0		6	2	m	0 1 1 0
	BCLR/NE Rn, @aa:8				7	F	a a a a a	a a a a a		6	2	m	0 1 1 0
	BCLR/NE Rn, @aa:16				6	A	0 0 0 0 1	1 0 0 0 0	a	6	2	m	0 1 1 0
	BCLR/NE Rn, @aa:32				6	A	0 0 0 1 1	1 0 0 0 0	a a	6	2	m	0 1 1 0
BFLD	BFLD #xx:8, @ERs, Rd				7	C	0 rs	0 0 0 0 0		F	rd	x x x x	x x x x
	BFLD #xx:8, @aa:8, Rd				7	E	a a a a a	a a a a a		F	rd	x x x x	x x x x
	BFLD #xx:8, @aa:16, Rd				6	A	0 0 0 0 1	0 0 0 0 0	a	F	rd	x x x x	x x x x
	BFLD #xx:8, @aa:32, Rd				6	A	0 0 0 1 1	0 0 0 0 0	a a	F	rd	x x x x	x x x x
BFST	BFST Rs, #xx:8, @ERd				7	D	0 rd	0 0 0 0 0		F	rs	x x x x	x x x x
	BFST Rs, #xx:8, @aa:8				7	F	a a a a a	a a a a a		F	rs	x x x x	x x x x
	BFST Rs, #xx:8, @aa:16				6	A	0 0 0 0 1	1 0 0 0 0	a	F	rs	x x x x	x x x x
	BFST Rs, #xx:8, @aa:32				6	A	0 0 0 1 1	1 0 0 0 0	a a	F	rs	x x x x	x x x x
BIAND	BIAND #xx:3, Rd									7	6	1 x x x	rd
	BIAND #xx:3, @ERd				7	C	0 rd	0 0 0 0 0		7	6	1 x x x	0 0 0 0
	BIAND #xx:3, @aa:8				7	E	a a a a a	a a a a a		7	6	1 x x x	0 0 0 0
	BIAND #xx:3, @aa:16				6	A	0 0 0 0 1	0 0 0 0 0	a	7	6	1 x x x	0 0 0 0
	BIAND #xx:3, @aa:32				6	A	0 0 0 1 1	0 0 0 0 0	a a	7	6	1 x x x	0 0 0 0
BILD	BILD #xx:3, Rd									7	7	1 x x x	rd
	BILD #xx:3, @ERd				7	C	0 rd	0 0 0 0 0		7	7	1 x x x	0 0 0 0
	BILD #xx:3, @aa:8				7	E	a a a a a	a a a a a		7	7	1 x x x	0 0 0 0
	BILD #xx:3, @aa:16				6	A	0 0 0 0 1	0 0 0 0 0	a	7	7	1 x x x	0 0 0 0
	BILD #xx:3, @aa:32				6	A	0 0 0 1 1	0 0 0 0 0	a a	7	7	1 x x x	0 0 0 0



Instruction	Mnemonic	Opcode			Opcode			EA Extension	Opcode			EA Extension					
		15	8	7	0	15	8		7	0	15		8	7	0		
BIOR	BIOR #xx:3,Rd									7	4	1	x	x	x	rd	
	BIOR #xx:3,@ERd				7		C	0	rd	0	0	0	0	0			
	BIOR #xx:3,@aa:8				7		E	a	a	a	a	a	a	a	a		
	BIOR #xx:3,@aa:16				6		A	0	0	0	1	0	0	0	0	a	
	BIOR #xx:3,@aa:32				6		A	0	0	1	1	0	0	0	0	a	a
BIST	BIST #xx:3,Rd										6	7	1	x	x	x	rd
	BIST #xx:3,@ERd				7		D	0	rd	0	0	0	0	0			
	BIST #xx:3,@aa:8				7		F	a	a	a	a	a	a	a	a		
	BIST #xx:3,@aa:16				6		A	0	0	0	1	1	0	0	0	a	
	BIST #xx:3,@aa:32				6		A	0	0	1	1	1	0	0	0	a	a
BISTZ	BISTZ #xx:3,@ERd				7		D	0	rd	0	0	0	0	0			
	BISTZ #xx:3,@aa:8				7		F	a	a	a	a	a	a	a	a		
	BISTZ #xx:3,@aa:16				6		A	0	0	0	1	1	0	0	0	a	
	BISTZ #xx:3,@aa:32				6		A	0	0	1	1	1	0	0	0	a	a
BIXOR	BIXOR #xx:3,Rd										7	5	1	x	x	x	rd
	BIXOR #xx:3,@ERd				7		C	0	rd	0	0	0	0	0			
	BIXOR #xx:3,@aa:8				7		E	a	a	a	a	a	a	a	a		
	BIXOR #xx:3,@aa:16				6		A	0	0	0	1	0	0	0	0	a	
	BIXOR #xx:3,@aa:32				6		A	0	0	1	1	0	0	0	0	a	a
BLD	BLD #xx:3,Rd										7	7	0	x	x	x	rd
	BLD #xx:3,@ERd				7		C	0	rd	0	0	0	0	0			
	BLD #xx:3,@aa:8				7		E	a	a	a	a	a	a	a	a		
	BLD #xx:3,@aa:16				6		A	0	0	0	1	0	0	0	0	a	
	BLD #xx:3,@aa:32				6		A	0	0	1	1	0	0	0	0	a	a
BNOT	BNOT #xx:3,Rd										7	1	0	x	x	x	rd
	BNOT #xx:3,@ERd				7		D	0	rd	0	0	0	0	0			
	BNOT #xx:3,@aa:8				7		F	a	a	a	a	a	a	a	a		
	BNOT #xx:3,@aa:16				6		A	0	0	0	1	1	0	0	0	a	
	BNOT #xx:3,@aa:32				6		A	0	0	1	1	1	0	0	0	a	a
	BNOT Rn,Rd											6	1	r	r	r	rd
	BNOT Rn,@ERd				7		D	0	rd	0	0	0	0	0			
	BNOT Rn,@aa:8				7		F	a	a	a	a	a	a	a	a		
	BNOT Rn,@aa:16				6		A	0	0	0	1	1	0	0	0	a	
BNOT Rn,@aa:32				6		A	0	0	1	1	1	0	0	0	a	a	

Instruction	Mnemonic	Opcode				EA Ex- tension	Opcode				EA										
		15	8	7	0		15	8	7	0		1	8	7	0						
BOR	BOR #xx:3,Rd										7	4	0	x	x	x	rd				
	BOR #xx:3,@ERd					7	C	0	rd	0	0	0	0								
	BOR #xx:3,@aa:8					7	E	a	a	a	a	a	a	a							
	BOR #xx:3,@aa:16					6	A	0	0	0	1	0	0	0	a						
	BOR #xx:3,@aa:32					6	A	0	0	1	1	0	0	0	0	a	a				
BRA	BRA Rn,B											5	9	0		m	0	1	0	1	
	BRA Rn,W											5	9	0		m	0	1	1	0	
	BRA ERn,L											5	9	0		m	0	1	1	1	
BRA/BC	BRA/BC #xx:3,@ERs,d:8					7	C	0	rs	0	0	0	0								
BRA/BS	BRA/BC #xx:3,@aa:8,d:8					7	E	a	a	a	a	a	a	a							
	BRA/BC #xx:3,@aa:16,d:8					6	A	0	0	0	1	0	0	0	0	a					
	BRA/BC #xx:3,@aa:32,d:8					6	A	0	0	1	1	0	0	0	0	a	a				
	BRA/BS #xx:3,@ERs,d:8					7	C	0	rs	0	0	0	0								
	BRA/BS #xx:3,@aa:8,d:8					7	E	a	a	a	a	a	a	a							
	BRA/BS #xx:3,@aa:16,d:8					6	A	0	0	0	1	0	0	0	0	a					
	BRA/BS #xx:3,@aa:32,d:8					6	A	0	0	1	1	0	0	0	0	a	a				
	BRA/BC #xx:3,@ERs,d:16					7	C	0	rs	0	0	0	0								
	BRA/BC #xx:3,@aa:8,d:16					7	E	a	a	a	a	a	a	a							
	BRA/BC #xx:3,@aa:16,d:16					6	A	0	0	0	1	0	0	0	0	a					
	BRA/BC #xx:3,@aa:32,d:16					6	A	0	0	1	1	0	0	0	0	a	a				
	BRA/BS #xx:3,@ERs,d:16					7	C	0	rs	0	0	0	0								
	BRA/BS #xx:3,@aa:8,d:16					7	E	a	a	a	a	a	a	a							
	BRA/BS #xx:3,@aa:16,d:16					6	A	0	0	0	1	0	0	0	0	a					
	BRA/BS #xx:3,@aa:32,d:16					6	A	0	0	1	1	0	0	0	0	a	a				
	BRA/S	BRA/S d:8											4	0	0	0	0	d	d	d	d
BSET	BSET #xx:3,Rd											7	0	0	x	x	rd				
	BSET #xx:3,@ERd					7	D	0	rd	0	0	0	0								
	BSET #xx:3,@aa:8					7	F	a	a	a	a	a	a	a							
	BSET #xx:3,@aa:16					6	A	0	0	0	1	1	0	0	0	a					
	BSET #xx:3,@aa:32					6	A	0	0	1	1	1	0	0	0	a	a				



Instruction	Mnemonic	Opcode			Opcode			EA Extension	Opcode			EA Extension			
		15	8	7	0	15	8		7	0	15		8	7	0
BSET	BSET Rn,Rd									6	0	m	rd		
	BSET Rn,@ERd				7	D	0	rd	0 0 0 0		6	0	m	0 0 0 0	
	BSET Rn,@aa:8				7	F	a a a a	a a a a			6	0	m	0 0 0 0	
	BSET Rn,@aa:16				6	A	0 0 0 1	1 0 0 0	a		6	0	m	0 0 0 0	
BSET Rn,@aa:32				6	A	0 0 1 1	1 0 0 0	a a		6	0	m	0 0 0 0		
BSET/EQ	BSET/EQ #xx:3,@ERd				7	D	0	rd	0 0 0 0		7	0	0 x x x	0 1 1 1	
	BSET/EQ #xx:3,@aa:8				7	F	a a a a	a a a a			7	0	0 x x x	0 1 1 1	
	BSET/EQ #xx:3,@aa:16				6	A	0 0 0 1	1 0 0 0	a		7	0	0 x x x	0 1 1 1	
	BSET/EQ #xx:3,@aa:32				6	A	0 0 1 1	1 0 0 0	a a		7	0	0 x x x	0 1 1 1	
	BSET/EQ Rn,@ERd				7	D	0	rd	0 0 0 0		6	0	m	0 1 1 1	
	BSET/EQ Rn,@aa:8				7	F	a a a a	a a a a			6	0	m	0 1 1 1	
	BSET/EQ Rn,@aa:16				6	A	0 0 0 1	1 0 0 0	a		6	0	m	0 1 1 1	
	BSET/EQ Rn,@aa:32				6	A	0 0 1 1	1 0 0 0	a a		6	0	m	0 1 1 1	
BSET/NE	BSET/NE #xx:3,@ERd				7	D	0	rd	0 0 0 0		7	0	0 x x x	0 1 1 0	
	BSET/NE #xx:3,@aa:8				7	F	a a a a	a a a a			7	0	0 x x x	0 1 1 0	
	BSET/NE #xx:3,@aa:16				6	A	0 0 0 1	1 0 0 0	a		7	0	0 x x x	0 1 1 0	
	BSET/NE #xx:3,@aa:32				6	A	0 0 1 1	1 0 0 0	a a		7	0	0 x x x	0 1 1 0	
	BSET/NE Rn,@ERd				7	D	0	rd	0 0 0 0		6	0	m	0 1 1 0	
	BSET/NE Rn,@aa:8				7	F	a a a a	a a a a			6	0	m	0 1 1 0	
	BSET/NE Rn,@aa:16				6	A	0 0 0 1	1 0 0 0	a		6	0	m	0 1 1 0	
	BSET/NE Rn,@aa:32				6	A	0 0 1 1	1 0 0 0	a a		6	0	m	0 1 1 0	
BSR	BSR d:8									5	5	d d d d	d d d d		
	BSR d:16									5	C	0 0 0 0	0 0 0 0	d	
	BSR Rn,B									5	D	0	m	0 1 0 1	
	BSR Rn,W									5	D	0	m	0 1 1 0	
	BSR ERn,L									5	D	0	m	0 1 1 1	
BSR/BC	BSR/BC #xx:3,@ERs.d:16				7	C	0	rs	0 0 0 0		5	C	0 x x x	0 0 0 0	d
	BSR/BC #xx:3,@aa:8.d:16				7	E	a a a a	a a a a			5	C	0 x x x	0 0 0 0	d
	BSR/BC #xx:3,@aa:16.d:16				6	A	0 0 0 1	0 0 0 0	a		5	C	0 x x x	0 0 0 0	d
	BSR/BC #xx:3,@aa:32.d:16				6	A	0 0 1 1	0 0 0 0	a a		5	C	0 x x x	0 0 0 0	d
BSR/BS	BSR/BS #xx:3,@ERs.d:16				7	C	0	rs	0 0 0 0		5	C	1 x x x	0 0 0 0	d
	BSR/BS #xx:3,@aa:8.d:16				7	E	a a a a	a a a a			5	C	1 x x x	0 0 0 0	d
	BSR/BS #xx:3,@aa:16.d:16				6	A	0 0 0 1	0 0 0 0	a		5	C	1 x x x	0 0 0 0	d
	BSR/BS #xx:3,@aa:32.d:16				6	A	0 0 1 1	0 0 0 0	a a		5	C	1 x x x	0 0 0 0	d

Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension																
		15	8	7	0	15	8	7	0		15	8	7	0																	
BST	BST #xx:3,Rd										6	7	0	x	x	x	rd														
	BST #xx:3,@ERd					7	D	0	rd	0	0	0	0		6	7	0	x	x	x	0	0	0	0							
	BST #xx:3,@aa:8					7	F	a	a	a	a	a	a	a		6	7	0	x	x	x	0	0	0	0						
	BST #xx:3,@aa:16					6	A	0	0	0	1	1	0	0	0	a		6	7	0	x	x	x	0	0	0	0				
BST #xx:3,@aa:32					6	A	0	0	0	1	1	1	0	0	0	a	a	6	7	0	x	x	x	0	0	0	0				
BSTZ	BSTZ #xx:3,@ERd					7	D	0	rd	0	0	0	0		6	7	0	x	x	x	0	1	1	1							
	BSTZ #xx:3,@aa:8					7	F	a	a	a	a	a	a	a		6	7	0	x	x	x	0	1	1	1						
	BSTZ #xx:3,@aa:16					6	A	0	0	0	1	1	1	0	0	0	a		6	7	0	x	x	x	0	1	1	1			
	BSTZ #xx:3,@aa:32					6	A	0	0	1	1	1	1	0	0	0	a	a	6	7	0	x	x	x	0	1	1	1			
BTST	BTST #xx:3,Rd														7	3	0	x	x	x	rd										
	BTST #xx:3,@ERd					7	C	0	rd	0	0	0	0		7	3	0	x	x	x	0	0	0	0							
	BTST #xx:3,@aa:8					7	E	a	a	a	a	a	a	a		7	3	0	x	x	x	0	0	0	0						
	BTST #xx:3,@aa:16					6	A	0	0	0	1	1	0	0	0	0	a		7	3	0	x	x	x	0	0	0	0			
	BTST #xx:3,@aa:32					6	A	0	0	1	1	1	0	0	0	0	a	a	7	3	0	x	x	x	0	0	0	0			
	BTST Rn,Rd														6	3						rd									
	BTST Rn,@ERd					7	C	0	rd	0	0	0	0		6	3					m	0	0	0	0						
	BTST Rn,@aa:8					7	E	a	a	a	a	a	a	a		6	3				m	0	0	0	0						
	BTST Rn,@aa:16					6	A	0	0	0	1	1	0	0	0	0	a		6	3				m	0	0	0	0			
BTST Rn,@aa:32					6	A	0	0	1	1	1	0	0	0	0	a	a	6	3				m	0	0	0	0				
BXOR	BXOR #xx:3,Rd														7	5	0	x	x	x	rd										
	BXOR #xx:3,@ERd					7	C	0	rd	0	0	0	0		7	5	0	x	x	x	0	0	0	0							
	BXOR #xx:3,@aa:8					7	E	a	a	a	a	a	a	a		7	5	0	x	x	x	0	0	0	0						
	BXOR #xx:3,@aa:16					6	A	0	0	0	1	1	0	0	0	0	a		7	5	0	x	x	x	0	0	0	0			
	BXOR #xx:3,@aa:32					6	A	0	0	1	1	1	0	0	0	0	a	a	7	5	0	x	x	x	0	0	0	0			
CLRMAC	CLRMAC													0	1					A	0	0	0	0							
CMP	CMP.B #xx:8,Rd														A		rd				x	x	x	x	x	x	x				
	CMP.B #xx:8,@ERd					7	D	0	rd	0	0	0	0		A	0	0	0	0		x	x	x	x	x	x	x				
	CMP.B #xx:8,@ERd+	0	1		7	0	1	0	0	6	C	0	rd	1	0	0	0		A	0	0	0	0	x	x	x	x	x	x		
	CMP.B #xx:8,@ERd-	0	1		7	0	1	1	0	6	C	0	rd	1	0	0	0		A	0	0	0	0	x	x	x	x	x	x		
	CMP.B #xx:8,@+ERd	0	1		7	0	1	0	1	6	C	0	rd	1	0	0	0		A	0	0	0	0	x	x	x	x	x	x		
	CMP.B #xx:8,@-ERd	0	1		7	0	1	1	1	6	C	0	rd	1	0	0	0		A	0	0	0	0	x	x	x	x	x	x		
	CMP.B #xx:8,@(d:2,ERd)	0	1		7	0	1	d	d	6	8	0	rd	1	0	0	0		A	0	0	0	0	x	x	x	x	x	x		
	CMP.B #xx:8,@(rd:16,ERd)	0	1		7	0	1	0	0	6	E	0	rd	1	0	0	0	d		A	0	0	0	0	x	x	x	x	x	x	
	CMP.B #xx:8,@(rd:32,ERd)	7	8	0	rd	0	1	0	0	6	A	0	0	1	0	1	0	0	d	d		A	0	0	0	0	x	x	x	x	x



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
CMP	CMP.B #xx:8,@(d:16,Rd,B)	0	1	7	0 1 0 1	6	E	0	rd	1 0 0 0	d	A	0 0 0 0	x x x x	x x x x	
	CMP.B #xx:8,@(d:16,Rd,W)	0	1	7	0 1 1 0	6	E	0	rd	1 0 0 0	d	A	0 0 0 0	x x x x	x x x x	
	CMP.B #xx:8,@(d:16,ERd,L)	0	1	7	0 1 1 1	6	E	0	rd	1 0 0 0	d	A	0 0 0 0	x x x x	x x x x	
	CMP.B #xx:8,@(d:32,Rd,B)	7	8	0	rd	0 1 0 1	6	A	0 0 1 0	1 0 0 0	d d	A	0 0 0 0	x x x x	x x x x	
	CMP.B #xx:8,@(d:32,Rd,W)	7	8	0	rd	0 1 1 0	6	A	0 0 1 0	1 0 0 0	d d	A	0 0 0 0	x x x x	x x x x	
	CMP.B #xx:8,@(d:32,ERd,L)	7	8	0	rd	0 1 1 1	6	A	0 0 1 0	1 0 0 0	d d	A	0 0 0 0	x x x x	x x x x	
	CMP.B #xx:8,@aa:8					7	F	a a a a	a a a a			A	0 0 0 0	x x x x	x x x x	
	CMP.B #xx:8,@aa:16					6	A	0 0 0 1	1 0 0 0	a		A	0 0 0 0	x x x x	x x x x	
	CMP.B #xx:8,@aa:32					6	A	0 0 1 1	1 0 0 0	a a		A	0 0 0 0	x x x x	x x x x	
	CMP.B Rs,Rd												1	C	rs	rd
	CMP.B Rs,@ERd					7	D	0	rd	0 0 0 0		1	C	rs	0 0 0 0	
	CMP.B Rs,@ERd+					0	1	7	1 0 0 1		1 0 0 0	0	rd	0 0 1 0	rs	
	CMP.B Rs,@ERd-					0	1	7	1 0 0 1		1 0 1 0	0	rd	0 0 1 0	rs	
	CMP.B Rs,@+ERd					0	1	7	1 0 0 1		1 0 0 1	0	rd	0 0 1 0	rs	
	CMP.B Rs,@-ERd					0	1	7	1 0 0 1		1 0 1 1	0	rd	0 0 1 0	rs	
	CMP.B Rs,@(d:2,ERd)					0	1	7	1 0 0 1		0 0 d d	0	rd	0 0 1 0	rs	
	CMP.B Rs,@(d:16,ERd)					0	1	7	1 0 0 1		1 1 0 0	0	rd	0 0 1 0	rs	d
	CMP.B Rs,@(d:32,ERd)					0	1	7	1 0 0 1		1 1 0 0	1	rd	0 0 1 0	rs	d d
	CMP.B Rs,@(d:16,Rd,B)					0	1	7	1 0 0 1		1 1 0 1	0	rd	0 0 1 0	rs	d
	CMP.B Rs,@(d:16,Rd,W)					0	1	7	1 0 0 1		1 1 1 0	0	rd	0 0 1 0	rs	d
	CMP.B Rs,@(d:16,ERd,L)					0	1	7	1 0 0 1		1 1 1 1	0	rd	0 0 1 0	rs	d
	CMP.B Rs,@(d:32,Rd,B)					0	1	7	1 0 0 1		1 1 0 1	1	rd	0 0 1 0	rs	d d
	CMP.B Rs,@(d:32,Rd,W)					0	1	7	1 0 0 1		1 1 1 0	1	rd	0 0 1 0	rs	d d
	CMP.B Rs,@(d:32,ERd,L)					0	1	7	1 0 0 1		1 1 1 1	1	rd	0 0 1 0	rs	d d
	CMP.B Rs,@aa:8					7	F	a a a a	a a a a		1	C	rs	0 0 0 0		
	CMP.B Rs,@aa:16					6	A	0 0 0 1	1 0 0 0	a		1	C	rs	0 0 0 0	
	CMP.B Rs,@aa:32					6	A	0 0 1 1	1 0 0 0	a a		1	C	rs	0 0 0 0	
	CMP.B @ERs,Rd					7	C	0	rs	0 0 0 0		1	C	0 0 0 0	rd	
	CMP.B @ERs+,Rd					0	1	7	1 0 1 0		1 0 0 0	0	rs	0 0 1 0	rd	
	CMP.B @ERs-,Rd					0	1	7	1 0 1 0		1 0 1 0	0	rs	0 0 1 0	rd	
	CMP.B @+ERs,Rd					0	1	7	1 0 1 0		1 0 0 1	0	rs	0 0 1 0	rd	
	CMP.B @-ERs,Rd					0	1	7	1 0 1 0		1 0 1 1	0	rs	0 0 1 0	rd	
CMP.B @(d:2,ERs),Rd					0	1	7	1 0 1 0		0 0 d d	0	rs	0 0 1 0	rd		
CMP.B @(d:16,ERs),Rd					0	1	7	1 0 1 0		1 1 0 0	0	rs	0 0 1 0	rd		

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension																									
		15	8	7	0		15	8	7	0																										
CMP	CMP.B @(d:32,ERs),Rd				0			1		7		1	0	0	1	0			1	1	0	0	1	0			1	rs	0	0	1	0		rd	d	d
	CMP.B @(d:16,Rs.B),Rd				0			1		7		1	0	0	1	0			1	1	0	1	0	rs	0	0	1	0		rd	d					
	CMP.B @(d:16,Rs.W),Rd				0			1		7		1	0	0	1	0			1	1	1	0	0	rs	0	0	1	0		rd	d					
	CMP.B @(d:16,ERs.L),Rd				0			1		7		1	0	0	1	0			1	1	1	1	0	rs	0	0	1	0		rd	d					
	CMP.B @(d:32,Rs.B),Rd				0			1		7		1	0	0	1	0			1	1	0	1	1	rs	0	0	1	0		rd	d	d				
	CMP.B @(d:32,Rs.W),Rd				0			1		7		1	0	0	1	0			1	1	1	0	1	rs	0	0	1	0		rd	d	d				
	CMP.B @(d:32,ERs.L),Rd				0			1		7		1	0	0	1	0			1	1	1	1	1	rs	0	0	1	0		rd	d	d				
	CMP.B @aa:8,Rd				7			E		a	a	a	a	a	a	a	a			1				C	0	0	0	0		rd						
	CMP.B @aa:16,Rd				6			A		0	0	0	0	1	0	0	0	0	a			1		C	0	0	0	0		rd						
	CMP.B @aa:32,Rd				6			A		0	0	0	1	1	0	0	0	0	a	a			1		C	0	0	0	0		rd					
	CMP.B @ERs,@ERd				7			C		0	rs	0	1	0	1	0	1			0	0	0	0	0	rd	0	0	1	0	0	0	0	0			
	CMP.B @ERs,@ERd+				7			C		0	rs	0	1	0	1	1			1	0	0	0	0	rd	0	0	1	0	0	0	0	0				
	CMP.B @ERs,@ERd-				7			C		0	rs	0	1	0	1	1			1	0	1	0	0	rd	0	0	1	0	0	0	0	0				
	CMP.B @ERs,@+ERd				7			C		0	rs	0	1	0	1	1			1	0	0	1	0	rd	0	0	1	0	0	0	0	0				
	CMP.B @ERs,@-ERd				7			C		0	rs	0	1	0	1	1			1	0	1	1	0	rd	0	0	1	0	0	0	0	0				
	CMP.B @ERs,@(d:2,ERd)				7			C		0	rs	0	1	0	1	1			0	0	d	d	0	rd	0	0	1	0	0	0	0	0				
	CMP.B @ERs,@(d:16,ERd)				7			C		0	rs	0	1	0	1	1			1	1	0	0	0	rd	0	0	1	0	0	0	0	0	d			
	CMP.B @ERs,@(d:32,ERd)				7			C		0	rs	0	1	0	1	1			1	1	0	0	1	rd	0	0	1	0	0	0	0	0	d	d		
	CMP.B @ERs,@(d:16,Rd.B)				7			C		0	rs	0	1	0	1	1			1	1	0	1	0	rd	0	0	1	0	0	0	0	0	d			
	CMP.B @ERs,@(d:16,Rd.W)				7			C		0	rs	0	1	0	1	1			1	1	1	0	0	rd	0	0	1	0	0	0	0	0	d			
	CMP.B @ERs,@(d:16,ERd.L)				7			C		0	rs	0	1	0	1	1			1	1	1	1	0	rd	0	0	1	0	0	0	0	0	d			
	CMP.B @ERs,@(d:32,Rd.B)				7			C		0	rs	0	1	0	1	1			1	1	0	1	1	rd	0	0	1	0	0	0	0	0	d	d		
	CMP.B @ERs,@(d:32,Rd.W)				7			C		0	rs	0	1	0	1	1			1	1	1	0	1	rd	0	0	1	0	0	0	0	0	d	d		
	CMP.B @ERs,@(d:32,ERd.L)				7			C		0	rs	0	1	0	1	1			1	1	1	1	1	rd	0	0	1	0	0	0	0	0	d	d		
	CMP.B @ERs,@aa:16				7			C		0	rs	0	1	0	1	1			0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	a		
	CMP.B @ERs,@aa:32				7			C		0	rs	0	1	0	1	1			0	1	0	0	1	0	0	0	0	0	0	0	0	0	a	a		
	CMP.B @ERs+,@ERd		0	1	7	0	1	0	0	6	C	0	rs	1	1	0	0			0	0	0	0	0	rd	0	0	1	0	0	0	0	0			
	CMP.B @ERs+,@ERd+		0	1	7	0	1	0	0	6	C	0	rs	1	1	0	0			1	0	0	0	0	rd	0	0	1	0	0	0	0	0			
	CMP.B @ERs+,@ERd-		0	1	7	0	1	0	0	6	C	0	rs	1	1	0	0			1	0	1	0	0	rd	0	0	1	0	0	0	0	0			
	CMP.B @ERs+,@+ERd		0	1	7	0	1	0	0	6	C	0	rs	1	1	0	0			1	0	0	1	0	rd	0	0	1	0	0	0	0	0			
	CMP.B @ERs+,@-ERd		0	1	7	0	1	0	0	6	C	0	rs	1	1	0	0			1	0	1	1	0	rd	0	0	1	0	0	0	0	0			
	CMP.B @ERs+,@(d:2,ERd)		0	1	7	0	1	0	0	6	C	0	rs	1	1	0	0			0	0	d	d	0	rd	0	0	1	0	0	0	0	0			
CMP.B @ERs+,@(d:16,ERd)		0	1	7	0	1	0	0	6	C	0	rs	1	1	0	0			1	1	0	0	0	rd	0	0	1	0	0	0	0	0	d			
CMP.B @ERs+,@(d:32,ERd)		0	1	7	0	1	0	0	6	C	0	rs	1	1	0	0			1	1	0	0	1	rd	0	0	1	0	0	0	0	0	d	d		



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension			
		15	8	7	0	15	8	7	0		15	8	7	0				
CMP	CMP.B @ERs+,@(d:16,Rd,B)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 1 0 1	0	rd	0 0 1 0	0 0 0 0	0	d
	CMP.B @ERs+,@(d:16,Rd,W)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 1 1 0	0	rd	0 0 1 0	0 0 0 0	0	d
	CMP.B @ERs+,@(d:16,ERd,L)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 1 1 1	0	rd	0 0 1 0	0 0 0 0	0	d
	CMP.B @ERs+,@(d:32,Rd,B)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 1 0 1	1	rd	0 0 1 0	0 0 0 0	0	d d
	CMP.B @ERs+,@(d:32,Rd,W)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 1 1 0	1	rd	0 0 1 0	0 0 0 0	0	d d
	CMP.B @ERs+,@(d:32,ERd,L)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 1 1 1	1	rd	0 0 1 0	0 0 0 0	0	d d
	CMP.B @ERs+,@aa:16	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		0 1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	0	a
	CMP.B @ERs+,@aa:32	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		0 1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	0	a a
	CMP.B @ERs-,@ERd	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		0 0 0 0	0	rd	0 0 1 0	0 0 0 0		
	CMP.B @ERs-,@ERd+	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 0 0 0	0	rd	0 0 1 0	0 0 0 0		
	CMP.B @ERs-,@ERd-	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 0 1 0	0	rd	0 0 1 0	0 0 0 0		
	CMP.B @ERs-,@+ERd	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 1 0	0 0 0 0		
	CMP.B @ERs-,@-ERd	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 0 1 1	0	rd	0 0 1 0	0 0 0 0		
	CMP.B @ERs-,@(d:2,ERd)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		0 0 d d	0	rd	0 0 1 0	0 0 0 0		
	CMP.B @ERs-,@(d:16,ERd)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 1 0 0	0	rd	0 0 1 0	0 0 0 0	0	d
	CMP.B @ERs-,@(d:32,ERd)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 1 0 0	1	rd	0 0 1 0	0 0 0 0	0	d d
	CMP.B @ERs-,@(d:16,Rd,B)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 1 0 1	0	rd	0 0 1 0	0 0 0 0	0	d
	CMP.B @ERs-,@(d:16,Rd,W)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 1 1 0	0	rd	0 0 1 0	0 0 0 0	0	d
	CMP.B @ERs-,@(d:16,ERd,L)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 1 1 1	0	rd	0 0 1 0	0 0 0 0	0	d
	CMP.B @ERs-,@(d:32,Rd,B)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 1 0 1	1	rd	0 0 1 0	0 0 0 0	0	d d
	CMP.B @ERs-,@(d:32,Rd,W)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 1 1 0	1	rd	0 0 1 0	0 0 0 0	0	d d
	CMP.B @ERs-,@(d:32,ERd,L)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 1 1 1	1	rd	0 0 1 0	0 0 0 0	0	d d
	CMP.B @ERs-,@aa:16	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		0 1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	0	a
	CMP.B @ERs-,@aa:32	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		0 1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	0	a a
	CMP.B @+ERs,@ERd	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		0 0 0 0	0	rd	0 0 1 0	0 0 0 0		
	CMP.B @+ERs,@ERd+	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		1 0 0 0	0	rd	0 0 1 0	0 0 0 0		
	CMP.B @+ERs,@ERd-	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		1 0 1 0	0	rd	0 0 1 0	0 0 0 0		
	CMP.B @+ERs,@+ERd	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 1 0	0 0 0 0		
CMP.B @+ERs,@-ERd	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		1 0 1 1	0	rd	0 0 1 0	0 0 0 0			
CMP.B @+ERs,@(d:2,ERd)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		0 0 d d	0	rd	0 0 1 0	0 0 0 0			
CMP.B @+ERs,@(d:16,ERd)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		1 1 0 0	0	rd	0 0 1 0	0 0 0 0	0	d	
CMP.B @+ERs,@(d:32,ERd)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		1 1 0 0	1	rd	0 0 1 0	0 0 0 0	0	d d	
CMP.B @+ERs,@(d:16,Rd,B)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		1 1 0 1	0	rd	0 0 1 0	0 0 0 0	0	d	
CMP.B @+ERs,@(d:16,Rd,W)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0		1 1 1 0	0	rd	0 0 1 0	0 0 0 0	0	d	

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
CMP	CMP.B @+ERs,@(d:16,ERd.L)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0 0		1 1 1 1	0	rd	0 0 1 0 0	0 0 0 0 0	d
	CMP.B @+ERs,@(d:32,Rd.B)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0 0		1 1 0 1	1	rd	0 0 1 0 0	0 0 0 0 0	d d
	CMP.B @+ERs,@(d:32,Rd.W)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0 0		1 1 1 0	1	rd	0 0 1 0 0	0 0 0 0 0	d d
	CMP.B @+ERs,@(d:32,ERd.L)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0 0		1 1 1 1	1	rd	0 0 1 0 0	0 0 0 0 0	d d
	CMP.B @+ERs,@aa:16	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0 0		0 1 0 0	0	0 0 0 0	0 0 1 0 0	0 0 0 0 0	a
	CMP.B @+ERs,@aa:32	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0 0		0 1 0 0	1	0 0 0 0	0 0 1 0 0	0 0 0 0 0	a a
	CMP.B @-ERs,@ERd	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		0 0 0 0	0	rd	0 0 1 0 0	0 0 0 0 0	
	CMP.B @-ERs,@ERd+	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		1 0 0 0	0	rd	0 0 1 0 0	0 0 0 0 0	
	CMP.B @-ERs,@ERd-	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		1 0 1 0	0	rd	0 0 1 0 0	0 0 0 0 0	
	CMP.B @-ERs,@+ERd	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		1 0 0 1	0	rd	0 0 1 0 0	0 0 0 0 0	
	CMP.B @-ERs,@-ERd	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		1 0 1 1	0	rd	0 0 1 0 0	0 0 0 0 0	
	CMP.B @-ERs,@(d:2,ERd)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		0 0 d d	0	rd	0 0 1 0 0	0 0 0 0 0	
	CMP.B @-ERs,@(d:16,ERd)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		1 1 0 0	0	rd	0 0 1 0 0	0 0 0 0 0	d
	CMP.B @-ERs,@(d:32,ERd)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		1 1 0 0	1	rd	0 0 1 0 0	0 0 0 0 0	d d
	CMP.B @-ERs,@(d:16,Rd.B)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		1 1 0 1	0	rd	0 0 1 0 0	0 0 0 0 0	d
	CMP.B @-ERs,@(d:16,Rd.W)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		1 1 1 0	0	rd	0 0 1 0 0	0 0 0 0 0	d
	CMP.B @-ERs,@(d:16,ERd.L)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		1 1 1 1	0	rd	0 0 1 0 0	0 0 0 0 0	d
	CMP.B @-ERs,@(d:32,Rd.B)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		1 1 0 1	1	rd	0 0 1 0 0	0 0 0 0 0	d d
	CMP.B @-ERs,@(d:32,Rd.W)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		1 1 1 0	1	rd	0 0 1 0 0	0 0 0 0 0	d d
	CMP.B @-ERs,@(d:32,ERd.L)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		1 1 1 1	1	rd	0 0 1 0 0	0 0 0 0 0	d d
	CMP.B @-ERs,@aa:16	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		0 1 0 0	0	0 0 0 0	0 0 1 0 0	0 0 0 0 0	a
	CMP.B @-ERs,@aa:32	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		0 1 0 0	1	0 0 0 0	0 0 1 0 0	0 0 0 0 0	a a
	CMP.B @(d:2,ERs),@ERd	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0 0		0 0 0 0	0	rd	0 0 1 0 0	0 0 0 0 0	
	CMP.B @(d:2,ERs),@ERd+	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0 0		1 0 0 0	0	rd	0 0 1 0 0	0 0 0 0 0	
	CMP.B @(d:2,ERs),@ERd-	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0 0		1 0 1 0	0	rd	0 0 1 0 0	0 0 0 0 0	
	CMP.B @(d:2,ERs),@+ERd	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0 0		1 0 0 1	0	rd	0 0 1 0 0	0 0 0 0 0	
	CMP.B @(d:2,ERs),@-ERd	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0 0		1 0 1 1	0	rd	0 0 1 0 0	0 0 0 0 0	
	CMP.B @(d:2,ERs),@(d:2,ERd)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0 0		0 0 d d	0	rd	0 0 1 0 0	0 0 0 0 0	
	CMP.B @(d:2,ERs),@(d:16,ERd)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0 0		1 1 0 0	0	rd	0 0 1 0 0	0 0 0 0 0	d
	CMP.B @(d:2,ERs),@(d:32,ERd)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0 0		1 1 0 0	1	rd	0 0 1 0 0	0 0 0 0 0	d d
	CMP.B @(d:2,ERs),@(d:16,Rd.B)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0 0		1 1 0 1	0	rd	0 0 1 0 0	0 0 0 0 0	d
	CMP.B @(d:2,ERs),@(d:16,Rd.W)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0 0		1 1 1 0	0	rd	0 0 1 0 0	0 0 0 0 0	d
	CMP.B @(d:2,ERs),@(d:16,ERd.L)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0 0		1 1 1 1	0	rd	0 0 1 0 0	0 0 0 0 0	d
	CMP.B @(d:2,ERs),@(d:32,Rd.B)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0 0		1 1 0 1	1	rd	0 0 1 0 0	0 0 0 0 0	d d



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
CMP	CMP.B @(d2,ERs),@(d32,Rd,W)	0	1	7	0 1 d d	6	8	0	rs 1 1 0 0 0		1 1 1 0 1	1	rd	0 0 1 0 0	0 0 0 0 0	d d
	CMP.B @(d2,ERs),@(d32,ERd,L)	0	1	7	0 1 d d	6	8	0	rs 1 1 0 0 0		1 1 1 1 1	1	rd	0 0 1 0 0	0 0 0 0 0	d d
	CMP.B @(d2,ERs),@aa:16	0	1	7	0 1 d d	6	8	0	rs 1 1 0 0 0		0 1 0 0 0	0 0 0 0 0	0 0 1 0 0	0 0 0 0 0	a	
	CMP.B @(d2,ERs),@aa:32	0	1	7	0 1 d d	6	8	0	rs 1 1 0 0 0		0 1 0 0 0	1 0 0 0 0	0 0 1 0 0	0 0 0 0 0	a a	
	CMP.B @(d:16,ERs),@ERd	0	1	7	0 1 0 0 0	6	E	0	rs 1 1 0 0 0 d		0 0 0 0 0	0	rd	0 0 1 0 0	0 0 0 0 0	
	CMP.B @(d:16,ERs),@ERd+	0	1	7	0 1 0 0 0	6	E	0	rs 1 1 0 0 0 d		1 0 0 0 0	0	rd	0 0 1 0 0	0 0 0 0 0	
	CMP.B @(d:16,ERs),@ERd-	0	1	7	0 1 0 0 0	6	E	0	rs 1 1 0 0 0 d		1 0 1 0 0	0	rd	0 0 1 0 0	0 0 0 0 0	
	CMP.B @(d:16,ERs),@+ERd	0	1	7	0 1 0 0 0	6	E	0	rs 1 1 0 0 0 d		1 0 0 0 1	0	rd	0 0 1 0 0	0 0 0 0 0	
	CMP.B @(d:16,ERs),@-ERd	0	1	7	0 1 0 0 0	6	E	0	rs 1 1 0 0 0 d		1 0 0 1 1	0	rd	0 0 1 0 0	0 0 0 0 0	
	CMP.B @(d:16,ERs),@(d2,ERd)	0	1	7	0 1 0 0 0	6	E	0	rs 1 1 0 0 0 d		0 0 d d	0	rd	0 0 1 0 0	0 0 0 0 0	
	CMP.B @(d:16,ERs),@(d:16,ERd)	0	1	7	0 1 0 0 0	6	E	0	rs 1 1 0 0 0 d		1 1 0 0 0	0	rd	0 0 1 0 0	0 0 0 0 0	d
	CMP.B @(d:16,ERs),@(d:32,ERd)	0	1	7	0 1 0 0 0	6	E	0	rs 1 1 0 0 0 d		1 1 0 0 1	1	rd	0 0 1 0 0	0 0 0 0 0	d d
	CMP.B @(d:16,ERs),@(d:16,Rd,B)	0	1	7	0 1 0 0 0	6	E	0	rs 1 1 0 0 0 d		1 1 0 1 0	0	rd	0 0 1 0 0	0 0 0 0 0	d
	CMP.B @(d:16,ERs),@(d:16,Rd,W)	0	1	7	0 1 0 0 0	6	E	0	rs 1 1 0 0 0 d		1 1 1 0 0	0	rd	0 0 1 0 0	0 0 0 0 0	d
	CMP.B @(d:16,ERs),@(d:16,ERd,L)	0	1	7	0 1 0 0 0	6	E	0	rs 1 1 0 0 0 d		1 1 1 1 0	0	rd	0 0 1 0 0	0 0 0 0 0	d
	CMP.B @(d:16,ERs),@(d:32,Rd,B)	0	1	7	0 1 0 0 0	6	E	0	rs 1 1 0 0 0 d		1 1 0 1 1	1	rd	0 0 1 0 0	0 0 0 0 0	d d
	CMP.B @(d:16,ERs),@(d:32,Rd,W)	0	1	7	0 1 0 0 0	6	E	0	rs 1 1 0 0 0 d		1 1 1 0 0	1	rd	0 0 1 0 0	0 0 0 0 0	d d
	CMP.B @(d:16,ERs),@(d:32,ERd,L)	0	1	7	0 1 0 0 0	6	E	0	rs 1 1 0 0 0 d		1 1 1 1 1	1	rd	0 0 1 0 0	0 0 0 0 0	d d
	CMP.B @(d:16,ERs),@aa:16	0	1	7	0 1 0 0 0	6	E	0	rs 1 1 0 0 0 d		0 1 0 0 0	0 0 0 0 0	0 0 1 0 0	0 0 0 0 0	a	
	CMP.B @(d:16,ERs),@aa:32	0	1	7	0 1 0 0 0	6	E	0	rs 1 1 0 0 0 d		0 1 0 0 0	1 0 0 0 0	0 0 1 0 0	0 0 0 0 0	a a	
	CMP.B @(d:32,ERs),@ERd	7	8	0	rs 0 1 0 0 0	6	A	0 0 1 0 0	1 1 0 0 0 d d		0 0 0 0 0	0	rd	0 0 1 0 0	0 0 0 0 0	
	CMP.B @(d:32,ERs),@ERd+	7	8	0	rs 0 1 0 0 0	6	A	0 0 1 0 0	1 1 0 0 0 d d		1 0 0 0 0	0	rd	0 0 1 0 0	0 0 0 0 0	
	CMP.B @(d:32,ERs),@ERd-	7	8	0	rs 0 1 0 0 0	6	A	0 0 1 0 0	1 1 0 0 0 d d		1 0 1 0 0	0	rd	0 0 1 0 0	0 0 0 0 0	
	CMP.B @(d:32,ERs),@+ERd	7	8	0	rs 0 1 0 0 0	6	A	0 0 1 0 0	1 1 0 0 0 d d		1 0 0 0 1	0	rd	0 0 1 0 0	0 0 0 0 0	
CMP.B @(d:32,ERs),@-ERd	7	8	0	rs 0 1 0 0 0	6	A	0 0 1 0 0	1 1 0 0 0 d d		1 0 1 1 0	0	rd	0 0 1 0 0	0 0 0 0 0		
CMP.B @(d:32,ERs),@(d2,ERd)	7	8	0	rs 0 1 0 0 0	6	A	0 0 1 0 0	1 1 0 0 0 d d		0 0 d d	0	rd	0 0 1 0 0	0 0 0 0 0		
CMP.B @(d:32,ERs),@(d:16,ERd)	7	8	0	rs 0 1 0 0 0	6	A	0 0 1 0 0	1 1 0 0 0 d d		1 1 0 0 0	0	rd	0 0 1 0 0	0 0 0 0 0	d	
CMP.B @(d:32,ERs),@(d:32,ERd)	7	8	0	rs 0 1 0 0 0	6	A	0 0 1 0 0	1 1 0 0 0 d d		1 1 0 0 1	1	rd	0 0 1 0 0	0 0 0 0 0	d d	
CMP.B @(d:32,ERs),@(d:16,Rd,B)	7	8	0	rs 0 1 0 0 0	6	A	0 0 1 0 0	1 1 0 0 0 d d		1 1 0 1 0	0	rd	0 0 1 0 0	0 0 0 0 0	d	
CMP.B @(d:32,ERs),@(d:16,Rd,W)	7	8	0	rs 0 1 0 0 0	6	A	0 0 1 0 0	1 1 0 0 0 d d		1 1 1 0 0	0	rd	0 0 1 0 0	0 0 0 0 0	d	
CMP.B @(d:32,ERs),@(d:16,ERd,L)	7	8	0	rs 0 1 0 0 0	6	A	0 0 1 0 0	1 1 0 0 0 d d		1 1 1 1 0	0	rd	0 0 1 0 0	0 0 0 0 0	d	
CMP.B @(d:32,ERs),@(d:32,Rd,B)	7	8	0	rs 0 1 0 0 0	6	A	0 0 1 0 0	1 1 0 0 0 d d		1 1 0 1 1	1	rd	0 0 1 0 0	0 0 0 0 0	d d	
CMP.B @(d:32,ERs),@(d:32,Rd,W)	7	8	0	rs 0 1 0 0 0	6	A	0 0 1 0 0	1 1 0 0 0 d d		1 1 1 0 1	1	rd	0 0 1 0 0	0 0 0 0 0	d d	
CMP.B @(d:32,ERs),@(d:32,ERd,L)	7	8	0	rs 0 1 0 0 0	6	A	0 0 1 0 0	1 1 0 0 0 d d		1 1 1 1 1	1	rd	0 0 1 0 0	0 0 0 0 0	d d	

Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension	
		15	8	7	0	15	8		7	0	15	8	7	0		
CMP	CMP.B @(d:32,ERs),@aa:16	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	0 0 0 0	0 0 1 0	0 0 0 0	a
	CMP.B @(d:32,ERs),@aa:32	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	1 0 0 0	0 0 1 0	0 0 0 0	a a
	CMP.B @(d:16,Rs,B),@ERd	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d		0 0 0 0	0 rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:16,Rs,B),@ERd+	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d		1 0 0 0	0 rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:16,Rs,B),@ERd-	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d		1 0 1 0	0 rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:16,Rs,B),@+ERd	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d		1 0 0 1	0 rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:16,Rs,B),@-ERd	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d		1 0 1 1	0 rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:16,Rs,B),@(d:2,ERd)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d		0 0 d d	0 rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:16,Rs,B),@(d:16,ERd)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d		1 1 0 0	0 rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:16,Rs,B),@(d:32,ERd)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d		1 1 0 0	1 rd	0 0 1 0	0 0 0 0	d d
	CMP.B @(d:16,Rs,B),@(d:16,Rd,B)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d		1 1 0 1	0 rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:16,Rs,B),@(d:16,Rd,W)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d		1 1 1 0	0 rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:16,Rs,B),@(d:16,ERd,L)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d		1 1 1 1	0 rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:16,Rs,B),@(d:32,Rd,B)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d		1 1 0 1	1 rd	0 0 1 0	0 0 0 0	d d
	CMP.B @(d:16,Rs,B),@(d:32,Rd,W)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d		1 1 1 0	1 rd	0 0 1 0	0 0 0 0	d d
	CMP.B @(d:16,Rs,B),@(d:32,ERd,L)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d		1 1 1 1	1 rd	0 0 1 0	0 0 0 0	d d
	CMP.B @(d:16,Rs,B),@aa:16	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d		0 1 0 0	0 0 0 0	0 0 1 0	0 0 0 0	a
	CMP.B @(d:16,Rs,B),@aa:32	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d		0 1 0 0	1 0 0 0	0 0 1 0	0 0 0 0	a a
	CMP.B @(d:16,Rs,W),@ERd	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d		0 0 0 0	0 rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:16,Rs,W),@ERd+	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d		1 0 0 0	0 rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:16,Rs,W),@ERd-	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d		1 0 1 0	0 rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:16,Rs,W),@+ERd	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d		1 0 0 1	0 rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:16,Rs,W),@-ERd	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d		1 0 1 1	0 rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:16,Rs,W),@(d:2,ERd)	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d		0 0 d d	0 rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:16,Rs,W),@(d:16,ERd)	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d		1 1 0 0	0 rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:16,Rs,W),@(d:32,ERd)	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d		1 1 0 0	1 rd	0 0 1 0	0 0 0 0	d d
	CMP.B @(d:16,Rs,W),@(d:16,Rd,B)	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d		1 1 0 1	0 rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:16,Rs,W),@(d:16,Rd,W)	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d		1 1 1 0	0 rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:16,Rs,W),@(d:16,ERd,L)	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d		1 1 1 1	0 rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:16,Rs,W),@(d:32,Rd,B)	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d		1 1 0 1	1 rd	0 0 1 0	0 0 0 0	d d
	CMP.B @(d:16,Rs,W),@(d:32,Rd,W)	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d		1 1 1 0	1 rd	0 0 1 0	0 0 0 0	d d
	CMP.B @(d:16,Rs,W),@(d:32,ERd,L)	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d		1 1 1 1	1 rd	0 0 1 0	0 0 0 0	d d
CMP.B @(d:16,Rs,W),@aa:16	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d		0 1 0 0	0 0 0 0	0 0 1 0	0 0 0 0	a	
CMP.B @(d:16,Rs,W),@aa:32	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d		0 1 0 0	1 0 0 0	0 0 1 0	0 0 0 0	a a	



Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
CMP	CMP.B @(d:16,ERs.L),@ERd	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:16,ERs.L),@ERd+	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:16,ERs.L),@ERd-	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:16,ERs.L),@+ERd	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:16,ERs.L),@-ERd	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:16,ERs.L),@(d:2,ERd)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:16,ERs.L),@(d:16,ERd)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:16,ERs.L),@(d:32,ERd)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @(d:16,ERs.L),@(d:16,Rd.B)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:16,ERs.L),@(d:16,Rd.W)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:16,ERs.L),@(d:16,ERd.L)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:16,ERs.L),@(d:32,Rd.B)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @(d:16,ERs.L),@(d:32,Rd.W)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @(d:16,ERs.L),@(d:32,ERd.L)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @(d:16,ERs.L),@aa:16	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 0 1 0	0 0 0 0	a	
	CMP.B @(d:16,ERs.L),@aa:32	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 0 1 0	0 0 0 0	a a	
	CMP.B @(d:32,Rs.B),@ERd	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:32,Rs.B),@ERd+	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:32,Rs.B),@ERd-	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:32,Rs.B),@+ERd	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:32,Rs.B),@-ERd	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:32,Rs.B),@(d:2,ERd)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:32,Rs.B),@(d:16,ERd)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:32,Rs.B),@(d:32,ERd)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @(d:32,Rs.B),@(d:16,Rd.B)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:32,Rs.B),@(d:16,Rd.W)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:32,Rs.B),@(d:16,ERd.L)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:32,Rs.B),@(d:32,Rd.B)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @(d:32,Rs.B),@(d:32,Rd.W)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @(d:32,Rs.B),@(d:32,ERd.L)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @(d:32,Rs.B),@aa:16	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	0 0 0 0	0 0 1 0	0 0 0 0	a	
	CMP.B @(d:32,Rs.B),@aa:32	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	1 0 0 0	0 0 1 0	0 0 0 0	a a	
CMP.B @(d:32,Rs.W),@ERd	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 0 1 0	0 0 0 0		
CMP.B @(d:32,Rs.W),@ERd+	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 0 1 0	0 0 0 0		

Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension		
		15	8	7	0	15	8		7	0	15	8	7	0			
CMP	CMP.B @(d:32.Rs,W),@ERd-	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:32.Rs,W),@+ERd	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:32.Rs,W),@-ERd	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:32.Rs,W),@(d:2,ERd)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:32.Rs,W),@(d:16,ERd)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:32.Rs,W),@(d:32,ERd)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @(d:32.Rs,W),@(d:16,Rd,B)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:32.Rs,W),@(d:16,Rd,W)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:32.Rs,W),@(d:16,ERd,L)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:32.Rs,W),@(d:32,Rd,B)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @(d:32.Rs,W),@(d:32,Rd,W)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @(d:32.Rs,W),@(d:32,ERd,L)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @(d:32.Rs,W),@aa:16	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	a
	CMP.B @(d:32.Rs,W),@aa:32	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	a a
	CMP.B @(d:32,ERs,L),@ERd	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:32,ERs,L),@ERd+	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:32,ERs,L),@ERd-	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:32,ERs,L),@+ERd	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:32,ERs,L),@-ERd	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:32,ERs,L),@(d:2,ERd)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @(d:32,ERs,L),@(d:16,ERd)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:32,ERs,L),@(d:32,ERd)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @(d:32,ERs,L),@(d:16,Rd,B)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:32,ERs,L),@(d:16,Rd,W)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:32,ERs,L),@(d:16,ERd,L)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @(d:32,ERs,L),@(d:32,Rd,B)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @(d:32,ERs,L),@(d:32,Rd,W)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @(d:32,ERs,L),@(d:32,ERd,L)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @(d:32,ERs,L),@aa:16	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	a
	CMP.B @(d:32,ERs,L),@aa:32	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	a a
CMP.B @aa:16,@ERd							6	A	0 0 0 1	0 1 0 1	a	0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
CMP.B @aa:16,@ERd+							6	A	0 0 0 1	0 1 0 1	a	1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
CMP.B @aa:16,@ERd-							6	A	0 0 0 1	0 1 0 1	a	1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
CMP.B @aa:16,@+ERd							6	A	0 0 0 1	0 1 0 1	a	1 0 0 1	0	rd	0 0 1 0	0 0 0 0	



Instruction	Mnemonic	Opcode			Opcode			EA Ex- tension	Opcode			EA Extension			
		15	8	7	0	15	8		7	0	15		8	7	0
CMP	CMP.B @aa:16,@ERd				6	A	0 0 0 0 1	0 1 0 1 a		1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @aa:16,@(d:2,ERd)				6	A	0 0 0 0 1	0 1 0 1 a		0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @aa:16,@(d:16,ERd)				6	A	0 0 0 0 1	0 1 0 1 a		1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @aa:16,@(d:32,ERd)				6	A	0 0 0 0	0 1 0 1 a		1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @aa:16,@(d:16,Rd.B)				6	A	0 0 0 0 1	0 1 0 1 a		1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @aa:16,@(d:16,Rd.W)				6	A	0 0 0 0 1	0 1 0 1 a		1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @aa:16,@(d:16,ERd.L)				6	A	0 0 0 0 1	0 1 0 1 a		1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @aa:16,@(d:32,Rd.B)				6	A	0 0 0 0 1	0 1 0 1 a		1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @aa:16,@(d:32,Rd.W)				6	A	0 0 0 0 1	0 1 0 1 a		1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @aa:16,@(d:32,ERd.L)				6	A	0 0 0 0 1	0 1 0 1 a		1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @aa:16,@aa:16				6	A	0 0 0 0 1	0 1 0 1 a		0 1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	a
	CMP.B @aa:16,@aa:32				6	A	0 0 0 0 1	0 1 0 1 a		0 1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	a a
	CMP.B @aa:32,@ERd				6	A	0 0 0 1	0 1 0 1 a	a	0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @aa:32,@ERd+				6	A	0 0 0 1	0 1 0 1 a	a a	1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @aa:32,@ERd-				6	A	0 0 0 1	0 1 0 1 a	a a	1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @aa:32,@+ERd				6	A	0 0 0 1	0 1 0 1 a	a a	1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @aa:32,@-ERd				6	A	0 0 0 1	0 1 0 1 a	a a	1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @aa:32,@(d:2,ERd)				6	A	0 0 0 1	0 1 0 1 a	a a	0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.B @aa:32,@(d:16,ERd)				6	A	0 0 0 1	0 1 0 1 a	a a	1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @aa:32,@(d:32,ERd)				6	A	0 0 0 1	0 1 0 1 a	a a	1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @aa:32,@(d:16,Rd.B)				6	A	0 0 0 1	0 1 0 1 a	a a	1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @aa:32,@(d:16,Rd.W)				6	A	0 0 0 1	0 1 0 1 a	a a	1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @aa:32,@(d:16,ERd.L)				6	A	0 0 0 1	0 1 0 1 a	a a	1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.B @aa:32,@(d:32,Rd.B)				6	A	0 0 0 1	0 1 0 1 a	a a	1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @aa:32,@(d:32,Rd.W)				6	A	0 0 0 1	0 1 0 1 a	a a	1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @aa:32,@(d:32,ERd.L)				6	A	0 0 0 1	0 1 0 1 a	a a	1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.B @aa:32,@aa:16				6	A	0 0 0 1	0 1 0 1 a	a a	0 1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	a
	CMP.B @aa:32,@aa:32				6	A	0 0 0 1	0 1 0 1 a	a a	0 1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	a a
	CMP.W #xx:3,Rd									1	F	0 x x x	rd		
	CMP.W #bxx:16,Rd									7	9	0 0 1 0	rd	x	
	CMP.W #xx:3,@ERd				7	D	1	rd	0 0 0 0	1	F	0 x x x	0 0 0 0		
	CMP.W #xx:3,@aa:16				6	B	0 0 0 1	1 0 0 0	a	1	F	0 x x x	0 0 0 0		
CMP.W #bxx:3,@aa:32				6	B	0 0 0 1	1 0 0 0	a a	1	F	0 x x x	0 0 0 0			
CMP.W #xx:16,@ERd				0	1	5	1 1 1 0		0 0 0 0	0	rd	0 0 1 0	0 0 0 0	x	

Instruction	Mnemonic	Opcode			Opcode			EA Ex- tension	Opcode			EA Extension					
		15	8	7	0	15	8		7	0	15			8	7	0	
CMP	CMP.W #xoc:16,@ERd+				0	1	5	1 1 1 0		1 0 0 0	0	rd	0 0 1 0	0 0 0 0			x
	CMP.W #xoc:16,@ERd-				0	1	5	1 1 1 0		1 0 1 0	0	rd	0 0 1 0	0 0 0 0			x
	CMP.W #xoc:16,@+ERd				0	1	5	1 1 1 0		1 0 0 1	0	rd	0 0 1 0	0 0 0 0			x
	CMP.W #xoc:16,@-ERd				0	1	5	1 1 1 0		1 0 1 1	0	rd	0 0 1 0	0 0 0 0			x
	CMP.W #xoc:16,@(d:2,ERd)				0	1	5	1 1 1 0		0 0 d d	0	rd	0 0 1 0	0 0 0 0			x
	CMP.W #xoc:16,@(d:16,ERd)				0	1	5	1 1 1 0		1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d		x
	CMP.W #xoc:16,@(d:32,ERd)				0	1	5	1 1 1 0		1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d	d	x
	CMP.W #xoc:16,@(d:16,Rd.B)				0	1	5	1 1 1 0		1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d		x
	CMP.W #xoc:16,@(d:16,Rd.W)				0	1	5	1 1 1 0		1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d		x
	CMP.W #xoc:16,@(d:16,ERd.L)				0	1	5	1 1 1 0		1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d		x
	CMP.W #xoc:16,@(d:32,Rd.B)				0	1	5	1 1 1 0		1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d	d	x
	CMP.W #xoc:16,@(d:32,Rd.W)				0	1	5	1 1 1 0		1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d	d	x
	CMP.W #xoc:16,@(d:32,ERd.L)				0	1	5	1 1 1 0		1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d	d	x
	CMP.W #xoc:16,@aa:16				0	1	5	1 1 1 0		0 1 0 0	0 0 0 0	0 0 1 0	0 0 0 0	0 0 0 0	a		x
	CMP.W #xoc:16,@aa:32				0	1	5	1 1 1 0		0 1 0 0	1 0 0 0	0 0 1 0	0 0 0 0	0 0 0 0	a	a	x
	CMP.W Rs,Rd									1	D	rs	rd				
	CMP.W Rs,@ERd				7	D	1	rd	0 0 0 0		1	D	rs	0 0 0 0			
	CMP.W Rs,@ERd+				0	1	5	1 0 0 1		1 0 0 0	0	rd	0 0 1 0	rs			
	CMP.W Rs,@ERd-				0	1	5	1 0 0 1		1 0 1 0	0	rd	0 0 1 0	rs			
	CMP.W Rs,@+ERd				0	1	5	1 0 0 1		1 0 0 1	0	rd	0 0 1 0	rs			
	CMP.W Rs,@-ERd				0	1	5	1 0 0 1		1 0 1 1	0	rd	0 0 1 0	rs			
	CMP.W Rs,@(d:2,ERd)				0	1	5	1 0 0 1		0 0 d d	0	rd	0 0 1 0	rs			
	CMP.W Rs,@(d:16,ERd)				0	1	5	1 0 0 1		1 1 0 0	0	rd	0 0 1 0	rs	d		
	CMP.W Rs,@(d:32,ERd)				0	1	5	1 0 0 1		1 1 0 0	1	rd	0 0 1 0	rs	d	d	
	CMP.W Rs,@(d:16,Rd.B)				0	1	5	1 0 0 1		1 1 0 1	0	rd	0 0 1 0	rs	d		
	CMP.W Rs,@(d:16,Rd.W)				0	1	5	1 0 0 1		1 1 1 0	0	rd	0 0 1 0	rs	d		
	CMP.W Rs,@(d:16,ERd.L)				0	1	5	1 0 0 1		1 1 1 1	0	rd	0 0 1 0	rs	d		
	CMP.W Rs,@(d:32,Rd.B)				0	1	5	1 0 0 1		1 1 0 1	1	rd	0 0 1 0	rs	d	d	
	CMP.W Rs,@(d:32,Rd.W)				0	1	5	1 0 0 1		1 1 1 0	1	rd	0 0 1 0	rs	d	d	
	CMP.W Rs,@(d:32,ERd.L)				0	1	5	1 0 0 1		1 1 1 1	1	rd	0 0 1 0	rs	d	d	
	CMP.W Rs,@aa:16				6	B	0 0 0 1	1 0 0 0	a		1	D	rs	0 0 0 0			
	CMP.W Rs,@aa:32				6	B	0 0 1 1	1 0 0 0	a a		1	D	rs	0 0 0 0			
	CMP.W @ERs,Rd				7	C	1	rs	0 0 0 0		1	D	0 0 0 0	rd			
	CMP.W @ERs+,Rd				0	1	5	1 0 1 0		1 0 0 0	0	rs	0 0 1 0	rd			

Instruction	Mnemonic	Opcode			Opcode			EA Extension	Opcode			EA Extension																
		15	8	7	0	15	8		7	0	15		8	7	0													
CMP	CMP.W @ERs,Rd				0	1	5	1 0 1 0		1	0	1	0	0	rs	0	0	1	0	rd								
	CMP.W @+ERs,Rd				0	1	5	1 0 1 0		1	0	0	1	0	rs	0	0	1	0	rd								
	CMP.W @-ERs,Rd				0	1	5	1 0 1 0		1	0	1	1	0	rs	0	0	1	0	rd								
	CMP.W @(d:2,ERs),Rd				0	1	5	1 0 1 0		0	0	d	d	0	rs	0	0	1	0	rd								
	CMP.W @(d:16,ERs),Rd				0	1	5	1 0 1 0		1	1	0	0	0	rs	0	0	1	0	rd								
	CMP.W @(d:32,ERs),Rd				0	1	5	1 0 1 0		1	1	0	0	1	rs	0	0	1	0	rd								
	CMP.W @(d:16,Rs,B),Rd				0	1	5	1 0 1 0		1	1	0	1	0	rs	0	0	1	0	rd								
	CMP.W @(d:16,Rs,W),Rd				0	1	5	1 0 1 0		1	1	1	0	0	rs	0	0	1	0	rd								
	CMP.W @(d:16,ERs,L),Rd				0	1	5	1 0 1 0		1	1	1	1	0	rs	0	0	1	0	rd								
	CMP.W @(d:32,Rs,B),Rd				0	1	5	1 0 1 0		1	1	0	1	1	rs	0	0	1	0	rd								
	CMP.W @(d:32,Rs,W),Rd				0	1	5	1 0 1 0		1	1	1	0	1	rs	0	0	1	0	rd								
	CMP.W @(d:32,ERs,L),Rd				0	1	5	1 0 1 0		1	1	1	1	1	rs	0	0	1	0	rd								
	CMP.W @aa:16,Rd				6	B	0	0	0	0	0	0	0	a		1		D	0	0	0	0	rd					
	CMP.W @aa:32,Rd				6	B	0	0	0	1	1	0	0	0	a	a		1		D	0	0	0	0	rd			
	CMP.W @ERs,@ERd				7	C	1	rs	0	1	0	1	1		0	0	0	0	0	rd	0	0	1	0	0	0	0	0
	CMP.W @ERs,@ERd+				7	C	1	rs	0	1	0	1	1		1	0	0	0	0	rd	0	0	1	0	0	0	0	0
	CMP.W @ERs,@ERd-				7	C	1	rs	0	1	0	1	1		1	0	1	0	0	rd	0	0	1	0	0	0	0	0
	CMP.W @ERs,@+ERd				7	C	1	rs	0	1	0	1	1		1	0	0	1	0	rd	0	0	1	0	0	0	0	0
	CMP.W @ERs,@-ERd				7	C	1	rs	0	1	0	1	1		1	0	1	1	0	rd	0	0	1	0	0	0	0	0
	CMP.W @ERs,@(d:2,ERd)				7	C	1	rs	0	1	0	1	1		0	0	d	d	0	rd	0	0	1	0	0	0	0	0
	CMP.W @ERs,@(d:16,ERd)				7	C	1	rs	0	1	0	1	1		1	1	0	0	0	rd	0	0	1	0	0	0	0	0
	CMP.W @ERs,@(d:32,ERd)				7	C	1	rs	0	1	0	1	1		1	1	0	0	1	rd	0	0	1	0	0	0	0	0
	CMP.W @ERs,@(d:16,Rd,B)				7	C	1	rs	0	1	0	1	1		1	1	0	1	0	rd	0	0	1	0	0	0	0	0
	CMP.W @ERs,@(d:16,Rd,W)				7	C	1	rs	0	1	0	1	1		1	1	1	0	0	rd	0	0	1	0	0	0	0	0
	CMP.W @ERs,@(d:16,ERd,L)				7	C	1	rs	0	1	0	1	1		1	1	1	1	0	rd	0	0	1	0	0	0	0	0
	CMP.W @ERs,@(d:32,Rd,B)				7	C	1	rs	0	1	0	1	1		1	1	0	1	1	rd	0	0	1	0	0	0	0	0
	CMP.W @ERs,@(d:32,Rd,W)				7	C	1	rs	0	1	0	1	1		1	1	1	0	1	rd	0	0	1	0	0	0	0	0
	CMP.W @ERs,@(d:32,ERd,L)				7	C	1	rs	0	1	0	1	1		1	1	1	1	1	rd	0	0	1	0	0	0	0	0
	CMP.W @ERs,@aa:16				7	C	1	rs	0	1	0	1	1		0	1	0	0	0	0	rd	0	0	1	0	0	0	0
	CMP.W @ERs,@aa:32				7	C	1	rs	0	1	0	1	1		0	1	0	0	1	0	0	0	rd	0	0	1	0	0
	CMP.W @ERs+,@ERd				0	1	5	0	1	0	0	0	6	D	0	rs	1	1	0	0	rd	0	0	1	0	0	0	0
	CMP.W @ERs+,@ERd+				0	1	5	0	1	0	0	0	6	D	0	rs	1	1	0	0	rd	0	0	1	0	0	0	0
CMP.W @ERs+,@ERd-				0	1	5	0	1	0	0	0	6	D	0	rs	1	1	0	0	rd	0	0	1	0	0	0	0	
CMP.W @ERs+,@+ERd				0	1	5	0	1	0	0	0	6	D	0	rs	1	1	0	0	rd	0	0	1	0	0	0	0	

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
CMP	CMP.W @ERs+, @ERd	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0 0		1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @ERs+, @(d:2,ERd)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0 0		0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @ERs+, @(d:16,ERd)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0 0		1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @ERs+, @(d:32,ERd)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0 0		1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @ERs+, @(d:16,Rd,B)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0 0		1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @ERs+, @(d:16,Rd,W)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0 0		1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @ERs+, @(d:16,ERd,L)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0 0		1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @ERs+, @(d:32,Rd,B)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0 0		1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @ERs+, @(d:32,Rd,W)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0 0		1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @ERs+, @(d:32,ERd,L)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0 0		1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @ERs+, @aa:16	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0 0		0 1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	a
	CMP.W @ERs+, @aa:32	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0 0		0 1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	a a
	CMP.W @ERs-, @ERd	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0 0		0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @ERs-, @ERd+	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0 0		1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @ERs-, @ERd-	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0 0		1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @ERs-, @+ERd	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0 0		1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @ERs-, @-ERd	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0 0		1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @ERs-, @(d:2,ERd)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0 0		0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @ERs-, @(d:16,ERd)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0 0		1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @ERs-, @(d:32,ERd)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0 0		1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @ERs-, @(d:16,Rd,B)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0 0		1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @ERs-, @(d:16,Rd,W)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0 0		1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @ERs-, @(d:16,ERd,L)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0 0		1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @ERs-, @(d:32,Rd,B)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0 0		1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @ERs-, @(d:32,Rd,W)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0 0		1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @ERs-, @(d:32,ERd,L)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0 0		1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @ERs-, @aa:16	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0 0		0 1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	a
	CMP.W @ERs-, @aa:32	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0 0		0 1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	a a
	CMP.W @+ERs, @ERd	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0 0		0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @+ERs, @ERd+	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0 0		1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @+ERs, @ERd-	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0 0		1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @+ERs, @+ERd	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0 0		1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
CMP.W @+ERs, @-ERd	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0 0		1 0 1 1	0	rd	0 0 1 0	0 0 0 0		
CMP.W @+ERs, @(d:2,ERd)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0 0		0 0 d d	0	rd	0 0 1 0	0 0 0 0		



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
CMP	CMP.W @+ERs,@(d:16,ERd)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @+ERs,@(d:32,ERd)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @+ERs,@(d:16,Rd.B)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @+ERs,@(d:16,Rd.W)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @+ERs,@(d:16,ERd.L)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @+ERs,@(d:32,Rd.B)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @+ERs,@(d:32,Rd.W)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @+ERs,@(d:32,ERd.L)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @+ERs,@aa:16	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		0 1 0 0	0 0 0 0	0 0 1 0	0 0 0 0	a	
	CMP.W @+ERs,@aa:32	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		0 1 0 0	1 0 0 0	0 0 1 0	0 0 0 0	a a	
	CMP.W @-ERs,@ERd	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @-ERs,@ERd+	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @-ERs,@ERd-	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @-ERs,@+ERd	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @-ERs,@-ERd	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @-ERs,@(d:2,ERd)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @-ERs,@(d:16,ERd)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @-ERs,@(d:32,ERd)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @-ERs,@(d:16,Rd.B)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @-ERs,@(d:16,Rd.W)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @-ERs,@(d:16,ERd.L)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @-ERs,@(d:32,Rd.B)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @-ERs,@(d:32,Rd.W)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @-ERs,@(d:32,ERd.L)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @-ERs,@aa:16	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		0 1 0 0	0 0 0 0	0 0 1 0	0 0 0 0	a	
	CMP.W @-ERs,@aa:32	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		0 1 0 0	1 0 0 0	0 0 1 0	0 0 0 0	a a	
	CMP.W @(d:2,ERs),@ERd	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:2,ERs),@ERd+	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:2,ERs),@ERd-	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:2,ERs),@+ERd	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:2,ERs),@-ERd	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:2,ERs),@(d:2,ERd)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		0 0 d d	0	rd	0 0 1 0	0 0 0 0	
CMP.W @(d:2,ERs),@(d:16,ERd)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d	
CMP.W @(d:2,ERs),@(d:32,ERd)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d	

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
CMP	CMP.W @(d:2,ERs),@(d:16,Rd,B)	0	1	5	0 1 d d	6	9	0	rs 1 1 0 0		1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @(d:2,ERs),@(d:16,Rd,W)	0	1	5	0 1 d d	6	9	0	rs 1 1 0 0		1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @(d:2,ERs),@(d:16,ERd,L)	0	1	5	0 1 d d	6	9	0	rs 1 1 0 0		1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @(d:2,ERs),@(d:32,Rd,B)	0	1	5	0 1 d d	6	9	0	rs 1 1 0 0		1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @(d:2,ERs),@(d:32,Rd,W)	0	1	5	0 1 d d	6	9	0	rs 1 1 0 0		1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @(d:2,ERs),@(d:32,ERd,L)	0	1	5	0 1 d d	6	9	0	rs 1 1 0 0		1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @(d:2,ERs),@aa:16	0	1	5	0 1 d d	6	9	0	rs 1 1 0 0		0 1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	a
	CMP.W @(d:2,ERs),@aa:32	0	1	5	0 1 d d	6	9	0	rs 1 1 0 0		0 1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	a a
	CMP.W @(d:16,ERs),@ERd	0	1	5	0 1 0 0	6	F	0	rs 1 1 0 0	d	0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:16,ERs),@ERd+	0	1	5	0 1 0 0	6	F	0	rs 1 1 0 0	d	1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:16,ERs),@ERd-	0	1	5	0 1 0 0	6	F	0	rs 1 1 0 0	d	1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:16,ERs),@+ERd	0	1	5	0 1 0 0	6	F	0	rs 1 1 0 0	d	1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:16,ERs),@-ERd	0	1	5	0 1 0 0	6	F	0	rs 1 1 0 0	d	1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:16,ERs),@(d:2,ERd)	0	1	5	0 1 0 0	6	F	0	rs 1 1 0 0	d	0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:16,ERs),@(d:16,ERd)	0	1	5	0 1 0 0	6	F	0	rs 1 1 0 0	d	1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @(d:16,ERs),@(d:32,ERd)	0	1	5	0 1 0 0	6	F	0	rs 1 1 0 0	d	1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @(d:16,ERs),@(d:16,Rd,B)	0	1	5	0 1 0 0	6	F	0	rs 1 1 0 0	d	1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @(d:16,ERs),@(d:16,Rd,W)	0	1	5	0 1 0 0	6	F	0	rs 1 1 0 0	d	1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @(d:16,ERs),@(d:16,ERd,L)	0	1	5	0 1 0 0	6	F	0	rs 1 1 0 0	d	1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @(d:16,ERs),@(d:32,Rd,B)	0	1	5	0 1 0 0	6	F	0	rs 1 1 0 0	d	1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @(d:16,ERs),@(d:32,Rd,W)	0	1	5	0 1 0 0	6	F	0	rs 1 1 0 0	d	1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @(d:16,ERs),@(d:32,ERd,L)	0	1	5	0 1 0 0	6	F	0	rs 1 1 0 0	d	1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @(d:16,ERs),@aa:16	0	1	5	0 1 0 0	6	F	0	rs 1 1 0 0	d	0 1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	a
	CMP.W @(d:16,ERs),@aa:32	0	1	5	0 1 0 0	6	F	0	rs 1 1 0 0	d	0 1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	a a
	CMP.W @(d:32,ERs),@ERd	7	8	0	rs 0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:32,ERs),@ERd+	7	8	0	rs 0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:32,ERs),@ERd-	7	8	0	rs 0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:32,ERs),@+ERd	7	8	0	rs 0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:32,ERs),@-ERd	7	8	0	rs 0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:32,ERs),@(d:2,ERd)	7	8	0	rs 0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:32,ERs),@(d:16,ERd)	7	8	0	rs 0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @(d:32,ERs),@(d:32,ERd)	7	8	0	rs 0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
CMP.W @(d:32,ERs),@(d:16,Rd,B)	7	8	0	rs 0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d	
CMP.W @(d:32,ERs),@(d:16,Rd,W)	7	8	0	rs 0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d	



Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension		
		15	8	7	0	15	8		7	0	15	8	7	0			
CMP	CMP.W @(d:32,ERs),@(d:16,ERd,L)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @(d:32,ERs),@(d:32,Rd,B)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @(d:32,ERs),@(d:32,Rd,W)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @(d:32,ERs),@(d:32,ERd,L)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @(d:32,ERs),@aa:16	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	a
	CMP.W @(d:32,ERs),@aa:32	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	a a
	CMP.W @(d:16,Rs,B),@ERd	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:16,Rs,B),@ERd+	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:16,Rs,B),@ERd-	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:16,Rs,B),@+ERd	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:16,Rs,B),@-ERd	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:16,Rs,B),@(d:2,ERd)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:16,Rs,B),@(d:16,ERd)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @(d:16,Rs,B),@(d:32,ERd)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @(d:16,Rs,B),@(d:16,Rd,B)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @(d:16,Rs,B),@(d:16,Rd,W)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @(d:16,Rs,B),@(d:16,ERd,L)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @(d:16,Rs,B),@(d:32,Rd,B)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @(d:16,Rs,B),@(d:32,Rd,W)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @(d:16,Rs,B),@(d:32,ERd,L)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @(d:16,Rs,B),@aa:16	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	a
	CMP.W @(d:16,Rs,B),@aa:32	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	a a
	CMP.W @(d:16,Rs,W),@ERd	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:16,Rs,W),@ERd+	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:16,Rs,W),@ERd-	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:16,Rs,W),@+ERd	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:16,Rs,W),@-ERd	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:16,Rs,W),@(d:2,ERd)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @(d:16,Rs,W),@(d:16,ERd)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @(d:16,Rs,W),@(d:32,ERd)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
CMP.W @(d:16,Rs,W),@(d:16,Rd,B)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d	
CMP.W @(d:16,Rs,W),@(d:16,Rd,W)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d	
CMP.W @(d:16,Rs,W),@(d:16,ERd,L)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d	
CMP.W @(d:16,Rs,W),@(d:32,Rd,B)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d	

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension			
		15	8	7	0	15	8	7	0		15	8	7	0				
CMP	CMP.W @(d:16,Rs.W),@(d:32,Rd.W)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 0 1 0	0 0 0 0	0 0 0 0	d d
	CMP.W @(d:16,Rs.W),@(d:32,ERd.L)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 0 1 0	0 0 0 0	0 0 0 0	d d
	CMP.W @(d:16,Rs.W),@aa:16	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	0 0 0 0	a
	CMP.W @(d:16,Rs.W),@aa:32	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	0 0 0 0	a a
	CMP.W @(d:16,ERs.L),@ERd	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 0 1 0	0 0 0 0	0 0 0 0	
	CMP.W @(d:16,ERs.L),@ERd+	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 0 1 0	0 0 0 0	0 0 0 0	
	CMP.W @(d:16,ERs.L),@ERd-	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 0 1 0	0 0 0 0	0 0 0 0	
	CMP.W @(d:16,ERs.L),@+ERd	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 0 1 0	0 0 0 0	0 0 0 0	
	CMP.W @(d:16,ERs.L),@-ERd	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 0 1 0	0 0 0 0	0 0 0 0	
	CMP.W @(d:16,ERs.L),@(d:2,ERd)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 0 1 0	0 0 0 0	0 0 0 0	
	CMP.W @(d:16,ERs.L),@(d:16,ERd)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 0 1 0	0 0 0 0	0 0 0 0	d
	CMP.W @(d:16,ERs.L),@(d:32,ERd)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 0 1 0	0 0 0 0	0 0 0 0	d d
	CMP.W @(d:16,ERs.L),@(d:16,Rd.B)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 0 1 0	0 0 0 0	0 0 0 0	d
	CMP.W @(d:16,ERs.L),@(d:16,Rd.W)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 0 1 0	0 0 0 0	0 0 0 0	d
	CMP.W @(d:16,ERs.L),@(d:16,ERd.L)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 0 1 0	0 0 0 0	0 0 0 0	d
	CMP.W @(d:16,ERs.L),@(d:32,Rd.B)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 0 1 0	0 0 0 0	0 0 0 0	d d
	CMP.W @(d:16,ERs.L),@(d:32,Rd.W)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 0 1 0	0 0 0 0	0 0 0 0	d d
	CMP.W @(d:16,ERs.L),@(d:32,ERd.L)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 0 1 0	0 0 0 0	0 0 0 0	d d
	CMP.W @(d:16,ERs.L),@aa:16	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	0 0 0 0	a
	CMP.W @(d:16,ERs.L),@aa:32	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	0 0 0 0	a a
	CMP.W @(d:32,Rs.B),@ERd	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 0 1 0	0 0 0 0	0 0 0 0	
	CMP.W @(d:32,Rs.B),@ERd+	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 0 1 0	0 0 0 0	0 0 0 0	
	CMP.W @(d:32,Rs.B),@ERd-	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 0 1 0	0 0 0 0	0 0 0 0	
	CMP.W @(d:32,Rs.B),@+ERd	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 0 1 0	0 0 0 0	0 0 0 0	
	CMP.W @(d:32,Rs.B),@-ERd	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 0 1 0	0 0 0 0	0 0 0 0	
	CMP.W @(d:32,Rs.B),@(d:2,ERd)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 0 1 0	0 0 0 0	0 0 0 0	
	CMP.W @(d:32,Rs.B),@(d:16,ERd)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 0 1 0	0 0 0 0	0 0 0 0	d
	CMP.W @(d:32,Rs.B),@(d:32,ERd)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 0 1 0	0 0 0 0	0 0 0 0	d d
	CMP.W @(d:32,Rs.B),@(d:16,Rd.B)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 0 1 0	0 0 0 0	0 0 0 0	d
	CMP.W @(d:32,Rs.B),@(d:16,Rd.W)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 0 1 0	0 0 0 0	0 0 0 0	d
	CMP.W @(d:32,Rs.B),@(d:16,ERd.L)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 0 1 0	0 0 0 0	0 0 0 0	d
	CMP.W @(d:32,Rs.B),@(d:32,Rd.B)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 0 1 0	0 0 0 0	0 0 0 0	d d



Instruction	Mnemonic	Opcode						EA Ex- tension	Opcode						EA Extension																									
		15	8	7	0				15	8	7	0																												
CMP	CMP.W @(d:32.Rs.B),@(d:32.Rd.W)	7	8	0	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	0	1	rd	0	0	1	0	0	0	0	0	0	d	d		
	CMP.W @(d:32.Rs.B),@(d:32.ERd.L)	7	8	0	rs	0	1	0	1	6	B	0	0	1	0	1	1	1	1	1	rd	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	d	d	
	CMP.W @(d:32.Rs.B),@aa:16	7	8	0	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	0	rd	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	a	
	CMP.W @(d:32.Rs.B),@aa:32	7	8	0	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	0	rd	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	a	a
	CMP.W @(d:32.Rs.W),@ERd	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	0	rd	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	CMP.W @(d:32.Rs.W),@ERd+	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	0	rd	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
	CMP.W @(d:32.Rs.W),@ERd-	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	0	rd	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0			
	CMP.W @(d:32.Rs.W),@+ERd	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	0	rd	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0			
	CMP.W @(d:32.Rs.W),@-ERd	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	0	rd	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0			
	CMP.W @(d:32.Rs.W),@(d:2.ERd)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	0	rd	0	0	d	d	0	0	0	0	0	0	0	0	0	0	0				
	CMP.W @(d:32.Rs.W),@(d:16.ERd)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	0	rd	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	d			
	CMP.W @(d:32.Rs.W),@(d:32.ERd)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	0	rd	1	1	0	0	1	0	0	0	0	0	0	0	0	0	d	d			
	CMP.W @(d:32.Rs.W),@(d:16.Rd.B)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	0	rd	1	1	0	1	0	0	0	0	0	0	0	0	0	d					
	CMP.W @(d:32.Rs.W),@(d:16.Rd.W)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	0	rd	1	1	1	0	0	0	0	0	0	0	0	0	0	d					
	CMP.W @(d:32.Rs.W),@(d:16.ERd.L)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	0	rd	1	1	1	1	0	0	0	0	0	0	0	0	0	d					
	CMP.W @(d:32.Rs.W),@(d:32.Rd.B)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	0	rd	1	1	0	1	1	0	0	0	0	0	0	0	0	d	d				
	CMP.W @(d:32.Rs.W),@(d:32.Rd.W)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	0	rd	1	1	1	0	1	0	0	0	0	0	0	0	d	d					
	CMP.W @(d:32.Rs.W),@(d:32.ERd.L)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	0	rd	1	1	1	1	1	0	0	0	0	0	0	0	d	d					
	CMP.W @(d:32.Rs.W),@aa:16	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	0	rd	0	1	0	0	0	0	0	0	0	0	0	0	0	a					
	CMP.W @(d:32.Rs.W),@aa:32	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	0	rd	0	1	0	0	1	0	0	0	0	0	0	0	0	a	a				
	CMP.W @(d:32.ERs.L),@ERd	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	0	rd	0	0	0	0	0	0	0	0	0	0	0	0	0						
	CMP.W @(d:32.ERs.L),@ERd+	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	0	rd	1	0	0	0	0	0	0	0	0	0	0	0	0						
	CMP.W @(d:32.ERs.L),@ERd-	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	0	rd	1	0	1	0	0	0	0	0	0	0	0	0	0						
	CMP.W @(d:32.ERs.L),@+ERd	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	0	rd	1	0	0	1	0	0	0	0	0	0	0	0	0						
	CMP.W @(d:32.ERs.L),@-ERd	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	0	rd	1	0	1	1	0	0	0	0	0	0	0	0							
	CMP.W @(d:32.ERs.L),@(d:2.ERd)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	0	rd	0	0	d	d	0	0	0	0	0	0	0	0							
	CMP.W @(d:32.ERs.L),@(d:16.ERd)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	0	rd	1	1	0	0	0	0	0	0	0	0	0	0	d						
	CMP.W @(d:32.ERs.L),@(d:32.ERd)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	0	rd	1	1	0	0	1	0	0	0	0	0	0	d	d						
	CMP.W @(d:32.ERs.L),@(d:16.Rd.B)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	0	rd	1	1	0	1	0	0	0	0	0	0	d								
	CMP.W @(d:32.ERs.L),@(d:16.Rd.W)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	0	rd	1	1	1	0	0	0	0	0	0	0	d								
	CMP.W @(d:32.ERs.L),@(d:16.Rd.W)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	0	rd	1	1	1	0	0	0	0	0	0	0	d								
	CMP.W @(d:32.ERs.L),@(d:16.ERd.L)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	0	rd	1	1	1	1	0	0	0	0	0	0	d								
CMP.W @(d:32.ERs.L),@(d:32.Rd.B)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	0	rd	1	1	0	1	1	0	0	0	0	0	d	d								
CMP.W @(d:32.ERs.L),@(d:32.Rd.W)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	0	rd	1	1	1	0	1	0	0	0	0	d	d									

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension						
		15	8	7	0		15	8	7	0							
CMP	CMP.W @(d:32,ERs.L),@(d:32,ERd.L)	7	8	0	rs 0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d	
	CMP.W @aa:16 @(d:32,ERs.L),@aa:16	7	8	0	rs 0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	a	
	CMP.W @aa:32 @(d:32,ERs.L),@aa:32	7	8	0	rs 0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	a a	
	CMP.W @aa:16, @ERd					6	B	0 0 0 1	0 1 0 1	a		0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @aa:16, @ERd+					6	B	0 0 0 1	0 1 0 1	a		1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @aa:16, @ERd-					6	B	0 0 0 1	0 1 0 1	a		1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @aa:16, @+ERd					6	B	0 0 0 1	0 1 0 1	a		1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @aa:16, @-ERd					6	B	0 0 0 1	0 1 0 1	a		1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @aa:16, @(rd:2,ERd)					6	B	0 0 0 1	0 1 0 1	a		0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @aa:16, @(rd:16,ERd)					6	B	0 0 0 1	0 1 0 1	a		1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @aa:16, @(d:32,ERd)					6	B	0 0 0 1	0 1 0 1	a		1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @aa:16, @(d:16,Rd,B)					6	B	0 0 0 1	0 1 0 1	a		1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @aa:16, @(d:16,Rd,W)					6	B	0 0 0 1	0 1 0 1	a		1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @aa:16, @(d:16,ERd,L)					6	B	0 0 0 1	0 1 0 1	a		1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @aa:16, @(d:32,Rd,B)					6	B	0 0 0 1	0 1 0 1	a		1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @aa:16, @(d:32,Rd,W)					6	B	0 0 0 1	0 1 0 1	a		1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @aa:16, @(d:32,ERd,L)					6	B	0 0 0 1	0 1 0 1	a		1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @aa:16, @aa:16					6	B	0 0 0 1	0 1 0 1	a		0 1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	a
	CMP.W @aa:16, @aa:32					6	B	0 0 0 1	0 1 0 1	a		0 1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	a a
	CMP.W @aa:32, @ERd					6	B	0 0 1 1	0 1 0 1	a a		0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @aa:32, @ERd+					6	B	0 0 1 1	0 1 0 1	a a		1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @aa:32, @ERd-					6	B	0 0 1 1	0 1 0 1	a a		1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @aa:32, @+ERd					6	B	0 0 1 1	0 1 0 1	a a		1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @aa:32, @-ERd					6	B	0 0 1 1	0 1 0 1	a a		1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @aa:32, @(rd:2,ERd)					6	B	0 0 1 1	0 1 0 1	a a		0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.W @aa:32, @(rd:16,ERd)					6	B	0 0 1 1	0 1 0 1	a a		1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @aa:32, @(d:32,ERd)					6	B	0 0 1 1	0 1 0 1	a a		1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @aa:32, @(d:16,Rd,B)					6	B	0 0 1 1	0 1 0 1	a a		1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @aa:32, @(d:16,Rd,W)					6	B	0 0 1 1	0 1 0 1	a a		1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @aa:32, @(d:16,ERd,L)					6	B	0 0 1 1	0 1 0 1	a a		1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.W @aa:32, @(d:32,Rd,B)					6	B	0 0 1 1	0 1 0 1	a a		1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.W @aa:32, @(d:32,Rd,W)					6	B	0 0 1 1	0 1 0 1	a a		1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
CMP.W @aa:32, @(d:32,ERd,L)					6	B	0 0 1 1	0 1 0 1	a a		1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d	



Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension			
		15	8	7	0		15	8	7	0				
CMP	CMP.W @aa:32, @aa:16		6	B	0 0 1 1	0 1 0 1	a a	0 1 0 0	0 0 0 0	0 0 1 0	0 0 0 0	a		
	CMP.W @aa:32, @aa:32		6	B	0 0 1 1	0 1 0 1	a a	0 1 0 0	1 0 0 0	0 0 1 0	0 0 0 0	a a		
	CMP.L #xx:3, ERd							1	F	1 x x x	1 1	rd		
	CMP.L #xx:16, ERd							7	A	0 0 1 0	1	rd	x	
	CMP.L #xx:32, ERd							7	A	0 0 1 0	0 0	rd	x x	
	CMP.L #xx:16, @ERd		0	1	0	1 1 1 0		0 0 0 0	0	rd	0 0 1 0	0 0 0 0		x
	CMP.L #xx:16, @ERd+		0	1	0	1 1 1 0		1 0 0 0	0	rd	0 0 1 0	0 0 0 0		x
	CMP.L #xx:16, @ERd-		0	1	0	1 1 1 0		1 0 1 0	0	rd	0 0 1 0	0 0 0 0		x
	CMP.L #xx:16, @+ERd		0	1	0	1 1 1 0		1 0 0 1	0	rd	0 0 1 0	0 0 0 0		x
	CMP.L #xx:16, @-ERd		0	1	0	1 1 1 0		1 0 1 1	0	rd	0 0 1 0	0 0 0 0		x
	CMP.L #xx:16, @(d:2, ERd)		0	1	0	1 1 1 0		0 0 d d	0	rd	0 0 1 0	0 0 0 0		x
	CMP.L #xx:16, @(d:16, ERd)		0	1	0	1 1 1 0		1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d	x
	CMP.L #xx:16, @(d:32, ERd)		0	1	0	1 1 1 0		1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d	x
	CMP.L #xx:16, @(d:16, Rd.B)		0	1	0	1 1 1 0		1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d	x
	CMP.L #xx:16, @(d:16, Rd.W)		0	1	0	1 1 1 0		1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d	x
	CMP.L #xx:16, @(d:16, ERd.L)		0	1	0	1 1 1 0		1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d	x
	CMP.L #xx:16, @(d:32, Rd.B)		0	1	0	1 1 1 0		1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d	x
	CMP.L #xx:16, @(d:32, Rd.W)		0	1	0	1 1 1 0		1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d	x
	CMP.L #xx:16, @(d:32, ERd.L)		0	1	0	1 1 1 0		1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d	x
	CMP.L #xx:16, @aa:16		0	1	0	1 1 1 0		0 1 0 0	0 0 0 0	0 0 1 0	0 0 0 0	a	x	
	CMP.L #xx:16, @aa:32		0	1	0	1 1 1 0		0 1 0 0	1 0 0 0	0 0 1 0	0 0 0 0	a a	x	
	CMP.L #xx:32, @ERd		0	1	0	1 1 1 0		0 0 0 0	0	rd	0 0 1 0	1 0 0 0		x x
	CMP.L #xx:32, @ERd+		0	1	0	1 1 1 0		1 0 0 0	0	rd	0 0 1 0	1 0 0 0		x x
	CMP.L #xx:32, @ERd-		0	1	0	1 1 1 0		1 0 1 0	0	rd	0 0 1 0	1 0 0 0		x x
	CMP.L #xx:32, @+ERd		0	1	0	1 1 1 0		1 0 0 1	0	rd	0 0 1 0	1 0 0 0		x x
	CMP.L #xx:32, @-ERd		0	1	0	1 1 1 0		1 0 1 1	0	rd	0 0 1 0	1 0 0 0		x x
CMP.L #xx:32, @(d:2, ERd)		0	1	0	1 1 1 0		0 0 d d	0	rd	0 0 1 0	1 0 0 0		x x	
CMP.L #xx:32, @(d:16, ERd)		0	1	0	1 1 1 0		1 1 0 0	0	rd	0 0 1 0	1 0 0 0	d	x x	
CMP.L #xx:32, @(d:32, ERd)		0	1	0	1 1 1 0		1 1 0 0	1	rd	0 0 1 0	1 0 0 0	d d	x x	
CMP.L #xx:32, @(d:16, Rd.B)		0	1	0	1 1 1 0		1 1 0 1	0	rd	0 0 1 0	1 0 0 0	d	x x	
CMP.L #xx:32, @(d:16, Rd.W)		0	1	0	1 1 1 0		1 1 1 0	0	rd	0 0 1 0	1 0 0 0	d	x x	
CMP.L #xx:32, @(d:16, ERd.L)		0	1	0	1 1 1 0		1 1 1 1	0	rd	0 0 1 0	1 0 0 0	d	x x	
CMP.L #xx:32, @(d:32, Rd.B)		0	1	0	1 1 1 0		1 1 0 1	1	rd	0 0 1 0	1 0 0 0	d d	x x	
CMP.L #xx:32, @(d:32, Rd.W)		0	1	0	1 1 1 0		1 1 1 0	1	rd	0 0 1 0	1 0 0 0	d d	x x	

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension				
		15	8	7	0		15	8	7	0					
CMP	CMP.L #xx:32,@(d:32,ERd.L)	0	1	0	1 1 1 0		1 1 1 1	1	rd	0 0 1 0	1 0 0 0	d	d	x	x
	CMP.L #xx:32,@aa:16	0	1	0	1 1 1 0		0 1 0 0	0	0 0 0 0	0 0 1 0	1 0 0 0	a		x	x
	CMP.L #xx:32,@aa:32	0	1	0	1 1 1 0		0 1 0 0	1	0 0 0 0	0 0 1 0	1 0 0 0	a	a	x	x
	CMP.L ERs,ERd						1	F	rs	0	rd				
	CMP.L ERs,@ERd	0	1	0	1 0 0 1		0 0 0 0	0	rd	0 0 1 0	0 0	rs			
	CMP.L ERs,@ERd+	0	1	0	1 0 0 1		1 0 0 0	0	rd	0 0 1 0	0	rs			
	CMP.L ERs,@ERd-	0	1	0	1 0 0 1		1 0 1 0	0	rd	0 0 1 0	0	rs			
	CMP.L ERs,@+ERd	0	1	0	1 0 0 1		1 0 0 1	0	rd	0 0 1 0	0	rs			
	CMP.L ERs,@-ERd	0	1	0	1 0 0 1		1 0 1 1	0	rd	0 0 1 0	0	rs			
	CMP.L ERs,@(d:2,ERd)	0	1	0	1 0 0 1		0 0 d	d	rd	0 0 1 0	0	rs			
	CMP.L ERs,@(d:16,ERd)	0	1	0	1 0 0 1		1 1 0 0	0	rd	0 0 1 0	0	rs	d		
	CMP.L ERs,@(d:32,ERd)	0	1	0	1 0 0 1		1 1 0 0	1	rd	0 0 1 0	0	rs	d	d	
	CMP.L ERs,@(d:16,Rd,B)	0	1	0	1 0 0 1		1 1 0 1	0	rd	0 0 1 0	0	rs	d		
	CMP.L ERs,@(d:16,Rd,W)	0	1	0	1 0 0 1		1 1 1 0	0	rd	0 0 1 0	0	rs	d		
	CMP.L ERs,@(d:16,ERd,L)	0	1	0	1 0 0 1		1 1 1 1	0	rd	0 0 1 0	0	rs	d		
	CMP.L ERs,@(d:32,Rd,B)	0	1	0	1 0 0 1		1 1 0 1	1	rd	0 0 1 0	0	rs	d	d	
	CMP.L ERs,@(d:32,Rd,W)	0	1	0	1 0 0 1		1 1 1 0	1	rd	0 0 1 0	0	rs	d	d	
	CMP.L ERs,@(d:32,ERd,L)	0	1	0	1 0 0 1		1 1 1 1	1	rd	0 0 1 0	0	rs	d	d	
	CMP.L ERs,@aa:16	0	1	0	1 0 0 1		0 1 0 0	0	0 0 0 0	0 0 1 0	0	rs	a		
	CMP.L ERs,@aa:32	0	1	0	1 0 0 1		0 1 0 0	1	0 0 0 0	0 0 1 0	0	rs	a	a	
	CMP.L @ERs,ERd	0	1	0	1 0 1 0		0 0 0 0	0	rs	0 0 1 0	0	rd			
	CMP.L @ERs+,ERd	0	1	0	1 0 1 0		1 0 0 0	0	rs	0 0 1 0	0	rd			
	CMP.L @ERs-,ERd	0	1	0	1 0 1 0		1 0 1 0	0	rs	0 0 1 0	0	rd			
	CMP.L @+ERs,ERd	0	1	0	1 0 1 0		1 0 0 1	0	rs	0 0 1 0	0	rd			
	CMP.L @-ERs,ERd	0	1	0	1 0 1 0		1 0 1 1	0	rs	0 0 1 0	0	rd			
	CMP.L @(d:2,ERs),ERd	0	1	0	1 0 1 0		0 0 d	d	rs	0 0 1 0	0	rd			
	CMP.L @(d:16,ERs),ERd	0	1	0	1 0 1 0		1 1 0 0	0	rs	0 0 1 0	0	rd	d		
	CMP.L @(d:32,ERs),ERd	0	1	0	1 0 1 0		1 1 0 0	1	rs	0 0 1 0	0	rd	d	d	
	CMP.L @(d:16,Rs,B),ERd	0	1	0	1 0 1 0		1 1 0 1	0	rs	0 0 1 0	0	rd	d		
	CMP.L @(d:16,Rs,W),ERd	0	1	0	1 0 1 0		1 1 1 0	0	rs	0 0 1 0	0	rd	d		
	CMP.L @(d:16,ERs,L),ERd	0	1	0	1 0 1 0		1 1 1 1	0	rs	0 0 1 0	0	rd	d		
	CMP.L @(d:32,Rs,B),ERd	0	1	0	1 0 1 0		1 1 0 1	1	rs	0 0 1 0	0	rd	d	d	
	CMP.L @(d:32,Rs,W),ERd	0	1	0	1 0 1 0		1 1 1 0	1	rs	0 0 1 0	0	rd	d	d	
	CMP.L @(d:32,ERs,L),ERd	0	1	0	1 0 1 0		1 1 1 1	1	rs	0 0 1 0	0	rd	d	d	



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension																						
		15	8	7	0	15	8	7	0		15	8	7	0																							
CMP	CMP.L @aa:16,ERd				0		1		0		1	0	1	0	0	1	0	0	rd	a																	
	CMP.L @aa:32,ERd				0		1		0		1	0	1	0	0	1	0	0	rd	a	a																
	CMP.L @ERs,@ERd	0		1		0		0	1	0	0	6		9		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	rd		
	CMP.L @ERs,@ERd+	0		1		0		0	1	0	0	6		9		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	rd			
	CMP.L @ERs,@ERd-	0		1		0		0	1	0	0	6		9		0		rs	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	rd			
	CMP.L @ERs,@+ERd	0		1		0		0	1	0	0	6		9		0		rs	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	rd			
	CMP.L @ERs,@-ERd	0		1		0		0	1	0	0	6		9		0		rs	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	rd			
	CMP.L @ERs,@(d:2,ERd)	0		1		0		0	1	0	0	6		9		0		rs	1	1	0	0	d	d	0	0	0	0	0	0	0	0	rd				
	CMP.L @ERs,@(d:16,ERd)	0		1		0		0	1	0	0	6		9		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	0	rd		d		
	CMP.L @ERs,@(d:32,ERd)	0		1		0		0	1	0	0	6		9		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	0	rd		d	d	
	CMP.L @ERs,@(d:16,Rd,B)	0		1		0		0	1	0	0	6		9		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	0	rd				
	CMP.L @ERs,@(d:16,Rd,W)	0		1		0		0	1	0	0	6		9		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	0	rd				
	CMP.L @ERs,@(d:16,ERd,L)	0		1		0		0	1	0	0	6		9		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	0	rd				
	CMP.L @ERs,@(d:32,Rd,B)	0		1		0		0	1	0	0	6		9		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	0	rd		d	d	
	CMP.L @ERs,@(d:32,Rd,W)	0		1		0		0	1	0	0	6		9		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	0	rd		d	d	
	CMP.L @ERs,@(d:32,ERd,L)	0		1		0		0	1	0	0	6		9		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	0	rd		d	d	
	CMP.L @ERs,@aa:16	0		1		0		0	1	0	0	6		9		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	0	rd				
	CMP.L @ERs,@aa:32	0		1		0		0	1	0	0	6		9		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	0	rd		a	a	
	CMP.L @ERs+,@ERd	0		1		0		0	1	0	0	6		D		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	rd					
	CMP.L @ERs+,@ERd+	0		1		0		0	1	0	0	6		D		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	rd					
	CMP.L @ERs+,@ERd-	0		1		0		0	1	0	0	6		D		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	rd					
	CMP.L @ERs+,@+ERd	0		1		0		0	1	0	0	6		D		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	rd					
	CMP.L @ERs+,@-ERd	0		1		0		0	1	0	0	6		D		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	rd					
	CMP.L @ERs+,@(d:2,ERd)	0		1		0		0	1	0	0	6		D		0		rs	1	1	0	0	0	d	d	0	0	0	0	0	0	rd					
CMP.L @ERs+,@(d:16,ERd)	0		1		0		0	1	0	0	6		D		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	rd		d				
CMP.L @ERs+,@(d:32,ERd)	0		1		0		0	1	0	0	6		D		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	rd		d	d			
CMP.L @ERs+,@(d:16,Rd,B)	0		1		0		0	1	0	0	6		D		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	rd						
CMP.L @ERs+,@(d:16,Rd,W)	0		1		0		0	1	0	0	6		D		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	rd						
CMP.L @ERs+,@(d:16,ERd,L)	0		1		0		0	1	0	0	6		D		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	rd						
CMP.L @ERs+,@(d:32,Rd,B)	0		1		0		0	1	0	0	6		D		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	rd		d	d			
CMP.L @ERs+,@(d:32,Rd,W)	0		1		0		0	1	0	0	6		D		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	rd		d	d			
CMP.L @ERs+,@(d:32,ERd,L)	0		1		0		0	1	0	0	6		D		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	rd		d	d			
CMP.L @ERs+,@aa:16	0		1		0		0	1	0	0	6		D		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	rd		a				
CMP.L @ERs+,@aa:32	0		1		0		0	1	0	0	6		D		0		rs	1	1	0	0	0	0	0	0	0	0	0	0	0	rd		a	a			

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
CMP	CMP.L @ERs, @ERd	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @ERs, @ERd+	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @ERs, @ERd-	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @ERs, @+ERd	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @ERs, @-ERd	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @ERs, @(d:2,ERd)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @ERs, @(d:16,ERd)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @ERs, @(d:32,ERd)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @ERs, @(d:16,Rd,B)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @ERs, @(d:16,Rd,W)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @ERs, @(d:16,ERd,L)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @ERs, @(d:32,Rd,B)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @ERs, @(d:32,Rd,W)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @ERs, @(d:32,ERd,L)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @ERs, @aa:16	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		0 1 0 0	0 0 0 0	0 0 1 0	0 0 0 0	a	
	CMP.L @ERs, @aa:32	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		0 1 0 0	1 0 0 0	0 0 1 0	0 0 0 0	a a	
	CMP.L @+ERs, @ERd	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @+ERs, @ERd+	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @+ERs, @ERd-	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @+ERs, @+ERd	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @+ERs, @-ERd	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @+ERs, @(d:2,ERd)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @+ERs, @(d:16,ERd)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @+ERs, @(d:32,ERd)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @+ERs, @(d:16,Rd,B)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @+ERs, @(d:16,Rd,W)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @+ERs, @(d:16,ERd,L)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @+ERs, @(d:32,Rd,B)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @+ERs, @(d:32,Rd,W)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @+ERs, @(d:32,ERd,L)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @+ERs, @aa:16	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		0 1 0 0	0 0 0 0	0 0 1 0	0 0 0 0	a	
	CMP.L @+ERs, @aa:32	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		0 1 0 0	1 0 0 0	0 0 1 0	0 0 0 0	a a	
CMP.L @-ERs, @ERd	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		0 0 0 0	0	rd	0 0 1 0	0 0 0 0		
CMP.L @-ERs, @ERd+	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 0 0	0	rd	0 0 1 0	0 0 0 0		



Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
CMP	CMP.L @-ERs,@ERd-	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @-ERs,@+ERd	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @-ERs,@-ERd	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @-ERs,@(d:2,ERd)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @-ERs,@(d:16,ERd)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @-ERs,@(d:32,ERd)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @-ERs,@(d:16,Rd,B)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @-ERs,@(d:16,Rd,W)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @-ERs,@(d:16,ERd,L)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @-ERs,@(d:32,Rd,B)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @-ERs,@(d:32,Rd,W)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @-ERs,@(d:32,ERd,L)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @-ERs,@aa:16	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		0 1 0 0	0 0 0 0	0 0 1 0	0 0 0 0	a	
	CMP.L @-ERs,@aa:32	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		0 1 0 0	1 0 0 0	0 0 1 0	0 0 0 0	a a	
	CMP.L @(d:2,ERs),@ERd	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:2,ERs),@ERd+	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:2,ERs),@ERd-	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:2,ERs),@+ERd	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:2,ERs),@-ERd	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:2,ERs),@(d:2,ERd)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:2,ERs),@(d:16,ERd)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:2,ERs),@(d:32,ERd)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:2,ERs),@(d:16,Rd,B)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:2,ERs),@(d:16,Rd,W)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:2,ERs),@(d:16,ERd,L)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:2,ERs),@(d:32,Rd,B)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:2,ERs),@(d:32,Rd,W)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:2,ERs),@(d:32,ERd,L)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:2,ERs),@aa:16	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		0 1 0 0	0 0 0 0	0 0 1 0	0 0 0 0	a	
	CMP.L @(d:2,ERs),@aa:32	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		0 1 0 0	1 0 0 0	0 0 1 0	0 0 0 0	a a	
CMP.L @(d:16,ERs),@ERd	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 0 1 0	0 0 0 0		
CMP.L @(d:16,ERs),@ERd+	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 0 1 0	0 0 0 0		
CMP.L @(d:16,ERs),@ERd-	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 0 1 0	0 0 0 0		
CMP.L @(d:16,ERs),@+ERd	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 0 1 0	0 0 0 0		

Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension				
		15	8	7	0	15	8	7	0		15	8	7	0					
CMP	CMP.L @(d:16,ERs),@-ERd	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	0 1 1	0	rd	0 0 1 0	0 0 0 0		
	CMP.L @(d:16,ERs),@(d:2,ERd)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	0	0 d d	0	rd	0 0 1 0	0 0 0 0		
	CMP.L @(d:16,ERs),@(d:16,ERd)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 0 0	0	rd	0 0 1 0	0 0 0 0	d	
	CMP.L @(d:16,ERs),@(d:32,ERd)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 0 0	1	rd	0 0 1 0	0 0 0 0	d d	
	CMP.L @(d:16,ERs),@(d:16,Rd,B)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 0 1	0	rd	0 0 1 0	0 0 0 0	d	
	CMP.L @(d:16,ERs),@(d:16,Rd,W)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 1 0	0	rd	0 0 1 0	0 0 0 0	d	
	CMP.L @(d:16,ERs),@(d:16,ERdL)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 1 1	0	rd	0 0 1 0	0 0 0 0	d	
	CMP.L @(d:16,ERs),@(d:32,Rd,B)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 0 1	1	rd	0 0 1 0	0 0 0 0	d d	
	CMP.L @(d:16,ERs),@(d:32,Rd,W)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 1 0	1	rd	0 0 1 0	0 0 0 0	d d	
	CMP.L @(d:16,ERs),@(d:32,ERdL)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 1 1	1	rd	0 0 1 0	0 0 0 0	d d	
	CMP.L @(d:16,ERs),@aa:16	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	0	1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	a	
	CMP.L @(d:16,ERs),@aa:32	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	0	1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	a a	
	CMP.L @(d:32,ERs),@ERd	7	8	1	rs	0 1 0 0	6	B	0	0 1 0	1 1 0 0	d d	0	0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32,ERs),@ERd+	7	8	1	rs	0 1 0 0	6	B	0	0 1 0	1 1 0 0	d d	1	0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32,ERs),@ERd-	7	8	1	rs	0 1 0 0	6	B	0	0 1 0	1 1 0 0	d d	1	0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32,ERs),@+ERd	7	8	1	rs	0 1 0 0	6	B	0	0 1 0	1 1 0 0	d d	1	0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32,ERs),@-ERd	7	8	1	rs	0 1 0 0	6	B	0	0 1 0	1 1 0 0	d d	1	0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32,ERs),@(d:2,ERd)	7	8	1	rs	0 1 0 0	6	B	0	0 1 0	1 1 0 0	d d	0	0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32,ERs),@(d:16,ERd)	7	8	1	rs	0 1 0 0	6	B	0	0 1 0	1 1 0 0	d d	1	1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:32,ERs),@(d:32,ERd)	7	8	1	rs	0 1 0 0	6	B	0	0 1 0	1 1 0 0	d d	1	1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:32,ERs),@(d:16,Rd,B)	7	8	1	rs	0 1 0 0	6	B	0	0 1 0	1 1 0 0	d d	1	1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:32,ERs),@(d:16,Rd,W)	7	8	1	rs	0 1 0 0	6	B	0	0 1 0	1 1 0 0	d d	1	1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:32,ERs),@(d:16,ERdL)	7	8	1	rs	0 1 0 0	6	B	0	0 1 0	1 1 0 0	d d	1	1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:32,ERs),@(d:32,Rd,B)	7	8	1	rs	0 1 0 0	6	B	0	0 1 0	1 1 0 0	d d	1	1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:32,ERs),@(d:32,Rd,W)	7	8	1	rs	0 1 0 0	6	B	0	0 1 0	1 1 0 0	d d	1	1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:32,ERs),@(d:32,ERdL)	7	8	1	rs	0 1 0 0	6	B	0	0 1 0	1 1 0 0	d d	1	1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:32,ERs),@aa:16	7	8	1	rs	0 1 0 0	6	B	0	0 1 0	1 1 0 0	d d	0	1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	a
	CMP.L @(d:32,ERs),@aa:32	7	8	1	rs	0 1 0 0	6	B	0	0 1 0	1 1 0 0	d d	0	1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	a a
	CMP.L @(d:16,Rs,B),@ERd	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	0	0 0 0	0	rd	0 0 1 0	0 0 0 0		
	CMP.L @(d:16,Rs,B),@ERd+	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1	0 0 0	0	rd	0 0 1 0	0 0 0 0		
CMP.L @(d:16,Rs,B),@ERd-	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1	0 1 0	0	rd	0 0 1 0	0 0 0 0			
CMP.L @(d:16,Rs,B),@+ERd	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1	0 0 1	0	rd	0 0 1 0	0 0 0 0			
CMP.L @(d:16,Rs,B),@-ERd	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1	0 1 1	0	rd	0 0 1 0	0 0 0 0			
CMP.L @(d:16,Rs,B),@(d:2,ERd)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	0	0 d d	0	rd	0 0 1 0	0 0 0 0			



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
CMP	CMP.L @(d:16,Rs.B),@(d:16,ERd)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:16,Rs.B),@(d:32,ERd)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:16,Rs.B),@(d:16,Rd.B)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:16,Rs.B),@(d:16,Rd.W)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:16,Rs.B),@(d:16,ERd.L)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:16,Rs.B),@(d:32,Rd.B)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:16,Rs.B),@(d:32,Rd.W)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:16,Rs.B),@(d:32,ERd.L)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:16,Rs.B),@aa:16	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 0 1 0	0 0 0 0	a	
	CMP.L @(d:16,Rs.B),@aa:32	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 0 1 0	0 0 0 0	a a	
	CMP.L @(d:16,Rs.W),@ERd	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:16,Rs.W),@ERd+	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:16,Rs.W),@ERd-	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:16,Rs.W),@+ERd	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:16,Rs.W),@-ERd	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:16,Rs.W),@(d:2,ERd)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:16,Rs.W),@(d:16,ERd)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:16,Rs.W),@(d:32,ERd)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:16,Rs.W),@(d:16,Rd.B)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:16,Rs.W),@(d:16,Rd.W)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
CMP.L @(d:16,Rs.W),@(d:16,ERd.L)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d	
CMP.L @(d:16,Rs.W),@(d:32,Rd.B)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d	
CMP.L @(d:16,Rs.W),@(d:32,Rd.W)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d	
CMP.L @(d:16,Rs.W),@(d:32,ERd.L)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d	
CMP.L @(d:16,Rs.W),@aa:16	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 0 1 0	0 0 0 0	a		
CMP.L @(d:16,Rs.W),@aa:32	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 0 1 0	0 0 0 0	a a		
CMP.L @(d:16,ERs.L),@ERd	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 0 1 0	0 0 0 0		
CMP.L @(d:16,ERs.L),@ERd+	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 0 1 0	0 0 0 0		
CMP.L @(d:16,ERs.L),@ERd-	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 0 1 0	0 0 0 0		
CMP.L @(d:16,ERs.L),@+ERd	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 0 1 0	0 0 0 0		
CMP.L @(d:16,ERs.L),@-ERd	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 0 1 0	0 0 0 0		
CMP.L @(d:16,ERs.L),@(d:2,ERd)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 0 1 0	0 0 0 0		
CMP.L @(d:16,ERs.L),@(d:16,ERd)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d	
CMP.L @(d:16,ERs.L),@(d:32,ERd)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d	

Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
CMP	CMP.L @(d:16,ERs.L),@(d:16,Rd,B)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:16,ERs.L),@(d:16,Rd,W)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:16,ERs.L),@(d:16,ERd,L)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:16,ERs.L),@(d:32,Rd,B)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:16,ERs.L),@(d:32,Rd,W)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:16,ERs.L),@(d:32,ERd,L)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:16,ERs.L),@aa:16	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	a
	CMP.L @(d:16,ERs.L),@aa:32	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	a a
	CMP.L @(d:32,Rs,B),@ERd	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32,Rs,B),@ERd+	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32,Rs,B),@ERd-	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32,Rs,B),@+ERd	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32,Rs,B),@-ERd	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32,Rs,B),@(d:2,ERd)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32,Rs,B),@(d:16,ERd)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:32,Rs,B),@(d:32,ERd)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:32,Rs,B),@(d:16,Rd,B)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:32,Rs,B),@(d:16,Rd,W)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:32,Rs,B),@(d:16,ERd,L)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:32,Rs,B),@(d:32,Rd,B)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:32,Rs,B),@(d:32,Rd,W)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:32,Rs,B),@(d:32,ERd,L)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:32,Rs,B),@aa:16	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	a
	CMP.L @(d:32,Rs,B),@aa:32	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	a a
	CMP.L @(d:32,Rs,W),@ERd	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32,Rs,W),@ERd+	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32,Rs,W),@ERd-	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32,Rs,W),@+ERd	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32,Rs,W),@-ERd	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32,Rs,W),@(d:2,ERd)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32,Rs,W),@(d:16,ERd)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:32,Rs,W),@(d:32,ERd)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
CMP.L @(d:32,Rs,W),@(d:16,Rd,B)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d	
CMP.L @(d:32,Rs,W),@(d:16,Rd,W)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d	

Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension		
		15	8	7	0	15	8		7	0	15	8	7	0			
CMP	CMP.L @(d:32.Rs.W),@(d:16.ERd.L)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:32.Rs.W),@(d:32.Rd.B)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:32.Rs.W),@(d:32.Rd.W)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:32.Rs.W),@(d:32.ERd.L)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:32.Rs.W),@aa:16	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	a
	CMP.L @(d:32.Rs.W),@aa:32	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	a a
	CMP.L @(d:32.ERs.L),@ERd	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32.ERs.L),@ERd+	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32.ERs.L),@ERd-	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32.ERs.L),@+ERd	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32.ERs.L),@-ERd	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32.ERs.L),@(d:2.ERd)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @(d:32.ERs.L),@(d:16.ERd)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:32.ERs.L),@(d:32.ERd)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:32.ERs.L),@(d:16.Rd.B)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:32.ERs.L),@(d:16.Rd.W)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:32.ERs.L),@(d:16.ERd.L)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @(d:32.ERs.L),@(d:32.Rd.B)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:32.ERs.L),@(d:32.Rd.W)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:32.ERs.L),@(d:32.ERd.L)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @(d:32.ERs.L),@aa:16	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	a
	CMP.L @(d:32.ERs.L),@aa:32	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	a a
	CMP.L @aa:16,@ERd	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @aa:16,@ERd+	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @aa:16,@ERd-	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @aa:16,@+ERd	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @aa:16,@-ERd	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @aa:16,@(d:2.ERd)	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 0 d d	0	rd	0 0 1 0	0 0 0 0	
CMP.L @aa:16,@(d:16.ERd)	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d	
CMP.L @aa:16,@(d:32.ERd)	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d	
CMP.L @aa:16,@(d:16.Rd.B)	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d	
CMP.L @aa:16,@(d:16.Rd.W)	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d	
CMP.L @aa:16,@(d:16.ERd.L)	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d	
CMP.L @aa:16,@(d:32.Rd.B)	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d	

Instruction	Mnemonic	Opcode				15	Opcode				EA Ex- tension	Opcode				EA Extension
		15	8	7	0		15	8	7	0		15	8	7	0	
CMP	CMP.L @aa:16,@(d:32,Rd,W)	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @aa:16,@(d:32,ERd,L)	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @aa:16,@aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	a
	CMP.L @aa:16,@aa:32	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	a a
	CMP.L @aa:32,@ERd	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @aa:32,@ERd+	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 0 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @aa:32,@ERd-	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 1 0	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @aa:32,@+ERd	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 0 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @aa:32,@-ERd	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 1 1	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @aa:32,@(d:2,ERd)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 0 d d	0	rd	0 0 1 0	0 0 0 0	
	CMP.L @aa:32,@(d:16,ERd)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @aa:32,@(d:32,ERd)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @aa:32,@(d:16,Rd,B)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @aa:32,@(d:16,Rd,W)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 0	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @aa:32,@(d:16,ERd,L)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 1	0	rd	0 0 1 0	0 0 0 0	d
	CMP.L @aa:32,@(d:32,Rd,B)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @aa:32,@(d:32,Rd,W)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 0	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @aa:32,@(d:32,ERd,L)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 1	1	rd	0 0 1 0	0 0 0 0	d d
	CMP.L @aa:32,@aa:16	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 1 0 0	0	0 0 0 0	0 0 1 0	0 0 0 0	a
	CMP.L @aa:32,@aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 1 0 0	1	0 0 0 0	0 0 1 0	0 0 0 0	a a
DAA	DAA Rd											0	F	0 0 0 0	rd	
DAS	DAS Rd											1	F	0 0 0 0	rd	
DEC	DEC.B Rd											1	A	0 0 0 0	rd	
	DEC.W #1,Rd											1	B	0 1 0 1	rd	
	DEC.W #2,Rd											1	B	1 1 0 1	rd	
	DEC.L #1,ERd											1	B	0 1 1 1	0 rd	
	DEC.L #2,ERd											1	B	1 1 1 1	0 rd	
DIVS	DIVS.W #xx:4,Rd					0	1	D	0 1 1 0			5	1	x x x x	rd	
	DIVS.W Rs,Rd					0	1	D	0 0 1 0			5	1	rs	rd	
	DIVS.L #xx:4,ERd					0	1	D	0 0 1 1 0			5	3	x x x x 0	rd	
	DIVS.L ERs,ERd					0	1	D	0 0 1 0			5	3	0 rs 0	rd	
DIVU	DIVU.W #xx:4,Rd					0	1	D	1 1 1 0			5	1	x x x x	rd	
	DIVU.W Rs,Rd					0	1	D	1 0 1 0			5	1	rs	rd	
	DIVU.L #xx:4,ERd					0	1	D	1 1 1 0			5	3	x x x x 0	rd	



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension																							
		15	8	7	0	15	8	7	0		15	8	7	0																								
DIVU	DIVU.L ERs,ERd					0		1		D		1	0	1	0	5		3	0	rs	0	rd																
DIVXS	DIVXS.B #xxx-4,Rd					0		1		D		0	1	0	0	5		1		x	x	x	x	rd														
	DIVXS.B Rs,Rd					0		1		D		0	0	0	0	5		1		rs				rd														
	DIVXS.W #xxx-4,ERd					0		1		D		0	1	0	0	5		3		x	x	x	x	0	rd													
	DIVXS.W Rs,ERd					0		1		D		0	0	0	0	5		3		rs				0	rd													
DIVXU	DIVXU.B #xxx-4,Rd					0		1		D		1	1	0	0	5		1		x	x	x	x	rd														
	DIVXU.B Rs,Rd															5		1		rs				rd														
	DIVXU.W #xxx-4,ERd					0		1		D		1	1	0	0	5		3		x	x	x	x	0	rd													
	DIVXU.W Rs,ERd															5		3		rs				0	rd													
EEPMOV	EEPMOV.B					7		B		0	1	0	1	1	1	0	0	0	5		9		8		F													
	EEPMOV.W					7		B		1	1	0	1	0	1	0	1	0	0	5		9		8		F												
EXTS	EXTS.W Rd															1		7		D				rd														
	EXTS.W @ERd					7		D		1		rd	0	0	0	0	1		7		D		0	0	0	0												
	EXTS.W @ERd+	0		1		5		0	1	0	0	0	6		D		0	rd	1	0	0	0	0	1		7		D		0	0	0	0					
	EXTS.W @ERd-	0		1		5		0	1	1	0	0	6		D		0	rd	1	0	0	0	0	1		7		D		0	0	0	0					
	EXTS.W @+ERd	0		1		5		0	1	0	1	0	6		D		0	rd	1	0	0	0	0	1		7		D		0	0	0	0					
	EXTS.W @-ERd	0		1		5		0	1	1	1	0	6		D		0	rd	1	0	0	0	0	1		7		D		0	0	0	0					
	EXTS.W @(d:2,ERd)	0		1		5		0	1	d	d	6		9		0	rd	1	0	0	0	0	1		7		D		0	0	0	0						
	EXTS.W @(d:16,ERd)	0		1		5		0	1	0	0	6		F		0	rd	1	0	0	0	d	1		7		D		0	0	0	0						
	EXTS.W @(d:32,ERd)	7		8		0		rd	0	1	0	0	6		B		0	0	1	0	1	0	0	0	d	d	1		7		D		0	0	0	0		
	EXTS.W @(d:16,Rd,B)	0		1		5		0	1	0	1	6		F		0	rd	1	0	0	0	d	1		7		D		0	0	0	0						
	EXTS.W @(d:16,Rd,W)	0		1		5		0	1	1	0	6		F		0	rd	1	0	0	0	d	1		7		D		0	0	0	0						
	EXTS.W @(d:16,ERd,L)	0		1		5		0	1	1	1	6		F		0	rd	1	0	0	0	d	1		7		D		0	0	0	0						
	EXTS.W @(d:32,Rd,B)	7		8		0		rd	0	1	0	1	6		B		0	0	1	0	1	0	0	0	d	d	1		7		D		0	0	0	0		
	EXTS.W @(d:32,Rd,W)	7		8		0		rd	0	1	1	0	6		B		0	0	1	0	1	0	0	0	d	d	1		7		D		0	0	0	0		
	EXTS.W @(d:32,ERd,L)	7		8		0		rd	0	1	1	1	6		B		0	0	1	0	1	0	0	0	d	d	1		7		D		0	0	0	0		
	EXTS.W @aa:16															6		B		0	0	0	1	1	0	0	0	a	1		7		D		0	0	0	0
EXTS.W @aa:32															6		B		0	0	1	1	1	0	0	0	a	a	1		7		D		0	0	0	0
EXTS.L ERd																									1		7		F		0	rd						
EXTS.L @ERd					0		1		0		0	1	0	0	6		9		0	rd	1	0	0	0	0	1		7		F		0	0	0	0			
EXTS.L @ERd+					0		1		0		0	1	0	0	6		D		0	rd	1	0	0	0	0	1		7		F		0	0	0	0			
EXTS.L @ERd-					0		1		0		0	1	1	0	6		D		0	rd	1	0	0	0	0	1		7		F		0	0	0	0			
EXTS.L @+ERd					0		1		0		0	1	0	1	6		D		0	rd	1	0	0	0	0	1		7		F		0	0	0	0			
EXTS.L @-ERd					0		1		0		0	1	1	1	6		D		0	rd	1	0	0	0	0	1		7		F		0	0	0	0			

Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension
		15	8	7	0	15	8	7	0		15	8	7	0	
EXTS	EXTS.L @(d2,ERd)	0	1	0	0 1 d d	6	9	0	rd	1 0 0 0		1	7	F	0 0 0 0
	EXTS.L @(d:16,ERd)	0	1	0	0 1 0 0	6	F	0	rd	1 0 0 0	d	1	7	F	0 0 0 0
	EXTS.L @(d:32,ERd)	7	8	1	rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0	d d	1	7	F	0 0 0 0
	EXTS.L @(d:16,Rd,B)	0	1	0	0 1 0 1	6	F	0	rd	1 0 0 0	d	1	7	F	0 0 0 0
	EXTS.L @(d:16,Rd,W)	0	1	0	0 1 1 0	6	F	0	rd	1 0 0 0	d	1	7	F	0 0 0 0
	EXTS.L @(d:16,ERd,L)	0	1	0	0 1 1 1	6	F	0	rd	1 0 0 0	d	1	7	F	0 0 0 0
	EXTS.L @(d:32,Rd,B)	7	8	1	rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0	d d	1	7	F	0 0 0 0
	EXTS.L @(d:32,Rd,W)	7	8	1	rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0	d d	1	7	F	0 0 0 0
	EXTS.L @(d:32,ERd,L)	7	8	1	rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0	d d	1	7	F	0 0 0 0
	EXTS.L @aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0	1 0 0 0	a		1	7	F	0 0 0 0
	EXTS.L @aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0	1 0 0 0	a a		1	7	F	0 0 0 0
	EXTS.L #2,ERd											1	7	E	0 rd
	EXTS.L #2,@ERd	0	1	0	0 1 0 0	6	9	0	rd	1 0 0 0		1	7	E	0 0 0 0
	EXTS.L #2,@ERd+	0	1	0	0 1 0 0	6	D	0	rd	1 0 0 0		1	7	E	0 0 0 0
	EXTS.L #2,@ERd-	0	1	0	0 1 1 0	6	D	0	rd	1 0 0 0		1	7	E	0 0 0 0
	EXTS.L #2,@+ERd	0	1	0	0 1 0 1	6	D	0	rd	1 0 0 0		1	7	E	0 0 0 0
	EXTS.L #2,@-ERd	0	1	0	0 1 1 1	6	D	0	rd	1 0 0 0		1	7	E	0 0 0 0
	EXTS.L #2,@(d2,ERd)	0	1	0	0 1 d d	6	9	0	rd	1 0 0 0		1	7	E	0 0 0 0
	EXTS.L #2,@(d:16,ERd)	0	1	0	0 1 0 0	6	F	0	rd	1 0 0 0	d	1	7	E	0 0 0 0
	EXTS.L #2,@(d:32,ERd)	7	8	1	rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0	d d	1	7	E	0 0 0 0
EXTS.L #2,@(d:16,Rd,B)	0	1	0	0 1 0 1	6	F	0	rd	1 0 0 0	d	1	7	E	0 0 0 0	
EXTS.L #2,@(d:16,Rd,W)	0	1	0	0 1 1 0	6	F	0	rd	1 0 0 0	d	1	7	E	0 0 0 0	
EXTS.L #2,@(d:16,ERd,L)	0	1	0	0 1 1 1	6	F	0	rd	1 0 0 0	d	1	7	E	0 0 0 0	
EXTS.L #2,@(d:32,Rd,B)	7	8	1	rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0	d d	1	7	E	0 0 0 0	
EXTS.L #2,@(d:32,Rd,W)	7	8	1	rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0	d d	1	7	E	0 0 0 0	
EXTS.L #2,@(d:32,ERd,L)	7	8	1	rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0	d d	1	7	E	0 0 0 0	
EXTS.L #2,@aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0	1 0 0 0	a		1	7	E	0 0 0 0	
EXTS.L #2,@aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0	1 0 0 0	a a		1	7	E	0 0 0 0	
EXTU	EXTU.W Rd										1	7	S	rd	
	EXTU.W @ERd					7	D	1	rd	0 0 0 0		1	7	S	0 0 0 0
	EXTU.W @ERd+	0	1	5	0 1 0 0	6	D	0	rd	1 0 0 0		1	7	S	0 0 0 0
	EXTU.W @ERd-	0	1	5	0 1 1 0	6	D	0	rd	1 0 0 0		1	7	S	0 0 0 0
	EXTU.W @+ERd	0	1	5	0 1 0 1	6	D	0	rd	1 0 0 0		1	7	S	0 0 0 0
	EXTU.W @-ERd	0	1	5	0 1 1 1	6	D	0	rd	1 0 0 0		1	7	S	0 0 0 0

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
EXTU	EXTU.W @(d:2,ERd)	0	1	5	0 1 d d	6	9	0 rd	1 0 0 0		1	7	5	0 0 0 0		
	EXTU.W @(d:16,ERd)	0	1	5	0 1 0 0	6	F	0 rd	1 0 0 0	d	1	7	5	0 0 0 0		
	EXTU.W @(d:32,ERd)	7	8	0	rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0	d d	1	7	5	0 0 0 0	
	EXTU.W @(d:16,Rd,B)	0	1	5	0 1 0 1	6	F	0 rd	1 0 0 0	d	1	7	5	0 0 0 0		
	EXTU.W @(d:16,Rd,W)	0	1	5	0 1 1 0	6	F	0 rd	1 0 0 0	d	1	7	5	0 0 0 0		
	EXTU.W @(d:16,ERd,L)	0	1	5	0 1 1 1	6	F	0 rd	1 0 0 0	d	1	7	5	0 0 0 0		
	EXTU.W @(d:32,Rd,B)	7	8	0	rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0	d d	1	7	5	0 0 0 0	
	EXTU.W @(d:32,Rd,W)	7	8	0	rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0	d d	1	7	5	0 0 0 0	
	EXTU.W @(d:32,ERd,L)	7	8	0	rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0	d d	1	7	5	0 0 0 0	
	EXTU.W @aa:16					6	B	0 0 0 1	1 0 0 0	a	1	7	5	0 0 0 0		
	EXTU.W @aa:32					6	B	0 0 1 1	1 0 0 0	a a	1	7	5	0 0 0 0		
	EXTU.L ERd											1	7	7	0 rd	
	EXTU.L @ERd	0	1	0	0 1 0 0	6	9	0 rd	1 0 0 0		1	7	7	0 0 0 0		
	EXTU.L @ERd+	0	1	0	0 1 0 0	6	D	0 rd	1 0 0 0		1	7	7	0 0 0 0		
	EXTU.L @ERd-	0	1	0	0 1 1 0	6	D	0 rd	1 0 0 0		1	7	7	0 0 0 0		
	EXTU.L @+ERd	0	1	0	0 1 0 1	6	D	0 rd	1 0 0 0		1	7	7	0 0 0 0		
	EXTU.L @-ERd	0	1	0	0 1 1 1	6	D	0 rd	1 0 0 0		1	7	7	0 0 0 0		
	EXTU.L @(d:2,ERd)	0	1	0	0 1 d d	6	9	0 rd	1 0 0 0		1	7	7	0 0 0 0		
	EXTU.L @(d:16,ERd)	0	1	0	0 1 0 0	6	F	0 rd	1 0 0 0	d	1	7	7	0 0 0 0		
	EXTU.L @(d:32,ERd)	7	8	1	rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0	d d	1	7	7	0 0 0 0	
	EXTU.L @(d:16,Rd,B)	0	1	0	0 1 0 1	6	F	0 rd	1 0 0 0	d	1	7	7	0 0 0 0		
	EXTU.L @(d:16,Rd,W)	0	1	0	0 1 1 0	6	F	0 rd	1 0 0 0	d	1	7	7	0 0 0 0		
	EXTU.L @(d:16,ERd,L)	0	1	0	0 1 1 1	6	F	0 rd	1 0 0 0	d	1	7	7	0 0 0 0		
	EXTU.L @(d:32,Rd,B)	7	8	1	rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0	d d	1	7	7	0 0 0 0	
EXTU.L @(d:32,Rd,W)	7	8	1	rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0	d d	1	7	7	0 0 0 0		
EXTU.L @(d:32,ERd,L)	7	8	1	rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0	d d	1	7	7	0 0 0 0		
EXTU.L @aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0	1 0 0 0	a	1	7	7	0 0 0 0			
EXTU.L @aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0	1 0 0 0	a a	1	7	7	0 0 0 0			
EXTU.L #2,ERd											1	7	6	0 rd		
EXTU.L #2,@ERd	0	1	0	0 1 0 0	6	9	0 rd	1 0 0 0		1	7	6	0 0 0 0			
EXTU.L #2,@ERd+	0	1	0	0 1 0 0	6	D	0 rd	1 0 0 0		1	7	6	0 0 0 0			
EXTU.L #2,@ERd-	0	1	0	0 1 1 0	6	D	0 rd	1 0 0 0		1	7	6	0 0 0 0			
EXTU.L #2,@+ERd	0	1	0	0 1 0 1	6	D	0 rd	1 0 0 0		1	7	6	0 0 0 0			
EXTU.L #2,@-ERd	0	1	0	0 1 1 1	6	D	0 rd	1 0 0 0		1	7	6	0 0 0 0			

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension
		15	8	7	0	15	8	7	0		15	8	7	0	
EXTU	EXTU.L #2,@(d:2,ERd)	0	1	0	0 1 d d	6	9	0 rd	1 0 0 0		1	7	6	0 0 0 0	
	EXTU.L #2,@(d:16,ERd)	0	1	0	0 1 0 0	6	F	0 rd	1 0 0 0	d	1	7	6	0 0 0 0	
	EXTU.L #2,@(d:32,ERd)	7	8	1 rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0	d d	1	7	6	0 0 0 0	
	EXTU.L #2,@(d:16,Rd,B)	0	1	0	0 1 0 1	6	F	0 rd	1 0 0 0	d	1	7	6	0 0 0 0	
	EXTU.L #2,@(d:16,Rd,W)	0	1	0	0 1 1 0	6	F	0 rd	1 0 0 0	d	1	7	6	0 0 0 0	
	EXTU.L #2,@(d:16,ERd,L)	0	1	0	0 1 1 1	6	F	0 rd	1 0 0 0	d	1	7	6	0 0 0 0	
	EXTU.L #2,@(d:32,Rd,B)	7	8	1 rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0	d d	1	7	6	0 0 0 0	
	EXTU.L #2,@(d:32,Rd,W)	7	8	1 rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0	d d	1	7	6	0 0 0 0	
	EXTU.L #2,@(d:32,ERd,L)	7	8	1 rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0	d d	1	7	6	0 0 0 0	
	EXTU.L #2,@aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0	1 0 0 0	a	1	7	6	0 0 0 0	
EXTU.L #2,@aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0	1 0 0 0	a a	1	7	6	0 0 0 0		
INC	INC.B Rd										0	A	0 0 0 0	rd	
	INC.W #1,Rd										0	B	0 1 0 1	rd	
	INC.W #2,Rd										0	B	1 1 0 1	rd	
	INCL #1,ERd										0	B	0 1 1 1	0 rd	
	INCL #2,ERd										0	B	1 1 1 1	0 rd	
JMP	JMP @ERn										5	9	0 m	0 0 0 0	
	JMP @aa:24										5	A	a a a a	a a a a	a
	JMP @aa:32										5	9	0 0 0 0	1 0 0 0	a a
	JMP @ @aa:8										5	B	a a a a	a a a a	
	JMP @ @vec:7										5	9	1 v v v	v v v v	
JSR	JSR @ERn										5	D	0 m	0 0 0 0	
	JSR @aa:24										5	E	a a a a	a a a a	a
	JSR @aa:32										5	D	0 0 0 0	1 0 0 0	a a
	JSR @ @aa:8										5	F	a a a a	a a a a	
JSR @ @vec:7										5	D	1 v v v	v v v v		
LDC	LDC.B #xx:8,CCR										0	7	x x x x	x x x x	
	LDC.B Rs,CCR										0	3	0 0 0 0	r r r r	
	LDC.W @ERs,CCR					0	1	4	0 0 0 0		6	9	0 rs	0 0 0 0	
	LDC.W @ERs+,CCR					0	1	4	0 0 0 0		6	D	0 rs	0 0 0 0	
	LDC.W @(d:16,ERs),CCR					0	1	4	0 0 0 0		6	F	0 rs	0 0 0 0	d
	LDC.W @(d:32,ERs),CCR	0	1	4	0 0 0 0	7	8	0 rs	0 0 0 0		6	B	0 0 1 0	0 0 0 0	d d
	LDC.W @aa:16,CCR					0	1	4	0 0 0 0		6	B	0 0 0 0	0 0 0 0	a
LDC.W @aa:32,CCR					0	1	4	0 0 0 0		6	B	0 0 1 0	0 0 0 0	a a	



Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension														
		15	8	7	0		15	8	7	0															
LDC	LDC.B #xx:8,EXR			0	1	4	0	0	0	0	1		0	7	x	x	x	x	x	x	x				
	LDC.B Rs,EXR												0	3	0	0	0	1	r	r	r	r			
	LDC.W @ERs,EXR			0	1	4	0	0	0	0	1		6	9	0	0	rs	0	0	0	0	0			
	LDC.W @ERs+,EXR			0	1	4	0	0	0	0	1		6	D	0	0	rs	0	0	0	0	0			
	LDC.W @(d:16,ERs),EXR			0	1	4	0	0	0	0	1		6	F	0	0	rs	0	0	0	0	0	d		
	LDC.W @(d:32,ERs),EXR		0	1	4	0	0	0	0	1		7	8	0	1	rs	0	0	0	0	0	0	d	d	
	LDC.W @aa:16,EXR				0	1	4	0	0	0	0	1		6	B	0	0	0	0	0	0	0	a		
	LDC.W @aa:32,EXR				0	1	4	0	0	0	0	1		6	B	0	0	1	0	0	0	0	a	a	
	LDC.L ERs,SBR													0	3	0	1	1	1	0		rs			
LDC.L ERs,VBR													0	3	0	1	1	0	0		rs				
LDM	LDM.L @SP+,(ERn-ERn+1)			0	1	1	0	0	0	0			6	D	0	1	1	1	0		rn+1				
	LDM.L @SP+,(ERn-ERn+2)			0	1	2	0	0	0	0			6	D	0	1	1	1	0		rn+2				
	LDM.L @SP+,(ERn-ERn+3)			0	1	3	0	0	0	0			6	D	0	1	1	1	0		rn+3				
LDMAC	LDMAC ERs,MACH												0	3	0	0	1	0	0		rs				
	LDMAC ERs,MACL												0	3	0	0	1	1	0		rs				
MAC	MAC @ERn+,@ERm+			0	1	6	0	0	0	0			6	D	0		m	0		mm					
MOV	MOV.B #xx:8,Rd												F	rd	x	x	x	x	x	x	x				
	MOV.B #xx:4,@aa:16												6	A	1	1	0	1	x	x	x	a			
	MOV.B #xx:4,@aa:32												6	A	1	1	1	1	x	x	x	a	a		
	MOV.B #xx:8,@ERd			0	1	7	1	1	0	1			0	0	0	0	0	rd	x	x	x	x	x		
	MOV.B #xx:8,@ERd+			0	1	7	1	1	0	1			1	0	0	0	0	rd	x	x	x	x	x		
	MOV.B #xx:8,@ERd-			0	1	7	1	1	0	1			1	0	1	0	0	rd	x	x	x	x	x		
	MOV.B #xx:8,@+ERd			0	1	7	1	1	0	1			1	0	0	1	0	rd	x	x	x	x	x		
	MOV.B #xx:8,@-ERd			0	1	7	1	1	0	1			1	0	1	1	0	rd	x	x	x	x	x		
	MOV.B #xx:8,@(d:2,ERd)			0	1	7	1	1	0	1			0	0	d	d	0	rd	x	x	x	x	x		
	MOV.B #xx:8,@(d:16,ERd)			0	1	7	1	1	0	1			1	1	0	0	0	rd	x	x	x	x	x	d	
	MOV.B #xx:8,@(d:32,ERd)			0	1	7	1	1	0	1			1	1	0	0	1	rd	x	x	x	x	x	d	d
	MOV.B #xx:8,@(d:16,Rd,B)			0	1	7	1	1	0	1			1	1	0	1	0	rd	x	x	x	x	x	d	
	MOV.B #xx:8,@(d:16,Rd,W)			0	1	7	1	1	0	1			1	1	1	0	0	rd	x	x	x	x	x	d	
	MOV.B #xx:8,@(d:16,ERd,L)			0	1	7	1	1	0	1			1	1	1	1	0	rd	x	x	x	x	x	d	
	MOV.B #xx:8,@(d:32,Rd,B)			0	1	7	1	1	0	1			1	1	0	1	1	rd	x	x	x	x	x	d	d
MOV.B #xx:8,@(d:32,Rd,W)			0	1	7	1	1	0	1			1	1	1	0	1	rd	x	x	x	x	x	d	d	
MOV.B #xx:8,@(d:32,ERd,L)			0	1	7	1	1	0	1			1	1	1	1	1	rd	x	x	x	x	x	d	d	
MOV.B #xx:8,@aa:16				0	1	7	1	1	0	1			0	1	0	0	0	0	x	x	x	x	x	a	

Instruction	Mnemonic	Opcode			EA Extension	Opcode			EA Extension					
		15	8	7		0	15	8		7	0			
MOV	MOV.B #xc:8,@aa:32				0	1	7	1 1 0 1	0 1 0 0	1 0 0 0	x x x x	x x x x	a a	
	MOV.B Rs,Rd								0	C	rs	rd		
	MOV.B Rs,@ERd								6	8	1	rd	rs	
	MOV.B Rs,@ERd+				0	1	7	0 0 1 1	6	C	1	rd	rs	
	MOV.B Rs,@ERd-				0	1	7	0 0 0 1	6	C	1	rd	rs	
	MOV.B Rs,@+ERd				0	1	7	0 0 1 0	6	C	1	rd	rs	
	MOV.B Rs,@-ERd								6	C	1	rd	rs	
	MOV.B Rs,@(d:2,ERd)				0	1	7	0 0 d d	6	8	1	rd	rs	
	MOV.B Rs,@(d:16,ERd)								6	E	1	rd	rs	d
	MOV.B Rs,@(d:32,ERd)				7	8	0	rd 0 0 0 0	6	A	1 0 1 0	rs	d d	
	MOV.B Rs,@(d:16,Rd.B)				0	1	7	0 0 0 1	6	E	1	rd	rs	d
	MOV.B Rs,@(d:16,Rd.W)				0	1	7	0 0 1 0	6	E	1	rd	rs	d
	MOV.B Rs,@(d:16,ERd.L)				0	1	7	0 0 1 1	6	E	1	rd	rs	d
	MOV.B Rs,@(d:32,Rd.B)				7	8	0	rd 0 0 0 1	6	A	1 0 1 0	rs	d d	
	MOV.B Rs,@(d:32,Rd.W)				7	8	0	rd 0 0 1 0	6	A	1 0 1 0	rs	d d	
	MOV.B Rs,@(d:32,ERd.L)				7	8	0	rd 0 0 1 1	6	A	1 0 1 0	rs	d d	
	MOV.B Rs,@aa:8								3	rs	a a a a	a a a a		
	MOV.B Rs,@aa:16								6	A	1 0 0 0	rs	a	
	MOV.B Rs,@aa:32								6	A	1 0 1 0	rs	a a	
	MOV.B @ERs,Rd								6	8	0	rs	rd	
	MOV.B @ERs+,Rd								6	C	0	rs	rd	
	MOV.B @ERs-,Rd				0	1	7	0 0 1 0	6	C	0	rs	rd	
	MOV.B @+ERs,Rd				0	1	7	0 0 0 1	6	C	0	rs	rd	
	MOV.B @-ERs,Rd				0	1	7	0 0 1 1	6	C	0	rs	rd	
	MOV.B @(d:2,ERs),Rd				0	1	7	0 0 d d	6	8	0	rs	rd	
	MOV.B @(d:16,ERs),Rd								6	E	0	rs	rd	d
	MOV.B @(d:32,ERs),Rd				7	8	0	rs 0 0 0 0	6	A	0 0 1 0	rd	d d	
	MOV.B @(d:16,Rs.B),Rd				0	1	7	0 0 0 1	6	E	0	rs	rd	d
	MOV.B @(d:16,Rs.W),Rd				0	1	7	0 0 1 0	6	E	0	rs	rd	d
	MOV.B @(d:16,ERs.L),Rd				0	1	7	0 0 1 1	6	E	0	rs	rd	d
	MOV.B @(d:32,Rs.B),Rd				7	8	0	rs 0 0 0 1	6	A	0 0 1 0	rd	d d	
	MOV.B @(d:32,Rs.W),Rd				7	8	0	rs 0 0 1 0	6	A	0 0 1 0	rd	d d	
	MOV.B @(d:32,ERs.L),Rd				7	8	0	rs 0 0 1 1	6	A	0 0 1 0	rd	d d	
	MOV.B @aa:8,Rd								2	rd	a a a a	a a a a		



Instruction	Mnemonic	Opcode			Opcode			EA Extension	Opcode			EA Extension												
		15	8	7	0	15	8		7	0	15		8	7	0									
MOV	MOV.B @aa:16,Rd								6	A	0	0	0	0	rd	a								
	MOV.B @aa:32,Rd								6	A	0	0	1	0	rd	a	a							
	MOV.B @ERs,@ERd				0	1	7	1	0	0	0	0	0	0	rs	0	0	0	0	0	rd			
	MOV.B @ERs,@ERd+				0	1	7	1	0	0	0	0	0	0	rs	1	0	0	0	0	rd			
	MOV.B @ERs,@ERd-				0	1	7	1	0	0	0	0	0	0	rs	1	0	1	0	0	rd			
	MOV.B @ERs,@+ERd				0	1	7	1	0	0	0	0	0	0	rs	1	0	0	1	0	rd			
	MOV.B @ERs,@-ERd				0	1	7	1	0	0	0	0	0	0	rs	1	0	1	1	0	rd			
	MOV.B @ERs,@(d:2,ERd)				0	1	7	1	0	0	0	0	0	0	rs	0	0	d	d	0	rd			
	MOV.B @ERs,@(d:16,ERd)				0	1	7	1	0	0	0	0	0	0	rs	1	1	0	0	0	rd		d	
	MOV.B @ERs,@(d:32,ERd)				0	1	7	1	0	0	0	0	0	0	rs	1	1	0	0	1	rd		d	
	MOV.B @ERs,@(d:16,Rd,B)				0	1	7	1	0	0	0	0	0	0	rs	1	1	0	1	0	rd		d	
	MOV.B @ERs,@(d:16,Rd,W)				0	1	7	1	0	0	0	0	0	0	rs	1	1	1	0	0	rd		d	
	MOV.B @ERs,@(d:16,ERd,L)				0	1	7	1	0	0	0	0	0	0	rs	1	1	1	1	0	rd		d	
	MOV.B @ERs,@(d:32,Rd,B)				0	1	7	1	0	0	0	0	0	0	rs	1	1	0	1	1	rd		d	
	MOV.B @ERs,@(d:32,Rd,W)				0	1	7	1	0	0	0	0	0	0	rs	1	1	1	0	1	rd		d	
	MOV.B @ERs,@(d:32,ERd,L)				0	1	7	1	0	0	0	0	0	0	rs	1	1	1	1	1	rd		d	
	MOV.B @ERs,@aa:16				0	1	7	1	0	0	0	0	0	0	rs	0	1	0	0	0	0	0	0	a
	MOV.B @ERs,@aa:32				0	1	7	1	0	0	0	0	0	0	rs	0	1	0	0	1	0	0	0	a
	MOV.B @ERs+,@ERd				0	1	7	1	0	0	0	1	0	0	rs	0	0	0	0	0	rd			
	MOV.B @ERs+,@ERd+				0	1	7	1	0	0	0	1	0	0	rs	1	0	0	0	0	rd			
	MOV.B @ERs+,@ERd-				0	1	7	1	0	0	0	1	0	0	rs	1	0	1	0	0	rd			
	MOV.B @ERs+,@+ERd				0	1	7	1	0	0	0	1	0	0	rs	1	0	0	1	0	rd			
	MOV.B @ERs+,@-ERd				0	1	7	1	0	0	0	1	0	0	rs	1	0	1	1	0	rd			
	MOV.B @ERs+,@(d:2,ERd)				0	1	7	1	0	0	0	1	0	0	rs	0	0	d	d	0	rd			
	MOV.B @ERs+,@(d:16,ERd)				0	1	7	1	0	0	0	1	0	0	rs	1	1	0	0	0	rd		d	
	MOV.B @ERs+,@(d:32,ERd)				0	1	7	1	0	0	0	1	0	0	rs	1	1	0	0	1	rd		d	
	MOV.B @ERs+,@(d:16,Rd,B)				0	1	7	1	0	0	0	1	0	0	rs	1	1	0	1	0	rd		d	
	MOV.B @ERs+,@(d:16,Rd,W)				0	1	7	1	0	0	0	1	0	0	rs	1	1	1	0	0	rd		d	
MOV.B @ERs+,@(d:16,ERd,L)				0	1	7	1	0	0	0	1	0	0	rs	1	1	1	1	0	rd		d		
MOV.B @ERs+,@(d:32,Rd,B)				0	1	7	1	0	0	0	1	0	0	rs	1	1	0	1	1	rd		d		
MOV.B @ERs+,@(d:32,Rd,W)				0	1	7	1	0	0	0	1	0	0	rs	1	1	1	0	1	rd		d		
MOV.B @ERs+,@(d:32,ERd,L)				0	1	7	1	0	0	0	1	0	0	rs	1	1	1	1	1	rd		d		
MOV.B @ERs+,@aa:16				0	1	7	1	0	0	0	1	0	0	rs	0	1	0	0	0	0	0	0	a	
MOV.B @ERs+,@aa:32				0	1	7	1	0	0	0	1	0	0	rs	0	1	0	0	1	0	0	0	a	

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension																					
		15	8	7	0		15	8	7	0																						
MOV	MOV.B @ERs, @ERd				0			1		7		1	0	0	0	0		1	0	1	0	0	rs	0	0	0	0	0	0	rd		
	MOV.B @ERs, @ERd+				0			1		7		1	0	0	0	0		1	0	1	0	0	rs	1	0	0	0	0	0	rd		
	MOV.B @ERs, @ERd-				0			1		7		1	0	0	0	0		1	0	1	0	0	rs	1	0	1	0	0	0	rd		
	MOV.B @ERs, @+ERd				0			1		7		1	0	0	0	0		1	0	1	0	0	rs	1	0	0	1	0	0	rd		
	MOV.B @ERs, @-ERd				0			1		7		1	0	0	0	0		1	0	1	0	0	rs	1	0	1	1	0	0	rd		
	MOV.B @ERs, @(d:2,ERd)				0			1		7		1	0	0	0	0		1	0	1	0	0	rs	0	0	d	d	0	0	rd		
	MOV.B @ERs, @(d:16,ERd)				0			1		7		1	0	0	0	0		1	0	1	0	0	rs	1	1	0	0	0	0	rd	d	
	MOV.B @ERs, @(d:32,ERd)				0			1		7		1	0	0	0	0		1	0	1	0	0	rs	1	1	0	0	1	1	rd	d d	
	MOV.B @ERs, @(d:16,Rd,B)				0			1		7		1	0	0	0	0		1	0	1	0	0	rs	1	1	0	1	0	0	rd	d	
	MOV.B @ERs, @(d:16,Rd,W)				0			1		7		1	0	0	0	0		1	0	1	0	0	rs	1	1	1	0	0	0	rd	d	
	MOV.B @ERs, @(d:16,ERd,L)				0			1		7		1	0	0	0	0		1	0	1	0	0	rs	1	1	1	1	0	0	rd	d	
	MOV.B @ERs, @(d:32,Rd,B)				0			1		7		1	0	0	0	0		1	0	1	0	0	rs	1	1	0	1	1	1	rd	d d	
	MOV.B @ERs, @(d:32,Rd,W)				0			1		7		1	0	0	0	0		1	0	1	0	0	rs	1	1	1	0	1	1	rd	d d	
	MOV.B @ERs, @(d:32,ERd,L)				0			1		7		1	0	0	0	0		1	0	1	0	0	rs	1	1	1	1	1	1	rd	d d	
	MOV.B @ERs, @aa:16				0			1		7		1	0	0	0	0		1	0	1	0	0	rs	0	1	0	0	0	0	0	0	a
	MOV.B @ERs, @aa:32				0			1		7		1	0	0	0	0		1	0	1	0	0	rs	0	1	0	0	1	0	0	0	a a
	MOV.B @+ERs, @ERd				0			1		7		1	0	0	0	0		1	0	0	1	0	rs	0	0	0	0	0	0	rd		
	MOV.B @+ERs, @ERd+				0			1		7		1	0	0	0	0		1	0	0	1	0	rs	1	0	0	0	0	0	rd		
	MOV.B @+ERs, @ERd-				0			1		7		1	0	0	0	0		1	0	0	1	0	rs	1	0	1	0	0	0	rd		
	MOV.B @+ERs, @+ERd				0			1		7		1	0	0	0	0		1	0	0	1	0	rs	1	0	0	1	0	0	rd		
	MOV.B @+ERs, @-ERd				0			1		7		1	0	0	0	0		1	0	0	1	0	rs	1	0	1	1	0	0	rd		
	MOV.B @+ERs, @(d:2,ERd)				0			1		7		1	0	0	0	0		1	0	0	1	0	rs	0	0	d	d	0	0	rd		
	MOV.B @+ERs, @(d:16,ERd)				0			1		7		1	0	0	0	0		1	0	0	1	0	rs	1	1	0	0	0	0	rd	d	
	MOV.B @+ERs, @(d:32,ERd)				0			1		7		1	0	0	0	0		1	0	0	1	0	rs	1	1	0	0	1	1	rd	d d	
	MOV.B @+ERs, @(d:16,Rd,B)				0			1		7		1	0	0	0	0		1	0	0	1	0	rs	1	1	0	1	0	0	rd	d	
	MOV.B @+ERs, @(d:16,Rd,W)				0			1		7		1	0	0	0	0		1	0	0	1	0	rs	1	1	1	0	0	0	rd	d	
	MOV.B @+ERs, @(d:16,ERd,L)				0			1		7		1	0	0	0	0		1	0	0	1	0	rs	1	1	1	1	0	0	rd	d	
	MOV.B @+ERs, @(d:32,Rd,B)				0			1		7		1	0	0	0	0		1	0	0	1	0	rs	1	1	0	1	1	1	rd	d d	
	MOV.B @+ERs, @(d:32,Rd,W)				0			1		7		1	0	0	0	0		1	0	0	1	0	rs	1	1	1	0	1	1	rd	d d	
	MOV.B @+ERs, @(d:32,ERd,L)				0			1		7		1	0	0	0	0		1	0	0	1	0	rs	1	1	1	1	1	1	rd	d d	
	MOV.B @+ERs, @aa:16				0			1		7		1	0	0	0	0		1	0	0	1	0	rs	0	1	0	0	0	0	0	0	a
	MOV.B @+ERs, @aa:32				0			1		7		1	0	0	0	0		1	0	0	1	0	rs	0	1	0	0	1	0	0	0	a a
	MOV.B @-ERs, @ERd				0			1		7		1	0	0	0	0		1	0	1	1	0	rs	0	0	0	0	0	0	rd		
	MOV.B @-ERs, @ERd+				0			1		7		1	0	0	0	0		1	0	1	1	0	rs	1	0	0	0	0	0	rd		



Instruction	Mnemonic	Opcode			Opcode			EA Extension	Opcode			EA Extension					
		15	8	7	0	15	8		7	0	15		8	7	0		
MOV	MOV.B @-ERs,@ERd-				0	1	7	1 0 0 0		1 0 1 1	0	rs	1 0 1 0	0	rd		
	MOV.B @-ERs,@+ERd				0	1	7	1 0 0 0		1 0 1 1	0	rs	1 0 0 1	0	rd		
	MOV.B @-ERs,@-ERd				0	1	7	1 0 0 0		1 0 1 1	0	rs	1 0 1 1	0	rd		
	MOV.B @(d2,ERd)				0	1	7	1 0 0 0		1 0 1 1	0	rs	0 0 d d	0	rd		
	MOV.B @-ERs,@(d:16,ERd)				0	1	7	1 0 0 0		1 0 1 1	0	rs	1 1 0 0	0	rd	d	
	MOV.B @-ERs,@(d:32,ERd)				0	1	7	1 0 0 0		1 0 1 1	0	rs	1 1 0 0	1	rd	d	d
	MOV.B @-ERs,@(d:16,Rd,B)				0	1	7	1 0 0 0		1 0 1 1	0	rs	1 1 0 1	0	rd	d	
	MOV.B @-ERs,@(d:16,Rd,W)				0	1	7	1 0 0 0		1 0 1 1	0	rs	1 1 1 0	0	rd	d	
	MOV.B @-ERs,@(d:16,ERd,L)				0	1	7	1 0 0 0		1 0 1 1	0	rs	1 1 1 1	0	rd	d	
	MOV.B @-ERs,@(d:32,Rd,B)				0	1	7	1 0 0 0		1 0 1 1	0	rs	1 1 0 1	1	rd	d	d
	MOV.B @-ERs,@(d:32,Rd,W)				0	1	7	1 0 0 0		1 0 1 1	0	rs	1 1 1 0	1	rd	d	d
	MOV.B @-ERs,@(d:32,ERd,L)				0	1	7	1 0 0 0		1 0 1 1	0	rs	1 1 1 1	1	rd	d	d
	MOV.B @-ERs,@aa:16				0	1	7	1 0 0 0		1 0 1 1	0	rs	0 1 0 0	0 0 0 0	0	a	
	MOV.B @-ERs,@aa:32				0	1	7	1 0 0 0		1 0 1 1	0	rs	0 1 0 0	1 0 0 0	0	a	a
	MOV.B @(d2,ERs),@ERd				0	1	7	1 0 0 0		0 0 d d	0	rs	0 0 0 0	0	rd		
	MOV.B @(d2,ERs),@ERd+				0	1	7	1 0 0 0		0 0 d d	0	rs	1 0 0 0	0	rd		
	MOV.B @(d2,ERs),@ERd-				0	1	7	1 0 0 0		0 0 d d	0	rs	1 0 1 0	0	rd		
	MOV.B @(d2,ERs),@+ERd				0	1	7	1 0 0 0		0 0 d d	0	rs	1 0 0 1	0	rd		
	MOV.B @(d2,ERs),@-ERd				0	1	7	1 0 0 0		0 0 d d	0	rs	1 0 1 1	0	rd		
	MOV.B @(d2,ERs),@(d2,ERd)				0	1	7	1 0 0 0		0 0 d d	0	rs	0 0 d d	0	rd		
	MOV.B @(d2,ERs),@(d:16,ERd)				0	1	7	1 0 0 0		0 0 d d	0	rs	1 1 0 0	0	rd	d	
	MOV.B @(d2,ERs),@(d:32,ERd)				0	1	7	1 0 0 0		0 0 d d	0	rs	1 1 0 0	1	rd	d	d
	MOV.B @(d2,ERs),@(d:16,Rd,B)				0	1	7	1 0 0 0		0 0 d d	0	rs	1 1 0 1	0	rd	d	
	MOV.B @(d2,ERs),@(d:16,Rd,W)				0	1	7	1 0 0 0		0 0 d d	0	rs	1 1 1 0	0	rd	d	
	MOV.B @(d2,ERs),@(d:16,ERd,L)				0	1	7	1 0 0 0		0 0 d d	0	rs	1 1 1 1	0	rd	d	
	MOV.B @(d2,ERs),@(d:32,Rd,B)				0	1	7	1 0 0 0		0 0 d d	0	rs	1 1 0 1	1	rd	d	d
	MOV.B @(d2,ERs),@(d:32,Rd,W)				0	1	7	1 0 0 0		0 0 d d	0	rs	1 1 1 0	1	rd	d	d
	MOV.B @(d2,ERs),@(d:32,ERd,L)				0	1	7	1 0 0 0		0 0 d d	0	rs	1 1 1 1	1	rd	d	d
	MOV.B @(d2,ERs),@aa:16				0	1	7	1 0 0 0		0 0 d d	0	rs	0 1 0 0	0 0 0 0	0	a	
	MOV.B @(d2,ERs),@aa:32				0	1	7	1 0 0 0		0 0 d d	0	rs	0 1 0 0	1 0 0 0	0	a	a
MOV.B @(d:16,ERs),@ERd				0	1	7	1 0 0 0		1 1 0 0	0	rs	0 0 0 0	0	rd	d		
MOV.B @(d:16,ERs),@ERd+				0	1	7	1 0 0 0		1 1 0 0	0	rs	1 0 0 0	0	rd	d		
MOV.B @(d:16,ERs),@ERd-				0	1	7	1 0 0 0		1 1 0 0	0	rs	1 0 1 0	0	rd	d		
MOV.B @(d:16,ERs),@+ERd				0	1	7	1 0 0 0		1 1 0 0	0	rs	1 0 0 1	0	rd	d		

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension																				
		15	8	7	0		15	8	7	0																					
MOV	MOV.B @(d:16,ERs),@-ERd				0	1		7	1	0	0	0	0	1	1	0	0	0	rs	1	0	1	1	0	rd	d					
	MOV.B @(d:16,ERs),@(d:2,ERd)				0	1		7	1	0	0	0	0	1	1	0	0	0	rs	0	0	d	d	0	rd	d					
	MOV.B @(d:16,ERs),@(d:16,ERd)				0	1		7	1	0	0	0	0	1	1	0	0	0	rs	1	1	0	0	0	rd	d	d				
	MOV.B @(d:16,ERs),@(d:32,ERd)				0	1		7	1	0	0	0	0	1	1	0	0	0	rs	1	1	0	0	1	rd	d	d				
	MOV.B @(d:16,ERs),@(d:16,Rd.B)				0	1		7	1	0	0	0	0	1	1	0	0	0	rs	1	1	0	1	0	rd	d	d				
	MOV.B @(d:16,ERs),@(d:16,Rd.W)				0	1		7	1	0	0	0	0	1	1	0	0	0	rs	1	1	1	0	0	rd	d	d				
	MOV.B @(d:16,ERs),@(d:16,ERd.L)				0	1		7	1	0	0	0	0	1	1	0	0	0	rs	1	1	1	1	0	rd	d	d				
	MOV.B @(d:16,ERs),@(d:32,Rd.B)				0	1		7	1	0	0	0	0	1	1	0	0	0	rs	1	1	0	1	1	rd	d	d				
	MOV.B @(d:16,ERs),@(d:32,Rd.W)				0	1		7	1	0	0	0	0	1	1	0	0	0	rs	1	1	1	0	1	rd	d	d				
	MOV.B @(d:16,ERs),@(d:32,ERd.L)				0	1		7	1	0	0	0	0	1	1	0	0	0	rs	1	1	1	1	1	rd	d	d				
	MOV.B @(d:16,ERs),@aa:16				0	1		7	1	0	0	0	0	1	1	0	0	0	rs	0	1	0	0	0	0	0	0	0	d	a	
	MOV.B @(d:16,ERs),@aa:32				0	1		7	1	0	0	0	0	1	1	0	0	0	rs	0	1	0	0	1	0	0	0	0	d	a	a
	MOV.B @(d:32,ERs),@ERd				0	1		7	1	0	0	0	0	1	1	0	0	1	rs	0	0	0	0	0	rd	d	d				
	MOV.B @(d:32,ERs),@ERd+				0	1		7	1	0	0	0	0	1	1	0	0	1	rs	1	0	0	0	0	rd	d	d				
	MOV.B @(d:32,ERs),@ERd-				0	1		7	1	0	0	0	0	1	1	0	0	1	rs	1	0	1	0	0	rd	d	d				
	MOV.B @(d:32,ERs),@+ERd				0	1		7	1	0	0	0	0	1	1	0	0	1	rs	1	0	0	1	0	rd	d	d				
	MOV.B @(d:32,ERs),@-ERd				0	1		7	1	0	0	0	0	1	1	0	0	1	rs	1	0	1	1	0	rd	d	d				
	MOV.B @(d:32,ERs),@(d:2,ERd)				0	1		7	1	0	0	0	0	1	1	0	0	1	rs	0	0	d	d	0	rd	d	d				
	MOV.B @(d:32,ERs),@(d:16,ERd)				0	1		7	1	0	0	0	0	1	1	0	0	1	rs	1	1	0	0	0	rd	d	d				
	MOV.B @(d:32,ERs),@(d:32,ERd)				0	1		7	1	0	0	0	0	1	1	0	0	1	rs	1	1	0	0	1	rd	d	d				
	MOV.B @(d:32,ERs),@(d:16,Rd.B)				0	1		7	1	0	0	0	0	1	1	0	0	1	rs	1	1	0	1	0	rd	d	d				
	MOV.B @(d:32,ERs),@(d:16,Rd.W)				0	1		7	1	0	0	0	0	1	1	0	0	1	rs	1	1	1	0	0	rd	d	d				
	MOV.B @(d:32,ERs),@(d:16,ERd.L)				0	1		7	1	0	0	0	0	1	1	0	0	1	rs	1	1	1	1	0	rd	d	d				
	MOV.B @(d:32,ERs),@(d:32,Rd.B)				0	1		7	1	0	0	0	0	1	1	0	0	1	rs	1	1	0	1	1	rd	d	d				
	MOV.B @(d:32,ERs),@(d:32,Rd.W)				0	1		7	1	0	0	0	0	1	1	0	0	1	rs	1	1	1	0	1	rd	d	d				
	MOV.B @(d:32,ERs),@(d:32,ERd.L)				0	1		7	1	0	0	0	0	1	1	0	0	1	rs	1	1	1	1	1	rd	d	d				
	MOV.B @(d:32,ERs),@aa:16				0	1		7	1	0	0	0	0	1	1	0	0	1	rs	0	1	0	0	0	0	0	0	0	d	a	
	MOV.B @(d:32,ERs),@aa:32				0	1		7	1	0	0	0	0	1	1	0	0	1	rs	0	1	0	0	1	0	0	0	0	d	a	a
	MOV.B @(d:16,Rs.B),@ERd				0	1		7	1	0	0	0	0	1	1	0	1	0	rs	0	0	0	0	0	rd	d					
	MOV.B @(d:16,Rs.B),@ERd+				0	1		7	1	0	0	0	0	1	1	0	1	0	rs	1	0	0	0	0	rd	d					
	MOV.B @(d:16,Rs.B),@ERd-				0	1		7	1	0	0	0	0	1	1	0	1	0	rs	1	0	1	0	0	rd	d					
	MOV.B @(d:16,Rs.B),@+ERd				0	1		7	1	0	0	0	0	1	1	0	1	0	rs	1	0	0	1	0	rd	d					
MOV.B @(d:16,Rs.B),@-ERd				0	1		7	1	0	0	0	0	1	1	0	1	0	rs	1	0	1	1	0	rd	d						
MOV.B @(d:16,Rs.B),@(d:2,ERd)				0	1		7	1	0	0	0	0	1	1	0	1	0	rs	0	0	d	d	0	rd	d						



Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0		15	8	7	0			
MOV	MOV.B @(d:16,Rs.B),@(d:16,ERd)	0	1	7	1 0 0 0	1 1 0 1	0	rs	1 1 0 0	0	rd	d	d
	MOV.B @(d:16,Rs.B),@(d:32,ERd)	0	1	7	1 0 0 0	1 1 0 1	0	rs	1 1 0 0	1	rd	d	d d
	MOV.B @(d:16,Rs.B),@(d:16,Rd.B)	0	1	7	1 0 0 0	1 1 0 1	0	rs	1 1 0 1	0	rd	d	d
	MOV.B @(d:16,Rs.B),@(d:16,Rd.W)	0	1	7	1 0 0 0	1 1 0 1	0	rs	1 1 1 0	0	rd	d	d
	MOV.B @(d:16,Rs.B),@(d:16,ERd.L)	0	1	7	1 0 0 0	1 1 0 1	0	rs	1 1 1 1	0	rd	d	d
	MOV.B @(d:16,Rs.B),@(d:32,Rd.B)	0	1	7	1 0 0 0	1 1 0 1	0	rs	1 1 0 1	1	rd	d	d d
	MOV.B @(d:16,Rs.B),@(d:32,Rd.W)	0	1	7	1 0 0 0	1 1 0 1	0	rs	1 1 1 0	1	rd	d	d d
	MOV.B @(d:16,Rs.B),@(d:32,ERd.L)	0	1	7	1 0 0 0	1 1 0 1	0	rs	1 1 1 1	1	rd	d	d d
	MOV.B @(d:16,Rs.B),@aa:16	0	1	7	1 0 0 0	1 1 0 1	0	rs	0 1 0 0	0 0 0 0	d	a	
	MOV.B @(d:16,Rs.B),@aa:32	0	1	7	1 0 0 0	1 1 0 1	0	rs	0 1 0 0	1 0 0 0	d	a a	
	MOV.B @(d:16,Rs.W),@ERd	0	1	7	1 0 0 0	1 1 1 0	0	rs	0 0 0 0	0	rd	d	
	MOV.B @(d:16,Rs.W),@ERd+	0	1	7	1 0 0 0	1 1 1 0	0	rs	1 0 0 0	0	rd	d	
	MOV.B @(d:16,Rs.W),@ERd-	0	1	7	1 0 0 0	1 1 1 0	0	rs	1 0 1 0	0	rd	d	
	MOV.B @(d:16,Rs.W),@+ERd	0	1	7	1 0 0 0	1 1 1 0	0	rs	1 0 0 1	0	rd	d	
	MOV.B @(d:16,Rs.W),@-ERd	0	1	7	1 0 0 0	1 1 1 0	0	rs	1 0 1 1	0	rd	d	
	MOV.B @(d:16,Rs.W),@(d:2,ERd)	0	1	7	1 0 0 0	1 1 1 0	0	rs	0 0 d d	0	rd	d	
	MOV.B @(d:16,Rs.W),@(d:16,ERd)	0	1	7	1 0 0 0	1 1 1 0	0	rs	1 1 0 0	0	rd	d	d
	MOV.B @(d:16,Rs.W),@(d:32,ERd)	0	1	7	1 0 0 0	1 1 1 0	0	rs	1 1 0 0	1	rd	d	d d
	MOV.B @(d:16,Rs.W),@(d:16,Rd.B)	0	1	7	1 0 0 0	1 1 1 0	0	rs	1 1 0 1	0	rd	d	d
	MOV.B @(d:16,Rs.W),@(d:16,Rd.W)	0	1	7	1 0 0 0	1 1 1 0	0	rs	1 1 1 0	0	rd	d	d
	MOV.B @(d:16,Rs.W),@(d:16,ERd.L)	0	1	7	1 0 0 0	1 1 1 0	0	rs	1 1 1 1	0	rd	d	d
	MOV.B @(d:16,Rs.W),@(d:32,Rd.B)	0	1	7	1 0 0 0	1 1 1 0	0	rs	1 1 0 1	1	rd	d	d d
	MOV.B @(d:16,Rs.W),@(d:32,Rd.W)	0	1	7	1 0 0 0	1 1 1 0	0	rs	1 1 1 0	1	rd	d	d d
	MOV.B @(d:16,Rs.W),@(d:32,ERd.L)	0	1	7	1 0 0 0	1 1 1 0	0	rs	1 1 1 1	1	rd	d	d d
	MOV.B @(d:16,Rs.W),@aa:16	0	1	7	1 0 0 0	1 1 1 0	0	rs	0 1 0 0	0 0 0 0	d	a	
	MOV.B @(d:16,Rs.W),@aa:32	0	1	7	1 0 0 0	1 1 1 0	0	rs	0 1 0 0	1 0 0 0	d	a a	
	MOV.B @(d:16,ERs.L),@ERd	0	1	7	1 0 0 0	1 1 1 1	0	rs	0 0 0 0	0	rd	d	
	MOV.B @(d:16,ERs.L),@ERd+	0	1	7	1 0 0 0	1 1 1 1	0	rs	1 0 0 0	0	rd	d	
	MOV.B @(d:16,ERs.L),@ERd-	0	1	7	1 0 0 0	1 1 1 1	0	rs	1 0 1 0	0	rd	d	
	MOV.B @(d:16,ERs.L),@+ERd	0	1	7	1 0 0 0	1 1 1 1	0	rs	1 0 0 1	0	rd	d	
MOV.B @(d:16,ERs.L),@-ERd	0	1	7	1 0 0 0	1 1 1 1	0	rs	1 0 1 1	0	rd	d		
MOV.B @(d:16,ERs.L),@(d:2,ERd)	0	1	7	1 0 0 0	1 1 1 1	0	rs	0 0 d d	0	rd	d		
MOV.B @(d:16,ERs.L),@(d:16,ERd)	0	1	7	1 0 0 0	1 1 1 1	0	rs	1 1 0 0	0	rd	d	d	
MOV.B @(d:16,ERs.L),@(d:32,ERd)	0	1	7	1 0 0 0	1 1 1 1	0	rs	1 1 0 0	1	rd	d	d d	

Instruction	Mnemonic	Opcode				EA Ex- tension	Opcode				EA Extension		
		15	8	7	0		15	8	7	0			
MOV	MOV.B @(d:16,ERs.L),@(d:16,Rd.B)	0	1	7	1 0 0 0	1 1 1 1	0	rs	1 1 0 1	0	rd	d	d
	MOV.B @(d:16,ERs.L),@(d:16,Rd.W)	0	1	7	1 0 0 0	1 1 1 1	0	rs	1 1 1 0	0	rd	d	d
	MOV.B @(d:16,ERs.L),@(d:16,ERd.L)	0	1	7	1 0 0 0	1 1 1 1	0	rs	1 1 1 1	0	rd	d	d
	MOV.B @(d:16,ERs.L),@(d:32,Rd.B)	0	1	7	1 0 0 0	1 1 1 1	0	rs	1 1 0 1	1	rd	d	d d
	MOV.B @(d:16,ERs.L),@(d:32,Rd.W)	0	1	7	1 0 0 0	1 1 1 1	0	rs	1 1 1 0	1	rd	d	d d d
	MOV.B @(d:16,ERs.L),@(d:32,ERd.L)	0	1	7	1 0 0 0	1 1 1 1	0	rs	1 1 1 1	1	rd	d	d d d
	MOV.B @(d:16,ERs.L),@aa:16	0	1	7	1 0 0 0	1 1 1 1	0	rs	0 1 0 0	0 0 0 0	d	a	
	MOV.B @(d:16,ERs.L),@aa:32	0	1	7	1 0 0 0	1 1 1 1	0	rs	0 1 0 0	1 0 0 0	d	a	a
	MOV.B @(d:32,Rs.B),@ERd	0	1	7	1 0 0 0	1 1 0 1	1	rs	0 0 0 0	0	rd	d	d
	MOV.B @(d:32,Rs.B),@ERd+	0	1	7	1 0 0 0	1 1 0 1	1	rs	1 0 0 0	0	rd	d	d
	MOV.B @(d:32,Rs.B),@ERd-	0	1	7	1 0 0 0	1 1 0 1	1	rs	1 0 0 1	0	rd	d	d
	MOV.B @(d:32,Rs.B),@+ERd	0	1	7	1 0 0 0	1 1 0 1	1	rs	1 0 0 1	0	rd	d	d
	MOV.B @(d:32,Rs.B),@-ERd	0	1	7	1 0 0 0	1 1 0 1	1	rs	1 0 1 1	0	rd	d	d
	MOV.B @(d:32,Rs.B),@(d:2,ERd)	0	1	7	1 0 0 0	1 1 0 1	1	rs	0 0 d d	0	rd	d	d
	MOV.B @(d:32,Rs.B),@(d:16,ERd)	0	1	7	1 0 0 0	1 1 0 1	1	rs	1 1 0 0	0	rd	d	d d
	MOV.B @(d:32,Rs.B),@(d:32,ERd)	0	1	7	1 0 0 0	1 1 0 1	1	rs	1 1 0 0	1	rd	d	d d d
	MOV.B @(d:32,Rs.B),@(d:16,Rd.B)	0	1	7	1 0 0 0	1 1 0 1	1	rs	1 1 0 1	0	rd	d	d d
	MOV.B @(d:32,Rs.B),@(d:16,Rd.W)	0	1	7	1 0 0 0	1 1 0 1	1	rs	1 1 1 0	0	rd	d	d d
	MOV.B @(d:32,Rs.B),@(d:16,ERd.L)	0	1	7	1 0 0 0	1 1 0 1	1	rs	1 1 1 1	0	rd	d	d d
	MOV.B @(d:32,Rs.B),@(d:32,Rd.B)	0	1	7	1 0 0 0	1 1 0 1	1	rs	1 1 0 1	1	rd	d	d d d
	MOV.B @(d:32,Rs.B),@(d:32,Rd.W)	0	1	7	1 0 0 0	1 1 0 1	1	rs	1 1 1 0	1	rd	d	d d d
	MOV.B @(d:32,Rs.B),@(d:32,ERd.L)	0	1	7	1 0 0 0	1 1 0 1	1	rs	1 1 1 1	1	rd	d	d d d
	MOV.B @(d:32,Rs.B),@aa:16	0	1	7	1 0 0 0	1 1 0 1	1	rs	0 1 0 0	0 0 0 0	d	a	
	MOV.B @(d:32,Rs.B),@aa:32	0	1	7	1 0 0 0	1 1 0 1	1	rs	0 1 0 0	1 0 0 0	d	a	a
	MOV.B @(d:32,Rs.W),@ERd	0	1	7	1 0 0 0	1 1 1 0	1	rs	0 0 0 0	0	rd	d	d
	MOV.B @(d:32,Rs.W),@ERd+	0	1	7	1 0 0 0	1 1 1 0	1	rs	1 0 0 0	0	rd	d	d
	MOV.B @(d:32,Rs.W),@ERd-	0	1	7	1 0 0 0	1 1 1 0	1	rs	1 0 1 0	0	rd	d	d
	MOV.B @(d:32,Rs.W),@+ERd	0	1	7	1 0 0 0	1 1 1 0	1	rs	1 0 0 1	0	rd	d	d
	MOV.B @(d:32,Rs.W),@-ERd	0	1	7	1 0 0 0	1 1 1 0	1	rs	1 0 1 1	0	rd	d	d
	MOV.B @(d:32,Rs.W),@(d:2,ERd)	0	1	7	1 0 0 0	1 1 1 0	1	rs	0 0 d d	0	rd	d	d
	MOV.B @(d:32,Rs.W),@(d:16,ERd)	0	1	7	1 0 0 0	1 1 1 0	1	rs	1 1 0 0	0	rd	d	d d
	MOV.B @(d:32,Rs.W),@(d:32,ERd)	0	1	7	1 0 0 0	1 1 1 0	1	rs	1 1 0 0	1	rd	d	d d d
MOV.B @(d:32,Rs.W),@(d:16,Rd.B)	0	1	7	1 0 0 0	1 1 1 0	1	rs	1 1 0 1	0	rd	d	d d	
MOV.B @(d:32,Rs.W),@(d:16,Rd.W)	0	1	7	1 0 0 0	1 1 1 0	1	rs	1 1 1 0	0	rd	d	d d	



Instruction	Mnemonic	Opcode			EA Extension	Opcode			EA Extension						
		15	8	7		0	15	8		7	0				
MOV	MOV.B @(d:32.Rs.W),@(d:16.ERd.L)	0	1	7	1 0 0 0	1	1 1 1 0	1	rs	1 1 1 1	0	rd	d	d	d
	MOV.B @(d:32.Rs.W),@(d:32.Rd.B)	0	1	7	1 0 0 0	1	1 1 1 0	1	rs	1 1 0 1	1	rd	d	d	d
	MOV.B @(d:32.Rs.W),@(d:32.Rd.W)	0	1	7	1 0 0 0	1	1 1 1 0	1	rs	1 1 1 0	1	rd	d	d	d
	MOV.B @(d:32.Rs.W),@(d:32.ERd.L)	0	1	7	1 0 0 0	1	1 1 1 0	1	rs	1 1 1 1	1	rd	d	d	d
	MOV.B @(d:32.Rs.W),@aa:16	0	1	7	1 0 0 0	1	1 1 1 0	1	rs	0 1 0 0	0 0 0 0	0 0 0 0	d	d	a
	MOV.B @(d:32.Rs.W),@aa:32	0	1	7	1 0 0 0	1	1 1 1 0	1	rs	0 1 0 0	1 0 0 0	0 0 0 0	d	d	a
	MOV.B @(d:32.ERs.L),@ERd	0	1	7	1 0 0 0	1	1 1 1 1	1	rs	0 0 0 0	0 0 0 0	rd	d	d	
	MOV.B @(d:32.ERs.L),@ERd+	0	1	7	1 0 0 0	1	1 1 1 1	1	rs	1 0 0 0	0 0 0 0	rd	d	d	
	MOV.B @(d:32.ERs.L),@ERd-	0	1	7	1 0 0 0	1	1 1 1 1	1	rs	1 0 1 0	0 0 0 0	rd	d	d	
	MOV.B @(d:32.ERs.L),@+ERd	0	1	7	1 0 0 0	1	1 1 1 1	1	rs	1 0 0 1	0 0 0 0	rd	d	d	
	MOV.B @(d:32.ERs.L),@-ERd	0	1	7	1 0 0 0	1	1 1 1 1	1	rs	0 1 0 1	0 0 0 0	rd	d	d	
	MOV.B @(d:32.ERs.L),@(d:2.ERd)	0	1	7	1 0 0 0	1	1 1 1 1	1	rs	0 0 d d	0 0 0 0	rd	d	d	
	MOV.B @(d:32.ERs.L),@(d:16.ERd)	0	1	7	1 0 0 0	1	1 1 1 1	1	rs	1 1 0 0	0 0 0 0	rd	d	d	d
	MOV.B @(d:32.ERs.L),@(d:32.ERd)	0	1	7	1 0 0 0	1	1 1 1 1	1	rs	1 1 0 0	0 1 0 0	rd	d	d	d
	MOV.B @(d:32.ERs.L),@(d:16.Rd.B)	0	1	7	1 0 0 0	1	1 1 1 1	1	rs	1 1 0 1	0 0 0 0	rd	d	d	d
	MOV.B @(d:32.ERs.L),@(d:16.Rd.W)	0	1	7	1 0 0 0	1	1 1 1 1	1	rs	1 1 1 0	0 0 0 0	rd	d	d	d
	MOV.B @(d:32.ERs.L),@(d:16.ERd.L)	0	1	7	1 0 0 0	1	1 1 1 1	1	rs	1 1 1 1	1 0 0 0	rd	d	d	d
	MOV.B @(d:32.ERs.L),@(d:32.Rd.B)	0	1	7	1 0 0 0	1	1 1 1 1	1	rs	1 1 0 1	1 0 0 0	rd	d	d	d
	MOV.B @(d:32.ERs.L),@(d:32.Rd.W)	0	1	7	1 0 0 0	1	1 1 1 1	1	rs	1 1 1 0	1 0 0 0	rd	d	d	d
	MOV.B @(d:32.ERs.L),@(d:32.ERd.L)	0	1	7	1 0 0 0	1	1 1 1 1	1	rs	1 1 1 1	1 1 0 0	rd	d	d	d
	MOV.B @(d:32.ERs.L),@aa:16	0	1	7	1 0 0 0	1	1 1 1 1	1	rs	0 1 0 0	0 0 0 0	0 0 0 0	d	d	a
	MOV.B @(d:32.ERs.L),@aa:32	0	1	7	1 0 0 0	1	1 1 1 1	1	rs	0 1 0 0	1 0 0 0	0 0 0 0	d	d	a
	MOV.B @aa:16,@ERd	0	1	7	1 0 0 0	0	1 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	rd	a		
	MOV.B @aa:16,@ERd+	0	1	7	1 0 0 0	0	1 0 0 0	0 0 0 0	0 0 0 0	1 0 0 0	0 0 0 0	rd	a		
	MOV.B @aa:16,@ERd-	0	1	7	1 0 0 0	0	1 0 0 0	0 0 0 0	0 0 0 0	1 0 1 0	0 0 0 0	rd	a		
	MOV.B @aa:16,@+ERd	0	1	7	1 0 0 0	0	1 0 0 0	0 0 0 0	0 0 0 0	1 0 0 1	0 0 0 0	rd	a		
	MOV.B @aa:16,@-ERd	0	1	7	1 0 0 0	0	1 0 0 0	0 0 0 0	0 0 0 0	1 0 1 1	0 0 0 0	rd	a		
	MOV.B @aa:16,@(d:2.ERd)	0	1	7	1 0 0 0	0	1 0 0 0	0 0 0 0	0 0 0 0	0 0 d d	0 0 0 0	rd	a		
	MOV.B @aa:16,@(d:16.ERd)	0	1	7	1 0 0 0	0	1 0 0 0	0 0 0 0	0 0 0 0	1 1 0 0	0 0 0 0	rd	a	d	
	MOV.B @aa:16,@(d:32.ERd)	0	1	7	1 0 0 0	0	1 0 0 0	0 0 0 0	0 0 0 0	1 1 0 0	0 1 0 0	rd	a	d	d
MOV.B @aa:16,@(d:16.Rd.B)	0	1	7	1 0 0 0	0	1 0 0 0	0 0 0 0	0 0 0 0	1 1 0 1	0 0 0 0	rd	a	d		
MOV.B @aa:16,@(d:16.Rd.W)	0	1	7	1 0 0 0	0	1 0 0 0	0 0 0 0	0 0 0 0	1 1 1 0	0 0 0 0	rd	a	d		
MOV.B @aa:16,@(d:16.ERd.L)	0	1	7	1 0 0 0	0	1 0 0 0	0 0 0 0	0 0 0 0	1 1 1 1	1 0 0 0	rd	a	d		
MOV.B @aa:16,@(d:32.Rd.B)	0	1	7	1 0 0 0	0	1 0 0 0	0 0 0 0	0 0 0 0	1 1 0 1	1 1 0 0	rd	a	d	d	

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension												
		15	8	7	0		15	8	7	0													
MOV	MOV.B @aa:16,@(d:32,Rd,W)			0	1	7	1	0	0	0	0	1	0	0	0	rd	a	d	d				
	MOV.B @aa:16,@(d:32,ERd,L)			0	1	7	1	0	0	0	0	0	1	1	1	1	rd	a	d	d			
	MOV.B @aa:16,@aa:16			0	1	7	1	0	0	0	0	1	0	0	0	0	0	0	0	a	a		
	MOV.B @aa:16,@aa:32			0	1	7	1	0	0	0	0	1	0	0	0	0	1	0	0	0	a	a	
	MOV.B @aa:32,@ERd			0	1	7	1	0	0	0	0	1	0	0	0	0	0	0	0	0	rd	a	a
	MOV.B @aa:32,@ERd+			0	1	7	1	0	0	0	0	1	0	0	1	0	0	0	0	0	rd	a	a
	MOV.B @aa:32,@ERd-			0	1	7	1	0	0	0	0	1	0	0	1	0	0	0	0	0	rd	a	a
	MOV.B @aa:32,@+ERd			0	1	7	1	0	0	0	0	1	0	0	1	0	0	0	0	0	rd	a	a
	MOV.B @aa:32,@-ERd			0	1	7	1	0	0	0	0	1	0	0	1	0	0	0	0	0	rd	a	a
	MOV.B @aa:32,@(d:2,ERd)			0	1	7	1	0	0	0	0	1	0	0	0	0	0	0	0	0	rd	a	a
	MOV.B @aa:32,@(d:16,ERd)			0	1	7	1	0	0	0	0	1	0	0	0	0	0	0	0	0	rd	a	a
	MOV.B @aa:32,@(d:32,ERd)			0	1	7	1	0	0	0	0	1	0	0	0	1	0	0	0	0	rd	a	a
	MOV.B @aa:32,@(d:16,Rd,B)			0	1	7	1	0	0	0	0	1	0	0	0	1	0	0	0	0	rd	a	a
	MOV.B @aa:32,@(d:16,Rd,W)			0	1	7	1	0	0	0	0	1	0	0	0	1	0	0	0	0	rd	a	a
	MOV.B @aa:32,@(d:16,ERd,L)			0	1	7	1	0	0	0	0	1	0	0	0	1	1	1	0	0	rd	a	a
	MOV.B @aa:32,@(d:32,Rd,B)			0	1	7	1	0	0	0	0	1	0	0	0	1	0	1	1	0	rd	a	a
	MOV.B @aa:32,@(d:32,Rd,W)			0	1	7	1	0	0	0	0	1	0	0	0	1	0	0	1	1	rd	a	a
	MOV.B @aa:32,@(d:32,ERd,L)			0	1	7	1	0	0	0	0	1	0	0	0	1	1	1	1	1	rd	a	a
	MOV.B @aa:32,@aa:16			0	1	7	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	a	a
	MOV.B @aa:32,@aa:32			0	1	7	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	a	a
	MOV.W #x:c3,Rd										0	F	0	x	x	x	rd						
	MOV.W #x:c16,Rd										7	9	0								x		
	MOV.W #x:c4,@aa:16										6	B	1	1	0	1	x	x	x	x	a		
	MOV.W #x:c4,@aa:32										6	B	1	1	1	1	x	x	x	x	a	a	
MOV.W #x:c8,@ERd			0	1	5	1	1	0	1	0	0	0	0	rd	x	x	x	x	x	x	x		
MOV.W #x:c8,@ERd+			0	1	5	1	1	0	1	1	0	0	0	rd	x	x	x	x	x	x	x		
MOV.W #x:c8,@ERd-			0	1	5	1	1	0	1	0	1	0	0	rd	x	x	x	x	x	x	x		
MOV.W #x:c8,@+ERd			0	1	5	1	1	0	1	0	0	1	0	rd	x	x	x	x	x	x	x		
MOV.W #x:c8,@-ERd			0	1	5	1	1	0	1	0	1	0	0	rd	x	x	x	x	x	x	x		
MOV.W #x:c8,@(d:2,ERd)			0	1	5	1	1	0	1	0	0	0	0	rd	x	x	x	x	x	x	x		
MOV.W #x:c8,@(d:16,ERd)			0	1	5	1	1	0	1	1	0	0	0	rd	x	x	x	x	x	x	d		
MOV.W #x:c8,@(d:32,ERd)			0	1	5	1	1	0	1	1	0	0	1	rd	x	x	x	x	x	x	d		
MOV.W #x:c8,@(d:16,Rd,B)			0	1	5	1	1	0	1	1	0	1	0	rd	x	x	x	x	x	x	d		
MOV.W #x:c8,@(d:16,Rd,W)			0	1	5	1	1	0	1	1	1	0	0	rd	x	x	x	x	x	x	d		

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension															
		15	8	7	0		15	8	7	0																
MOV	MOV.W #hoc:8,@(d:16,ERd.L)				0	1	5	1	1	0	1	1	1	1	0	rd	x	x	x	x	x	x	x	d		
	MOV.W #hoc:8,@(d:32,Rd.B)				0	1	5	1	1	0	1	1	1	0	1	rd	x	x	x	x	x	x	x	d	d	
	MOV.W #hoc:8,@(d:32,Rd.W)				0	1	5	1	1	0	1	1	1	0	1	rd	x	x	x	x	x	x	x	d	d	
	MOV.W #hoc:8,@(d:32,ERd.L)				0	1	5	1	1	0	1	1	1	1	1	rd	x	x	x	x	x	x	x	d	d	
	MOV.W #hoc:8,@aa:16				0	1	5	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	MOV.W #hoc:8,@aa:32				0	1	5	1	1	0	1	0	0	1	0	0	0	0	x	x	x	x	x	x	a	a
	MOV.W #oc:16,@ERd				7	9	0	1	1	1	0	1	0	0	0	x	0	0	0	0	0	rd	0	0	0	0
	MOV.W #oc:16,@ERd+				7	9	0	1	1	1	0	1	0	0	0	x	1	0	0	0	0	rd	0	0	0	0
	MOV.W #oc:16,@ERd-				7	9	0	1	1	1	0	1	0	0	0	x	1	0	1	0	0	rd	0	0	0	0
	MOV.W #oc:16,@+ERd				7	9	0	1	1	1	0	1	0	0	0	x	1	0	0	1	0	rd	0	0	0	0
	MOV.W #oc:16,@-ERd				7	9	0	1	1	1	0	1	0	0	0	x	1	0	1	1	0	rd	0	0	0	0
	MOV.W #oc:16,@(d:2,ERd)				7	9	0	1	1	1	0	1	0	0	0	x	0	0	d	d	0	rd	0	0	0	0
	MOV.W #oc:16,@(d:16,ERd)				7	9	0	1	1	1	0	1	0	0	0	x	1	1	0	0	0	rd	0	0	0	0
	MOV.W #oc:16,@(d:32,ERd)				7	9	0	1	1	1	0	1	0	0	0	x	1	1	0	0	1	rd	0	0	0	0
	MOV.W #oc:16,@(d:16,Rd.B)				7	9	0	1	1	1	0	1	0	0	0	x	1	1	0	1	0	rd	0	0	0	0
	MOV.W #oc:16,@(d:16,Rd.W)				7	9	0	1	1	1	0	1	0	0	0	x	1	1	1	0	0	rd	0	0	0	0
	MOV.W #oc:16,@(d:16,ERd.L)				7	9	0	1	1	1	0	1	0	0	0	x	1	1	1	1	0	rd	0	0	0	0
	MOV.W #oc:16,@(d:32,Rd.B)				7	9	0	1	1	1	0	1	0	0	0	x	1	1	0	1	1	rd	0	0	0	0
	MOV.W #oc:16,@(d:32,Rd.W)				7	9	0	1	1	1	0	1	0	0	0	x	1	1	1	0	1	rd	0	0	0	0
	MOV.W #oc:16,@(d:32,ERd.L)				7	9	0	1	1	1	0	1	0	0	0	x	1	1	1	1	1	rd	0	0	0	0
MOV.W #oc:16,@aa:16				7	9	0	1	1	1	0	1	0	0	0	x	0	1	0	0	0	0	0	0	0	0	
MOV.W #oc:16,@aa:32				7	9	0	1	1	1	0	1	0	0	0	x	0	1	0	0	0	1	0	0	0	0	
MOV.W Rs,Rd																0		D		rs		rd				
MOV.W Rs,@ERd																6		9		rd		rs				
MOV.W Rs,@ERd+				0	1	5	0	0	1	1						6		D		rd		rs				
MOV.W Rs,@ERd-				0	1	5	0	0	0	1						6		D		rd		rs				
MOV.W Rs,@+ERd				0	1	5	0	0	1	0						6		D	1	rd		rs				
MOV.W Rs,@-ERd																6		D	1	rd		rs				
MOV.W Rs,@(d:2,ERd)				0	1	5	0	0	d	d						6		9		rd		rs				
MOV.W Rs,@(d:16,ERd)																6		F		rd		rs	d			
MOV.W Rs,@(d:32,ERd)				7	8	0	rd	0	0	0	0					6		B	1	0	1	0	rs	d		
MOV.W Rs,@(d:16,Rd.B)				0	1	5	0	0	0	1						6		F		rd		rs	d			
MOV.W Rs,@(d:16,Rd.W)				0	1	5	0	0	1	0						6		F		rd		rs	d			
MOV.W Rs,@(d:16,ERd.L)				0	1	5	0	0	1	1						6		F	1	rd		rs	d			

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension									
		15	8	7	0		15	8	7	0										
MOV	MOV.W Rs,@(d:32,Rd.B)		7	8	0	rd	0	0	0	1	6	B	1	0	1	0	rs	d	d	
	MOV.W Rs,@(d:32,Rd.W)		7	8	0	rd	0	0	1	0	6	B	1	0	1	0	rs	d	d	
	MOV.W Rs,@(d:32,ERd.L)		7	8	0	rd	0	0	1	1	6	B	1	0	1	0	rs	d	d	
	MOV.W Rs,@aa:16										6	B	1	0	0	0	rs	a		
	MOV.W Rs,@aa:32										6	B	1	0	1	0	rs	a	a	
	MOV.W @ERs,Rd										6	9	0				rs	rd		
	MOV.W @ERs+,Rd										6	D	0				rs	rd		
	MOV.W @ERs-,Rd		0	1	5	0	0	0	1	0	6	D	0				rs	rd		
	MOV.W @+ERs,Rd		0	1	5	0	0	0	1	1	6	D	0				rs	rd		
	MOV.W @-ERs,Rd		0	1	5	0	0	1	1		6	D	0				rs	rd		
	MOV.W @(d:2,ERs),Rd		0	1	5	0	0	d	d		6	9	0				rs	rd		
	MOV.W @(d:16,ERs),Rd										6	F	0				rs	rd	d	
	MOV.W @(d:32,ERs),Rd		7	8	0	rs	0	0	0	0	6	B	0	0	1	0	rd	d	d	
	MOV.W @(d:16,Rs.B),Rd		0	1	5	0	0	0	1		6	F	0				rs	rd	d	
	MOV.W @(d:16,Rs.W),Rd		0	1	5	0	0	1	0		6	F	0				rs	rd	d	
	MOV.W @(d:16,ERs.L),Rd		0	1	5	0	0	1	1		6	F	0				rs	rd	d	
	MOV.W @(d:32,Rs.B),Rd		7	8	0	rs	0	0	0	1	6	B	0	0	1	0	rd	d	d	
	MOV.W @(d:32,Rs.W),Rd		7	8	0	rs	0	0	1	0	6	B	0	0	1	0	rd	d	d	
	MOV.W @(d:32,ERs.L),Rd		7	8	0	rs	0	0	1	1	6	B	0	0	1	0	rd	d	d	
	MOV.W @aa:16,Rd										6	B	0	0	0	0	rd	a		
	MOV.W @aa:32,Rd										6	B	0	0	1	0	rd	a	a	
	MOV.W @ERs,@ERd		0	1	5	1	0	0	0		0	0	0	0	0	0	rs	0	0	0
	MOV.W @ERs,@ERd+		0	1	5	1	0	0	0		0	0	0	0	0	rs	1	0	0	
	MOV.W @ERs,@ERd-		0	1	5	1	0	0	0		0	0	0	0	rs	1	0	1		
	MOV.W @ERs,@+ERd		0	1	5	1	0	0	0		0	0	0	0	rs	1	0	0		
	MOV.W @ERs,@-ERd		0	1	5	1	0	0	0		0	0	0	0	rs	1	0	1		
	MOV.W @ERs,@(d:2,ERd)		0	1	5	1	0	0	0		0	0	0	0	rs	0	0	d		
	MOV.W @ERs,@(d:16,ERd)		0	1	5	1	0	0	0		0	0	0	0	rs	1	1	0		
	MOV.W @ERs,@(d:32,ERd)		0	1	5	1	0	0	0		0	0	0	0	rs	1	1	0		
	MOV.W @ERs,@(d:16,Rd.B)		0	1	5	1	0	0	0		0	0	0	0	rs	1	1	0		
	MOV.W @ERs,@(d:16,Rd.W)		0	1	5	1	0	0	0		0	0	0	0	rs	1	1	1		
	MOV.W @ERs,@(d:16,ERd.L)		0	1	5	1	0	0	0		0	0	0	0	rs	1	1	1		
MOV.W @ERs,@(d:32,Rd.B)		0	1	5	1	0	0	0		0	0	0	0	rs	1	1	0			
MOV.W @ERs,@(d:32,Rd.W)		0	1	5	1	0	0	0		0	0	0	0	rs	1	1	0			



Instruction	Mnemonic	Opcode			EA Ex- tension	Opcode			EA Extension																								
		15	8	7		0	15	8		7	0																						
MOV	MOV.W @ERs,@(d:32,ERd.L)				0	1		5	1	0	0	0	0	0	0	0	0	0	0	rs	1	1	1	1	1	1	rd			d	d		
	MOV.W @ERs,@aa:16				0	1		5	1	0	0	0	0	0	0	0	0	0	0	rs	0	1	0	0	0	0	0	0	0	0	0	0	a
	MOV.W @ERs,@aa:32				0	1		5	1	0	0	0	0	0	0	0	0	0	0	rs	0	1	0	0	1	1	0	0	0	0	0	a	a
	MOV.W @ERs+,@ERd				0	1		5	1	0	0	0	0	0	0	0	0	0	0	rs	0	0	0	0	0	0	0	rd					
	MOV.W @ERs+,@ERd+				0	1		5	1	0	0	0	0	1	0	0	0	0	0	rs	1	0	0	0	0	0	0	rd					
	MOV.W @ERs+,@ERd-				0	1		5	1	0	0	0	0	1	0	0	0	0	0	rs	1	0	1	0	0	0	0	rd					
	MOV.W @ERs+,@+ERd				0	1		5	1	0	0	0	0	1	0	0	0	0	0	rs	1	0	0	1	0	0	0	rd					
	MOV.W @ERs+,@-ERd				0	1		5	1	0	0	0	0	1	0	0	0	0	0	rs	1	0	1	1	0	0	0	rd					
	MOV.W @ERs+,@(d:2,ERd)				0	1		5	1	0	0	0	0	1	0	0	0	0	0	rs	0	0	d	d	0	0	rd						
	MOV.W @ERs+,@(d:16,ERd)				0	1		5	1	0	0	0	0	1	0	0	0	0	0	rs	1	1	0	0	0	0	rd			d			
	MOV.W @ERs+,@(d:32,ERd)				0	1		5	1	0	0	0	0	1	0	0	0	0	0	rs	1	1	0	0	1	1	rd			d	d		
	MOV.W @ERs+,@(d:16,Rd.B)				0	1		5	1	0	0	0	0	1	0	0	0	0	0	rs	1	1	0	1	0	0	rd			d			
	MOV.W @ERs+,@(d:16,Rd.W)				0	1		5	1	0	0	0	0	1	0	0	0	0	0	rs	1	1	1	0	0	0	rd			d			
	MOV.W @ERs+,@(d:16,ERd.L)				0	1		5	1	0	0	0	0	1	0	0	0	0	0	rs	1	1	1	1	0	0	rd			d			
	MOV.W @ERs+,@(d:32,Rd.B)				0	1		5	1	0	0	0	0	1	0	0	0	0	0	rs	1	1	0	1	1	1	rd			d	d		
	MOV.W @ERs+,@(d:32,Rd.W)				0	1		5	1	0	0	0	0	1	0	0	0	0	0	rs	1	1	1	0	1	1	rd			d	d		
	MOV.W @ERs+,@(d:32,ERd.L)				0	1		5	1	0	0	0	0	1	0	0	0	0	0	rs	1	1	1	1	1	1	rd			d	d		
	MOV.W @ERs+,@aa:16				0	1		5	1	0	0	0	0	1	0	0	0	0	0	rs	0	1	0	0	0	0	0	0	0	0	0	a	
	MOV.W @ERs+,@aa:32				0	1		5	1	0	0	0	0	1	0	0	0	0	0	rs	0	1	0	0	1	0	0	0	0	0	a	a	
	MOV.W @ERs-,@ERd				0	1		5	1	0	0	0	0	1	0	1	0	0	0	rs	0	0	0	0	0	0	0	rd					
	MOV.W @ERs-,@ERd+				0	1		5	1	0	0	0	0	1	0	1	0	0	0	rs	1	0	0	0	0	0	0	rd					
	MOV.W @ERs-,@ERd-				0	1		5	1	0	0	0	0	1	0	1	0	0	0	rs	1	0	1	0	0	0	0	rd					
	MOV.W @ERs-,@+ERd				0	1		5	1	0	0	0	0	1	0	1	0	0	0	rs	1	0	0	1	0	0	0	rd					
	MOV.W @ERs-,@-ERd				0	1		5	1	0	0	0	0	1	0	1	0	0	0	rs	1	0	1	1	0	0	0	rd					
	MOV.W @ERs-,@(d:2,ERd)				0	1		5	1	0	0	0	0	1	0	1	0	0	0	rs	0	0	d	d	0	0	rd						
	MOV.W @ERs-,@(d:16,ERd)				0	1		5	1	0	0	0	0	1	0	1	0	0	0	rs	1	1	0	0	0	0	rd			d			
	MOV.W @ERs-,@(d:32,ERd)				0	1		5	1	0	0	0	0	1	0	1	0	0	0	rs	1	1	0	0	1	1	rd			d	d		
	MOV.W @ERs-,@(d:16,Rd.B)				0	1		5	1	0	0	0	0	1	0	1	0	0	0	rs	1	1	0	1	0	0	rd			d			
	MOV.W @ERs-,@(d:16,Rd.W)				0	1		5	1	0	0	0	0	1	0	1	0	0	0	rs	1	1	1	0	0	0	rd			d			
	MOV.W @ERs-,@(d:16,ERd.L)				0	1		5	1	0	0	0	0	1	0	1	0	0	0	rs	1	1	1	1	0	0	rd			d			
	MOV.W @ERs-,@(d:32,Rd.B)				0	1		5	1	0	0	0	0	1	0	1	0	0	0	rs	1	1	0	1	1	1	rd			d	d		
	MOV.W @ERs-,@(d:32,Rd.W)				0	1		5	1	0	0	0	0	1	0	1	0	0	0	rs	1	1	1	0	1	1	rd			d	d		
MOV.W @ERs-,@(d:32,ERd.L)				0	1		5	1	0	0	0	0	1	0	1	0	0	0	rs	1	1	1	1	1	1	rd			d	d			
MOV.W @ERs-,@aa:16				0	1		5	1	0	0	0	0	1	0	1	0	0	0	rs	0	1	0	0	0	0	0	0	0	0	a			

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension																						
		15	8	7	0		15	8	7	0																							
MOV	MOV.W @ERs, @aa:32			0	1	5	1	0	0	0	0	1	0	1	0	0	rs	0	1	0	0	0	1	0	0	0	0	rd		a	a		
	MOV.W @+ERs, @ERd			0	1	5	1	0	0	0	0	1	0	0	1	0	rs	0	0	0	0	0	0	rd									
	MOV.W @+ERs, @ERd+			0	1	5	1	0	0	0	0	1	0	0	1	0	rs	1	0	0	0	0	0	rd									
	MOV.W @+ERs, @ERd-			0	1	5	1	0	0	0	0	1	0	0	1	0	rs	1	0	1	0	0	0	rd									
	MOV.W @+ERs, @+ERd			0	1	5	1	0	0	0	0	1	0	0	1	0	rs	1	0	0	1	0	0	rd									
	MOV.W @+ERs, @-ERd			0	1	5	1	0	0	0	0	1	0	0	1	0	rs	1	0	1	1	0	0	rd									
	MOV.W @+ERs, @(d:2,ERd)			0	1	5	1	0	0	0	0	1	0	0	1	0	rs	0	0	d	d	0	0	rd									
	MOV.W @+ERs, @(d:16,ERd)			0	1	5	1	0	0	0	0	1	0	0	1	0	rs	1	1	0	0	0	0	rd			d						
	MOV.W @+ERs, @(d:32,ERd)			0	1	5	1	0	0	0	0	1	0	0	1	0	rs	1	1	0	0	1	0	rd			d	d					
	MOV.W @+ERs, @(d:16,Rd,B)			0	1	5	1	0	0	0	0	1	0	0	1	0	rs	1	1	0	1	0	0	rd			d						
	MOV.W @+ERs, @(d:16,Rd,W)			0	1	5	1	0	0	0	0	1	0	0	1	0	rs	1	1	1	0	0	0	rd			d						
	MOV.W @+ERs, @(d:16,ERd,L)			0	1	5	1	0	0	0	0	1	0	0	1	0	rs	1	1	1	1	0	0	rd			d						
	MOV.W @+ERs, @(d:32,Rd,B)			0	1	5	1	0	0	0	0	1	0	0	1	0	rs	1	1	0	1	1	0	rd			d	d					
	MOV.W @+ERs, @(d:32,Rd,W)			0	1	5	1	0	0	0	0	1	0	0	1	0	rs	1	1	1	0	1	0	rd			d	d					
	MOV.W @+ERs, @(d:32,ERd,L)			0	1	5	1	0	0	0	0	1	0	0	1	0	rs	1	1	1	1	1	0	rd			d	d					
	MOV.W @+ERs, @aa:16			0	1	5	1	0	0	0	0	1	0	0	1	0	rs	0	1	0	0	0	0	0	0	0	0	0	0	0	0	a	
	MOV.W @+ERs, @aa:32			0	1	5	1	0	0	0	0	1	0	0	1	0	rs	0	1	0	0	1	0	0	0	0	0	0	0	0	0	a	a
	MOV.W @-ERs, @ERd			0	1	5	1	0	0	0	0	1	0	1	1	0	rs	0	0	0	0	0	0	rd									
	MOV.W @-ERs, @ERd+			0	1	5	1	0	0	0	0	1	0	1	1	0	rs	1	0	0	0	0	0	rd									
	MOV.W @-ERs, @ERd-			0	1	5	1	0	0	0	0	1	0	1	1	0	rs	1	0	1	0	0	0	rd									
	MOV.W @-ERs, @+ERd			0	1	5	1	0	0	0	0	1	0	1	1	0	rs	1	0	0	1	0	0	rd									
	MOV.W @-ERs, @-ERd			0	1	5	1	0	0	0	0	1	0	1	1	0	rs	1	0	1	1	0	0	rd									
	MOV.W @-ERs, @(d:2,ERd)			0	1	5	1	0	0	0	0	1	0	1	1	0	rs	0	0	d	d	0	0	rd									
	MOV.W @-ERs, @(d:16,ERd)			0	1	5	1	0	0	0	0	1	0	1	1	0	rs	1	1	0	0	0	0	rd			d						
	MOV.W @-ERs, @(d:32,ERd)			0	1	5	1	0	0	0	0	1	0	1	1	0	rs	1	1	0	0	1	0	rd			d	d					
	MOV.W @-ERs, @(d:16,Rd,B)			0	1	5	1	0	0	0	0	1	0	1	1	0	rs	1	1	0	1	0	0	rd			d						
	MOV.W @-ERs, @(d:16,Rd,W)			0	1	5	1	0	0	0	0	1	0	1	1	0	rs	1	1	1	0	0	0	rd			d						
	MOV.W @-ERs, @(d:16,ERd,L)			0	1	5	1	0	0	0	0	1	0	1	1	0	rs	1	1	1	1	0	0	rd			d						
	MOV.W @-ERs, @(d:32,Rd,B)			0	1	5	1	0	0	0	0	1	0	1	1	0	rs	1	1	0	1	1	0	rd			d	d					
	MOV.W @-ERs, @(d:32,Rd,W)			0	1	5	1	0	0	0	0	1	0	1	1	0	rs	1	1	1	0	1	0	rd			d	d					
	MOV.W @-ERs, @(d:32,ERd,L)			0	1	5	1	0	0	0	0	1	0	1	1	0	rs	1	1	1	1	1	0	rd			d	d					
	MOV.W @-ERs, @aa:16			0	1	5	1	0	0	0	0	1	0	1	1	0	rs	0	1	0	0	0	0	0	0	0	0	0	0	0	a		
MOV.W @-ERs, @aa:32			0	1	5	1	0	0	0	0	1	0	1	1	0	rs	0	1	0	0	1	0	0	0	0	0	0	0	0	a	a		
MOV.W @(d:2,ERs), @ERd			0	1	5	1	0	0	0	0	0	0	d	d	0	rs	0	0	0	0	0	0	rd										



Instruction	Mnemonic	Opcode				EA Ex- tension	Opcode				EA Extension																								
		15	8	7	0		15	8	7	0																									
MOV	MOV.W @(d:2,ERs),@ERd+				0		1		5		1	0	0	0	0		0	0	d	d		0	rs	1	0	0	0	0		rd					
	MOV.W @(d:2,ERs),@ERd-				0		1		5		1	0	0	0	0		0	0	d	d		0	rs	1	0	1	0	0		rd					
	MOV.W @(d:2,ERs),@+ERd				0		1		5		1	0	0	0	0		0	0	d	d		0	rs	1	0	0	1	0		rd					
	MOV.W @(d:2,ERs),@-ERd				0		1		5		1	0	0	0	0		0	0	d	d		0	rs	1	0	1	1	0		rd					
	MOV.W @(d:2,ERs),@(d:2,ERd)				0		1		5		1	0	0	0	0		0	0	d	d		0	rs	0	0	0	d	0		rd					
	MOV.W @(d:2,ERs),@(d:16,ERd)				0		1		5		1	0	0	0	0		0	0	d	d		0	rs	1	1	0	0	0		rd		d			
	MOV.W @(d:2,ERs),@(d:32,ERd)				0		1		5		1	0	0	0	0		0	0	d	d		0	rs	1	1	0	0	1		rd		d	d		
	MOV.W @(d:2,ERs),@(d:16,Rd.B)				0		1		5		1	0	0	0	0		0	0	d	d		0	rs	1	1	0	1	0		rd		d			
	MOV.W @(d:2,ERs),@(d:16,Rd.W)				0		1		5		1	0	0	0	0		0	0	d	d		0	rs	1	1	1	0	0		rd		d			
	MOV.W @(d:2,ERs),@(d:16,ERd.L)				0		1		5		1	0	0	0	0		0	0	d	d		0	rs	1	1	1	1	0		rd		d			
	MOV.W @(d:2,ERs),@(d:32,Rd.B)				0		1		5		1	0	0	0	0		0	0	d	d		0	rs	1	1	0	1	1		rd		d	d		
	MOV.W @(d:2,ERs),@(d:32,Rd.W)				0		1		5		1	0	0	0	0		0	0	d	d		0	rs	1	1	1	0	1		rd		d	d		
	MOV.W @(d:2,ERs),@(d:32,ERd.L)				0		1		5		1	0	0	0	0		0	0	d	d		0	rs	1	1	1	1	1		rd		d	d		
	MOV.W @(d:2,ERs),@aa:16				0		1		5		1	0	0	0	0		0	0	d	d		0	rs	0	1	0	0	0	0	0	0		a		
	MOV.W @(d:2,ERs),@aa:32				0		1		5		1	0	0	0	0		0	0	d	d		0	rs	0	1	0	0	1	0	0	0		a	a	
	MOV.W @(d:16,ERs),@ERd				0		1		5		1	0	0	0	0		1	1	0	0		0	rs	0	0	0	0	0		rd		d			
	MOV.W @(d:16,ERs),@ERd+				0		1		5		1	0	0	0	0		1	1	0	0		0	rs	1	0	0	0	0		rd		d			
	MOV.W @(d:16,ERs),@ERd-				0		1		5		1	0	0	0	0		1	1	0	0		0	rs	1	0	1	0	0		rd		d			
	MOV.W @(d:16,ERs),@+ERd				0		1		5		1	0	0	0	0		1	1	0	0		0	rs	1	0	0	1	0		rd		d			
	MOV.W @(d:16,ERs),@-ERd				0		1		5		1	0	0	0	0		1	1	0	0		0	rs	1	0	1	1	0		rd		d			
	MOV.W @(d:16,ERs),@(d:2,ERd)				0		1		5		1	0	0	0	0		1	1	0	0		0	rs	0	0	d	d	0		rd		d			
	MOV.W @(d:16,ERs),@(d:16,ERd)				0		1		5		1	0	0	0	0		1	1	0	0		0	rs	1	1	0	0	0		rd		d	d		
	MOV.W @(d:16,ERs),@(d:32,ERd)				0		1		5		1	0	0	0	0		1	1	0	0		0	rs	1	1	0	0	1		rd		d	d	d	
	MOV.W @(d:16,ERs),@(d:16,Rd.B)				0		1		5		1	0	0	0	0		1	1	0	0		0	rs	1	1	0	1	0		rd		d	d		
	MOV.W @(d:16,ERs),@(d:16,Rd.W)				0		1		5		1	0	0	0	0		1	1	0	0		0	rs	1	1	1	0	0		rd		d	d		
	MOV.W @(d:16,ERs),@(d:16,ERd.L)				0		1		5		1	0	0	0	0		1	1	0	0		0	rs	1	1	1	1	0		rd		d	d		
	MOV.W @(d:16,ERs),@(d:32,Rd.B)				0		1		5		1	0	0	0	0		1	1	0	0		0	rs	1	1	0	1	1		rd		d	d	d	
	MOV.W @(d:16,ERs),@(d:32,Rd.W)				0		1		5		1	0	0	0	0		1	1	0	0		0	rs	1	1	1	0	1		rd		d	d	d	
	MOV.W @(d:16,ERs),@(d:32,ERd.L)				0		1		5		1	0	0	0	0		1	1	0	0		0	rs	1	1	1	1	1		rd		d	d	d	
	MOV.W @(d:16,ERs),@aa:16				0		1		5		1	0	0	0	0		1	1	0	0		0	rs	0	1	0	0	0	0	0	0	0		a	
	MOV.W @(d:16,ERs),@aa:32				0		1		5		1	0	0	0	0		1	1	0	0		0	rs	0	1	0	0	1	0	0	0	0		a	a
	MOV.W @(d:32,ERs),@ERd				0		1		5		1	0	0	0	0		1	1	0	0		1	rs	0	0	0	0	0		rd		d	d		
	MOV.W @(d:32,ERs),@ERd+				0		1		5		1	0	0	0	0		1	1	0	0		1	rs	1	0	0	0	0		rd		d	d		
	MOV.W @(d:32,ERs),@ERd-				0		1		5		1	0	0	0	0		1	1	0	0		1	rs	1	0	1	0	0		rd		d	d		

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension																						
		15	8	7	0		15	8	7	0																							
MOV	MOV.W @(d:32,ERs),@+ERd				0		1		5	1	0	0	0		1	1	0	0	1	rs	1	0	0	1	0	rd	d	d					
	MOV.W @(d:32,ERs),@-ERd				0		1		5	1	0	0	0		1	1	0	0	1	rs	1	0	1	1	0	rd	d	d					
	MOV.W @(d:32,ERs),@(d:2,ERd)				0		1		5	1	0	0	0		1	1	0	0	1	rs	0	0	d	d	0	rd	d	d					
	MOV.W @(d:32,ERs),@(d:16,ERd)				0		1		5	1	0	0	0		1	1	0	0	1	rs	1	1	0	0	0	rd	d	d	d				
	MOV.W @(d:32,ERs),@(d:32,ERd)				0		1		5	1	0	0	0		1	1	0	0	1	rs	1	1	0	0	1	rd	d	d	d	d			
	MOV.W @(d:32,ERs),@(d:16,Rd,B)				0		1		5	1	0	0	0		1	1	0	0	1	rs	1	1	0	1	0	rd	d	d	d				
	MOV.W @(d:32,ERs),@(d:16,Rd,W)				0		1		5	1	0	0	0		1	1	0	0	1	rs	1	1	1	0	0	rd	d	d	d				
	MOV.W @(d:32,ERs),@(d:16,ERd,L)				0		1		5	1	0	0	0		1	1	0	0	1	rs	1	1	1	1	0	rd	d	d	d				
	MOV.W @(d:32,ERs),@(d:32,Rd,B)				0		1		5	1	0	0	0		1	1	0	0	1	rs	1	1	0	1	1	rd	d	d	d	d			
	MOV.W @(d:32,ERs),@(d:32,Rd,W)				0		1		5	1	0	0	0		1	1	0	0	1	rs	1	1	1	0	1	rd	d	d	d	d			
	MOV.W @(d:32,ERs),@(d:32,ERd,L)				0		1		5	1	0	0	0		1	1	0	0	1	rs	1	1	1	1	1	rd	d	d	d	d			
	MOV.W @(d:32,ERs),@aa:16				0		1		5	1	0	0	0		1	1	0	0	1	rs	0	1	0	0	0	0	0	0	0	d	d	a	
	MOV.W @(d:32,ERs),@aa:32				0		1		5	1	0	0	0		1	1	0	0	1	rs	0	1	0	0	1	0	0	0	0	d	d	a	a
	MOV.W @(d:16,Rs,B),@ERd				0		1		5	1	0	0	0		1	1	0	1	0	rs	0	0	0	0	0	rd	d						
	MOV.W @(d:16,Rs,B),@ERd+				0		1		5	1	0	0	0		1	1	0	1	0	rs	1	0	0	0	0	rd	d						
	MOV.W @(d:16,Rs,B),@ERd-				0		1		5	1	0	0	0		1	1	0	1	0	rs	1	0	1	0	0	rd	d						
	MOV.W @(d:16,Rs,B),@+ERd				0		1		5	1	0	0	0		1	1	0	1	0	rs	1	0	0	1	0	rd	d						
	MOV.W @(d:16,Rs,B),@-ERd				0		1		5	1	0	0	0		1	1	0	1	0	rs	1	0	1	1	0	rd	d						
	MOV.W @(d:16,Rs,B),@(d:2,ERd)				0		1		5	1	0	0	0		1	1	0	1	0	rs	0	0	d	d	0	rd	d						
	MOV.W @(d:16,Rs,B),@(d:16,ERd)				0		1		5	1	0	0	0		1	1	0	1	0	rs	1	1	0	0	0	rd	d		d				
	MOV.W @(d:16,Rs,B),@(d:32,ERd)				0		1		5	1	0	0	0		1	1	0	1	0	rs	1	1	0	0	1	rd	d		d	d			
	MOV.W @(d:16,Rs,B),@(d:16,Rd,B)				0		1		5	1	0	0	0		1	1	0	1	0	rs	1	1	0	1	0	rd	d		d				
	MOV.W @(d:16,Rs,B),@(d:16,Rd,W)				0		1		5	1	0	0	0		1	1	0	1	0	rs	1	1	1	0	0	rd	d		d				
	MOV.W @(d:16,Rs,B),@(d:16,ERd,L)				0		1		5	1	0	0	0		1	1	0	1	0	rs	1	1	1	1	0	rd	d		d				
	MOV.W @(d:16,Rs,B),@(d:32,Rd,B)				0		1		5	1	0	0	0		1	1	0	1	0	rs	1	1	0	1	1	rd	d		d	d			
	MOV.W @(d:16,Rs,B),@(d:32,Rd,W)				0		1		5	1	0	0	0		1	1	0	1	0	rs	1	1	1	0	1	rd	d		d	d			
	MOV.W @(d:16,Rs,B),@(d:32,ERd,L)				0		1		5	1	0	0	0		1	1	0	1	0	rs	1	1	1	1	1	rd	d		d	d			
	MOV.W @(d:16,Rs,B),@aa:16				0		1		5	1	0	0	0		1	1	0	1	0	rs	0	1	0	0	0	0	0	0	0	d	a		
	MOV.W @(d:16,Rs,B),@aa:32				0		1		5	1	0	0	0		1	1	0	1	0	rs	0	1	0	0	1	0	0	0	0	d	a	a	
	MOV.W @(d:16,Rs,W),@ERd				0		1		5	1	0	0	0		1	1	1	0	0	rs	0	0	0	0	0	rd	d						
	MOV.W @(d:16,Rs,W),@ERd+				0		1		5	1	0	0	0		1	1	1	0	0	rs	1	0	0	0	0	rd	d						
	MOV.W @(d:16,Rs,W),@ERd-				0		1		5	1	0	0	0		1	1	1	0	0	rs	1	0	1	0	0	rd	d						
	MOV.W @(d:16,Rs,W),@+ERd				0		1		5	1	0	0	0		1	1	1	0	0	rs	1	0	0	1	0	rd	d						
	MOV.W @(d:16,Rs,W),@-ERd				0		1		5	1	0	0	0		1	1	1	0	0	rs	1	0	1	1	0	rd	d						



Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension																						
		15	8	7	0		15	8	7	0																							
MOV	MOV.W @(d:16,Rs,W),@(d:2,ERd)				0			1		5		1	0	0	0	0		1	1	1	0	0	rs	0	0	d	d	0	rd	d			
	MOV.W @(d:16,Rs,W),@(d:16,ERd)				0			1		5		1	0	0	0		1	1	1	0	0	rs	1	1	0	0	0	rd	d		d		
	MOV.W @(d:16,Rs,W),@(d:32,ERd)				0			1		5		1	0	0	0		1	1	1	0	0	rs	1	1	0	0	1	rd	d		d		
	MOV.W @(d:16,Rs,W),@(d:16,Rd,B)				0			1		5		1	0	0	0		1	1	1	0	0	rs	1	1	0	1	0	rd	d		d		
	MOV.W @(d:16,Rs,W),@(d:16,Rd,W)				0			1		5		1	0	0	0		1	1	1	0	0	rs	1	1	1	0	0	rd	d		d		
	MOV.W @(d:16,Rs,W),@(d:16,ERd,L)				0			1		5		1	0	0	0		1	1	1	0	0	rs	1	1	1	1	0	rd	d		d		
	MOV.W @(d:16,Rs,W),@(d:32,Rd,B)				0			1		5		1	0	0	0		1	1	1	0	0	rs	1	1	0	1	1	rd	d		d		
	MOV.W @(d:16,Rs,W),@(d:32,Rd,W)				0			1		5		1	0	0	0		1	1	1	0	0	rs	1	1	1	0	1	rd	d		d		
	MOV.W @(d:16,Rs,W),@(d:32,ERd,L)				0			1		5		1	0	0	0		1	1	1	0	0	rs	1	1	1	1	1	rd	d		d		
	MOV.W @(d:16,Rs,W),@aa:16				0			1		5		1	0	0	0		1	1	1	0	0	rs	0	1	0	0	0	0	0	0	d	a	
	MOV.W @(d:16,Rs,W),@aa:32				0			1		5		1	0	0	0		1	1	1	0	0	rs	0	1	0	0	1	1	0	0	0	d	a
	MOV.W @(d:16,ERs,L),@ERd				0			1		5		1	0	0	0		1	1	1	0	0	rs	0	0	0	0	0	rd	d				
	MOV.W @(d:16,ERs,L),@ERd+				0			1		5		1	0	0	0		1	1	1	0	0	rs	1	0	0	0	0	rd	d				
	MOV.W @(d:16,ERs,L),@ERd-				0			1		5		1	0	0	0		1	1	1	0	0	rs	1	0	1	0	0	rd	d				
	MOV.W @(d:16,ERs,L),@+ERd				0			1		5		1	0	0	0		1	1	1	0	0	rs	1	0	0	1	0	rd	d				
	MOV.W @(d:16,ERs,L),@-ERd				0			1		5		1	0	0	0		1	1	1	0	0	rs	1	0	1	1	0	rd	d				
	MOV.W @(d:16,ERs,L),@(d:2,ERd)				0			1		5		1	0	0	0		1	1	1	0	0	rs	0	0	d	d	0	rd	d				
	MOV.W @(d:16,ERs,L),@(d:16,ERd)				0			1		5		1	0	0	0		1	1	1	0	0	rs	1	1	0	0	0	rd	d		d		
	MOV.W @(d:16,ERs,L),@(d:32,ERd)				0			1		5		1	0	0	0		1	1	1	0	0	rs	1	1	0	0	1	rd	d		d		
	MOV.W @(d:16,ERs,L),@(d:16,Rd,B)				0			1		5		1	0	0	0		1	1	1	0	0	rs	1	1	0	1	0	rd	d		d		
	MOV.W @(d:16,ERs,L),@(d:16,Rd,W)				0			1		5		1	0	0	0		1	1	1	0	0	rs	1	1	1	0	0	rd	d		d		
	MOV.W @ (d:16,ERs,L),@(d:16,ERd,L)				0			1		5		1	0	0	0		1	1	1	0	0	rs	1	1	1	1	0	rd	d		d		
	MOV.W @(d:16,ERs,L),@(d:32,Rd,B)				0			1		5		1	0	0	0		1	1	1	0	0	rs	1	1	0	1	1	rd	d		d		
	MOV.W @(d:16,ERs,L),@(d:32,Rd,W)				0			1		5		1	0	0	0		1	1	1	0	0	rs	1	1	1	0	1	rd	d		d		
	MOV.W @(d:16,ERs,L),@(d:32,ERd,L)				0			1		5		1	0	0	0		1	1	1	0	0	rs	1	1	1	1	1	rd	d		d		
	MOV.W @(d:16,ERs,L),@aa:16				0			1		5		1	0	0	0		1	1	1	0	0	rs	0	1	0	0	0	0	0	0	d	a	
	MOV.W @(d:16,ERs,L),@aa:32				0			1		5		1	0	0	0		1	1	1	0	0	rs	0	1	0	0	1	1	0	0	0	d	a
	MOV.W @(d:32,Rs,B),@ERd				0			1		5		1	0	0	0		1	1	0	1	1	rs	0	0	0	0	0	rd	d	d			
	MOV.W @(d:32,Rs,B),@ERd+				0			1		5		1	0	0	0		1	1	0	1	1	rs	1	0	0	0	0	rd	d	d			
	MOV.W @(d:32,Rs,B),@ERd-				0			1		5		1	0	0	0		1	1	0	1	1	rs	1	0	1	0	0	rd	d	d			
MOV.W @(d:32,Rs,B),@+ERd				0			1		5		1	0	0	0		1	1	0	1	1	rs	1	0	0	1	0	rd	d	d				
MOV.W @(d:32,Rs,B),@-ERd				0			1		5		1	0	0	0		1	1	0	1	1	rs	1	0	1	1	0	rd	d	d				

Instruction	Mnemonic	Opcode				EA Ex- tension	Opcode				EA Extension																	
		15	8	7	0		15	8	7	0																		
MOV	MOV.W @(d:32,Rs.B),@(d:2,ERd)				0	1	5	1	0	0	0	1	1	0	1	rs	0	0	d	d	0	rd	d	d				
	MOV.W @(d:32,Rs.B),@(d:16,ERd)				0	1	5	1	0	0	0	1	1	0	1	rs	1	1	0	0	0	rd	d	d	d			
	MOV.W @(d:32,Rs.B),@(d:32,ERd)				0	1	5	1	0	0	0	1	1	0	1	rs	1	1	0	0	1	rd	d	d	d			
	MOV.W @(d:32,Rs.B),@(d:16,Rd.B)				0	1	5	1	0	0	0	1	1	0	1	rs	1	1	0	1	0	rd	d	d	d			
	MOV.W @(d:32,Rs.B),@(d:16,Rd.W)				0	1	5	1	0	0	0	1	1	0	1	rs	1	1	1	0	0	rd	d	d	d			
	MOV.W @(d:32,Rs.B),@(d:16,ERd.L)				0	1	5	1	0	0	0	1	1	0	1	rs	1	1	1	1	0	rd	d	d	d			
	MOV.W @(d:32,Rs.B),@(d:32,Rd.B)				0	1	5	1	0	0	0	1	1	0	1	rs	1	1	0	1	1	rd	d	d	d			
	MOV.W @(d:32,Rs.B),@(d:32,Rd.W)				0	1	5	1	0	0	0	1	1	0	1	rs	1	1	1	0	1	rd	d	d	d			
	MOV.W @(d:32,Rs.B),@(d:32,ERd.L)				0	1	5	1	0	0	0	1	1	0	1	rs	1	1	1	1	1	rd	d	d	d			
	MOV.W @(d:32,Rs.B),@aa:16				0	1	5	1	0	0	0	1	1	0	1	rs	0	1	0	0	0	0	0	0	0	d	d	a
	MOV.W @(d:32,Rs.B),@aa:32				0	1	5	1	0	0	0	1	1	0	1	rs	0	1	0	0	1	0	0	0	0	d	d	a
	MOV.W @(d:32,Rs.W),@ERd				0	1	5	1	0	0	0	1	1	1	0	rs	0	0	0	0	0	rd	d	d				
	MOV.W @(d:32,Rs.W),@ERd+				0	1	5	1	0	0	0	1	1	1	0	rs	1	0	0	0	0	rd	d	d				
	MOV.W @(d:32,Rs.W),@ERd-				0	1	5	1	0	0	0	1	1	1	0	rs	1	0	1	0	0	rd	d	d				
	MOV.W @(d:32,Rs.W),@+ERd				0	1	5	1	0	0	0	1	1	1	0	rs	1	0	0	1	0	rd	d	d				
	MOV.W @(d:32,Rs.W),@-ERd				0	1	5	1	0	0	0	1	1	1	0	rs	1	0	1	1	0	rd	d	d				
	MOV.W @(d:32,Rs.W),@(d:2,ERd)				0	1	5	1	0	0	0	1	1	1	0	rs	0	0	d	d	0	rd	d	d				
	MOV.W @(d:32,Rs.W),@(d:16,ERd)				0	1	5	1	0	0	0	1	1	1	0	rs	1	1	0	0	0	rd	d	d	d			
	MOV.W @(d:32,Rs.W),@(d:32,ERd)				0	1	5	1	0	0	0	1	1	1	0	rs	1	1	0	0	1	rd	d	d	d			
	MOV.W @(d:32,Rs.W),@(d:16,Rd.B)				0	1	5	1	0	0	0	1	1	1	0	rs	1	1	0	1	0	rd	d	d	d			
	MOV.W @(d:32,Rs.W),@(d:16,Rd.W)				0	1	5	1	0	0	0	1	1	1	0	rs	1	1	1	0	0	rd	d	d	d			
	MOV.W @(d:32,Rs.W),@(d:16,ERd.L)				0	1	5	1	0	0	0	1	1	1	0	rs	1	1	1	1	0	rd	d	d	d			
	MOV.W @(d:32,Rs.W),@(d:32,Rd.B)				0	1	5	1	0	0	0	1	1	1	0	rs	1	1	0	1	1	rd	d	d	d			
	MOV.W @(d:32,Rs.W),@(d:32,Rd.W)				0	1	5	1	0	0	0	1	1	1	0	rs	1	1	1	0	1	rd	d	d	d			
	MOV.W @(d:32,Rs.W),@(d:32,ERd.L)				0	1	5	1	0	0	0	1	1	1	0	rs	1	1	1	1	1	rd	d	d	d			
	MOV.W @(d:32,Rs.W),@aa:16				0	1	5	1	0	0	0	1	1	1	0	rs	0	1	0	0	0	0	0	0	0	d	d	a
	MOV.W @(d:32,Rs.W),@aa:32				0	1	5	1	0	0	0	1	1	1	0	rs	0	1	0	0	1	0	0	0	0	d	d	a
	MOV.W @(d:32,ERs.L),@ERd				0	1	5	1	0	0	0	1	1	1	1	rs	0	0	0	0	0	rd	d	d				
	MOV.W @(d:32,ERs.L),@ERd+				0	1	5	1	0	0	0	1	1	1	1	rs	1	0	0	0	0	rd	d	d				
	MOV.W @(d:32,ERs.L),@ERd-				0	1	5	1	0	0	0	1	1	1	1	rs	1	0	1	0	0	rd	d	d				
	MOV.W @(d:32,ERs.L),@+ERd				0	1	5	1	0	0	0	1	1	1	1	rs	1	0	0	1	0	rd	d	d				
	MOV.W @(d:32,ERs.L),@-ERd				0	1	5	1	0	0	0	1	1	1	1	rs	1	0	1	1	0	rd	d	d				
MOV.W @(d:32,ERs.L),@(d:2,ERd)				0	1	5	1	0	0	0	1	1	1	1	rs	0	0	d	d	0	rd	d	d					
MOV.W @(d:32,ERs.L),@(d:16,ERd)				0	1	5	1	0	0	0	1	1	1	1	rs	1	1	0	0	0	rd	d	d	d				



Instruction	Mnemonic	Opcode			EA Extension	Opcode			EA Extension					
		15	8	7		0	15	8		7	0	EA Extension		
MOV	MOV.W @(d:32,ERs.L),@(d:32,ERd)	0	1	5	1 0 0 0	1 1 1 1	1	rs	1 1 0 0	1	rd	d	d	d
	MOV.W @(d:32,ERs.L),@(d:16,Rd.B)	0	1	5	1 0 0 0	1 1 1 1	1	rs	1 1 0 1	0	rd	d	d	d
	MOV.W @(d:32,ERs.L),@(d:16,Rd.W)	0	1	5	1 0 0 0	1 1 1 1	1	rs	1 1 1 0	0 0	rd	d	d	d
	MOV.W @(d:32,ERs.L),@(d:16,ERd.L)	0	1	5	1 0 0 0	1 1 1 1	1	rs	1 1 1 1	0	rd	d	d	d
	MOV.W @(d:32,ERs.L),@(d:32,Rd.B)	0	1	5	1 0 0 0	1 1 1 1	1	rs	1 1 0 1	1	rd	d	d	d
	MOV.W @(d:32,ERs.L),@(d:32,Rd.W)	0	1	5	1 0 0 0	1 1 1 1	1	rs	1 1 1 0	1	rd	d	d	d
	MOV.W @(d:32,ERs.L),@(d:32,ERd.L)	0	1	5	1 0 0 0	1 1 1 1	1	rs	1 1 1 1	1	rd	d	d	d
	MOV.W @aa:16,@aa:16	0	1	5	1 0 0 0	1 1 1 1	1	rs	0 1 0 0	0 0 0 0	0	d	d	a
	MOV.W @aa:16,@aa:32	0	1	5	1 0 0 0	1 1 1 1	1	rs	0 1 0 0	1 0 0 0	0	d	d	a
	MOV.W @aa:16,@ERd	0	1	5	1 0 0 0	0 1 0 0	0	0 0 0 0	0 0 0 0	0	rd	a		
	MOV.W @aa:16,@ERd+	0	1	5	1 0 0 0	0 1 0 0	0	0 0 0 0	1 0 0 0	0	rd	a		
	MOV.W @aa:16,@ERd-	0	1	5	1 0 0 0	0 1 0 0	0	0 0 0 0	1 0 1 0	0	rd	a		
	MOV.W @aa:16,@+ERd	0	1	5	1 0 0 0	0 1 0 0	0	0 0 0 0	1 0 0 1	0	rd	a		
	MOV.W @aa:16,@-ERd	0	1	5	1 0 0 0	0 1 0 0	0	0 0 0 0	1 0 1 1	0	rd	a		
	MOV.W @aa:16,@(d:2,ERd)	0	1	5	1 0 0 0	0 1 0 0	0	0 0 0 0	0 0 d d	0	rd	a		
	MOV.W @aa:16,@(d:16,ERd)	0	1	5	1 0 0 0	0 1 0 0	0	0 0 0 0	1 1 0 0	0	rd	a	d	
	MOV.W @aa:16,@(d:32,ERd)	0	1	5	1 0 0 0	0 1 0 0	0	0 0 0 0	1 1 0 0	1	rd	a	d	d
	MOV.W @aa:16,@(d:16,Rd.B)	0	1	5	1 0 0 0	0 1 0 0	0	0 0 0 0	1 1 0 1	0	rd	a	d	
	MOV.W @aa:16,@(d:16,Rd.W)	0	1	5	1 0 0 0	0 1 0 0	0	0 0 0 0	1 1 1 0	0	rd	a	d	
	MOV.W @aa:16,@(d:16,ERd.L)	0	1	5	1 0 0 0	0 1 0 0	0	0 0 0 0	1 1 1 1	0	rd	a	d	
	MOV.W @aa:16,@(d:32,Rd.B)	0	1	5	1 0 0 0	0 1 0 0	0	0 0 0 0	1 1 0 1	1	rd	a	d	d
	MOV.W @aa:16,@(d:32,Rd.W)	0	1	5	1 0 0 0	0 1 0 0	0	0 0 0 0	1 1 1 0	1	rd	a	d	d
	MOV.W @aa:16,@(d:32,ERd.L)	0	1	5	1 0 0 0	0 1 0 0	0	0 0 0 0	1 1 1 1	1	rd	a	d	d
	MOV.W @aa:16,@aa:16	0	1	5	1 0 0 0	0 1 0 0	0	0 0 0 0	0 1 0 0	0 0 0 0	0	a	a	
	MOV.W @aa:16,@aa:32	0	1	5	1 0 0 0	0 1 0 0	0	0 0 0 0	0 1 0 0	1 0 0 0	0	a	a	
	MOV.W @aa:32,@ERd	0	1	5	1 0 0 0	0 1 0 0	1	0 0 0 0	0 0 0 0	0	rd	a	a	
	MOV.W @aa:32,@ERd+	0	1	5	1 0 0 0	0 1 0 0	1	0 0 0 0	1 0 0 0	0	rd	a	a	
	MOV.W @aa:32,@ERd-	0	1	5	1 0 0 0	0 1 0 0	1	0 0 0 0	1 0 1 0	0	rd	a	a	
	MOV.W @aa:32,@+ERd	0	1	5	1 0 0 0	0 1 0 0	1	0 0 0 0	1 0 0 1	0	rd	a	a	
	MOV.W @aa:32,@-ERd	0	1	5	1 0 0 0	0 1 0 0	1	0 0 0 0	1 0 1 1	0	rd	a	a	
	MOV.W @aa:32,@(d:2,ERd)	0	1	5	1 0 0 0	0 1 0 0	1	0 0 0 0	0 0 d d	0	rd	a	a	
	MOV.W @aa:32,@(d:16,ERd)	0	1	5	1 0 0 0	0 1 0 0	1	0 0 0 0	1 1 0 0	0	rd	a	a	d

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension														
		15	8	7	0		15	8	7	0															
MOV	MOV.W @aa:32,@(d:32,ERd)			0	1	5	1	0	0	0	0	1	0	0	0	1	rd	a	a	d	d				
	MOV.W @aa:32,@(d:16,Rd.B)			0	1	5	1	0	0	0	0	1	0	0	0	1	1	0	1	0	rd	a	a	d	
	MOV.W @aa:32,@(d:16,Rd.W)			0	1	5	1	0	0	0	0	1	0	0	0	1	1	1	0	0	rd	a	a	d	
	MOV.W @aa:32,@(d:16,ERd.L)			0	1	5	1	0	0	0	0	1	0	0	0	1	1	1	1	0	rd	a	a	d	
	MOV.W @aa:32,@(d:32,Rd.B)			0	1	5	1	0	0	0	0	1	0	0	0	1	1	0	1	1	rd	a	a	d	
	MOV.W @aa:32,@(d:32,Rd.W)			0	1	5	1	0	0	0	0	1	0	0	0	1	1	1	0	1	rd	a	a	d	
	MOV.W @aa:32,@(d:32,ERd.L)			0	1	5	1	0	0	0	0	1	0	0	0	1	1	1	1	1	rd	a	a	d	
	MOV.W @aa:32,@aa:16			0	1	5	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
	MOV.W @aa:32,@aa:32			0	1	5	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
	MOV.L #xx:3,ERd										0					F	1	x	x	x	1	rd			
	MOV.L #xx:16,ERd											7			A	0	0	0	0	1	rd	x			
	MOV.L #xx:32,ERd												7		A	0	0	0	0	0	rd	x	x		
	MOV.L #xx:8,@ERd			0	1	0	1	1	0	1	0	0	0	0	0	rd	x	x	x	x	x	x	x	x	
	MOV.L #xx:8,@ERd+			0	1	0	1	1	0	1	0	1	0	0	0	rd	x	x	x	x	x	x	x		
	MOV.L #xx:8,@ERd-			0	1	0	1	1	0	1	0	1	0	1	0	rd	x	x	x	x	x	x			
	MOV.L #xx:8,@+ERd			0	1	0	1	1	0	1	0	1	0	1	0	rd	x	x	x	x	x	x			
	MOV.L #xx:8,@-ERd			0	1	0	1	1	0	1	0	1	0	1	0	rd	x	x	x	x	x	x			
	MOV.L #xx:8,@(d:2,ERd)			0	1	0	1	1	0	1	0	0	0	d	d	rd	x	x	x	x	x				
	MOV.L #xx:8,@(d:16,ERd)			0	1	0	1	1	0	1	0	1	1	0	0	rd	x	x	x	x	x	d			
	MOV.L #xx:8,@(d:32,ERd)			0	1	0	1	1	0	1	0	1	1	0	0	rd	x	x	x	x	x	d	d		
	MOV.L #xx:8,@(d:16,Rd.B)			0	1	0	1	1	0	1	0	1	1	0	1	rd	x	x	x	x	x	d			
	MOV.L #xx:8,@(d:16,Rd.W)			0	1	0	1	1	0	1	0	1	1	1	0	rd	x	x	x	x	x	d			
	MOV.L #xx:8,@(d:16,ERd.L)			0	1	0	1	1	0	1	0	1	1	1	1	rd	x	x	x	x	x	d			
	MOV.L #xx:8,@(d:32,Rd.B)			0	1	0	1	1	0	1	0	1	1	0	1	rd	x	x	x	x	x	d	d		
	MOV.L #xx:8,@(d:32,Rd.W)			0	1	0	1	1	0	1	0	1	1	1	0	rd	x	x	x	x	x	d	d		
	MOV.L #xx:8,@(d:32,ERd.L)			0	1	0	1	1	0	1	0	1	1	1	1	rd	x	x	x	x	x	d	d		
	MOV.L #xx:8,@aa:16			0	1	0	1	1	0	1	0	0	1	0	0	0	x	x	x	x	x	a			
	MOV.L #xx:8,@aa:32			0	1	0	1	1	0	1	0	0	1	0	0	0	x	x	x	x	x	a	a		
	MOV.L #xx:16,@ERd			7	A	0	1	1	1	1	1	0	0	0	0	rd	0	0	0	0	0	0	0	0	0
	MOV.L #xx:16,@ERd+			7	A	0	1	1	1	1	1	0	0	0	0	rd	0	0	0	0	0	0	0	0	0
MOV.L #xx:16,@ERd-			7	A	0	1	1	1	1	1	0	0	0	0	rd	0	0	0	0	0	0	0	0	0	
MOV.L #xx:16,@+ERd			7	A	0	1	1	1	1	1	0	0	0	0	rd	0	0	0	0	0	0	0	0	0	
MOV.L #xx:16,@-ERd			7	A	0	1	1	1	1	1	0	0	0	0	rd	0	0	0	0	0	0	0	0	0	
MOV.L #xx:16,@(d:2,ERd)			7	A	0	1	1	1	1	1	0	0	d	d	rd	0	0	0	0	0	0	0	0	0	



Instruction	Mnemonic	Opcode			Opcode			EA Ex- tension	Opcode			EA Extension				
		15	8	7	0	15	8		7	0	15		8	7	0	
MOV	MOV.L #xx:16,@(d:16,ERd)				7	A	0 1 1 1	1 1 0 0	x	1 1 0 0	0	rd	0 0 0 0	0 0 0 0	d	
	MOV.L #xx:16,@(d:32,ERd)				7	A	0 1 1 1	1 1 0 0	x	1 1 0 0	1	rd	0 0 0 0	0 0 0 0	d d	
	MOV.L #xx:16,@(d:16,Rd.B)				7	A	0 1 1 1	1 1 0 0	x	1 1 0 1	0	rd	0 0 0 0	0 0 0 0	d	
	MOV.L #xx:16,@(d:16,Rd.W)				7	A	0 1 1 1	1 1 0 0	x	1 1 1 0	0	rd	0 0 0 0	0 0 0 0	d	
	MOV.L #xx:16,@(d:16,ERd.L)				7	A	0 1 1 1	1 1 0 0	x	1 1 1 1	0	rd	0 0 0 0	0 0 0 0	d	
	MOV.L #xx:16,@(d:32,Rd.B)				7	A	0 1 1 1	1 1 0 0	x	1 1 0 1	1	rd	0 0 0 0	0 0 0 0	d d	
	MOV.L #xx:16,@(d:32,Rd.W)				7	A	0 1 1 1	1 1 0 0	x	1 1 1 0	1	rd	0 0 0 0	0 0 0 0	d d	
	MOV.L #xx:16,@(d:32,ERd.L)				7	A	0 1 1 1	1 1 0 0	x	1 1 1 1	1	rd	0 0 0 0	0 0 0 0	d d	
	MOV.L #xx:16,@aa:16				7	A	0 1 1 1	1 1 0 0	x	0 1 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	a	
	MOV.L #xx:16,@aa:32				7	A	0 1 1 1	1 1 0 0	x	0 1 0 0	1 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	a a	
	MOV.L #xx:32,@ERd				7	A	0 1 1 1	0 1 0 0	x x	0 0 0 0	0 0	rd	0 0 0 0	0 0 0 0		
	MOV.L #xx:32,@ERd+				7	A	0 1 1 1	0 1 0 0	x x	1 0 0 0	0 0	rd	0 0 0 0	0 0 0 0		
	MOV.L #xx:32,@ERd-				7	A	0 1 1 1	0 1 0 0	x x	1 0 1 0	0 0	rd	0 0 0 0	0 0 0 0		
	MOV.L #xx:32,@+ERd				7	A	0 1 1 1	0 1 0 0	x x	1 0 0 1	0 0	rd	0 0 0 0	0 0 0 0		
	MOV.L #xx:32,@-ERd				7	A	0 1 1 1	0 1 0 0	x x	1 0 1 1	0 0	rd	0 0 0 0	0 0 0 0		
	MOV.L #xx:32,@(d:2,ERd)				7	A	0 1 1 1	0 1 0 0	x x	0 0 d d	0 0	rd	0 0 0 0	0 0 0 0		
	MOV.L #xx:32,@(d:16,ERd)				7	A	0 1 1 1	0 1 0 0	x x	1 1 0 0	0 0	rd	0 0 0 0	0 0 0 0	d	
	MOV.L #xx:32,@(d:32,ERd)				7	A	0 1 1 1	0 1 0 0	x x	1 1 0 0	1	rd	0 0 0 0	0 0 0 0	d d	
	MOV.L #xx:32,@(d:16,Rd.B)				7	A	0 1 1 1	0 1 0 0	x x	1 1 0 1	0 0	rd	0 0 0 0	0 0 0 0	d	
	MOV.L #xx:32,@(d:16,Rd.W)				7	A	0 1 1 1	0 1 0 0	x x	1 1 1 0	0 0	rd	0 0 0 0	0 0 0 0	d	
	MOV.L #xx:32,@(d:16,ERd.L)				7	A	0 1 1 1	0 1 0 0	x x	1 1 1 1	0 0	rd	0 0 0 0	0 0 0 0	d	
	MOV.L #xx:32,@(d:32,Rd.B)				7	A	0 1 1 1	0 1 0 0	x x	1 1 0 1	1	rd	0 0 0 0	0 0 0 0	d d	
	MOV.L #xx:32,@(d:32,Rd.W)				7	A	0 1 1 1	0 1 0 0	x x	1 1 1 0	1	rd	0 0 0 0	0 0 0 0	d d	
	MOV.L #xx:32,@(d:32,ERd.L)				7	A	0 1 1 1	0 1 0 0	x x	1 1 1 1	1	rd	0 0 0 0	0 0 0 0	d d	
	MOV.L #xx:32,@aa:16				7	A	0 1 1 1	0 1 0 0	x x	0 1 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	a	
	MOV.L #xx:32,@aa:32				7	A	0 1 1 1	0 1 0 0	x x	0 1 0 0	1 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	a a	
	MOV.L ERs,ERd										0	F	1	rs	0	rd
MOV.L ERs,@ERd				0	1	0	0			6	9	1	rd	0	rs	
MOV.L ERs,@ERd+				0	1	0	0 0 0 1			6	D		rd	0	rs	
MOV.L ERs,@ERd-				0	1	0	0 0 0 1			6	D		rd	0	rs	
MOV.L ERs,@+ERd				0	1	0	0 0 1 0			6	D		rd	0	rs	
MOV.L ERs,@-ERd				0	1	0	0			6	D		rd	0	rs	
MOV.L ERs,@(d:2,ERd)				0	1	0	0 0 d d			6	9		rd	0	rs	
MOV.L ERs,@(d:16,ERd)				0	1	0	0			6	F		rd	0	rs d	

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension												
		15	8	7	0		15	8	7	0													
MOV	MOV.L ERs, @(d:32,ERd)*			7	8	1	rd	0	0	0	0	0	6	B	1	0	1	0	0	rs	d	d	
	MOV.L ERs, @(d:16,Rd.B)			0	1		0	0	0	0	0	1	6	F	1	rd	0			rs	d		
	MOV.L ERs, @(d:16,Rd.W)			0	1		0	0	0	1	0		6	F	1	rd	0			rs	d		
	MOV.L ERs, @(d:16,ERd.L)			0	1		0	0	0	1	1		6	F	1	rd	0			rs	d		
	MOV.L ERs, @(d:32,Rd.B)			7	8	1	rd	0	0	0	0	1	6	B	1	0	1	0	0	rs	d	d	
	MOV.L ERs, @(d:32,Rd.W)			7	8	1	rd	0	0	1	0		6	B	1	0	1	0	0	rs	d	d	
	MOV.L ERs, @(d:32,ERd.L)			7	8	1	rd	0	0	1	1		6	B	1	0	1	0	0	rs	d	d	
	MOV.L ERs, @aa:16			0	1		0					0	6	B	1	0	0	0	0	rs	a		
	MOV.L ERs, @aa:32			0	1		0				0	6	B	1	0	1	0	0	0	rs	a	a	
	MOV.L @ERs,ERd			0	1		0				0	6	9	0	rs	0				rd			
	MOV.L @ERs+,ERd			0	1		0				0	6	D	0	rs	0				rd			
	MOV.L @ERs-,ERd			0	1		0	0	0	1	0	6	D	0	rs	0				rd			
	MOV.L @+ERs,ERd			0	1		0	0	0	0	1	6	D	0	rs	0				rd			
	MOV.L @-ERs,ERd			0	1		0	0	0	1	1	6	D	0	rs	0				rd			
	MOV.L @(d:2,ERs),ERd			0	1		0	0	0	d	d	6	9	0	rs	0				rd			
	MOV.L @(d:16,ERs),ERd			0	1		0				0	6	F	0	rs	0				rd	d		
	MOV.L @(d:32,ERs),ERd*			7	8	1	rs	0	0	0	0	6	B	0	0	1	0	0	0	rd	d	d	
	MOV.L @(d:16,Rs.B),ERd			0	1		0	0	0	0	1	6	F	0	rs	0				rd	d		
	MOV.L @(d:16,Rs.W),ERd			0	1		0	0	0	1	0	6	F	0	rs	0				rd	d		
	MOV.L @(d:16,ERs.L),ERd			0	1		0	0	0	1	1	6	F	0	rs	0				rd	d		
	MOV.L @(d:32,Rs.B),ERd			7	8	1	rs	0	0	0	1	6	B	0	0	1	0	0	0	rd	d	d	
	MOV.L @(d:32,Rs.W),ERd			7	8	1	rs	0	0	1	0	6	B	0	0	1	0	0	0	rd	d	d	
	MOV.L @(d:32,ERs.L),ERd			7	8	1	rs	0	0	1	1	6	B	0	0	1	0	0	0	rd	d	d	
	MOV.L @aa:16,ERd			0	1		0				0	6	B	0	0	0	0	0	0	rd	a		
	MOV.L @aa:32,ERd			0	1		0				0	6	B	0	0	1	0	0	0	rd	a	a	
	MOV.L @ERs, @ERd			0	1		0	1	0	0	0	0	0	0	0	0	rs	0	0	0	0	0	rd
	MOV.L @ERs, @ERd+			0	1		0	1	0	0	0	0	0	0	0	0	rs	1	0	0	0	0	rd
	MOV.L @ERs, @ERd-			0	1		0	1	0	0	0	0	0	0	0	0	rs	1	0	1	0	0	rd
	MOV.L @ERs, @+ERd			0	1		0	1	0	0	0	0	0	0	0	0	rs	1	0	0	1	0	rd
	MOV.L @ERs, @-ERd			0	1		0	1	0	0	0	0	0	0	0	0	rs	1	0	1	1	0	rd
	MOV.L @ERs, @(d:2,ERd)			0	1		0	1	0	0	0	0	0	0	0	0	rs	0	0	d	d	0	rd
	MOV.L @ERs, @(d:16,ERd)			0	1		0	1	0	0	0	0	0	0	0	0	rs	1	1	0	0	0	rd
	MOV.L @ERs, @(d:32,ERd)			0	1		0	1	0	0	0	0	0	0	0	0	rs	1	1	0	0	1	rd
	MOV.L @ERs, @(d:16,Rd.B)			0	1		0	1	0	0	0	0	0	0	0	0	rs	1	1	0	1	0	rd



Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension																
		15	8	7	0		15	8	7	0																	
MOV	MOV.L @ERs, @(d:16,Rd,W)				0	1		0	1	0	0	0	0	0	0	0	rs	1	1	1	0	0	rd		d		
	MOV.L @ERs, @(d:16,ERd,L)				0	1		0	1	0	0	0	0	0	0	0	rs	1	1	1	1	0	rd		d		
	MOV.L @ERs, @(d:32,Rd,B)				0	1		0	1	0	0	0	0	0	0	0	rs	1	1	0	1	1	rd		d	d	
	MOV.L @ERs, @(d:32,Rd,W)				0	1		0	1	0	0	0	0	0	0	0	rs	1	1	1	0	1	rd		d	d	
	MOV.L @ERs, @(d:32,ERd,L)				0	1		0	1	0	0	0	0	0	0	0	rs	1	1	1	1	1	rd		d	d	
	MOV.L @ERs, @aa:16				0	1		0	1	0	0	0	0	0	0	0	rs	0	1	0	0	0	0	0	0	0	a
	MOV.L @ERs, @aa:32				0	1		0	1	0	0	0	0	0	0	0	rs	0	1	0	0	1	0	0	0	a	a
	MOV.L @ERs+, @ERd				0	1		0	1	0	0	0	0	0	0	0	rs	0	0	0	0	0	0	rd			
	MOV.L @ERs+, @ERd+				0	1		0	1	0	0	0	0	0	0	0	rs	1	0	0	0	0	0	rd			
	MOV.L @ERs+, @ERd-				0	1		0	1	0	0	0	0	0	0	0	rs	1	0	1	0	0	0	rd			
	MOV.L @ERs+, @+ERd				0	1		0	1	0	0	0	0	0	0	0	rs	1	0	0	1	0	0	rd			
	MOV.L @ERs+, @-ERd				0	1		0	1	0	0	0	0	0	0	0	rs	1	0	1	1	0	0	rd			
	MOV.L @ERs+, @(d:2,ERd)				0	1		0	1	0	0	0	0	0	0	0	rs	0	0	d	d	0	0	rd			
	MOV.L @ERs+, @(d:16,ERd)				0	1		0	1	0	0	0	0	0	0	0	rs	1	1	0	0	0	0	rd		d	
	MOV.L @ERs+, @(d:32,ERd)				0	1		0	1	0	0	0	0	0	0	0	rs	1	1	0	0	1	rd		d	d	
	MOV.L @ERs+, @(d:16,Rd,B)				0	1		0	1	0	0	0	0	0	0	0	rs	1	1	0	1	0	rd		d		
	MOV.L @ERs+, @(d:16,Rd,W)				0	1		0	1	0	0	0	0	0	0	0	rs	1	1	1	0	0	rd		d		
	MOV.L @ERs+, @(d:16,ERd,L)				0	1		0	1	0	0	0	0	0	0	0	rs	1	1	1	1	0	rd		d		
	MOV.L @ERs+, @(d:32,Rd,B)				0	1		0	1	0	0	0	0	0	0	0	rs	1	1	0	1	1	rd		d	d	
	MOV.L @ERs+, @(d:32,Rd,W)				0	1		0	1	0	0	0	0	0	0	0	rs	1	1	1	0	1	rd		d	d	
	MOV.L @ERs+, @(d:32,ERd,L)				0	1		0	1	0	0	0	0	0	0	0	rs	1	1	1	1	1	rd		d	d	
	MOV.L @ERs+, @aa:16				0	1		0	1	0	0	0	0	0	0	0	rs	0	1	0	0	0	0	0	0	0	a
	MOV.L @ERs+, @aa:32				0	1		0	1	0	0	0	0	0	0	0	rs	0	1	0	0	1	0	0	0	a	a
	MOV.L @ERs-, @ERd				0	1		0	1	0	0	0	0	0	0	0	rs	0	0	0	0	0	0	rd			
	MOV.L @ERs-, @ERd+				0	1		0	1	0	0	0	0	0	0	0	rs	1	0	0	0	0	0	rd			
	MOV.L @ERs-, @ERd-				0	1		0	1	0	0	0	0	0	0	0	rs	1	0	1	0	0	0	rd			
	MOV.L @ERs-, @+ERd				0	1		0	1	0	0	0	0	0	0	0	rs	1	0	0	1	0	0	rd			
	MOV.L @ERs-, @-ERd				0	1		0	1	0	0	0	0	0	0	0	rs	1	0	1	1	0	0	rd			
	MOV.L @ERs-, @(d:2,ERd)				0	1		0	1	0	0	0	0	0	0	0	rs	0	0	d	d	0	0	rd			
	MOV.L @ERs-, @(d:16,ERd)				0	1		0	1	0	0	0	0	0	0	0	rs	1	1	0	0	0	0	rd		d	
MOV.L @ERs-, @(d:32,ERd)				0	1		0	1	0	0	0	0	0	0	0	rs	1	1	0	0	1	rd		d	d		
MOV.L @ERs-, @(d:16,Rd,B)				0	1		0	1	0	0	0	0	0	0	0	rs	1	1	0	1	0	rd		d			
MOV.L @ERs-, @(d:16,Rd,W)				0	1		0	1	0	0	0	0	0	0	0	rs	1	1	1	0	0	rd		d			
MOV.L @ERs-, @(d:16,ERd,L)				0	1		0	1	0	0	0	0	0	0	0	rs	1	1	1	1	0	rd		d			

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension																			
		15	8	7	0		15	8	7	0																				
MOV	MOV.L @ERs-,@(d:32,Rd,B)				0	1			0	1	0	0	0	0	0	0	0	rs	1	1	0	1	1	1	rd			d	d	
	MOV.L @ERs-,@(d:32,Rd,W)				0	1			0	1	0	0	0	0	0	0	0	rs	1	1	1	0	1	1	rd			d	d	
	MOV.L @ERs-,@(d:32,ERd,L)				0	1			0	1	0	0	0	0	0	0	0	rs	1	1	1	1	1	1	rd			d	d	
	MOV.L @ERs-,@aa:16				0	1			0	1	0	0	0	0	0	0	0	rs	0	1	0	0	0	0	0	0			a	
	MOV.L @ERs-,@aa:32				0	1			0	1	0	0	0	0	0	0	0	rs	0	1	0	0	1	0	0	0			a	a
	MOV.L @+ERs,@ERd				0	1			0	1	0	0	0	0	0	0	0	rs	0	0	0	0	0	0	rd					
	MOV.L @+ERs,@ERd+				0	1			0	1	0	0	0	0	0	0	0	rs	1	0	0	0	0	0	rd					
	MOV.L @+ERs,@ERd-				0	1			0	1	0	0	0	0	0	0	0	rs	1	0	1	0	0	0	rd					
	MOV.L @+ERs,@+ERd				0	1			0	1	0	0	0	0	0	0	0	rs	1	0	0	1	0	0	rd					
	MOV.L @+ERs,@-ERd				0	1			0	1	0	0	0	0	0	0	0	rs	1	0	1	1	0	0	rd					
	MOV.L @+ERs,@(d:2,ERd)				0	1			0	1	0	0	0	0	0	0	0	rs	0	0	d	d	0	0	rd					
	MOV.L @+ERs,@(d:16,ERd)				0	1			0	1	0	0	0	0	0	0	0	rs	1	1	0	0	0	0	rd			d		
	MOV.L @+ERs,@(d:32,ERd)				0	1			0	1	0	0	0	0	0	0	0	rs	1	1	0	0	1	1	rd			d	d	
	MOV.L @+ERs,@(d:16,Rd,B)				0	1			0	1	0	0	0	0	0	0	0	rs	1	1	0	1	0	0	rd			d		
	MOV.L @+ERs,@(d:16,Rd,W)				0	1			0	1	0	0	0	0	0	0	0	rs	1	1	1	0	0	0	rd			d		
	MOV.L @+ERs,@(d:16,ERd,L)				0	1			0	1	0	0	0	0	0	0	0	rs	1	1	1	1	0	0	rd			d		
	MOV.L @+ERs,@(d:32,Rd,B)				0	1			0	1	0	0	0	0	0	0	0	rs	1	1	0	1	1	1	rd			d	d	
	MOV.L @+ERs,@(d:32,Rd,W)				0	1			0	1	0	0	0	0	0	0	0	rs	1	1	1	0	1	1	rd			d	d	
	MOV.L @+ERs,@(d:32,ERd,L)				0	1			0	1	0	0	0	0	0	0	0	rs	1	1	1	1	1	1	rd			d	d	
	MOV.L @+ERs,@aa:16				0	1			0	1	0	0	0	0	0	0	0	rs	0	1	0	0	0	0	0	0			a	
	MOV.L @+ERs,@aa:32				0	1			0	1	0	0	0	0	0	0	0	rs	0	1	0	0	1	0	0	0			a	a
	MOV.L @-ERs,@ERd				0	1			0	1	0	0	0	0	0	0	0	rs	0	0	0	0	0	0	rd					
	MOV.L @-ERs,@ERd+				0	1			0	1	0	0	0	0	0	0	0	rs	1	0	0	0	0	0	rd					
	MOV.L @-ERs,@ERd-				0	1			0	1	0	0	0	0	0	0	0	rs	1	0	1	0	0	0	rd					
	MOV.L @-ERs,@+ERd				0	1			0	1	0	0	0	0	0	0	0	rs	1	0	0	1	0	0	rd					
	MOV.L @-ERs,@-ERd				0	1			0	1	0	0	0	0	0	0	0	rs	1	0	1	1	0	0	rd					
	MOV.L @-ERs,@(d:2,ERd)				0	1			0	1	0	0	0	0	0	0	0	rs	0	0	d	d	0	0	rd					
	MOV.L @-ERs,@(d:16,ERd)				0	1			0	1	0	0	0	0	0	0	0	rs	1	1	0	0	0	0	rd			d		
	MOV.L @-ERs,@(d:32,ERd)				0	1			0	1	0	0	0	0	0	0	0	rs	1	1	0	0	1	1	rd			d	d	
	MOV.L @-ERs,@(d:16,Rd,B)				0	1			0	1	0	0	0	0	0	0	0	rs	1	1	0	1	0	0	rd			d		
	MOV.L @-ERs,@(d:16,Rd,W)				0	1			0	1	0	0	0	0	0	0	0	rs	1	1	1	0	0	0	rd			d		
	MOV.L @-ERs,@(d:16,ERd,L)				0	1			0	1	0	0	0	0	0	0	0	rs	1	1	1	1	0	0	rd			d		
	MOV.L @-ERs,@(d:32,Rd,B)				0	1			0	1	0	0	0	0	0	0	0	rs	1	1	0	1	1	1	rd			d	d	
	MOV.L @-ERs,@(d:32,Rd,W)				0	1			0	1	0	0	0	0	0	0	0	rs	1	1	1	0	1	1	rd			d	d	



Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension																			
		15	8	7	0		15	8	7	0																				
MOV	MOV.L @-ERs,@(d:32,ERd.L)				0	1			0	1	0	0	0	0	0	0	0	rs	1	1	1	1	1	rd		d	d			
	MOV.L @-ERs,@aa:16				0	1			0	1	0	1	1	0	rs	0	1	0	0	0	0	0	0	0	0	0	0		a	
	MOV.L @-ERs,@aa:32				0	1			0	1	0	1	1	0	rs	0	1	0	0	1	1	0	0	0	0	0	0		a	a
	MOV.L @(d:2,ERs),@ERd				0	1			0	1	0	0	0	0	rs	0	0	d	d	0	rd									
	MOV.L @(d:2,ERs),@ERd+				0	1			0	1	0	0	0	0	rs	0	0	d	d	0	rd									
	MOV.L @(d:2,ERs),@ERd-				0	1			0	1	0	0	0	0	rs	0	0	d	d	0	rd									
	MOV.L @(d:2,ERs),@+ERd				0	1			0	1	0	0	0	0	rs	0	0	d	d	0	rd									
	MOV.L @(d:2,ERs),@-ERd				0	1			0	1	0	0	0	0	rs	0	0	d	d	0	rd									
	MOV.L @(d:2,ERs),@(d:2,ERd)				0	1			0	1	0	0	0	0	rs	0	0	d	d	0	rd									
	MOV.L @(d:2,ERs),@(d:16,ERd)				0	1			0	1	0	0	0	0	rs	0	0	d	d	0	rd							d		
	MOV.L @(d:2,ERs),@(d:32,ERd)				0	1			0	1	0	0	0	0	rs	0	0	d	d	0	rd							d	d	
	MOV.L @(d:2,ERs),@(d:16,Rd.B)				0	1			0	1	0	0	0	0	rs	0	0	d	d	0	rd							d		
	MOV.L @(d:2,ERs),@(d:16,Rd.W)				0	1			0	1	0	0	0	0	rs	0	0	d	d	0	rd							d		
	MOV.L @(d:2,ERs),@(d:16,ERd.L)				0	1			0	1	0	0	0	0	rs	0	0	d	d	0	rd							d		
	MOV.L @(d:2,ERs),@(d:32,Rd.B)				0	1			0	1	0	0	0	0	rs	0	0	d	d	0	rd							d	d	
	MOV.L @(d:2,ERs),@(d:32,Rd.W)				0	1			0	1	0	0	0	0	rs	0	0	d	d	0	rd							d	d	
	MOV.L @(d:2,ERs),@(d:32,ERd.L)				0	1			0	1	0	0	0	0	rs	0	0	d	d	0	rd							d	d	
	MOV.L @(d:2,ERs),@aa:16				0	1			0	1	0	0	0	0	rs	0	0	d	d	0	rd							a		
	MOV.L @(d:2,ERs),@aa:32				0	1			0	1	0	0	0	0	rs	0	0	d	d	0	rd							a	a	
	MOV.L @(d:16,ERs),@ERd				0	1			0	1	0	0	0	1	rs	1	1	0	0	0	rd							d		
	MOV.L @(d:16,ERs),@ERd+				0	1			0	1	0	0	0	1	rs	1	1	0	0	0	rd							d		
	MOV.L @(d:16,ERs),@ERd-				0	1			0	1	0	0	0	1	rs	1	1	0	0	0	rd							d		
	MOV.L @(d:16,ERs),@+ERd				0	1			0	1	0	0	0	1	rs	1	1	0	0	0	rd							d		
	MOV.L @(d:16,ERs),@-ERd				0	1			0	1	0	0	0	1	rs	1	1	0	0	0	rd							d		
	MOV.L @(d:16,ERs),@(d:2,ERd)				0	1			0	1	0	0	0	1	rs	1	1	0	0	0	rd							d		
	MOV.L @(d:16,ERs),@(d:16,ERd)				0	1			0	1	0	0	0	1	rs	1	1	0	0	0	rd							d		
MOV.L @(d:16,ERs),@(d:32,ERd)				0	1			0	1	0	0	0	1	rs	1	1	0	0	1	rd							d	d		
MOV.L @(d:16,ERs),@(d:16,Rd.B)				0	1			0	1	0	0	0	1	rs	1	1	0	0	0	rd							d	d		
MOV.L @(d:16,ERs),@(d:16,Rd.W)				0	1			0	1	0	0	0	1	rs	1	1	0	0	0	rd							d	d		
MOV.L @(d:16,ERs),@(d:16,ERd.L)				0	1			0	1	0	0	0	1	rs	1	1	0	0	0	rd							d	d		
MOV.L @(d:16,ERs),@(d:32,Rd.B)				0	1			0	1	0	0	0	1	rs	1	1	0	0	1	rd							d	d		
MOV.L @(d:16,ERs),@(d:32,Rd.W)				0	1			0	1	0	0	0	1	rs	1	1	0	0	1	rd							d	d		
MOV.L @(d:16,ERs),@(d:32,ERd.L)				0	1			0	1	0	0	0	1	rs	1	1	0	0	1	rd							d	d		
MOV.L @(d:16,ERs),@aa:16				0	1			0	1	0	0	0	1	rs	0	1	0	0	0	rd							d	a		

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension																						
		15	8	7	0		15	8	7	0																							
MOV	MOV.L @(d:16,ERs),@aa:32			0	1	0	1	0	0	0	0	1	1	0	0	0	rs	0	1	0	0	0	1	0	0	0	0	rd	d	a	a		
	MOV.L @(d:32,ERs),@ERd			0	1	0	1	0	0	0	0	1	1	0	0	1	rs	0	0	0	0	0	0	rd	d	d							
	MOV.L @(d:32,ERs),@ERd+			0	1	0	1	0	0	0	0	1	1	0	0	1	rs	1	0	0	0	0	0	rd	d	d							
	MOV.L @(d:32,ERs),@ERd-			0	1	0	1	0	0	0	0	1	1	0	0	1	rs	1	0	1	0	0	0	rd	d	d							
	MOV.L @(d:32,ERs),@+ERd			0	1	0	0	1	0	0	0	1	1	0	0	1	rs	1	0	0	1	0	0	rd	d	d							
	MOV.L @(d:32,ERs),@-ERd			0	1	0	1	0	0	0	0	1	1	0	0	1	rs	1	0	1	1	0	0	rd	d	d							
	MOV.L @(d:32,ERs),@(d:2,ERd)			0	1	0	1	0	0	0	0	1	1	0	0	1	rs	0	0	d	d	0	0	rd	d	d							
	MOV.L @(d:32,ERs),@(d:16,ERd)			0	1	0	1	0	0	0	0	1	1	0	0	1	rs	1	1	0	0	0	0	rd	d	d	d						
	MOV.L @(d:32,ERs),@(d:32,ERd)			0	1	0	1	0	0	0	0	1	1	0	0	1	rs	1	1	0	0	1	0	rd	d	d	d	d					
	MOV.L @(d:32,ERs),@(d:16,Rd,B)			0	1	0	1	0	0	0	0	1	1	0	0	1	rs	1	1	0	1	0	0	rd	d	d	d						
	MOV.L @(d:32,ERs),@(d:16,Rd,W)			0	1	0	0	1	0	0	0	1	1	0	0	1	rs	1	1	1	0	0	0	rd	d	d	d						
	MOV.L @(d:32,ERs),@(d:16,ERd,L)			0	1	0	1	0	0	0	0	1	1	0	0	1	rs	1	1	1	1	0	0	rd	d	d	d						
	MOV.L @(d:32,ERs),@(d:32,Rd,B)			0	1	0	1	0	0	0	0	1	1	0	0	1	rs	1	1	0	1	1	0	rd	d	d	d	d					
	MOV.L @(d:32,ERs),@(d:32,Rd,W)			0	1	0	1	0	0	0	0	1	1	0	0	1	rs	1	1	1	0	1	0	rd	d	d	d	d					
	MOV.L @(d:32,ERs),@(d:32,ERd,L)			0	1	0	1	0	0	0	0	1	1	0	0	1	rs	1	1	1	1	1	0	rd	d	d	d	d					
	MOV.L @(d:32,ERs),@aa:16			0	1	0	1	0	0	0	0	1	1	0	0	1	rs	0	1	0	0	0	0	0	0	0	0	0	d	d	a		
	MOV.L @(d:32,ERs),@aa:32			0	1	0	0	1	0	0	0	1	1	0	0	1	rs	0	1	0	0	1	0	0	0	0	0	0	d	d	a	a	
	MOV.L @(d:16,Rs,B),@ERd			0	1	0	0	1	0	0	0	1	1	0	1	0	rs	0	0	0	0	0	0	rd	d								
	MOV.L @(d:16,Rs,B),@ERd+			0	1	0	0	1	0	0	0	1	1	0	1	0	rs	1	0	0	0	0	0	rd	d								
	MOV.L @(d:16,Rs,B),@ERd-			0	1	0	0	1	0	0	0	1	1	0	1	0	rs	1	0	1	0	0	0	rd	d								
	MOV.L @(d:16,Rs,B),@+ERd			0	1	0	0	1	0	0	0	1	1	0	1	0	rs	1	0	0	1	0	0	rd	d								
	MOV.L @(d:16,Rs,B),@-ERd			0	1	0	0	1	0	0	0	1	1	0	1	0	rs	1	0	1	1	0	0	rd	d								
	MOV.L @(d:16,Rs,B),@(d:2,ERd)			0	1	0	0	1	0	0	0	1	1	0	1	0	rs	0	0	d	d	0	0	rd	d								
	MOV.L @(d:16,Rs,B),@(d:16,ERd)			0	1	0	0	1	0	0	0	1	1	0	1	0	rs	1	1	0	0	0	0	rd	d			d					
	MOV.L @(d:16,Rs,B),@(d:32,ERd)			0	1	0	0	1	0	0	0	1	1	0	1	0	rs	1	1	0	0	1	0	rd	d			d	d				
	MOV.L @(d:16,Rs,B),@(d:16,Rd,B)			0	1	0	0	1	0	0	0	1	1	0	1	0	rs	1	1	0	1	0	0	rd	d			d					
	MOV.L @(d:16,Rs,B),@(d:16,Rd,W)			0	1	0	0	1	0	0	0	1	1	0	1	0	rs	1	1	1	0	0	0	rd	d			d					
	MOV.L @(d:16,Rs,B),@(d:16,ERd,L)			0	1	0	0	1	0	0	0	1	1	0	1	0	rs	1	1	1	1	0	0	rd	d			d					
	MOV.L @(d:16,Rs,B),@(d:32,Rd,B)			0	1	0	0	1	0	0	0	1	1	0	1	0	rs	1	1	0	1	1	0	rd	d			d	d				
	MOV.L @(d:16,Rs,B),@(d:32,Rd,W)			0	1	0	0	1	0	0	0	1	1	0	1	0	rs	1	1	1	0	1	1	rd	d			d	d				
	MOV.L @(d:16,Rs,B),@(d:32,ERd,L)			0	1	0	0	1	0	0	0	1	1	0	1	0	rs	1	1	1	1	1	0	rd	d			d	d				
	MOV.L @(d:16,Rs,B),@aa:16			0	1	0	0	1	0	0	0	1	1	0	1	0	rs	0	1	0	0	0	0	0	0	0	0	d		a			
	MOV.L @(d:16,Rs,B),@aa:32			0	1	0	0	1	0	0	0	1	1	0	1	0	rs	0	1	0	0	1	0	0	0	0	d		a	a			
	MOV.L @(d:16,Rs,W),@ERd			0	1	0	0	1	0	0	0	1	1	1	0	0	rs	0	0	0	0	0	0	rd	d								



Instruction	Mnemonic	Opcode			Opcode			EA Extension	Opcode			EA Extension													
		15	8	7	0	15	8		7	0	15		8	7	0										
MOV	MOV.L @(d:16,Rs,W),@ERd+				0	1	0	1 0 0 0		1	1	1	0	rs	1	0	0	0	0	rd	d				
	MOV.L @(d:16,Rs,W),@ERd-				0	1	0	1 0 0 0		1	1	1	0	0	rs	1	0	1	0	0	rd	d			
	MOV.L @(d:16,Rs,W),@+ERd				0	1	0	1 0 0 0		1	1	1	0	0	rs	1	0	0	1	0	rd	d			
	MOV.L @(d:16,Rs,W),@-ERd				0	1	0	1 0 0 0		1	1	1	0	0	rs	1	0	1	1	0	rd	d			
	MOV.L @(d:16,Rs,W),@(d:2,ERd)				0	1	0	1 0 0 0		1	1	1	0	0	rs	0	0	d	d	0	rd	d			
	MOV.L @(d:16,Rs,W),@(d:16,ERd)				0	1	0	1 0 0 0		1	1	1	0	0	rs	1	1	0	0	0	rd	d	d		
	MOV.L @(d:16,Rs,W),@(d:32,ERd)				0	1	0	1 0 0 0		1	1	1	0	0	rs	1	1	0	0	1	rd	d	d		
	MOV.L @(d:16,Rs,W),@(d:16,Rd,B)				0	1	0	1 0 0 0		1	1	1	0	0	rs	1	1	0	1	0	rd	d	d		
	MOV.L @(d:16,Rs,W),@(d:16,Rd,W)				0	1	0	1 0 0 0		1	1	1	0	0	rs	1	1	1	0	0	rd	d	d		
	MOV.L @(d:16,Rs,W),@(d:16,ERd,L)				0	1	0	1 0 0 0		1	1	1	0	0	rs	1	1	1	1	0	rd	d	d		
	MOV.L @(d:16,Rs,W),@(d:32,Rd,B)				0	1	0	1 0 0 0		1	1	1	0	0	rs	1	1	0	1	1	rd	d	d		
	MOV.L @(d:16,Rs,W),@(d:32,Rd,W)				0	1	0	1 0 0 0		1	1	1	0	0	rs	1	1	1	0	1	rd	d	d		
	MOV.L @(d:16,Rs,W),@(d:32,ERd,L)				0	1	0	1 0 0 0		1	1	1	0	0	rs	1	1	1	1	1	rd	d	d		
	MOV.L @(d:16,Rs,W),@aa:16				0	1	0	1 0 0 0		1	1	1	0	0	rs	0	1	0	0	0	0	0	0	d	a
	MOV.L @(d:16,Rs,W),@aa:32				0	1	0	1 0 0 0		1	1	1	0	0	rs	0	1	0	0	1	0	0	0	d	a
	MOV.L @(d:16,ERs,L),@ERd				0	1	0	1 0 0 0		1	1	1	1	0	rs	0	0	0	0	0	rd	d			
	MOV.L @(d:16,ERs,L),@ERd+				0	1	0	1 0 0 0		1	1	1	1	0	rs	1	0	0	0	0	rd	d			
	MOV.L @(d:16,ERs,L),@ERd-				0	1	0	1 0 0 0		1	1	1	1	0	rs	1	0	1	0	0	rd	d			
	MOV.L @(d:16,ERs,L),@+ERd				0	1	0	1 0 0 0		1	1	1	1	0	rs	1	0	0	1	0	rd	d			
	MOV.L @(d:16,ERs,L),@-ERd				0	1	0	1 0 0 0		1	1	1	1	0	rs	1	0	1	1	0	rd	d			
	MOV.L @(d:16,ERs,L),@(d:2,ERd)				0	1	0	1 0 0 0		1	1	1	1	0	rs	0	0	d	d	0	rd	d			
	MOV.L @(d:16,ERs,L),@(d:16,ERd)				0	1	0	1 0 0 0		1	1	1	1	0	rs	1	1	0	0	0	rd	d	d		
	MOV.L @(d:16,ERs,L),@(d:32,ERd)				0	1	0	1 0 0 0		1	1	1	1	0	rs	1	1	0	0	1	rd	d	d		
	MOV.L @(d:16,ERs,L),@(d:16,Rd,B)				0	1	0	1 0 0 0		1	1	1	1	0	rs	1	1	0	1	0	rd	d	d		
	MOV.L @(d:16,ERs,L),@(d:16,Rd,W)				0	1	0	1 0 0 0		1	1	1	1	0	rs	1	1	1	0	0	rd	d	d		
	MOV.L @(d:16,ERs,L),@(d:16,ERd,L)				0	1	0	1 0 0 0		1	1	1	1	0	rs	1	1	1	1	0	rd	d	d		
	MOV.L @(d:16,ERs,L),@(d:32,Rd,B)				0	1	0	1 0 0 0		1	1	1	1	0	rs	1	1	0	1	1	rd	d	d		
	MOV.L @(d:16,ERs,L),@(d:32,Rd,W)				0	1	0	1 0 0 0		1	1	1	1	0	rs	1	1	1	0	1	rd	d	d		
	MOV.L @(d:16,ERs,L),@(d:32,ERd,L)				0	1	0	1 0 0 0		1	1	1	1	0	rs	1	1	1	1	1	rd	d	d		
	MOV.L @(d:16,ERs,L),@aa:16				0	1	0	1 0 0 0		1	1	1	1	0	rs	0	1	0	0	0	0	0	0	d	a
	MOV.L @(d:16,ERs,L),@aa:32				0	1	0	1 0 0 0		1	1	1	1	0	rs	0	1	0	0	1	0	0	0	d	a
	MOV.L @(d:32,Rs,B),@ERd				0	1	0	1 0 0 0		1	1	0	1	1	rs	0	0	0	0	0	rd	d	d		
MOV.L @(d:32,Rs,B),@ERd+				0	1	0	1 0 0 0		1	1	0	1	1	rs	1	0	0	0	0	rd	d	d			
MOV.L @(d:32,Rs,B),@ERd-				0	1	0	1 0 0 0		1	1	0	1	1	rs	1	0	1	0	0	rd	d	d			

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension																			
		15	8	7	0		15	8	7	0																				
MOV	MOV.L @(d:32,Rs.B),@+ERd				0		1		0	1	0	0	0	1	1	0	1	rs	1	0	0	1	0	rd	d	d				
	MOV.L @(d:32,Rs.B),@-ERd				0		1		0	1	0	0	0	1	1	0	1	rs	1	0	1	1	0	rd	d	d				
	MOV.L @(d:32,Rs.B),@(d:2,ERd)				0		1		0	1	0	0	0	1	1	0	1	rs	0	0	d	d	0	rd	d	d				
	MOV.L @(d:32,Rs.B),@(d:16,ERd)				0		1		0	1	0	0	0	1	1	0	1	rs	1	1	0	0	0	rd	d	d	d			
	MOV.L @(d:32,Rs.B),@(d:32,ERd)				0		1		0	1	0	0	0	1	1	0	1	rs	1	1	0	0	1	rd	d	d	d	d		
	MOV.L @(d:32,Rs.B),@(d:16,Rd.B)				0		1		0	1	0	0	0	1	1	0	1	rs	1	1	0	1	0	rd	d	d	d			
	MOV.L @(d:32,Rs.B),@(d:16,Rd.W)				0		1		0	1	0	0	0	1	1	0	1	rs	1	1	1	0	0	rd	d	d	d			
	MOV.L @(d:32,Rs.B),@(d:16,ERd.L)				0		1		0	1	0	0	0	1	1	0	1	rs	1	1	1	1	0	rd	d	d	d			
	MOV.L @(d:32,Rs.B),@(d:32,Rd.B)				0		1		0	1	0	0	0	1	1	0	1	rs	1	1	0	1	1	rd	d	d	d	d		
	MOV.L @(d:32,Rs.B),@(d:32,Rd.W)				0		1		0	1	0	0	0	1	1	0	1	rs	1	1	1	0	1	rd	d	d	d	d		
	MOV.L @(d:32,Rs.B),@(d:32,ERd.L)				0		1		0	1	0	0	0	1	1	0	1	rs	1	1	1	1	1	rd	d	d	d	d		
	MOV.L @(d:32,Rs.B),@aa:16				0		1		0	1	0	0	0	1	1	0	1	rs	0	1	0	0	0	0	0	0	0	d	d	a
	MOV.L @(d:32,Rs.B),@aa:32				0		1		0	1	0	0	0	1	1	0	1	rs	0	1	0	0	1	0	0	0	d	d	a	a
	MOV.L @(d:32,Rs.W),@ERd				0		1		0	1	0	0	0	1	1	1	0	rs	0	0	0	0	0	rd	d	d				
	MOV.L @(d:32,Rs.W),@ERd+				0		1		0	1	0	0	0	1	1	1	0	rs	1	0	0	0	0	rd	d	d				
	MOV.L @(d:32,Rs.W),@ERd-				0		1		0	1	0	0	0	1	1	1	0	rs	1	0	1	0	0	rd	d	d				
	MOV.L @(d:32,Rs.W),@+ERd				0		1		0	1	0	0	0	1	1	1	0	rs	1	0	0	1	0	rd	d	d				
	MOV.L @(d:32,Rs.W),@-ERd				0		1		0	1	0	0	0	1	1	1	0	rs	1	0	1	1	0	rd	d	d				
	MOV.L @(d:32,Rs.W),@(d:2,ERd)				0		1		0	1	0	0	0	1	1	1	0	rs	0	0	d	d	0	rd	d	d				
	MOV.L @(d:32,Rs.W),@(d:16,ERd)				0		1		0	1	0	0	0	1	1	1	0	rs	1	1	0	0	0	rd	d	d	d			
	MOV.L @(d:32,Rs.W),@(d:32,ERd)				0		1		0	1	0	0	0	1	1	1	0	rs	1	1	0	0	1	rd	d	d	d	d		
	MOV.L @(d:32,Rs.W),@(d:16,Rd.B)				0		1		0	1	0	0	0	1	1	1	0	rs	1	1	0	1	0	rd	d	d	d			
	MOV.L @(d:32,Rs.W),@(d:16,Rd.W)				0		1		0	1	0	0	0	1	1	1	0	rs	1	1	1	0	0	rd	d	d	d			
	MOV.L @(d:32,Rs.W),@(d:16,ERd.L)				0		1		0	1	0	0	0	1	1	1	0	rs	1	1	1	1	0	rd	d	d	d			
	MOV.L @(d:32,Rs.W),@(d:32,Rd.B)				0		1		0	1	0	0	0	1	1	1	0	rs	1	1	0	1	1	rd	d	d	d	d		
	MOV.L @(d:32,Rs.W),@(d:32,Rd.W)				0		1		0	1	0	0	0	1	1	1	0	rs	1	1	1	0	1	rd	d	d	d	d		
	MOV.L @(d:32,Rs.W),@(d:32,ERd.L)				0		1		0	1	0	0	0	1	1	1	0	rs	1	1	1	1	1	rd	d	d	d	d		
	MOV.L @(d:32,Rs.W),@aa:16				0		1		0	1	0	0	0	1	1	1	0	rs	0	1	0	0	0	0	0	0	d	d	a	
	MOV.L @(d:32,Rs.W),@aa:32				0		1		0	1	0	0	0	1	1	1	0	rs	0	1	0	0	1	0	0	0	d	d	a	a
	MOV.L @(d:32,ERs.L),@ERd				0		1		0	1	0	0	0	1	1	1	1	rs	0	0	0	0	0	rd	d	d				
	MOV.L @(d:32,ERs.L),@ERd+				0		1		0	1	0	0	0	1	1	1	1	rs	1	0	0	0	0	rd	d	d				
	MOV.L @(d:32,ERs.L),@ERd-				0		1		0	1	0	0	0	1	1	1	1	rs	1	0	1	0	0	rd	d	d				
MOV.L @(d:32,ERs.L),@+ERd				0		1		0	1	0	0	0	1	1	1	1	rs	1	0	0	1	0	rd	d	d					
MOV.L @(d:32,ERs.L),@-ERd				0		1		0	1	0	0	0	1	1	1	1	rs	1	0	1	1	0	rd	d	d					



Instruction	Mnemonic	Opcode			EA Ex- tension	Opcode			EA Extension				
		15	8	7		0	15	8		7	0		
MOV	MOV.L @(d:32,ERs.L),@(d:2,ERd)	0	1	0	1 0 0 0	1 1 1 1	1	rs	0 0 d d	0	rd	d d	
	MOV.L @(d:32,ERs.L),@(d:16,ERd)	0	1	0	1 0 0 0	1 1 1 1	1	rs	1 1 0 0	0	rd	d d d	
	MOV.L @(d:32,ERs.L),@(d:32,ERd)	0	1	0	1 0 0 0	1 1 1 1	1	rs	1 1 0 0	1	rd	d d d d	
	MOV.L @(d:32,ERs.L),@(d:16,Rd.B)	0	1	0	1 0 0 0	1 1 1 1	1	rs	1 1 0 1	0	rd	d d d	
	MOV.L @(d:32,ERs.L),@(d:16,Rd.W)	0	1	0	1 0 0 0	1 1 1 1	1	rs	1 1 1 0	0	rd	d d d	
	MOV.L @(d:32,ERs.L),@(d:16,ERd.L)	0	1	0	1 0 0 0	1 1 1 1	1	rs	1 1 1 1	0	rd	d d d	
	MOV.L @(d:32,ERs.L),@(d:32,Rd.B)	0	1	0	1 0 0 0	1 1 1 1	1	rs	1 1 0 1	1	rd	d d d d	
	MOV.L @(d:32,ERs.L),@(d:32,Rd.W)	0	1	0	1 0 0 0	1 1 1 1	1	rs	1 1 1 0	1	rd	d d d d	
	MOV.L @(d:32,ERs.L),@(d:32,ERd.L)	0	1	0	1 0 0 0	1 1 1 1	1	rs	1 1 1 1	1	rd	d d d d	
	MOV.L @(d:32,ERs.L),@aa:16	0	1	0	1 0 0 0	1 1 1 1	1	rs	0 1 0 0	0 0 0 0		d d a	
	MOV.L @(d:32,ERs.L),@aa:32	0	1	0	1 0 0 0	1 1 1 1	1	rs	0 1 0 0	1 0 0 0		d d a a	
	MOV.L @aa:16,@ERd	0	1	0	1 0 0 0	0 1 0 0	0	0 0 0 0	0 0 0 0	0 0	rd	a	
	MOV.L @aa:16,@ERd+	0	1	0	1 0 0 0	0 1 0 0	0	0 0 0 0	1 0 0 0	0	rd	a	
	MOV.L @aa:16,@ERd-	0	1	0	1 0 0 0	0 1 0 0	0	0 0 0 0	1 0 1 0	0	rd	a	
	MOV.L @aa:16,@+ERd	0	1	0	1 0 0 0	0 1 0 0	0	0 0 0 0	1 0 0 1	0	rd	a	
	MOV.L @aa:16,@-ERd	0	1	0	1 0 0 0	0 1 0 0	0	0 0 0 0	1 0 1 1	0	rd	a	
	MOV.L @aa:16,@(d:2,ERd)	0	1	0	1 0 0 0	0 1 0 0	0	0 0 0 0	0 0 d d	0	rd	a	
	MOV.L @aa:16,@(d:16,ERd)	0	1	0	1 0 0 0	0 1 0 0	0	0 0 0 0	1 1 0 0	0	rd	a	d
	MOV.L @aa:16,@(d:32,ERd)	0	1	0	1 0 0 0	0 1 0 0	0	0 0 0 0	1 1 0 0	1	rd	a	d d
	MOV.L @aa:16,@(d:16,Rd.B)	0	1	0	1 0 0 0	0 1 0 0	0	0 0 0 0	1 1 0 1	0	rd	a	d
	MOV.L @aa:16,@(d:16,Rd.W)	0	1	0	1 0 0 0	0 1 0 0	0	0 0 0 0	1 1 1 0	0	rd	a	d
	MOV.L @aa:16,@(d:16,ERd.L)	0	1	0	1 0 0 0	0 1 0 0	0	0 0 0 0	1 1 1 1	0	rd	a	d
	MOV.L @aa:16,@(d:32,Rd.B)	0	1	0	1 0 0 0	0 1 0 0	0	0 0 0 0	1 1 0 1	1	rd	a	d d
	MOV.L @aa:16,@(d:32,Rd.W)	0	1	0	1 0 0 0	0 1 0 0	0	0 0 0 0	1 1 1 0	1	rd	a	d d
	MOV.L @aa:16,@(d:32,ERd.L)	0	1	0	1 0 0 0	0 1 0 0	0	0 0 0 0	1 1 1 1	1	rd	a	d d
	MOV.L @aa:16,@aa:16	0	1	0	1 0 0 0	0 1 0 0	0	0 0 0 0	0 1 0 0	0 0 0 0		a	a
	MOV.L @aa:16,@aa:32	0	1	0	1 0 0 0	0 1 0 0	0	0 0 0 0	0 1 0 0	1 0 0 0		a	a a
	MOV.L @aa:32,@ERd	0	1	0	1 0 0 0	0 1 0 0	1	0 0 0 0	0 0 0 0	0	rd	a a	
	MOV.L @aa:32,@ERd+	0	1	0	1 0 0 0	0 1 0 0	1	0 0 0 0	1 0 0 0	0	rd	a a	
	MOV.L @aa:32,@ERd-	0	1	0	1 0 0 0	0 1 0 0	1	0 0 0 0	1 0 1 0	0	rd	a a	
	MOV.L @aa:32,@+ERd	0	1	0	1 0 0 0	0 1 0 0	1	0 0 0 0	1 0 0 1	0	rd	a a	
	MOV.L @aa:32,@-ERd	0	1	0	1 0 0 0	0 1 0 0	1	0 0 0 0	1 0 1 1	0	rd	a a	
MOV.L @aa:32,@(d:2,ERd)	0	1	0	1 0 0 0	0 1 0 0	1	0 0 0 0	0 0 d d	0	rd	a a		
MOV.L @aa:32,@(d:16,ERd)	0	1	0	1 0 0 0	0 1 0 0	1	0 0 0 0	1 1 0 0	0	rd	a a d		

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension										
		15	8	7	0		15	8	7	0											
MOV	MOV.L @aa:32,@(d:32,ERd)			0	1	0	1	0	0	0	0	0	1	rd	a	a	d	d			
	MOV.L @aa:32,@(d:16,Rd,B)			0	1	0	1	0	0	0	0	0	1	rd	a	a	d				
	MOV.L @aa:32,@(d:16,Rd,W)			0	1	0	1	0	0	0	0	1	rd	a	a	d					
	MOV.L @aa:32,@(d:16,ERd,L)			0	1	0	1	0	0	0	0	1	rd	a	a	d					
	MOV.L @aa:32,@(d:32,Rd,B)			0	1	0	1	0	0	0	0	1	rd	a	a	d	d				
	MOV.L @aa:32,@(d:32,Rd,W)			0	1	0	1	0	0	0	0	1	rd	a	a	d	d				
	MOV.L @aa:32,@(d:32,ERd,L)			0	1	0	1	0	0	0	0	1	rd	a	a	d	d				
	MOV.L @aa:32,@aa:16			0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0		
MOV.L @aa:32,@aa:32			0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0			
MOVA	MOVA.B.L @(d:16,Rn,B),ERn										7	A	1	0	0	0	1	m	d		
	MOVA.B.L @(d:16,Rn,W),ERn										7	A	1	0	0	1	1	m	d		
	MOVA.W.L @(d:16,Rn,B),ERn										7	A	1	0	1	0	1	m	d		
	MOVA.W.L @(d:16,Rn,W),ERn										7	A	1	0	1	1	1	m	d		
	MOVA.L.L @(d:16,Rn,B),ERn										7	A	1	1	0	0	1	m	d		
	MOVA.L.L @(d:16,Rn,W),ERn										7	A	1	1	0	1	1	m	d		
	MOVA.B.L @(d:32,Rn,B),ERn											7	A	1	0	0	0	0	m	d	d
	MOVA.B.L @(d:32,Rn,W),ERn											7	A	1	0	0	1	0	m	d	d
	MOVA.W.L @(d:32,Rn,B),ERn											7	A	1	0	1	0	0	m	d	d
	MOVA.W.L @(d:32,Rn,W),ERn											7	A	1	0	1	1	0	m	d	d
	MOVA.L.L @(d:32,Rn,B),ERn											7	A	1	1	0	0	0	m	d	d
	MOVA.L.L @(d:32,Rn,W),ERn											7	A	1	1	0	1	0	m	d	d
	MOVA.B.L @(d:16,Rs,B),ERd			7	8		rs	1	0	0	0	7	A	1	0	0	0	1	rd	d	
	MOVA.B.L @(d:16,Rs,W),ERd			7	8		rs	1	0	0	1	7	A	1	0	0	1	1	rd	d	
	MOVA.W.L @(d:16,Rs,B),ERd			7	8		rs	1	0	0	0	7	A	1	0	1	0	1	rd	d	
	MOVA.W.L @(d:16,Rs,W),ERd			7	8		rs	1	0	0	1	7	A	1	0	1	1	1	rd	d	
	MOVA.L.L @(d:16,Rs,B),ERd			7	8		rs	1	0	0	0	7	A	1	1	0	0	1	rd	d	
	MOVA.L.L @(d:16,Rs,W),ERd			7	8		rs	1	0	0	1	7	A	1	1	0	1	1	rd	d	
	MOVA.B.L @(d:32,Rs,B),ERd			7	8		rs	1	0	0	0	7	A	1	0	0	0	0	rd	d	d
	MOVA.B.L @(d:32,Rs,W),ERd			7	8		rs	1	0	0	1	7	A	1	0	0	1	0	rd	d	d
	MOVA.W.L @(d:32,Rs,B),ERd			7	8		rs	1	0	0	0	7	A	1	0	1	0	0	rd	d	d
	MOVA.W.L @(d:32,Rs,W),ERd			7	8		rs	1	0	0	1	7	A	1	0	1	1	0	rd	d	d
	MOVA.L.L @(d:32,Rs,B),ERd			7	8		rs	1	0	0	0	7	A	1	1	0	0	0	rd	d	d
	MOVA.L.L @(d:32,Rs,W),ERd			7	8		rs	1	0	0	1	7	A	1	1	0	1	0	rd	d	d
MOVA.B.L @(d:16,@ERs,B),ERd			0	1		7	1	1	1	1	0	0	0	0	0	0	0	rd		d	



Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0		15	8	7	0		
MOVA	MOVA.B.L @(d:16,@ERs+.B),ERd	0	1	7	1 1 1 1	1 0 0 0	0	rs	1 0 0 0	0	rd	d
	MOVA.B.L @(d:16,@ERs.B),ERd	0	1	7	1 1 1 1	1 0 1 0	0	rs	1 0 0 0	0	rd	d
	MOVA.B.L @(d:16,@+ERs.B),ERd	0	1	7	1 1 1 1	1 0 0 1	0	rs	1 0 0 0	0	rd	d
	MOVA.B.L @(d:16,@-ERs.B),ERd	0	1	7	1 1 1 1	1 0 1 1	0	rs	1 0 0 0	0	rd	d
	MOVA.B.L @(d:16,@(d2,ERs).B),ERd	0	1	7	1 1 1 1	0 0 d d	0	rs	1 0 0 0	0	rd	d
	MOVA.B.L @(d:16,@(d:16,ERs).B),ERd	0	1	7	1 1 1 1	1 1 0 0	0	rs	1 0 0 0	0	rd	d d
	MOVA.B.L @(d:16,@(d:32,ERs).B),ERd	0	1	7	1 1 1 1	1 1 0 0	1	rs	1 0 0 0	0	rd	d d d
	MOVA.B.L @(d:16,@(d:16,Rs.B),B),ERd	0	1	7	1 1 1 1	1 1 0 1	0	rs	1 0 0 0	0	rd	d d
	MOVA.B.L @(d:16,@(d:16,Rs.W).B),ERd	0	1	7	1 1 1 1	1 1 1 0	0	rs	1 0 0 0	0	rd	d d
	MOVA.B.L @(d:16,@(d:16,ERs.L).B),ERd	0	1	7	1 1 1 1	1 1 1 1	0	rs	1 0 0 0	0	rd	d d
	MOVA.B.L @(d:16,@(d:32,Rs.B).B),ERd	0	1	7	1 1 1 1	1 1 0 1	1	rs	1 0 0 0	0	rd	d d d
	MOVA.B.L @(d:16,@(d:32,Rs.W).B),ERd	0	1	7	1 1 1 1	1 1 1 0	1	rs	1 0 0 0	0	rd	d d d
	MOVA.B.L @(d:16,@(d:32,ERs.L).B),ERd	0	1	7	1 1 1 1	1 1 1 1	1	rs	1 0 0 0	0	rd	d d d
	MOVA.B.L @(d:16,@aa:16.B),ERd	0	1	7	1 1 1 1	0 1 0 0	0 0 0 0	1 0 0 0	0 0	rd	a d	
	MOVA.B.L @(d:16,@aa:32.B),ERd	0	1	7	1 1 1 1	0 1 0 0	1 0 0 0	1 0 0 0	0 0	rd	a a d	
	MOVA.B.L @(d:16,@ERs.W),ERd	0	1	5	1 1 1 1	0 0 0 0	0	rs	1 0 0 1	0	rd	d
	MOVA.B.L @(d:16,@ERs+.W),ERd	0	1	5	1 1 1 1	1 0 0 0	0	rs	1 0 0 1	0	rd	d
	MOVA.B.L @(d:16,@ERs-.W),ERd	0	1	5	1 1 1 1	1 0 1 0	0	rs	1 0 0 1	0	rd	d
	MOVA.B.L @(d:16,@+ERs.W),ERd	0	1	5	1 1 1 1	1 0 0 1	0	rs	1 0 0 1	0	rd	d
	MOVA.B.L @(d:16,@-ERs.W),ERd	0	1	5	1 1 1 1	1 0 1 1	0	rs	1 0 0 1	0	rd	d
MOVA.B.L @(d:16,@(d2,ERs).W),ERd	0	1	5	1 1 1 1	0 0 d d	0	rs	1 0 0 1	0	rd	d	
MOVA.B.L @(d:16,@(d:16,ERs).W),ERd	0	1	5	1 1 1 1	1 1 0 0	0	rs	1 0 0 1	0	rd	d d	
MOVA.B.L @(d:16,@(d:32,ERs).W),ERd	0	1	5	1 1 1 1	1 1 0 0	1	rs	1 0 0 1	0	rd	d d d	

Instruction	Mnemonic	Opcode			EA Extension	Opcode			EA Extension							
		15	8	7		0	15	8		7	0					
MOVA	MOVAB.L				0	1	5	1 1 1 1	1 1 0 1	0	rs	1 0 0 1	0	rd	d	d
	@(d:16,@(d:16,Rs.B),W),ERd				0	1	5	1 1 1 1	1 1 1 0	0	rs	1 0 0 1	0	rd	d	d
	MOVAB.L				0	1	5	1 1 1 1	1 1 1 1	0	rs	1 0 0 1	0	rd	d	d
	@(d:16,@(d:16,ERs.L),W),ERd				0	1	5	1 1 1 1	1 1 0 1	1	rs	1 0 0 1	0	rd	d	d
	MOVAB.L				0	1	5	1 1 1 1	1 1 1 0	1	rs	1 0 0 1	0	rd	d	d
	@(d:16,@(d:32,Rs.B),W),ERd				0	1	5	1 1 1 1	1 1 1 1	1	rs	1 0 0 1	0	rd	d	d
	MOVAB.L				0	1	5	1 1 1 1	0 1 0 0	0 0 0 0	1 0 0 1	0	rd	a	d	
	@(d:16,@aa:16,W),ERd				0	1	5	1 1 1 1	0 1 0 0	1 0 0 0	1 0 0 1	0	rd	a	a	d
	MOVAB.L				0	1	7	1 1 1 1	0 0 0 0	0	rs	1 0 1 0	0	rd		d
	@(d:16,@ERs.B),ERd				0	1	7	1 1 1 1	1 0 0 0	0	rs	1 0 1 0	0	rd		d
	MOVAW.L				0	1	7	1 1 1 1	1 0 1 0	0	rs	1 0 1 0	0	rd		d
	@(d:16,@ERs+.B),ERd				0	1	7	1 1 1 1	1 0 0 1	0	rs	1 0 1 0	0	rd		d
	MOVAW.L				0	1	7	1 1 1 1	1 0 1 1	0	rs	1 0 1 0	0	rd		d
	@(d:16,@ERs-.B),ERd				0	1	7	1 1 1 1	0 0 d d	0	rs	1 0 1 0	0	rd		d
	MOVAW.L				0	1	7	1 1 1 1	1 1 0 0	0	rs	1 0 1 0	0	rd	d	d
	@(d:16,@(d:16,ERs),B),ERd				0	1	7	1 1 1 1	1 1 0 0	1	rs	1 0 1 0	0	rd	d	d
	MOVAW.L				0	1	7	1 1 1 1	1 1 0 1	0	rs	1 0 1 0	0	rd	d	d
	@(d:16,@(d:16,Rs.B),B),ERd				0	1	7	1 1 1 1	1 1 1 0	0	rs	1 0 1 0	0	rd	d	d
	MOVAW.L				0	1	7	1 1 1 1	1 1 1 1	0	rs	1 0 1 0	0	rd	d	d
	@(d:16,@(d:16,Rs.W),B),ERd				0	1	7	1 1 1 1	1 1 1 1	0	rs	1 0 1 0	0	rd	d	d
	MOVAW.L				0	1	7	1 1 1 1	1 1 0 1	1	rs	1 0 1 0	0	rd	d	d
	@(d:16,@(d:32,ERs.L),B),ERd				0	1	7	1 1 1 1	1 1 1 0	1	rs	1 0 1 0	0	rd	d	d
	MOVAW.L				0	1	7	1 1 1 1	1 1 1 1	1	rs	1 0 1 0	0	rd	d	d
	@(d:16,@(d:32,Rs.B),B),ERd				0	1	7	1 1 1 1	1 1 1 1	1	rs	1 0 1 0	0	rd	d	d
MOVAW.L				0	1	7	1 1 1 1	1 1 1 1	1	rs	1 0 1 0	0	rd	d	d	
@(d:16,@(d:32,Rs.W),B),ERd				0	1	7	1 1 1 1	1 1 1 1	1	rs	1 0 1 0	0	rd	d	d	
MOVAW.L				0	1	7	1 1 1 1	1 1 1 1	1	rs	1 0 1 0	0	rd	d	d	
@(d:16,@(d:32,ERs.L),B),ERd				0	1	7	1 1 1 1	1 1 1 1	1	rs	1 0 1 0	0	rd	d	d	



Instruction	Mnemonic	Opcode			Opcode			EA Extension	Opcode			EA Extension																
		15	8	7	0	15	8		7	0	15		8	7	0													
MOVA	MOVA.W.L @(d.16,@aa:16.B),ERd				0	1	7	1	1	1	1	1	0	1	0	0	0	0	0	1	0	1	0	0	rd	a	d	
	MOVA.W.L @(d.16,@aa:32.B),ERd				0	1	7	1	1	1	1	1	0	0	1	0	0	0	1	0	1	0	0	0	rd	a	a	d
	MOVA.W.L @(d.16,@ERs.W),ERd				0	1	5	1	1	1	1	1	0	0	0	0	0	rs	1	0	1	1	0	0	rd		d	
	MOVA.W.L @(d.16,@ERs+.W),ERd				0	1	5	1	1	1	1	1	0	0	0	0	rs	1	0	1	1	0	0	rd		d		
	MOVA.W.L @(d.16,@ERs-.W),ERd				0	1	5	1	1	1	1	1	0	1	0	0	rs	1	0	1	1	0	0	rd		d		
	MOVA.W.L @(d.16,@+ERs.W),ERd				0	1	5	1	1	1	1	1	0	1	0	0	rs	1	0	1	1	0	0	rd		d		
	MOVA.W.L @(d.16,@-ERs.W),ERd				0	1	5	1	1	1	1	1	0	1	0	0	rs	1	0	1	1	0	0	rd		d		
	MOVA.W.L @ (d.16,@(d2,ERs).W),ERd				0	1	5	1	1	1	1	1	0	0	d	d	0	rs	1	0	1	1	0	0	rd		d	
	MOVA.W.L @ (d.16,@(d:16,ERs).W),ERd				0	1	5	1	1	1	1	1	0	0	0	0	rs	1	0	1	1	0	0	rd	d		d	
	MOVA.W.L @ (d.16,@(d:32,ERs).W),ERd				0	1	5	1	1	1	1	1	0	0	1	rs	1	0	1	1	0	0	0	rd	d	d	d	
	MOVA.W.L @ (d.16,@(d:16,Rs.B).W),ERd				0	1	5	1	1	1	1	1	0	1	0	rs	1	0	1	1	0	0	0	rd	d		d	
	MOVA.W.L @ (d.16,@(d:16,Rs.W).W),ERd				0	1	5	1	1	1	1	1	0	0	0	rs	1	0	1	1	0	0	0	rd	d		d	
	MOVA.W.L @ (d.16,@(d:16,ERs.L).W),ERd				0	1	5	1	1	1	1	1	0	0	0	rs	1	0	1	1	0	0	0	rd	d		d	
	MOVA.W.L @ (d.16,@(d:32,Rs.B).W),ERd				0	1	5	1	1	1	1	1	0	1	1	rs	1	0	1	1	0	0	0	rd	d	d	d	
	MOVA.W.L @ (d.16,@(d:32,Rs.W).W),ERd				0	1	5	1	1	1	1	1	0	1	1	rs	1	0	1	1	0	0	0	rd	d	d	d	
	MOVA.W.L @ (d.16,@(d:32,ERs.L).W),ERd				0	1	5	1	1	1	1	1	0	1	1	rs	1	0	1	1	0	0	0	rd	d	d	d	
	MOVA.W.L @(d.16,@aa:16.W),ERd				0	1	5	1	1	1	1	1	0	0	1	0	0	0	1	0	1	1	0	0	rd	a	d	
	MOVA.W.L @(d.16,@aa:32.W),ERd				0	1	5	1	1	1	1	1	0	0	1	0	0	0	1	0	1	1	0	0	rd	a	a	d
	MOVA.L.L @(d.16,@ERs.B),ERd				0	1	7	1	1	1	1	1	0	0	0	0	rs	1	1	0	0	0	0	0	rd		d	
	MOVA.L.L @(d.16,@ERs+.B),ERd				0	1	7	1	1	1	1	1	0	0	0	0	rs	1	1	0	0	0	0	0	rd		d	
	MOVA.L.L @(d.16,@ERs-.B),ERd				0	1	7	1	1	1	1	1	0	1	0	0	rs	1	1	0	0	0	0	0	rd		d	
	MOVA.L.L @(d.16,@+ERs.B),ERd				0	1	7	1	1	1	1	1	0	0	1	0	rs	1	1	0	0	0	0	0	rd		d	
	MOVA.L.L @(d.16,@-ERs.B),ERd				0	1	7	1	1	1	1	1	0	1	0	0	rs	1	1	0	0	0	0	0	rd		d	
	MOVA.L.L @ (d.16,@(d2,ERs).B),ERd				0	1	7	1	1	1	1	1	0	d	d	0	rs	1	1	0	0	0	0	0	rd		d	
MOVA.L.L @ (d.16,@(d:16,ERs).B),ERd				0	1	7	1	1	1	1	1	0	0	0	0	rs	1	1	0	0	0	0	0	rd	d		d	

Instruction	Mnemonic	Opcode			EA Ex- tension	Opcode			EA Extension										
		15	8	7		0	15	8		7	0								
MOVA	MOVALL					0	1	7	1 1 1 1		1 1 0 0	1	rs	1 1 0 0	0 0	rd	d	d	d
	@(d:16,@(d:32,ERs),B),ERd																		
	MOVALL					0	1	7	1 1 1 1		1 1 0 1	0	rs	1 1 0 0	0 0	rd	d		d
	@(d:16,@(d:16,Rs,B),B),ERd																		
	MOVALL					0	1	7	1 1 1 1		1 1 1 0	0	rs	1 1 0 0	0 0	rd	d		d
	@(d:16,@(d:16,Rs,W),B),ERd																		
	MOVALL					0	1	7	1 1 1 1		1 1 1 1	0	rs	1 1 0 0	0 0	rd	d		d
	@(d:16,@(d:16,ERs,L),B),ERd																		
	MOVALL					0	1	7	1 1 1 1		1 1 0 1	1	rs	1 1 0 0	0 0	rd	d	d	d
	@(d:16,@(d:32,Rs,B),B),ERd																		
	MOVALL					0	1	7	1 1 1 1		1 1 1 0		rs	1 1 0 0	0 0	rd	d	d	d
	@(d:16,@(d:32,Rs,W),B),ERd																		
	MOVALL					0	1	7	1 1 1 1		1 1 1 1	1	rs	1 1 0 0	0 0	rd	d	d	d
	@(d:16,@(d:32,ERs,L),B),ERd																		
	MOVALL @ (d:16,@aa:16B),ERd					0	1	7	1 1 1 1		0 1 0 0	0 0 0 0	1 1 0 0	0 0	rd	a		d	
	MOVALL @ (d:16,@aa:32B),ERd					0	1	7	1 1 1 1		0 1 0 0	1 0 0 0	1 1 0 0	0 0	rd	a	a	d	
	MOVALL @(d:16,@ERs,W),ERd					0	1	5	1 1 1 1		0 0 0 0	0	rs	1 1 0 1	1 0	rd			d
	MOVALL @(d:16,@ERs+,W),ERd					0	1	5	1 1 1 1		1 0 0 0	0	rs	1 1 0 1	1 0	rd			d
	MOVALL @(d:16,@ERs-,W),ERd					0	1	5	1 1 1 1		1 0 1 0	0	rs	1 1 0 1	1 0	rd			d
	MOVALL @(d:16,@+ERs,W),ERd					0	1	5	1 1 1 1		1 0 0 1	0	rs	1 1 0 1	1 0	rd			d
	MOVALL @(d:16,@-ERs,W),ERd					0	1	5	1 1 1 1		1 0 1 1	0	rs	1 1 0 1	1 0	rd			d
	MOVALL @(d:16,@(d:2,ERs),W),ERd					0	1	5	1 1 1 1		0 0 d d	0	rs	1 1 0 1	1 0	rd			d
	MOVALL @(d:16,@(d:16,ERs),W),ERd					0	1	5	1 1 1 1		1 1 0 0	0 0	rs	1 1 0 1	1 0	rd	d		d
	MOVALL @(d:16,@(d:32,ERs),W),ERd					0	1	5	1 1 1 1		1 1 0 0	1	rs	1 1 0 1	1 0	rd	d	d	d
	MOVALL @(d:16,@(d:16,Rs,B),W),ERd					0	1	5	1 1 1 1		1 1 0 1	0	rs	1 1 0 1	1 0	rd	d		d
	MOVALL @(d:16,@(d:16,Rs,W),W),ERd					0	1	5	1 1 1 1		1 1 1 0	0 0	rs	1 1 0 1	1 0	rd	d		d
	MOVALL @(d:16,@(d:16,ERs,L),W),ERd					0	1	5	1 1 1 1		1 1 1 1	0	rs	1 1 0 1	1 0	rd	d		d
MOVALL @(d:16,@(d:32,Rs,B),W),ERd					0	1	5	1 1 1 1		1 1 0 1	1	rs	1 1 0 1	1 0	rd	d	d	d	
MOVALL @(d:16,@(d:32,Rs,W),W),ERd					0	1	5	1 1 1 1		1 1 1 0	1	rs	1 1 0 1	1 0	rd	d	d	d	



Instruction	Mnemonic	Opcode			EA Extension	Opcode			EA Extension										
		15	8	7		0	15	8		7	0	EA Extension							
MOVA	MOVAL.L				0	1	5	1 1 1 1		1 1 1 1	1	rs	1 1 0 1	0	rd	d	d	d	
	@(d:16,@(d:32,ERs.L),W),ERd																		
	MOVAL.L				0	1	5	1 1 1 1		0 1 0 0	0	0 0 0 0	1 1 0 1	0	rd	a		d	
	@(d:16,@aa:16,W),ERd																		
	MOVAL.L				0	1	5	1 1 1 1		0 1 0 0	1	0 0 0 0	1 1 0 1	0	rd	a	a	d	
	@(d:16,@aa:32,W),ERd																		
	MOVA.B.L				0	1	7	1 1 1 1		0 0 0 0	0	rs	1 0 0 0	1	rd			d	d
	@(d:32,@ERs.B),ERd																		
	MOVA.B.L				0	1	7	1 1 1 1		1 0 0 0	0	rs	1 0 0 0	1	rd			d	d
	@(d:32,@ERs+.B),ERd																		
	MOVA.B.L				0	1	7	1 1 1 1		1 0 1 0	0	rs	1 0 0 0	1	rd			d	d
	@(d:32,@ERs-.B),ERd																		
	MOVA.B.L				0	1	7	1 1 1 1		1 0 0 1	0	rs	1 0 0 0	1	rd			d	d
	@(d:32,@+ERs.B),ERd																		
	MOVA.B.L				0	1	7	1 1 1 1		1 0 1 1	0	rs	1 0 0 0	1	rd			d	d
	@(d:32,@-ERs.B),ERd																		
	MOVA.B.L				0	1	7	1 1 1 1		0 0 d d	0	rs	1 0 0 0	1	rd			d	d
	@(d:32,@(d:2,ERs.B),ERd																		
	MOVA.B.L				0	1	7	1 1 1 1		1 1 0 0	0	rs	1 0 0 0	1	rd	d		d	d
	@(d:32,@(d:16,ERs),B),ERd																		
	MOVA.B.L				0	1	7	1 1 1 1		1 1 0 0	1	rs	1 0 0 0	1	rd	d	d	d	d
	@(d:32,@(d:32,ERs),B),ERd																		
	MOVA.B.L				0	1	7	1 1 1 1		1 1 0 1	0	rs	1 0 0 0	1	rd	d		d	d
@(d:32,@(d:16,Rs.B),ERd																			
MOVA.B.L				0	1	7	1 1 1 1		1 1 1 0	0	rs	1 0 0 0	1	rd	d		d	d	
@(d:32,@(d:16,Rs.W),B),ERd																			
MOVA.B.L				0	1	7	1 1 1 1		1 1 1 1	0	rs	1 0 0 0	1	rd	d		d	d	
@(d:32,@(d:16,ERs.L),B),ERd																			
MOVA.B.L				0	1	7	1 1 1 1		1 1 0 1	1	rs	1 0 0 0	1	rd	d	d	d	d	
@(d:32,@(d:32,Rs.B),B),ERd																			
MOVA.B.L				0	1	7	1 1 1 1		1 1 1 0	1	rs	1 0 0 0	1	rd	d	d	d	d	
@(d:32,@(d:32,Rs.W),B),ERd																			
MOVA.B.L				0	1	7	1 1 1 1		1 1 1 1	1	rs	1 0 0 0	1	rd	d	d	d	d	
@(d:32,@(d:32,ERs.L),B),ERd																			
MOVA.B.L				0	1	7	1 1 1 1		0 1 0 0	0	0 0 0 0	1 0 0 0	0	1	rd	a		d	
@(d:32,@aa:16,B),ERd																			
MOVA.B.L				0	1	7	1 1 1 1		0 1 0 0	1	0 0 0 0	1 0 0 0	0	1	rd	a	a	d	
@(d:32,@aa:32,B),ERd																			
MOVA.B.L				0	1	5	1 1 1 1		0 0 0 0	0	rs	1 0 0 1	1	rd			d	d	
@(d:32,@ERs.W),ERd																			
MOVA.B.L				0	1	5	1 1 1 1		1 0 0 0	0	rs	1 0 0 1	1	rd			d	d	
@(d:32,@ERs+.W),ERd																			
MOVA.B.L				0	1	5	1 1 1 1		1 0 1 0	0	rs	1 0 0 1	1	rd			d	d	
@(d:32,@ERs-.W),ERd																			
MOVA.B.L				0	1	5	1 1 1 1		1 0 0 1	0	rs	1 0 0 1	1	rd			d	d	
@(d:32,@+ERs.W),ERd																			
MOVA.B.L				0	1	5	1 1 1 1		1 0 1 1	0	rs	1 0 0 1	1	rd			d	d	
@(d:32,@-ERs.W),ERd																			
MOVA.B.L				0	1	5	1 1 1 1		0 0 d d	0	rs	1 0 0 1	1	rd			d	d	
@(d:32,@(d:2,ERs),W),ERd																			

Instruction	Mnemonic	Opcode			EA Extension	Opcode			EA Extension																					
		15	8	7		0	15	8		7	0																			
MOVA	MOVA.B.L				0		1		5	1	1	1	1	1	1	1	0	0	rs	1	0	0	1	1	1	rd	d	d	d	
	@(d:32,@(d:16,ERs),W),ERd				0		1		5	1	1	1	1	1	1	1	0	0	rs	1	0	0	1	1	1	1	rd	d	d	d
	MOVA.B.L				0		1		5	1	1	1	1	1	1	1	0	0	rs	1	0	0	1	1	1	1	rd	d	d	d
	@(d:32,@(d:16,Rs.B),W),ERd				0		1		5	1	1	1	1	1	1	0	0	rs	1	0	0	1	1	1	1	rd	d	d	d	
	MOVA.B.L				0		1		5	1	1	1	1	1	1	1	0	0	rs	1	0	0	1	1	1	1	rd	d	d	d
	@(d:32,@(d:16,Rs.W),W),ERd				0		1		5	1	1	1	1	1	1	0	0	rs	1	0	0	1	1	1	1	rd	d	d	d	
	MOVA.B.L				0		1		5	1	1	1	1	1	1	1	0	0	rs	1	0	0	1	1	1	1	rd	d	d	d
	@(d:32,@(d:16,ERs.L),W),ERd				0		1		5	1	1	1	1	1	1	0	0	rs	1	0	0	1	1	1	1	rd	d	d	d	
	MOVA.B.L				0		1		5	1	1	1	1	1	1	1	0	0	rs	1	0	0	1	1	1	1	rd	d	d	d
	@(d:32,@(d:32,Rs.B),W),ERd				0		1		5	1	1	1	1	1	1	0	0	rs	1	0	0	1	1	1	1	rd	d	d	d	
	MOVA.B.L				0		1		5	1	1	1	1	1	1	1	0	0	rs	1	0	0	1	1	1	1	rd	d	d	d
	@(d:32,@(d:32,Rs.W),W),ERd				0		1		5	1	1	1	1	1	1	0	0	rs	1	0	0	1	1	1	1	rd	d	d	d	
	MOVA.B.L				0		1		5	1	1	1	1	1	1	1	0	0	rs	1	0	0	1	1	1	1	rd	d	d	d
	@(d:32,@(d:32,ERs.L),W),ERd				0		1		5	1	1	1	1	1	1	0	0	rs	1	0	0	1	1	1	1	rd	d	d	d	
	MOVA.B.L @(d:32,@aa:16.W),ERd				0		1		5	1	1	1	1	1	1	0	0	0	0	rs	1	0	0	1	1	1	rd	a	d	d
	MOVA.B.L @(d:32,@aa:32.W),ERd				0		1		5	1	1	1	1	1	1	0	0	0	0	rs	1	0	0	1	1	1	rd	a	a	d
	MOVA.W.L @(d:32,@ERs.B),ERd				0		1		7	1	1	1	1	1	1	0	0	0	0	rs	1	0	1	0	1	1	rd	d	d	d
	MOVA.W.L @(d:32,@ERs+.B),ERd				0		1		7	1	1	1	1	1	1	0	0	0	0	rs	1	0	1	0	1	1	rd	d	d	d
	MOVA.W.L @(d:32,@ERs-.B),ERd				0		1		7	1	1	1	1	1	1	0	0	0	0	rs	1	0	1	0	1	1	rd	d	d	d
	MOVA.W.L @(d:32,@+ERs.B),ERd				0		1		7	1	1	1	1	1	1	0	0	0	0	rs	1	0	1	0	1	1	rd	d	d	d
	MOVA.W.L @(d:32,@-ERs.B),ERd				0		1		7	1	1	1	1	1	1	0	0	0	0	rs	1	0	1	0	1	1	rd	d	d	d
	MOVA.W.L				0		1		7	1	1	1	1	1	1	0	0	d	d	0	rs	1	0	1	0	1	rd	d	d	d
	@(d:32,@(d:2,ERs.B),ERd				0		1		7	1	1	1	1	1	1	0	0	d	d	0	rs	1	0	1	0	1	rd	d	d	d
	MOVA.W.L				0		1		7	1	1	1	1	1	1	1	0	0	0	rs	1	0	1	0	1	1	rd	d	d	d
	@(d:32,@(d:16,ERs),B),ERd				0		1		7	1	1	1	1	1	1	1	0	0	1	rs	1	0	1	0	1	1	rd	d	d	d
	MOVA.W.L				0		1		7	1	1	1	1	1	1	1	0	1	0	rs	1	0	1	0	1	1	rd	d	d	d
	@(d:32,@(d:32,ERs),B),ERd				0		1		7	1	1	1	1	1	1	0	1	0	0	rs	1	0	1	0	1	1	rd	d	d	d
MOVA.W.L				0		1		7	1	1	1	1	1	1	1	0	1	0	rs	1	0	1	0	1	1	rd	d	d	d	
@(d:32,@(d:16,Rs.B),B),ERd				0		1		7	1	1	1	1	1	1	0	1	0	0	rs	1	0	1	0	1	1	rd	d	d	d	
MOVA.W.L				0		1		7	1	1	1	1	1	1	0	0	0	rs	1	0	1	0	1	1	rd	d	d	d		
@(d:32,@(d:16,Rs.W),B),ERd				0		1		7	1	1	1	1	1	1	0	0	0	rs	1	0	1	0	1	1	rd	d	d	d		
MOVA.W.L				0		1		7	1	1	1	1	1	1	1	0	0	rs	1	0	1	0	1	1	rd	d	d	d		
@(d:32,@(d:16,ERs.L),B),ERd				0		1		7	1	1	1	1	1	1	0	0	0	rs	1	0	1	0	1	1	rd	d	d	d		
MOVA.W.L				0		1		7	1	1	1	1	1	1	1	0	1	1	rs	1	0	1	0	1	1	rd	d	d	d	
@(d:32,@(d:32,Rs.B),B),ERd				0		1		7	1	1	1	1	1	1	0	1	1	1	rs	1	0	1	0	1	1	rd	d	d	d	



Instruction	Mnemonic	Opcode			EA Extension	Opcode			EA Extension									
		15	8	7		0	15	8		7	0							
MOVA	MOVA.W.L				0	1	7	1 1 1 1	1 1 1 0	1	rs	1 0 1 0	1	rd	d	d	d	
	@(d:32,@(d:32,Rs,W),B),ERd																	
	MOVA.W.L				0	1	7	1 1 1 1	1 1 1 1	1	rs	1 0 1 0	1	rd	d	d	d	
	@(d:32,@(d:32,ERs,L),B),ERd																	
	MOVA.W.L @ (d:32,@aa:16,B),ERd				0	1	7	1 1 1 1	0 1 0 0	0	0 0 0 0	1 0 1 0	1	rd	a		d	d
	MOVA.W.L @ (d:32,@aa:32,B),ERd				0	1	7	1 1 1 1	0 1 0 0	1	0 0 0 0	1 0 1 0	1	rd	a	a	d	d
	MOVA.W.L @ (d:32,@ERs,W),ERd				0	1	5	1 1 1 1	0 0 0 0	0	rs	1 0 1 1	1	rd			d	d
	MOVA.W.L @ (d:32,@ERs+,W),ERd				0	1	5	1 1 1 1	1 0 0 0	0	rs	1 0 1 1	1	rd			d	d
	MOVA.W.L @ (d:32,@ERs-,W),ERd				0	1	5	1 1 1 1	1 0 1 0	0	rs	1 0 1 1	1	rd			d	d
	MOVA.W.L @ (d:32,@+ERs,W),ERd				0	1	5	1 1 1 1	1 0 0 1	0	rs	1 0 1 1	1	rd			d	d
	MOVA.W.L @ (d:32,@-ERs,W),ERd				0	1	5	1 1 1 1	1 0 1 1	0	rs	1 0 1 1	1	rd			d	d
	MOVA.W.L				0	1	5	1 1 1 1	0 0 d d	0	rs	1 0 1 1	1	rd			d	d
	@(d:32,@(d:2,ERs),W),ERd																	
	MOVA.W.L				0	1	5	1 1 1 1	1 1 0 0	0	rs	1 0 1 1	1	rd	d		d	d
	@(d:32,@(d:16,ERs),W),ERd																	
	MOVA.W.L				0	1	5	1 1 1 1	1 1 0 0	1	rs	1 0 1 1	1	rd	d	d	d	d
	@(d:32,@(d:32,ERs),W),ERd																	
	MOVA.W.L				0	1	5	1 1 1 1	1 1 0 1	0	rs	1 0 1 1	1	rd	d		d	d
	@(d:32,@(d:16,Rs,B),W),ERd																	
	MOVA.W.L				0	1	5	1 1 1 1	1 1 1 0	0	rs	1 0 1 1	1	rd	d		d	d
@(d:32,@(d:16,ERs,L),W),ERd																		
MOVA.W.L				0	1	5	1 1 1 1	1 1 0 1	1	rs	1 0 1 1	1	rd	d	d	d	d	
@(d:32,@(d:32,Rs,B),W),ERd																		
MOVA.W.L				0	1	5	1 1 1 1	1 1 1 0	1	rs	1 0 1 1	1	rd	d	d	d	d	
@(d:32,@(d:32,Rs,W),W),ERd																		
MOVA.W.L @ (d:32,@aa:16,W),ERd				0	1	5	1 1 1 1	0 1 0 0	0	0 0 0 0	1 0 1 1	1	rd	a		d	d	
MOVA.W.L @ (d:32,@aa:32,W),ERd				0	1	5	1 1 1 1	0 1 0 0	1	0 0 0 0	1 0 1 1	1	rd	a	a	d	d	
MOVA.L.L @ (d:32,@ERs,B),ERd				0	1	7	1 1 1 1	0 0 0 0	0	rs	1 1 0 0	1	rd			d	d	
MOVA.L.L @ (d:32,@ERs+,B),ERd				0	1	7	1 1 1 1	1 0 0 0	0	rs	1 1 0 0	1	rd			d	d	
MOVA.L.L @ (d:32,@ERs-,B),ERd				0	1	7	1 1 1 1	1 0 1 0	0	rs	1 1 0 0	1	rd			d	d	
MOVA.L.L @ (d:32,@+ERs,B),ERd				0	1	7	1 1 1 1	1 0 0 1	0	rs	1 1 0 0	1	rd			d	d	
MOVA.L.L @ (d:32,@-ERs,B),ERd				0	1	7	1 1 1 1	1 0 1 1	0	rs	1 1 0 0	1	rd			d	d	

Instruction	Mnemonic	Opcode			EA Extension	Opcode			EA Extension																						
		15	8	7		0	15	8		7	0																				
MOVA	MOVALL					0	1	7	1 1 1 1		0	0	d	d	0	rs	1	1	0	0	1	1	rd			d	d				
	@(d:32,@(d:2,ERs),B),ERd																														
	MOVALL					0	1	7	1 1 1 1			1	1	0	0	0	rs	1	1	0	0	1	1	rd	d		d	d			
	@(d:32,@(d:16,ERs),B),ERd																														
	MOVALL					0	1	7	1 1 1 1			1	1	0	0	rs	1	1	0	0	1	1	rd	d	d	d	d	d	d		
	@(d:32,@(d:32,ERs),B),ERd																														
	MOVALL					0	1	7	1 1 1 1			1	1	0	1	0	rs	1	1	0	0	1	1	rd	d		d	d	d		
	@(d:32,@(d:16,Rs,B),B),ERd																														
	MOVALL					0	1	7	1 1 1 1			1	1	1	0	0	rs	1	1	0	0	1	1	rd	d		d	d	d	d	
	@(d:32,@(d:16,Rs,W),B),ERd																														
	MOVALL					0	1	7	1 1 1 1			1	1	1	1	0	rs	1	1	0	0	1	1	rd	d		d	d	d	d	
	@(d:32,@(d:16,ERs,L),B),ERd																														
	MOVALL					0	1	7	1 1 1 1			1	1	0	1	1	rs	1	1	0	0	1	1	rd	d	d	d	d	d	d	
	@(d:32,@(d:32,Rs,B),B),ERd																														
	MOVALL					0	1	7	1 1 1 1			1	1	1	0	1	rs	1	1	0	0	1	1	rd	d	d	d	d	d	d	
	@(d:32,@(d:32,Rs,W),B),ERd																														
	MOVALL					0	1	7	1 1 1 1			1	1	1	1	1	rs	1	1	0	0	1	1	rd	d	d	d	d	d	d	
	@(d:32,@(d:32,ERs,L),B),ERd																														
	MOVALL					0	1	7	1 1 1 1			0	1	0	0	0	0	rs	1	1	0	0	1	1	rd	a		d	d	d	
	@(d:32,@aa:16,B),ERd																														
	MOVALL					0	1	7	1 1 1 1			0	1	0	0	1	0	0	rs	1	1	0	0	1	rd	a	a	d	d	d	
	@(d:32,@aa:32,B),ERd																														
	MOVALL					0	1	5	1 1 1 1			0	0	0	0	0	rs	1	1	0	1	1	1	rd			d	d	d		
	@(d:32,@ERs,W),ERd																														
	MOVALL					0	1	5	1 1 1 1			1	0	0	0	0	rs	1	1	0	1	1	1	rd			d	d	d	d	
	@(d:32,@ERs+,W),ERd																														
	MOVALL					0	1	5	1 1 1 1			1	0	1	0	0	rs	1	1	0	1	1	1	rd			d	d	d	d	
	@(d:32,@ERs-,W),ERd																														
MOVALL					0	1	5	1 1 1 1			0	0	d	d	0	rs	1	1	0	1	1	1	rd			d	d	d	d		
@(d:32,@(d:2,ERs),W),ERd																															
MOVALL					0	1	5	1 1 1 1			1	1	0	0	0	rs	1	1	0	1	1	1	rd	d		d	d	d	d		
@(d:32,@(d:16,ERs),W),ERd																															
MOVALL					0	1	5	1 1 1 1			1	1	0	0	1	rs	1	1	0	1	1	1	rd	d	d	d	d	d	d		
@(d:32,@(d:32,ERs),W),ERd																															
MOVALL					0	1	5	1 1 1 1			1	1	0	1	0	rs	1	1	0	1	1	1	rd	d		d	d	d	d		
@(d:32,@(d:16,Rs,B),W),ERd																															
MOVALL					0	1	5	1 1 1 1			1	1	1	0	0	rs	1	1	0	1	1	1	rd	d		d	d	d	d		
@(d:32,@(d:16,Rs,W),W),ERd																															
MOVALL					0	1	5	1 1 1 1			1	1	1	1	0	rs	1	1	0	1	1	1	rd	d		d	d	d	d		
@(d:32,@(d:16,ERs,L),W),ERd																															



Instruction	Mnemonic	Opcode			Opcode			EA Extension	Opcode			EA Extension						
		15	8	7	0	15	8		7	0	15		8	7	0			
MOVA	MOVALL @(d.32,@(d.32,Rs.B),W),ERd				0	1	5	1 1 1 1		1 1 0 1	1	rs	1 1 0 1	1	rd	d	d	d
	MOVALL @(d.32,@(d.32,Rs.W),W),ERd				0	1	5	1 1 1 1		1 1 1 0		rs	1 1 0 1	1	rd	d	d	d
	MOVALL @(d.32,@(d.32,ERs.L),W),ERd				0	1	5	1 1 1 1		1 1 1 1		rs	1 1 0 1	1	rd	d	d	d
	MOVALL @(d.32,@aa:16,W),ERd				0	1	5	1 1 1 1		0 1 0 0	0 0 0 0		1 1 0 1	1	rd	a		d
	MOVALL @(d.32,@aa:32,W),ERd				0	1	5	1 1 1 1		0 1 0 0	1 0 0 0		1 1 0 1	1	rd	a	a	d
MOVFP	MOVFP @aa:16,Rd									6	A		0 1 0 0		rd	a		
MOVMD	MOVMD.B									7	B		1 0 0 1	0 1 0 0				
	MOVMD.W									7	B		1 0 1 0	0 0 1 0	0			
	MOVMD.L									7	B		1 0 1 1	0 0 1 0	0			
MOVSD	MOVSD.B d:16									7	B		1 0 0 0	0 1 0 0	0			d
MOVTP	MOVTP Rs,@aa:16									6	A		1 1 0 0		rs	a		
MULS	MULS.W #xx:4,Rd				0	1	C	0 1 1 0		5	0	x x x x			rd			
	MULS.W Rs,Rd				0	1	C	0 0 1 0		5	0		rs		rd			
	MULS.L #xx:4,ERd				0	1	C	0 1 1 0		5	2	x x x x	0		rd			
	MULS.L ERs,ERd				0	1	C	0 0 1 0		5	2	0	rs	0		rd		
MULU	MULU.W #xx:4,Rd				0	1	C	1 1 1 0		5	0	x x x x			rd			
	MULU.W Rs,Rd				0	1	C	1 0 1 0		5	0		rs		rd			
	MULU.L #xx:4,ERd				0	1	C	1 1 1 0		5	2	x x x x	0		rd			
	MULU.L ERs,ERd				0	1	C	1 0 1 0		5	2	0	rs	0		rd		
MULSU	MULSU.L #xx:4,ERd				0	1	C	0 1 1 1		5	2	x x x x	0		rd			
	MULSU.L ERs,ERd				0	1	C	0 0 1 1		5	2	0	rs	0		rd		
MULSU	MULSU.L #xx:4,ERd				0	1	C	1 1 1 1		5	2	x x x x	0		rd			
	MULSU.L ERs,ERd				0	1	C	1 0 1 1		5	2	0	rs	0		rd		
MULXS	MULXS.B #xx:4,Rd				0	1	C	0 1 0 0		5	0	x x x x			rd			
	MULXS.B Rs,Rd				0	1	C	0 0 0 0		5	0		rs		rd			
	MULXS.W #xx:4,ERd				0	1	C	0 1 0 0		5	2	x x x x	0		rd			
	MULXS.W Rs,ERd				0	1	C	0 0 0 0		5	2		rs	0		rd		
MULXU	MULXU.B #xx:4,Rd				0	1	C	1 1 0 0		5	0	x x x x			rd			

Instruction	Mnemonic	Opcode			Opcode			EA Extension	Opcode			EA Extension			
		15	8	7	0	15	8		7	0	15		8	7	0
MULXU	MULXU.B Rs,Rd									5	0	rs	rd		
	MULXU.W #xc4,ERd				0	1	C	1 1 0 0		5	2	x x x x 0	rd		
	MULXU.W Rs,ERd									5	2	rs	rd		
NEG.B	NEG.B Rd									1	7	8	rd		
	NEG.B @ERd				7	D	0	rd 0 0 0 0		1	7	8	0 0 0 0		
	NEG.B @ERd+	0	1	7	0 1 0 0	6	C	0	rd 1 0 0 0		1	7	8	0 0 0 0	
	NEG.B @ERd-	0	1	7	0 1 1 0	6	C	0	rd 1 0 0 0		1	7	8	0 0 0 0	
	NEG.B @+ERd	0	1	7	0 1 0 1	6	C	0	rd 1 0 0 0		1	7	8	0 0 0 0	
	NEG.B @-ERd	0	1	7	0 1 1 1	6	C	0	rd 1 0 0 0		1	7	8	0 0 0 0	
	NEG.B @(d:2,ERd)	0	1	7	0 1 d d	6	8	0	rd 1 0 0 0		1	7	8	0 0 0 0	
	NEG.B @(d:16,ERd)	0	1	7	0 1 0 0	6	E	0	rd 1 0 0 0 d		1	7	8	0 0 0 0	
	NEG.B @(d:32,ERd)	7	8	0	rd 0 1 0 0	6	A	0 0 1 0	1 0 0 0 d d		1	7	8	0 0 0 0	
	NEG.B @(d:16,Rd,B)	0	1	7	0 1 0 1	6	E	0	rd 1 0 0 0 d		1	7	8	0 0 0 0	
	NEG.B @(d:16,Rd,W)	0	1	7	0 1 1 0	6	E	0	rd 1 0 0 0 d		1	7	8	0 0 0 0	
	NEG.B @(d:16,ERd,L)	0	1	7	0 1 1 1	6	E	0	rd 1 0 0 0 d		1	7	8	0 0 0 0	
	NEG.B @(d:32,Rd,B)	7	8	0	rd 0 1 0 1	6	A	0 0 1 0	1 0 0 0 d d		1	7	8	0 0 0 0	
	NEG.B @(d:32,Rd,W)	7	8	0	rd 0 1 1 0	6	A	0 0 1 0	1 0 0 0 d d		1	7	8	0 0 0 0	
	NEG.B @(d:32,ERd,L)	7	8	0	rd 0 1 1 1	6	A	0 0 1 0	1 0 0 0 d d		1	7	8	0 0 0 0	
	NEG.B @aa:8					7	F	a a a a a a a a		1	7	8	0 0 0 0		
	NEG.B @aa:16					6	A	0 0 0 1	1 0 0 0 a		1	7	8	0 0 0 0	
	NEG.B @aa:32					6	A	0 0 1 1	1 0 0 0 a a		1	7	8	0 0 0 0	
	NEG.W Rd										1	7	9	rd	
	NEG.W @ERd					7	D	1	rd 0 0 0 0		1	7	9	0 0 0 0	
	NEG.W @ERd+	0	1	5	0 1 0 0	6	D	0	rd 1 0 0 0		1	7	9	0 0 0 0	
	NEG.W @ERd-	0	1	5	0 1 1 0	6	D	0	rd 1 0 0 0		1	7	9	0 0 0 0	
	NEG.W @+ERd	0	1	5	0 1 0 1	6	D	0	rd 1 0 0 0		1	7	9	0 0 0 0	
	NEG.W @-ERd	0	1	5	0 1 1 1	6	D	0	rd 1 0 0 0		1	7	9	0 0 0 0	
	NEG.W @(d:2,ERd)	0	1	5	0 1 d d	6	9	0	rd 1 0 0 0		1	7	9	0 0 0 0	
	NEG.W @(d:16,ERd)	0	1	5	0 1 0 0	6	F	0	rd 1 0 0 0 d		1	7	9	0 0 0 0	
	NEG.W @(d:32,ERd)	7	8	0	rd 0 1 0 0	6	B	0 0 1 0	1 0 0 0 d d		1	7	9	0 0 0 0	
NEG.W @(d:16,Rd,B)	0	1	5	0 1 0 1	6	F	0	rd 1 0 0 0 d		1	7	9	0 0 0 0		
NEG.W @(d:16,Rd,W)	0	1	5	0 1 1 0	6	F	0	rd 1 0 0 0 d		1	7	9	0 0 0 0		
NEG.W @(d:16,ERd,L)	0	1	5	0 1 1 1	6	F	0	rd 1 0 0 0 d		1	7	9	0 0 0 0		
NEG.W @(d:32,Rd,B)	7	8	0	rd 0 1 0 1	6	B	0 0 1 0	1 0 0 0 d d		1	7	9	0 0 0 0		



Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension				
		15	8	7	0		15	8	7	0					
MULXU	NEG.W @(d:32,Rd,W)	7	8	0	rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0	d d	1	7	9	0 0 0 0
	NEG.W @(d:32,ERd,L)	7	8	0	rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0	d d	1	7	9	0 0 0 0
	NEG.W @aa:16						6	B	0 0 0 1	1 0 0 0	a	1	7	9	0 0 0 0
	NEG.W @aa:32						6	B	0 0 1 1	1 0 0 0	a a	1	7	9	0 0 0 0
	NEG.L ERd											1	7	B	0 rd
	NEG.L @ERd	0	1	0	0 1 0 0	6	9	0	rd	1 0 0 0		1	7	B	0 0 0 0
	NEG.L @ERd+	0	1	0	0 1 0 0	6	D	0	rd	1 0 0 0		1	7	B	0 0 0 0
	NEG.L @ERd-	0	1	0	0 1 1 0	6	D	0	rd	1 0 0 0		1	7	B	0 0 0 0
	NEG.L @+ERd	0	1	0	0 1 0 1	6	D	0	rd	1 0 0 0		1	7	B	0 0 0 0
	NEG.L @-ERd	0	1	0	0 1 1 1	6	D	0	rd	1 0 0 0		1	7	B	0 0 0 0
	NEG.L @(d:2,ERd)	0	1	0	0 1 d d	6	9	0	rd	1 0 0 0		1	7	B	0 0 0 0
	NEG.L @(d:16,ERd)	0	1	0	0 1 0 0	6	F	0	rd	1 0 0 0	d	1	7	B	0 0 0 0
	NEG.L @(d:32,ERd)	7	8	1	rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0	d d	1	7	B	0 0 0 0
	NEG.L @(d:16,Rd,B)	0	1	0	0 1 0 1	6	F	0	rd	1 0 0 0	d	1	7	B	0 0 0 0
	NEG.L @(d:16,Rd,W)	0	1	0	0 1 1 0	6	F	0	rd	1 0 0 0	d	1	7	B	0 0 0 0
	NEG.L @(d:16,ERd,L)	0	1	0	0 1 1 1	6	F	0	rd	1 0 0 0	d	1	7	B	0 0 0 0
	NEG.L @(d:32,Rd,B)	7	8	1	rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0	d d	1	7	B	0 0 0 0
NEG.L @(d:32,Rd,W)	7	8	1	rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0	d d	1	7	B	0 0 0 0	
NEG.L @(d:32,ERd,L)	7	8	1	rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0	d d	1	7	B	0 0 0 0	
NEG.L @aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0	1 0 0 0	a		1	7	B	0 0 0 0	
NEG.L @aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0	1 0 0 0	a a		1	7	B	0 0 0 0	
NOP	NOP										0	0	0	0	
NOT	NOT.B Rd											1	7	0	rd
	NOT.B @ERd						7	D	0	rd	0 0 0 0		1	7	0 0 0 0 0 0
	NOT.B @ERd+	0	1	7	0 1 0 0	6	C	0	rd	1 0 0 0		1	7	0 0 0 0 0 0	
	NOT.B @ERd-	0	1	7	0 1 1 0	6	C	0	rd	1 0 0 0		1	7	0 0 0 0 0 0	
	NOT.B @+ERd	0	1	7	0 1 0 1	6	C	0	rd	1 0 0 0		1	7	0 0 0 0 0 0	
	NOT.B @-ERd	0	1	7	0 1 1 1	6	C	0	rd	1 0 0 0		1	7	0 0 0 0 0 0	
	NOT.B @(d:2,ERd)	0	1	7	0 1 d d	6	8	0	rd	1 0 0 0		1	7	0 0 0 0 0 0	
	NOT.B @(d:16,ERd)	0	1	7	0 1 0 0	6	E	0	rd	1 0 0 0	d	1	7	0 0 0 0 0 0	
	NOT.B @(d:32,ERd)	7	8	0	rd	0 1 0 0	6	A	0 0 1 0	1 0 0 0	d d	1	7	0 0 0 0 0 0	
	NOT.B @(d:16,Rd,B)	0	1	7	0 1 0 1	6	E	0	rd	1 0 0 0	d	1	7	0 0 0 0 0 0	
NOT.B @(d:16,Rd,W)	0	1	7	0 1 1 0	6	E	0	rd	1 0 0 0	d	1	7	0 0 0 0 0 0		
NOT.B @(d:16,ERd,L)	0	1	7	0 1 1 1	6	E	0	rd	1 0 0 0	d	1	7	0 0 0 0 0 0		

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension
		15	8	7	0	15	8	7	0		15	8	7	0	
NOT	NOT.B @(d:32,Rd.B)	7	8	0	rd	0 1 0 1	6	A	0 0 1 0	1 1 0 0 0 0	d d	1	7	0	0 0 0 0 0
	NOT.B @(d:32,Rd.W)	7	8	0	rd	0 1 1 0	6	A	0 0 1 0	1 1 0 0 0 0	d d	1	7	0	0 0 0 0 0
	NOT.B @(d:32,ERd.L)	7	8	0	rd	0 1 1 1	6	A	0 0 1 0	1 1 0 0 0 0	d d	1	7	0	0 0 0 0 0
	NOT.B @aa:8						7	F	a a a a	a a a a a		1	7	0	0 0 0 0 0
	NOT.B @aa:16						6	A	0 0 0 0 1	1 0 0 0 0 a		1	7	0	0 0 0 0 0
	NOT.B @aa:32						6	A	0 0 1 1	1 1 0 0 0 a	a	1	7	0	0 0 0 0 0
	NOT.W Rd											1	7	1	rd
	NOT.W @ERd						7	D	1 rd	0 0 0 0 0		1	7	1	0 0 0 0 0
	NOT.W @ERd+	0	1	5	0 1 0 0	6	D	0 rd	1 0 0 0 0		1	7	1	0 0 0 0 0	
	NOT.W @ERd-	0	1	5	0 1 1 0	6	D	0 rd	1 0 0 0 0		1	7	1	0 0 0 0 0	
	NOT.W @+ERd	0	1	5	0 1 0 1	6	D	0 rd	1 0 0 0 0		1	7	1	0 0 0 0 0	
	NOT.W @-ERd	0	1	5	0 1 1 1	6	D	0 rd	1 0 0 0 0		1	7	1	0 0 0 0 0	
	NOT.W @(d:2,ERd)	0	1	5	0 1 d d	6	9	0 rd	1 0 0 0 0		1	7	1	0 0 0 0 0	
	NOT.W @(d:16,ERd)	0	1	5	0 1 0 0	6	F	0 rd	1 0 0 0 0 d		1	7	1	0 0 0 0 0	
	NOT.W @(d:32,ERd)	7	8	0	rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0 0 d d		1	7	1	0 0 0 0 0
	NOT.W @(d:16,Rd.B)	0	1	5	0 1 0 1	6	F	0 rd	1 0 0 0 0 d		1	7	1	0 0 0 0 0	
	NOT.W @(d:16,Rd.W)	0	1	5	0 1 1 0	6	F	0 rd	1 0 0 0 0 d		1	7	1	0 0 0 0 0	
	NOT.W @(d:16,ERd.L)	0	1	5	0 1 1 1	6	F	0 rd	1 0 0 0 0 d		1	7	1	0 0 0 0 0	
	NOT.W @(d:32,Rd.B)	7	8	0	rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0 0 d d		1	7	1	0 0 0 0 0
	NOT.W @(d:32,Rd.W)	7	8	0	rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0 0 d d		1	7	1	0 0 0 0 0
	NOT.W @(d:32,ERd.L)	7	8	0	rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0 0 d d		1	7	1	0 0 0 0 0
	NOT.W @aa:16						6	B	0 0 0 0 1	1 0 0 0 0 a		1	7	1	0 0 0 0 0
	NOT.W @aa:32						6	B	0 0 1 1	1 0 0 0 0 a a		1	7	1	0 0 0 0 0
	NOT.L ERd											1	7	3	0 rd
	NOT.L @ERd	0	1	0	0 1 0 0	6	9	0 rd	1 0 0 0 0		1	7	3	0 0 0 0 0	
	NOT.L @ERd+	0	1	0	0 1 0 0	6	D	0 rd	1 0 0 0 0		1	7	3	0 0 0 0 0	
	NOT.L @ERd-	0	1	0	0 1 1 0	6	D	0 rd	1 0 0 0 0		1	7	3	0 0 0 0 0	
	NOT.L @+ERd	0	1	0	0 1 0 1	6	D	0 rd	1 0 0 0 0		1	7	3	0 0 0 0 0	
	NOT.L @-ERd	0	1	0	0 1 1 1	6	D	0 rd	1 0 0 0 0		1	7	3	0 0 0 0 0	
	NOT.L @(d:2,ERd)	0	1	0	0 1 d d	6	9	0 rd	1 0 0 0 0		1	7	3	0 0 0 0 0	
	NOT.L @(d:16,ERd)	0	1	0	0 1 0 0	6	F	0 rd	1 0 0 0 0 d		1	7	3	0 0 0 0 0	
	NOT.L @(d:32,ERd)	7	8	1	rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0 0 d d		1	7	3	0 0 0 0 0
NOT.L @(d:16,Rd.B)	0	1	0	0 1 0 1	6	F	0 rd	1 0 0 0 0 d		1	7	3	0 0 0 0 0		
NOT.L @(d:16,Rd.W)	0	1	0	0 1 1 0	6	F	0 rd	1 0 0 0 0 d		1	7	3	0 0 0 0 0		



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension								
		15	8	7	0	15	8	7	0		15	8	7	0									
NOT	NOT.L @(d:16,ERd.L)	0	1	0	0 1 1 1	6	F	0 rd	1 0 0 0 0 d		1	7	3	0 0 0 0									
	NOT.L @(d:32,Rd.B)	7	8	1 rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0 0 d d		1	7	3	0 0 0 0									
	NOT.L @(d:32,Rd.W)	7	8	1 rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0 0 d d		1	7	3	0 0 0 0									
	NOT.L @(d:32,ERd.L)	7	8	1 rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0 0 d d		1	7	3	0 0 0 0									
	NOT.L @aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0	1 0 0 0 0 a		1	7	3	0 0 0 0									
	NOT.L @aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0	1 0 0 0 0 a a		1	7	3	0 0 0 0									
OR	OR.B #xc:8,Rd													C	rd	x x x x	x x x x						
	OR.B #xc:8,@ERd													7	D	0 rd	0 0 0 0 0		C	0 0 0 0 0	x x x x	x x x x	
	OR.B #xc:8,@ERd+	0	1	7	0 1 0 0	6	C	0 rd	1 0 0 0 0		C	0 0 0 0 0	x x x x	x x x x									
	OR.B #xc:8,@ERd-	0	1	7	0 1 1 0	6	C	0 rd	1 0 0 0 0		C	0 0 0 0 0	x x x x	x x x x									
	OR.B #xc:8,@+ERd	0	1	7	0 1 0 1	6	C	0 rd	1 0 0 0 0		C	0 0 0 0 0	x x x x	x x x x									
	OR.B #xc:8,@-ERd	0	1	7	0 1 1 1	6	C	0 rd	1 0 0 0 0		C	0 0 0 0 0	x x x x	x x x x									
	OR.B #xc:8,@(d:2,ERd)	0	1	7	0 1 d d	6	8	0 rd	1 0 0 0 0		C	0 0 0 0 0	x x x x	x x x x									
	OR.B #xc:8,@(d:16,ERd)	0	1	7	0 1 0 0	6	E	0 rd	1 0 0 0 0 d		C	0 0 0 0 0	x x x x	x x x x									
	OR.B #xc:8,@(d:32,ERd)	7	8	0 rd	0 1 0 0	6	A	0 0 1 0	1 0 0 0 0 d d		C	0 0 0 0 0	x x x x	x x x x									
	OR.B #xc:8,@(d:16,Rd.B)	0	1	7	0 1 0 1	6	E	0 rd	1 0 0 0 0 d		C	0 0 0 0 0	x x x x	x x x x									
	OR.B #xc:8,@(d:16,Rd.W)	0	1	7	0 1 1 0	6	E	0 rd	1 0 0 0 0 d		C	0 0 0 0 0	x x x x	x x x x									
	OR.B #xc:8,@(d:16,ERd.L)	0	1	7	0 1 1 1	6	E	0 rd	1 0 0 0 0 d		C	0 0 0 0 0	x x x x	x x x x									
	OR.B #xc:8,@(d:32,Rd.B)	7	8	0 rd	0 1 0 1	6	A	0 0 1 0	1 0 0 0 0 d d		C	0 0 0 0 0	x x x x	x x x x									
	OR.B #xc:8,@(d:32,Rd.W)	7	8	0 rd	0 1 1 0	6	A	0 0 1 0	1 0 0 0 0 d d		C	0 0 0 0 0	x x x x	x x x x									
	OR.B #xc:8,@(d:32,ERd.L)	7	8	0 rd	0 1 1 1	6	A	0 0 1 0	1 0 0 0 0 d d		C	0 0 0 0 0	x x x x	x x x x									
	OR.B #xc:8,@aa:8													7	F	a a a a	a a a a		C	0 0 0 0 0	x x x x	x x x x	
	OR.B #xc:8,@aa:16													6	A	0 0 0 0	1 0 0 0 0 a		C	0 0 0 0 0	x x x x	x x x x	
	OR.B #xc:8,@aa:32													6	A	0 0 1 1	1 0 0 0 0 a a		C	0 0 0 0 0	x x x x	x x x x	
	OR.B Rs,Rd																		1	4	rs	rd	
	OR.B Rs,@ERd													7	D	0 rd	0 0 0 0 0		1	4	rs	0 0 0 0 0	
	OR.B Rs,@ERd+	0	1	7	1 0 0 1				1 0 0 0 0		1	0 0 0 0	0 rd	0 1 0 0	rs								
	OR.B Rs,@ERd-	0	1	7	1 0 0 1				1 0 1 0 0		0 rd	0 1 0 0		rs									
	OR.B Rs,@+ERd	0	1	7	1 0 0 1				1 0 0 1 0		0 rd	0 1 0 0		rs									
OR.B Rs,@-ERd	0	1	7	1 0 0 1				1 0 1 1 0		0 rd	0 1 0 0		rs										
OR.B Rs,@(d:2,ERd)	0	1	7	1 0 0 1				0 0 d d 0		0 rd	0 1 0 0		rs										
OR.B Rs,@(d:16,ERd)								1 1 0 0 0		0 rd	0 1 0 0		rs	d									
OR.B Rs,@(d:32,ERd)								1 1 0 0 1		1 rd	0 1 0 0		rs	d d									
OR.B Rs,@(d:16,Rd.B)								1 1 0 1 0		0 rd	0 1 0 0		rs	d									

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension												
		15	8	7	0		15	8	7	0													
OR	OR.B Rs, @(d:16,Rd,W)			0	1	7	1	0	0	1	1	1	1	0	rd	0	1	0	0	rs	d		
	OR.B Rs, @(d:16,ERd,L)			0	1	7	1	0	0	1	1	1	1	0	rd	0	1	0	0	rs	d		
	OR.B Rs, @(d:32,Rd,B)			0	1	7	1	0	0	1	1	1	0	1	rd	0	1	0	0	rs	d	d	
	OR.B Rs, @(d:32,Rd,W)			0	1	7	1	0	0	1	1	1	1	0	rd	0	1	0	0	rs	d	d	
	OR.B Rs, @(d:32,ERd,L)			0	1	7	1	0	0	1	1	1	1	1	rd	0	1	0	0	rs	d	d	
	OR.B Rs, @aa:8			7	F	a	a	a	a	a	a	a	a	a	1	4	rs	0	0	0	0	0	0
	OR.B Rs, @aa:16			6	A	0	0	0	1	1	0	0	0	0	a	1	4	rs	0	0	0	0	0
	OR.B Rs, @aa:32			6	A	0	0	1	1	1	0	0	0	0	a	a	1	4	rs	0	0	0	0
	OR.B @ERs,Rd			7	C	0	rs	0	0	0	0	0	0	0	1	4	0	0	0	0	rd		
	OR.B @ERs+,Rd			0	1	7	1	0	1	0	1	0	0	0	0	rs	0	1	0	0	rd		
	OR.B @ERs-,Rd			0	1	7	1	0	1	0	0	1	0	0	rs	0	1	0	0	rd			
	OR.B @+ERs,Rd			0	1	7	1	0	1	0	1	0	0	0	rs	0	1	0	0	rd			
	OR.B @-ERs,Rd			0	1	7	1	0	1	0	0	1	0	0	rs	0	1	0	0	rd			
	OR.B @(d:2,ERs),Rd			0	1	7	1	0	1	0	0	0	d	d	0	rs	0	1	0	0	rd		
	OR.B @(d:16,ERs),Rd			0	1	7	1	0	1	0	1	1	0	0	0	rs	0	1	0	0	rd	d	
	OR.B @(d:32,ERs),Rd			0	1	7	1	0	1	0	1	1	0	0	1	rs	0	1	0	0	rd	d	d
	OR.B @(d:16,Rs,B),Rd			0	1	7	1	0	1	0	1	1	0	1	0	rs	0	1	0	0	rd	d	
	OR.B @(d:16,Rs,W),Rd			0	1	7	1	0	1	0	1	1	1	0	0	rs	0	1	0	0	rd	d	
	OR.B @(d:16,ERs,L),Rd			0	1	7	1	0	1	0	1	1	1	1	0	rs	0	1	0	0	rd	d	
	OR.B @(d:32,Rs,B),Rd			0	1	7	1	0	1	0	1	1	0	1	1	rs	0	1	0	0	rd	d	d
	OR.B @(d:32,Rs,W),Rd			0	1	7	1	0	1	0	1	1	1	0	1	rs	0	1	0	0	rd	d	d
	OR.B @(d:32,ERs,L),Rd			0	1	7	1	0	1	0	1	1	1	1	1	rs	0	1	0	0	rd	d	d
	OR.B @aa:8,Rd			7	E	a	a	a	a	a	a	a	a	a	1	4	0	0	0	0	rd		
	OR.B @aa:16,Rd			6	A	0	0	0	1	1	0	0	0	0	a	1	4	0	0	0	0	rd	
	OR.B @aa:32,Rd			6	A	0	0	1	1	1	0	0	0	0	a	a	1	4	0	0	0	0	rd
	OR.B @ERs,@ERd			7	C	0	rs	0	1	0	1	0	0	0	0	rd	0	1	0	0	0	0	0
	OR.B @ERs,@ERd+			7	C	0	rs	0	1	0	1	1	0	0	0	rd	0	1	0	0	0	0	0
	OR.B @ERs,@ERd-			7	C	0	rs	0	1	0	1	0	1	0	0	rd	0	1	0	0	0	0	0
	OR.B @ERs,@+ERd			7	C	0	rs	0	1	0	1	1	0	0	1	rd	0	1	0	0	0	0	0
	OR.B @ERs,@-ERd			7	C	0	rs	0	1	0	1	0	1	0	0	rd	0	1	0	0	0	0	0
	OR.B @ERs,@(d:2,ERd)			7	C	0	rs	0	1	0	1	0	0	d	d	0	rd	0	1	0	0	0	0
	OR.B @ERs,@(d:16,ERd)			7	C	0	rs	0	1	0	1	1	1	0	0	rd	0	1	0	0	0	0	0
OR.B @ERs,@(d:32,ERd)			7	C	0	rs	0	1	0	1	1	1	0	0	rd	0	1	0	0	0	0	0	
OR.B @ERs,@(d:16,Rd,B)			7	C	0	rs	0	1	0	1	1	1	0	1	rd	0	1	0	0	0	0	0	



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
OR	OR.B @ERs,@(d:16,Rd,W)				7	C	0	rs	0 1 0 1		1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d	
	OR.B @ERs,@(d:16,ERd,L)				7	C	0	rs	0 1 0 1		1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d	
	OR.B @ERs,@(d:32,Rd,B)				7	C	0	rs	0 1 0 1		1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d d	
	OR.B @ERs,@(d:32,Rd,W)				7	C	0	rs	0 1 0 1		1 1 1 0	1	rd	0 1 0 0	0 0 0 0	d d	
	OR.B @ERs,@(d:32,ERd,L)				7	C	0	rs	0 1 0 1		1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d d	
	OR.B @ERs,@aa:16				7	C	0	rs	0 1 0 1		0 1 0 0	0 0 0 0	0 1 0 0	0 0 0 0	a		
	OR.B @ERs,@aa:32				7	C	0	rs	0 1 0 1		0 1 0 0	1 0 0 0	0 1 0 0	0 0 0 0	a a		
	OR.B @ERs+,@ERd	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		0 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	OR.B @ERs+,@ERd+	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	OR.B @ERs+,@ERd-	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 0 1 0	0	rd	0 1 0 0	0 0 0 0	
	OR.B @ERs+,@+ERd	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 0 0 1	0	rd	0 1 0 0	0 0 0 0	
	OR.B @ERs+,@-ERd	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 0 1 1	0	rd	0 1 0 0	0 0 0 0	
	OR.B @ERs+,@(d2,ERd)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		0 0 d d	0	rd	0 1 0 0	0 0 0 0	
	OR.B @ERs+,@(d:16,ERd)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 1 0 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.B @ERs+,@(d:32,ERd)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 1 0 0	1	rd	0 1 0 0	0 0 0 0	d d
	OR.B @ERs+,@(d:16,Rd,B)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 1 0 1	0	rd	0 1 0 0	0 0 0 0	d
	OR.B @ERs+,@(d:16,Rd,W)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.B @ERs+,@(d:16,ERd,L)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d
	OR.B @ERs+,@(d:32,Rd,B)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.B @ERs+,@(d:32,Rd,W)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 1 1 0	1	rd	0 1 0 0	0 0 0 0	d d
	OR.B @ERs+,@(d:32,ERd,L)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.B @ERs+,@aa:16	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		0 1 0 0	0 0 0 0	0 1 0 0	0 0 0 0	a	
	OR.B @ERs+,@aa:32	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0		0 1 0 0	1 0 0 0	0 1 0 0	0 0 0 0	a a	
	OR.B @ERs,@ERd	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		0 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	OR.B @ERs,@ERd+	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	OR.B @ERs,@ERd-	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 0 1 0	0	rd	0 1 0 0	0 0 0 0	
	OR.B @ERs,@+ERd	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 0 0 1	0	rd	0 1 0 0	0 0 0 0	
	OR.B @ERs,@-ERd	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 0 1 1	0	rd	0 1 0 0	0 0 0 0	
OR.B @ERs,@(d2,ERd)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		0 0 d d	0	rd	0 1 0 0	0 0 0 0		
OR.B @ERs,@(d:16,ERd)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 1 0 0	0	rd	0 1 0 0	0 0 0 0	d	
OR.B @ERs,@(d:32,ERd)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 1 0 0	1	rd	0 1 0 0	0 0 0 0	d d	
OR.B @ERs,@(d:16,Rd,B)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 1 0 1	0	rd	0 1 0 0	0 0 0 0	d	
OR.B @ERs,@(d:16,Rd,W)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d	
OR.B @ERs,@(d:16,ERd,L)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0		1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d	

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension			
		15	8	7	0	15	8	7	0		15	8	7	0				
OR	OR.B @ERs, @(d:32,Rd.B)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0 0		1 1 0 0 1	1	rd	0 1 0 0 0	0 0 0 0 0	d	d
	OR.B @ERs, @(d:32,Rd.W)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0 0		1 1 1 0 1	1	rd	0 1 0 0 0	0 0 0 0 0	d	d
	OR.B @ERs, @(d:32,ERd.L)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0 0		1 1 1 1 1	1	rd	0 1 0 0 0	0 0 0 0 0	d	d
	OR.B @ERs, @aa:16	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0 0		0 1 0 0 0	0	0 0 0 0 0	0 1 0 0 0	0 0 0 0 0	a	
	OR.B @ERs, @aa:32	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0 0		0 1 0 0 0	1	0 0 0 0 0	0 1 0 0 0	0 0 0 0 0	a	a
	OR.B @+ERs, @ERd	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0 0		0 0 0 0 0	0	rd	0 1 0 0 0	0 0 0 0 0		
	OR.B @+ERs, @ERd+	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0 0		1 0 0 0 0	0	rd	0 1 0 0 0	0 0 0 0 0		
	OR.B @+ERs, @ERd-	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0 0		1 0 1 0 0	0	rd	0 1 0 0 0	0 0 0 0 0		
	OR.B @+ERs, @+ERd	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0 0		1 0 0 0 1	0	rd	0 1 0 0 0	0 0 0 0 0		
	OR.B @+ERs, @-ERd	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0 0		1 0 1 1 0	0	rd	0 1 0 0 0	0 0 0 0 0		
	OR.B @+ERs, @(d:2,ERd)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0 0		0 0 d d	0	rd	0 1 0 0 0	0 0 0 0 0		
	OR.B @+ERs, @(d:16,ERd)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0 0		1 1 0 0 0	0	rd	0 1 0 0 0	0 0 0 0 0	d	
	OR.B @+ERs, @(d:32,ERd)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0 0		1 1 0 0 1	1	rd	0 1 0 0 0	0 0 0 0 0	d	d
	OR.B @+ERs, @(d:16,Rd.B)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0 0		1 1 0 1 0	0	rd	0 1 0 0 0	0 0 0 0 0	d	
	OR.B @+ERs, @(d:16,Rd.W)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0 0		1 1 1 0 0	0	rd	0 1 0 0 0	0 0 0 0 0	d	
	OR.B @+ERs, @(d:16,ERd.L)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0 0		1 1 1 1 0	0	rd	0 1 0 0 0	0 0 0 0 0	d	
	OR.B @+ERs, @(d:32,Rd.B)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0 0		1 1 0 1 1	1	rd	0 1 0 0 0	0 0 0 0 0	d	d
	OR.B @+ERs, @(d:32,Rd.W)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0 0		1 1 1 0 1	1	rd	0 1 0 0 0	0 0 0 0 0	d	d
	OR.B @+ERs, @(d:32,ERd.L)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0 0		1 1 1 1 1	1	rd	0 1 0 0 0	0 0 0 0 0	d	d
	OR.B @+ERs, @aa:16	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0 0		0 1 0 0 0	0	0 0 0 0 0	0 1 0 0 0	0 0 0 0 0	a	
	OR.B @+ERs, @aa:32	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0 0		0 1 0 0 0	1	0 0 0 0 0	0 1 0 0 0	0 0 0 0 0	a	a
	OR.B @-ERs, @ERd	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		0 0 0 0 0	0	rd	0 1 0 0 0	0 0 0 0 0		
	OR.B @-ERs, @ERd+	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		1 0 0 0 0	0	rd	0 1 0 0 0	0 0 0 0 0		
	OR.B @-ERs, @ERd-	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		1 0 1 0 0	0	rd	0 1 0 0 0	0 0 0 0 0		
	OR.B @-ERs, @+ERd	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		1 0 0 0 1	0	rd	0 1 0 0 0	0 0 0 0 0		
	OR.B @-ERs, @-ERd	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		1 0 1 1 0	0	rd	0 1 0 0 0	0 0 0 0 0		
	OR.B @-ERs, @(d:2,ERd)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		0 0 d d	0	rd	0 1 0 0 0	0 0 0 0 0		
	OR.B @-ERs, @(d:16,ERd)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		1 1 0 0 0	0	rd	0 1 0 0 0	0 0 0 0 0	d	
	OR.B @-ERs, @(d:32,ERd)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		1 1 0 0 1	1	rd	0 1 0 0 0	0 0 0 0 0	d	d
	OR.B @-ERs, @(d:16,Rd.B)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		1 1 0 1 0	0	rd	0 1 0 0 0	0 0 0 0 0	d	
	OR.B @-ERs, @(d:16,Rd.W)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		1 1 1 0 0	0	rd	0 1 0 0 0	0 0 0 0 0	d	
	OR.B @-ERs, @(d:16,ERd.L)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		1 1 1 1 0	0	rd	0 1 0 0 0	0 0 0 0 0	d	
OR.B @-ERs, @(d:32,Rd.B)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		1 1 0 1 1	1	rd	0 1 0 0 0	0 0 0 0 0	d	d	
OR.B @-ERs, @(d:32,Rd.W)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0 0		1 1 1 0 1	1	rd	0 1 0 0 0	0 0 0 0 0	d	d	



Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension			
		15	8	7	0	15	8	7	0		15	8	7	0				
OR	OR.B @-ERs, @(d:32,ERd,L)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0		1 1 1 1	1	rd	0 1 0 0	0 0 0 0	0	d d
	OR.B @-ERs, @aa:16	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0		0 1 0 0	0	0 0 0 0	0 1 0 0	0 0 0 0	0	a
	OR.B @-ERs, @aa:32	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0		0 1 0 0	1	0 0 0 0	0 1 0 0	0 0 0 0	0	a a
	OR.B @(d:2,ERs), @ERd	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		0 0 0 0	0	rd	0 1 0 0	0 0 0 0	0	
	OR.B @(d:2,ERs), @ERd+	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 0 0 0	0	rd	0 1 0 0	0 0 0 0	0	
	OR.B @(d:2,ERs), @ERd-	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 0 1 0	0	rd	0 1 0 0	0 0 0 0	0	
	OR.B @(d:2,ERs), @+ERd	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 0 0 1	0	rd	0 1 0 0	0 0 0 0	0	
	OR.B @(d:2,ERs), @-ERd	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 0 1 1	0	rd	0 1 0 0	0 0 0 0	0	
	OR.B @(d:2,ERs), @(d:2,ERd)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		0 0 d d	0	rd	0 1 0 0	0 0 0 0	0	
	OR.B @(d:2,ERs), @(d:16,ERd)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 1 0 0	0	rd	0 1 0 0	0 0 0 0	0	d
	OR.B @(d:2,ERs), @(d:32,ERd)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 1 0 0	1	rd	0 1 0 0	0 0 0 0	0	d d
	OR.B @(d:2,ERs), @(d:16,Rd,B)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 1 0 1	0	rd	0 1 0 0	0 0 0 0	0	d
	OR.B @(d:2,ERs), @(d:16,Rd,W)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 1 1 0	0	rd	0 1 0 0	0 0 0 0	0	d
	OR.B @(d:2,ERs), @(d:16,ERd,L)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 1 1 1	0	rd	0 1 0 0	0 0 0 0	0	d
	OR.B @(d:2,ERs), @(d:32,Rd,B)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 1 0 1	1	rd	0 1 0 0	0 0 0 0	0	d d
	OR.B @(d:2,ERs), @(d:32,Rd,W)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 1 1 0	1	rd	0 1 0 0	0 0 0 0	0	d d
	OR.B @(d:2,ERs), @(d:32,ERd,L)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		1 1 1 1	1	rd	0 1 0 0	0 0 0 0	0	d d
	OR.B @(d:2,ERs), @aa:16	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		0 1 0 0	0	0 0 0 0	0 1 0 0	0 0 0 0	0	a
	OR.B @(d:2,ERs), @aa:32	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0		0 1 0 0	1	0 0 0 0	0 1 0 0	0 0 0 0	0	a a
	OR.B @(d:16,ERs), @ERd	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 1 0 0	0 0 0 0	0	
	OR.B @(d:16,ERs), @ERd+	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 1 0 0	0 0 0 0	0	
	OR.B @(d:16,ERs), @ERd-	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 0 0	0 0 0 0	0	
	OR.B @(d:16,ERs), @+ERd	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 0 0	0 0 0 0	0	
	OR.B @(d:16,ERs), @-ERd	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 1 0 0	0 0 0 0	0	
	OR.B @(d:16,ERs), @(d:2,ERd)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 1 0 0	0 0 0 0	0	
	OR.B @(d:16,ERs), @(d:16,ERd)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 1 0 0	0 0 0 0	0	d
	OR.B @(d:16,ERs), @(d:32,ERd)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 1 0 0	0 0 0 0	0	d d
	OR.B @(d:16,ERs), @(d:16,Rd,B)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 1 0 0	0 0 0 0	0	d
OR.B @(d:16,ERs), @(d:16,Rd,W)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 1 0 0	0 0 0 0	0	d	
OR.B @(d:16,ERs), @(d:16,ERd,L)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 1 0 0	0 0 0 0	0	d	
OR.B @(d:16,ERs), @(d:32,Rd,B)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	0	d d	
OR.B @(d:16,ERs), @(d:32,Rd,W)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 1 0 0	0 0 0 0	0	d d	
OR.B @(d:16,ERs), @(d:32,ERd,L)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 1 0 0	0 0 0 0	0	d d	
OR.B @(d:16,ERs), @aa:16	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	0 1 0 0	0	0 0 0 0	0 1 0 0	0 0 0 0	0	a	

Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension
		15	8	7	0	15	8	7	0		15	8	7	0	
OR	OR.B @(d:16,ERs),@aa:32	0	1	7	0 1 0 0	6	E	0 rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 0 0	0 0 0 0	a a
	OR.B @(d:32,ERs),@ERd	7	8	0	rs 0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	0 0 0 0	0 rd	0 1 0 0	0 0 0 0	
	OR.B @(d:32,ERs),@ERd+	7	8	0	rs 0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 0	0 rd	0 1 0 0	0 0 0 0	
	OR.B @(d:32,ERs),@ERd-	7	8	0	rs 0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 0	0 rd	0 1 0 0	0 0 0 0	
	OR.B @(d:32,ERs),@+ERd	7	8	0	rs 0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 1	0 rd	0 1 0 0	0 0 0 0	
	OR.B @(d:32,ERs),@-ERd	7	8	0	rs 0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 1	0 rd	0 1 0 0	0 0 0 0	
	OR.B @(d:32,ERs),@(d:2,ERd)	7	8	0	rs 0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	0 0 d d	0 rd	0 1 0 0	0 0 0 0	
	OR.B @(d:32,ERs),@(d:16,ERd)	7	8	0	rs 0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	0 rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:32,ERs),@(d:32,ERd)	7	8	0	rs 0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	1 rd	0 1 0 0	0 0 0 0	d d
	OR.B @(d:32,ERs),@(d:16,Rd,B)	7	8	0	rs 0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	0 rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:32,ERs),@(d:16,Rd,W)	7	8	0	rs 0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	0 rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:32,ERs),@(d:16,ERd,L)	7	8	0	rs 0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	0 rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:32,ERs),@(d:32,Rd,B)	7	8	0	rs 0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	1 rd	0 1 0 0	0 0 0 0	d d
	OR.B @(d:32,ERs),@(d:32,Rd,W)	7	8	0	rs 0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	1 rd	0 1 0 0	0 0 0 0	d d
	OR.B @(d:32,ERs),@(d:32,ERd,L)	7	8	0	rs 0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	1 rd	0 1 0 0	0 0 0 0	d d
	OR.B @(d:32,ERs),@aa:16	7	8	0	rs 0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	0 0 0 0	0 1 0 0	0 0 0 0	a
	OR.B @(d:32,ERs),@aa:32	7	8	0	rs 0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	1 0 0 0	0 1 0 0	0 0 0 0	a a
	OR.B @(d:16,Rs,B),@ERd	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	0 0 0 0	0 rd	0 1 0 0	0 0 0 0	
	OR.B @(d:16,Rs,B),@ERd+	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 0 0 0	0 rd	0 1 0 0	0 0 0 0	
	OR.B @(d:16,Rs,B),@ERd-	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 0 1 0	0 rd	0 1 0 0	0 0 0 0	
	OR.B @(d:16,Rs,B),@+ERd	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 0 0 1	0 rd	0 1 0 0	0 0 0 0	
	OR.B @(d:16,Rs,B),@-ERd	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 0 1 1	0 rd	0 1 0 0	0 0 0 0	
	OR.B @(d:16,Rs,B),@(d:2,ERd)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	0 0 d d	0 rd	0 1 0 0	0 0 0 0	
	OR.B @(d:16,Rs,B),@(d:16,ERd)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 1 0 0	0 rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:16,Rs,B),@(d:32,ERd)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 1 0 0	1 rd	0 1 0 0	0 0 0 0	d d
	OR.B @(d:16,Rs,B),@(d:16,Rd,B)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 1 0 1	0 rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:16,Rs,B),@(d:16,Rd,W)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 1 1 0	0 rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:16,Rs,B),@(d:16,ERd,L)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 1 1 1	0 rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:16,Rs,B),@(d:32,Rd,B)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 1 0 1	1 rd	0 1 0 0	0 0 0 0	d d
	OR.B @(d:16,Rs,B),@(d:32,Rd,W)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 1 1 0	1 rd	0 1 0 0	0 0 0 0	d d
OR.B @(d:16,Rs,B),@(d:32,ERd,L)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 1 1 1	1 rd	0 1 0 0	0 0 0 0	d d	
OR.B @(d:16,Rs,B),@aa:16	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 1 0 0	0 0 0 0	a	
OR.B @(d:16,Rs,B),@aa:32	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 0 0	0 0 0 0	a a	
OR.B @(d:16,Rs,W),@ERd	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d	0 0 0 0	0 rd	0 1 0 0	0 0 0 0		



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
OR	OR.B @(d:16,Rs,W),@ERd+	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	OR.B @(d:16,Rs,W),@ERd-	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 0 0	0 0 0 0	
	OR.B @(d:16,Rs,W),@+ERd	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 0 0	0 0 0 0	
	OR.B @(d:16,Rs,W),@-ERd	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 1 0 0	0 0 0 0	
	OR.B @(d:16,Rs,W),@(d:2,ERd)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 1 0 0	0 0 0 0	
	OR.B @(d:16,Rs,W),@(d:16,ERd)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:16,Rs,W),@(d:32,ERd)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 1 0 0	0 0 0 0	d d
	OR.B @(d:16,Rs,W),@(d:16,Rd,B)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:16,Rs,W),@(d:16,Rd,W)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:16,Rs,W),@(d:16,ERdL)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:16,Rs,W),@(d:32,Rd,B)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.B @(d:16,Rs,W),@(d:32,Rd,W)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 1 0 0	0 0 0 0	d d
	OR.B @(d:16,Rs,W),@(d:32,ERdL)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.B @(d:16,Rs,W),@aa:16	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 1 0 0	0 0 0 0	a	
	OR.B @(d:16,Rs,W),@aa:32	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 0 0	0 0 0 0	a a	
	OR.B @(d:16,ERs,L),@ERd	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	OR.B @(d:16,ERs,L),@ERd+	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	OR.B @(d:16,ERs,L),@ERd-	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 0 0	0 0 0 0	
	OR.B @(d:16,ERs,L),@+ERd	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 0 0	0 0 0 0	
	OR.B @(d:16,ERs,L),@-ERd	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 1 0 0	0 0 0 0	
	OR.B @(d:16,ERs,L),@(d:2,ERd)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 1 0 0	0 0 0 0	
	OR.B @(d:16,ERs,L),@(d:16,ERd)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:16,ERs,L),@(d:32,ERd)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 1 0 0	0 0 0 0	d d
	OR.B @(d:16,ERs,L),@(d:16,Rd,B)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:16,ERs,L),@(d:16,Rd,W)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:16,ERs,L),@(d:16,ERdL)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:16,ERs,L),@(d:32,Rd,B)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.B @(d:16,ERs,L),@(d:32,Rd,W)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 1 0 0	0 0 0 0	d d
	OR.B @(d:16,ERs,L),@(d:32,ERdL)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.B @(d:16,ERs,L),@aa:16	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 1 0 0	0 0 0 0	a	
OR.B @(d:16,ERs,L),@aa:32	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 0 0	0 0 0 0	a a		
OR.B @(d:32,Rs,B),@ERd	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 1 0 0	0 0 0 0		
OR.B @(d:32,Rs,B),@ERd+	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 1 0 0	0 0 0 0		
OR.B @(d:32,Rs,B),@ERd-	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 1 0 0	0 0 0 0		

Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension		
		15	8	7	0	15	8		7	0	15	8	7	0			
OR	OR.B @(d:32,Rs.B),@+ERd	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 1 0 0	0 0 0 0	
	OR.B @(d:32,Rs.B),@-ERd	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 1 0 0	0 0 0 0	
	OR.B @(d:32,Rs.B),@(d:2,ERd)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 1 0 0	0 0 0 0	
	OR.B @(d:32,Rs.B),@(d:16,ERd)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:32,Rs.B),@(d:32,ERd)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 1 0 0	0 0 0 0	d d
	OR.B @(d:32,Rs.B),@(d:16,Rd.B)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:32,Rs.B),@(d:16,Rd.W)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:32,Rs.B),@(d:16,ERd.L)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:32,Rs.B),@(d:32,Rd.B)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.B @(d:32,Rs.B),@(d:32,Rd.W)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 1 0 0	0 0 0 0	d d
	OR.B @(d:32,Rs.B),@(d:32,ERd.L)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.B @(d:32,Rs.B),@aa:16	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 1 0 0	0 0 0 0	a
	OR.B @(d:32,Rs.B),@aa:32	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 1 0 0	0 0 0 0	a a
	OR.B @(d:32,Rs.W),@ERd	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	OR.B @(d:32,Rs.W),@ERd+	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	OR.B @(d:32,Rs.W),@ERd-	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 1 0 0	0 0 0 0	
	OR.B @(d:32,Rs.W),@+ERd	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 1 0 0	0 0 0 0	
	OR.B @(d:32,Rs.W),@-ERd	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 1 0 0	0 0 0 0	
	OR.B @(d:32,Rs.W),@(d:2,ERd)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 1 0 0	0 0 0 0	
	OR.B @(d:32,Rs.W),@(d:16,ERd)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:32,Rs.W),@(d:32,ERd)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 1 0 0	0 0 0 0	d d
	OR.B @(d:32,Rs.W),@(d:16,Rd.B)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:32,Rs.W),@(d:16,Rd.W)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:32,Rs.W),@(d:16,ERd.L)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d
	OR.B @(d:32,Rs.W),@(d:32,Rd.B)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.B @(d:32,Rs.W),@(d:32,Rd.W)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 1 0 0	0 0 0 0	d d
	OR.B @(d:32,Rs.W),@(d:32,ERd.L)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.B @(d:32,Rs.W),@aa:16	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 1 0 0	0 0 0 0	a
	OR.B @(d:32,Rs.W),@aa:32	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 1 0 0	0 0 0 0	a a
	OR.B @(d:32,ERs.L),@ERd	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	OR.B @(d:32,ERs.L),@ERd+	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	OR.B @(d:32,ERs.L),@ERd-	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 1 0 0	0 0 0 0	
OR.B @(d:32,ERs.L),@+ERd	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 1 0 0	0 0 0 0		
OR.B @(d:32,ERs.L),@-ERd	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 1 0 0	0 0 0 0		



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension																					
		15	8	7	0	15	8	7	0		15	8	7	0																						
OR	OR.B @(d:32.ERs.L),@(d:2,ERd)	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	d	d	0	0	0	0	0	0	0	0	0	0	0				
	OR.B @(d:32.ERs.L),@(d:16,ERd)	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	d	d	1	1	0	0	0	0	rd	0	1	0	0	0	0	d	
	OR.B @(d:32.ERs.L),@(d:32,ERd)	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	d	d	1	1	0	0	1	rd	0	1	0	0	0	0	0	d	d
	OR.B @(d:32.ERs.L),@(d:16,Rd.B)	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	d	d	1	1	0	1	0	rd	0	1	0	0	0	0	0	d	
	OR.B @(d:32.ERs.L),@(d:16,Rd.W)	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	d	d	1	1	1	0	0	rd	0	1	0	0	0	0	0	d	
	OR.B @(d:32.ERs.L),@(d:16,ERd.L)	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	d	d	1	1	1	1	0	rd	0	1	0	0	0	0	0	d	
	OR.B @(d:32.ERs.L),@(d:32,Rd.B)	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	d	d	1	1	0	1	1	rd	0	1	0	0	0	0	0	d	d
	OR.B @(d:32.ERs.L),@(d:32,Rd.W)	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	d	d	1	1	1	0	1	rd	0	1	0	0	0	0	0	d	d
	OR.B @(d:32.ERs.L),@(d:32,ERd.L)	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	d	d	1	1	1	1	1	rd	0	1	0	0	0	0	0	d	d
	OR.B @(d:32.ERs.L),@aa:16	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	d	d	0	1	0	0	0	0	rd	0	1	0	0	0	0	a	
	OR.B @(d:32.ERs.L),@aa:32	7	8	0	rs	0	1	1	1	6	A	0	0	1	0	1	1	0	0	d	d	0	1	0	0	0	0	rd	0	1	0	0	0	0	a	a
	OR.B @aa:16,@ERd									6	A	0	0	0	1	0	1	0	1	a			0	0	0	0	0	rd	0	1	0	0	0	0		
	OR.B @aa:16,@ERd+									6	A	0	0	0	1	0	1	0	1	a			1	0	0	0	0	rd	0	1	0	0	0	0		
	OR.B @aa:16,@ERd-									6	A	0	0	0	1	0	1	0	1	a			1	0	1	0	0	rd	0	1	0	0	0	0		
	OR.B @aa:16,@+ERd									6	A	0	0	0	1	0	1	0	1	a			1	0	0	1	0	rd	0	1	0	0	0	0		
	OR.B @aa:16,@-ERd									6	A	0	0	0	1	0	1	0	1	a			1	0	1	1	0	rd	0	1	0	0	0	0		
	OR.B @aa:16,@(d:2,ERd)									6	A	0	0	0	1	0	1	0	1	a			0	0	0	d	0	rd	0	1	0	0	0	0		
	OR.B @aa:16,@(d:16,ERd)									6	A	0	0	0	1	0	1	0	1	a			1	1	0	0	0	rd	0	1	0	0	0	0	d	
	OR.B @aa:16,@(d:32,ERd)									6	A	0	0	0	1	0	1	0	1	a			1	1	0	0	1	rd	0	1	0	0	0	0	d	d
	OR.B @aa:16,@(d:16,Rd.B)									6	A	0	0	0	1	0	1	0	1	a			1	1	0	1	0	rd	0	1	0	0	0	0	d	
	OR.B @aa:16,@(d:16,Rd.W)									6	A	0	0	0	1	0	1	0	1	a			1	1	1	0	0	rd	0	1	0	0	0	0	d	
	OR.B @aa:16,@(d:16,ERd.L)									6	A	0	0	0	1	0	1	0	1	a			1	1	1	1	0	rd	0	1	0	0	0	0	d	
	OR.B @aa:16,@(d:32,Rd.B)									6	A	0	0	0	1	0	1	0	1	a			1	1	0	1	1	rd	0	1	0	0	0	0	d	d
	OR.B @aa:16,@(d:32,Rd.W)									6	A	0	0	0	1	0	1	0	1	a			1	1	1	0	1	rd	0	1	0	0	0	0	d	d
	OR.B @aa:16,@(d:32,ERd.L)									6	A	0	0	0	1	0	1	0	1	a			1	1	1	1	1	rd	0	1	0	0	0	0	d	d
	OR.B @aa:16,@aa:16									6	A	0	0	0	1	0	1	0	1	a			0	1	0	0	0	rd	0	1	0	0	0	0	a	
	OR.B @aa:16,@aa:32									6	A	0	0	0	1	0	1	0	1	a			0	1	0	0	0	rd	0	1	0	0	0	0	a	a
	OR.B @aa:32,@ERd									6	A	0	0	1	1	0	1	0	1	a			0	0	0	0	0	rd	0	1	0	0	0	0		
	OR.B @aa:32,@ERd+									6	A	0	0	1	1	0	1	0	1	a			1	0	0	0	0	rd	0	1	0	0	0	0		
	OR.B @aa:32,@ERd-									6	A	0	0	1	1	0	1	0	1	a			1	0	1	0	0	rd	0	1	0	0	0	0		
	OR.B @aa:32,@+ERd									6	A	0	0	1	1	0	1	0	1	a			1	0	0	1	0	rd	0	1	0	0	0	0		
	OR.B @aa:32,@-ERd									6	A	0	0	1	1	0	1	0	1	a			1	0	1	1	0	rd	0	1	0	0	0	0		
OR.B @aa:32,@(d:2,ERd)									6	A	0	0	1	1	0	1	0	1	a			0	0	d	d	0	rd	0	1	0	0	0	0			
OR.B @aa:32,@(d:16,ERd)									6	A	0	0	1	1	0	1	0	1	a			1	1	0	0	0	rd	0	1	0	0	0	0	d		

Instruction	Mnemonic	Opcode			Opcode			EA Extension	Opcode			EA Extension				
		15	8	7	0	15	8		7	0	15		8	7	0	
OR	OR.B @aa:32,@(d:32,ERd)				6	A	0 0 1 1	0 1 0 1	a a	1 1 0 0	1	rd	0 1 0 0	0 0 0 0	d	d
	OR.B @aa:32,@(d:16,Rd.B)				6	A	0 0 1 1	0 1 0 1	a a	1 1 0 1	0	rd	0 1 0 0	0 0 0 0	d	
	OR.B @aa:32,@(d:16,Rd.W)				6	A	0 0 1 1	0 1 0 1	a a	1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d	
	OR.B @aa:32,@(d:16,ERd.L)				6	A	0 0 1 1	0 1 0 1	a a	1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d	
	OR.B @aa:32,@(d:32,Rd.B)				6	A	0 0 1 1	0 1 0 1	a a	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d	d
	OR.B @aa:32,@(d:32,Rd.W)				6	A	0 0 1 1	0 1 0 1	a a	1 1 1 0	1	rd	0 1 0 0	0 0 0 0	d	d
	OR.B @aa:32,@(d:32,ERd.L)				6	A	0 0 1 1	0 1 0 1	a a	1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d	d
	OR.B @aa:32,@aa:16				6	A	0 0 1 1	0 1 0 1	a a	0 1 0 0	0	0 0 0 0	0 1 0 0	0 0 0 0	a	
	OR.B @aa:32,@aa:32				6	A	0 0 1 1	0 1 0 1	a a	0 1 0 0	1	0 0 0 0	0 1 0 0	0 0 0 0	a	a
	OR.W #xx:16,Rd										7		g	0 1 0 0	rd	x
	OR.W #xx:16,@ERd				0	1	5	1 1 1 0		0 0 0 0	0	rd	0 1 0 0	0 0 0 0		x
	OR.W #xx:16,@ERd+				0	1	5	1 1 1 0		1 0 0 0	0	rd	0 1 0 0	0 0 0 0		x
	OR.W #xx:16,@ERd-				0	1	5	1 1 1 0		1 0 1 0	0	rd	0 1 0 0	0 0 0 0		x
	OR.W #xx:16,@+ERd				0	1	5	1 1 1 0		1 0 0 1	0	rd	0 1 0 0	0 0 0 0		x
	OR.W #xx:16,@-ERd				0	1	5	1 1 1 0		1 0 1 1	0	rd	0 1 0 0	0 0 0 0		x
	OR.W #xx:16,@(d:2,ERd)				0	1	5	1 1 1 0		0 0 d d	0	rd	0 1 0 0	0 0 0 0		x
	OR.W #xx:16,@(d:16,ERd)				0	1	5	1 1 1 0		1 1 0 0	0	rd	0 1 0 0	0 0 0 0	d	x
	OR.W #xx:16,@(d:32,ERd)				0	1	5	1 1 1 0		1 1 0 0	1	rd	0 1 0 0	0 0 0 0	d	d
	OR.W #xx:16,@(d:16,Rd.B)				0	1	5	1 1 1 0		1 1 0 1	0	rd	0 1 0 0	0 0 0 0	d	x
	OR.W #xx:16,@(d:16,Rd.W)				0	1	5	1 1 1 0		1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d	x
	OR.W #xx:16,@(d:16,ERd.L)				0	1	5	1 1 1 0		1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d	x
	OR.W #xx:16,@(d:32,Rd.B)				0	1	5	1 1 1 0		1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d	d
	OR.W #xx:16,@(d:32,Rd.W)				0	1	5	1 1 1 0		1 1 1 0	1	rd	0 1 0 0	0 0 0 0	d	d
	OR.W #xx:16,@(d:32,ERd.L)				0	1	5	1 1 1 0		1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d	d
	OR.W #xx:16,@aa:16				0	1	5	1 1 1 0		0 1 0 0	0	0 0 0 0	0 1 0 0	0 0 0 0	a	x
	OR.W #xx:16,@aa:32				0	1	5	1 1 1 0		0 1 0 0	1	0 0 0 0	0 1 0 0	0 0 0 0	a	a
	OR.W Rs,Rd										6		4	rs	rd	
	OR.W Rs,@ERd				7	D	1	rd	0 0 0 0		6		4	rs	0 0 0 0	
	OR.W Rs,@ERd+				0	1	5	1 0 0 1		1 0 0 0	0	rd	0 1 0 0	rs		
	OR.W Rs,@ERd-				0	1	5	1 0 0 1		1 0 1 0	0	rd	0 1 0 0	rs		
OR.W Rs,@+ERd				0	1	5	1 0 0 1		1 0 0 1	0	rd	0 1 0 0	rs			
OR.W Rs,@-ERd				0	1	5	1 0 0 1		1 0 1 1	0	rd	0 1 0 0	rs			
OR.W Rs,@(d:2,ERd)				0	1	5	1 0 0 1		0 0 d d	0	rd	0 1 0 0	rs			
OR.W Rs,@(d:16,ERd)				0	1	5	1 0 0 1		1 1 0 0	0	rd	0 1 0 0	rs	d		



Instruction	Mnemonic	Opcode			Opcode			EA Extension	Opcode			EA Extension			
		15	8	7	0	15	8		7	0	15		8	7	0
OR	OR.W Rs.@(d:32,ERd)				0	1	5	1 0 0 1		1	1 0 0 1	rd	0 1 0 0	rs	d d
	OR.W Rs.@(d:16,Rd.B)				0	1	5	1 0 0 1		1	1 0 1 0	rd	0 1 0 0	rs	d
	OR.W Rs.@(d:16,Rd.W)				0	1	5	1 0 0 1		1	1 1 0 0	rd	0 1 0 0	rs	d
	OR.W Rs.@(d:16,ERd.L)				0	1	5	1 0 0 1		1	1 1 1 0	rd	0 1 0 0	rs	d
	OR.W Rs.@(d:32,Rd.B)				0	1	5	1 0 0 1		1	1 0 1 1	rd	0 1 0 0	rs	d d
	OR.W Rs.@(d:32,Rd.W)				0	1	5	1 0 0 1		1	1 1 0 1	rd	0 1 0 0	rs	d d
	OR.W Rs.@(d:32,ERd.L)				0	1	5	1 0 0 1		1	1 1 1 1	rd	0 1 0 0	rs	d d
	OR.W Rs.@aa:16				6	B	0 0 0 1	1 0 0 0	a	6	4	rs	0 0 0 0		
	OR.W Rs.@aa:32				6	B	0 0 1 1	1 0 0 0	a a	6	4	rs	0 0 0 0		
	OR.W @ERs,Rd				7	C	1 rs	0 0 0 0		6	4		0 0 0 0	rd	
	OR.W @ERs+,Rd				0	1	5	1 0 1 0		1	0 0 0 0	rs	0 1 0 0	rd	
	OR.W @ERs-,Rd				0	1	5	1 0 1 0		1	0 1 0 0	rs	0 1 0 0	rd	
	OR.W @+ERs,Rd				0	1	5	1 0 1 0		1	0 0 1 0	rs	0 1 0 0	rd	
	OR.W @-ERs,Rd				0	1	5	1 0 1 0		1	0 1 1 0	rs	0 1 0 0	rd	
	OR.W @(d:2,ERs),Rd				0	1	5	1 0 1 0		0	0 d d 0	rs	0 1 0 0	rd	
	OR.W @(d:16,ERs),Rd				0	1	5	1 0 1 0		1	1 0 0 0	rs	0 1 0 0	rd	d
	OR.W @(d:32,ERs),Rd				0	1	5	1 0 1 0		1	1 0 0 1	rs	0 1 0 0	rd	d d
	OR.W @(d:16,Rs.B),Rd				0	1	5	1 0 1 0		1	1 0 1 0	rs	0 1 0 0	rd	d
	OR.W @(d:16,Rs.W),Rd				0	1	5	1 0 1 0		1	1 1 0 0	rs	0 1 0 0	rd	d
	OR.W @(d:16,ERs.L),Rd				0	1	5	1 0 1 0		1	1 1 1 0	rs	0 1 0 0	rd	d
	OR.W @(d:32,Rs.B),Rd				0	1	5	1 0 1 0		1	1 0 1 1	rs	0 1 0 0	rd	d d
	OR.W @(d:32,Rs.W),Rd				0	1	5	1 0 1 0		1	1 1 0 1	rs	0 1 0 0	rd	d d
	OR.W @(d:32,ERs.L),Rd				0	1	5	1 0 1 0		1	1 1 1 1	rs	0 1 0 0	rd	d d
	OR.W @aa:16,Rd				6	B	0 0 0 1	0 0 0 0	a	6	4		0 0 0 0	rd	
	OR.W @aa:32,Rd				6	B	0 0 1 1	0 0 0 0	a a	6	4		0 0 0 0	rd	
	OR.W @ERs,@ERd				7	C	1 rs	0 1 0 1		0	0 0 0 0	rd	0 1 0 0	0 0 0 0	
	OR.W @ERs,@ERd+				7	C	1 rs	0 1 0 1		1	0 0 0 0	rd	0 1 0 0	0 0 0 0	
	OR.W @ERs,@ERd-				7	C	1 rs	0 1 0 1		1	0 1 0 0	rd	0 1 0 0	0 0 0 0	
	OR.W @ERs,@+ERd				7	C	1 rs	0 1 0 1		1	0 0 1 0	rd	0 1 0 0	0 0 0 0	
	OR.W @ERs,@-ERd				7	C	1 rs	0 1 0 1		1	0 1 1 0	rd	0 1 0 0	0 0 0 0	
OR.W @ERs,@(d:2,ERd)				7	C	1 rs	0 1 0 1		0	0 d d 0	rd	0 1 0 0	0 0 0 0		
OR.W @ERs,@(d:16,ERd)				7	C	1 rs	0 1 0 1		1	1 0 0 0	rd	0 1 0 0	0 0 0 0	d	
OR.W @ERs,@(d:32,ERd)				7	C	1 rs	0 1 0 1		1	1 0 0 1	rd	0 1 0 0	0 0 0 0	d d	
OR.W @ERs,@(d:16,Rd.B)				7	C	1 rs	0 1 0 1		1	1 0 1 0	rd	0 1 0 0	0 0 0 0	d	

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
OR	OR.W @ERs, @(d:16,Rd,W)				7	C	1	rs	0 1 0 1		1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d	
	OR.W @ERs, @(d:16,ERd,L)				7	C	1	rs	0 1 0 1		1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d	
	OR.W @ERs, @(d:32,Rd,B)				7	C	1	rs	0 1 0 1		1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d d	
	OR.W @ERs, @(d:32,Rd,W)				7	C	1	rs	0 1 0 1		1 1 1 0	1	rd	0 1 0 0	0 0 0 0	d d	
	OR.W @ERs, @(d:32,ERd,L)				7	C	1	rs	0 1 0 1		1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d d	
	OR.W @ERs, @aa:16				7	C	1	rs	0 1 0 1		0 1 0 0	0 0 0 0	0 1 0 0	0 0 0 0	a		
	OR.W @ERs, @aa:32				7	C	1	rs	0 1 0 1		0 1 0 0	1 0 0 0	0 1 0 0	0 0 0 0	a a		
	OR.W @ERs+, @ERd	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0		0 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	OR.W @ERs+, @ERd+	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0		1 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	OR.W @ERs+, @ERd-	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0		1 0 1 0	0	rd	0 1 0 0	0 0 0 0	
	OR.W @ERs+, @+ERd	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0		1 0 0 1	0	rd	0 1 0 0	0 0 0 0	
	OR.W @ERs+, @-ERd	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0		1 0 1 1	0	rd	0 1 0 0	0 0 0 0	
	OR.W @ERs+, @(d2,ERd)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0		0 0 d d	0	rd	0 1 0 0	0 0 0 0	
	OR.W @ERs+, @(d:16,ERd)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0		1 1 0 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.W @ERs+, @(d:32,ERd)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0		1 1 0 0	1	rd	0 1 0 0	0 0 0 0	d d
	OR.W @ERs+, @(d:16,Rd,B)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0		1 1 0 1	0	rd	0 1 0 0	0 0 0 0	d
	OR.W @ERs+, @(d:16,Rd,W)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0		1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.W @ERs+, @(d:16,ERd,L)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0		1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d
	OR.W @ERs+, @(d:32,Rd,B)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0		1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.W @ERs+, @(d:32,Rd,W)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0		1 1 1 0	1	rd	0 1 0 0	0 0 0 0	d d
	OR.W @ERs+, @(d:32,ERd,L)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0		1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.W @ERs+, @aa:16	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0		0 1 0 0	0 0 0 0	0 1 0 0	0 0 0 0	a	
	OR.W @ERs+, @aa:32	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0		0 1 0 0	1 0 0 0	0 1 0 0	0 0 0 0	a a	
	OR.W @ERs-, @ERd	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		0 0 0 0	0 0 0 0	0 1 0 0	0 0 0 0		
	OR.W @ERs-, @ERd+	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	OR.W @ERs-, @ERd-	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 0 1 0	0	rd	0 1 0 0	0 0 0 0	
	OR.W @ERs-, @+ERd	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 0 0 1	0	rd	0 1 0 0	0 0 0 0	
	OR.W @ERs-, @-ERd	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 0 1 1	0	rd	0 1 0 0	0 0 0 0	
	OR.W @ERs-, @(d2,ERd)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		0 0 d d	0	rd	0 1 0 0	0 0 0 0	
	OR.W @ERs-, @(d:16,ERd)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 0 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.W @ERs-, @(d:32,ERd)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 0 0	1	rd	0 1 0 0	0 0 0 0	d d
	OR.W @ERs-, @(d:16,Rd,B)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 0 1	0	rd	0 1 0 0	0 0 0 0	d
OR.W @ERs-, @(d:16,Rd,W)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d	
OR.W @ERs-, @(d:16,ERd,L)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d	



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension			
		15	8	7	0	15	8	7	0		15	8	7	0				
OR	OR.W @ERs, @(d:32,Rd.B)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0	1	1 0 0 1	1	rd	0 1 0 0	0 0 0 0	0	d d
	OR.W @ERs, @(d:32,Rd.W)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0	1	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	0	d d
	OR.W @ERs, @(d:32,ERd.L)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0	1	1 1 1 1	1	rd	0 1 0 0	0 0 0 0	0	d d
	OR.W @ERs, @aa:16	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0	0	1 0 0 0	0	rd	0 1 0 0	0 0 0 0	0	a
	OR.W @ERs, @aa:32	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0	0	1 0 0 0	1	0 0 0 0	0	1 0 0 0	0 0 0 0	a a
	OR.W @+ERs, @ERd	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0	0	0 0 0 0	0	rd	0 1 0 0	0 0 0 0	0	
	OR.W @+ERs, @ERd+	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0	1	0 0 0 0	0	rd	0 1 0 0	0 0 0 0	0	
	OR.W @+ERs, @ERd-	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0	1	0 1 0 0	0	rd	0 1 0 0	0 0 0 0	0	
	OR.W @+ERs, @+ERd	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0	1	0 0 0 1	0	rd	0 1 0 0	0 0 0 0	0	
	OR.W @+ERs, @-ERd	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0	1	0 1 1 0	0	rd	0 1 0 0	0 0 0 0	0	
	OR.W @+ERs, @(d:2,ERd)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0	0	0 d d d	0	rd	0 1 0 0	0 0 0 0	0	
	OR.W @+ERs, @(d:16,ERd)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0	1	1 0 0 0	0	rd	0 1 0 0	0 0 0 0	0	d
	OR.W @+ERs, @(d:32,ERd)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0	1	1 0 0 1	1	rd	0 1 0 0	0 0 0 0	0	d d
	OR.W @+ERs, @(d:16,Rd.B)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0	1	1 0 1 0	0	rd	0 1 0 0	0 0 0 0	0	d
	OR.W @+ERs, @(d:16,Rd.W)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0	1	1 1 0 0	0	rd	0 1 0 0	0 0 0 0	0	d
	OR.W @+ERs, @(d:16,ERd.L)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0	1	1 1 1 0	0	rd	0 1 0 0	0 0 0 0	0	d
	OR.W @+ERs, @(d:32,Rd.B)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0	1	1 0 1 1	1	rd	0 1 0 0	0 0 0 0	0	d d
	OR.W @+ERs, @(d:32,Rd.W)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0	1	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	0	d d
	OR.W @+ERs, @(d:32,ERd.L)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0	1	1 1 1 1	1	rd	0 1 0 0	0 0 0 0	0	d d
	OR.W @+ERs, @aa:16	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0	0	1 0 0 0	0	0 0 0 0	0	1 0 0 0	0 0 0 0	a
	OR.W @+ERs, @aa:32	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0	0	1 0 0 0	1	0 0 0 0	0	1 0 0 0	0 0 0 0	a a
	OR.W @-ERs, @ERd	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0	0	0 0 0 0	0	rd	0 1 0 0	0 0 0 0	0	
	OR.W @-ERs, @ERd+	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0	1	0 0 0 0	0	rd	0 1 0 0	0 0 0 0	0	
	OR.W @-ERs, @ERd-	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0	1	0 1 0 0	0	rd	0 1 0 0	0 0 0 0	0	
	OR.W @-ERs, @+ERd	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0	1	0 0 0 1	0	rd	0 1 0 0	0 0 0 0	0	
	OR.W @-ERs, @-ERd	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0	1	0 1 1 0	0	rd	0 1 0 0	0 0 0 0	0	
	OR.W @-ERs, @(d:2,ERd)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0	0	0 d d d	0	rd	0 1 0 0	0 0 0 0	0	
	OR.W @-ERs, @(d:16,ERd)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0	1	1 0 0 0	0	rd	0 1 0 0	0 0 0 0	0	d
	OR.W @-ERs, @(d:32,ERd)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0	1	1 0 0 1	1	rd	0 1 0 0	0 0 0 0	0	d d
	OR.W @-ERs, @(d:16,Rd.B)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0	1	1 0 1 0	0	rd	0 1 0 0	0 0 0 0	0	d
OR.W @-ERs, @(d:16,Rd.W)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0	1	1 1 0 0	0	rd	0 1 0 0	0 0 0 0	0	d	
OR.W @-ERs, @(d:16,ERd.L)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0	1	1 1 1 0	0	rd	0 1 0 0	0 0 0 0	0	d	
OR.W @-ERs, @(d:32,Rd.B)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0	1	1 0 1 1	1	rd	0 1 0 0	0 0 0 0	0	d d	
OR.W @-ERs, @(d:32,Rd.W)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0	1	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	0	d d	

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension			
		15	8	7	0	15	8	7	0		15	8	7	0				
OR	OR.W @.ERs, @(d:32,ERd.L)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 1	1	rd	0 1 0 0	0 0 0 0	0 0 0 0	d d
	OR.W @.ERs, @aa:16	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		0 1 0 0	0	0 0 0 0	0 1 0 0	0 0 0 0	0 0 0 0	a
	OR.W @.ERs, @aa:32	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		0 1 0 0	1	0 0 0 0	0 1 0 0	0 0 0 0	0 0 0 0	a a
	OR.W @(d:2,ERs), @ERd	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		0 0 0 0	0	rd	0 1 0 0	0 0 0 0	0 0 0 0	
	OR.W @(rd:2,ERs), @ERd+	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 0 0 0	0	rd	0 1 0 0	0 0 0 0	0 0 0 0	
	OR.W @(d:2,ERs), @ERd-	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 0 1 0	0	rd	0 1 0 0	0 0 0 0	0 0 0 0	
	OR.W @(d:2,ERs), @+ERd	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 0 0 1	0	rd	0 1 0 0	0 0 0 0	0 0 0 0	
	OR.W @(rd:2,ERs), @-ERd	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 0 1 1	0	rd	0 1 0 0	0 0 0 0	0 0 0 0	
	OR.W @(d:2,ERs), @(d:2,ERd)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		0 0 d d	0	rd	0 1 0 0	0 0 0 0	0 0 0 0	
	OR.W @(d:2,ERs), @(d:16,ERd)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 0	0	rd	0 1 0 0	0 0 0 0	0 0 0 0	d
	OR.W @(rd:2,ERs), @(d:32,ERd)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 0	1	rd	0 1 0 0	0 0 0 0	0 0 0 0	d d
	OR.W @(d:2,ERs), @(d:16,Rd.B)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 1	0	rd	0 1 0 0	0 0 0 0	0 0 0 0	d
	OR.W @(d:2,ERs), @(d:16,Rd.W)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 1 0	0	rd	0 1 0 0	0 0 0 0	0 0 0 0	d
	OR.W @(d:2,ERs), @(d:16,ERd.L)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 1 1	0	rd	0 1 0 0	0 0 0 0	0 0 0 0	d
	OR.W @(d:2,ERs), @(d:32,Rd.B)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 1	1	rd	0 1 0 0	0 0 0 0	0 0 0 0	d d
	OR.W @(d:2,ERs), @(d:32,Rd.W)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 1 0	1	rd	0 1 0 0	0 0 0 0	0 0 0 0	d d
	OR.W @(d:2,ERs), @(d:32,ERd.L)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 1 1	1	rd	0 1 0 0	0 0 0 0	0 0 0 0	d d
	OR.W @(d:2,ERs), @aa:16	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		0 1 0 0	0	0 0 0 0	0 1 0 0	0 0 0 0	0 0 0 0	a
	OR.W @(d:2,ERs), @aa:32	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		0 1 0 0	1	0 0 0 0	0 1 0 0	0 0 0 0	0 0 0 0	a a
	OR.W @(d:16,ERs), @ERd	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 1 0 0	0 0 0 0	0 0 0 0	
	OR.W @(d:16,ERs), @ERd+	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 1 0 0	0 0 0 0	0 0 0 0	
	OR.W @(d:16,ERs), @ERd-	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 0 0	0 0 0 0	0 0 0 0	
	OR.W @(d:16,ERs), @+ERd	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 0 0	0 0 0 0	0 0 0 0	
	OR.W @(d:16,ERs), @-ERd	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 1 0 0	0 0 0 0	0 0 0 0	
	OR.W @(d:16,ERs), @(d:2,ERd)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 1 0 0	0 0 0 0	0 0 0 0	
	OR.W @(d:16,ERs), @(d:16,ERd)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 1 0 0	0 0 0 0	0 0 0 0	d
	OR.W @(d:16,ERs), @(d:32,ERd)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 1 0 0	0 0 0 0	0 0 0 0	d d
	OR.W @(d:16,ERs), @(d:16,Rd.B)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 1 0 0	0 0 0 0	0 0 0 0	d
	OR.W @(d:16,ERs), @(d:16,Rd.W)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 1 0 0	0 0 0 0	0 0 0 0	d
	OR.W @(d:16,ERs), @(d:16,ERd.L)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 1 0 0	0 0 0 0	0 0 0 0	d
OR.W @(d:16,ERs), @(d:32,Rd.B)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	0 0 0 0	d d	
OR.W @(d:16,ERs), @(d:32,Rd.W)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 1 0 0	0 0 0 0	0 0 0 0	d d	
OR.W @(d:16,ERs), @(d:32,ERd.L)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 1 0 0	0 0 0 0	0 0 0 0	d d	
OR.W @(d:16,ERs), @aa:16	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	0	0 0 0 0	0 1 0 0	0 0 0 0	0 0 0 0	a	



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
OR	OR.W @(d:16,ERs),@aa:32	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 0 0	0 0 0 0	a a
	OR.W @(d:32,ERs),@ERd	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 1 0 0	0 0 0 0
	OR.W @(d:32,ERs),@ERd+	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 1 0 0	0 0 0 0
	OR.W @(d:32,ERs),@ERd-	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 1 0 0	0 0 0 0
	OR.W @(d:32,ERs),@+ERd	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 1 0 0	0 0 0 0
	OR.W @(d:32,ERs),@-ERd	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 1 0 0	0 0 0 0
	OR.W @(d:32,ERs),@(d:2,ERd)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 1 0 0	0 0 0 0
	OR.W @(d:32,ERs),@(d:16,ERd)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 1 0 0	0 0 0 0
	OR.W @(d:32,ERs),@(d:32,ERd)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 1 0 0	0 0 0 0
	OR.W @(d:32,ERs),@(d:16,Rd,B)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 1 0 0	0 0 0 0
	OR.W @(d:32,ERs),@(d:16,Rd,W)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 1 0 0	0 0 0 0
	OR.W @(d:32,ERs),@(d:16,ERd,L)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 1 0 0	0 0 0 0
	OR.W @(d:32,ERs),@(d:32,Rd,B)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 1 0 0	0 0 0 0
	OR.W @(d:32,ERs),@(d:32,Rd,W)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 1 0 0	0 0 0 0
	OR.W @(d:32,ERs),@(d:32,ERd,L)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 1 0 0	0 0 0 0
	OR.W @(d:32,ERs),@aa:16	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0 0 0 0	0 1 0 0	0 0 0 0	a
	OR.W @(d:32,ERs),@aa:32	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1 0 0 0	0 1 0 0	0 0 0 0	a a
	OR.W @(d:16,Rs,B),@ERd	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 1 0 0	0 0 0 0
	OR.W @(d:16,Rs,B),@ERd+	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 1 0 0	0 0 0 0
	OR.W @(d:16,Rs,B),@ERd-	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 0 0	0 0 0 0
	OR.W @(d:16,Rs,B),@+ERd	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 0 0	0 0 0 0
	OR.W @(d:16,Rs,B),@-ERd	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 1 0 0	0 0 0 0
	OR.W @(d:16,Rs,B),@(d:2,ERd)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 1 0 0	0 0 0 0
	OR.W @(d:16,Rs,B),@(d:16,ERd)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 1 0 0	0 0 0 0
	OR.W @(d:16,Rs,B),@(d:32,ERd)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 1 0 0	0 0 0 0
	OR.W @(d:16,Rs,B),@(d:16,Rd,B)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 1 0 0	0 0 0 0
	OR.W @(d:16,Rs,B),@(d:16,Rd,W)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 1 0 0	0 0 0 0
	OR.W @(d:16,Rs,B),@(d:16,ERd,L)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 1 0 0	0 0 0 0
	OR.W @(d:16,Rs,B),@(d:32,Rd,B)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 1 0 0	0 0 0 0
	OR.W @(d:16,Rs,B),@(d:32,Rd,W)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 1 0 0	0 0 0 0
	OR.W @(d:16,Rs,B),@(d:32,ERd,L)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 1 0 0	0 0 0 0
	OR.W @(d:16,Rs,B),@aa:16	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 1 0 0	0 0 0 0	a
OR.W @(d:16,Rs,B),@aa:32	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 0 0	0 0 0 0	a a	
OR.W @(d:16,Rs,W),@ERd	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 1 0 0	0 0 0 0	

Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension			
		15	8	7	0	15	8	7	0		15	8	7	0				
OR	OR.W @(d:16,Rs.W),@ERd+	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1	0 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	OR.W @(d:16,Rs.W),@ERd-	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1	0 1 0 0	0	rd	0 1 0 0	0 0 0 0	
	OR.W @(d:16,Rs.W),@+ERd	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1	0 0 1 0	0	rd	0 1 0 0	0 0 0 0	
	OR.W @(d:16,Rs.W),@-ERd	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1	0 1 1 0	0	rd	0 1 0 0	0 0 0 0	
	OR.W @(d:16,Rs.W),@(d:2,ERd)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0	0 d d 0	0	rd	0 1 0 0	0 0 0 0	
	OR.W @(d:16,Rs.W),@(d:16,ERd)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1	1 0 0 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.W @(d:16,Rs.W),@(d:32,ERd)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1	1 0 0 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.W @(d:16,Rs.W),@(d:16,Rd.B)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1	1 0 1 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.W @(d:16,Rs.W),@(d:16,Rd.W)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1	1 1 0 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.W @(d:16,Rs.W),@(d:16,ERd.L)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1	1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.W @(d:16,Rs.W),@(d:32,Rd.B)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1	1 0 1 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.W @(d:16,Rs.W),@(d:32,Rd.W)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.W @(d:16,Rs.W),@(d:32,ERd.L)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1	1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.W @(d:16,Rs.W),@aa:16	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0	1 0 0 0	0 0 0 0	0 1 0 0	0 0 0 0	a	
	OR.W @(d:16,Rs.W),@aa:32	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0	1 0 0 1	1 0 0 0	0 1 0 0	0 0 0 0	a a	
	OR.W @(d:16,ERs.L),@ERd	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0	0 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	OR.W @(d:16,ERs.L),@ERd+	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1	0 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	OR.W @(d:16,ERs.L),@ERd-	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1	0 1 0 0	0	rd	0 1 0 0	0 0 0 0	
	OR.W @(d:16,ERs.L),@+ERd	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1	0 0 1 0	0	rd	0 1 0 0	0 0 0 0	
	OR.W @(d:16,ERs.L),@-ERd	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1	0 1 1 0	0	rd	0 1 0 0	0 0 0 0	
	OR.W @(d:16,ERs.L),@(d:2,ERd)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0	0 d d 0	0	rd	0 1 0 0	0 0 0 0	
	OR.W @(d:16,ERs.L),@(d:16,ERd)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1	1 0 0 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.W @(d:16,ERs.L),@(d:32,ERd)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1	1 0 0 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.W @(d:16,ERs.L),@(d:16,Rd.B)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1	1 0 1 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.W @(d:16,ERs.L),@(d:16,Rd.W)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1	1 1 0 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.W @(d:16,ERs.L),@(d:16,ERd.L)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1	1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.W @(d:16,ERs.L),@(d:32,Rd.B)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1	1 0 1 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.W @(d:16,ERs.L),@(d:32,Rd.W)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.W @(d:16,ERs.L),@(d:32,ERd.L)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1	1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.W @(d:16,ERs.L),@aa:16	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0	1 0 0 0	0 0 0 0	0 1 0 0	0 0 0 0	a	
	OR.W @(d:16,ERs.L),@aa:32	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0	1 0 0 1	1 0 0 0	0 1 0 0	0 0 0 0	a a	
	OR.W @(d:32,Rs.B),@ERd	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0	0 0 0 0	0	rd	0 1 0 0	0 0 0 0	
OR.W @(d:32,Rs.B),@ERd+	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1	0 0 0 0	0	rd	0 1 0 0	0 0 0 0		
OR.W @(d:32,Rs.B),@ERd-	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1	0 1 0 0	0	rd	0 1 0 0	0 0 0 0		



Instruction	Mnemonic	Opcode						EA Ex- tension	Opcode						EA Extension		
		15	8	7	0	15	8		7	0	15	8	7	0			
OR	OR.W @(d:32,Rs.B),@+ERd	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 1 0 0	0 0 0 0	
	OR.W @(d:32,Rs.B),@-ERd	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 1 0 0	0 0 0 0	
	OR.W @(d:32,Rs.B),@(d:2,ERd)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 1 0 0	0 0 0 0	
	OR.W @(d:32,Rs.B),@(d:16,ERd)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.W @(d:32,Rs.B),@(d:32,ERd)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 1 0 0	0 0 0 0	d d
	OR.W @(d:32,Rs.B),@(d:16,Rd.B)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 1 0 0	0 0 0 0	d
	OR.W @(d:32,Rs.B),@(d:16,Rd.W)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.W @(d:32,Rs.B),@(d:16,ERdL)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d
	OR.W @(d:32,Rs.B),@(d:32,Rd.B)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.W @(d:32,Rs.B),@(d:32,Rd.W)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 1 0 0	0 0 0 0	d d
	OR.W @(d:32,Rs.B),@(d:32,ERdL)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.W @(d:32,Rs.B),@aa:16	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 1 0 0	0 0 0 0	a
	OR.W @(d:32,Rs.B),@aa:32	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 1 0 0	0 0 0 0	a a
	OR.W @(d:32,Rs.W),@ERd	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	OR.W @(d:32,Rs.W),@ERd+	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	OR.W @(d:32,Rs.W),@ERd-	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 1 0 0	0 0 0 0	
	OR.W @(d:32,Rs.W),@+ERd	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 1 0 0	0 0 0 0	
	OR.W @(d:32,Rs.W),@-ERd	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 1 0 0	0 0 0 0	
	OR.W @(d:32,Rs.W),@(d:2,ERd)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 1 0 0	0 0 0 0	
	OR.W @(d:32,Rs.W),@(d:16,ERd)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.W @(d:32,Rs.W),@(d:32,ERd)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 1 0 0	0 0 0 0	d d
	OR.W @(d:32,Rs.W),@(d:16,Rd.B)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 1 0 0	0 0 0 0	d
	OR.W @(d:32,Rs.W),@(d:16,Rd.W)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d
	OR.W @(d:32,Rs.W),@(d:16,ERdL)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d
	OR.W @(d:32,Rs.W),@(d:32,Rd.B)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.W @(d:32,Rs.W),@(d:32,Rd.W)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 1 0 0	0 0 0 0	d d
	OR.W @(d:32,Rs.W),@(d:32,ERdL)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d d
	OR.W @(d:32,Rs.W),@aa:16	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 1 0 0	0 0 0 0	a
OR.W @(d:32,Rs.W),@aa:32	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 1 0 0	0 0 0 0	a a	
OR.W @(d:32,ERs.L),@ERd	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 1 0 0	0 0 0 0		
OR.W @(d:32,ERs.L),@ERd+	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 1 0 0	0 0 0 0		
OR.W @(d:32,ERs.L),@ERd-	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 1 0 0	0 0 0 0		
OR.W @(d:32,ERs.L),@+ERd	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 1 0 0	0 0 0 0		
OR.W @(d:32,ERs.L),@-ERd	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 1 0 0	0 0 0 0		

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension																						
		15	8	7	0	15	8	7	0		15	8	7	0																							
OR	OR.W @(d:32,ERs.L),@(d:2,ERd)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	0	0	d	d	0	rd	0	1	0	0	0	0	0	0	0	
	OR.W @(d:32,ERs.L),@(d:16,ERd)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	0	0	rd	0	1	0	0	0	0	0	0	0	d
	OR.W @(d:32,ERs.L),@(d:32,ERd)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	0	1	rd	0	1	0	0	0	0	0	0	0	d
	OR.W @(d:32,ERs.L),@(d:16,Rd.B)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	1	0	rd	0	1	0	0	0	0	0	0	0	d
	OR.W @(d:32,ERs.L),@(d:16,Rd.W)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	0	0	rd	0	1	0	0	0	0	0	0	0	d
	OR.W @(d:32,ERs.L),@(d:16,ERd.L)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	1	0	rd	0	1	0	0	0	0	0	0	d	
	OR.W @(d:32,ERs.L),@(d:32,Rd.B)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	1	1	rd	0	1	0	0	0	0	0	0	d	
	OR.W @(d:32,ERs.L),@(d:32,Rd.W)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	0	1	rd	0	1	0	0	0	0	0	0	d	
	OR.W @(d:32,ERs.L),@(d:32,ERd.L)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	1	1	rd	0	1	0	0	0	0	0	0	d	
	OR.W @(d:32,ERs.L),@aa:16	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	0	1	0	0	0	rd	0	1	0	0	0	0	0	0	a	
	OR.W @(d:32,ERs.L),@aa:32	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	0	1	0	0	1	rd	0	1	0	0	0	0	0	0	a	
	OR.W @aa:16,@ERd									6	B	0	0	0	1	0	1	0	1	a			0	0	0	0	rd	0	1	0	0	0	0	0	0		
	OR.W @aa:16,@ERd+									6	B	0	0	0	1	0	1	0	1	a			1	0	0	0	rd	0	1	0	0	0	0	0	0		
	OR.W @aa:16,@ERd-									6	B	0	0	0	1	0	1	0	1	a			1	0	1	0	rd	0	1	0	0	0	0	0	0		
	OR.W @aa:16,@+ERd									6	B	0	0	0	1	0	1	0	1	a			1	0	0	1	rd	0	1	0	0	0	0	0	0		
	OR.W @aa:16,@-ERd									6	B	0	0	0	1	0	1	0	1	a			1	0	1	1	rd	0	1	0	0	0	0	0	0		
	OR.W @aa:16,@(d:2,ERd)									6	B	0	0	0	1	0	1	0	1	a			0	0	d	d	rd	0	1	0	0	0	0	0	0		
	OR.W @aa:16,@(d:16,ERd)									6	B	0	0	0	1	0	1	0	1	a			1	1	0	0	rd	0	1	0	0	0	0	0	0	d	
	OR.W @aa:16,@(d:32,ERd)									6	B	0	0	0	1	0	1	0	1	a			1	1	0	0	rd	0	1	0	0	0	0	0	0	d	
	OR.W @aa:16,@(d:16,Rd.B)									6	B	0	0	0	1	0	1	0	1	a			1	1	0	1	rd	0	1	0	0	0	0	0	0	d	
	OR.W @aa:16,@(d:16,Rd.W)									6	B	0	0	0	1	0	1	0	1	a			1	1	1	0	rd	0	1	0	0	0	0	0	0	d	
	OR.W @aa:16,@(d:16,ERd.L)									6	B	0	0	0	1	0	1	0	1	a			1	1	1	1	rd	0	1	0	0	0	0	0	0	d	
	OR.W @aa:16,@(d:32,Rd.B)									6	B	0	0	0	1	0	1	0	1	a			1	1	0	1	rd	0	1	0	0	0	0	0	0	d	
	OR.W @aa:16,@(d:32,Rd.W)									6	B	0	0	0	1	0	1	0	1	a			1	1	1	0	rd	0	1	0	0	0	0	0	0	d	
	OR.W @aa:16,@(d:32,ERd.L)									6	B	0	0	0	1	0	1	0	1	a			1	1	1	1	rd	0	1	0	0	0	0	0	0	d	
	OR.W @aa:16,@aa:16									6	B	0	0	0	1	0	1	0	1	a			0	1	0	0	rd	0	1	0	0	0	0	0	0	a	
	OR.W @aa:16,@aa:32									6	B	0	0	0	1	0	1	0	1	a			0	1	0	0	rd	0	1	0	0	0	0	0	0	a	
	OR.W @aa:32,@ERd									6	B	0	0	1	1	0	1	0	1	a			0	0	0	0	rd	0	1	0	0	0	0	0	0		
	OR.W @aa:32,@ERd+									6	B	0	0	1	1	0	1	0	1	a			1	0	0	0	rd	0	1	0	0	0	0	0	0		
	OR.W @aa:32,@ERd-									6	B	0	0	1	1	0	1	0	1	a			1	0	1	0	rd	0	1	0	0	0	0	0	0		
	OR.W @aa:32,@+ERd									6	B	0	0	1	1	0	1	0	1	a			1	0	0	1	rd	0	1	0	0	0	0	0	0		
	OR.W @aa:32,@-ERd									6	B	0	0	1	1	0	1	0	1	a			1	0	1	1	rd	0	1	0	0	0	0	0	0		
OR.W @aa:32,@(d:2,ERd)									6	B	0	0	1	1	0	1	0	1	a			0	0	d	d	rd	0	1	0	0	0	0	0	0			
OR.W @aa:32,@(d:16,ERd)									6	B	0	0	1	1	0	1	0	1	a			1	1	0	0	rd	0	1	0	0	0	0	0	0	d		



Instruction	Mnemonic	Opcode			Opcode			EA Ex- tension	Opcode			EA Extension				
		15	8	7	0	15	8		7	0	15		8	7	0	
OR	OR.W @aa:32,@(d:32,ERd)				6	B	0 0 1 1	0 1 0 1	a a	1 1 0 0	1	rd	0 1 0 0	0 0 0 0	d d	
	OR.W @aa:32,@(d:16,Rd.B)				6	B	0 0 1 1	0 1 0 1	a a	1 1 0 1	0	rd	0 1 0 0	0 0 0 0	d	
	OR.W @aa:32,@(d:16,Rd.W)				6	B	0 0 1 1	0 1 0 1	a a	1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d	
	OR.W @aa:32,@(d:16,ERd.L)				6	B	0 0 1 1	0 1 0 1	a a	1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d	
	OR.W @aa:32,@(d:32,Rd.B)				6	B	0 0 1 1	0 1 0 1	a a	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d d	
	OR.W @aa:32,@(d:32,Rd.W)				6	B	0 0 1 1	0 1 0 1	a a	1 1 1 0	1	rd	0 1 0 0	0 0 0 0	d d	
	OR.W @aa:32,@(d:32,ERd.L)				6	B	0 0 1 1	0 1 0 1	a a	1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d d	
	OR.W @aa:32,@aa:16				6	B	0 0 1 1	0 1 0 1	a a	0 1 0 0	0	0 0 0 0	0 1 0 0	0 0 0 0	a	
	OR.W @aa:32,@aa:32				6	B	0 0 1 1	0 1 0 1	a a	0 1 0 0	1	0 0 0 0	0 1 0 0	0 0 0 0	a a	
	OR.L #xc:16,ERd										7	A	0 1 0 0	1	rd	x
	OR.L #xc:32,ERd										7	A	0 1 0 0	0	rd	x x
	OR.L #xc:16,@ERd				0	1	0	1 1 1 0		0 0 0 0	0	rd	0 1 0 0	0 0 0 0		x
	OR.L #xc:16,@ERd+				0	1	0	1 1 1 0		1 0 0 0	0	rd	0 1 0 0	0 0 0 0		x
	OR.L #xc:16,@ERd-				0	1	0	1 1 1 0		1 0 1 0	0	rd	0 1 0 0	0 0 0 0		x
	OR.L #xc:16,@+ERd				0	1	0	1 1 1 0		1 0 0 1	0	rd	0 1 0 0	0 0 0 0		x
	OR.L #xc:16,@-ERd				0	1	0	1 1 1 0		1 0 1 1	0	rd	0 1 0 0	0 0 0 0		x
	OR.L #xc:16,@(d:2,ERd)				0	1	0	1 1 1 0		0 0 d d	0	rd	0 1 0 0	0 0 0 0		x
	OR.L #xc:16,@(d:16,ERd)				0	1	0	1 1 1 0		1 1 0 0	0	rd	0 1 0 0	0 0 0 0	d	x
	OR.L #xc:16,@(d:32,ERd)				0	1	0	1 1 1 0		1 1 0 0	1	rd	0 1 0 0	0 0 0 0	d d	x
	OR.L #xc:16,@(d:16,Rd.B)				0	1	0	1 1 1 0		1 1 0 1	0	rd	0 1 0 0	0 0 0 0	d	x
	OR.L #xc:16,@(d:16,Rd.W)				0	1	0	1 1 1 0		1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d	x
	OR.L #xc:16,@(d:16,ERd.L)				0	1	0	1 1 1 0		1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d	x
	OR.L #xc:16,@(d:32,Rd.B)				0	1	0	1 1 1 0		1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d d	x
	OR.L #xc:16,@(d:32,Rd.W)				0	1	0	1 1 1 0		1 1 1 0	1	rd	0 1 0 0	0 0 0 0	d d	x
	OR.L #xc:16,@(d:32,ERd.L)				0	1	0	1 1 1 0		1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d d	x
	OR.L #xc:16,@aa:16				0	1	0	1 1 1 0		0 1 0 0	0	0 0 0 0	0 1 0 0	0 0 0 0	a	x
	OR.L #xc:16,@aa:32				0	1	0	1 1 1 0		0 1 0 0	1	0 0 0 0	0 1 0 0	0 0 0 0	a a	x
	OR.L #xc:32,@ERd				0	1	0	1 1 1 0		0 0 0 0	0	rd	0 1 0 0	1 0 0 0		x x
	OR.L #xc:32,@ERd+				0	1	0	1 1 1 0		1 0 0 0	0	rd	0 1 0 0	1 0 0 0		x x
	OR.L #xc:32,@ERd-				0	1	0	1 1 1 0		1 0 1 0	0	rd	0 1 0 0	1 0 0 0		x x
	OR.L #xc:32,@+ERd				0	1	0	1 1 1 0		1 0 0 1	0	rd	0 1 0 0	1 0 0 0		x x
	OR.L #xc:32,@-ERd				0	1	0	1 1 1 0		1 0 1 1	0	rd	0 1 0 0	1 0 0 0		x x
OR.L #xc:32,@(d:2,ERd)				0	1	0	1 1 1 0		0 0 d d	0	rd	0 1 0 0	1 0 0 0		x x	
OR.L #xc:32,@(d:16,ERd)				0	1	0	1 1 1 0		1 1 0 0	0	rd	0 1 0 0	1 0 0 0	d	x x	

Instruction	Mnemonic	Opcode			Opcode			EA Extension	Opcode			EA Extension						
		15	8	7	0	15	8		7	0	15		8	7	0			
OR	OR.L #xc:32,@(d:32,ERd)				0	1	0	1 1 1 0		1 1 0 0	1	rd	0 1 0 0	1 0 0 0	d	d	x	x
	OR.L #xc:32,@(d:16,Rd,B)				0	1	0	1 1 1 0		1 1 0 1	0	rd	0 1 0 0	1 0 0 0	d		x	x
	OR.L #xc:32,@(d:16,Rd,W)				0	1	0	1 1 1 0		1 1 1 0	0	rd	0 1 0 0	1 0 0 0	d		x	x
	OR.L #xc:32,@(d:16,ERd,L)				0	1	0	1 1 1 0		1 1 1 1	0	rd	0 1 0 0	1 0 0 0	d		x	x
	OR.L #xc:32,@(d:32,Rd,B)				0	1	0	1 1 1 0		1 1 0 1	1	rd	0 1 0 0	1 0 0 0	d	d	x	x
	OR.L #xc:32,@(d:32,Rd,W)				0	1	0	1 1 1 0		1 1 1 0	1	rd	0 1 0 0	1 0 0 0	d	d	x	x
	OR.L #xc:32,@(d:32,ERd,L)				0	1	0	1 1 1 0		1 1 1 1	1	rd	0 1 0 0	1 0 0 0	d	d	x	x
	OR.L #xc:32,@aa:16				0	1	0	1 1 1 0		0 1 0 0	0	0 0 0 0	0 1 0 0	1 0 0 0	a		x	x
	OR.L #xc:32,@aa:32				0	1	0	1 1 1 0		0 1 0 0	1	0 0 0 0	0 1 0 0	1 0 0 0	a	a	x	x
	OR.L ERs,ERd				0	1	F	0 0 0 0		6		4	0	rs	0	rd		
	OR.L ERs,@ERd				0	1	0	1 0 0 1		0 0 0 0	0	rd	0 1 0 0	0	rs			
	OR.L ERs,@ERd+				0	1	0	1 0 0 1		1 0 0 0	0	rd	0 1 0 0	0	rs			
	OR.L ERs,@ERd-				0	1	0	1 0 0 1		1 0 1 0	0	rd	0 1 0 0	0	rs			
	OR.L ERs,@+ERd				0	1	0	1 0 0 1		1 0 0 1	0	rd	0 1 0 0	0	rs			
	OR.L ERs,@-ERd				0	1	0	1 0 0 1		1 0 1 1	0	rd	0 1 0 0	0	rs			
	OR.L ERs,@(d:2,ERd)				0	1	0	1 0 0 1		0 0 d d	0	rd	0 1 0 0	0	rs			
	OR.L ERs,@(d:16,ERd)				0	1	0	1 0 0 1		1 1 0 0	0	rd	0 1 0 0	0	rs	d		
	OR.L ERs,@(d:32,ERd)				0	1	0	1 0 0 1		1 1 0 0	1	rd	0 1 0 0	0	rs	d	d	
	OR.L ERs,@(d:16,Rd,B)				0	1	0	1 0 0 1		1 1 0 1	0	rd	0 1 0 0	0	rs	d		
	OR.L ERs,@(d:16,Rd,W)				0	1	0	1 0 0 1		1 1 1 0	0	rd	0 1 0 0	0	rs	d		
	OR.L ERs,@(d:16,ERd,L)				0	1	0	1 0 0 1		1 1 1 1	0	rd	0 1 0 0	0	rs	d		
	OR.L ERs,@(d:32,Rd,B)				0	1	0	1 0 0 1		1 1 0 1	1	rd	0 1 0 0	0	rs	d	d	
	OR.L ERs,@(d:32,Rd,W)				0	1	0	1 0 0 1		1 1 1 0	1	rd	0 1 0 0	0	rs	d	d	
	OR.L ERs,@(d:32,ERd,L)				0	1	0	1 0 0 1		1 1 1 1	1	rd	0 1 0 0	0	rs	d	d	
	OR.L ERs,@aa:16				0	1	0	1 0 0 1		0 1 0 0	0	0 0 0 0	0 1 0 0	0	rs	a		
	OR.L ERs,@aa:32				0	1	0	1 0 0 1		0 1 0 0	1	0 0 0 0	0 1 0 0	0	rs	a	a	
	OR.L @ERs,ERd				0	1	0	1 0 1 0		0 0 0 0	0	rs	0 1 0 0	0	rd			
	OR.L @ERs+,ERd				0	1	0	1 0 1 0		1 0 0 0	0	rs	0 1 0 0	0	rd			
	OR.L @ERs-,ERd				0	1	0	1 0 1 0		1 0 1 0	0	rs	0 1 0 0	0	rd			
	OR.L @+ERs,ERd				0	1	0	1 0 1 0		1 0 0 1	0	rs	0 1 0 0	0	rd			
	OR.L @-ERs,ERd				0	1	0	1 0 1 0		1 0 1 1	0	rs	0 1 0 0	0	rd			
	OR.L @(d:2,ERs),ERd				0	1	0	1 0 1 0		0 0 d d	0	rs	0 1 0 0	0	rd			
	OR.L @(d:16,ERs),ERd				0	1	0	1 0 1 0		1 1 0 0	0	rs	0 1 0 0	0	rd	d		
	OR.L @(d:32,ERs),ERd				0	1	0	1 0 1 0		1 1 0 0	1	rs	0 1 0 0	0	rd	d	d	



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension																	
		15	8	7	0	15	8	7	0		15	8	7	0																		
OR	OR.L @(d:16,Rs,B),ERd				0		1		0		1	0	1	0	1	1	0	1	0	rs	0	1	0	0	0	rd	d					
	OR.L @(d:16,Rs,W),ERd				0		1		0		1	0	1	0	1	1	1	0	0	rs	0	1	0	0	0	rd	d					
	OR.L @(d:16,ERs,L),ERd				0		1		0		1	0	1	0	1	1	1	1	0	rs	0	1	0	0	0	rd	d					
	OR.L @(d:32,Rs,B),ERd				0		1		0		1	0	1	0	1	1	0	1	1	rs	0	1	0	0	0	rd	d d					
	OR.L @(d:32,Rs,W),ERd				0		1		0		1	0	1	0	1	1	1	0	1	rs	0	1	0	0	0	rd	d d					
	OR.L @(d:32,ERs,L),ERd				0		1		0		1	0	1	0	1	1	1	1	1	rs	0	1	0	0	0	rd	d d					
	OR.L @aa:16,ERd				0		1		0		1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	rd	a					
	OR.L @aa:32,ERd				0		1		0		1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	rd	a a					
	OR.L @ERs,@ERd	0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	0	0	0	0	0	0	0	0	rd	0	0	0	0	0	0
	OR.L @ERs,@ERd+	0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	0	0	0	0	0	0	0	rd	0	1	0	0	0	0
	OR.L @ERs,@ERd-	0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	0	1	0	0	0	0	rd	0	1	0	0	0	0	
	OR.L @ERs,@+ERd	0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	0	0	1	0	0	0	rd	0	1	0	0	0	0	
	OR.L @ERs,@-ERd	0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	0	1	1	0	0	0	rd	0	1	0	0	0	0	
	OR.L @ERs,@(d:2,ERd)	0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	0	0	d	d	0	0	0	rd	0	1	0	0	0	0	
	OR.L @ERs,@(d:16,ERd)	0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	1	0	0	0	0	0	rd	0	1	0	0	0	0	
	OR.L @ERs,@(d:32,ERd)	0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	1	0	0	1	0	0	rd	0	1	0	0	0	0	
	OR.L @ERs,@(d:16,Rd,B)	0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	1	0	1	0	0	0	rd	0	1	0	0	0	0	
	OR.L @ERs,@(d:16,Rd,W)	0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	1	1	0	0	0	0	rd	0	1	0	0	0	0	
	OR.L @ERs,@(d:16,ERd,L)	0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	1	1	1	0	0	0	rd	0	1	0	0	0	0	
	OR.L @ERs,@(d:32,Rd,B)	0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	1	0	1	1	0	0	rd	0	1	0	0	0	0	
	OR.L @ERs,@(d:32,Rd,W)	0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	1	1	0	1	0	0	rd	0	1	0	0	0	0	
	OR.L @ERs,@(d:32,ERd,L)	0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	1	1	1	1	1	0	0	rd	0	1	0	0	0	0	
	OR.L @ERs,@aa:16	0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	0	1	0	0	0	0	0	rd	0	1	0	0	0	0	
	OR.L @ERs,@aa:32	0	1	0	0	0	1	0	0	6	9	0	rs	1	1	0	0	0	1	0	0	1	0	0	rd	0	1	0	0	0	0	
	OR.L @ERs+,@ERd	0	1	0	0	0	1	0	0	6	D	0	rs	1	1	0	0	0	0	0	0	0	0	0	rd	0	1	0	0	0	0	
	OR.L @ERs+,@ERd+	0	1	0	0	0	1	0	0	6	D	0	rs	1	1	0	0	1	0	0	0	0	0	0	rd	0	1	0	0	0	0	
	OR.L @ERs+,@ERd-	0	1	0	0	0	1	0	0	6	D	0	rs	1	1	0	0	1	0	1	0	0	0	0	rd	0	1	0	0	0	0	
	OR.L @ERs+,@+ERd	0	1	0	0	0	1	0	0	6	D	0	rs	1	1	0	0	1	0	0	1	0	0	0	rd	0	1	0	0	0	0	
	OR.L @ERs+,@-ERd	0	1	0	0	0	1	0	0	6	D	0	rs	1	1	0	0	1	0	1	1	0	0	0	rd	0	1	0	0	0	0	
	OR.L @ERs+,@(d:2,ERd)	0	1	0	0	0	1	0	0	6	D	0	rs	1	1	0	0	0	0	d	d	0	0	0	rd	0	1	0	0	0	0	
	OR.L @ERs+,@(d:16,ERd)	0	1	0	0	0	1	0	0	6	D	0	rs	1	1	0	0	1	1	0	0	0	0	0	rd	0	1	0	0	0	0	
	OR.L @ERs+,@(d:32,ERd)	0	1	0	0	0	1	0	0	6	D	0	rs	1	1	0	0	1	1	0	0	1	0	0	rd	0	1	0	0	0	0	
OR.L @ERs+,@(d:16,Rd,B)	0	1	0	0	0	1	0	0	6	D	0	rs	1	1	0	0	1	1	0	1	0	0	0	rd	0	1	0	0	0	0		
OR.L @ERs+,@(d:16,Rd,W)	0	1	0	0	0	1	0	0	6	D	0	rs	1	1	0	0	1	1	1	0	0	0	0	rd	0	1	0	0	0	0		

Instruction	Mnemonic	Opcode				15	Opcode				EA Extension	Opcode				EA Extension
		15	8	7	0		15	8	7	0		15	8	7	0	
OR	ORL @ERs+, @ (d:16,ERdL)	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0	1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d
	ORL @ERs+, @ (d:32,Rd.B)	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @ERs+, @ (d:32,Rd.W)	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0	1 1 1 0	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @ERs+, @ (d:32,ERdL)	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0	1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @ERs+, @aa:16	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0	0 1 0 0	0	0 0 0 0	0 1 0 0	0 0 0 0	a
	ORL @ERs+, @aa:32	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0	0 1 0 0	1	0 0 0 0	0 1 0 0	0 0 0 0	a a
	ORL @ERs-, @ERd	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	0 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	ORL @ERs-, @ERd+	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	ORL @ERs-, @ERd-	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1 0 1 0	0	rd	0 1 0 0	0 0 0 0	
	ORL @ERs+, @+ERd	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1 0 0 1	0	rd	0 1 0 0	0 0 0 0	
	ORL @ERs-, @-ERd	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1 0 1 1	0	rd	0 1 0 0	0 0 0 0	
	ORL @ERs-, @ (d:2,ERd)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	0 0 d d	0	rd	0 1 0 0	0 0 0 0	
	ORL @ERs-, @ (d:16,ERd)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1 1 0 0	0	rd	0 1 0 0	0 0 0 0	d
	ORL @ERs-, @ (d:32,ERd)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1 1 0 0	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @ERs-, @ (d:16,Rd.B)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1 1 0 1	0	rd	0 1 0 0	0 0 0 0	d
	ORL @ERs-, @ (d:16,Rd.W)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d
	ORL @ERs-, @ (d:16,ERdL)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d
	ORL @ERs-, @ (d:32,Rd.B)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @ERs-, @ (d:32,Rd.W)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1 1 1 0	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @ERs-, @ (d:32,ERdL)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @ERs-, @aa:16	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	0 1 0 0	0	0 0 0 0	0 1 0 0	0 0 0 0	a
	ORL @ERs-, @aa:32	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	0 1 0 0	1	0 0 0 0	0 1 0 0	0 0 0 0	a a
	ORL @+ERs, @ERd	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	0 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	ORL @+ERs, @ERd+	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	ORL @+ERs, @ERd-	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1 0 1 0	0	rd	0 1 0 0	0 0 0 0	
	ORL @+ERs, @+ERd	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1 0 0 1	0	rd	0 1 0 0	0 0 0 0	
	ORL @+ERs, @-ERd	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1 0 1 1	0	rd	0 1 0 0	0 0 0 0	
	ORL @+ERs, @ (d:2,ERd)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	0 0 d d	0	rd	0 1 0 0	0 0 0 0	
	ORL @+ERs, @ (d:16,ERd)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1 1 0 0	0	rd	0 1 0 0	0 0 0 0	d
	ORL @+ERs, @ (d:32,ERd)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1 1 0 0	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @+ERs, @ (d:16,Rd.B)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1 1 0 1	0	rd	0 1 0 0	0 0 0 0	d
	ORL @+ERs, @ (d:16,Rd.W)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d
ORL @+ERs, @ (d:16,ERdL)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d	
ORL @+ERs, @ (d:32,Rd.B)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d d	



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
OR	ORL @+ERs,@(d:32,Rd,W)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0 0		1 1 1 0	1	rd	0 1 0 0 0	0 0 0 0 0	d d
	ORL @+ERs,@(d:32,ERd,L)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0 0		1 1 1 1	1	rd	0 1 0 0 0	0 0 0 0 0	d d
	ORL @+ERs,@aa:16	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0 0		0 1 0 0 0	0 0 0 0 0	0 1 0 0 0	0 0 0 0 0	a	
	ORL @+ERs,@aa:32	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0 0		0 1 0 0 0	1 0 0 0 0	0 1 0 0 0	0 0 0 0 0	a a	
	ORL @-ERs,@ERd	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		0 0 0 0 0	0	rd	0 1 0 0 0	0 0 0 0 0	
	ORL @-ERs,@ERd+	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 0 0 0 0	0	rd	0 1 0 0 0	0 0 0 0 0	
	ORL @-ERs,@ERd-	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 0 1 0 0	0	rd	0 1 0 0 0	0 0 0 0 0	
	ORL @-ERs,@+ERd	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 0 0 1 0	0	rd	0 1 0 0 0	0 0 0 0 0	
	ORL @-ERs,@-ERd	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 0 1 1 0	0	rd	0 1 0 0 0	0 0 0 0 0	
	ORL @-ERs,@(d:2,ERd)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		0 0 d d	0	rd	0 1 0 0 0	0 0 0 0 0	
	ORL @-ERs,@(d:16,ERd)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 1 0 0 0	0	rd	0 1 0 0 0	0 0 0 0 0	d
	ORL @-ERs,@(d:32,ERd)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 1 0 0 1	1	rd	0 1 0 0 0	0 0 0 0 0	d d
	ORL @-ERs,@(d:16,Rd,B)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 1 0 1 0	0	rd	0 1 0 0 0	0 0 0 0 0	d
	ORL @-ERs,@(d:16,Rd,W)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 1 1 0 0	0	rd	0 1 0 0 0	0 0 0 0 0	d
	ORL @-ERs,@(d:16,ERd,L)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 1 1 1 0	0	rd	0 1 0 0 0	0 0 0 0 0	d
	ORL @-ERs,@(d:32,Rd,B)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 1 0 1 1	1	rd	0 1 0 0 0	0 0 0 0 0	d d
	ORL @-ERs,@(d:32,Rd,W)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 1 1 0 1	1	rd	0 1 0 0 0	0 0 0 0 0	d d
	ORL @-ERs,@(d:32,ERd,L)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		1 1 1 1 1	1	rd	0 1 0 0 0	0 0 0 0 0	d d
	ORL @-ERs,@aa:16	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		0 1 0 0 0	0 0 0 0 0	0 1 0 0 0	0 0 0 0 0	a	
	ORL @-ERs,@aa:32	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0 0		0 1 0 0 0	1 0 0 0 0	0 1 0 0 0	0 0 0 0 0	a a	
	ORL @(d:2,ERs),@ERd	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		0 0 0 0 0	0	rd	0 1 0 0 0	0 0 0 0 0	
	ORL @(d:2,ERs),@ERd+	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		1 0 0 0 0	0	rd	0 1 0 0 0	0 0 0 0 0	
	ORL @(d:2,ERs),@ERd-	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		1 0 1 0 0	0	rd	0 1 0 0 0	0 0 0 0 0	
	ORL @(d:2,ERs),@+ERd	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		1 0 0 1 0	0	rd	0 1 0 0 0	0 0 0 0 0	
	ORL @(d:2,ERs),@-ERd	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		1 0 1 1 0	0	rd	0 1 0 0 0	0 0 0 0 0	
	ORL @(d:2,ERs),@(d:2,ERd)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		0 0 d d	0	rd	0 1 0 0 0	0 0 0 0 0	
	ORL @(d:2,ERs),@(d:16,ERd)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		1 1 0 0 0	0	rd	0 1 0 0 0	0 0 0 0 0	d
	ORL @(d:2,ERs),@(d:32,ERd)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		1 1 0 0 1	1	rd	0 1 0 0 0	0 0 0 0 0	d d
	ORL @(d:2,ERs),@(d:16,Rd,B)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		1 1 0 1 0	0	rd	0 1 0 0 0	0 0 0 0 0	d
	ORL @(d:2,ERs),@(d:16,Rd,W)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		1 1 1 0 0	0	rd	0 1 0 0 0	0 0 0 0 0	d
ORL @(d:2,ERs),@(d:16,ERd,L)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		1 1 1 1 0	0	rd	0 1 0 0 0	0 0 0 0 0	d	
ORL @(d:2,ERs),@(d:32,Rd,B)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		1 1 0 1 1	1	rd	0 1 0 0 0	0 0 0 0 0	d d	
ORL @(d:2,ERs),@(d:32,ERd,L)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		1 1 1 1 1	1	rd	0 1 0 0 0	0 0 0 0 0	d d	
ORL @(d:2,ERs),@aa:16	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		0 1 0 0 0	0 0 0 0 0	0 1 0 0 0	0 0 0 0 0	a		

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
OR	ORL @(d:2,ERs),@aa:32	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0 0		0 1 0 0 0	1 0 0 0 0	0 1 0 0 0	0 0 0 0 0	a a
	ORL @(d:16,ERs),@ERd	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0 0	d	0 0 0 0 0	0 rd	0 1 0 0 0	0 0 0 0 0	
	ORL @(d:16,ERs),@ERd+	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0 0	d	1 0 0 0 0	0 rd	0 1 0 0 0	0 0 0 0 0	
	ORL @(d:16,ERs),@ERd-	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0 0	d	1 0 0 0 0	0 rd	0 1 0 0 0	0 0 0 0 0	
	ORL @(d:16,ERs),@+ERd	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0 0	d	1 0 0 0 1	0 rd	0 1 0 0 0	0 0 0 0 0	
	ORL @(d:16,ERs),@-ERd	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0 0	d	1 0 1 1 0	0 rd	0 1 0 0 0	0 0 0 0 0	
	ORL @(d:16,ERs),@(d:2,ERd)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0 0	d	0 0 d d	0 rd	0 1 0 0 0	0 0 0 0 0	
	ORL @(d:16,ERs),@(d:16,ERd)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0 0	d	1 1 0 0 0	0 rd	0 1 0 0 0	0 0 0 0 0	d
	ORL @(d:16,ERs),@(d:32,ERd)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0 0	d	1 1 0 0 0	1 rd	0 1 0 0 0	0 0 0 0 0	d d
	ORL @(d:16,ERs),@(d:16,Rd,B)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0 0	d	1 1 0 0 1	0 rd	0 1 0 0 0	0 0 0 0 0	d
	ORL @(d:16,ERs),@(d:16,Rd,W)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0 0	d	1 1 1 0 0	0 rd	0 1 0 0 0	0 0 0 0 0	d
	ORL @(d:16,ERs),@(d:16,ERd,L)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0 0	d	1 1 1 1 0	0 rd	0 1 0 0 0	0 0 0 0 0	d
	ORL @(d:16,ERs),@(d:32,Rd,B)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0 0	d	1 1 0 0 1	1 rd	0 1 0 0 0	0 0 0 0 0	d d
	ORL @(d:16,ERs),@(d:32,Rd,W)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0 0	d	1 1 1 0 0	1 rd	0 1 0 0 0	0 0 0 0 0	d d
	ORL @(d:16,ERs),@(d:32,ERd,L)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0 0	d	1 1 1 1 1	1 rd	0 1 0 0 0	0 0 0 0 0	d d
	ORL @(d:16,ERs),@aa:16	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0 0	d	0 1 0 0 0	0 0 0 0 0	0 1 0 0 0	0 0 0 0 0	a
	ORL @(d:16,ERs),@aa:32	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0 0	d	0 1 0 0 0	1 0 0 0 0	0 1 0 0 0	0 0 0 0 0	a a
	ORL @(d:32,ERs),@ERd	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0 0	d d	0 0 0 0 0	0 rd	0 1 0 0 0	0 0 0 0 0	
	ORL @(d:32,ERs),@ERd+	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0 0	d d	1 0 0 0 0	0 rd	0 1 0 0 0	0 0 0 0 0	
	ORL @(d:32,ERs),@ERd-	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0 0	d d	1 0 1 0 0	0 rd	0 1 0 0 0	0 0 0 0 0	
	ORL @(d:32,ERs),@+ERd	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0 0	d d	1 0 0 1 0	0 rd	0 1 0 0 0	0 0 0 0 0	
	ORL @(d:32,ERs),@-ERd	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0 0	d d	1 0 1 1 0	0 rd	0 1 0 0 0	0 0 0 0 0	
	ORL @(d:32,ERs),@(d:2,ERd)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0 0	d d	0 0 d d	0 rd	0 1 0 0 0	0 0 0 0 0	
	ORL @(d:32,ERs),@(d:16,ERd)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0 0	d d	1 1 0 0 0	0 rd	0 1 0 0 0	0 0 0 0 0	d
	ORL @(d:32,ERs),@(d:32,ERd)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0 0	d d	1 1 0 0 0	1 rd	0 1 0 0 0	0 0 0 0 0	d d
	ORL @(d:32,ERs),@(d:16,Rd,B)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0 0	d d	1 1 0 0 1	0 rd	0 1 0 0 0	0 0 0 0 0	d
	ORL @(d:32,ERs),@(d:16,Rd,W)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0 0	d d	1 1 1 0 0	0 rd	0 1 0 0 0	0 0 0 0 0	d
	ORL @(d:32,ERs),@(d:16,ERd,L)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0 0	d d	1 1 1 1 0	0 rd	0 1 0 0 0	0 0 0 0 0	d
	ORL @(d:32,ERs),@(d:32,Rd,B)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0 0	d d	1 1 0 0 1	1 rd	0 1 0 0 0	0 0 0 0 0	d d
	ORL @(d:32,ERs),@(d:32,Rd,W)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0 0	d d	1 1 1 0 0	1 rd	0 1 0 0 0	0 0 0 0 0	d d
	ORL @(d:32,ERs),@(d:32,ERd,L)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0 0	d d	1 1 1 1 1	1 rd	0 1 0 0 0	0 0 0 0 0	d d
	ORL @(d:32,ERs),@aa:16	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0 0	d d	0 1 0 0 0	0 0 0 0 0	0 1 0 0 0	0 0 0 0 0	a
ORL @(d:32,ERs),@aa:32	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0 0	d d	0 1 0 0 0	1 0 0 0 0	0 1 0 0 0	0 0 0 0 0	a a	
ORL @(d:16,Rs,B),@ERd	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0 0	d	0 0 0 0 0	0 rd	0 1 0 0 0	0 0 0 0 0		



Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
OR	ORL @(d:16,Rs.B),@ERd+	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:16,Rs.B),@ERd-	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:16,Rs.B),@+ERd	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:16,Rs.B),@ERd	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:16,Rs.B),@(d:2,ERd)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:16,Rs.B),@(d:16,ERd)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:16,Rs.B),@(d:32,ERd)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @(d:16,Rs.B),@(d:16,Rd.B)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:16,Rs.B),@(d:16,Rd.W)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:16,Rs.B),@(d:16,ERd.L)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:16,Rs.B),@(d:32,Rd.B)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @(d:16,Rs.B),@(d:32,Rd.W)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @(d:16,Rs.B),@(d:32,ERd.L)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @(d:16,Rs.B),@aa:16	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 1 0 0	0 0 0 0	a	
	ORL @(d:16,Rs.B),@aa:32	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 0 0	0 0 0 0	a a	
	ORL @(d:16,Rs.W),@ERd	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:16,Rs.W),@ERd+	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:16,Rs.W),@ERd-	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:16,Rs.W),@+ERd	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:16,Rs.W),@ERd	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:16,Rs.W),@(d:2,ERd)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:16,Rs.W),@(d:16,ERd)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:16,Rs.W),@(d:32,ERd)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @(d:16,Rs.W),@(d:16,Rd.B)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:16,Rs.W),@(d:16,Rd.W)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:16,Rs.W),@(d:16,ERd.L)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:16,Rs.W),@(d:32,Rd.B)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @(d:16,Rs.W),@(d:32,Rd.W)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @(d:16,Rs.W),@(d:32,ERd.L)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @(d:16,Rs.W),@aa:16	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 1 0 0	0 0 0 0	a	
	ORL @(d:16,Rs.W),@aa:32	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 0 0	0 0 0 0	a a	
	ORL @(d:16,ERs.L),@ERd	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 1 0 0	0 0 0 0	
ORL @(d:16,ERs.L),@ERd+	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 1 0 0	0 0 0 0		
ORL @(d:16,ERs.L),@ERd-	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 0 0	0 0 0 0		

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension			
		15	8	7	0	15	8	7	0		15	8	7	0				
OR	ORL @(d:16,ERs.L),@+ERd	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1	0 0 1	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:16,ERs.L),@-ERd	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1	0 1 1	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:16,ERs.L),@(d:2,ERd)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0	0 d d	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:16,ERs.L),@(d:16,ERd)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1	1 0 0	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:16,ERs.L),@(d:32,ERd)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1	1 0 0	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @(d:16,ERs.L),@(d:16,Rd,B)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1	1 0 1	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:16,ERs.L),@(d:16,Rd,W)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1	1 1 0	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:16,ERs.L),@(d:16,ERd,L)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1	1 1 1	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:16,ERs.L),@(d:32,Rd,B)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1	1 0 1	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @(d:16,ERs.L),@(d:32,Rd,W)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1	1 1 0	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @(d:16,ERs.L),@(d:32,ERd,L)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1	1 1 1	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @(d:16,ERs.L),@aa:16	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0	1 0 0	0	rd	0 1 0 0	0 0 0 0	a
	ORL @(d:16,ERs.L),@aa:32	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0	1 0 0	1	rd	0 1 0 0	0 0 0 0	a a
	ORL @(d:32,Rs,B),@ERd	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0	0 0 0	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:32,Rs,B),@ERd+	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1	0 0 0	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:32,Rs,B),@ERd-	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1	0 1 0	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:32,Rs,B),@+ERd	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1	0 0 1	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:32,Rs,B),@-ERd	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1	0 1 1	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:32,Rs,B),@(d:2,ERd)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0	0 d d	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:32,Rs,B),@(d:16,ERd)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1	1 0 0	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:32,Rs,B),@(d:32,ERd)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1	1 0 0	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @(d:32,Rs,B),@(d:16,Rd,B)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1	1 0 1	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:32,Rs,B),@(d:16,Rd,W)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1	1 1 0	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:32,Rs,B),@(d:16,ERd,L)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1	1 1 1	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:32,Rs,B),@(d:32,Rd,B)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1	1 0 1	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @(d:32,Rs,B),@(d:32,Rd,W)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1	1 1 0	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @(d:32,Rs,B),@(d:32,ERd,L)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1	1 1 1	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @(d:32,Rs,B),@aa:16	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0	1 0 0	0	rd	0 1 0 0	0 0 0 0	a
	ORL @(d:32,Rs,B),@aa:32	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0	1 0 0	1	rd	0 1 0 0	0 0 0 0	a a
	ORL @(d:32,Rs,W),@ERd	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0	0 0 0	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:32,Rs,W),@ERd+	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1	0 0 0	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:32,Rs,W),@ERd-	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1	0 1 0	0	rd	0 1 0 0	0 0 0 0	
ORL @(d:32,Rs,W),@+ERd	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1	0 0 1	0	rd	0 1 0 0	0 0 0 0		
ORL @(d:32,Rs,W),@-ERd	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1	0 1 1	0	rd	0 1 0 0	0 0 0 0		



Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension		
		15	8	7	0	15	8		7	0	15	8	7	0			
OR	ORL @(d:32,Rs,W),@(d2,ERd)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:32,Rs,W),@(d:16,ERd)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:32,Rs,W),@(d:32,ERd)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @(d:32,Rs,W),@(d:16,Rd,B)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:32,Rs,W),@(d:16,Rd,W)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:32,Rs,W),@(d:16,ERd,L)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:32,Rs,W),@(d:32,Rd,B)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @(d:32,Rs,W),@(d:32,Rd,W)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @(d:32,Rs,W),@(d:32,ERd,L)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @(d:32,Rs,W),@aa:16	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 1 0 0	0 0 0 0	a
	ORL @(d:32,Rs,W),@aa:32	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 1 0 0	0 0 0 0	a a
	ORL @(d:32,ERs,L),@ERd	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:32,ERs,L),@ERd+	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:32,ERs,L),@ERd-	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:32,ERs,L),@+ERd	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:32,ERs,L),@-ERd	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:32,ERs,L),@(d2,ERd)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 1 0 0	0 0 0 0	
	ORL @(d:32,ERs,L),@(d:16,ERd)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:32,ERs,L),@(d:32,ERd)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @(d:32,ERs,L),@(d:16,Rd,B)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:32,ERs,L),@(d:16,Rd,W)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:32,ERs,L),@(d:16,ERd,L)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d
	ORL @(d:32,ERs,L),@(d:32,Rd,B)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @(d:32,ERs,L),@(d:32,Rd,W)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 1 0 0	0 0 0 0	d d
ORL @(d:32,ERs,L),@(d:32,ERd,L)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d d	
ORL @(d:32,ERs,L),@aa:16	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 1 0 0	0 0 0 0	a	
ORL @(d:32,ERs,L),@aa:32	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 1 0 0	0 0 0 0	a a	
ORL @aa:16,@ERd	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 0 0 0	0	rd	0 1 0 0	0 0 0 0		
ORL @aa:16,@ERd+	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 0 0	0	rd	0 1 0 0	0 0 0 0		
ORL @aa:16,@ERd-	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 1 0	0	rd	0 1 0 0	0 0 0 0		
ORL @aa:16,@+ERd	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 0 1	0	rd	0 1 0 0	0 0 0 0		
ORL @aa:16,@-ERd	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 1 1	0	rd	0 1 0 0	0 0 0 0		
ORL @aa:16,@(d2,ERd)	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 0 d d	0	rd	0 1 0 0	0 0 0 0		
ORL @aa:16,@(d:16,ERd)	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 0	0	rd	0 1 0 0	0 0 0 0	d	

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
OR	ORL @aa:16,@(d:32,ERd)	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 0	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @aa:16,@(d:16,Rd,B)	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 1	0	rd	0 1 0 0	0 0 0 0	d
	ORL @aa:16,@(d:16,Rd,W)	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d
	ORL @aa:16,@(d:16,ERd,L)	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d
	ORL @aa:16,@(d:32,Rd,B)	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @aa:16,@(d:32,Rd,W)	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 0	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @aa:16,@(d:32,ERd,L)	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @aa:16,@aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 1 0 0	0 0 0 0	0 1 0 0	0 0 0 0	0 0 0 0	a
	ORL @aa:16,@aa:32	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 1 0 0	1 0 0 0	0 1 0 0	0 0 0 0	0 0 0 0	a a
	ORL @aa:32,@ERd	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	ORL @aa:32,@ERd+	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 0 0	0	rd	0 1 0 0	0 0 0 0	
	ORL @aa:32,@ERd-	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 1 0	0	rd	0 1 0 0	0 0 0 0	
	ORL @aa:32,@+ERd	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 0 1	0	rd	0 1 0 0	0 0 0 0	
	ORL @aa:32,@-ERd	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 1 1	0	rd	0 1 0 0	0 0 0 0	
	ORL @aa:32,@(d:2,ERd)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 0 d d	0	rd	0 1 0 0	0 0 0 0	
	ORL @aa:32,@(d:16,ERd)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 0	0	rd	0 1 0 0	0 0 0 0	d
	ORL @aa:32,@(d:32,ERd)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 0	1	rd	0 1 0 0	0 0 0 0	d d
	ORL @aa:32,@(d:16,Rd,B)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 1	0	rd	0 1 0 0	0 0 0 0	d
	ORL @aa:32,@(d:16,Rd,W)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 0	0	rd	0 1 0 0	0 0 0 0	d
	ORL @aa:32,@(d:16,ERd,L)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 1	0	rd	0 1 0 0	0 0 0 0	d
ORL @aa:32,@(d:32,Rd,B)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 1	1	rd	0 1 0 0	0 0 0 0	d d	
ORL @aa:32,@(d:32,Rd,W)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 0	1	rd	0 1 0 0	0 0 0 0	d d	
ORL @aa:32,@(d:32,ERd,L)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 1	1	rd	0 1 0 0	0 0 0 0	d d	
ORL @aa:32,@aa:16	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 1 0 0	0 0 0 0	0 1 0 0	0 0 0 0	0 0 0 0	a	
ORL @aa:32,@aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 1 0 0	1 0 0 0	0 1 0 0	0 0 0 0	0 0 0 0	a a	
ORC	ORC #xc:8,CCR											0	4	x x x x	x x x x	
	ORC #xc:8,EXR											0	4	x x x x	x x x x	
POP	POP.W Rn											6	D	0 1 1 1	m	
	POP.L ERn											6	D	0 1 1 1	0 m	
PUSH	PUSH.W Rn											6	D	1 1 1 1	m	
	PUSH.L ERn											6	D	1 1 1 1	0 m	
ROTL	ROTL.B Rd											1	2	1 0 0 0	rd	
	ROTL.B @ERd											7	D	0 rd	0 0 0 0	
	ROTL.B @ERd+	0	1	7	0 1 0 0	6	C	0 rd	1 0 0 0		1	2	1 0 0 0	0 0 0 0		



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
ROTL	ROTLB @ERd-	0	1	7	0 1 1 0	6	C	0	rd	1 0 0 0		1	2	1 0 0 0	0 0 0 0	
	ROTLB @+ERd	0	1	7	0 1 0 1	6	C	0	rd	1 0 0 0		1	2	1 0 0 0	0 0 0 0	
	ROTLB @-ERd	0	1	7	0 1 1 1	6	C	0	rd	1 0 0 0		1	2	1 0 0 0	0 0 0 0	
	ROTLB @(d2,ERd)	0	1	7	0 1 d d	6	8	0	rd	1 0 0 0		1	2	1 0 0 0	0 0 0 0	
	ROTLB @(d:16,ERd)	0	1	7	0 1 0 0	6	E	0	rd	1 0 0 0	d	1	2	1 0 0 0	0 0 0 0	
	ROTLB @(d:32,ERd)	7	8	0	rd	0 1 0 0	6	A	0 0 1 0	1 0 0 0	d d	1	2	1 0 0 0	0 0 0 0	
	ROTLB @(d:16,Rd,B)	0	1	7	0 1 0 1	6	E	0	rd	1 0 0 0	d	1	2	1 0 0 0	0 0 0 0	
	ROTLB @(d:16,Rd,W)	0	1	7	0 1 1 0	6	E	0	rd	1 0 0 0	d	1	2	1 0 0 0	0 0 0 0	
	ROTLB @(d:16,ERd,L)	0	1	7	0 1 1 1	6	E	0	rd	1 0 0 0	d	1	2	1 0 0 0	0 0 0 0	
	ROTLB @(d:32,Rd,B)	7	8	0	rd	0 1 0 1	6	A	0 0 1 0	1 0 0 0	d d	1	2	1 0 0 0	0 0 0 0	
	ROTLB @(d:32,Rd,W)	7	8	0	rd	0 1 1 0	6	A	0 0 1 0	1 0 0 0	d d	1	2	1 0 0 0	0 0 0 0	
	ROTLB @(d:32,ERd,L)	7	8	0	rd	0 1 1 1	6	A	0 0 1 0	1 0 0 0	d d	1	2	1 0 0 0	0 0 0 0	
	ROTLB @aa:8					7	F	a a a a	a a a a			1	2	1 0 0 0	0 0 0 0	
	ROTLB @aa:16					6	A	0 0 0 1	1 0 0 0	a		1	2	1 0 0 0	0 0 0 0	
	ROTLB @aa:32					6	A	0 0 1 1	1 0 0 0	a a		1	2	1 0 0 0	0 0 0 0	
	ROTLB #2,Rd											1	2	1 1 0 0	rd	
	ROTLB #2,@ERd					7	D	0	rd	0 0 0 0		1	2	1 1 0 0	0 0 0 0	
	ROTLB #2,@ERd+	0	1	7	0 1 0 0	6	C	0	rd	1 0 0 0		1	2	1 1 0 0	0 0 0 0	
	ROTLB #2,@ERd-	0	1	7	0 1 1 0	6	C	0	rd	1 0 0 0		1	2	1 1 0 0	0 0 0 0	
	ROTLB #2,@+ERd	0	1	7	0 1 0 1	6	C	0	rd	1 0 0 0		1	2	1 1 0 0	0 0 0 0	
ROTLB #2,@-ERd	0	1	7	0 1 1 1	6	C	0	rd	1 0 0 0		1	2	1 1 0 0	0 0 0 0		
ROTLB #2,@(d2,ERd)	0	1	7	0 1 d d	6	8	0	rd	1 0 0 0		1	2	1 1 0 0	0 0 0 0		
ROTLB #2,@(d:16,ERd)	0	1	7	0 1 0 0	6	E	0	rd	1 0 0 0	d	1	2	1 1 0 0	0 0 0 0		
ROTLB #2,@(d:32,ERd)	7	8	0	rd	0 1 0 0	6	A	0 0 1 0	1 0 0 0	d d	1	2	1 1 0 0	0 0 0 0		
ROTLB #2,@(d:16,Rd,B)	0	1	7	0 1 0 1	6	E	0	rd	1 0 0 0	d	1	2	1 1 0 0	0 0 0 0		
ROTLB #2,@(d:16,Rd,W)	0	1	7	0 1 1 0	6	E	0	rd	1 0 0 0	d	1	2	1 1 0 0	0 0 0 0		
ROTLB #2,@(d:16,ERd,L)	0	1	7	0 1 1 1	6	E	0	rd	1 0 0 0	d	1	2	1 1 0 0	0 0 0 0		
ROTLB #2,@(d:32,Rd,B)	7	8	0	rd	0 1 0 1	6	A	0 0 1 0	1 0 0 0	d d	1	2	1 1 0 0	0 0 0 0		
ROTLB #2,@(d:32,Rd,W)	7	8	0	rd	0 1 1 0	6	A	0 0 1 0	1 0 0 0	d d	1	2	1 1 0 0	0 0 0 0		
ROTLB #2,@(d:32,ERd,L)	7	8	0	rd	0 1 1 1	6	A	0 0 1 0	1 0 0 0	d d	1	2	1 1 0 0	0 0 0 0		
ROTLB #2,@aa:8					7	F	a a a a	a a a a			1	2	1 1 0 0	0 0 0 0		
ROTLB #2,@aa:16					6	A	0 0 0 1	1 0 0 0	a		1	2	1 1 0 0	0 0 0 0		
ROTLB #2,@aa:32					6	A	0 0 1 1	1 0 0 0	a a		1	2	1 1 0 0	0 0 0 0		
ROTLW Rd											1	2	1 0 0 1	rd		

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension																
		15	8	7	0	15	8	7	0		15	8	7	0																	
	ROTLW @ERd					7	D	1	rd	0	0	0	0	1	2	1	0	0	1	0	0	0	0	0							
	ROTLW @ERd+	0	1	5	0	1	0	0	0	6	D	0	rd	1	0	0	0	0	1	2	1	0	0	1	0	0	0	0			
	ROTLW @ERd-	0	1	5	0	1	1	1	0	6	D	0	rd	1	0	0	0	0	1	2	1	0	0	1	0	0	0	0			
	ROTLW @+ERd	0	1	5	0	1	0	1	0	6	D	0	rd	1	0	0	0	0	1	2	1	0	0	1	0	0	0	0			
	ROTLW @-ERd	0	1	5	0	1	1	1	1	6	D	0	rd	1	0	0	0	0	1	2	1	0	0	1	0	0	0	0			
	ROTLW @(d:2,ERd)	0	1	5	0	1	d	d	d	6	9	0	rd	1	0	0	0	0	1	2	1	0	0	1	0	0	0	0			
	ROTLW @(d:16,ERd)	0	1	5	0	1	0	0	0	6	F	0	rd	1	0	0	0	d	1	2	1	0	0	1	0	0	0	0			
	ROTLW @(d:32,ERd)	7	8	0	rd	0	1	0	0	6	B	0	0	1	0	1	0	0	0	d	d	1	2	1	0	0	1	0	0	0	0
	ROTLW @(d:16,Rd,B)	0	1	5	0	1	0	1	0	6	F	0	rd	1	0	0	0	d	1	2	1	0	0	1	0	0	0	0			
	ROTLW @(d:16,Rd,W)	0	1	5	0	1	1	1	0	6	F	0	rd	1	0	0	0	d	1	2	1	0	0	1	0	0	0	0			
	ROTLW @(d:16,ERd,L)	0	1	5	0	1	1	1	1	6	F	0	rd	1	0	0	0	d	1	2	1	0	0	1	0	0	0	0			
	ROTLW @(d:32,Rd,B)	7	8	0	rd	0	1	0	1	6	B	0	0	1	0	1	0	0	0	d	d	1	2	1	0	0	1	0	0	0	0
	ROTLW @(d:32,Rd,W)	7	8	0	rd	0	1	1	0	6	B	0	0	1	0	1	0	0	0	d	d	1	2	1	0	0	1	0	0	0	0
	ROTLW @(d:32,ERd,L)	7	8	0	rd	0	1	1	1	6	B	0	0	1	0	1	0	0	0	d	d	1	2	1	0	0	1	0	0	0	0
	ROTLW @aa:16					6	B	0	0	0	1	1	0	0	0	a	1	2	1	0	0	1	0	0	0	0	0				
	ROTLW @aa:32					6	B	0	0	1	1	1	0	0	0	a	a	1	2	1	0	0	1	0	0	0	0	0			
	ROTLW #2,Rd																	1	2	1	1	0	1	rd							
	ROTLW #2,@ERd					7	D	1	rd	0	0	0	0	1	2	1	1	0	1	0	0	0	0	0	0	0					
	ROTLW #2,@ERd+	0	1	5	0	1	0	0	0	6	D	0	rd	1	0	0	0	0	1	2	1	1	0	1	0	0	0	0			
	ROTLW #2,@ERd-	0	1	5	0	1	1	1	0	6	D	0	rd	1	0	0	0	0	1	2	1	1	0	1	0	0	0	0			
	ROTLW #2,@+ERd	0	1	5	0	1	0	1	0	6	D	0	rd	1	0	0	0	0	1	2	1	1	0	1	0	0	0	0			
	ROTLW #2,@-ERd	0	1	5	0	1	1	1	1	6	D	0	rd	1	0	0	0	0	1	2	1	1	0	1	0	0	0	0			
	ROTLW #2,@(d:2,ERd)	0	1	5	0	1	d	d	d	6	9	0	rd	1	0	0	0	0	1	2	1	1	0	1	0	0	0	0			
	ROTLW #2,@(d:16,ERd)	0	1	5	0	1	0	0	0	6	F	0	rd	1	0	0	0	d	1	2	1	1	0	1	0	0	0	0			
	ROTLW #2,@(d:32,ERd)	7	8	0	rd	0	1	0	0	6	B	0	0	1	0	1	0	0	0	d	d	1	2	1	1	0	1	0	0	0	0
	ROTLW #2,@(d:16,Rd,B)	0	1	5	0	1	0	1	0	6	F	0	rd	1	0	0	0	d	1	2	1	1	0	1	0	0	0	0			
	ROTLW #2,@(d:16,Rd,W)	0	1	5	0	1	1	1	0	6	F	0	rd	1	0	0	0	d	1	2	1	1	0	1	0	0	0	0			
	ROTLW #2,@(d:16,ERd,L)	0	1	5	0	1	1	1	1	6	F	0	rd	1	0	0	0	d	1	2	1	1	0	1	0	0	0	0			
	ROTLW #2,@(d:32,Rd,B)	7	8	0	rd	0	1	0	1	6	B	0	0	1	0	1	0	0	0	d	d	1	2	1	1	0	1	0	0	0	0
	ROTLW #2,@(d:32,Rd,W)	7	8	0	rd	0	1	1	0	6	B	0	0	1	0	1	0	0	0	d	d	1	2	1	1	0	1	0	0	0	0
	ROTLW #2,@(d:32,ERd,L)	7	8	0	rd	0	1	1	1	6	B	0	0	1	0	1	0	0	0	d	d	1	2	1	1	0	1	0	0	0	0
	ROTLW #2,@aa:16					6	B	0	0	0	1	1	0	0	0	a	1	2	1	1	0	1	0	0	0	0	0				
	ROTLW #2,@aa:32					6	B	0	0	1	1	1	0	0	0	a	a	1	2	1	1	0	1	0	0	0	0	0			
	ROTL L,ERd													1	2	1	0	1	1	0	rd										



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
ROTL	ROTLL @ERd	0	1	0	0 1 0 0	6	9	0	rd	1 0 0 0		1	2	1 0 1 1	0 0 0 0	
	ROTLL @ERd+	0	1	0	0 1 0 0	6	D	0	rd	1 0 0 0		1	2	1 0 1 1	0 0 0 0	
	ROTLL @ERd-	0	1	0	0 1 1 0	6	D	0	rd	1 0 0 0		1	2	1 0 1 1	0 0 0 0	
	ROTLL @+ERd	0	1	0	0 1 0 1	6	D	0	rd	1 0 0 0		1	2	1 0 1 1	0 0 0 0	
	ROTLL @-ERd	0	1	0	0 1 1 1	6	D	0	rd	1 0 0 0		1	2	1 0 1 1	0 0 0 0	
	ROTLL @(d:2,ERd)	0	1	0	0 1 d d	6	9	0	rd	1 0 0 0		1	2	1 0 1 1	0 0 0 0	
	ROTLL @(d:16,ERd)	0	1	0	0 1 0 0	6	F	0	rd	1 0 0 0	d	1	2	1 0 1 1	0 0 0 0	
	ROTLL @(d:32,ERd)	7	8	1	rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0	d d	1	2	1 0 1 1	0 0 0 0	
	ROTLL @(d:16,Rd,B)	0	1	0	0 1 0 1	6	F	0	rd	1 0 0 0	d	1	2	1 0 1 1	0 0 0 0	
	ROTLL @(d:16,Rd,W)	0	1	0	0 1 1 0	6	F	0	rd	1 0 0 0	d	1	2	1 0 1 1	0 0 0 0	
	ROTLL @(d:16,ERd,L)	0	1	0	0 1 1 1	6	F	0	rd	1 0 0 0	d	1	2	1 0 1 1	0 0 0 0	
	ROTLL @(d:32,Rd,B)	7	8	1	rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0	d d	1	2	1 0 1 1	0 0 0 0	
	ROTLL @(d:32,Rd,W)	7	8	1	rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0	d d	1	2	1 0 1 1	0 0 0 0	
	ROTLL @(d:32,ERd,L)	7	8	1	rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0	d d	1	2	1 0 1 1	0 0 0 0	
	ROTLL @aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0	1 0 0 0	a	1	2	1 0 1 1	0 0 0 0		
	ROTLL @aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0	1 0 0 0	a a	1	2	1 0 1 1	0 0 0 0		
	ROTLL #2,ERd											1	2	1 1 1 1	0 1 rd	
	ROTLL #2,@ERd	0	1	0	0 1 0 0	6	9	0	rd	1 0 0 0		1	2	1 1 1 1	0 0 0 0	
	ROTLL #2,@ERd+	0	1	0	0 1 0 0	6	D	0	rd	1 0 0 0		1	2	1 1 1 1	0 0 0 0	
	ROTLL #2,@ERd-	0	1	0	0 1 1 0	6	D	0	rd	1 0 0 0		1	2	1 1 1 1	0 0 0 0	
ROTLL #2,@+ERd	0	1	0	0 1 0 1	6	D	0	rd	1 0 0 0		1	2	1 1 1 1	0 0 0 0		
ROTLL #2,@-ERd	0	1	0	0 1 1 1	6	D	0	rd	1 0 0 0		1	2	1 1 1 1	0 0 0 0		
ROTLL #2,@(d:2,ERd)	0	1	0	0 1 d d	6	9	0	rd	1 0 0 0		1	2	1 1 1 1	0 0 0 0		
ROTLL #2,@(d:16,ERd)	0	1	0	0 1 0 0	6	F	0	rd	1 0 0 0	d	1	2	1 1 1 1	0 0 0 0		
ROTLL #2,@(d:32,ERd)	7	8	1	rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0	d d	1	2	1 1 1 1	0 0 0 0		
ROTLL #2,@(d:16,Rd,B)	0	1	0	0 1 0 1	6	F	0	rd	1 0 0 0	d	1	2	1 1 1 1	0 0 0 0		
ROTLL #2,@(d:16,Rd,W)	0	1	0	0 1 1 0	6	F	0	rd	1 0 0 0	d	1	2	1 1 1 1	0 0 0 0		
ROTLL #2,@(d:16,ERd,L)	0	1	0	0 1 1 1	6	F	0	rd	1 0 0 0	d	1	2	1 1 1 1	0 0 0 0		
ROTLL #2,@(d:32,Rd,B)	7	8	1	rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0	d d	1	2	1 1 1 1	0 0 0 0		
ROTLL #2,@(d:32,Rd,W)	7	8	1	rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0	d d	1	2	1 1 1 1	0 0 0 0		
ROTLL #2,@(d:32,ERd,L)	7	8	1	rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0	d d	1	2	1 1 1 1	0 0 0 0		
ROTLL #2,@aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0	1 0 0 0	a	1	2	1 1 1 1	0 0 0 0			
ROTLL #2,@aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0	1 0 0 0	a a	1	2	1 1 1 1	0 0 0 0			
ROTR	ROTR,B Rd										1	3	1 0 0 0	rd		

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
ROTR	ROTR.B @ERd				7	D	0	rd	0 0 0 0		1	3	1 0 0 0	0 0 0 0		
	ROTR.B @ERd+	0	1	7	0 1 0 0	6	C	0	rd	1 0 0 0		1	3	1 0 0 0	0 0 0 0	
	ROTR.B @ERd-	0	1	7	0 1 1 0	6	C	0	rd	1 0 0 0		1	3	1 0 0 0	0 0 0 0	
	ROTR.B @+ERd	0	1	7	0 1 0 1	6	C	0	rd	1 0 0 0		1	3	1 0 0 0	0 0 0 0	
	ROTR.B @-ERd	0	1	7	0 1 1 1	6	C	0	rd	1 0 0 0		1	3	1 0 0 0	0 0 0 0	
	ROTR.B @(d:2,ERd)	0	1	7	0 1 d d	6	8	0	rd	1 0 0 0		1	3	1 0 0 0	0 0 0 0	
	ROTR.B @(d:16,ERd)	0	1	7	0 1 0 0	6	E	0	rd	1 0 0 0	d	1	3	1 0 0 0	0 0 0 0	
	ROTR.B @(d:32,ERd)	7	8	0	rd	0 1 0 0	6	A	0 0 1 0	1 0 0 0	d d	1	3	1 0 0 0	0 0 0 0	
	ROTR.B @(d:16,Rd,B)	0	1	7	0 1 0 1	6	E	0	rd	1 0 0 0	d	1	3	1 0 0 0	0 0 0 0	
	ROTR.B @(d:16,Rd,W)	0	1	7	0 1 1 0	6	E	0	rd	1 0 0 0	d	1	3	1 0 0 0	0 0 0 0	
	ROTR.B @(d:16,ERd,L)	0	1	7	0 1 1 1	6	E	0	rd	1 0 0 0	d	1	3	1 0 0 0	0 0 0 0	
	ROTR.B @(d:32,Rd,B)	7	8	0	rd	0 1 0 1	6	A	0 0 1 0	1 0 0 0	d d	1	3	1 0 0 0	0 0 0 0	
	ROTR.B @(d:32,Rd,W)	7	8	0	rd	0 1 1 0	6	A	0 0 1 0	1 0 0 0	d d	1	3	1 0 0 0	0 0 0 0	
	ROTR.B @(d:32,ERd,L)	7	8	0	rd	0 1 1 1	6	A	0 0 1 0	1 0 0 0	d d	1	3	1 0 0 0	0 0 0 0	
	ROTR.B @aa:8					7	F	a a a a	a a a a		1	3	1 0 0 0	0 0 0 0		
	ROTR.B @aa:16					6	A	0 0 0 1	1 0 0 0	a	1	3	1 0 0 0	0 0 0 0		
	ROTR.B @aa:32					6	A	0 0 1 1	1 0 0 0	a a	1	3	1 0 0 0	0 0 0 0		
	ROTR.B #2,Rd										1	3	1 1 0 0	rd		
	ROTR.B #2,@ERd				7	D	0	rd	0 0 0 0		1	3	1 1 0 0	0 0 0 0		
	ROTR.B #2,@ERd+	0	1	7	0 1 0 0	6	C	0	rd	1 0 0 0		1	3	1 1 0 0	0 0 0 0	
	ROTR.B #2,@ERd-	0	1	7	0 1 1 0	6	C	0	rd	1 0 0 0		1	3	1 1 0 0	0 0 0 0	
	ROTR.B #2,@+ERd	0	1	7	0 1 0 1	6	C	0	rd	1 0 0 0		1	3	1 1 0 0	0 0 0 0	
	ROTR.B #2,@-ERd	0	1	7	0 1 1 1	6	C	0	rd	1 0 0 0		1	3	1 1 0 0	0 0 0 0	
	ROTR.B #2,@(d:2,ERd)	0	1	7	0 1 d d	6	8	0	rd	1 0 0 0		1	3	1 1 0 0	0 0 0 0	
	ROTR.B #2,@(d:16,ERd)	0	1	7	0 1 0 0	6	E	0	rd	1 0 0 0	d	1	3	1 1 0 0	0 0 0 0	
	ROTR.B #2,@(d:32,ERd)	7	8	0	rd	0 1 0 0	6	A	0 0 1 0	1 0 0 0	d d	1	3	1 1 0 0	0 0 0 0	
	ROTR.B #2,@(d:16,Rd,B)	0	1	7	0 1 0 1	6	E	0	rd	1 0 0 0	d	1	3	1 1 0 0	0 0 0 0	
	ROTR.B #2,@(d:16,Rd,W)	0	1	7	0 1 1 0	6	E	0	rd	1 0 0 0	d	1	3	1 1 0 0	0 0 0 0	
	ROTR.B #2,@(d:16,ERd,L)	0	1	7	0 1 1 1	6	E	0	rd	1 0 0 0	d	1	3	1 1 0 0	0 0 0 0	
	ROTR.B #2,@(d:32,Rd,B)	7	8	0	rd	0 1 0 1	6	A	0 0 1 0	1 0 0 0	d d	1	3	1 1 0 0	0 0 0 0	
	ROTR.B #2,@(d:32,Rd,W)	7	8	0	rd	0 1 1 0	6	A	0 0 1 0	1 0 0 0	d d	1	3	1 1 0 0	0 0 0 0	
	ROTR.B #2,@(d:32,ERd,L)	7	8	0	rd	0 1 1 1	6	A	0 0 1 0	1 0 0 0	d d	1	3	1 1 0 0	0 0 0 0	
ROTR.B #2,@aa:8					7	F	a a a a	a a a a		1	3	1 1 0 0	0 0 0 0			
ROTR.B #2,@aa:16					6	A	0 0 0 1	1 0 0 0	a	1	3	1 1 0 0	0 0 0 0			



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension
		15	8	7	0	15	8	7	0		15	8	7	0	
ROTR	ROTR.B #2, @aa:32					6	A	0 0 1 1	1 0 0 0 0	a a	1	3	1 1 0 0	0 0 0 0 0	
	ROTR.W Rd										1	3	1 0 0 1	rd	
	ROTR.W @ERd					7	D	1	rd 0 0 0 0 0		1	3	1 0 0 1	0 0 0 0 0	
	ROTR.W @ERd+	0	1	5	0 1 0 0	6	D	0	rd 1 0 0 0 0		1	3	1 0 0 1	0 0 0 0 0	
	ROTR.W @ERd-	0	1	5	0 1 1 0	6	D	0	rd 1 0 0 0 0		1	3	1 0 0 1	0 0 0 0 0	
	ROTR.W @+ERd	0	1	5	0 1 0 1	6	D	0	rd 1 0 0 0 0		1	3	1 0 0 1	0 0 0 0 0	
	ROTR.W @-ERd	0	1	5	0 1 1 1	6	D	0	rd 1 0 0 0 0		1	3	1 0 0 1	0 0 0 0 0	
	ROTR.W @(d:2,ERd)	0	1	5	0 1 d d	6	9	0	rd 1 0 0 0 0		1	3	1 0 0 1	0 0 0 0 0	
	ROTR.W @(d:16,ERd)	0	1	5	0 1 0 0	6	F	0	rd 1 0 0 0 0	d	1	3	1 0 0 1	0 0 0 0 0	
	ROTR.W @(d:32,ERd)	7	8	0	rd 0 1 0 0	6	B	0 0 1 0	1 0 0 0 0	d d	1	3	1 0 0 1	0 0 0 0 0	
	ROTR.W @(d:16,Rd,B)	0	1	5	0 1 0 1	6	F	0	rd 1 0 0 0 0	d	1	3	1 0 0 1	0 0 0 0 0	
	ROTR.W @(d:16,Rd,W)	0	1	5	0 1 1 0	6	F	0	rd 1 0 0 0 0	d	1	3	1 0 0 1	0 0 0 0 0	
	ROTR.W @(d:16,ERd,L)	0	1	5	0 1 1 1	6	F	0	rd 1 0 0 0 0	d	1	3	1 0 0 1	0 0 0 0 0	
	ROTR.W @(d:32,Rd,B)	7	8	0	rd 0 1 0 1	6	B	0 0 1 0	1 0 0 0 0	d d	1	3	1 0 0 1	0 0 0 0 0	
	ROTR.W @(d:32,Rd,W)	7	8	0	rd 0 1 1 0	6	B	0 0 1 0	1 0 0 0 0	d d	1	3	1 0 0 1	0 0 0 0 0	
	ROTR.W @(d:32,ERd,L)	7	8	0	rd 0 1 1 1	6	B	0 0 1 0	1 0 0 0 0	d d	1	3	1 0 0 1	0 0 0 0 0	
	ROTR.W @aa:16					6	B	0 0 0 1	1 0 0 0 0	a	1	3	1 0 0 1	0 0 0 0 0	
	ROTR.W @aa:32					6	B	0 0 1 1	1 0 0 0 0	a a	1	3	1 0 0 1	0 0 0 0 0	
	ROTR.W #2,Rd										1	3	1 1 0 1	rd	
	ROTR.W #2,@ERd					7	D	1	rd 0 0 0 0 0		1	3	1 1 0 1	0 0 0 0 0	
	ROTR.W #2,@ERd+	0	1	5	0 1 0 0	6	D	0	rd 1 0 0 0 0		1	3	1 1 0 1	0 0 0 0 0	
	ROTR.W #2,@ERd-	0	1	5	0 1 1 0	6	D	0	rd 1 0 0 0 0		1	3	1 1 0 1	0 0 0 0 0	
	ROTR.W #2,@+ERd	0	1	5	0 1 0 1	6	D	0	rd 1 0 0 0 0		1	3	1 1 0 1	0 0 0 0 0	
	ROTR.W #2,@-ERd	0	1	5	0 1 1 1	6	D	0	rd 1 0 0 0 0		1	3	1 1 0 1	0 0 0 0 0	
	ROTR.W #2,@(d:2,ERd)	0	1	5	0 1 d d	6	9	0	rd 1 0 0 0 0		1	3	1 1 0 1	0 0 0 0 0	
	ROTR.W #2,@(d:16,ERd)	0	1	5	0 1 0 0	6	F	0	rd 1 0 0 0 0	d	1	3	1 1 0 1	0 0 0 0 0	
	ROTR.W #2,@(d:32,ERd)	7	8	0	rd 0 1 0 0	6	B	0 0 1 0	1 0 0 0 0	d d	1	3	1 1 0 1	0 0 0 0 0	
	ROTR.W #2,@(d:16,Rd,B)	0	1	5	0 1 0 1	6	F	0	rd 1 0 0 0 0	d	1	3	1 1 0 1	0 0 0 0 0	
ROTR.W #2,@(d:16,Rd,W)	0	1	5	0 1 1 0	6	F	0	rd 1 0 0 0 0	d	1	3	1 1 0 1	0 0 0 0 0		
ROTR.W #2,@(d:16,ERd,L)	0	1	5	0 1 1 1	6	F	0	rd 1 0 0 0 0	d	1	3	1 1 0 1	0 0 0 0 0		
ROTR.W #2,@(d:32,Rd,B)	7	8	0	rd 0 1 0 1	6	B	0 0 1 0	1 0 0 0 0	d d	1	3	1 1 0 1	0 0 0 0 0		
ROTR.W #2,@(d:32,Rd,W)	7	8	0	rd 0 1 1 0	6	B	0 0 1 0	1 0 0 0 0	d d	1	3	1 1 0 1	0 0 0 0 0		
ROTR.W #2,@(d:32,ERd,L)	7	8	0	rd 0 1 1 1	6	B	0 0 1 0	1 0 0 0 0	d d	1	3	1 1 0 1	0 0 0 0 0		
ROTR.W #2,@aa:16					6	B	0 0 0 1	1 0 0 0 0	a	1	3	1 1 0 1	0 0 0 0 0		

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension
		15	8	7	0	15	8	7	0		15	8	7	0	
ROTR	ROTR.W #2,@aa:32					6	B	0 0 1 1	1 1 0 0 0 0	a a	1	3	1 1 0 1	0 0 0 0 0	
	ROTR.L ERd										1	3	1 0 1 1	0 rd	
	ROTR.L @ERd	0	1	0	0 1 0 0	6	9	0 rd	1 0 0 0 0		1	3	1 0 1 1	0 0 0 0 0	
	ROTR.L @ERd+	0	1	0	0 1 0 0	6	D	0 rd	1 0 0 0 0		1	3	1 0 1 1	0 0 0 0 0	
	ROTR.L @ERd-	0	1	0	0 1 1 0	6	D	0 rd	1 0 0 0 0		1	3	1 0 1 1	0 0 0 0 0	
	ROTR.L @+ERd	0	1	0	0 1 0 1	6	D	0 rd	1 0 0 0 0		1	3	1 0 1 1	0 0 0 0 0	
	ROTR.L @-ERd	0	1	0	0 1 1 1	6	D	0 rd	1 0 0 0 0		1	3	1 0 1 1	0 0 0 0 0	
	ROTR.L @(d:2,ERd)	0	1	0	0 1 d d	6	9	0 rd	1 0 0 0 0		1	3	1 0 1 1	0 0 0 0 0	
	ROTR.L @(d:16,ERd)	0	1	0	0 1 0 0	6	F	0 rd	1 0 0 0 0	d	1	3	1 0 1 1	0 0 0 0 0	
	ROTR.L @(d:32,ERd)	7	8	1 rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0 0	d d	1	3	1 0 1 1	0 0 0 0 0	
	ROTR.L @(d:16,Rd,B)	0	1	0	0 1 0 1	6	F	0 rd	1 0 0 0 0	d	1	3	1 0 1 1	0 0 0 0 0	
	ROTR.L @(d:16,Rd,W)	0	1	0	0 1 1 0	6	F	0 rd	1 0 0 0 0	d	1	3	1 0 1 1	0 0 0 0 0	
	ROTR.L @(d:16,ERd,L)	0	1	0	0 1 1 1	6	F	0 rd	1 0 0 0 0	d	1	3	1 0 1 1	0 0 0 0 0	
	ROTR.L @(d:32,Rd,B)	7	8	1 rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0 0	d d	1	3	1 0 1 1	0 0 0 0 0	
	ROTR.L @(d:32,Rd,W)	7	8	1 rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0 0	d d	1	3	1 0 1 1	0 0 0 0 0	
	ROTR.L @(d:32,ERd,L)	7	8	1 rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0 0	d d	1	3	1 0 1 1	0 0 0 0 0	
	ROTR.L @aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0	1 0 0 0 0	a	1	3	1 0 1 1	0 0 0 0 0	
	ROTR.L @aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0	1 0 0 0 0	a a	1	3	1 0 1 1	0 0 0 0 0	
	ROTR.L #2,ERd										1	3	1 1 1 1	0 rd	
	ROTR.L #2,@ERd	0	1	0	0 1 0 0	6	9	0 rd	1 0 0 0 0		1	3	1 1 1 1	0 0 0 0 0	
	ROTR.L #2,@ERd+	0	1	0	0 1 0 0	6	D	0 rd	1 0 0 0 0		1	3	1 1 1 1	0 0 0 0 0	
	ROTR.L #2,@ERd-	0	1	0	0 1 1 0	6	D	0 rd	1 0 0 0 0		1	3	1 1 1 1	0 0 0 0 0	
	ROTR.L #2,@+ERd	0	1	0	0 1 0 1	6	D	0 rd	1 0 0 0 0		1	3	1 1 1 1	0 0 0 0 0	
	ROTR.L #2,@-ERd	0	1	0	0 1 1 1	6	D	0 rd	1 0 0 0 0		1	3	1 1 1 1	0 0 0 0 0	
	ROTR.L #2,@(d:2,ERd)	0	1	0	0 1 d d	6	9	0 rd	1 0 0 0 0		1	3	1 1 1 1	0 0 0 0 0	
	ROTR.L #2,@(d:16,ERd)	0	1	0	0 1 0 0	6	F	0 rd	1 0 0 0 0	d	1	3	1 1 1 1	0 0 0 0 0	
	ROTR.L #2,@(d:32,ERd)	7	8	1 rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0 0	d d	1	3	1 1 1 1	0 0 0 0 0	
	ROTR.L #2,@(d:16,Rd,B)	0	1	0	0 1 0 1	6	F	0 rd	1 0 0 0 0	d	1	3	1 1 1 1	0 0 0 0 0	
	ROTR.L #2,@(d:16,Rd,W)	0	1	0	0 1 1 0	6	F	0 rd	1 0 0 0 0	d	1	3	1 1 1 1	0 0 0 0 0	
	ROTR.L #2,@(d:16,ERd,L)	0	1	0	0 1 1 1	6	F	0 rd	1 0 0 0 0	d	1	3	1 1 1 1	0 0 0 0 0	
	ROTR.L #2,@(d:32,Rd,B)	7	8	1 rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0 0	d d	1	3	1 1 1 1	0 0 0 0 0	
	ROTR.L #2,@(d:32,Rd,W)	7	8	1 rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0 0	d d	1	3	1 1 1 1	0 0 0 0 0	
ROTR.L #2,@(d:32,ERd,L)	7	8	1 rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0 0	d d	1	3	1 1 1 1	0 0 0 0 0		
ROTR.L #2,@aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0	1 0 0 0 0	a	1	3	1 1 1 1	0 0 0 0 0		



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension									
		15	8	7	0	15	8	7	0		15	8	7	0										
ROTR	ROTRL #2, @aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0	1 0 0 0 0	a a	1	3	1 1 1 1	0 0 0 0										
ROTXL	ROTXL B Rd														1	2	0 0 0 0	rd						
	ROTXL B @ERd														7	D	0	rd	0 0 0 0 0					
	ROTXL B @ERd+	0	1	7	0 1 0 0	6	C	0	rd	1 0 0 0 0		1	2	0 0 0 0	0 0 0 0 0									
	ROTXL B @ERd-	0	1	7	0 1 1 0	6	C	0	rd	1 0 0 0 0		1	2	0 0 0 0	0 0 0 0 0									
	ROTXL B @+ERd	0	1	7	0 1 0 1	6	C	0	rd	1 0 0 0 0		1	2	0 0 0 0	0 0 0 0 0									
	ROTXL B @-ERd	0	1	7	0 1 1 1	6	C	0	rd	1 0 0 0 0		1	2	0 0 0 0	0 0 0 0 0									
	ROTXL B @(d:2,ERd)	0	1	7	0 1 d d	6	8	0	rd	1 0 0 0 0		1	2	0 0 0 0	0 0 0 0 0									
	ROTXL B @(d:16,ERd)	0	1	7	0 1 0 0	6	E	0	rd	1 0 0 0 0	d	1	2	0 0 0 0	0 0 0 0 0									
	ROTXL B @(d:32,ERd)	7	8	0	rd	0 1 0 0	6	A	0 0 1 0	1 0 0 0 0	d d	1	2	0 0 0 0	0 0 0 0 0									
	ROTXL B @(d:16,Rd,B)	0	1	7	0 1 0 1	6	E	0	rd	1 0 0 0 0	d	1	2	0 0 0 0	0 0 0 0 0									
	ROTXL B @(d:16,Rd,W)	0	1	7	0 1 1 0	6	E	0	rd	1 0 0 0 0	d	1	2	0 0 0 0	0 0 0 0 0									
	ROTXL B @(d:16,ERd,L)	0	1	7	0 1 1 1	6	E	0	rd	1 0 0 0 0	d	1	2	0 0 0 0	0 0 0 0 0									
	ROTXL B @(d:32,Rd,B)	7	8	0	rd	0 1 0 1	6	A	0 0 1 0	1 0 0 0 0	d d	1	2	0 0 0 0	0 0 0 0 0									
	ROTXL B @(d:32,Rd,W)	7	8	0	rd	0 1 1 0	6	A	0 0 1 0	1 0 0 0 0	d d	1	2	0 0 0 0	0 0 0 0 0									
	ROTXL B @(d:32,ERd,L)	7	8	0	rd	0 1 1 1	6	A	0 0 1 0	1 0 0 0 0	d d	1	2	0 0 0 0	0 0 0 0 0									
	ROTXL B @aa:8														7	F	a a a a a	a a a a a						
	ROTXL B @aa:16														6	A	0 0 0 1	1 0 0 0 0	a	1	2	0 0 0 0	0 0 0 0 0	
	ROTXL B @aa:32														6	A	0 0 1 1	1 0 0 0 0	a a	1	2	0 0 0 0	0 0 0 0 0	
	ROTXL B #2,Rd														1	2	0 1 0 0	rd						
	ROTXL B #2,@ERd														7	D	0	rd	0 0 0 0 0					
	ROTXL B #2,@ERd+	0	1	7	0 1 0 0	6	C	0	rd	1 0 0 0 0		1	2	0 1 0 0	0 0 0 0 0									
ROTXL B #2,@ERd-	0	1	7	0 1 1 0	6	C	0	rd	1 0 0 0 0		1	2	0 1 0 0	0 0 0 0 0										
ROTXL B #2,@+ERd	0	1	7	0 1 0 1	6	C	0	rd	1 0 0 0 0		1	2	0 1 0 0	0 0 0 0 0										
ROTXL B #2,@-ERd	0	1	7	0 1 1 1	6	C	0	rd	1 0 0 0 0		1	2	0 1 0 0	0 0 0 0 0										
ROTXL B #2,@(d:2,ERd)	0	1	7	0 1 d d	6	8	0	rd	1 0 0 0 0		1	2	0 1 0 0	0 0 0 0 0										
ROTXL B #2,@(d:16,ERd)	0	1	7	0 1 0 0	6	E	0	rd	1 0 0 0 0	d	1	2	0 1 0 0	0 0 0 0 0										
ROTXL B #2,@(d:32,ERd)	7	8	0	rd	0 1 0 0	6	A	0 0 1 0	1 0 0 0 0	d d	1	2	0 1 0 0	0 0 0 0 0										
ROTXL B #2,@(d:16,Rd,B)	0	1	7	0 1 0 1	6	E	0	rd	1 0 0 0 0	d	1	2	0 1 0 0	0 0 0 0 0										
ROTXL B #2,@(d:16,Rd,W)	0	1	7	0 1 1 0	6	E	0	rd	1 0 0 0 0	d	1	2	0 1 0 0	0 0 0 0 0										
ROTXL B #2,@(d:16,ERd,L)	0	1	7	0 1 1 1	6	E	0	rd	1 0 0 0 0	d	1	2	0 1 0 0	0 0 0 0 0										
ROTXL B #2,@(d:32,Rd,B)	7	8	0	rd	0 1 0 1	6	A	0 0 1 0	1 0 0 0 0	d d	1	2	0 1 0 0	0 0 0 0 0										
ROTXL B #2,@(d:32,Rd,W)	7	8	0	rd	0 1 1 0	6	A	0 0 1 0	1 0 0 0 0	d d	1	2	0 1 0 0	0 0 0 0 0										
ROTXL B #2,@(d:32,ERd,L)	7	8	0	rd	0 1 1 1	6	A	0 0 1 0	1 0 0 0 0	d d	1	2	0 1 0 0	0 0 0 0 0										

Instruction	Mnemonic	Opcode			Opcode			EA Extension	Opcode			EA Extension				
		15	8	7	0	15	8		7	0	15		8	7	0	
ROTXL	ROTXL B #2, @aa:8				7	F	a a a a a	a a a a a		1	2	0 1 0 0	0 0 0 0	0		
	ROTXL B #2, @aa:16				6	A	0 0 0 0 1	1 0 0 0 0 a		1	2	0 1 0 0	0 0 0 0	0		
	ROTXL B #2, @aa:32				6	A	0 0 0 1 1	1 0 0 0 0 a a		1	2	0 1 0 0	0 0 0 0	0		
	ROTXL W Rd									1	2	0 0 0 1	rd			
	ROTXL W @ERd				7	D	1	rd	0 0 0 0 0		1	2	0 0 0 1	0 0 0 0	0	
	ROTXL W @ERd+	0	1	5	0 1 0 0	6	D	0	rd	1 0 0 0 0		1	2	0 0 0 1	0 0 0 0	0
	ROTXL W @ERd-	0	1	5	0 1 1 0	6	D	0	rd	1 0 0 0 0		1	2	0 0 0 1	0 0 0 0	0
	ROTXL W @+ERd	0	1	5	0 1 0 1	6	D	0	rd	1 0 0 0 0		1	2	0 0 0 1	0 0 0 0	0
	ROTXL W @-ERd	0	1	5	0 1 1 1	6	D	0	rd	1 0 0 0 0		1	2	0 0 0 1	0 0 0 0	0
	ROTXL W @(d:2,ERd)	0	1	5	0 1 d d	6	9	0	rd	1 0 0 0 0		1	2	0 0 0 1	0 0 0 0	0
	ROTXL W @(d:16,ERd)	0	1	5	0 1 0 0	6	F	0	rd	1 0 0 0 0 d		1	2	0 0 0 1	0 0 0 0	0
	ROTXL W @(d:32,ERd)	7	8	0	rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0 0 d d		1	2	0 0 0 1	0 0 0 0	0
	ROTXL W @(d:16,Rd,B)	0	1	5	0 1 0 1	6	F	0	rd	1 0 0 0 0 d		1	2	0 0 0 1	0 0 0 0	0
	ROTXL W @(d:16,Rd,W)	0	1	5	0 1 1 0	6	F	0	rd	1 0 0 0 0 d		1	2	0 0 0 1	0 0 0 0	0
	ROTXL W @(d:16,ERd,L)	0	1	5	0 1 1 1	6	F	0	rd	1 0 0 0 0 d		1	2	0 0 0 1	0 0 0 0	0
	ROTXL W @(d:32,Rd,B)	7	8	0	rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0 0 d d		1	2	0 0 0 1	0 0 0 0	0
	ROTXL W @(d:32,Rd,W)	7	8	0	rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0 0 d d		1	2	0 0 0 1	0 0 0 0	0
	ROTXL W @(d:32,ERd,L)	7	8	0	rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0 0 d d		1	2	0 0 0 1	0 0 0 0	0
	ROTXL W @aa:16					6	B	0 0 0 0 1	1 0 0 0 0 a		1	2	0 0 0 1	0 0 0 0	0	
	ROTXL W @aa:32					6	B	0 0 0 1 1	1 0 0 0 0 a a		1	2	0 0 0 1	0 0 0 0	0	
	ROTXL W #2,Rd									1	2	0 1 0 1	rd			
	ROTXL W #2,@ERd				7	D	1	rd	0 0 0 0 0		1	2	0 1 0 1	0 0 0 0	0	
	ROTXL W #2,@ERd+	0	1	5	0 1 0 0	6	D	0	rd	1 0 0 0 0		1	2	0 1 0 1	0 0 0 0	0
	ROTXL W #2,@ERd-	0	1	5	0 1 1 0	6	D	0	rd	1 0 0 0 0		1	2	0 1 0 1	0 0 0 0	0
	ROTXL W #2,@+ERd	0	1	5	0 1 0 1	6	D	0	rd	1 0 0 0 0		1	2	0 1 0 1	0 0 0 0	0
	ROTXL W #2,@-ERd	0	1	5	0 1 1 1	6	D	0	rd	1 0 0 0 0		1	2	0 1 0 1	0 0 0 0	0
	ROTXL W #2,@(d:2,ERd)	0	1	5	0 1 d d	6	9	0	rd	1 0 0 0 0		1	2	0 1 0 1	0 0 0 0	0
	ROTXL W #2,@(d:16,ERd)	0	1	5	0 1 0 0	6	F	0	rd	1 0 0 0 0 d		1	2	0 1 0 1	0 0 0 0	0
	ROTXL W #2,@(d:32,ERd)	7	8	0	rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0 0 d d		1	2	0 1 0 1	0 0 0 0	0
	ROTXL W #2,@(d:16,Rd,B)	0	1	5	0 1 0 1	6	F	0	rd	1 0 0 0 0 d		1	2	0 1 0 1	0 0 0 0	0
	ROTXL W #2,@(d:16,Rd,W)	0	1	5	0 1 1 0	6	F	0	rd	1 0 0 0 0 d		1	2	0 1 0 1	0 0 0 0	0
	ROTXL W #2,@(d:16,ERd,L)	0	1	5	0 1 1 1	6	F	0	rd	1 0 0 0 0 d		1	2	0 1 0 1	0 0 0 0	0
ROTXL W #2,@(d:32,Rd,B)	7	8	0	rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0 0 d d		1	2	0 1 0 1	0 0 0 0	0	
ROTXL W #2,@(d:32,Rd,W)	7	8	0	rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0 0 d d		1	2	0 1 0 1	0 0 0 0	0	



Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension	
		15	8	7	0	15	8		7	0	15	8	7	0		
ROTXL	ROTXL.W #2, @(d:32,ERd.L)	7	8	0	rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0	d d	1	2	0 1 0 1	0 0 0 0	
	ROTXL.W #2, @aa:16						6	B	0 0 0 1	1 0 0 0	a	1	2	0 1 0 1	0 0 0 0	
	ROTXL.W #2, @aa:32						6	B	0 0 1 1	1 0 0 0	a a	1	2	0 1 0 1	0 0 0 0	
	ROTXL.L ERd											1	2	0 0 1 1	0 rd	
	ROTXL.L @ERd	0	1	0	0 1 0 0	6	9	0	rd	1 0 0 0		1	2	0 0 1 1	0 0 0 0	
	ROTXL.L @ERd+	0	1	0	0 1 0 0	6	D	0	rd	1 0 0 0		1	2	0 0 1 1	0 0 0 0	
	ROTXL.L @ERd-	0	1	0	0 1 1 0	6	D	0	rd	1 0 0 0		1	2	0 0 1 1	0 0 0 0	
	ROTXL.L @+ERd	0	1	0	0 1 0 1	6	D	0	rd	1 0 0 0		1	2	0 0 1 1	0 0 0 0	
	ROTXL.L @-ERd	0	1	0	0 1 1 1	6	D	0	rd	1 0 0 0		1	2	0 0 1 1	0 0 0 0	
	ROTXL.L @(d:2,ERd)	0	1	0	0 1 d d	6	9	0	rd	1 0 0 0		1	2	0 0 1 1	0 0 0 0	
	ROTXL.L @(d:16,ERd)	0	1	0	0 1 0 0	6	F	0	rd	1 0 0 0	d	1	2	0 0 1 1	0 0 0 0	
	ROTXL.L @(d:32,ERd)	7	8	1	rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0	d d	1	2	0 0 1 1	0 0 0 0	
	ROTXL.L @(d:16,Rd.B)	0	1	0	0 1 0 1	6	F	0	rd	1 0 0 0	d	1	2	0 0 1 1	0 0 0 0	
	ROTXL.L @(d:16,Rd.W)	0	1	0	0 1 1 0	6	F	0	rd	1 0 0 0	d	1	2	0 0 1 1	0 0 0 0	
	ROTXL.L @(d:16,ERd.L)	0	1	0	0 1 1 1	6	F	0	rd	1 0 0 0	d	1	2	0 0 1 1	0 0 0 0	
	ROTXL.L @(d:32,Rd.B)	7	8	1	rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0	d d	1	2	0 0 1 1	0 0 0 0	
	ROTXL.L @(d:32,Rd.W)	7	8	1	rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0	d d	1	2	0 0 1 1	0 0 0 0	
	ROTXL.L @(d:32,ERd.L)	7	8	1	rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0	d d	1	2	0 0 1 1	0 0 0 0	
	ROTXL.L @aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0	1 0 0 0	a		1	2	0 0 1 1	0 0 0 0	
	ROTXL.L @aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0	1 0 0 0	a a		1	2	0 0 1 1	0 0 0 0	
	ROTXL.L #2,ERd											1	2	0 1 1 1	0 rd	
	ROTXL.L #2,@ERd	0	1	0	0 1 0 0	6	9	0	rd	1 0 0 0		1	2	0 1 1 1	0 0 0 0	
	ROTXL.L #2,@ERd+	0	1	0	0 1 0 0	6	D	0	rd	1 0 0 0		1	2	0 1 1 1	0 0 0 0	
	ROTXL.L #2,@ERd-	0	1	0	0 1 1 0	6	D	0	rd	1 0 0 0		1	2	0 1 1 1	0 0 0 0	
	ROTXL.L #2,@+ERd	0	1	0	0 1 0 1	6	D	0	rd	1 0 0 0		1	2	0 1 1 1	0 0 0 0	
	ROTXL.L #2,@-ERd	0	1	0	0 1 1 1	6	D	0	rd	1 0 0 0		1	2	0 1 1 1	0 0 0 0	
	ROTXL.L #2,@(d:2,ERd)	0	1	0	0 1 d d	6	9	0	rd	1 0 0 0		1	2	0 1 1 1	0 0 0 0	
	ROTXL.L #2,@(d:16,ERd)	0	1	0	0 1 0 0	6	F	0	rd	1 0 0 0	d	1	2	0 1 1 1	0 0 0 0	
	ROTXL.L #2,@(d:32,ERd)	7	8	1	rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0	d d	1	2	0 1 1 1	0 0 0 0	
	ROTXL.L #2,@(d:16,Rd.B)	0	1	0	0 1 0 1	6	F	0	rd	1 0 0 0	d	1	2	0 1 1 1	0 0 0 0	
ROTXL.L #2,@(d:16,Rd.W)	0	1	0	0 1 1 0	6	F	0	rd	1 0 0 0	d	1	2	0 1 1 1	0 0 0 0		
ROTXL.L #2,@(d:16,ERd.L)	0	1	0	0 1 1 1	6	F	0	rd	1 0 0 0	d	1	2	0 1 1 1	0 0 0 0		
ROTXL.L #2,@(d:32,Rd.B)	7	8	1	rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0	d d	1	2	0 1 1 1	0 0 0 0		
ROTXL.L #2,@(d:32,Rd.W)	7	8	1	rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0	d d	1	2	0 1 1 1	0 0 0 0		

Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension								
		15	8	7	0	15	8	7	0		15	8	7	0									
ROTXL	ROTXLL #2,@(d:32,ERdL)	7	8	1	rd	0 1 1 1	6	B	0 0 1 0	1 1 0 0 0 0	d d	1	2	0 1 1 1	0 0 0 0 0								
	ROTXLL #2,@aa:16	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0 0 0	a	1	2	0 1 1 1	0 0 0 0 0								
	ROTXLL #2,@aa:32	0	1		0	0 1 0 0	6	B	0 0 1 0	1 1 0 0 0 0	a a	1	2	0 1 1 1	0 0 0 0 0								
ROTXR	ROTXR.B Rd														1	3	0 0 0 0 0	rd					
	ROTXR.B @ERd														7	D	0	rd	0 0 0 0 0	1	3	0 0 0 0 0	0 0 0 0 0
	ROTXR.B @ERd+	0	1		7	0 1 0 0	6	C	0	rd	1 0 0 0 0		1	3	0 0 0 0 0	0 0 0 0 0							
	ROTXR.B @ERd-	0	1		7	0 1 1 0	6	C	0	rd	1 0 0 0 0		1	3	0 0 0 0 0	0 0 0 0 0							
	ROTXR.B @+ERd	0	1		7	0 1 0 1	6	C	0	rd	1 0 0 0 0		1	3	0 0 0 0 0	0 0 0 0 0							
	ROTXR.B @-ERd	0	1		7	0 1 1 1	6	C	0	rd	1 0 0 0 0		1	3	0 0 0 0 0	0 0 0 0 0							
	ROTXR.B @(d:2,ERd)	0	1		7	0 1 d d	6	8	0	rd	1 0 0 0 0		1	3	0 0 0 0 0	0 0 0 0 0							
	ROTXR.B @(d:16,ERd)	0	1		7	0 1 0 0	6	E	0	rd	1 0 0 0 0	d	1	3	0 0 0 0 0	0 0 0 0 0							
	ROTXR.B @(d:32,ERd)	7	8	0	1	rd	0 1 0 0	6	A	0 0 1 0	1 1 0 0 0 0	d d	1	3	0 0 0 0 0	0 0 0 0 0							
	ROTXR.B @(d:16,Rd.B)	0	1		7	0 1 0 1	6	E	0	rd	1 0 0 0 0	d	1	3	0 0 0 0 0	0 0 0 0 0							
	ROTXR.B @(d:16,Rd.W)	0	1		7	0 1 1 0	6	E	0	rd	1 0 0 0 0	d	1	3	0 0 0 0 0	0 0 0 0 0							
	ROTXR.B @(d:16,ERd.L)	0	1		7	0 1 1 1	6	E	0	rd	1 0 0 0 0	d	1	3	0 0 0 0 0	0 0 0 0 0							
	ROTXR.B @(d:32,Rd.B)	7	8	0	1	rd	0 1 0 1	6	A	0 0 1 0	1 1 0 0 0 0	d d	1	3	0 0 0 0 0	0 0 0 0 0							
	ROTXR.B @(d:32,Rd.W)	7	8	0	1	rd	0 1 1 0	6	A	0 0 1 0	1 1 0 0 0 0	d d	1	3	0 0 0 0 0	0 0 0 0 0							
	ROTXR.B @(d:32,ERd.L)	7	8	0	1	rd	0 1 1 1	6	A	0 0 1 0	1 1 0 0 0 0	d d	1	3	0 0 0 0 0	0 0 0 0 0							
	ROTXR.B @aa:8														7	F	a a a a a	a a a a a	1	3	0 0 0 0 0	0 0 0 0 0	
	ROTXR.B @aa:16														6	A	0 0 0 0 1	1 1 0 0 0 0	a	1	3	0 0 0 0 0	0 0 0 0 0
	ROTXR.B @aa:32														6	A	0 0 1 1 1	1 1 0 0 0 0	a a	1	3	0 0 0 0 0	0 0 0 0 0
	ROTXR.B #2,Rd														1	3	0 1 0 0	rd					
	ROTXR.B #2,@ERd														7	D	0	rd	0 0 0 0 0	1	3	0 1 0 0 0	0 0 0 0 0
	ROTXR.B #2,@ERd+	0	1		7	0 1 0 0	6	C	0	rd	1 0 0 0 0		1	3	0 1 0 0 0	0 0 0 0 0							
	ROTXR.B #2,@ERd-	0	1		7	0 1 1 0	6	C	0	rd	1 0 0 0 0		1	3	0 1 0 0 0	0 0 0 0 0							
	ROTXR.B #2,@+ERd	0	1		7	0 1 0 1	6	C	0	rd	1 0 0 0 0		1	3	0 1 0 0 0	0 0 0 0 0							
	ROTXR.B #2,@-ERd	0	1		7	0 1 1 1	6	C	0	rd	1 0 0 0 0		1	3	0 1 0 0 0	0 0 0 0 0							
ROTXR.B #2,@(d:2,ERd)	0	1		7	0 1 d d	6	8	0	rd	1 0 0 0 0		1	3	0 1 0 0 0	0 0 0 0 0								
ROTXR.B #2,@(d:16,ERd)	0	1		7	0 1 0 0	6	E	0	rd	1 0 0 0 0	d	1	3	0 1 0 0 0	0 0 0 0 0								
ROTXR.B #2,@(d:32,ERd)	7	8	0	1	rd	0 1 0 0	6	A	0 0 1 0	1 1 0 0 0 0	d d	1	3	0 1 0 0 0	0 0 0 0 0								
ROTXR.B #2,@(d:16,Rd.B)	0	1		7	0 1 0 1	6	E	0	rd	1 0 0 0 0	d	1	3	0 1 0 0 0	0 0 0 0 0								
ROTXR.B #2,@(d:16,Rd.W)	0	1		7	0 1 1 0	6	E	0	rd	1 0 0 0 0	d	1	3	0 1 0 0 0	0 0 0 0 0								
ROTXR.B #2,@(d:16,ERd.L)	0	1		7	0 1 1 1	6	E	0	rd	1 0 0 0 0	d	1	3	0 1 0 0 0	0 0 0 0 0								
ROTXR.B #2,@(d:32,Rd.B)	7	8	0	1	rd	0 1 0 1	6	A	0 0 1 0	1 1 0 0 0 0	d d	1	3	0 1 0 0 0	0 0 0 0 0								



Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension	
		15	8	7	0	15	8		7	0	15	8	7	0		
ROTXR	ROTXR.B #2, @(d:32,Rd,W)	7	8	0	rd	0 1 1 0	6	A	0 0 1 0	1 1 0 0 0	d d	1	3	0 1 0 0	0 0 0 0	
	ROTXR.B #2, @(d:32,ERd,L)	7	8	0	rd	0 1 1 1	6	A	0 0 1 0	1 1 0 0 0	d d	1	3	0 1 0 0	0 0 0 0	
	ROTXR.B #2, @aa:8						7	F	a a a a	a a a a a		1	3	0 1 0 0	0 0 0 0	
	ROTXR.B #2, @aa:16						6	A	0 0 0 0	1 1 0 0 0	a	1	3	0 1 0 0	0 0 0 0	
	ROTXR.B #2, @aa:32						6	A	0 0 1 1	1 1 0 0 0	a a	1	3	0 1 0 0	0 0 0 0	
	ROTXR.W Rd											1	3	0 0 0 1	rd	
	ROTXR.W @ERd						7	D	1 rd	0 0 0 0		1	3	0 0 0 1	0 0 0 0	
	ROTXR.W @ERd+	0	1	5	0 1 0 0		6	D	0 rd	1 0 0 0		1	3	0 0 0 1	0 0 0 0	
	ROTXR.W @ERd-	0	1	5	0 1 1 0		6	D	0 rd	1 0 0 0		1	3	0 0 0 1	0 0 0 0	
	ROTXR.W @+ERd	0	1	5	0 1 0 1		6	D	0 rd	1 0 0 0		1	3	0 0 0 1	0 0 0 0	
	ROTXR.W @-ERd	0	1	5	0 1 1 1		6	D	0 rd	1 0 0 0		1	3	0 0 0 1	0 0 0 0	
	ROTXR.W @(d:2,ERd)	0	1	5	0 1 d d		6	9	0 rd	1 0 0 0		1	3	0 0 0 1	0 0 0 0	
	ROTXR.W @(d:16,ERd)	0	1	5	0 1 0 0		6	F	0 rd	1 0 0 0	d	1	3	0 0 0 1	0 0 0 0	
	ROTXR.W @(d:32,ERd)	7	8	0	rd	0 1 0 0	6	B	0 0 1 0	1 1 0 0 0	d d	1	3	0 0 0 1	0 0 0 0	
	ROTXR.W @(d:16,Rd,B)	0	1	5	0 1 0 1		6	F	0 rd	1 0 0 0	d	1	3	0 0 0 1	0 0 0 0	
	ROTXR.W @(d:16,Rd,W)	0	1	5	0 1 1 0		6	F	0 rd	1 0 0 0	d	1	3	0 0 0 1	0 0 0 0	
	ROTXR.W @(d:16,ERd,L)	0	1	5	0 1 1 1		6	F	0 rd	1 0 0 0	d	1	3	0 0 0 1	0 0 0 0	
	ROTXR.W @(d:32,Rd,B)	7	8	0	rd	0 1 0 1	6	B	0 0 1 0	1 1 0 0 0	d d	1	3	0 0 0 1	0 0 0 0	
	ROTXR.W @(d:32,Rd,W)	7	8	0	rd	0 1 1 0	6	B	0 0 1 0	1 1 0 0 0	d d	1	3	0 0 0 1	0 0 0 0	
	ROTXR.W @(d:32,ERd,L)	7	8	0	rd	0 1 1 1	6	B	0 0 1 0	1 1 0 0 0	d d	1	3	0 0 0 1	0 0 0 0	
	ROTXR.W @aa:16						6	B	0 0 0 1	1 1 0 0 0	a	1	3	0 0 0 1	0 0 0 0	
	ROTXR.W @aa:32						6	B	0 0 1 1	1 1 0 0 0	a a	1	3	0 0 0 1	0 0 0 0	
	ROTXR.W #2,Rd											1	3	0 1 0 1	rd	
	ROTXR.W #2, @ERd						7	D	1 rd	0 0 0 0		1	3	0 1 0 1	0 0 0 0	
	ROTXR.W #2, @ERd+	0	1	5	0 1 0 0		6	D	0 rd	1 0 0 0		1	3	0 1 0 1	0 0 0 0	
	ROTXR.W #2, @ERd-	0	1	5	0 1 1 0		6	D	0 rd	1 0 0 0		1	3	0 1 0 1	0 0 0 0	
	ROTXR.W #2, @+ERd	0	1	5	0 1 0 1		6	D	0 rd	1 0 0 0		1	3	0 1 0 1	0 0 0 0	
	ROTXR.W #2, @-ERd	0	1	5	0 1 1 1		6	D	0 rd	1 0 0 0		1	3	0 1 0 1	0 0 0 0	
ROTXR.W #2, @(d:2,ERd)	0	1	5	0 1 d d		6	9	0 rd	1 0 0 0		1	3	0 1 0 1	0 0 0 0		
ROTXR.W #2, @(d:16,ERd)	0	1	5	0 1 0 0		6	F	0 rd	1 0 0 0	d	1	3	0 1 0 1	0 0 0 0		
ROTXR.W #2, @(d:32,ERd)	7	8	0	rd	0 1 0 0	6	B	0 0 1 0	1 1 0 0 0	d d	1	3	0 1 0 1	0 0 0 0		
ROTXR.W #2, @(d:16,Rd,B)	0	1	5	0 1 0 1		6	F	0 rd	1 0 0 0	d	1	3	0 1 0 1	0 0 0 0		
ROTXR.W #2, @(d:16,Rd,W)	0	1	5	0 1 1 0		6	F	0 rd	1 0 0 0	d	1	3	0 1 0 1	0 0 0 0		
ROTXR.W #2, @(d:16,ERd,L)	0	1	5	0 1 1 1		6	F	0 rd	1 0 0 0	d	1	3	0 1 0 1	0 0 0 0		

Instruction	Mnemonic	Opcode						EA Ex- tension	Opcode						EA Extension																							
		15	8	7	0				15	8	7	0																										
ROTXR	ROTXR.W #2,@(d:32,Rd.B)	7	8	0	rd	0	1	0	1	6	B	0	0	1	0	1	0	0	0	0	d	d	1	3	0	1	0	1	0	0	0	0	0	0				
	ROTXR.W #2,@(d:32,Rd.W)	7	8	0	rd	0	1	1	0	6	B	0	0	1	0	1	0	0	0	0	0	d	d	1	3	0	1	0	1	0	0	0	0	0				
	ROTXR.W #2,@(d:32,ERd.L)	7	8	0	rd	0	1	1	1	6	B	0	0	1	0	1	0	0	0	0	0	d	d	1	3	0	1	0	1	0	0	0	0	0	0			
	ROTXR.W #2,@aa:16					6	B	0	0	0	0	1	0	0	0	0	0	0	0	0	0	a		1	3	0	1	0	1	0	0	0	0	0	0	0		
	ROTXR.W #2,@aa:32					6	B	0	0	1	1	1	0	0	0	0	0	0	0	0	0	a	a	1	3	0	1	0	1	0	0	0	0	0	0	0		
	ROTXR.L ERd																							1	3	0	0	1	1	0	rd							
	ROTXR.L @ERd	0	1	0	0	1	0	0	6	9	0	rd	1	0	0	0								1	3	0	0	1	1	0	0	0	0	0	0	0		
	ROTXR.L @ERd+	0	1	0	0	1	0	0	6	D	0	rd	1	0	0	0								1	3	0	0	1	1	0	0	0	0	0	0	0		
	ROTXR.L @ERd-	0	1	0	0	1	1	0	6	D	0	rd	1	0	0	0								1	3	0	0	1	1	0	0	0	0	0	0	0	0	
	ROTXR.L @+ERd	0	1	0	0	1	0	1	6	D	0	rd	1	0	0	0								1	3	0	0	1	1	0	0	0	0	0	0	0	0	
	ROTXR.L @-ERd	0	1	0	0	1	1	1	6	D	0	rd	1	0	0	0								1	3	0	0	1	1	0	0	0	0	0	0	0	0	
	ROTXR.L @(d:2,ERd)	0	1	0	0	1	d	d	6	9	0	rd	1	0	0	0								1	3	0	0	1	1	0	0	0	0	0	0	0	0	
	ROTXR.L @(d:16,ERd)	0	1	0	0	1	0	0	6	F	0	rd	1	0	0	0	d							1	3	0	0	1	1	0	0	0	0	0	0	0	0	
	ROTXR.L @(d:32,ERd)	7	8	1	rd	0	1	0	0	6	B	0	0	1	0	1	0	0	0	0	0	d	d	1	3	0	0	1	1	0	0	0	0	0	0	0	0	
	ROTXR.L @(d:16,Rd.B)	0	1	0	0	1	0	1	6	F	0	rd	1	0	0	0	d							1	3	0	0	1	1	0	0	0	0	0	0	0	0	
	ROTXR.L @(d:16,Rd.W)	0	1	0	0	1	1	0	6	F	0	rd	1	0	0	0	d							1	3	0	0	1	1	0	0	0	0	0	0	0	0	
	ROTXR.L @(d:16,ERd.L)	0	1	0	0	1	1	1	6	F	0	rd	1	0	0	0	d							1	3	0	0	1	1	0	0	0	0	0	0	0	0	
	ROTXR.L @(d:32,Rd.B)	7	8	1	rd	0	1	0	1	6	B	0	0	1	0	1	0	0	0	0	0	d	d	1	3	0	0	1	1	0	0	0	0	0	0	0	0	
	ROTXR.L @(d:32,Rd.W)	7	8	1	rd	0	1	1	0	6	B	0	0	1	0	1	0	0	0	0	0	d	d	1	3	0	0	1	1	0	0	0	0	0	0	0	0	
	ROTXR.L @(d:32,ERd.L)	7	8	1	rd	0	1	1	1	6	B	0	0	1	0	1	0	0	0	0	0	d	d	1	3	0	0	1	1	0	0	0	0	0	0	0	0	0
	ROTXR.L @aa:16	0	1	0	0	1	0	0	6	B	0	0	0	0	1	0	0	0	0	0	0	a		1	3	0	0	1	1	0	0	0	0	0	0	0	0	
	ROTXR.L @aa:32	0	1	0	0	1	0	0	6	B	0	0	1	0	1	0	0	0	0	0	0	a	a	1	3	0	0	1	1	0	0	0	0	0	0	0	0	
	ROTXR.L #2,ERd																							1	3	0	1	1	0	rd								
	ROTXR.L #2,@ERd	0	1	0	0	1	0	0	6	9	0	rd	1	0	0	0								1	3	0	1	1	1	0	0	0	0	0	0	0	0	
	ROTXR.L #2,@ERd+	0	1	0	0	1	0	0	6	D	0	rd	1	0	0	0								1	3	0	1	1	1	0	0	0	0	0	0	0	0	
	ROTXR.L #2,@ERd-	0	1	0	0	1	1	0	6	D	0	rd	1	0	0	0								1	3	0	1	1	1	0	0	0	0	0	0	0	0	0
	ROTXR.L #2,@+ERd	0	1	0	0	1	0	1	6	D	0	rd	1	0	0	0								1	3	0	1	1	1	0	0	0	0	0	0	0	0	0
	ROTXR.L #2,@-ERd	0	1	0	0	1	1	1	6	D	0	rd	1	0	0	0								1	3	0	1	1	1	0	0	0	0	0	0	0	0	0
	ROTXR.L #2,@(d:2,ERd)	0	1	0	0	1	d	d	6	9	0	rd	1	0	0	0								1	3	0	1	1	1	0	0	0	0	0	0	0	0	
	ROTXR.L #2,@(d:16,ERd)	0	1	0	0	1	0	0	6	F	0	rd	1	0	0	0	d							1	3	0	1	1	1	0	0	0	0	0	0	0	0	0
	ROTXR.L #2,@(d:32,ERd)	7	8	1	rd	0	1	0	0	6	B	0	0	1	0	1	0	0	0	0	0	d	d	1	3	0	1	1	1	0	0	0	0	0	0	0	0	0
	ROTXR.L #2,@(d:16,Rd.B)	0	1	0	0	1	0	1	6	F	0	rd	1	0	0	0	d							1	3	0	1	1	1	0	0	0	0	0	0	0	0	0
ROTXR.L #2,@(d:16,Rd.W)	0	1	0	0	1	1	0	6	F	0	rd	1	0	0	0	d							1	3	0	1	1	1	0	0	0	0	0	0	0	0	0	
ROTXR.L #2,@(d:16,ERd.L)	0	1	0	0	1	1	1	6	F	0	rd	1	0	0	0	d							1	3	0	1	1	1	0	0	0	0	0	0	0	0	0	



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
ROTXR	ROTXR.L #2, @(d:32,Rd.B)	7	8	1	rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0	d d	1	3	0 1 1 1	0 0 0 0	
	ROTXR.L #2, @(d:32,Rd.W)	7	8	1	rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0	d d	1	3	0 1 1 1	0 0 0 0	
	ROTXR.L #2, @(d:32,ERd.L)	7	8	1	rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0	d d	1	3	0 1 1 1	0 0 0 0	
	ROTXR.L #2, @aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0	1 0 0 0	a	1	3	0 1 1 1	0 0 0 0		
	ROTXR.L #2, @aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0	1 0 0 0	a a	1	3	0 1 1 1	0 0 0 0		
RTE	RTE										5	6	0 1 1 1	0 0 0 0		
RTE/L	RTE/L ERn										5	6	0 0 0 0	0	m	
	RTE/L (ERn-ERn+1)										5	6	0 0 0 1	0	rn+1	
	RTE/L (ERn-ERn+2)										5	6	0 0 1 0	0	rn+2	
	RTE/L (ERn-ERn+3)										5	6	0 0 1 1	0	rn+3	
RTS	RTS										5	4	0 1 1 1	0 0 0 0		
RTS/L	RTS/L ERn										5	4	0 0 0 0	0	m	
	RTS/L (ERn-ERn+1)										5	4	0 0 0 1	0	rn+1	
	RTS/L (ERn-ERn+2)										5	4	0 0 1 0	0	rn+2	
	RTS/L (ERn-ERn+3)										5	4	0 0 1 1	0	rn+3	
SHAL	SHALB Rd										1	0	1 0 0 0	rd		
	SHALB @ ERd					7	D	0 rd	0 0 0 0		1	0	1 0 0 0	0 0 0 0		
	SHALB @ ERd+	0	1	7	0 1 0 0	6	C	0 rd	1 0 0 0		1	0	1 0 0 0	0 0 0 0		
	SHALB @ ERd-	0	1	7	0 1 1 0	6	C	0 rd	1 0 0 0		1	0	1 0 0 0	0 0 0 0		
	SHALB @+ERd	0	1	7	0 1 0 1	6	C	0 rd	1 0 0 0		1	0	1 0 0 0	0 0 0 0		
	SHALB @-ERd	0	1	7	0 1 1 1	6	C	0 rd	1 0 0 0		1	0	1 0 0 0	0 0 0 0		
	SHALB @(d:2,ERd)	0	1	7	0 1 d d	6	8	0 rd	1 0 0 0		1	0	1 0 0 0	0 0 0 0		
	SHALB @(d:16,ERd)	0	1	7	0 1 0 0	6	E	0 rd	1 0 0 0	d	1	0	1 0 0 0	0 0 0 0		
	SHALB @(d:32,ERd)	7	8	0	rd	0 1 0 0	6	A	0 0 1 0	1 0 0 0	d d	1	0	1 0 0 0	0 0 0 0	
	SHALB @(d:16,Rd.B)	0	1	7	0 1 0 1	6	E	0 rd	1 0 0 0	d	1	0	1 0 0 0	0 0 0 0		
	SHALB @(d:16,Rd.W)	0	1	7	0 1 1 0	6	E	0 rd	1 0 0 0	d	1	0	1 0 0 0	0 0 0 0		
	SHALB @(d:16,ERd.L)	0	1	7	0 1 1 1	6	E	0 rd	1 0 0 0	d	1	0	1 0 0 0	0 0 0 0		
	SHALB @(d:32,Rd.B)	7	8	0	rd	0 1 0 1	6	A	0 0 1 0	1 0 0 0	d d	1	0	1 0 0 0	0 0 0 0	
	SHALB @(d:32,Rd.W)	7	8	0	rd	0 1 1 0	6	A	0 0 1 0	1 0 0 0	d d	1	0	1 0 0 0	0 0 0 0	
	SHALB @(d:32,ERd.L)	7	8	0	rd	0 1 1 1	6	A	0 0 1 0	1 0 0 0	d d	1	0	1 0 0 0	0 0 0 0	
	SHALB @aa:8					7	F	a a a a	a a a a		1	0	1 0 0 0	0 0 0 0		
	SHALB @aa:16					6	A	0 0 0 1	1 0 0 0	a	1	0	1 0 0 0	0 0 0 0		
SHALB @aa:32					6	A	0 0 1 1	1 0 0 0	a a	1	0	1 0 0 0	0 0 0 0			
SHALB #2,Rd											1	0	1 1 0 0	rd		

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
SHAL	SHAL.B #2, @ERd				7	D	0	rd	0 0 0 0		1	0	1 1 0 0	0 0 0 0		
	SHAL.B #2, @ERd+	0	1	7	0 1 0 0	6	C	0	rd	1 0 0 0		1	0	1 1 0 0	0 0 0 0	
	SHAL.B #2, @ERd-	0	1	7	0 1 1 0	6	C	0	rd	1 0 0 0		1	0	1 1 0 0	0 0 0 0	
	SHAL.B #2, @+ERd	0	1	7	0 1 0 1	6	C	0	rd	1 0 0 0		1	0	1 1 0 0	0 0 0 0	
	SHAL.B #2, @-ERd	0	1	7	0 1 1 1	6	C	0	rd	1 0 0 0		1	0	1 1 0 0	0 0 0 0	
	SHAL.B #2, @(d:2,ERd)	0	1	7	0 1 d d	6	8	0	rd	1 0 0 0		1	0	1 1 0 0	0 0 0 0	
	SHAL.B #2, @(d:16,ERd)	0	1	7	0 1 0 0	6	E	0	rd	1 0 0 0	d	1	0	1 1 0 0	0 0 0 0	
	SHAL.B #2, @(d:32,ERd)	7	8	0	rd	0 1 0 0	6	A	0 0 1 0	1 0 0 0	d d	1	0	1 1 0 0	0 0 0 0	
	SHAL.B #2, @(d:16,Rd,B)	0	1	7	0 1 0 1	6	E	0	rd	1 0 0 0	d	1	0	1 1 0 0	0 0 0 0	
	SHAL.B #2, @(d:16,Rd,W)	0	1	7	0 1 1 0	6	E	0	rd	1 0 0 0	d	1	0	1 1 0 0	0 0 0 0	
	SHAL.B #2, @(d:16,ERd,L)	0	1	7	0 1 1 1	6	E	0	rd	1 0 0 0	d	1	0	1 1 0 0	0 0 0 0	
	SHAL.B #2, @(d:32,Rd,B)	7	8	0	rd	0 1 0 1	6	A	0 0 1 0	1 0 0 0	d d	1	0	1 1 0 0	0 0 0 0	
	SHAL.B #2, @(d:32,Rd,W)	7	8	0	rd	0 1 1 0	6	A	0 0 1 0	1 0 0 0	d d	1	0	1 1 0 0	0 0 0 0	
	SHAL.B #2, @(d:32,ERd,L)	7	8	0	rd	0 1 1 1	6	A	0 0 1 0	1 0 0 0	d d	1	0	1 1 0 0	0 0 0 0	
	SHAL.B #2, @aa:8					7	F	a a a a	a a a a		1	0	1 1 0 0	0 0 0 0		
	SHAL.B #2, @aa:16					6	A	0 0 0 1	1 0 0 0	a	1	0	1 1 0 0	0 0 0 0		
	SHAL.B #2, @aa:32					6	A	0 0 1 1	1 0 0 0	a a	1	0	1 1 0 0	0 0 0 0		
	SHAL.W Rd										1	0	1 0 0 1	rd		
	SHAL.W @ERd				7	D	1	rd	0 0 0 0		1	0	1 0 0 1	0 0 0 0		
	SHAL.W @ERd+	0	1	5	0 1 0 0	6	D	0	rd	1 0 0 0		1	0	1 0 0 1	0 0 0 0	
	SHAL.W @ERd-	0	1	5	0 1 1 0	6	D	0	rd	1 0 0 0		1	0	1 0 0 1	0 0 0 0	
	SHAL.W @+ERd	0	1	5	0 1 0 1	6	D	0	rd	1 0 0 0		1	0	1 0 0 1	0 0 0 0	
	SHAL.W @-ERd	0	1	5	0 1 1 1	6	D	0	rd	1 0 0 0		1	0	1 0 0 1	0 0 0 0	
	SHAL.W @(d:2,ERd)	0	1	5	0 1 d d	6	9	0	rd	1 0 0 0		1	0	1 0 0 1	0 0 0 0	
	SHAL.W @(d:16,ERd)	0	1	5	0 1 0 0	6	F	0	rd	1 0 0 0	d	1	0	1 0 0 1	0 0 0 0	
	SHAL.W @(d:32,ERd)	7	8	0	rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0	d d	1	0	1 0 0 1	0 0 0 0	
	SHAL.W @(d:16,Rd,B)	0	1	5	0 1 0 1	6	F	0	rd	1 0 0 0	d	1	0	1 0 0 1	0 0 0 0	
	SHAL.W @(d:16,Rd,W)	0	1	5	0 1 1 0	6	F	0	rd	1 0 0 0	d	1	0	1 0 0 1	0 0 0 0	
	SHAL.W @(d:16,ERd,L)	0	1	5	0 1 1 1	6	F	0	rd	1 0 0 0	d	1	0	1 0 0 1	0 0 0 0	
	SHAL.W @(d:32,Rd,B)	7	8	0	rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0	d d	1	0	1 0 0 1	0 0 0 0	
	SHAL.W @(d:32,Rd,W)	7	8	0	rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0	d d	1	0	1 0 0 1	0 0 0 0	
	SHAL.W @(d:32,ERd,L)	7	8	0	rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0	d d	1	0	1 0 0 1	0 0 0 0	
SHAL.W @aa:16					6	B	0 0 0 1	1 0 0 0	a	1	0	1 0 0 1	0 0 0 0			
SHAL.W @aa:32					6	B	0 0 1 1	1 0 0 0	a a	1	0	1 0 0 1	0 0 0 0			



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension														
		15	8	7	0	15	8	7	0		15	8	7	0															
SHAL	SHAL.W #2,Rd												1	0	1	1	0	1	rd										
	SHAL.W #2,@ERd					7	D	1	rd	0	0	0	0	1	0	1	1	0	0	0	0								
	SHAL.W #2,@ERd+	0	1	5	0	1	0	6	D	0	rd	1	0	0	0	1	0	1	1	0	0	0	0						
	SHAL.W #2,@ERd-	0	1	5	0	1	1	0	6	D	0	rd	1	0	0	0	1	0	1	0	0	0	0						
	SHAL.W #2,@+ERd	0	1	5	0	1	0	1	6	D	0	rd	1	0	0	0	1	0	0	1	0	0	0						
	SHAL.W #2,@-ERd	0	1	5	0	1	1	1	6	D	0	rd	1	0	0	0	1	0	1	0	0	0	0						
	SHAL.W #2,@(d:2,ERd)	0	1	5	0	1	d	d	6	9	0	rd	1	0	0	0	1	0	1	0	0	0	0						
	SHAL.W #2,@(d:16,ERd)	0	1	5	0	1	0	0	6	F	0	rd	1	0	0	0	d	1	0	1	0	0	0						
	SHAL.W #2,@(d:32,ERd)	7	8	0	rd	0	1	0	0	6	B	0	0	1	0	1	0	0	0	d	d	1	0	0	1	0	0	0	0
	SHAL.W #2,@(d:16,Rd,B)	0	1	5	0	1	0	1	6	F	0	rd	1	0	0	0	d	1	0	1	0	0	0	0	0	0	0	0	
	SHAL.W #2,@(d:16,Rd,W)	0	1	5	0	1	1	0	6	F	0	rd	1	0	0	0	d	1	0	1	0	0	0	0	0	0	0	0	
	SHAL.W #2,@(d:16,ERd,L)	0	1	5	0	1	1	1	6	F	0	rd	1	0	0	0	d	1	0	1	0	0	0	0	0	0	0	0	
	SHAL.W #2,@(d:32,Rd,B)	7	8	0	rd	0	1	0	1	6	B	0	0	1	0	1	0	0	0	d	d	1	0	0	1	0	0	0	0
	SHAL.W #2,@(d:32,Rd,W)	7	8	0	rd	0	1	1	0	6	B	0	0	1	0	1	0	0	0	d	d	1	0	0	1	0	0	0	0
	SHAL.W #2,@(d:32,ERd,L)	7	8	0	rd	0	1	1	1	6	B	0	0	1	0	1	0	0	0	d	d	1	0	0	1	0	0	0	0
	SHAL.W #2,@aa:16								6	B	0	0	0	0	1	1	0	0	0	a		1	0	0	1	0	0	0	0
	SHAL.W #2,@aa:32								6	B	0	0	1	1	1	1	0	0	0	a	a	1	0	0	1	0	0	0	0
	SHALL.ERd																												
	SHALL.@ERd	0	1	0	0	1	0	0	6	9	0	rd	1	0	0	0		1	0	1	0	1	1	0	0	0	0	0	0
	SHALL.@ERd+	0	1	0	0	1	0	0	6	D	0	rd	1	0	0	0		1	0	1	0	1	1	0	0	0	0	0	0
	SHALL.@ERd-	0	1	0	0	1	1	0	6	D	0	rd	1	0	0	0		1	0	1	0	1	1	0	0	0	0	0	0
	SHALL.@+ERd	0	1	0	0	1	0	1	6	D	0	rd	1	0	0	0		1	0	1	0	1	1	0	0	0	0	0	0
	SHALL.@-ERd	0	1	0	0	1	1	1	6	D	0	rd	1	0	0	0		1	0	1	0	1	1	0	0	0	0	0	0
	SHALL.@(d:2,ERd)	0	1	0	0	1	d	d	6	9	0	rd	1	0	0	0		1	0	1	0	1	1	0	0	0	0	0	0
SHALL.@(d:16,ERd)	0	1	0	0	1	0	0	6	F	0	rd	1	0	0	0	d		1	0	1	0	1	1	0	0	0	0	0	
SHALL.@(d:32,ERd)	7	8	1	rd	0	1	0	0	6	B	0	0	1	0	1	0	0	0	d	d	1	0	0	1	0	0	0	0	
SHALL.@(d:16,Rd,B)	0	1	0	0	1	0	1	6	F	0	rd	1	0	0	0	d		1	0	1	0	1	1	0	0	0	0	0	
SHALL.@(d:16,Rd,W)	0	1	0	0	1	1	0	6	F	0	rd	1	0	0	0	d		1	0	1	0	1	1	0	0	0	0	0	
SHALL.@(d:16,ERd,L)	0	1	0	0	1	1	1	6	F	0	rd	1	0	0	0	d		1	0	1	0	1	1	0	0	0	0	0	
SHALL.@(d:32,Rd,B)	7	8	1	rd	0	1	0	1	6	B	0	0	1	0	1	0	0	0	d	d	1	0	0	1	0	0	0	0	
SHALL.@(d:32,Rd,W)	7	8	1	rd	0	1	1	0	6	B	0	0	1	0	1	0	0	0	d	d	1	0	0	1	0	0	0	0	
SHALL.@(d:32,ERd,L)	7	8	1	rd	0	1	1	1	6	B	0	0	1	0	1	0	0	0	d	d	1	0	0	1	0	0	0	0	
SHALL.@aa:16	0	1	0	0	1	0	0	6	B	0	0	0	0	1	1	0	0	0	a		1	0	0	1	0	0	0	0	
SHALL.@aa:32	0	1	0	0	1	0	0	6	B	0	0	1	0	1	1	0	0	0	a	a	1	0	0	1	0	0	0	0	

Instruction	Mnemonic	Opcode						EA Ex- tension	Opcode						EA Extension													
		15	8	7	0	15	8		7	0	15	8	7	0														
SHAL	SHALL #2,ERd													1	0	1	1	1	1	0	rd							
	SHALL #2,@ERd	0	1	0	0	1	0	0	6	9	0	rd	1	0	0	0	0	0	0	0	0	0	0					
	SHALL #2,@ERd+	0	1	0	0	0	1	0	0	6	D	0	rd	1	0	0	0	0	0	0	0	0	0					
	SHALL #2,@ERd-	0	1	0	0	0	1	1	0	6	D	0	rd	1	0	0	0	0	0	0	0	0	0					
	SHALL #2,@+ERd	0	1	0	0	0	1	0	1	6	D	0	rd	1	0	0	0	0	0	0	0	0	0					
	SHALL #2,@-ERd	0	1	0	0	0	1	1	1	6	D	0	rd	1	0	0	0	0	0	0	0	0	0					
	SHALL #2,@(d2,ERd)	0	1	0	0	0	1	d	d	6	9	0	rd	1	0	0	0	0	0	0	0	0	0					
	SHALL #2,@(d:16,ERd)	0	1	0	0	0	1	0	0	6	F	0	rd	1	0	0	0	d		1	0	0	0	0				
	SHALL #2,@(d:32,ERd)	7	8	1	rd	0	1	0	0	6	B	0	0	1	0	1	0	0	d	d	1	0	0	0	0			
	SHALL #2,@(d:16,Rd,B)	0	1	0	0	0	1	0	1	6	F	0	rd	1	0	0	0	d		1	0	0	0	0	0			
	SHALL #2,@(d:16,Rd,W)	0	1	0	0	0	1	1	0	6	F	0	rd	1	0	0	0	d		1	0	0	0	0	0			
	SHALL #2,@(d:16,ERd,L)	0	1	0	0	0	1	1	1	6	F	0	rd	1	0	0	0	d		1	0	0	0	0	0			
	SHALL #2,@(d:32,Rd,B)	7	8	1	rd	0	1	0	1	6	B	0	0	1	0	1	0	0	d	d	1	0	0	0	0			
	SHALL #2,@(d:32,Rd,W)	7	8	1	rd	0	1	1	0	6	B	0	0	1	0	1	0	0	d	d	1	0	0	0	0			
SHALL #2,@(d:32,ERd,L)	7	8	1	rd	0	1	1	1	6	B	0	0	1	0	1	0	0	d	d	1	0	0	0	0				
SHALL #2,@aa:16	0	1	0	0	0	1	0	0	6	B	0	0	0	0	1	0	0	0	a		1	0	0	0	0			
SHALL #2,@aa:32	0	1	0	0	0	1	0	0	6	B	0	0	1	0	1	0	0	0	a	a	1	0	0	0	0			
SHAR	SHAR,B Rd													1	1	1	0	0	0	rd								
SHAR,B @ERd								7	D	0	rd	0	0	0	0			1	1	1	0	0	0	0	0	0		
SHAR,B @ERd+	0	1	7	0	1	0	0	6	C	0	rd	1	0	0	0			1	1	1	0	0	0	0	0	0		
SHAR,B @ERd-	0	1	7	0	1	1	0	6	C	0	rd	1	0	0	0			1	1	1	0	0	0	0	0	0		
SHAR,B @+ERd	0	1	7	0	1	0	1	6	C	0	rd	1	0	0	0			1	1	1	0	0	0	0	0	0		
SHAR,B @-ERd	0	1	7	0	1	1	1	6	C	0	rd	1	0	0	0			1	1	1	0	0	0	0	0	0		
SHAR,B @(d:2,ERd)	0	1	7	0	1	d	d	6	8	0	rd	1	0	0	0			1	1	1	0	0	0	0	0	0		
SHAR,B @(d:16,ERd)	0	1	7	0	1	0	0	6	E	0	rd	1	0	0	0	d		1	1	1	0	0	0	0	0	0		
SHAR,B @(d:32,ERd)	7	8	0	rd	0	1	0	0	6	A	0	0	1	0	1	0	0	d	d	1	1	1	0	0	0	0	0	0
SHAR,B @(d:16,Rd,B)	0	1	7	0	1	0	1	6	E	0	rd	1	0	0	0	d		1	1	1	0	0	0	0	0	0		
SHAR,B @(d:16,Rd,W)	0	1	7	0	1	1	0	6	E	0	rd	1	0	0	0	d		1	1	1	0	0	0	0	0	0		
SHAR,B @(d:16,ERd,L)	0	1	7	0	1	1	1	6	E	0	rd	1	0	0	0	d		1	1	1	0	0	0	0	0	0		
SHAR,B @(d:32,Rd,B)	7	8	0	rd	0	1	0	1	6	A	0	0	1	0	1	0	0	d	d	1	1	1	0	0	0	0	0	0
SHAR,B @(d:32,Rd,W)	7	8	0	rd	0	1	1	0	6	A	0	0	1	0	1	0	0	d	d	1	1	1	0	0	0	0	0	0
SHAR,B @(d:32,ERd,L)	7	8	0	rd	0	1	1	1	6	A	0	0	1	0	1	0	0	d	d	1	1	1	0	0	0	0	0	0
SHAR,B @aa:8								7	F	a	a	a	a	a	a	a	a			1	1	1	0	0	0	0	0	0
SHAR,B @aa:16								6	A	0	0	0	1	1	0	0	0	a		1	1	1	0	0	0	0	0	0



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension
		15	8	7	0	15	8	7	0		15	8	7	0	
SHAR	SHAR.B @aa:32					6	A	0 0 1 1	1 0 0 0 0	a a	1	1	1 0 0 0	0 0 0 0	
	SHAR.B #2,Rd										1	1	1 1 0 0	rd	
	SHAR.B #2,@ERd					7	D	0 rd	0 0 0 0 0		1	1	1 1 0 0	0 0 0 0	
	SHAR.B #2,@ERd+	0	1	7	0 1 0 0	6	C	0 rd	1 0 0 0 0		1	1	1 1 0 0	0 0 0 0	
	SHAR.B #2,@ERd-	0	1	7	0 1 1 0	6	C	0 rd	1 0 0 0 0		1	1	1 1 0 0	0 0 0 0	
	SHAR.B #2,@+ERd	0	1	7	0 1 0 1	6	C	0 rd	1 0 0 0 0		1	1	1 1 0 0	0 0 0 0	
	SHAR.B #2,@-ERd	0	1	7	0 1 1 1	6	C	0 rd	1 0 0 0 0		1	1	1 1 0 0	0 0 0 0	
	SHAR.B #2,@(d:2,ERd)	0	1	7	0 1 d d	6	8	0 rd	1 0 0 0 0		1	1	1 1 0 0	0 0 0 0	
	SHAR.B #2,@(d:16,ERd)	0	1	7	0 1 0 0	6	E	0 rd	1 0 0 0 0	d	1	1	1 1 0 0	0 0 0 0	
	SHAR.B #2,@(d:32,ERd)	7	8	0	rd 0 1 0 0	6	A	0 0 1 0	1 0 0 0 0	d d	1	1	1 1 0 0	0 0 0 0	
	SHAR.B #2,@(d:16,Rd,B)	0	1	7	0 1 0 1	6	E	0 rd	1 0 0 0 0	d	1	1	1 1 0 0	0 0 0 0	
	SHAR.B #2,@(d:16,Rd,W)	0	1	7	0 1 1 0	6	E	0 rd	1 0 0 0 0	d	1	1	1 1 0 0	0 0 0 0	
	SHAR.B #2,@(d:16,ERd,L)	0	1	7	0 1 1 1	6	E	0 rd	1 0 0 0 0	d	1	1	1 1 0 0	0 0 0 0	
	SHAR.B #2,@(d:32,Rd,B)	7	8	0	rd 0 1 0 1	6	A	0 0 1 0	1 0 0 0 0	d d	1	1	1 1 0 0	0 0 0 0	
	SHAR.B #2,@(d:32,Rd,W)	7	8	0	rd 0 1 1 0	6	A	0 0 1 0	1 0 0 0 0	d d	1	1	1 1 0 0	0 0 0 0	
	SHAR.B #2,@(d:32,ERd,L)	7	8	0	rd 0 1 1 1	6	A	0 0 1 0	1 0 0 0 0	d d	1	1	1 1 0 0	0 0 0 0	
	SHAR.B #2,@aa:8					7	F	a a a a	a a a a		1	1	1 1 0 0	0 0 0 0	
	SHAR.B #2,@aa:16					6	A	0 0 0 1	1 0 0 0 0	a	1	1	1 1 0 0	0 0 0 0	
	SHAR.B #2,@aa:32					6	A	0 0 1 1	1 0 0 0 0	a a	1	1	1 1 0 0	0 0 0 0	
	SHAR.W Rd										1	1	1 0 0 1	rd	
	SHAR.W @ERd					7	D	1 rd	0 0 0 0 0		1	1	1 0 0 1	0 0 0 0	
	SHAR.W @ERd+	0	1	5	0 1 0 0	6	D	0 rd	1 0 0 0 0		1	1	1 0 0 1	0 0 0 0	
	SHAR.W @ERd-	0	1	5	0 1 1 0	6	D	0 rd	1 0 0 0 0		1	1	1 0 0 1	0 0 0 0	
SHAR.W @+ERd	0	1	5	0 1 0 1	6	D	0 rd	1 0 0 0 0		1	1	1 0 0 1	0 0 0 0		
SHAR.W @-ERd	0	1	5	0 1 1 1	6	D	0 rd	1 0 0 0 0		1	1	1 0 0 1	0 0 0 0		
SHAR.W @(d:2,ERd)	0	1	5	0 1 d d	6	9	0 rd	1 0 0 0 0		1	1	1 0 0 1	0 0 0 0		
SHAR.W @(d:16,ERd)	0	1	5	0 1 0 0	6	F	0 rd	1 0 0 0 0	d	1	1	1 0 0 1	0 0 0 0		
SHAR.W @(d:32,ERd)	7	8	0	rd 0 1 0 0	6	B	0 0 1 0	1 0 0 0 0	d d	1	1	1 0 0 1	0 0 0 0		
SHAR.W @(d:16,Rd,B)	0	1	5	0 1 0 1	6	F	0 rd	1 0 0 0 0	d	1	1	1 0 0 1	0 0 0 0		
SHAR.W @(d:16,Rd,W)	0	1	5	0 1 1 0	6	F	0 rd	1 0 0 0 0	d	1	1	1 0 0 1	0 0 0 0		
SHAR.W @(d:16,ERd,L)	0	1	5	0 1 1 1	6	F	0 rd	1 0 0 0 0	d	1	1	1 0 0 1	0 0 0 0		
SHAR.W @(d:32,Rd,B)	7	8	0	rd 0 1 0 1	6	B	0 0 1 0	1 0 0 0 0	d d	1	1	1 0 0 1	0 0 0 0		
SHAR.W @(d:32,Rd,W)	7	8	0	rd 0 1 1 0	6	B	0 0 1 0	1 0 0 0 0	d d	1	1	1 0 0 1	0 0 0 0		
SHAR.W @(d:32,ERd,L)	7	8	0	rd 0 1 1 1	6	B	0 0 1 0	1 0 0 0 0	d d	1	1	1 0 0 1	0 0 0 0		

Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
SHAR	SHAR.W @aa:16					6	B	0 0 0 1	1 0 0 0 0	a		1	1	1 0 0 1	0 0 0 0 0	
	SHAR.W @aa:32					6	B	0 0 1 1	1 0 0 0 0	a a		1	1	1 0 0 1	0 0 0 0 0	
	SHAR.W #2,Rd											1	1	1 1 0 1	rd	
	SHAR.W #2,@ERd					7	D	1 rd	0 0 0 0 0			1	1	1 1 0 1	0 0 0 0 0	
	SHAR.W #2,@ERd+	0	1	5	0 1 0 0	6	D	0 rd	1 0 0 0 0			1	1	1 1 0 1	0 0 0 0 0	
	SHAR.W #2,@ERd-	0	1	5	0 1 1 0	6	D	0 rd	1 0 0 0 0			1	1	1 1 0 1	0 0 0 0 0	
	SHAR.W #2,@+ERd	0	1	5	0 1 0 1	6	D	0 rd	1 0 0 0 0			1	1	1 1 0 1	0 0 0 0 0	
	SHAR.W #2,@-ERd	0	1	5	0 1 1 1	6	D	0 rd	1 0 0 0 0			1	1	1 1 0 1	0 0 0 0 0	
	SHAR.W #2,@(d:2,ERd)	0	1	5	0 1 d d	6	9	0 rd	1 0 0 0 0			1	1	1 1 0 1	0 0 0 0 0	
	SHAR.W #2,@(d:16,ERd)	0	1	5	0 1 0 0	6	F	0 rd	1 0 0 0 0	d		1	1	1 1 0 1	0 0 0 0 0	
	SHAR.W #2,@(d:32,ERd)	7	8	0	rd 0 1 0 0	6	B	0 0 1 0	1 0 0 0 0	d d		1	1	1 1 0 1	0 0 0 0 0	
	SHAR.W #2,@(d:16,Rd,B)	0	1	5	0 1 0 1	6	F	0 rd	1 0 0 0 0	d		1	1	1 1 0 1	0 0 0 0 0	
	SHAR.W #2,@(d:16,Rd,W)	0	1	5	0 1 1 0	6	F	0 rd	1 0 0 0 0	d		1	1	1 1 0 1	0 0 0 0 0	
	SHAR.W #2,@(d:16,ERd,L)	0	1	5	0 1 1 1	6	F	0 rd	1 0 0 0 0	d		1	1	1 1 0 1	0 0 0 0 0	
	SHAR.W #2,@(d:32,Rd,B)	7	8	0	rd 0 1 0 1	6	B	0 0 1 0	1 0 0 0 0	d d		1	1	1 1 0 1	0 0 0 0 0	
	SHAR.W #2,@(d:32,Rd,W)	7	8	0	rd 0 1 1 0	6	B	0 0 1 0	1 0 0 0 0	d d		1	1	1 1 0 1	0 0 0 0 0	
	SHAR.W #2,@(d:32,ERd,L)	7	8	0	rd 0 1 1 1	6	B	0 0 1 0	1 0 0 0 0	d d		1	1	1 1 0 1	0 0 0 0 0	
	SHAR.W #2,@aa:16					6	B	0 0 0 1	1 0 0 0 0	a		1	1	1 1 0 1	0 0 0 0 0	
	SHAR.W #2,@aa:32					6	B	0 0 1 1	1 0 0 0 0	a a		1	1	1 1 0 1	0 0 0 0 0	
	SHAR.L ERd											1	1	1 0 1 1	0 rd	
	SHAR.L @ERd	0	1	0	0 1 0 0	6	9	0 rd	1 0 0 0 0			1	1	1 0 1 1	0 0 0 0 0	
	SHAR.L @ERd+	0	1	0	0 1 0 0	6	D	0 rd	1 0 0 0 0			1	1	1 0 1 1	0 0 0 0 0	
	SHAR.L @ERd-	0	1	0	0 1 1 0	6	D	0 rd	1 0 0 0 0			1	1	1 0 1 1	0 0 0 0 0	
	SHAR.L @+ERd	0	1	0	0 1 0 1	6	D	0 rd	1 0 0 0 0			1	1	1 0 1 1	0 0 0 0 0	
	SHAR.L @-ERd	0	1	0	0 1 1 1	6	D	0 rd	1 0 0 0 0			1	1	1 0 1 1	0 0 0 0 0	
	SHAR.L @(d:2,ERd)	0	1	0	0 1 d d	6	9	0 rd	1 0 0 0 0			1	1	1 0 1 1	0 0 0 0 0	
	SHAR.L @(d:16,ERd)	0	1	0	0 1 0 0	6	F	0 rd	1 0 0 0 0	d		1	1	1 0 1 1	0 0 0 0 0	
	SHAR.L @(d:32,ERd)	7	8	1	rd 0 1 0 0	6	B	0 0 1 0	1 0 0 0 0	d d		1	1	1 0 1 1	0 0 0 0 0	
	SHAR.L @(d:16,Rd,B)	0	1	0	0 1 0 1	6	F	0 rd	1 0 0 0 0	d		1	1	1 0 1 1	0 0 0 0 0	
	SHAR.L @(d:16,Rd,W)	0	1	0	0 1 1 0	6	F	0 rd	1 0 0 0 0	d		1	1	1 0 1 1	0 0 0 0 0	
	SHAR.L @(d:16,ERd,L)	0	1	0	0 1 1 1	6	F	0 rd	1 0 0 0 0	d		1	1	1 0 1 1	0 0 0 0 0	
	SHAR.L @(d:32,Rd,B)	7	8	1	rd 0 1 0 1	6	B	0 0 1 0	1 0 0 0 0	d d		1	1	1 0 1 1	0 0 0 0 0	
SHAR.L @(d:32,Rd,W)	7	8	1	rd 0 1 1 0	6	B	0 0 1 0	1 0 0 0 0	d d		1	1	1 0 1 1	0 0 0 0 0		
SHAR.L @(d:32,ERd,L)	7	8	1	rd 0 1 1 1	6	B	0 0 1 0	1 0 0 0 0	d d		1	1	1 0 1 1	0 0 0 0 0		



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension
		15	8	7	0	15	8	7	0		15	8	7	0	
SHARL	@aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0	1 0 0 0	a	1	1	1 0 1 1	0 0 0 0	
	@aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0	1 0 0 0	a a	1	1	1 0 1 1	0 0 0 0	
	#2,ERd										1	1	1 1 1 1	0 rd	
	#2,@ERd	0	1	0	0 1 0 0	6	9	0 rd	1 0 0 0		1	1	1 1 1 1	0 0 0 0	
	#2,@ERd+	0	1	0	0 1 0 0	6	D	0 rd	1 0 0 0		1	1	1 1 1 1	0 0 0 0	
	#2,@ERd-	0	1	0	0 1 1 0	6	D	0 rd	1 0 0 0		1	1	1 1 1 1	0 0 0 0	
	#2,@+ERd	0	1	0	0 1 0 1	6	D	0 rd	1 0 0 0		1	1	1 1 1 1	0 0 0 0	
	#2,@-ERd	0	1	0	0 1 1 1	6	D	0 rd	1 0 0 0		1	1	1 1 1 1	0 0 0 0	
	#2,@(d:2,ERd)	0	1	0	0 1 d d	6	9	0 rd	1 0 0 0		1	1	1 1 1 1	0 0 0 0	
	#2,@(d:16,ERd)	0	1	0	0 1 0 0	6	F	0 rd	1 0 0 0	d	1	1	1 1 1 1	0 0 0 0	
	#2,@(d:32,ERd)	7	8	1	rd 0 1 0 0	6	B	0 0 1 0	1 0 0 0	d d	1	1	1 1 1 1	0 0 0 0	
	#2,@(d:16,Rd,B)	0	1	0	0 1 0 1	6	F	0 rd	1 0 0 0	d	1	1	1 1 1 1	0 0 0 0	
	#2,@(d:16,Rd,W)	0	1	0	0 1 1 0	6	F	0 rd	1 0 0 0	d	1	1	1 1 1 1	0 0 0 0	
	#2,@(d:16,ERd,L)	0	1	0	0 1 1 1	6	F	0 rd	1 0 0 0	d	1	1	1 1 1 1	0 0 0 0	
	#2,@(d:32,Rd,B)	7	8	1	rd 0 1 0 1	6	B	0 0 1 0	1 0 0 0	d d	1	1	1 1 1 1	0 0 0 0	
	#2,@(d:32,Rd,W)	7	8	1	rd 0 1 1 0	6	B	0 0 1 0	1 0 0 0	d d	1	1	1 1 1 1	0 0 0 0	
	#2,@(d:32,ERd,L)	7	8	1	rd 0 1 1 1	6	B	0 0 1 0	1 0 0 0	d d	1	1	1 1 1 1	0 0 0 0	
	#2,@aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0	1 0 0 0	a	1	1	1 1 1 1	0 0 0 0	
#2,@aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0	1 0 0 0	a a	1	1	1 1 1 1	0 0 0 0		
SHLL	#xx:5,Rd					0	3	1 0 0 x	x x x x		1	0	0 0 0 0	rd	
	Rn,Rd					7	8	m	1 0 0 0		1	0	0 0 0 0	rd	
	Rd										1	0	0 0 0 0	rd	
	@ERd					7	D	0 rd	0 0 0 0		1	0	0 0 0 0	0 0 0 0	
	@ERd+	0	1	7	0 1 0 0	6	C	0 rd	1 0 0 0		1	0	0 0 0 0	0 0 0 0	
	@ERd-	0	1	7	0 1 1 0	6	C	0 rd	1 0 0 0		1	0	0 0 0 0	0 0 0 0	
	@+ERd	0	1	7	0 1 0 1	6	C	0 rd	1 0 0 0		1	0	0 0 0 0	0 0 0 0	
	@-ERd	0	1	7	0 1 1 1	6	C	0 rd	1 0 0 0		1	0	0 0 0 0	0 0 0 0	
	@(d:2,ERd)	0	1	7	0 1 d d	6	8	0 rd	1 0 0 0		1	0	0 0 0 0	0 0 0 0	
	@(d:16,ERd)	0	1	7	0 1 0 0	6	E	0 rd	1 0 0 0	d	1	0	0 0 0 0	0 0 0 0	
	@(d:32,ERd)	7	8	0	rd 0 1 0 0	6	A	0 0 1 0	1 0 0 0	d d	1	0	0 0 0 0	0 0 0 0	
	@(d:16,Rd,B)	0	1	7	0 1 0 1	6	E	0 rd	1 0 0 0	d	1	0	0 0 0 0	0 0 0 0	
@(d:16,Rd,W)	0	1	7	0 1 1 0	6	E	0 rd	1 0 0 0	d	1	0	0 0 0 0	0 0 0 0		
@(d:16,ERd,L)	0	1	7	0 1 1 1	6	E	0 rd	1 0 0 0	d	1	0	0 0 0 0	0 0 0 0		
@(d:32,Rd,B)	7	8	0	rd 0 1 0 1	6	A	0 0 1 0	1 0 0 0	d d	1	0	0 0 0 0	0 0 0 0		

Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension
		15	8	7	0	15	8		7	0	15	8	7	0	
SHLL	SHLL.B @(d:32,Rd,W)	7	8	0	rd	0 1 1 0	6	A	0 0 1 0	1 0 0 0	d d	1	0	0 0 0 0	0 0 0 0
	SHLL.B @(d:32,ERd,L)	7	8	0	rd	0 1 1 1	6	A	0 0 1 0	1 0 0 0	d d	1	0	0 0 0 0	0 0 0 0
	SHLL.B @aa:8						7	F	a a a a	a a a a		1	0	0 0 0 0	0 0 0 0
	SHLL.B @aa:16						6	A	0 0 0 0	1 0 0 0	a	1	0	0 0 0 0	0 0 0 0
	SHLL.B @aa:32						6	A	0 0 1 1	1 0 0 0	a a	1	0	0 0 0 0	0 0 0 0
	SHLL.B #2,Rd											1	0	0 1 0 0	rd
	SHLL.B #2,@ERd						7	D	0 rd	0 0 0 0		1	0	0 1 0 0	0 0 0 0
	SHLL.B #2,@ERd+	0	1	7	0 1 0 0	6	C	0 rd	1 0 0 0		1	0	0 1 0 0	0 0 0 0	
	SHLL.B #2,@ERd-	0	1	7	0 1 1 0	6	C	0 rd	1 0 0 0		1	0	0 1 0 0	0 0 0 0	
	SHLL.B #2,@+ERd	0	1	7	0 1 0 1	6	C	0 rd	1 0 0 0		1	0	0 1 0 0	0 0 0 0	
	SHLL.B #2,@-ERd	0	1	7	0 1 1 1	6	C	0 rd	1 0 0 0		1	0	0 1 0 0	0 0 0 0	
	SHLL.B #2,@(d2,ERd)	0	1	7	0 1 d d	6	8	0 rd	1 0 0 0		1	0	0 1 0 0	0 0 0 0	
	SHLL.B #2,@(d:16,ERd)	0	1	7	0 1 0 0	6	E	0 rd	1 0 0 0	d	1	0	0 1 0 0	0 0 0 0	
	SHLL.B #2,@(d:32,ERd)	7	8	0	rd	0 1 0 0	6	A	0 0 1 0	1 0 0 0	d d	1	0	0 1 0 0	0 0 0 0
	SHLL.B #2,@(d:16,Rd,B)	0	1	7	0 1 0 1	6	E	0 rd	1 0 0 0	d	1	0	0 1 0 0	0 0 0 0	
	SHLL.B #2,@(d:16,Rd,W)	0	1	7	0 1 1 0	6	E	0 rd	1 0 0 0	d	1	0	0 1 0 0	0 0 0 0	
	SHLL.B #2,@(d:16,ERd,L)	0	1	7	0 1 1 1	6	E	0 rd	1 0 0 0	d	1	0	0 1 0 0	0 0 0 0	
	SHLL.B #2,@(d:32,Rd,B)	7	8	0	rd	0 1 0 1	6	A	0 0 1 0	1 0 0 0	d d	1	0	0 1 0 0	0 0 0 0
	SHLL.B #2,@(d:32,Rd,W)	7	8	0	rd	0 1 1 0	6	A	0 0 1 0	1 0 0 0	d d	1	0	0 1 0 0	0 0 0 0
	SHLL.B #2,@(d:32,ERd,L)	7	8	0	rd	0 1 1 1	6	A	0 0 1 0	1 0 0 0	d d	1	0	0 1 0 0	0 0 0 0
	SHLL.B #2,@aa:8						7	F	a a a a	a a a a		1	0	0 1 0 0	0 0 0 0
	SHLL.B #2,@aa:16						6	A	0 0 0 0	1 0 0 0	a	1	0	0 1 0 0	0 0 0 0
	SHLL.B #2,@aa:32						6	A	0 0 1 1	1 0 0 0	a a	1	0	0 1 0 0	0 0 0 0
	SHLL.B #4,Rd											1	0	1 0 1 0	rd
	SHLL.B #4,@ERd						7	D	0 rd	0 0 0 0		1	0	1 0 1 0	0 0 0 0
	SHLL.B #4,@ERd+	0	1	7	0 1 0 0	6	C	0 rd	1 0 0 0		1	0	1 0 1 0	0 0 0 0	
	SHLL.B #4,@ERd-	0	1	7	0 1 1 0	6	C	0 rd	1 0 0 0		1	0	1 0 1 0	0 0 0 0	
	SHLL.B #4,@+ERd	0	1	7	0 1 0 1	6	C	0 rd	1 0 0 0		1	0	1 0 1 0	0 0 0 0	
	SHLL.B #4,@-ERd	0	1	7	0 1 1 1	6	C	0 rd	1 0 0 0		1	0	1 0 1 0	0 0 0 0	
	SHLL.B #4,@(d2,ERd)	0	1	7	0 1 d d	6	8	0 rd	1 0 0 0		1	0	1 0 1 0	0 0 0 0	
	SHLL.B #4,@(d:16,ERd)	0	1	7	0 1 0 0	6	E	0 rd	1 0 0 0	d	1	0	1 0 1 0	0 0 0 0	
	SHLL.B #4,@(d:32,ERd)	7	8	0	rd	0 1 0 0	6	A	0 0 1 0	1 0 0 0	d d	1	0	1 0 1 0	0 0 0 0
SHLL.B #4,@(d:16,Rd,B)	0	1	7	0 1 0 1	6	E	0 rd	1 0 0 0	d	1	0	1 0 1 0	0 0 0 0		
SHLL.B #4,@(d:16,Rd,W)	0	1	7	0 1 1 0	6	E	0 rd	1 0 0 0	d	1	0	1 0 1 0	0 0 0 0		

Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
SHLL	SHLL B #4,@(d:16,ERd.L)	0	1	7	0 1 1 1	6	E	0	rd	1 0 0 0	d	1	0	1 0 1 0	0 0 0 0	
	SHLL B #4,@(d:32,Rd.B)	7	8	0	rd	6	A	0 0 1 0	1 0 0 0	d d	1	0	1 0 1 0	0 0 0 0		
	SHLL B #4,@(d:32,Rd.W)	7	8	0	rd	6	A	0 0 1 0	1 0 0 0	d d	1	0	1 0 1 0	0 0 0 0		
	SHLL B #4,@(d:32,ERd.L)	7	8	0	rd	6	A	0 0 1 0	1 0 0 0	d d	1	0	1 0 1 0	0 0 0 0		
	SHLL B #4,@aa:8					7	F	a a a a	a a a a		1	0	1 0 1 0	0 0 0 0		
	SHLL B #4,@aa:16					6	A	0 0 0 1	1 0 0 0	a	1	0	1 0 1 0	0 0 0 0		
	SHLL B #4,@aa:32					6	A	0 0 1 1	1 0 0 0	a a	1	0	1 0 1 0	0 0 0 0		
	SHLL W #xx:5,Rd					0	3	1 0 0 x	x x x x		1	0	0 0 0 1	rd		
	SHLL W Rn,Rd					7	8	m	1 0 0 0		1	0	0 0 0 1	rd		
	SHLL W Rd										1	0	0 0 0 1	rd		
	SHLL W @ERd					7	D	1	rd	0 0 0 0		1	0	0 0 0 1	0 0 0 0	
	SHLL W @ERd+	0	1	5	0 1 0 0	6	D	0	rd	1 0 0 0		1	0	0 0 0 1	0 0 0 0	
	SHLL W @ERd-	0	1	5	0 1 1 0	6	D	0	rd	1 0 0 0		1	0	0 0 0 1	0 0 0 0	
	SHLL W @+ERd	0	1	5	0 1 0 1	6	D	0	rd	1 0 0 0		1	0	0 0 0 1	0 0 0 0	
	SHLL W @-ERd	0	1	5	0 1 1 1	6	D	0	rd	1 0 0 0		1	0	0 0 0 1	0 0 0 0	
	SHLL W @(d:2,ERd)	0	1	5	0 1 d d	6	9	0	rd	1 0 0 0		1	0	0 0 0 1	0 0 0 0	
	SHLL W @(d:16,ERd)	0	1	5	0 1 0 0	6	F	0	rd	1 0 0 0	d	1	0	0 0 0 1	0 0 0 0	
	SHLL W @(d:32,ERd)	7	8	0	rd	6	B	0 0 1 0	1 0 0 0	d d	1	0	0 0 0 1	0 0 0 0		
	SHLL W @(d:16,Rd.B)	0	1	5	0 1 0 1	6	F	0	rd	1 0 0 0	d	1	0	0 0 0 1	0 0 0 0	
	SHLL W @(d:16,Rd.W)	0	1	5	0 1 1 0	6	F	0	rd	1 0 0 0	d	1	0	0 0 0 1	0 0 0 0	
	SHLL W @(d:16,ERd.L)	0	1	5	0 1 1 1	6	F	0	rd	1 0 0 0	d	1	0	0 0 0 1	0 0 0 0	
	SHLL W @(d:32,Rd.B)	7	8	0	rd	6	B	0 0 1 0	1 0 0 0	d d	1	0	0 0 0 1	0 0 0 0		
	SHLL W @(d:32,Rd.W)	7	8	0	rd	6	B	0 0 1 0	1 0 0 0	d d	1	0	0 0 0 1	0 0 0 0		
	SHLL W @(d:32,ERd.L)	7	8	0	rd	6	B	0 0 1 0	1 0 0 0	d d	1	0	0 0 0 1	0 0 0 0		
	SHLL W @aa:16					6	B	0 0 0 1	1 0 0 0	a	1	0	0 0 0 1	0 0 0 0		
	SHLL W @aa:32					6	B	0 0 1 1	1 0 0 0	a a	1	0	0 0 0 1	0 0 0 0		
	SHLL W #2,Rd										1	0	0 1 0 1	rd		
	SHLL W #2,@ERd					7	D	1	rd	0 0 0 0		1	0	0 1 0 1	0 0 0 0	
SHLL W #2,@ERd+	0	1	5	0 1 0 0	6	D	0	rd	1 0 0 0		1	0	0 1 0 1	0 0 0 0		
SHLL W #2,@ERd-	0	1	5	0 1 1 0	6	D	0	rd	1 0 0 0		1	0	0 1 0 1	0 0 0 0		
SHLL W #2,@+ERd	0	1	5	0 1 0 1	6	D	0	rd	1 0 0 0		1	0	0 1 0 1	0 0 0 0		
SHLL W #2,@-ERd	0	1	5	0 1 1 1	6	D	0	rd	1 0 0 0		1	0	0 1 0 1	0 0 0 0		
SHLL W #2,@(d:2,ERd)	0	1	5	0 1 d d	6	9	0	rd	1 0 0 0		1	0	0 1 0 1	0 0 0 0		
SHLL W #2,@(d:16,ERd)	0	1	5	0 1 0 0	6	F	0	rd	1 0 0 0	d	1	0	0 1 0 1	0 0 0 0		

Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension																									
		15	8	7	0	15	8	7	0		15	8	7	0																										
SHLL	SHLL.W #2, @(d:32,ERd)	7	8	0	rd	0	1	0	0	6	B	0	0	1	0	1	0	0	0	d	d	1	0	0	1	0	0	0	0	0	0									
	SHLL.W #2, @(d:16,Rd.B)	0	1		5	0	1	0	1	6	F	0	rd	1	0	0	0	0	d			1	0	0	1	0	1	0	0	0	0	0	0							
	SHLL.W #2, @(d:16,Rd.W)	0	1		5	0	1	1	1	6	F	0	rd	1	0	0	0	0	d			1	0	0	1	0	1	0	0	0	0	0	0	0						
	SHLL.W #2, @(d:16,ERd.L)	0	1		5	0	1	1	1	6	F	0	rd	1	0	0	0	0	d			1	0	0	1	0	1	0	0	0	0	0	0	0	0					
	SHLL.W #2, @(d:32,Rd.B)	7	8	0	rd	0	1	0	1	6	B	0	0	1	0	1	0	0	0	d	d	1	0	0	1	0	1	0	0	0	0	0	0	0	0					
	SHLL.W #2, @(d:32,Rd.W)	7	8	0	rd	0	1	1	0	6	B	0	0	1	0	1	0	0	0	d	d	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0				
	SHLL.W #2, @(d:32,ERd.L)	7	8	0	rd	0	1	1	1	6	B	0	0	1	0	1	0	0	0	d	d	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0				
	SHLL.W #2, @aa:16									6	B	0	0	0	1	1	0	0	0	a		1	0	0	1	0	1	0	0	0	0	0	0	0	0	0				
	SHLL.W #2, @aa:32									6	B	0	0	1	1	1	0	0	0	a	a	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0			
	SHLL.W #4,Rd																					1	0	0	0	1	0	rd												
	SHLL.W #4, @ERd									7	D	1	rd	0	0	0	0	0				1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0		
	SHLL.W #4, @ERd+	0	1		5	0	1	0	0	6	D	0	rd	1	0	0	0	0				1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0		
	SHLL.W #4, @ERd-	0	1		5	0	1	1	0	6	D	0	rd	1	0	0	0	0				1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0		
	SHLL.W #4, @+ERd	0	1		5	0	1	0	1	6	D	0	rd	1	0	0	0	0				1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
	SHLL.W #4, @-ERd	0	1		5	0	1	1	1	6	D	0	rd	1	0	0	0	0				1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
	SHLL.W #4, @(d:2,ERd)	0	1		5	0	1	d	d	6	9	0	rd	1	0	0	0	0				1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0		
	SHLL.W #4, @(d:16,ERd)	0	1		5	0	1	0	0	6	F	0	rd	1	0	0	0	0	d			1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
	SHLL.W #4, @(d:32,ERd)	7	8	0	rd	0	1	0	0	6	B	0	0	1	0	1	0	0	0	d	d	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
	SHLL.W #4, @(d:16,Rd.B)	0	1		5	0	1	0	1	6	F	0	rd	1	0	0	0	0	d			1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
	SHLL.W #4, @(d:16,Rd.W)	0	1		5	0	1	1	0	6	F	0	rd	1	0	0	0	0	d			1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
	SHLL.W #4, @(d:16,ERd.L)	0	1		5	0	1	1	1	6	F	0	rd	1	0	0	0	0	d			1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
	SHLL.W #4, @(d:32,Rd.B)	7	8	0	rd	0	1	0	1	6	B	0	0	1	0	1	0	0	0	d	d	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
	SHLL.W #4, @(d:32,Rd.W)	7	8	0	rd	0	1	1	0	6	B	0	0	1	0	1	0	0	0	d	d	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
	SHLL.W #4, @(d:32,ERd.L)	7	8	0	rd	0	1	1	1	6	B	0	0	1	0	1	0	0	0	d	d	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
	SHLL.W #4, @aa:16									6	B	0	0	0	1	1	0	0	0	a		1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
	SHLL.W #4, @aa:32									6	B	0	0	1	1	1	0	0	0	a	a	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
	SHLL.W #8,Rd																					1	0	0	0	1	1	0	rd											
	SHLL.W #8, @ERd									7	D	1	rd	0	0	0	0	0				1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	
	SHLL.W #8, @ERd+	0	1		5	0	1	0	0	6	D	0	rd	1	0	0	0	0				1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
	SHLL.W #8, @ERd-	0	1		5	0	1	1	0	6	D	0	rd	1	0	0	0	0				1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	SHLL.W #8, @+ERd	0	1		5	0	1	0	1	6	D	0	rd	1	0	0	0	0				1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	SHLL.W #8, @-ERd	0	1		5	0	1	1	1	6	D	0	rd	1	0	0	0	0				1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
SHLL.W #8, @(d:2,ERd)	0	1		5	0	1	d	d	6	9	0	rd	1	0	0	0	0				1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
SHLL.W #8, @(d:16,ERd)	0	1		5	0	1	0	0	6	F	0	rd	1	0	0	0	0	d			1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	



Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension																		
		15	8	7	0	15	8	7	0		15	8	7	0																			
SHLL	SHLL.W #8, @(d:32,ERd)	7	8	0	rd	0	1	0	0	6	B	0	0	1	0	1	0	0	0	d	d	1	0	0	1	1	0	0	0	0	0	0	
	SHLL.W #8, @(d:16,Rd.B)	0	1		5	0	1	0	1	6	F	0	rd	1	0	0	0	0	d			1	0	0	1	1	0	0	0	0	0	0	
	SHLL.W #8, @(d:16,Rd.W)	0	1		5	0	1	1	0	6	F	0	rd	1	0	0	0	d			1	0	0	1	1	0	0	0	0	0	0	0	
	SHLL.W #8, @(d:16,ERd.L)	0	1		5	0	1	1	1	6	F	0	rd	1	0	0	0	d			1	0	0	1	1	0	0	0	0	0	0	0	
	SHLL.W #8, @(d:32,Rd.B)	7	8	0	rd	0	1	0	1	6	B	0	0	1	0	1	0	0	d	d		1	0	0	1	1	0	0	0	0	0	0	
	SHLL.W #8, @(d:32,Rd.W)	7	8	0	rd	0	1	1	0	6	B	0	0	1	0	1	0	0	d	d		1	0	0	1	1	0	0	0	0	0	0	
	SHLL.W #8, @(d:32,ERd.L)	7	8	0	rd	0	1	1	1	6	B	0	0	1	0	1	0	0	d	d		1	0	0	1	1	0	0	0	0	0	0	
	SHLL.W #8, @aa:16									6	B	0	0	0	1	1	0	0	a			1	0	0	1	1	0	0	0	0	0	0	
	SHLL.W #8, @aa:32									6	B	0	0	1	1	1	0	0	a	a		1	0	0	1	1	0	0	0	0	0	0	
	SHLLL #xx:5,ERd									0	3	1	0	0	x	x	x	x				1	0	0	0	1	1	0	rd				
	SHLLL Rn,ERd									7	8		m	1	0	0	0	0				1	0	0	0	1	1	0	rd				
	SHLLL ERd																					1	0	0	0	1	1	0	rd				
	SHLLL @ERd	0	1		0	0	1	0	0	6	9	0	rd	1	0	0	0					1	0	0	0	1	1	0	0	0	0	0	0
	SHLLL @ERd+	0	1		0	0	1	0	0	6	D	0	rd	1	0	0	0					1	0	0	0	1	1	0	0	0	0	0	0
	SHLLL @ERd-	0	1		0	0	1	1	0	6	D	0	rd	1	0	0	0					1	0	0	0	1	1	0	0	0	0	0	0
	SHLLL @+ERd	0	1		0	0	1	0	1	6	D	0	rd	1	0	0	0					1	0	0	0	1	1	0	0	0	0	0	0
	SHLLL @-ERd	0	1		0	0	1	1	1	6	D	0	rd	1	0	0	0					1	0	0	0	1	1	0	0	0	0	0	0
	SHLLL @(d:2,ERd)	0	1		0	0	1	d	d	6	9	0	rd	1	0	0	0					1	0	0	0	1	1	0	0	0	0	0	0
	SHLLL @(d:16,ERd)	0	1		0	0	1	0	0	6	F	0	rd	1	0	0	0	d				1	0	0	0	1	1	0	0	0	0	0	0
	SHLLL @(d:32,ERd)	7	8	1	rd	0	1	0	0	6	B	0	0	1	0	1	0	0	d	d		1	0	0	0	1	1	0	0	0	0	0	0
	SHLLL @(d:16,Rd.B)	0	1		0	0	1	0	1	6	F	0	rd	1	0	0	0	d				1	0	0	0	1	1	0	0	0	0	0	0
	SHLLL @(d:16,Rd.W)	0	1		0	0	1	1	0	6	F	0	rd	1	0	0	0	d				1	0	0	0	1	1	0	0	0	0	0	0
	SHLLL @(d:16,ERd.L)	0	1		0	0	1	1	1	6	F	0	rd	1	0	0	0	d				1	0	0	0	1	1	0	0	0	0	0	0
	SHLLL @(d:32,Rd.B)	7	8	1	rd	0	1	0	1	6	B	0	0	1	0	1	0	0	d	d		1	0	0	0	1	1	0	0	0	0	0	0
	SHLLL @(d:32,Rd.W)	7	8	1	rd	0	1	1	0	6	B	0	0	1	0	1	0	0	d	d		1	0	0	0	1	1	0	0	0	0	0	0
	SHLLL @(d:32,ERd.L)	7	8	1	rd	0	1	1	1	6	B	0	0	1	0	1	0	0	d	d		1	0	0	0	1	1	0	0	0	0	0	0
	SHLLL @aa:16	0	1		0	0	1	0	0	6	B	0	0	0	0	1	0	0	a			1	0	0	0	1	1	0	0	0	0	0	0
	SHLLL @aa:32	0	1		0	0	1	0	0	6	B	0	0	1	0	1	0	0	a	a		1	0	0	0	1	1	0	0	0	0	0	0
	SHLLL #2,ERd																					1	0	0	0	1	1	0	rd				
	SHLLL #2, @ERd	0	1		0	0	1	0	0	6	9	0	rd	1	0	0	0					1	0	0	0	1	1	1	0	0	0	0	0
	SHLLL #2, @ERd+	0	1		0	0	1	0	0	6	D	0	rd	1	0	0	0					1	0	0	0	1	1	1	0	0	0	0	0
	SHLLL #2, @ERd-	0	1		0	0	1	1	0	6	D	0	rd	1	0	0	0					1	0	0	0	1	1	1	0	0	0	0	0
SHLLL #2, @+ERd	0	1		0	0	1	0	1	6	D	0	rd	1	0	0	0					1	0	0	0	1	1	1	0	0	0	0	0	
SHLLL #2, @-ERd	0	1		0	0	1	1	1	6	D	0	rd	1	0	0	0					1	0	0	0	1	1	1	0	0	0	0	0	

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
SHLL	SHLLL #2,@(d:2,ERd)	0	1	0	0 1 d d	6	9	0 rd	1 0 0 0		1	0	0 1 1 1	0 0 0 0		
	SHLLL #2,@(d:16,ERd)	0	1	0	0 1 0 0	6	F	0 rd	1 0 0 0	d	1	0	0 1 1 1	0 0 0 0		
	SHLLL #2,@(d:32,ERd)	7	8	1 rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0	d d	1	0	0 1 1 1	0 0 0 0		
	SHLLL #2,@(d:16,Rd,B)	0	1	0	0 1 0 1	6	F	0 rd	1 0 0 0	d	1	0	0 1 1 1	0 0 0 0		
	SHLLL #2,@(d:16,Rd,W)	0	1	0	0 1 1 0	6	F	0 rd	1 0 0 0	d	1	0	0 1 1 1	0 0 0 0		
	SHLLL #2,@(d:16,ERd,L)	0	1	0	0 1 1 1	6	F	0 rd	1 0 0 0	d	1	0	0 1 1 1	0 0 0 0		
	SHLLL #2,@(d:32,Rd,B)	7	8	1 rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0	d d	1	0	0 1 1 1	0 0 0 0		
	SHLLL #2,@(d:32,Rd,W)	7	8	1 rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0	d d	1	0	0 1 1 1	0 0 0 0		
	SHLLL #2,@(d:32,ERd,L)	7	8	1 rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0	d d	1	0	0 1 1 1	0 0 0 0		
	SHLLL #2,@aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0	1 0 0 0	a	1	0	0 1 1 1	0 0 0 0		
	SHLLL #2,@aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0	1 0 0 0	a a	1	0	0 1 1 1	0 0 0 0		
	SHLLL #4,ERd											1	0	0 0 1 1	1 rd	
	SHLLL #4,@ERd	0	1	0	0 1 0 0	6	9	0 rd	1 0 0 0		1	0	0 0 1 1	1 0 0 0		
	SHLLL #4,@ERd+	0	1	0	0 1 0 0	6	D	0 rd	1 0 0 0		1	0	0 0 1 1	1 0 0 0		
	SHLLL #4,@ERd-	0	1	0	0 1 1 0	6	D	0 rd	1 0 0 0		1	0	0 0 1 1	1 0 0 0		
	SHLLL #4,@+ERd	0	1	0	0 1 0 1	6	D	0 rd	1 0 0 0		1	0	0 0 1 1	1 0 0 0		
	SHLLL #4,@-ERd	0	1	0	0 1 1 1	6	D	0 rd	1 0 0 0		1	0	0 0 1 1	1 0 0 0		
	SHLLL #4,@(d:2,ERd)	0	1	0	0 1 d d	6	9	0 rd	1 0 0 0		1	0	0 0 1 1	1 0 0 0		
	SHLLL #4,@(d:16,ERd)	0	1	0	0 1 0 0	6	F	0 rd	1 0 0 0	d	1	0	0 0 1 1	1 0 0 0		
	SHLLL #4,@(d:32,ERd)	7	8	1 rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0	d d	1	0	0 0 1 1	1 0 0 0		
	SHLLL #4,@(d:16,Rd,B)	0	1	0	0 1 0 1	6	F	0 rd	1 0 0 0	d	1	0	0 0 1 1	1 0 0 0		
	SHLLL #4,@(d:16,Rd,W)	0	1	0	0 1 1 0	6	F	0 rd	1 0 0 0	d	1	0	0 0 1 1	1 0 0 0		
	SHLLL #4,@(d:16,ERd,L)	0	1	0	0 1 1 1	6	F	0 rd	1 0 0 0	d	1	0	0 0 1 1	1 0 0 0		
	SHLLL #4,@(d:32,Rd,B)	7	8	1 rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0	d d	1	0	0 0 1 1	1 0 0 0		
	SHLLL #4,@(d:32,Rd,W)	7	8	1 rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0	d d	1	0	0 0 1 1	1 0 0 0		
	SHLLL #4,@(d:32,ERd,L)	7	8	1 rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0	d d	1	0	0 0 1 1	1 0 0 0		
	SHLLL #4,@aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0	1 0 0 0	a	1	0	0 0 1 1	1 0 0 0		
	SHLLL #4,@aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0	1 0 0 0	a a	1	0	0 0 1 1	1 0 0 0		
	SHLLL #8,ERd											1	0	0 1 1 1	1 rd	
	SHLLL #8,@ERd	0	1	0	0 1 0 0	6	9	0 rd	1 0 0 0		1	0	0 1 1 1	1 0 0 0		
	SHLLL #8,@ERd+	0	1	0	0 1 0 0	6	D	0 rd	1 0 0 0		1	0	0 1 1 1	1 0 0 0		
	SHLLL #8,@ERd-	0	1	0	0 1 1 0	6	D	0 rd	1 0 0 0		1	0	0 1 1 1	1 0 0 0		
SHLLL #8,@+ERd	0	1	0	0 1 0 1	6	D	0 rd	1 0 0 0		1	0	0 1 1 1	1 0 0 0			
SHLLL #8,@-ERd	0	1	0	0 1 1 1	6	D	0 rd	1 0 0 0		1	0	0 1 1 1	1 0 0 0			



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
SHLL	SHLL #8,@(d:2,ERd)	0	1	0	0 1 d d	6	9	0	rd	1 0 0 0		1	0	0 1 1 1	1 0 0 0		
	SHLL #8,@(d:16,ERd)	0	1	0	0 1 0 0	6	F	0	rd	1 0 0 0	d	1	0	0 1 1 1	1 0 0 0		
	SHLL #8,@(d:32,ERd)	7	8	1	rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0	d d	1	0	0 1 1 1	1 0 0 0		
	SHLL #8,@(d:16,Rd,B)	0	1	0	0 1 0 1	6	F	0	rd	1 0 0 0	d	1	0	0 1 1 1	1 0 0 0		
	SHLL #8,@(d:16,Rd,W)	0	1	0	0 1 1 0	6	F	0	rd	1 0 0 0	d	1	0	0 1 1 1	1 0 0 0		
	SHLL #8,@(d:16,ERd,L)	0	1	0	0 1 1 1	6	F	0	rd	1 0 0 0	d	1	0	0 1 1 1	1 0 0 0		
	SHLL #8,@(d:32,Rd,B)	7	8	1	rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0	d d	1	0	0 1 1 1	1 0 0 0		
	SHLL #8,@(d:32,Rd,W)	7	8	1	rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0	d d	1	0	0 1 1 1	1 0 0 0		
	SHLL #8,@(d:32,ERd,L)	7	8	1	rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0	d d	1	0	0 1 1 1	1 0 0 0		
	SHLL #8,@aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0		1 0 0 0	a	1	0	0 1 1 1	1 0 0 0		
	SHLL #8,@aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0		1 0 0 0	a a	1	0	0 1 1 1	1 0 0 0		
	SHLL #16,ERd												1	0	1 1 1 1	1 rd	
	SHLL #16,@ERd	0	1	0	0 1 0 0	6	9	0	rd	1 0 0 0		1	0	1 1 1 1	1 0 0 0		
	SHLL #16,@ERd+	0	1	0	0 1 0 0	6	D	0	rd	1 0 0 0		1	0	1 1 1 1	1 0 0 0		
	SHLL #16,@ERd-	0	1	0	0 1 1 0	6	D	0	rd	1 0 0 0		1	0	1 1 1 1	1 0 0 0		
	SHLL #16,@+ERd	0	1	0	0 1 0 1	6	D	0	rd	1 0 0 0		1	0	1 1 1 1	1 0 0 0		
	SHLL #16,@-ERd	0	1	0	0 1 1 1	6	D	0	rd	1 0 0 0		1	0	1 1 1 1	1 0 0 0		
	SHLL #16,@(d:2,ERd)	0	1	0	0 1 d d	6	9	0	rd	1 0 0 0		1	0	1 1 1 1	1 0 0 0		
	SHLL #16,@(d:16,ERd)	0	1	0	0 1 0 0	6	F	0	rd	1 0 0 0	d	1	0	1 1 1 1	1 0 0 0		
	SHLL #16,@(d:32,ERd)	7	8	1	rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0	d d	1	0	1 1 1 1	1 0 0 0		
SHLL #16,@(d:16,Rd,B)	0	1	0	0 1 0 1	6	F	0	rd	1 0 0 0	d	1	0	1 1 1 1	1 0 0 0			
SHLL #16,@(d:16,Rd,W)	0	1	0	0 1 1 0	6	F	0	rd	1 0 0 0	d	1	0	1 1 1 1	1 0 0 0			
SHLL #16,@(d:16,ERd,L)	0	1	0	0 1 1 1	6	F	0	rd	1 0 0 0	d	1	0	1 1 1 1	1 0 0 0			
SHLL #16,@(d:32,Rd,B)	7	8	1	rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0	d d	1	0	1 1 1 1	1 0 0 0			
SHLL #16,@(d:32,Rd,W)	7	8	1	rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0	d d	1	0	1 1 1 1	1 0 0 0			
SHLL #16,@(d:32,ERd,L)	7	8	1	rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0	d d	1	0	1 1 1 1	1 0 0 0			
SHLL #16,@aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0		1 0 0 0	a	1	0	1 1 1 1	1 0 0 0			
SHLL #16,@aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0		1 0 0 0	a a	1	0	1 1 1 1	1 0 0 0			
SHLR	SHLR,B #xx:5,Rd					0	3	1 0 0 x	x x x x			1	1	0 0 0 0	rd		
	SHLR,B Rn,Rd					7	8	m	1 0 0 0			1	1	0 0 0 0	rd		
	SHLR,B Rd											1	1	0 0 0 0	rd		
	SHLR,B @ERd					7	D	0	rd	0 0 0 0		1	1	0 0 0 0	0 0 0 0		
	SHLR,B @ERd+	0	1	7	0 1 0 0	6	C	0	rd	1 0 0 0		1	1	0 0 0 0	0 0 0 0		
	SHLR,B @ERd-	0	1	7	0 1 1 0	6	C	0	rd	1 0 0 0		1	1	0 0 0 0	0 0 0 0		

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
SHLR	SHLR.B @+ERd	0	1	7	0 1 0 1	6	C	0	rd	1 0 0 0		1	1	0 0 0 0	0 0 0 0		
	SHLR.B @-ERd	0	1	7	0 1 1 1	6	C	0	rd	1 0 0 0		1	1	0 0 0 0	0 0 0 0		
	SHLR.B @(d:2,ERd)	0	1	7	0 1 d d	6	8	0	rd	1 0 0 0		1	1	0 0 0 0	0 0 0 0		
	SHLR.B @(d:16,ERd)	0	1	7	0 1 0 0	6	E	0	rd	1 0 0 0	d	1	1	0 0 0 0	0 0 0 0		
	SHLR.B @(d:32,ERd)	7	8	0	rd	0 1 0 0	6	A	0 0 1 0	1 0 0 0	d d	1	1	0 0 0 0	0 0 0 0		
	SHLR.B @(d:16,Rd.B)	0	1	7	0 1 0 1	6	E	0	rd	1 0 0 0	d	1	1	0 0 0 0	0 0 0 0		
	SHLR.B @(d:16,Rd.W)	0	1	7	0 1 1 0	6	E	0	rd	1 0 0 0	d	1	1	0 0 0 0	0 0 0 0		
	SHLR.B @(d:16,ERd.L)	0	1	7	0 1 1 1	6	E	0	rd	1 0 0 0	d	1	1	0 0 0 0	0 0 0 0		
	SHLR.B @(d:32,Rd.B)	7	8	0	rd	0 1 0 1	6	A	0 0 1 0	1 0 0 0	d d	1	1	0 0 0 0	0 0 0 0		
	SHLR.B @(d:32,Rd.W)	7	8	0	rd	0 1 1 0	6	A	0 0 1 0	1 0 0 0	d d	1	1	0 0 0 0	0 0 0 0		
	SHLR.B @(d:32,ERd.L)	7	8	0	rd	0 1 1 1	6	A	0 0 1 0	1 0 0 0	d d	1	1	0 0 0 0	0 0 0 0		
	SHLR.B @aa:8						7	F	a a a a	a a a a		1	1	0 0 0 0	0 0 0 0		
	SHLR.B @aa:16						6	A	0 0 0 1	1 0 0 0	a	1	1	0 0 0 0	0 0 0 0		
	SHLR.B @aa:32						6	A	0 0 1 1	1 0 0 0	a a	1	1	0 0 0 0	0 0 0 0		
	SHLR.B #2,Rd											1	1	0 1 0 0	rd		
	SHLR.B #2,@ERd						7	D	0	rd	0 0 0 0		1	1	0 1 0 0	0 0 0 0	
	SHLR.B #2,@ERd+	0	1	7	0 1 0 0	6	C	0	rd	1 0 0 0		1	1	0 1 0 0	0 0 0 0		
	SHLR.B #2,@ERd-	0	1	7	0 1 1 0	6	C	0	rd	1 0 0 0		1	1	0 1 0 0	0 0 0 0		
	SHLR.B #2,@+ERd	0	1	7	0 1 0 1	6	C	0	rd	1 0 0 0		1	1	0 1 0 0	0 0 0 0		
	SHLR.B #2,@-ERd	0	1	7	0 1 1 1	6	C	0	rd	1 0 0 0		1	1	0 1 0 0	0 0 0 0		
	SHLR.B #2,@(d:2,ERd)	0	1	7	0 1 d d	6	8	0	rd	1 0 0 0		1	1	0 1 0 0	0 0 0 0		
	SHLR.B #2,@(d:16,ERd)	0	1	7	0 1 0 0	6	E	0	rd	1 0 0 0	d	1	1	0 1 0 0	0 0 0 0		
	SHLR.B #2,@(d:32,ERd)	7	8	0	rd	0 1 0 0	6	A	0 0 1 0	1 0 0 0	d d	1	1	0 1 0 0	0 0 0 0		
	SHLR.B #2,@(d:16,Rd.B)	0	1	7	0 1 0 1	6	E	0	rd	1 0 0 0	d	1	1	0 1 0 0	0 0 0 0		
	SHLR.B #2,@(d:16,Rd.W)	0	1	7	0 1 1 0	6	E	0	rd	1 0 0 0	d	1	1	0 1 0 0	0 0 0 0		
	SHLR.B #2,@(d:16,ERd.L)	0	1	7	0 1 1 1	6	E	0	rd	1 0 0 0	d	1	1	0 1 0 0	0 0 0 0		
	SHLR.B #2,@(d:32,Rd.B)	7	8	0	rd	0 1 0 1	6	A	0 0 1 0	1 0 0 0	d d	1	1	0 1 0 0	0 0 0 0		
	SHLR.B #2,@(d:32,Rd.W)	7	8	0	rd	0 1 1 0	6	A	0 0 1 0	1 0 0 0	d d	1	1	0 1 0 0	0 0 0 0		
	SHLR.B #2,@(d:32,ERd.L)	7	8	0	rd	0 1 1 1	6	A	0 0 1 0	1 0 0 0	d d	1	1	0 1 0 0	0 0 0 0		
	SHLR.B #2,@aa:8						7	F	a a a a	a a a a		1	1	0 1 0 0	0 0 0 0		
	SHLR.B #2,@aa:16						6	A	0 0 0 1	1 0 0 0	a	1	1	0 1 0 0	0 0 0 0		
	SHLR.B #2,@aa:32						6	A	0 0 1 1	1 0 0 0	a a	1	1	0 1 0 0	0 0 0 0		
SHLR.B #4,Rd											1	1	1 0 1 0	rd			
SHLR.B #4,@ERd						7	D	0	rd	0 0 0 0		1	1	1 0 1 0	0 0 0 0		



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
SHLR	SHLR.B #4, @ERd+	0	1	7	0 1 0 0	6	C	0	rd	1 0 0 0		1	1	1 0 1 0	0 0 0 0	
	SHLR.B #4, @ERd-	0	1	7	0 1 1 0	6	C	0	rd	1 0 0 0		1	1	1 0 1 0	0 0 0 0	
	SHLR.B #4, @+ERd	0	1	7	0 1 0 1	6	C	0	rd	1 0 0 0		1	1	1 0 1 0	0 0 0 0	
	SHLR.B #4, @-ERd	0	1	7	0 1 1 1	6	C	0	rd	1 0 0 0		1	1	1 0 1 0	0 0 0 0	
	SHLR.B #4, @(d:2,ERd)	0	1	7	0 1 d d	6	8	0	rd	1 0 0 0		1	1	1 0 1 0	0 0 0 0	
	SHLR.B #4, @(d:16,ERd)	0	1	7	0 1 0 0	6	E	0	rd	1 0 0 0	d	1	1	1 0 1 0	0 0 0 0	
	SHLR.B #4, @(d:32,ERd)	7	8	0	rd	0 1 0 0	6	A	0 0 1 0	1 0 0 0	d d	1	1	1 0 1 0	0 0 0 0	
	SHLR.B #4, @(d:16,Rd.B)	0	1	7	0 1 0 1	6	E	0	rd	1 0 0 0	d	1	1	1 0 1 0	0 0 0 0	
	SHLR.B #4, @(d:16,Rd.W)	0	1	7	0 1 1 0	6	E	0	rd	1 0 0 0	d	1	1	1 0 1 0	0 0 0 0	
	SHLR.B #4, @(d:16,ERd.L)	0	1	7	0 1 1 1	6	E	0	rd	1 0 0 0	d	1	1	1 0 1 0	0 0 0 0	
	SHLR.B #4, @(d:32,Rd.B)	7	8	0	rd	0 1 0 1	6	A	0 0 1 0	1 0 0 0	d d	1	1	1 0 1 0	0 0 0 0	
	SHLR.B #4, @(d:32,Rd.W)	7	8	0	rd	0 1 1 0	6	A	0 0 1 0	1 0 0 0	d d	1	1	1 0 1 0	0 0 0 0	
	SHLR.B #4, @(d:32,ERd.L)	7	8	0	rd	0 1 1 1	6	A	0 0 1 0	1 0 0 0	d d	1	1	1 0 1 0	0 0 0 0	
	SHLR.B #4, @aa:8					7	F	a a a a	a a a a			1	1	1 0 1 0	0 0 0 0	
	SHLR.B #4, @aa:16					6	A	0 0 0 1	1 0 0 0	a		1	1	1 0 1 0	0 0 0 0	
	SHLR.B #4, @aa:32					6	A	0 0 1 1	1 0 0 0	a a		1	1	1 0 1 0	0 0 0 0	
	SHLR.W #xc:5,Rd					0	3	1 0 0 x	x x x x			1	1	0 0 0 1	rd	
	SHLR.W Rn,Rd					7	8	m	1 0 0 0			1	1	0 0 0 1	rd	
	SHLR.W Rd											1	1	0 0 0 1	rd	
	SHLR.W @ERd					7	D	1	rd	0 0 0 0		1	1	0 0 0 1	0 0 0 0	
	SHLR.W @ERd+	0	1	5	0 1 0 0	6	D	0	rd	1 0 0 0		1	1	0 0 0 1	0 0 0 0	
	SHLR.W @ERd-	0	1	5	0 1 1 0	6	D	0	rd	1 0 0 0		1	1	0 0 0 1	0 0 0 0	
	SHLR.W @+ERd	0	1	5	0 1 0 1	6	D	0	rd	1 0 0 0		1	1	0 0 0 1	0 0 0 0	
	SHLR.W @-ERd	0	1	5	0 1 1 1	6	D	0	rd	1 0 0 0		1	1	0 0 0 1	0 0 0 0	
	SHLR.W @(d:2,ERd)	0	1	5	0 1 d d	6	9	0	rd	1 0 0 0		1	1	0 0 0 1	0 0 0 0	
	SHLR.W @(d:16,ERd)	0	1	5	0 1 0 0	6	F	0	rd	1 0 0 0	d	1	1	0 0 0 1	0 0 0 0	
	SHLR.W @(d:32,ERd)	7	8	0	rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0	d d	1	1	0 0 0 1	0 0 0 0	
	SHLR.W @(d:16,Rd.B)	0	1	5	0 1 0 1	6	F	0	rd	1 0 0 0	d	1	1	0 0 0 1	0 0 0 0	
	SHLR.W @(d:16,Rd.W)	0	1	5	0 1 1 0	6	F	0	rd	1 0 0 0	d	1	1	0 0 0 1	0 0 0 0	
	SHLR.W @(d:16,ERd.L)	0	1	5	0 1 1 1	6	F	0	rd	1 0 0 0	d	1	1	0 0 0 1	0 0 0 0	
	SHLR.W @(d:32,Rd.B)	7	8	0	rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0	d d	1	1	0 0 0 1	0 0 0 0	
	SHLR.W @(d:32,Rd.W)	7	8	0	rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0	d d	1	1	0 0 0 1	0 0 0 0	
SHLR.W @(d:32,ERd.L)	7	8	0	rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0	d d	1	1	0 0 0 1	0 0 0 0		
SHLR.W @aa:16					6	B	0 0 0 1	1 0 0 0	a		1	1	0 0 0 1	0 0 0 0		

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension
		15	8	7	0	15	8	7	0		15	8	7	0	
SHLR	SHLR.W @aa:32					6	B	0 0 1 1	1 0 0 0 0	a a	1	1	0 0 0 1	0 0 0 0	
	SHLR.W #2,Rd										1	1	0 1 0 1	rd	
	SHLR.W #2,@ERd					7	D	1 rd	0 0 0 0		1	1	0 1 0 1	0 0 0 0	
	SHLR.W #2,@ERd+	0	1	5	0 1 0 0	6	D	0 rd	1 0 0 0		1	1	0 1 0 1	0 0 0 0	
	SHLR.W #2,@ERd-	0	1	5	0 1 1 0	6	D	0 rd	1 0 0 0		1	1	0 1 0 1	0 0 0 0	
	SHLR.W #2,@+ERd	0	1	5	0 1 0 1	6	D	0 rd	1 0 0 0		1	1	0 1 0 1	0 0 0 0	
	SHLR.W #2,@-ERd	0	1	5	0 1 1 1	6	D	0 rd	1 0 0 0		1	1	0 1 0 1	0 0 0 0	
	SHLR.W #2,@(d:2,ERd)	0	1	5	0 1 d d	6	9	0 rd	1 0 0 0		1	1	0 1 0 1	0 0 0 0	
	SHLR.W #2,@(d:16,ERd)	0	1	5	0 1 0 0	6	F	0 rd	1 0 0 0	d	1	1	0 1 0 1	0 0 0 0	
	SHLR.W #2,@(d:32,ERd)	7	8	0	rd 0 1 0 0	6	B	0 0 1 0	1 0 0 0	d d	1	1	0 1 0 1	0 0 0 0	
	SHLR.W #2,@(d:16,Rd,B)	0	1	5	0 1 0 1	6	F	0 rd	1 0 0 0	d	1	1	0 1 0 1	0 0 0 0	
	SHLR.W #2,@(d:16,Rd,W)	0	1	5	0 1 1 0	6	F	0 rd	1 0 0 0	d	1	1	0 1 0 1	0 0 0 0	
	SHLR.W #2,@(d:16,ERd,L)	0	1	5	0 1 1 1	6	F	0 rd	1 0 0 0	d	1	1	0 1 0 1	0 0 0 0	
	SHLR.W #2,@(d:32,Rd,B)	7	8	0	rd 0 1 0 1	6	B	0 0 1 0	1 0 0 0	d d	1	1	0 1 0 1	0 0 0 0	
	SHLR.W #2,@(d:32,Rd,W)	7	8	0	rd 0 1 1 0	6	B	0 0 1 0	1 0 0 0	d d	1	1	0 1 0 1	0 0 0 0	
	SHLR.W #2,@(d:32,ERd,L)	7	8	0	rd 0 1 1 1	6	B	0 0 1 0	1 0 0 0	d d	1	1	0 1 0 1	0 0 0 0	
	SHLR.W #2,@aa:16					6	B	0 0 0 1	1 0 0 0	a	1	1	0 1 0 1	0 0 0 0	
	SHLR.W #2,@aa:32					6	B	0 0 1 1	1 0 0 0	a a	1	1	0 1 0 1	0 0 0 0	
	SHLR.W #4,Rd										1	1	0 0 1 0	rd	
	SHLR.W #4,@ERd					7	D	1 rd	0 0 0 0		1	1	0 0 1 0	0 0 0 0	
	SHLR.W #4,@ERd+	0	1	5	0 1 0 0	6	D	0 rd	1 0 0 0		1	1	0 0 1 0	0 0 0 0	
	SHLR.W #4,@ERd-	0	1	5	0 1 1 0	6	D	0 rd	1 0 0 0		1	1	0 0 1 0	0 0 0 0	
	SHLR.W #4,@+ERd	0	1	5	0 1 0 1	6	D	0 rd	1 0 0 0		1	1	0 0 1 0	0 0 0 0	
	SHLR.W #4,@-ERd	0	1	5	0 1 1 1	6	D	0 rd	1 0 0 0		1	1	0 0 1 0	0 0 0 0	
	SHLR.W #4,@(d:2,ERd)	0	1	5	0 1 d d	6	9	0 rd	1 0 0 0		1	1	0 0 1 0	0 0 0 0	
	SHLR.W #4,@(d:16,ERd)	0	1	5	0 1 0 0	6	F	0 rd	1 0 0 0	d	1	1	0 0 1 0	0 0 0 0	
	SHLR.W #4,@(d:32,ERd)	7	8	0	rd 0 1 0 0	6	B	0 0 1 0	1 0 0 0	d d	1	1	0 0 1 0	0 0 0 0	
	SHLR.W #4,@(d:16,Rd,B)	0	1	5	0 1 0 1	6	F	0 rd	1 0 0 0	d	1	1	0 0 1 0	0 0 0 0	
	SHLR.W #4,@(d:16,Rd,W)	0	1	5	0 1 1 0	6	F	0 rd	1 0 0 0	d	1	1	0 0 1 0	0 0 0 0	
	SHLR.W #4,@(d:16,ERd,L)	0	1	5	0 1 1 1	6	F	0 rd	1 0 0 0	d	1	1	0 0 1 0	0 0 0 0	
	SHLR.W #4,@(d:32,Rd,B)	7	8	0	rd 0 1 0 1	6	B	0 0 1 0	1 0 0 0	d d	1	1	0 0 1 0	0 0 0 0	
	SHLR.W #4,@(d:32,Rd,W)	7	8	0	rd 0 1 1 0	6	B	0 0 1 0	1 0 0 0	d d	1	1	0 0 1 0	0 0 0 0	
SHLR.W #4,@(d:32,ERd,L)	7	8	0	rd 0 1 1 1	6	B	0 0 1 0	1 0 0 0	d d	1	1	0 0 1 0	0 0 0 0		
SHLR.W #4,@aa:16					6	B	0 0 0 1	1 0 0 0	a	1	1	0 0 1 0	0 0 0 0		



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension
		15	8	7	0	15	8	7	0		15	8	7	0	
SHLR	SHLR.W #4, @aa:32					6	B	0 0 1 1	1 1 0 0 0 0	a a	1	1	0 0 1 0	0 0 0 0 0	
	SHLR.W #8,Rd										1	1	0 1 1 0	rd	
	SHLR.W #8,@ERd					7	D	1	rd 0 0 0 0 0		1	1	0 1 1 0	0 0 0 0 0	
	SHLR.W #8,@ERd+	0	1	5	0 1 0 0	6	D	0	rd 1 0 0 0 0		1	1	0 1 1 0	0 0 0 0 0	
	SHLR.W #8,@ERd-	0	1	5	0 1 1 0	6	D	0	rd 1 0 0 0 0		1	1	0 1 1 0	0 0 0 0 0	
	SHLR.W #8,@+ERd	0	1	5	0 1 0 1	6	D	0	rd 1 0 0 0 0		1	1	0 1 1 0	0 0 0 0 0	
	SHLR.W #8,@-ERd	0	1	5	0 1 1 1	6	D	0	rd 1 0 0 0 0		1	1	0 1 1 0	0 0 0 0 0	
	SHLR.W #8,@(d:2,ERd)	0	1	5	0 1 d d	6	9	0	rd 1 0 0 0 0		1	1	0 1 1 0	0 0 0 0 0	
	SHLR.W #8,@(d:16,ERd)	0	1	5	0 1 0 0	6	F	0	rd 1 0 0 0 0	d	1	1	0 1 1 0	0 0 0 0 0	
	SHLR.W #8,@(d:32,ERd)	7	8	0	rd 0 1 0 0	6	B	0 0 1 0	1 0 0 0 0	d d	1	1	0 1 1 0	0 0 0 0 0	
	SHLR.W #8,@(d:16,Rd,B)	0	1	5	0 1 0 1	6	F	0	rd 1 0 0 0 0	d	1	1	0 1 1 0	0 0 0 0 0	
	SHLR.W #8,@(d:16,Rd,W)	0	1	5	0 1 1 0	6	F	0	rd 1 0 0 0 0	d	1	1	0 1 1 0	0 0 0 0 0	
	SHLR.W #8,@(d:16,ERd,L)	0	1	5	0 1 1 1	6	F	0	rd 1 0 0 0 0	d	1	1	0 1 1 0	0 0 0 0 0	
	SHLR.W #8,@(d:32,Rd,B)	7	8	0	rd 0 1 0 1	6	B	0 0 1 0	1 0 0 0 0	d d	1	1	0 1 1 0	0 0 0 0 0	
	SHLR.W #8,@(d:32,Rd,W)	7	8	0	rd 0 1 1 0	6	B	0 0 1 0	1 0 0 0 0	d d	1	1	0 1 1 0	0 0 0 0 0	
	SHLR.W #8,@(d:32,ERd,L)	7	8	0	rd 0 1 1 1	6	B	0 0 1 0	1 0 0 0 0	d d	1	1	0 1 1 0	0 0 0 0 0	
	SHLR.W #8,@aa:16					6	B	0 0 0 1	1 0 0 0 0	a	1	1	0 1 1 0	0 0 0 0 0	
	SHLR.W #8,@aa:32					6	B	0 0 1 1	1 0 0 0 0	a a	1	1	0 1 1 0	0 0 0 0 0	
	SHLR.L #xx:5,ERd					0	3	1 0 0 x	x x x x x		1	1	0 0 1 1	0 rd	
	SHLR.L Rn,ERd					7	8	m	1 0 0 0 0		1	1	0 0 1 1	0 rd	
	SHLR.L ERd										1	1	0 0 1 1	0 rd	
	SHLR.L @ERd	0	1	0	0 1 0 0	6	9	0	rd 1 0 0 0 0		1	1	0 0 1 1	0 0 0 0 0	
	SHLR.L @ERd+	0	1	0	0 1 0 0	6	D	0	rd 1 0 0 0 0		1	1	0 0 1 1	0 0 0 0 0	
SHLR.L @ERd-	0	1	0	0 1 1 0	6	D	0	rd 1 0 0 0 0		1	1	0 0 1 1	0 0 0 0 0		
SHLR.L @+ERd	0	1	0	0 1 0 1	6	D	0	rd 1 0 0 0 0		1	1	0 0 1 1	0 0 0 0 0		
SHLR.L @-ERd	0	1	0	0 1 1 1	6	D	0	rd 1 0 0 0 0		1	1	0 0 1 1	0 0 0 0 0		
SHLR.L @(d:2,ERd)	0	1	0	0 1 d d	6	9	0	rd 1 0 0 0 0		1	1	0 0 1 1	0 0 0 0 0		
SHLR.L @(d:16,ERd)	0	1	0	0 1 0 0	6	F	0	rd 1 0 0 0 0	d	1	1	0 0 1 1	0 0 0 0 0		
SHLR.L @(d:32,ERd)	7	8	1	rd 0 1 0 0	6	B	0 0 1 0	1 0 0 0 0	d d	1	1	0 0 1 1	0 0 0 0 0		
SHLR.L @(d:16,Rd,B)	0	1	0	0 1 0 1	6	F	0	rd 1 0 0 0 0	d	1	1	0 0 1 1	0 0 0 0 0		
SHLR.L @(d:16,Rd,W)	0	1	0	0 1 1 0	6	F	0	rd 1 0 0 0 0	d	1	1	0 0 1 1	0 0 0 0 0		
SHLR.L @(d:16,ERd,L)	0	1	0	0 1 1 1	6	F	0	rd 1 0 0 0 0	d	1	1	0 0 1 1	0 0 0 0 0		
SHLR.L @(d:32,Rd,B)	7	8	1	rd 0 1 0 1	6	B	0 0 1 0	1 0 0 0 0	d d	1	1	0 0 1 1	0 0 0 0 0		
SHLR.L @(d:32,Rd,W)	7	8	1	rd 0 1 1 0	6	B	0 0 1 0	1 0 0 0 0	d d	1	1	0 0 1 1	0 0 0 0 0		

Instruction	Mnemonic	Opcode						EA Ex- tension	Opcode						EA Extension
		15	8	7	0				15	8	7	0			
SHLR	SHLRL @(d:32,ERd.L)	7	8	1	rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0	d d	1	1	0 0 1 1	0 0 0 0
	SHLRL @aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0	1 0 0 0	a	1	1	0 0 1 1	0 0 0 0	
	SHLRL @aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0	1 0 0 0	a a	1	1	0 0 1 1	0 0 0 0	
	SHLRL #2,ERd											1	1	0 1 1 1	0 rd
	SHLRL #2,@ERd	0	1	0	0 1 0 0	6	9	0 rd	1 0 0 0			1	1	0 1 1 1	0 0 0 0
	SHLRL #2,@ERd+	0	1	0	0 1 0 0	6	D	0 rd	1 0 0 0			1	1	0 1 1 1	0 0 0 0
	SHLRL #2,@ERd-	0	1	0	0 1 1 0	6	D	0 rd	1 0 0 0			1	1	0 1 1 1	0 0 0 0
	SHLRL #2,@+ERd	0	1	0	0 1 0 1	6	D	0 rd	1 0 0 0			1	1	0 1 1 1	0 0 0 0
	SHLRL #2,@-ERd	0	1	0	0 1 1 1	6	D	0 rd	1 0 0 0			1	1	0 1 1 1	0 0 0 0
	SHLRL #2,@(d:2,ERd)	0	1	0	0 1 d d	6	9	0 rd	1 0 0 0			1	1	0 1 1 1	0 0 0 0
	SHLRL #2,@(d:16,ERd)	0	1	0	0 1 0 0	6	F	0 rd	1 0 0 0	d		1	1	0 1 1 1	0 0 0 0
	SHLRL #2,@(d:32,ERd)	7	8	1	rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0	d d	1	1	0 1 1 1	0 0 0 0
	SHLRL #2,@(d:16,Rd,B)	0	1	0	0 1 0 1	6	F	0 rd	1 0 0 0	d		1	1	0 1 1 1	0 0 0 0
	SHLRL #2,@(d:16,Rd,W)	0	1	0	0 1 1 0	6	F	0 rd	1 0 0 0	d		1	1	0 1 1 1	0 0 0 0
	SHLRL #2,@(d:16,ERd.L)	0	1	0	0 1 1 1	6	F	0 rd	1 0 0 0	d		1	1	0 1 1 1	0 0 0 0
	SHLRL #2,@(d:32,Rd,B)	7	8	1	rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0	d d	1	1	0 1 1 1	0 0 0 0
	SHLRL #2,@(d:32,Rd,W)	7	8	1	rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0	d d	1	1	0 1 1 1	0 0 0 0
	SHLRL #2,@(d:32,ERd.L)	7	8	1	rd	0 1 1 1	6	B	0 0 1 0	1 0 0 0	d d	1	1	0 1 1 1	0 0 0 0
	SHLRL #2,@aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0	1 0 0 0	a		1	1	0 1 1 1	0 0 0 0
	SHLRL #2,@aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0	1 0 0 0	a a		1	1	0 1 1 1	0 0 0 0
	SHLRL #4,ERd											1	1	0 0 1 1	1 rd
	SHLRL #4,@ERd	0	1	0	0 1 0 0	6	9	0 rd	1 0 0 0			1	1	0 0 1 1	1 0 0 0
	SHLRL #4,@ERd+	0	1	0	0 1 0 0	6	D	0 rd	1 0 0 0			1	1	0 0 1 1	1 0 0 0
	SHLRL #4,@ERd-	0	1	0	0 1 1 0	6	D	0 rd	1 0 0 0			1	1	0 0 1 1	1 0 0 0
	SHLRL #4,@+ERd	0	1	0	0 1 0 1	6	D	0 rd	1 0 0 0			1	1	0 0 1 1	1 0 0 0
	SHLRL #4,@-ERd	0	1	0	0 1 1 1	6	D	0 rd	1 0 0 0			1	1	0 0 1 1	1 0 0 0
	SHLRL #4,@(d:2,ERd)	0	1	0	0 1 d d	6	9	0 rd	1 0 0 0			1	1	0 0 1 1	1 0 0 0
	SHLRL #4,@(d:16,ERd)	0	1	0	0 1 0 0	6	F	0 rd	1 0 0 0	d		1	1	0 0 1 1	1 0 0 0
	SHLRL #4,@(d:32,ERd)	7	8	1	rd	0 1 0 0	6	B	0 0 1 0	1 0 0 0	d d	1	1	0 0 1 1	1 0 0 0
	SHLRL #4,@(d:16,Rd,B)	0	1	0	0 1 0 1	6	F	0 rd	1 0 0 0	d		1	1	0 0 1 1	1 0 0 0
	SHLRL #4,@(d:16,Rd,W)	0	1	0	0 1 1 0	6	F	0 rd	1 0 0 0	d		1	1	0 0 1 1	1 0 0 0
	SHLRL #4,@(d:16,ERd.L)	0	1	0	0 1 1 1	6	F	0 rd	1 0 0 0	d		1	1	0 0 1 1	1 0 0 0
SHLRL #4,@(d:32,Rd,B)	7	8	1	rd	0 1 0 1	6	B	0 0 1 0	1 0 0 0	d d	1	1	0 0 1 1	1 0 0 0	
SHLRL #4,@(d:32,Rd,W)	7	8	1	rd	0 1 1 0	6	B	0 0 1 0	1 0 0 0	d d	1	1	0 0 1 1	1 0 0 0	



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension																				
		15	8	7	0	15	8	7	0		15	8	7	0																					
SHLR	SHLRL #4,@(d:32,ERd.L)	7	8	1	rd	0	1	1	1	6	B	0	0	1	0	1	0	0	0	d	d	1	1	0	0	1	1	1	0	0	0	0			
	SHLRL #4,@aa:16	0	1		0	0	1	0	0	6	B	0	0	0	0	1	0	0	0	a		1	1	0	0	1	1	1	1	0	0	0	0		
	SHLRL #4,@aa:32	0	1		0	0	1	0	0	6	B	0	0	1	0	1	0	0	0	a	a	1	1	0	0	1	1	1	1	0	0	0	0		
	SHLRL #8,ERd																					1	1	0	1	1	1	1	1	rd					
	SHLRL #8,@ERd	0	1		0	0	1	0	0	6	9	0	rd	1	0	0	0	0				1	1	0	1	1	1	1	0	0	0	0	0	0	
	SHLRL #8,@ERd+	0	1		0	0	1	0	0	6	D	0	rd	1	0	0	0	0				1	1	0	1	1	1	1	1	0	0	0	0	0	
	SHLRL #8,@ERd-	0	1		0	0	1	1	0	6	D	0	rd	1	0	0	0	0				1	1	0	1	1	1	1	1	0	0	0	0	0	
	SHLRL #8,@-ERd	0	1		0	0	1	0	1	6	D	0	rd	1	0	0	0	0				1	1	0	1	1	1	1	1	0	0	0	0	0	
	SHLRL #8,@-ERd	0	1		0	0	1	1	1	6	D	0	rd	1	0	0	0	0				1	1	0	1	1	1	1	1	0	0	0	0	0	0
	SHLRL #8,@(d:2,ERd)	0	1		0	0	1	d	d	6	9	0	rd	1	0	0	0	0				1	1	0	1	1	1	1	1	0	0	0	0	0	0
	SHLRL #8,@(d:16,ERd)	0	1		0	0	1	0	0	6	F	0	rd	1	0	0	0	0	d			1	1	0	1	1	1	1	1	0	0	0	0	0	0
	SHLRL #8,@(d:32,ERd)	7	8	1	rd	0	1	0	0	6	B	0	0	1	0	1	0	0	0	d	d	1	1	0	1	1	1	1	1	0	0	0	0	0	0
	SHLRL #8,@(d:16,Rd,B)	0	1		0	0	1	0	1	6	F	0	rd	1	0	0	0	0	d			1	1	0	1	1	1	1	1	0	0	0	0	0	0
	SHLRL #8,@(d:16,Rd,W)	0	1		0	0	1	1	0	6	F	0	rd	1	0	0	0	0	d			1	1	0	1	1	1	1	1	0	0	0	0	0	0
	SHLRL #8,@(d:16,ERd.L)	0	1		0	0	1	1	1	6	F	0	rd	1	0	0	0	0	d			1	1	0	1	1	1	1	1	0	0	0	0	0	0
	SHLRL #8,@(d:32,Rd,B)	7	8	1	rd	0	1	0	1	6	B	0	0	1	0	1	0	0	0	d	d	1	1	0	1	1	1	1	1	0	0	0	0	0	0
	SHLRL #8,@(d:32,Rd,W)	7	8	1	rd	0	1	1	0	6	B	0	0	1	0	1	0	0	0	d	d	1	1	0	1	1	1	1	1	0	0	0	0	0	0
	SHLRL #8,@(d:32,ERd.L)	7	8	1	rd	0	1	1	1	6	B	0	0	1	0	1	0	0	0	d	d	1	1	0	1	1	1	1	1	0	0	0	0	0	0
	SHLRL #8,@aa:16	0	1		0	0	1	0	0	6	B	0	0	0	0	1	0	0	0	a		1	1	0	1	1	1	1	1	0	0	0	0	0	0
	SHLRL #8,@aa:32	0	1		0	0	1	0	0	6	B	0	0	1	0	1	0	0	0	a	a	1	1	0	1	1	1	1	1	0	0	0	0	0	0
	SHLRL #16,ERd																					1	1	1	1	1	1	1	1	rd					
	SHLRL #16,@ERd	0	1		0	0	1	0	0	6	9	0	rd	1	0	0	0	0				1	1	1	1	1	1	1	1	0	0	0	0	0	0
	SHLRL #16,@ERd+	0	1		0	0	1	0	0	6	D	0	rd	1	0	0	0	0				1	1	1	1	1	1	1	1	0	0	0	0	0	0
	SHLRL #16,@ERd-	0	1		0	0	1	1	0	6	D	0	rd	1	0	0	0	0				1	1	1	1	1	1	1	1	0	0	0	0	0	0
	SHLRL #16,@+ERd	0	1		0	0	1	0	1	6	D	0	rd	1	0	0	0	0				1	1	1	1	1	1	1	1	0	0	0	0	0	0
	SHLRL #16,@-ERd	0	1		0	0	1	1	1	6	D	0	rd	1	0	0	0	0				1	1	1	1	1	1	1	1	0	0	0	0	0	0
	SHLRL #16,@(d:2,ERd)	0	1		0	0	1	d	d	6	9	0	rd	1	0	0	0	0				1	1	1	1	1	1	1	1	0	0	0	0	0	0
	SHLRL #16,@(d:16,ERd)	0	1		0	0	1	0	0	6	F	0	rd	1	0	0	0	0	d			1	1	1	1	1	1	1	1	0	0	0	0	0	0
	SHLRL #16,@(d:32,ERd)	7	8	1	rd	0	1	0	0	6	B	0	0	1	0	1	0	0	0	d	d	1	1	0	1	1	1	1	1	0	0	0	0	0	0
	SHLRL #16,@(d:16,Rd,B)	0	1		0	0	1	0	1	6	F	0	rd	1	0	0	0	0	d			1	1	1	1	1	1	1	1	0	0	0	0	0	0
SHLRL #16,@(d:16,Rd,W)	0	1		0	0	1	1	0	6	F	0	rd	1	0	0	0	0	d			1	1	1	1	1	1	1	1	0	0	0	0	0	0	
SHLRL #16,@(d:16,ERd.L)	0	1		0	0	1	1	1	6	F	0	rd	1	0	0	0	0	d			1	1	1	1	1	1	1	1	0	0	0	0	0	0	
SHLRL #16,@(d:32,Rd,B)	7	8	1	rd	0	1	0	1	6	B	0	0	1	0	1	0	0	0	d	d	1	1	0	1	1	1	1	1	0	0	0	0	0	0	
SHLRL #16,@(d:32,Rd,W)	7	8	1	rd	0	1	1	0	6	B	0	0	1	0	1	0	0	0	d	d	1	1	0	1	1	1	1	1	0	0	0	0	0	0	

Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension																									
		15	8	7	0	15	8	7	0		15	8	7	0																										
SHLR	SHL,R,L #16,@(d:32,ERd,L)	7	8	1	rd	0	1	1	1	6	B	0	0	1	0	1	0	0	0	0	d	d	1	1	1	1	1	1	1	0	0	0	0							
	SHL,R,L #16,@aa:16	0	1	0	0	1	0	0	0	6	B	0	0	0	0	1	0	0	0	0	a	1	1	1	1	1	1	1	0	0	0	0								
	SHL,R,L #16,@aa:32	0	1	0	0	1	0	0	0	6	B	0	0	1	0	1	0	0	0	a	a	1	1	1	1	1	1	1	0	0	0	0								
SLEEP	SLEEP																				0	1	8	0	0	0	0	0												
STC	STC,B CCR,Rd																				0	2	0	0	0	0	rd													
	STC,W CCR,@ERd																				0	1	4	0	0	0	0		6	9	rd	0	0	0	0					
	STC,W CCR,@-ERd																				0	1	4	0	0	0	0		6	D	rd	0	0	0	0					
	STC,W CCR,@(d:16,ERd)																				0	1	4	0	0	0	0		6	F	rd	0	0	0	0	d				
	STC,W CCR,@(d:32,ERd)	0	1	4	0	0	0	0	0	7	8	0	rd	0	0	0	0	0	6	B	1	0	1	0	0	0	0	0	0	0	0	d	d							
	STC,W CCR,@aa:16																				0	1	4	0	0	0	0		6	B	1	0	0	0	0	0	0	a		
	STC,W CCR,@aa:32																				0	1	4	0	0	0	0		6	B	1	0	1	0	0	0	0	a	a	
	STC,B EXR,Rd																				0	2	0	0	0	1	rd													
	STC,W EXR,@ERd																				0	1	4	0	0	0	1		6	9	rd	0	0	0	0					
	STC,W EXR,@-ERd																				0	1	4	0	0	0	1		6	D	rd	0	0	0	0					
	STC,W EXR,@(d:16,ERd)																				0	1	4	0	0	0	1		6	F	rd	0	0	0	0	d				
	STC,W EXR,@(d:32,ERd)	0	1	4	0	0	0	1		7	8	0	rd	0	0	0	0	6	B	1	0	1	0	0	0	0	0	0	0	0	d	d								
	STC,W EXR,@aa:16																				0	1	4	0	0	0	1		6	B	1	0	0	0	0	0	0	a		
	STC,W EXR,@aa:32																				0	1	4	0	0	0	1		6	B	1	0	1	0	0	0	0	a	a	
	STC,L SBR,ERd																				0	2	0	1	1	0	rd													
STC,L VBR,ERd																				0	2	0	1	1	0	rd														
STM	STM,L (ERn-ERn+1),@-SP																				0	1	1	0	0	0	0		6	D	1	1	1	1	0	m				
	STM,L (ERn-ERn+2),@-SP																				0	1	2	0	0	0	0		6	D	1	1	1	1	0	m				
	STM,L (ERn-ERn+3),@-SP																				0	1	3	0	0	0	0		6	D	1	1	1	1	0	m				
STMAC	STMAC MACH,ERd																				0	2	0	0	1	0	0	rd												
	STMAC MACL,ERd																				0	2	0	0	1	1	0	rd												
SUB	SUB,B #xx:8,@ERd																				7	D	0	rd	0	0	0	0	A	0	0	0	1	x	x	x	x	x	x	x
	SUB,B #xx:8,@ERd+	0	1	7	0	1	0	0	6	C	0	rd	1	0	0	0	A	0	0	0	1	x	x	x	x	x	x	x	x	x	x									
	SUB,B #xx:8,@ERd-	0	1	7	0	1	1	0	6	C	0	rd	1	0	0	0	A	0	0	0	1	x	x	x	x	x	x	x	x	x	x									
	SUB,B #xx:8,@+ERd	0	1	7	0	1	0	1	6	C	0	rd	1	0	0	0	A	0	0	0	1	x	x	x	x	x	x	x	x	x	x									
	SUB,B #xx:8,@-ERd	0	1	7	0	1	1	1	6	C	0	rd	1	0	0	0	A	0	0	0	1	x	x	x	x	x	x	x	x	x	x									
	SUB,B #xx:8,@(d:2,ERd)	0	1	7	0	1	d	d	6	8	0	rd	1	0	0	0	A	0	0	0	1	x	x	x	x	x	x	x	x	x	x									
	SUB,B #xx:8,@(d:16,ERd)	0	1	7	0	1	0	0	6	E	0	rd	1	0	0	0	d	A	0	0	0	1	x	x	x	x	x	x	x	x	x									
	SUB,B #xx:8,@(d:32,ERd)	7	8	0	rd	0	1	0	0	6	A	0	0	1	0	1	0	0	d	d	A	0	0	0	1	x	x	x	x	x	x	x								
	SUB,B #xx:8,@(d:16,Rd,B)	0	1	7	0	1	0	1	6	E	0	rd	1	0	0	0	d	A	0	0	0	1	x	x	x	x	x	x	x	x	x									



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
SUB	SUB.B #xx:8,@(d:16,Rd,W)	0	1	7	0 1 1 0	6	E	0	rd	1 0 0 0	d	A	0 0 0 1	x x x x	x x x x	
	SUB.B #xx:8,@(d:16,ERd,L)	0	1	7	0 1 1 1	6	E	0	rd	1 0 0 0	d	A	0 0 0 1	x x x x	x x x x	
	SUB.B #xx:8,@(d:32,Rd,B)	7	8	0	rd	0 1 0 1	6	A	0 0 1 0	1 0 0 0	d d	A	0 0 0 1	x x x x	x x x x	
	SUB.B #xx:8,@(d:32,Rd,W)	7	8	0	rd	0 1 1 0	6	A	0 0 1 0	1 0 0 0	d d	A	0 0 0 1	x x x x	x x x x	
	SUB.B #xx:8,@(d:32,ERd,L)	7	8	0	rd	0 1 1 1	6	A	0 0 1 0	1 0 0 0	d d	A	0 0 0 1	x x x x	x x x x	
	SUB.B #xx:8,@aa:8					7	F	a a a a	a a a a			A	0 0 0 1	x x x x	x x x x	
	SUB.B #xx:8,@aa:16					6	A	0 0 0 1	1 0 0 0	a		A	0 0 0 1	x x x x	x x x x	
	SUB.B #xx:8,@aa:32					6	A	0 0 1 1	1 0 0 0	a a		A	0 0 0 1	x x x x	x x x x	
	SUB.B Rs,Rd											1	8	rs	rd	
	SUB.B Rs,@ERd					7	D	0	rd	0 0 0 0		1	8	rs	0 0 0 0	
	SUB.B Rs,@ERd+					0	1	7	1 0 0 1		1 0 0 0	0	rd	0 0 1 1	rs	
	SUB.B Rs,@ERd-					0	1	7	1 0 0 1		1 0 1 0	0	rd	0 0 1 1	rs	
	SUB.B Rs,@+ERd					0	1	7	1 0 0 1		1 0 0 1	0	rd	0 0 1 1	rs	
	SUB.B Rs,@-ERd					0	1	7	1 0 0 1		1 0 1 1	0	rd	0 0 1 1	rs	
	SUB.B Rs,@(d:2,ERd)					0	1	7	1 0 0 1		0 0 d d	0	rd	0 0 1 1	rs	
	SUB.B Rs,@(d:16,ERd)					0	1	7	1 0 0 1		1 1 0 0	0	rd	0 0 1 1	rs	d
	SUB.B Rs,@(d:32,ERd)					0	1	7	1 0 0 1		1 1 0 0	1	rd	0 0 1 1	rs	d d
	SUB.B Rs,@(d:16,Rd,B)					0	1	7	1 0 0 1		1 1 0 1	0	rd	0 0 1 1	rs	d
	SUB.B Rs,@(d:16,Rd,W)					0	1	7	1 0 0 1		1 1 1 0	0	rd	0 0 1 1	rs	d
	SUB.B Rs,@(d:16,ERd,L)					0	1	7	1 0 0 1		1 1 1 1	0	rd	0 0 1 1	rs	d
	SUB.B Rs,@(d:32,Rd,B)					0	1	7	1 0 0 1		1 1 0 1	1	rd	0 0 1 1	rs	d d
	SUB.B Rs,@(d:32,Rd,W)					0	1	7	1 0 0 1		1 1 1 0	1	rd	0 0 1 1	rs	d d
	SUB.B Rs,@(d:32,ERd,L)					0	1	7	1 0 0 1		1 1 1 1	1	rd	0 0 1 1	rs	d d
	SUB.B Rs,@aa:8					7	F	a a a a	a a a a			1	8	rs	0 0 0 0	
	SUB.B Rs,@aa:16					6	A	0 0 0 1	1 0 0 0	a		1	8	rs	0 0 0 0	
	SUB.B Rs,@aa:32					6	A	0 0 1 1	1 0 0 0	a a		1	8	rs	0 0 0 0	
	SUB.B @ERs,Rd					7	C	0	rs	0 0 0 0		1	8	0 0 0 0	rd	
SUB.B @ERs+,Rd					0	1	7	1 0 1 0		1 0 0 0	0	rs	0 0 1 1	rd		
SUB.B @ERs-,Rd					0	1	7	1 0 1 0		1 0 1 0	0	rs	0 0 1 1	rd		
SUB.B @+ERs,Rd					0	1	7	1 0 1 0		1 0 0 1	0	rs	0 0 1 1	rd		
SUB.B @-ERs,Rd					0	1	7	1 0 1 0		1 0 1 1	0	rs	0 0 1 1	rd		
SUB.B @(d:2,ERs),Rd					0	1	7	1 0 1 0		0 0 d d	0	rs	0 0 1 1	rd		
SUB.B @(d:16,ERs),Rd					0	1	7	1 0 1 0		1 1 0 0	0	rs	0 0 1 1	rd	d	
SUB.B @(d:32,ERs),Rd					0	1	7	1 0 1 0		1 1 0 0	1	rs	0 0 1 1	rd	d d	

Instruction	Mnemonic	Opcode				EA Ex- tension	Opcode				EA Extension																							
		15	8	7	0		15	8	7	0																								
SUB	SUB.B @(d:16,Rs.B),Rd				0	1		7	1	0	1	0	1	0			1	1	0	1	0	rs	0	0	1	1	rd	d						
	SUB.B @(d:16,Rs.W),Rd				0	1		7	1	0	1	0	1	0			1	1	1	0	0	rs	0	0	1	1	rd	d						
	SUB.B @(d:16,ERs.L),Rd				0	1		7	1	0	1	0	1	0			1	1	1	1	0	rs	0	0	1	1	rd	d						
	SUB.B @(d:32,Rs.B),Rd				0	1		7	1	0	1	0	1	0			1	1	0	1	1	rs	0	0	1	1	rd	d d						
	SUB.B @(d:32,Rs.W),Rd				0	1		7	1	0	1	0	1	0			1	1	1	0	1	rs	0	0	1	1	rd	d d						
	SUB.B @(d:32,ERs.L),Rd				0	1		7	1	0	1	0	1	0			1	1	1	1	1	rs	0	0	1	1	rd	d d						
	SUB.B @aa:8,Rd				7		E	a	a	a	a	a	a	a	a			1																
	SUB.B @aa:16,Rd				6		A	0	0	0	1	0	0	0	0	a			1															
	SUB.B @aa:32,Rd				6		A	0	0	1	1	0	0	0	0	a	a			1														
	SUB.B @ERs,@ERd				7		C	0	rs	0	1	0	1	0	1			0	0	0	0	0	rd	0	0	1	1	0	0	0	0			
	SUB.B @ERs,@ERd+				7		C	0	rs	0	1	0	1	0	1			1	0	0	0	0	rd	0	0	1	1	0	0	0	0			
	SUB.B @ERs,@ERd-				7		C	0	rs	0	1	0	1	0	1			1	0	1	0	0	rd	0	0	1	1	0	0	0	0			
	SUB.B @ERs,@+ERd				7		C	0	rs	0	1	0	1	0	1			1	0	0	1	0	rd	0	0	1	1	0	0	0	0			
	SUB.B @ERs,@-ERd				7		C	0	rs	0	1	0	1	0	1			1	0	1	1	0	rd	0	0	1	1	0	0	0	0			
	SUB.B @ERs,@(d:2,ERd)				7		C	0	rs	0	1	0	1	0	1			0	0	d	d	0	rd	0	0	1	1	0	0	0	0			
	SUB.B @ERs,@(d:16,ERd)				7		C	0	rs	0	1	0	1	0	1			1	1	0	0	0	rd	0	0	1	1	0	0	0	0	d		
	SUB.B @ERs,@(d:32,ERd)				7		C	0	rs	0	1	0	1	0	1			1	1	0	0	1	rd	0	0	1	1	0	0	0	0	d d		
	SUB.B @ERs,@(d:16,Rd.B)				7		C	0	rs	0	1	0	1	0	1			1	1	0	1	0	rd	0	0	1	1	0	0	0	0	d		
	SUB.B @ERs,@(d:16,Rd.W)				7		C	0	rs	0	1	0	1	0	1			1	1	1	0	0	rd	0	0	1	1	0	0	0	0	d		
	SUB.B @ERs,@(d:16,ERd.L)				7		C	0	rs	0	1	0	1	0	1			1	1	1	1	0	rd	0	0	1	1	0	0	0	0	d		
	SUB.B @ERs,@(d:32,Rd.B)				7		C	0	rs	0	1	0	1	0	1			1	1	0	1	1	rd	0	0	1	1	0	0	0	0	d d		
	SUB.B @ERs,@(d:32,Rd.W)				7		C	0	rs	0	1	0	1	0	1			1	1	1	0	1	rd	0	0	1	1	0	0	0	0	d d		
	SUB.B @ERs,@(d:32,ERd.L)				7		C	0	rs	0	1	0	1	0	1			1	1	1	1	1	rd	0	0	1	1	0	0	0	0	d d		
	SUB.B @ERs,@aa:16				7		C	0	rs	0	1	0	1	0	1			0	1	0	0	0	0	0	0	0	1	1	0	0	0	a		
	SUB.B @ERs,@aa:32				7		C	0	rs	0	1	0	1	0	1			0	1	0	0	1	0	0	0	0	0	1	1	0	0	a a		
	SUB.B @ERs+,@ERd		0		1		7		0	1	0	0	0	6		C	0	rs	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	
	SUB.B @ERs+,@ERd+		0		1		7		0	1	0	0	0	6		C	0	rs	1	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0
	SUB.B @ERs+,@ERd-		0		1		7		0	1	0	0	0	6		C	0	rs	1	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0
	SUB.B @ERs+,@+ERd		0		1		7		0	1	0	0	0	6		C	0	rs	1	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0
	SUB.B @ERs+,@-ERd		0		1		7		0	1	0	0	0	6		C	0	rs	1	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0
	SUB.B @ERs+,@(d:2,ERd)		0		1		7		0	1	0	0	0	6		C	0	rs	1	1	0	0	0	d	d	0	0	0	1	1	0	0	0	0
	SUB.B @ERs+,@(d:16,ERd)		0		1		7		0	1	0	0	0	6		C	0	rs	1	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0
SUB.B @ERs+,@(d:32,ERd)		0		1		7		0	1	0	0	0	6		C	0	rs	1	1	0	0	1	0	0	1	0	0	1	1	0	0	0	0	d d
SUB.B @ERs+,@(d:16,Rd.B)		0		1		7		0	1	0	0	0	6		C	0	rs	1	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	d



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
SUB	SUB.B @ERs+,@(d:16,Rd,W)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @ERs+,@(d:16,ERd,L)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @ERs+,@(d:32,Rd,B)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @ERs+,@(d:32,Rd,W)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @ERs+,@(d:32,ERd,L)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @ERs+,@aa:16	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0	0 1 0 0	0	0 0 0 0	0 0 1 1	0 0 0 0	a
	SUB.B @ERs+,@aa:32	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0	0 1 0 0	1	0 0 0 0	0 0 1 1	0 0 0 0	a a
	SUB.B @ERs-,@ERd	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @ERs-,@ERd+	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @ERs-,@ERd-	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 0 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @ERs-,@+ERd	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @ERs-,@-ERd	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 0 1 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @ERs-,@(d2,ERd)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	0 0 d d	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @ERs-,@(d:16,ERd)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @ERs-,@(d:32,ERd)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @ERs-,@(d:16,Rd,B)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @ERs-,@(d:16,Rd,W)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @ERs-,@(d:16,ERd,L)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @ERs-,@(d:32,Rd,B)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @ERs-,@(d:32,Rd,W)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @ERs-,@(d:32,ERd,L)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @ERs-,@aa:16	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	0 1 0 0	0	0 0 0 0	0 0 1 1	0 0 0 0	a
	SUB.B @ERs-,@aa:32	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	0 1 0 0	1	0 0 0 0	0 0 1 1	0 0 0 0	a a
	SUB.B @+ERs,@ERd	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @+ERs,@ERd+	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @+ERs,@ERd-	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 0 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @+ERs,@+ERd	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @+ERs,@-ERd	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 0 1 1	0	rd	0 0 1 1	0 0 0 0	
SUB.B @+ERs,@(d2,ERd)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	0 0 d d	0	rd	0 0 1 1	0 0 0 0		
SUB.B @+ERs,@(d:16,ERd)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d	
SUB.B @+ERs,@(d:32,ERd)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d	
SUB.B @+ERs,@(d:16,Rd,B)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d	
SUB.B @+ERs,@(d:16,Rd,W)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d	
SUB.B @+ERs,@(d:16,ERd,L)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d	

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
SUB	SUB.B @+ERs, @(d:32,Rd,B)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @+ERs, @(d:32,Rd,W)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @+ERs, @(d:32,ERd,L)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @+ERs, @aa:16	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	0 1 0 0	0	rd	0 0 1 1	0 0 0 0	a
	SUB.B @+ERs, @aa:32	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	0 1 0 0	1	0 0 0 0	0 0 1 1	0 0 0 0	a a
	SUB.B @-ERs, @ERd	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @-ERs, @ERd+	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @-ERs, @ERd-	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	1 0 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @-ERs, @+ERd	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @-ERs, @-ERd	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	1 0 1 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @-ERs, @(d:2,ERd)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	0 0 d d	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @-ERs, @(d:16,ERd)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @-ERs, @(d:32,ERd)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @-ERs, @(d:16,Rd,B)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @-ERs, @(d:16,Rd,W)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @-ERs, @(d:16,ERd,L)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @-ERs, @(d:32,Rd,B)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @-ERs, @(d:32,Rd,W)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @-ERs, @(d:32,ERd,L)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @-ERs, @aa:16	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	0 1 0 0	0	0 0 0 0	0 0 1 1	0 0 0 0	a
	SUB.B @-ERs, @aa:32	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	0 1 0 0	1	0 0 0 0	0 0 1 1	0 0 0 0	a a
	SUB.B @(d:2,ERs), @ERd	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:2,ERs), @ERd+	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:2,ERs), @ERd-	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	1 0 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:2,ERs), @+ERd	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:2,ERs), @-ERd	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	1 0 1 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:2,ERs), @(d:2,ERd)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	0 0 d d	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:2,ERs), @(d:16,ERd)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @(d:2,ERs), @(d:32,ERd)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @(d:2,ERs), @(d:16,Rd,B)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @(d:2,ERs), @(d:16,Rd,W)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @(d:2,ERs), @(d:16,ERd,L)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
SUB.B @(d:2,ERs), @(d:32,Rd,B)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d	
SUB.B @(d:2,ERs), @(d:32,Rd,W)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d	



Instruction	Mnemonic	Opcode						EA Ex- tension	Opcode						EA Extension																							
		15	8	7	0	15	8		7	0	15	8	7	0																								
SUB	SUB.B @(d:2,ERs),@(d:32,ERd.L)	0	1	7	0	1	d	d	6	8	0	rs	1	1	0	0		1	1	1	1	rd	0	0	1	1	0	0	0	0	0	d	d					
	SUB.B @(d:2,ERs),@aa:16	0	1	7	0	1	d	d	6	8	0	rs	1	1	0	0		0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	a			
	SUB.B @(d:2,ERs),@aa:32	0	1	7	0	1	d	d	6	8	0	rs	1	1	0	0		0	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	a	a		
	SUB.B @(d:16,ERs),@ERd	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	0	0	0	0	0	rd	0	0	1	1	0	0	0	0	0						
	SUB.B @(d:16,ERs),@ERd+	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	0	0	0	0	rd	0	0	1	1	0	0	0	0	0						
	SUB.B @(d:16,ERs),@ERd-	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	0	1	0	0	rd	0	0	1	1	0	0	0	0	0						
	SUB.B @(d:16,ERs),@+ERd	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	0	0	1	0	rd	0	0	1	1	0	0	0	0	0						
	SUB.B @(d:16,ERs),@-ERd	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	0	1	1	0	rd	0	0	1	1	0	0	0	0	0						
	SUB.B @(d:16,ERs),@(d:2,ERd)	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	0	0	d	d	0	rd	0	0	1	1	0	0	0	0	0						
	SUB.B @(d:16,ERs),@(d:16,ERd)	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	1	0	0	0	rd	0	0	1	1	0	0	0	0	0	d					
	SUB.B @(d:16,ERs),@(d:32,ERd)	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	1	0	0	1	rd	0	0	1	1	0	0	0	0	0	d	d				
	SUB.B @(d:16,ERs),@(d:16,Rd.B)	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	1	0	1	0	rd	0	0	1	1	0	0	0	0	0	d					
	SUB.B @(d:16,ERs),@(d:16,Rd.W)	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	1	1	0	0	rd	0	0	1	1	0	0	0	0	0	d					
	SUB.B @(d:16,ERs),@(d:16,ERd.L)	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	1	1	1	0	rd	0	0	1	1	0	0	0	0	0	d					
	SUB.B @(d:16,ERs),@(d:32,Rd.B)	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	1	0	1	1	rd	0	0	1	1	0	0	0	0	0	d	d				
	SUB.B @(d:16,ERs),@(d:32,Rd.W)	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	1	1	0	1	rd	0	0	1	1	0	0	0	0	0	d	d				
	SUB.B @(d:16,ERs),@(d:32,ERd.L)	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	1	1	1	1	1	rd	0	0	1	1	0	0	0	0	0	d	d				
	SUB.B @(d:16,ERs),@aa:16	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	0	1	0	0	0	rd	0	0	1	1	0	0	0	0	0	a					
	SUB.B @(d:16,ERs),@aa:32	0	1	7	0	1	0	0	6	E	0	rs	1	1	0	0	d	0	1	0	0	1	rd	0	0	1	1	0	0	0	0	0	a	a				
	SUB.B @(d:32,ERs),@ERd	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	1	1	0	0	d	d	0	0	0	0	0	rd	0	0	1	1	0	0	0	0	0		
	SUB.B @(d:32,ERs),@ERd+	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	1	1	0	0	d	d	1	0	0	0	0	rd	0	0	1	1	0	0	0	0	0		
	SUB.B @(d:32,ERs),@ERd-	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	1	1	0	0	d	d	1	0	1	0	0	rd	0	0	1	1	0	0	0	0	0		
	SUB.B @(d:32,ERs),@+ERd	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	1	1	0	0	d	d	1	0	0	1	0	rd	0	0	1	1	0	0	0	0	0		
	SUB.B @(d:32,ERs),@-ERd	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	1	1	0	0	d	d	1	0	1	1	0	rd	0	0	1	1	0	0	0	0	0		
	SUB.B @(d:32,ERs),@(d:2,ERd)	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	1	1	0	0	d	d	0	0	d	d	0	rd	0	0	1	1	0	0	0	0	0		
	SUB.B @(d:32,ERs),@(d:16,ERd)	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	1	1	0	0	d	d	1	1	0	0	0	rd	0	0	1	1	0	0	0	0	0	d	
	SUB.B @(d:32,ERs),@(d:32,ERd)	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	1	1	0	0	d	d	1	1	0	0	1	rd	0	0	1	1	0	0	0	0	0	d	d
	SUB.B @(d:32,ERs),@(d:16,Rd.B)	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	1	1	0	0	d	d	1	1	0	1	0	rd	0	0	1	1	0	0	0	0	0	d	
	SUB.B @(d:32,ERs),@(d:16,Rd.W)	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	1	1	0	0	d	d	1	1	1	0	0	rd	0	0	1	1	0	0	0	0	0	d	
	SUB.B @(d:32,ERs),@(d:16,ERd.L)	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	1	1	0	0	d	d	1	1	1	1	0	rd	0	0	1	1	0	0	0	0	0	d	
	SUB.B @(d:32,ERs),@(d:32,Rd.B)	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	1	1	0	0	d	d	1	1	0	1	1	rd	0	0	1	1	0	0	0	0	0	d	d
	SUB.B @(d:32,ERs),@(d:32,Rd.W)	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	1	1	0	0	d	d	1	1	1	0	1	rd	0	0	1	1	0	0	0	0	0	d	d
SUB.B @(d:32,ERs),@(d:32,ERd.L)	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	1	1	0	0	d	d	1	1	1	1	1	rd	0	0	1	1	0	0	0	0	0	d	d	
SUB.B @(d:32,ERs),@aa:16	7	8	0	rs	0	1	0	0	6	A	0	0	1	0	1	1	0	0	d	d	0	1	0	0	0	rd	0	0	1	1	0	0	0	0	0	a		

Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension	
		15	8	7	0	15	8		7	0	15	8	7	0		
SUB	SUB.B @(d:32,ERs),@aa:32	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	1 0 0 0	0 0 1 1	0 0 0 0	a a
	SUB.B @(d:16,Rs.B),@ERd	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	0 0 0 0	0 rd	0 0 1 1	0 0 0 0		
	SUB.B @(d:16,Rs.B),@ERd+	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 0 0 0	0 rd	0 0 1 1	0 0 0 0		
	SUB.B @(d:16,Rs.B),@ERd-	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 0 1 0	0 rd	0 0 1 1	0 0 0 0		
	SUB.B @(d:16,Rs.B),@+ERd	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 0 0 1	0 rd	0 0 1 1	0 0 0 0		
	SUB.B @(d:16,Rs.B),@-ERd	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 0 1 1	0 rd	0 0 1 1	0 0 0 0		
	SUB.B @(d:16,Rs.B),@(d:2,ERd)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	0 0 d d	0 rd	0 0 1 1	0 0 0 0		
	SUB.B @(d:16,Rs.B),@(d:16,ERd)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 1 0 0	0 rd	0 0 1 1	0 0 0 0	d	
	SUB.B @(d:16,Rs.B),@(d:32,ERd)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 1 0 0	1 rd	0 0 1 1	0 0 0 0	d d	
	SUB.B @(d:16,Rs.B),@(d:16,Rd.B)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 1 0 1	0 rd	0 0 1 1	0 0 0 0	d	
	SUB.B @(d:16,Rs.B),@(d:16,Rd.W)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 1 1 0	0 rd	0 0 1 1	0 0 0 0	d	
	SUB.B @(d:16,Rs.B),@(d:16,ERd.L)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 1 1 1	0 rd	0 0 1 1	0 0 0 0	d	
	SUB.B @(d:16,Rs.B),@(d:32,Rd.B)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 1 0 1	1 rd	0 0 1 1	0 0 0 0	d d	
	SUB.B @(d:16,Rs.B),@(d:32,Rd.W)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 1 1 0	1 rd	0 0 1 1	0 0 0 0	d d	
	SUB.B @(d:16,Rs.B),@(d:32,ERd.L)	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	1 1 1 1	1 rd	0 0 1 1	0 0 0 0	d d	
	SUB.B @(d:16,Rs.B),@aa:16	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 0 1 1	0 0 0 0	a	
	SUB.B @(d:16,Rs.B),@aa:32	0	1	7	0 1 0 1	6	E	0 rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 0 1 1	0 0 0 0	a a	
	SUB.B @(d:16,Rs.W),@ERd	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d	0 0 0 0	0 rd	0 0 1 1	0 0 0 0		
	SUB.B @(d:16,Rs.W),@ERd+	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d	1 0 0 0	0 rd	0 0 1 1	0 0 0 0		
	SUB.B @(d:16,Rs.W),@ERd-	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d	1 0 1 0	0 rd	0 0 1 1	0 0 0 0		
	SUB.B @(d:16,Rs.W),@+ERd	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d	1 0 0 1	0 rd	0 0 1 1	0 0 0 0		
	SUB.B @(d:16,Rs.W),@-ERd	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d	1 0 1 1	0 rd	0 0 1 1	0 0 0 0		
	SUB.B @(d:16,Rs.W),@(d:2,ERd)	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d	0 0 d d	0 rd	0 0 1 1	0 0 0 0		
	SUB.B @(d:16,Rs.W),@(d:16,ERd)	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d	1 1 0 0	0 rd	0 0 1 1	0 0 0 0	d	
	SUB.B @(d:16,Rs.W),@(d:32,ERd)	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d	1 1 0 0	1 rd	0 0 1 1	0 0 0 0	d d	
	SUB.B @(d:16,Rs.W),@(d:16,Rd.B)	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d	1 1 0 1	0 rd	0 0 1 1	0 0 0 0	d	
	SUB.B @(d:16,Rs.W),@(d:16,Rd.W)	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d	1 1 1 0	0 rd	0 0 1 1	0 0 0 0	d	
	SUB.B @(d:16,Rs.W),@(d:16,ERd.L)	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d	1 1 1 1	0 rd	0 0 1 1	0 0 0 0	d	
	SUB.B @(d:16,Rs.W),@(d:32,Rd.B)	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d	1 1 0 1	1 rd	0 0 1 1	0 0 0 0	d d	
	SUB.B @(d:16,Rs.W),@(d:32,Rd.W)	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d	1 1 1 0	1 rd	0 0 1 1	0 0 0 0	d d	
	SUB.B @(d:16,Rs.W),@(d:32,ERd.L)	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d	1 1 1 1	1 rd	0 0 1 1	0 0 0 0	d d	
	SUB.B @(d:16,Rs.W),@aa:16	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 0 1 1	0 0 0 0	a	
SUB.B @(d:16,Rs.W),@aa:32	0	1	7	0 1 1 0	6	E	0 rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 0 1 1	0 0 0 0	a a		
SUB.B @(d:16,ERs.L),@ERd	0	1	7	0 1 1 1	6	E	0 rs	1 1 0 0	d	0 0 0 0	0 rd	0 0 1 1	0 0 0 0			



Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
SUB	SUB.B @(d:16,ERs.L),@ERd+	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:16,ERs.L),@ERd-	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:16,ERs.L),@+ERd	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:16,ERs.L),@-ERd	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:16,ERs.L),@(d:2,ERd)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:16,ERs.L),@(d:16,ERd)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @(d:16,ERs.L),@(d:32,ERd)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @(d:16,ERs.L),@(d:16,Rd.B)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @(d:16,ERs.L),@(d:16,Rd.W)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @(d:16,ERs.L),@(d:16,ERd.L)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @(d:16,ERs.L),@(d:32,Rd.B)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @(d:16,ERs.L),@(d:32,Rd.W)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @(d:16,ERs.L),@(d:32,ERd.L)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @(d:16,ERs.L),@aa:16	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 0 1 1	0 0 0 0	a	
	SUB.B @(d:16,ERs.L),@aa:32	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 0 1 1	0 0 0 0	a a	
	SUB.B @(d:32,Rs.B),@ERd	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:32,Rs.B),@ERd+	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:32,Rs.B),@ERd-	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:32,Rs.B),@+ERd	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:32,Rs.B),@-ERd	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:32,Rs.B),@(d:2,ERd)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:32,Rs.B),@(d:16,ERd)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @(d:32,Rs.B),@(d:32,ERd)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @(d:32,Rs.B),@(d:16,Rd.B)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @(d:32,Rs.B),@(d:16,Rd.W)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @(d:32,Rs.B),@(d:16,ERd.L)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @(d:32,Rs.B),@(d:32,Rd.B)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @(d:32,Rs.B),@(d:32,Rd.W)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @(d:32,Rs.B),@aa:16	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	0 0 0 0	0 0 1 1	0 0 0 0	a	
	SUB.B @(d:32,Rs.B),@aa:32	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	1 0 0 0	0 0 1 1	0 0 0 0	a a	
	SUB.B @(d:32,Rs.W),@ERd	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:32,Rs.W),@ERd+	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:32,Rs.W),@ERd-	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 0 1 1	0 0 0 0	

Instruction	Mnemonic	Opcode						EA Ex- tension	Opcode						EA Extension		
		15	8	7	0	15	8		7	0	15	8	7	0			
SUB	SUB.B @(d:32.Rs.W),@+ERd	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:32.Rs.W),@-ERd	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:32.Rs.W),@(d:2.ERd)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:32.Rs.W),@(d:16.ERd)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @(d:32.Rs.W),@(d:32.ERd)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @(d:32.Rs.W),@(d:16.Rd.B)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @(d:32.Rs.W),@(d:16.Rd.W)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @(d:32.Rs.W),@(d:16.ERd.L)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @(d:32.Rs.W),@(d:32.Rd.B)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @(d:32.Rs.W),@(d:32.Rd.W)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @(d:32.Rs.W),@(d:32.ERd.L)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @(d:32.Rs.W),@aa:16	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 0 1 1	0 0 0 0	a
	SUB.B @(d:32.Rs.W),@aa:32	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 0 1 1	0 0 0 0	a a
	SUB.B @(d:32.ERs.L),@ERd	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:32.ERs.L),@ERd+	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:32.ERs.L),@ERd-	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:32.ERs.L),@+ERd	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:32.ERs.L),@-ERd	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:32.ERs.L),@(d:2.ERd)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 0 1 1	0 0 0 0	
	SUB.B @(d:32.ERs.L),@(d:16.ERd)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @(d:32.ERs.L),@(d:32.ERd)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @(d:32.ERs.L),@(d:16.Rd.B)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @(d:32.ERs.L),@(d:16.Rd.W)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @(d:32.ERs.L),@(d:16.ERd.L)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.B @(d:32.ERs.L),@(d:32.Rd.B)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @(d:32.ERs.L),@(d:32.Rd.W)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @(d:32.ERs.L),@(d:32.ERd.L)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.B @(d:32.ERs.L),@aa:16	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 0 1 1	0 0 0 0	a
	SUB.B @(d:32.ERs.L),@aa:32	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 0 1 1	0 0 0 0	a a
	SUB.B @aa:16,@ERd						6	A	0 0 0 1	0 1 0 1	a		0 0 0 0	0	rd	0 0 1 1	0 0 0 0
SUB.B @aa:16,@ERd+						6	A	0 0 0 1	0 1 0 1	a		1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
SUB.B @aa:16,@ERd-						6	A	0 0 0 1	0 1 0 1	a		1 0 1 0	0	rd	0 0 1 1	0 0 0 0	
SUB.B @aa:16,@+ERd						6	A	0 0 0 1	0 1 0 1	a		1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
SUB.B @aa:16,@-ERd						6	A	0 0 0 1	0 1 0 1	a		1 0 1 1	0	rd	0 0 1 1	0 0 0 0	



Instruction	Mnemonic	Opcode			Opcode			EA Extension	Opcode			EA Extension		
		15	8	7	0	15	8		7	0	15		8	7
SUB	SUB.B @aa:16,@(d:2,ERd)				6	A	0 0 0 0 1	0 1 0 1 a		0 0 d d	0 rd	0 0 1 1	0 0 0 0	
	SUB.B @aa:16,@(d:16,ERd)				6	A	0 0 0 0 1	0 1 0 1 a		1 1 0 0	0 rd	0 0 1 1	0 0 0 0	d
	SUB.B @aa:16,@(d:32,ERd)				6	A	0 0 0 0 1	0 1 0 1 a		1 1 0 0	1 rd	0 0 1 1	0 0 0 0	d d
	SUB.B @aa:16,@(d:16,Rd,B)				6	A	0 0 0 0	0 1 0 1 a		1 1 0 1	0 rd	0 0 1 1	0 0 0 0	d
	SUB.B @aa:16,@(d:16,Rd,W)				6	A	0 0 0 0 1	0 1 0 1 a		1 1 1 0	0 rd	0 0 1 1	0 0 0 0	d
	SUB.B @aa:16,@(d:16,ERd,L)				6	A	0 0 0 0 1	0 1 0 1 a		1 1 1 1	0 rd	0 0 1 1	0 0 0 0	d
	SUB.B @aa:16,@(d:32,Rd,B)				6	A	0 0 0 0	0 1 0 1 a		1 1 0 1	1 rd	0 0 1 1	0 0 0 0	d d
	SUB.B @aa:16,@(d:32,Rd,W)				6	A	0 0 0 0 1	0 1 0 1 a		1 1 1 0	1 rd	0 0 1 1	0 0 0 0	d d
	SUB.B @aa:16,@(d:32,ERd,L)				6	A	0 0 0 0 1	0 1 0 1 a		1 1 1 1	1 rd	0 0 1 1	0 0 0 0	d d
	SUB.B @aa:16,@aa:16				6	A	0 0 0 0 1	0 1 0 1 a		0 1 0 0	0 0 0 0	0 0 1 1	0 0 0 0	a
	SUB.B @aa:16,@aa:32				6	A	0 0 0 0 1	0 1 0 1 a		0 1 0 0	1 0 0 0	0 0 1 1	0 0 0 0	a a
	SUB.B @aa:32,@ERd				6	A	0 0 1 1	0 1 0 1 a a		0 0 0 0	0 rd	0 0 1 1	0 0 0 0	
	SUB.B @aa:32,@ERd+				6	A	0 0 1 1	0 1 0 1 a a		1 0 0 0	0 rd	0 0 1 1	0 0 0 0	
	SUB.B @aa:32,@ERd-				6	A	0 0 1 1	0 1 0 1 a a		1 0 1 0	0 rd	0 0 1 1	0 0 0 0	
	SUB.B @aa:32,@+ERd				6	A	0 0 1 1	0 1 0 1 a a		1 0 0 1	0 rd	0 0 1 1	0 0 0 0	
	SUB.B @aa:32,@-ERd				6	A	0 0 1 1	0 1 0 1 a a		1 0 1 1	0 rd	0 0 1 1	0 0 0 0	
	SUB.B @aa:32,@(d:2,ERd)				6	A	0 0 1 1	0 1 0 1 a a		0 0 d d	0 rd	0 0 1 1	0 0 0 0	
	SUB.B @aa:32,@(d:16,ERd)				6	A	0 0 1 1	0 1 0 1 a a		1 1 0 0	0 rd	0 0 1 1	0 0 0 0	d
	SUB.B @aa:32,@(d:32,ERd)				6	A	0 0 1 1	0 1 0 1 a a		1 1 0 0	1 rd	0 0 1 1	0 0 0 0	d d
	SUB.B @aa:32,@(d:16,Rd,B)				6	A	0 0 1 1	0 1 0 1 a a		1 1 0 1	0 rd	0 0 1 1	0 0 0 0	d
	SUB.B @aa:32,@(d:16,Rd,W)				6	A	0 0 1 1	0 1 0 1 a a		1 1 1 0	0 rd	0 0 1 1	0 0 0 0	d
	SUB.B @aa:32,@(d:16,ERd,L)				6	A	0 0 1 1	0 1 0 1 a a		1 1 1 1	0 rd	0 0 1 1	0 0 0 0	d
	SUB.B @aa:32,@(d:32,Rd,B)				6	A	0 0 1 1	0 1 0 1 a a		1 1 0 1	1 rd	0 0 1 1	0 0 0 0	d d
	SUB.B @aa:32,@(d:32,Rd,W)				6	A	0 0 1 1	0 1 0 1 a a		1 1 1 0	1 rd	0 0 1 1	0 0 0 0	d d
SUB.B @aa:32,@(d:32,ERd,L)				6	A	0 0 1 1	0 1 0 1 a a		1 1 1 1	1 rd	0 0 1 1	0 0 0 0	d d	
SUB.B @aa:32,@aa:16				6	A	0 0 1 1	0 1 0 1 a a		0 1 0 0	0 0 0 0	0 0 1 1	0 0 0 0	a	
SUB.B @aa:32,@aa:32				6	A	0 0 1 1	0 1 0 1 a a		0 1 0 0	1 0 0 0	0 0 1 1	0 0 0 0	a a	
SUB.W #xc:3,Rd									1	A	0 x x x	rd		
SUB.W #xc:16,Rd									7	9	0 0 1 1	rd	x	
SUB.W #xc:3,@ERd				7	D	1	rd	0 0 0 0	1	A	0 x x x	0 0 0 0		
SUB.W #xc:3,@aa:16				6	B	0 0 0 0 1	1 0 0 0 a		1	A	0 x x x	0 0 0 0		
SUB.W #xc:3,@aa:32				6	B	0 0 0 1	1 0 0 0 a a		1	A	0 x x x	0 0 0 0		
SUB.W #xc:16,@ERd				0	1	5	1 1 1 0		0 0 0 0	0 rd	0 0 1 1	0 0 0 0	x	
SUB.W #xc:16,@ERd+				0	1	5	1 1 1 0		1 0 0 0	0 rd	0 0 1 1	0 0 0 0	x	

Instruction	Mnemonic	Opcode			Opcode			EA Ex- tension	Opcode			EA Extension					
		15	8	7	0	15	8		7	0	15		8	7	0		
SUB	SUB.W #xx:16,@ERd				0	1	5	1 1 1 0		1 0 1 0	0	rd	0 0 1 1	0 0 0 0			x
	SUB.W #xx:16,@+ERd				0	1	5	1 1 1 0		1 0 0 1	0	rd	0 0 1 1	0 0 0 0			x
	SUB.W #xx:16,@-ERd				0	1	5	1 1 1 0		1 0 1 1	0	rd	0 0 1 1	0 0 0 0			x
	SUB.W #xx:16,@(d:2,ERd)				0	1	5	1 1 1 0		0 0 d d	0	rd	0 0 1 1	0 0 0 0			x
	SUB.W #xx:16,@(d:16,ERd)				0	1	5	1 1 1 0		1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d		x
	SUB.W #xx:16,@(d:32,ERd)				0	1	5	1 1 1 0		1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d	d	x
	SUB.W #xx:16,@(d:16,Rd,B)				0	1	5	1 1 1 0		1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d		x
	SUB.W #xx:16,@(d:16,Rd,W)				0	1	5	1 1 1 0		1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d		x
	SUB.W #xx:16,@(d:16,ERd,L)				0	1	5	1 1 1 0		1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d		x
	SUB.W #xx:16,@(d:32,Rd,B)				0	1	5	1 1 1 0		1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d	d	x
	SUB.W #xx:16,@(d:32,Rd,W)				0	1	5	1 1 1 0		1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d	d	x
	SUB.W #xx:16,@(d:32,ERd,L)				0	1	5	1 1 1 0		1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d	d	x
	SUB.W #xx:16,@aa:16				0	1	5	1 1 1 0		0 1 0 0	0 0 0 0		0 0 1 1	0 0 0 0	a		x
	SUB.W #xx:16,@aa:32				0	1	5	1 1 1 0		0 1 0 0	1 0 0 0		0 0 1 1	0 0 0 0	a	a	x
	SUB.W Rs,Rd										1	9	rs	rd			
	SUB.W Rs,@ERd				7	D	1	rd			1	9	rs	0 0 0 0			
	SUB.W Rs,@ERd+				0	1	5	1 0 0 1		1 0 0 0	0	rd	0 0 1 1	rs			
	SUB.W Rs,@ERd-				0	1	5	1 0 0 1		1 0 1 0	0	rd	0 0 1 1	rs			
	SUB.W Rs,@+ERd				0	1	5	1 0 0 1		1 0 0 1	0	rd	0 0 1 1	rs			
	SUB.W Rs,@-ERd				0	1	5	1 0 0 1		1 0 1 1	0	rd	0 0 1 1	rs			
	SUB.W Rs,@(d:2,ERd)				0	1	5	1 0 0 1		0 0 d d	0	rd	0 0 1 1	rs			
	SUB.W Rs,@(d:16,ERd)				0	1	5	1 0 0 1		1 1 0 0	0	rd	0 0 1 1	rs	d		
	SUB.W Rs,@(d:32,ERd)				0	1	5	1 0 0 1		1 1 0 0	1	rd	0 0 1 1	rs	d	d	
	SUB.W Rs,@(d:16,Rd,B)				0	1	5	1 0 0 1		1 1 0 1	0	rd	0 0 1 1	rs	d		
	SUB.W Rs,@(d:16,Rd,W)				0	1	5	1 0 0 1		1 1 1 0	0	rd	0 0 1 1	rs	d		
	SUB.W Rs,@(d:16,ERd,L)				0	1	5	1 0 0 1		1 1 1 1	0	rd	0 0 1 1	rs	d		
	SUB.W Rs,@(d:32,Rd,B)				0	1	5	1 0 0 1		1 1 0 1	1	rd	0 0 1 1	rs	d	d	
	SUB.W Rs,@(d:32,Rd,W)				0	1	5	1 0 0 1		1 1 1 0	1	rd	0 0 1 1	rs	d	d	
	SUB.W Rs,@(d:32,ERd,L)				0	1	5	1 0 0 1		1 1 1 1	1	rd	0 0 1 1	rs	d	d	
	SUB.W Rs,@aa:16				6	B	0 0 0 1	1 0 0 0	a		1	9	rs	0 0 0 0			
SUB.W Rs,@aa:32				6	B	0 0 1 1	1 0 0 0	a a		1	9	rs	0 0 0 0				
SUB.W @ERs,Rd				7	C	1	rs			1	9	0 0 0 0	rd				
SUB.W @ERs+,Rd				0	1	5	1 0 1 0		1 0 0 0	0	rs	0 0 1 1	rd				
SUB.W @ERs-,Rd				0	1	5	1 0 1 0		1 0 1 0	0	rs	0 0 1 1	rd				



Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0		15	8	7	0		
SUB	SUB.W @+ERs,Rd	0	1	5	1 0 1 0		1 0 0 1	0	rs	0 0 1 1	rd	
	SUB.W @-ERs,Rd	0	1	5	1 0 1 0		1 0 1 1	0	rs	0 0 1 1	rd	
	SUB.W @(d:2,ERs),Rd	0	1	5	1 0 1 0		0 0 d d	0	rs	0 0 1 1	rd	
	SUB.W @(d:16,ERs),Rd	0	1	5	1 0 1 0		1 1 0 0	0	rs	0 0 1 1	rd	d
	SUB.W @(d:32,ERs),Rd	0	1	5	1 0 1 0		1 1 0 0	1	rs	0 0 1 1	rd	d d
	SUB.W @(d:16,Rs,B),Rd	0	1	5	1 0 1 0		1 1 0 1	0	rs	0 0 1 1	rd	d
	SUB.W @(d:16,Rs,W),Rd	0	1	5	1 0 1 0		1 1 1 0	0	rs	0 0 1 1	rd	d
	SUB.W @(d:16,ERs,L),Rd	0	1	5	1 0 1 0		1 1 1 1	0	rs	0 0 1 1	rd	d
	SUB.W @(d:32,Rs,B),Rd	0	1	5	1 0 1 0		1 1 0 1	1	rs	0 0 1 1	rd	d d
	SUB.W @(d:32,Rs,W),Rd	0	1	5	1 0 1 0		1 1 1 0	1	rs	0 0 1 1	rd	d d
	SUB.W @(d:32,ERs,L),Rd	0	1	5	1 0 1 0		1 1 1 1	1	rs	0 0 1 1	rd	d d
	SUB.W @aa:16,Rd	6	B	0 0 0 1	0 0 0 0	a	1	9	0 0 0 0	rd		
	SUB.W @aa:32,Rd	6	B	0 0 1 1	0 0 0 0	a a	1	9	0 0 0 0	rd		
	SUB.W @ERs,@ERd	7	C	1 rs	0 1 0 1		0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @ERs,@ERd+	7	C	1 rs	0 1 0 1		1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @ERs,@ERd-	7	C	1 rs	0 1 0 1		1 0 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @ERs,@+ERd	7	C	1 rs	0 1 0 1		1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @ERs,@-ERd	7	C	1 rs	0 1 0 1		1 0 1 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @ERs,@(d:2,ERd)	7	C	1 rs	0 1 0 1		0 0 d d	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @ERs,@(d:16,ERd)	7	C	1 rs	0 1 0 1		1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @ERs,@(d:32,ERd)	7	C	1 rs	0 1 0 1		1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @ERs,@(d:16,Rd,B)	7	C	1 rs	0 1 0 1		1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @ERs,@(d:16,Rd,W)	7	C	1 rs	0 1 0 1		1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @ERs,@(d:16,ERd,L)	7	C	1 rs	0 1 0 1		1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @ERs,@(d:32,Rd,B)	7	C	1 rs	0 1 0 1		1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @ERs,@(d:32,Rd,W)	7	C	1 rs	0 1 0 1		1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @ERs,@(d:32,ERd,L)	7	C	1 rs	0 1 0 1		1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @ERs,@aa:16	7	C	1 rs	0 1 0 1		0 1 0 0	0 0 0 0	0 0 1 1	0 0 0 0	a	
SUB.W @ERs,@aa:32	7	C	1 rs	0 1 0 1		0 1 0 0	1 0 0 0	0 0 1 1	0 0 0 0	a a		
SUB.W @ERs+,@ERd	0	1	5	0 1 0 0	6	D	0 rs	1 1 0 0	0 0 0 0	0 0 rd	0 0 1 1	0 0 0 0
SUB.W @ERs+,@ERd+	0	1	5	0 1 0 0	6	D	0 rs	1 1 0 0	1 0 0 0	0 0 rd	0 0 1 1	0 0 0 0
SUB.W @ERs+,@ERd-	0	1	5	0 1 0 0	6	D	0 rs	1 1 0 0	1 0 1 0	0 0 rd	0 0 1 1	0 0 0 0
SUB.W @ERs+,@+ERd	0	1	5	0 1 0 0	6	D	0 rs	1 1 0 0	1 0 0 1	0 0 rd	0 0 1 1	0 0 0 0
SUB.W @ERs+,@-ERd	0	1	5	0 1 0 0	6	D	0 rs	1 1 0 0	1 0 1 1	0 0 rd	0 0 1 1	0 0 0 0

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension				
		15	8	7	0	15	8	7	0		15	8	7	0					
SUB	SUB.W @ERs+,@(d:2,ERd)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0	0	0	d	d	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @ERs+,@(d:16,ERd)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0	1	1	0 0	0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @ERs+,@(d:32,ERd)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0	1	1	0 0	1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @ERs+,@(d:16,Rd,B)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0	1	1	0 1	0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @ERs+,@(d:16,Rd,W)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0	1	1	1 0	0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @ERs+,@(d:16,ERd,L)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0	1	1	1 1	0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @ERs+,@(d:32,Rd,B)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0	1	1	0 1	1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @ERs+,@(d:32,Rd,W)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0	1	1	1 0	1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @ERs+,@(d:32,ERd,L)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0	1	1	1 1	1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @ERs+,@aa:16	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0	0	1	0 0	0	0	0 0 0 0	0 0 1 1	0 0 0 0	a
	SUB.W @ERs+,@aa:32	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0	0	1	0 0	1	0 0 0 0	1 0 0 0	0 0 1 1	0 0 0 0	a a
	SUB.W @ERs+,@ERd	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0	0	0	0 0	0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @ERs+,@ERd+	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0	1	0	0 0	0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @ERs+,@ERd-	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0	1	0	1 0	0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @ERs+,@+ERd	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0	1	0	0 1	0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @ERs+,@-ERd	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0	1	0	1 1	0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @ERs+,@(d:2,ERd)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0	0	0	d	d	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @ERs+,@(d:16,ERd)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0	1	1	0 0	0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @ERs+,@(d:32,ERd)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0	1	1	0 0	1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @ERs+,@(d:16,Rd,B)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0	1	1	0 1	0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @ERs+,@(d:16,Rd,W)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0	1	1	1 0	0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @ERs+,@(d:16,ERd,L)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0	1	1	1 1	0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @ERs+,@(d:32,Rd,B)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0	1	1	0 1	1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @ERs+,@(d:32,Rd,W)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0	1	1	1 0	1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @ERs+,@(d:32,ERd,L)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0	1	1	1 1	1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @ERs+,@aa:16	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0	0	1	0 0	0	0	0 0 0 0	0 0 1 1	0 0 0 0	a
	SUB.W @ERs+,@aa:32	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0	0	1	0 0	1	0 0 0 0	1 0 0 0	0 0 1 1	0 0 0 0	a a
	SUB.W @+ERs,@ERd	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0	0	0	0 0	0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @+ERs,@ERd+	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0	1	0	0 0	0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @+ERs,@ERd-	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0	1	0	1 0	0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @+ERs,@+ERd	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0	1	0	0 1	0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @+ERs,@-ERd	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0	1	0	1 1	0	0	rd	0 0 1 1	0 0 0 0	
SUB.W @+ERs,@(d:2,ERd)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0	0	0	d	d	0	rd	0 0 1 1	0 0 0 0		
SUB.W @+ERs,@(d:16,ERd)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0	1	1	0 0	0	0	rd	0 0 1 1	0 0 0 0	d	



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
SUB	SUB.W @+ERs,@(d:32,ERd)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @+ERs,@(d:16,Rd.B)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @+ERs,@(d:16,Rd.W)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @+ERs,@(d:16,ERd.L)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @+ERs,@(d:32,Rd.B)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @+ERs,@(d:32,Rd.W)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @+ERs,@(d:32,ERd.L)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @+ERs,@aa:16	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		0 1 0 0	0	0 0 0 0	0 0 1 1	0 0 0 0	a
	SUB.W @+ERs,@aa:32	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		0 1 0 0	1	0 0 0 0	0 0 1 1	0 0 0 0	a a
	SUB.W @-ERs,@ERd	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @-ERs,@ERd+	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @-ERs,@ERd-	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @-ERs,@+ERd	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @-ERs,@-ERd	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 1 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @-ERs,@(d:2,ERd)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		0 0 d d	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @-ERs,@(d:16,ERd)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @-ERs,@(d:32,ERd)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @-ERs,@(d:16,Rd.B)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @-ERs,@(d:16,Rd.W)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @-ERs,@(d:16,ERd.L)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @-ERs,@(d:32,Rd.B)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @-ERs,@(d:32,Rd.W)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @-ERs,@(d:32,ERd.L)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @-ERs,@aa:16	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		0 1 0 0	0	0 0 0 0	0 0 1 1	0 0 0 0	a
	SUB.W @-ERs,@aa:32	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		0 1 0 0	1	0 0 0 0	0 0 1 1	0 0 0 0	a a
	SUB.W @(d2,ERs),@ERd	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d2,ERs),@ERd+	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d2,ERs),@ERd-	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 0 1 0	0	rd	0 0 1 1	0 0 0 0	
SUB.W @(d2,ERs),@+ERd	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 1 1	0 0 0 0		
SUB.W @(d2,ERs),@-ERd	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 0 1 1	0	rd	0 0 1 1	0 0 0 0		
SUB.W @(d2,ERs),@(d:2,ERd)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		0 0 d d	0	rd	0 0 1 1	0 0 0 0		
SUB.W @(d2,ERs),@(d:16,ERd)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d	
SUB.W @(d2,ERs),@(d:32,ERd)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d	
SUB.W @(d2,ERs),@(d:16,Rd.B)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d	

Instruction	Mnemonic	Opcode						EA Ex- tension	Opcode						EA Extension																					
		15	8	7	0	15	8		7	0	15	8	7	0																						
SUB	SUB.W @(d:2,ERs),@(d:16,Rd,W)	0	1	5	0	1	d	d	6	9	0	rs	1	1	0	0	1	1	1	0	0	rd	0	0	1	1	0	0	0	0	0	d				
	SUB.W @(d:2,ERs),@(d:16,ERd,L)	0	1	5	0	1	d	d	6	9	0	rs	1	1	0	0	1	1	1	1	0	rd	0	0	1	1	0	0	0	0	0	d				
	SUB.W @(d:2,ERs),@(d:32,Rd,B)	0	1	5	0	1	d	d	6	9	0	rs	1	1	0	0	1	1	0	1	1	rd	0	0	1	1	0	0	0	0	0	d	d			
	SUB.W @(d:2,ERs),@(d:32,Rd,W)	0	1	5	0	1	d	d	6	9	0	rs	1	1	0	0	1	1	1	0	1	rd	0	0	1	1	0	0	0	0	0	d	d			
	SUB.W @(d:2,ERs),@(d:32,ERd,L)	0	1	5	0	1	d	d	6	9	0	rs	1	1	0	0	1	1	1	1	1	rd	0	0	1	1	0	0	0	0	0	d	d			
	SUB.W @(d:2,ERs),@aa:16	0	1	5	0	1	d	d	6	9	0	rs	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	a		
	SUB.W @(d:2,ERs),@aa:32	0	1	5	0	1	d	d	6	9	0	rs	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	a	a	
	SUB.W @(d:16,ERs),@ERd	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	d	0	0	0	0	0	0	rd	0	0	1	1	0	0	0	0				
	SUB.W @(d:16,ERs),@ERd+	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	0	0	0	0	0	rd	0	0	1	1	0	0	0	0				
	SUB.W @(d:16,ERs),@ERd-	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	0	1	0	0	0	rd	0	0	1	1	0	0	0	0				
	SUB.W @(d:16,ERs),@+ERd	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	0	0	1	0	rd	0	0	1	1	0	0	0	0	0				
	SUB.W @(d:16,ERs),@-ERd	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	0	1	1	0	rd	0	0	1	1	0	0	0	0	0				
	SUB.W @(d:16,ERs),@(d:2,ERd)	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	d	0	0	d	d	0	rd	0	0	1	1	0	0	0	0					
	SUB.W @(d:16,ERs),@(d:16,ERd)	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	1	0	0	0	rd	0	0	1	1	0	0	0	0	d				
	SUB.W @(d:16,ERs),@(d:32,ERd)	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	1	0	0	1	rd	0	0	1	1	0	0	0	0	d	d			
	SUB.W @(d:16,ERs),@(d:16,Rd,B)	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	1	0	1	0	rd	0	0	1	1	0	0	0	0	d				
	SUB.W @(d:16,ERs),@(d:16,Rd,W)	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	1	1	0	0	rd	0	0	1	1	0	0	0	0	d				
	SUB.W @(d:16,ERs),@(d:16,ERd,L)	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	1	1	1	0	rd	0	0	1	1	0	0	0	0	d				
	SUB.W @(d:16,ERs),@(d:32,Rd,B)	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	1	0	1	1	rd	0	0	1	1	0	0	0	0	d	d			
	SUB.W @(d:16,ERs),@(d:32,Rd,W)	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	1	1	0	1	rd	0	0	1	1	0	0	0	0	d	d			
	SUB.W @(d:16,ERs),@(d:32,ERd,L)	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	1	1	1	1	rd	0	0	1	1	0	0	0	0	d	d			
	SUB.W @(d:16,ERs),@aa:16	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	d	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	a	
	SUB.W @(d:16,ERs),@aa:32	0	1	5	0	1	0	0	6	F	0	rs	1	1	0	0	d	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	a	a
	SUB.W @(d:32,ERs),@ERd	7	8	0	rs	0	1	0	0	6	B	0	0	1	0	1	1	0	0	d	0	0	0	0	0	rd	0	0	1	1	0	0	0	0		
	SUB.W @(d:32,ERs),@ERd+	7	8	0	rs	0	1	0	0	6	B	0	0	1	0	1	1	0	0	d	1	0	0	0	0	rd	0	0	1	1	0	0	0	0		
	SUB.W @(d:32,ERs),@ERd-	7	8	0	rs	0	1	0	0	6	B	0	0	1	0	1	1	0	0	d	1	0	1	0	0	rd	0	0	1	1	0	0	0	0		
	SUB.W @(d:32,ERs),@+ERd	7	8	0	rs	0	1	0	0	6	B	0	0	1	0	1	1	0	0	d	1	0	0	1	0	rd	0	0	1	1	0	0	0	0		
	SUB.W @(d:32,ERs),@-ERd	7	8	0	rs	0	1	0	0	6	B	0	0	1	0	1	1	0	0	d	1	0	1	1	0	rd	0	0	1	1	0	0	0	0		
	SUB.W @(d:32,ERs),@(d:2,ERd)	7	8	0	rs	0	1	0	0	6	B	0	0	1	0	1	1	0	0	d	0	0	d	d	0	rd	0	0	1	1	0	0	0	0		
	SUB.W @(d:32,ERs),@(d:16,ERd)	7	8	0	rs	0	1	0	0	6	B	0	0	1	0	1	1	0	0	d	1	1	0	0	0	rd	0	0	1	1	0	0	0	0	d	
	SUB.W @(d:32,ERs),@(d:32,ERd)	7	8	0	rs	0	1	0	0	6	B	0	0	1	0	1	1	0	0	d	1	1	0	0	1	rd	0	0	1	1	0	0	0	0	d	d
	SUB.W @(d:32,ERs),@(d:16,Rd,B)	7	8	0	rs	0	1	0	0	6	B	0	0	1	0	1	1	0	0	d	1	1	0	1	0	rd	0	0	1	1	0	0	0	0	d	
SUB.W @(d:32,ERs),@(d:16,Rd,W)	7	8	0	rs	0	1	0	0	6	B	0	0	1	0	1	1	0	0	d	1	1	1	0	0	rd	0	0	1	1	0	0	0	0	d		
SUB.W @(d:32,ERs),@(d:16,ERd,L)	7	8	0	rs	0	1	0	0	6	B	0	0	1	0	1	1	0	0	d	1	1	1	1	0	rd	0	0	1	1	0	0	0	0	d		



Instruction	Mnemonic	Opcode						EA Ex- tension	Opcode						EA Extension				
		15	8	7	0	15	8		7	0	15	8	7	0					
SUB	SUB.W @(d:32,ERs),@(d:32,Rd,B)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d		
	SUB.W @(d:32,ERs),@(d:32,Rd,W)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d		
	SUB.W @(d:32,ERs),@(d:32,ERd,L)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d		
	SUB.W @(d:32,ERs),@aa:16	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 0 1 1	0 0 0 0	a		
	SUB.W @(d:32,ERs),@aa:32	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 0 1 1	0 0 0 0	a a		
	SUB.W @(d:16,Rs,B),@ERd	0	1		5	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 0 1 1	0 0 0 0		
	SUB.W @(d:16,Rs,B),@ERd+	0	1		5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 0 1 1	0 0 0 0		
	SUB.W @(d:16,Rs,B),@ERd-	0	1		5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 0 1 1	0 0 0 0		
	SUB.W @(d:16,Rs,B),@+ERd	0	1		5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 0 1 1	0 0 0 0		
	SUB.W @(d:16,Rs,B),@-ERd	0	1		5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 0 1 1	0 0 0 0		
	SUB.W @(d:16,Rs,B),@(d:2,ERd)	0	1		5	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 0 1 1	0 0 0 0		
	SUB.W @(d:16,Rs,B),@(d:16,ERd)	0	1		5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d	
	SUB.W @(d:16,Rs,B),@(d:32,ERd)	0	1		5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d	
	SUB.W @(d:16,Rs,B),@(d:16,Rd,B)	0	1		5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d	
	SUB.W @(d:16,Rs,B),@(d:16,Rd,W)	0	1		5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d	
	SUB.W @(d:16,Rs,B),@(d:16,ERd,L)	0	1		5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d	
	SUB.W @(d:16,Rs,B),@(d:32,Rd,B)	0	1		5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d	
	SUB.W @(d:16,Rs,B),@(d:32,Rd,W)	0	1		5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d	
	SUB.W @(d:16,Rs,B),@(d:32,ERd,L)	0	1		5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d	
	SUB.W @(d:16,Rs,B),@aa:16	0	1		5	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	0	0 0 0 0	0 0 1 1	0 0 0 0	a	
	SUB.W @(d:16,Rs,B),@aa:32	0	1		5	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	1	0 0 0 0	0 0 1 1	0 0 0 0	a a	
	SUB.W @(d:16,Rs,W),@ERd	0	1		5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 0 1 1	0 0 0 0		
	SUB.W @(d:16,Rs,W),@ERd+	0	1		5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 0 1 1	0 0 0 0		
	SUB.W @(d:16,Rs,W),@ERd-	0	1		5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 0 1 1	0 0 0 0		
SUB.W @(d:16,Rs,W),@+ERd	0	1		5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 0 1 1	0 0 0 0			
SUB.W @(d:16,Rs,W),@-ERd	0	1		5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 0 1 1	0 0 0 0			
SUB.W @(d:16,Rs,W),@(d:2,ERd)	0	1		5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 0 1 1	0 0 0 0			
SUB.W @(d:16,Rs,W),@(d:16,ERd)	0	1		5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d		
SUB.W @(d:16,Rs,W),@(d:32,ERd)	0	1		5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d		
SUB.W @(d:16,Rs,W),@(d:16,Rd,B)	0	1		5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d		
SUB.W @(d:16,Rs,W),@(d:16,Rd,W)	0	1		5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d		
SUB.W @(d:16,Rs,W),@(d:16,ERd,L)	0	1		5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d		
SUB.W @(d:16,Rs,W),@(d:32,Rd,B)	0	1		5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d		
SUB.W @(d:16,Rs,W),@(d:32,Rd,W)	0	1		5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d		

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
SUB	SUB.W @(d:16,Rs.W),@(d:32,ERd.L)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @(d:16,Rs.W),@aa:16	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	0	0 0 0 0	0 0 1 1	0 0 0 0	a
	SUB.W @(d:16,Rs.W),@aa:32	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	1	0 0 0 0	0 0 1 1	0 0 0 0	a a
	SUB.W @(d:16,ERs.L),@ERd	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d:16,ERs.L),@ERd+	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d:16,ERs.L),@ERd-	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d:16,ERs.L),@+ERd	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d:16,ERs.L),@-ERd	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d:16,ERs.L),@(d:2,ERd)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d:16,ERs.L),@(d:16,ERd)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @(d:16,ERs.L),@(d:32,ERd)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @(d:16,ERs.L),@(d:16,Rd.B)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @(d:16,ERs.L),@(d:16,Rd.W)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @(d:16,ERs.L),@(d:16,ERd.L)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @(d:16,ERs.L),@(d:32,Rd.B)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @(d:16,ERs.L),@(d:32,Rd.W)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @(d:16,ERs.L),@(d:32,ERd.L)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @(d:16,ERs.L),@aa:16	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	0	0 0 0 0	0 0 1 1	0 0 0 0	a
	SUB.W @(d:16,ERs.L),@aa:32	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	1	0 0 0 0	0 0 1 1	0 0 0 0	a a
	SUB.W @(d:32,Rs.B),@ERd	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d:32,Rs.B),@ERd+	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d:32,Rs.B),@ERd-	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d:32,Rs.B),@+ERd	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d:32,Rs.B),@-ERd	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d:32,Rs.B),@(d:2,ERd)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d:32,Rs.B),@(d:16,ERd)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @(d:32,Rs.B),@(d:32,ERd)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @(d:32,Rs.B),@(d:16,Rd.B)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @(d:32,Rs.B),@(d:16,Rd.W)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @(d:32,Rs.B),@(d:16,ERd.L)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @(d:32,Rs.B),@(d:32,Rd.B)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @(d:32,Rs.B),@(d:32,Rd.W)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d



Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension		
		15	8	7	0	15	8		7	0	15	8	7	0			
SUB	SUB.W @(d:32,Rs.B),@(d:32,ERd.L)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @(d:32,Rs.B),@aa:16	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 0 1 1	0 0 0 0	a
	SUB.W @(d:32,Rs.B),@aa:32	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 0 1 1	0 0 0 0	a a
	SUB.W @(d:32,Rs.W),@ERd	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d:32,Rs.W),@ERd+	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d:32,Rs.W),@ERd-	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d:32,Rs.W),@+ERd	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d:32,Rs.W),@-ERd	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d:32,Rs.W),@(d:2,ERd)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d:32,Rs.W),@(d:16,ERd)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @(d:32,Rs.W),@(d:32,ERd)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @(d:32,Rs.W),@(d:16,Rd.B)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @(d:32,Rs.W),@(d:16,Rd.W)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @(d:32,Rs.W),@(d:16,ERd.L)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.W @(d:32,Rs.W),@(d:32,Rd.B)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @(d:32,Rs.W),@(d:32,Rd.W)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @(d:32,Rs.W),@(d:32,ERd.L)	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.W @(d:32,Rs.W),@aa:16	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 0 1 1	0 0 0 0	a
	SUB.W @(d:32,Rs.W),@aa:32	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 0 1 1	0 0 0 0	a a
	SUB.W @(d:32,ERs.L),@ERd	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d:32,ERs.L),@ERd+	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d:32,ERs.L),@ERd-	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d:32,ERs.L),@+ERd	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.W @(d:32,ERs.L),@-ERd	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 0 1 1	0 0 0 0	
SUB.W @(d:32,ERs.L),@(d:2,ERd)	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 0 1 1	0 0 0 0		
SUB.W @(d:32,ERs.L),@(d:16,ERd)	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d	
SUB.W @(d:32,ERs.L),@(d:32,ERd)	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d	
SUB.W @(d:32,ERs.L),@(d:16,Rd.B)	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d	
SUB.W @(d:32,ERs.L),@(d:16,Rd.W)	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d	
SUB.W	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d	
@(d:32,ERs.L),@(d:16,ERd.L)																	
SUB.W @(d:32,ERs.L),@(d:32,Rd.B)	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d	
SUB.W @(d:32,ERs.L),@(d:32,Rd.W)	7	8	0	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d	

Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension																							
		15	8	7	0	15	8		7	0	15	8	7	0																								
SUB	SUB.W	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	1	1	rd	0	0	1	1	0	0	0	0	d	d	
	@(d:32,ERs.L),@(d:32,ERd.L)																																					
	SUB.W @(d:32,ERs.L),@aa:16	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	a	
	SUB.W @(d:32,ERs.L),@aa:32	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	a	a
	SUB.W @aa:16,@ERd									6	B	0	0	0	1	0	1	0	1	a			0	0	0	0	0	rd	0	0	1	1	0	0	0	0		
	SUB.W @aa:16,@ERd+									6	B	0	0	0	1	0	1	0	1	a			1	0	0	0	0	rd	0	0	1	1	0	0	0	0		
	SUB.W @aa:16,@ERd-									6	B	0	0	0	1	0	1	0	1	a			1	0	1	0	0	rd	0	0	1	1	0	0	0	0		
	SUB.W @aa:16,@+ERd									6	B	0	0	0	1	0	1	0	1	a			1	0	0	1	0	rd	0	0	1	1	0	0	0	0		
	SUB.W @aa:16,@-ERd									6	B	0	0	0	1	0	1	0	1	a			1	0	1	1	0	rd	0	0	1	1	0	0	0	0		
	SUB.W @aa:16,@(d:2,ERd)									6	B	0	0	0	1	0	1	0	1	a			0	0	d	d	0	rd	0	0	1	1	0	0	0	0		
	SUB.W @aa:16,@(d:16,ERd)									6	B	0	0	0	1	0	1	0	1	a			1	1	0	0	0	rd	0	0	1	1	0	0	0	0	d	
	SUB.W @aa:16,@(d:32,ERd)									6	B	0	0	0	1	0	1	0	1	a			1	1	0	0	1	rd	0	0	1	1	0	0	0	0	d	d
	SUB.W @aa:16,@(d:16,Rd,B)									6	B	0	0	0	1	0	1	0	1	a			1	1	0	1	0	rd	0	0	1	1	0	0	0	0	d	
	SUB.W @aa:16,@(d:16,Rd,W)									6	B	0	0	0	1	0	1	0	1	a			1	1	1	0	0	rd	0	0	1	1	0	0	0	0	d	
	SUB.W @aa:16,@(d:16,ERd,L)									6	B	0	0	0	1	0	1	0	1	a			1	1	1	1	0	rd	0	0	1	1	0	0	0	0	d	
	SUB.W @aa:16,@(d:32,Rd,B)									6	B	0	0	0	1	0	1	0	1	a			1	1	0	1	1	rd	0	0	1	1	0	0	0	0	d	d
	SUB.W @aa:16,@(d:32,Rd,W)									6	B	0	0	0	1	0	1	0	1	a			1	1	1	0	1	rd	0	0	1	1	0	0	0	0	d	d
	SUB.W @aa:16,@(d:32,ERd,L)									6	B	0	0	0	1	0	1	0	1	a			1	1	1	1	1	rd	0	0	1	1	0	0	0	0	d	d
	SUB.W @aa:16,@aa:16									6	B	0	0	0	1	0	1	0	1	a			0	1	0	0	0	rd	0	0	1	1	0	0	0	0	a	
	SUB.W @aa:16,@aa:32									6	B	0	0	0	1	0	1	0	1	a			0	1	0	0	1	rd	0	0	1	1	0	0	0	0	a	a
	SUB.W @aa:32,@ERd									6	B	0	0	1	1	0	1	0	1	a	a		0	0	0	0	0	rd	0	0	1	1	0	0	0	0		
	SUB.W @aa:32,@ERd+									6	B	0	0	1	1	0	1	0	1	a	a		1	0	0	0	0	rd	0	0	1	1	0	0	0	0		
	SUB.W @aa:32,@ERd-									6	B	0	0	1	1	0	1	0	1	a	a		1	0	1	0	0	rd	0	0	1	1	0	0	0	0		
	SUB.W @aa:32,@+ERd									6	B	0	0	1	1	0	1	0	1	a	a		1	0	0	1	0	rd	0	0	1	1	0	0	0	0		
	SUB.W @aa:32,@-ERd									6	B	0	0	1	1	0	1	0	1	a	a		1	0	1	1	0	rd	0	0	1	1	0	0	0	0		
	SUB.W @aa:32,@(d:2,ERd)									6	B	0	0	1	1	0	1	0	1	a	a		0	0	d	d	0	rd	0	0	1	1	0	0	0	0		
	SUB.W @aa:32,@(d:16,ERd)									6	B	0	0	1	1	0	1	0	1	a	a		1	1	0	0	0	rd	0	0	1	1	0	0	0	0	d	
	SUB.W @aa:32,@(d:32,ERd)									6	B	0	0	1	1	0	1	0	1	a	a		1	1	0	0	1	rd	0	0	1	1	0	0	0	0	d	d
	SUB.W @aa:32,@(d:16,Rd,B)									6	B	0	0	1	1	0	1	0	1	a	a		1	1	0	1	0	rd	0	0	1	1	0	0	0	0	d	
	SUB.W @aa:32,@(d:16,Rd,W)									6	B	0	0	1	1	0	1	0	1	a	a		1	1	1	0	0	rd	0	0	1	1	0	0	0	0	d	
	SUB.W @aa:32,@(d:16,ERd,L)									6	B	0	0	1	1	0	1	0	1	a	a		1	1	1	1	0	rd	0	0	1	1	0	0	0	0	d	
	SUB.W @aa:32,@(d:32,Rd,B)									6	B	0	0	1	1	0	1	0	1	a	a		1	1	0	1	1	rd	0	0	1	1	0	0	0	0	d	d
SUB.W @aa:32,@(d:32,Rd,W)									6	B	0	0	1	1	0	1	0	1	a	a		1	1	1	0	1	rd	0	0	1	1	0	0	0	0	d	d	
SUB.W @aa:32,@(d:32,ERd,L)									6	B	0	0	1	1	0	1	0	1	a	a		1	1	1	1	1	rd	0	0	1	1	0	0	0	0	d	d	



Instruction	Mnemonic	Opcode				EA Ex- tension	Opcode				EA Extension		
		15	8	7	0		15	8	7	0			
SUB	SUB.W @aa:32,@aa:16		6	B	0 0 1 1	0 1 0 1	a a	0 1 0 0	0 0 0 0	0 0 1 1	0 0 0 0	a	
	SUB.W @aa:32,@aa:32		6	B	0 0 1 1	0 1 0 1	a a	0 1 0 0	1 0 0 0	0 0 1 1	0 0 0 0	a a	
	SUB.L #xx:3,ERd							1	A	1 x x x	1 rd		
	SUB.L #xx:16,ERd							7	A	0 0 1 1	1 rd	x	
	SUB.L #xx:32,ERd							7	A	0 0 1 1	0 rd	x x	
	SUB.L #xx:16,@ERd		0	1	0	1 1 1 0		0 0 0 0	0 rd	0 0 1 1	0 0 0 0		x
	SUB.L #xx:16,@ERd+		0	1	0	1 1 1 0		1 0 0 0	0 rd	0 0 1 1	0 0 0 0		x
	SUB.L #xx:16,@ERd-		0	1	0	1 1 1 0		1 0 1 0	0 rd	0 0 1 1	0 0 0 0		x
	SUB.L #xx:16,@+ERd		0	1	0	1 1 1 0		1 0 0 1	0 rd	0 0 1 1	0 0 0 0		x
	SUB.L #xx:16,@-ERd		0	1	0	1 1 1 0		1 0 1 1	0 rd	0 0 1 1	0 0 0 0		x
	SUB.L #xx:16,@(d:2,ERd)		0	1	0	1 1 1 0		0 0 d d	0 rd	0 0 1 1	0 0 0 0		x
	SUB.L #xx:16,@(d:16,ERd)		0	1	0	1 1 1 0		1 1 0 0	0 rd	0 0 1 1	0 0 0 0	d	x
	SUB.L #xx:16,@(d:32,ERd)		0	1	0	1 1 1 0		1 1 0 0	1 rd	0 0 1 1	0 0 0 0	d d	x
	SUB.L #xx:16,@(d:16,Rd.B)		0	1	0	1 1 1 0		1 1 0 1	0 rd	0 0 1 1	0 0 0 0	d	x
	SUB.L #xx:16,@(d:16,Rd.W)		0	1	0	1 1 1 0		1 1 1 0	0 rd	0 0 1 1	0 0 0 0	d	x
	SUB.L #xx:16,@(d:16,ERd.L)		0	1	0	1 1 1 0		1 1 1 1	0 rd	0 0 1 1	0 0 0 0	d	x
	SUB.L #xx:16,@(d:32,Rd.B)		0	1	0	1 1 1 0		1 1 0 1	1 rd	0 0 1 1	0 0 0 0	d d	x
	SUB.L #xx:16,@(d:32,Rd.W)		0	1	0	1 1 1 0		1 1 1 0	1 rd	0 0 1 1	0 0 0 0	d d	x
	SUB.L #xx:16,@(d:32,ERd.L)		0	1	0	1 1 1 0		1 1 1 1	1 rd	0 0 1 1	0 0 0 0	d d	x
	SUB.L #xx:16,@aa:16		0	1	0	1 1 1 0		0 1 0 0	0 0 0 0	0 0 1 1	0 0 0 0	a	x
	SUB.L #xx:16,@aa:32		0	1	0	1 1 1 0		0 1 0 0	1 0 0 0	0 0 1 1	0 0 0 0	a a	x
	SUB.L #xx:32,@ERd		0	1	0	1 1 1 0		0 0 0 0	0 rd	0 0 1 1	1 0 0 0		x x
	SUB.L #xx:32,@ERd+		0	1	0	1 1 1 0		1 0 0 0	0 rd	0 0 1 1	1 0 0 0		x x
	SUB.L #xx:32,@ERd-		0	1	0	1 1 1 0		1 0 1 0	0 rd	0 0 1 1	1 0 0 0		x x
	SUB.L #xx:32,@+ERd		0	1	0	1 1 1 0		1 0 0 1	0 rd	0 0 1 1	1 0 0 0		x x
	SUB.L #xx:32,@-ERd		0	1	0	1 1 1 0		1 0 1 1	0 rd	0 0 1 1	1 0 0 0		x x
	SUB.L #xx:32,@(d:2,ERd)		0	1	0	1 1 1 0		0 0 d d	0 rd	0 0 1 1	1 0 0 0		x x
	SUB.L #xx:32,@(d:16,ERd)		0	1	0	1 1 1 0		1 1 0 0	0 rd	0 0 1 1	1 0 0 0	d	x x
	SUB.L #xx:32,@(d:32,ERd)		0	1	0	1 1 1 0		1 1 0 0	1 rd	0 0 1 1	1 0 0 0	d d	x x
	SUB.L #xx:32,@(d:16,Rd.B)		0	1	0	1 1 1 0		1 1 0 1	0 rd	0 0 1 1	1 0 0 0	d	x x
SUB.L #xx:32,@(d:16,Rd.W)		0	1	0	1 1 1 0		1 1 1 0	0 rd	0 0 1 1	1 0 0 0	d	x x	
SUB.L #xx:32,@(d:16,ERd.L)		0	1	0	1 1 1 0		1 1 1 1	0 rd	0 0 1 1	1 0 0 0	d	x x	
SUB.L #xx:32,@(d:32,Rd.B)		0	1	0	1 1 1 0		1 1 0 1	1 rd	0 0 1 1	1 0 0 0	d d	x x	
SUB.L #xx:32,@(d:32,Rd.W)		0	1	0	1 1 1 0		1 1 1 0	1 rd	0 0 1 1	1 0 0 0	d d	x x	

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension				
		15	8	7	0		15	8	7	0					
SUB	SUB.L #xx:32, @(d:32,ERd,L)		0	1	0	1 1 1 0		1	1 1 1 1	rd	0 0 1 1	1 0 0 0	d d	x x	
	SUB.L #xx:32, @aa:16		0	1	0	1 1 1 0		0	1 0 0 0	0 0 0 0	0 0 1 1	1 0 0 0	a	x x	
	SUB.L #xx:32, @aa:32		0	1	0	1 1 1 0		0	1 0 0 0	1 0 0 0	0 0 1 1	1 0 0 0	a a	x x	
	SUB.L ERs,ERd							1		A		rs	0	rd	
	SUB.L ERs, @ERd		0	1	0	1 0 0 1		0	0 0 0 0	0	rd	0 0 1 1	0	rs	
	SUB.L ERs, @ERd+		0	1	0	1 0 0 1		1	0 0 0 0	0	rd	0 0 1 1	0	rs	
	SUB.L ERs, @ERd-		0	1	0	1 0 0 1		1	0 1 0 0	0	rd	0 0 1 1	0	rs	
	SUB.L ERs, @+ERd		0	1	0	1 0 0 1		1	0 0 0 1	0	rd	0 0 1 1	0	rs	
	SUB.L ERs, @-ERd		0	1	0	1 0 0 1		1	0 1 1 0	0	rd	0 0 1 1	0	rs	
	SUB.L ERs, @(d:2,ERd)		0	1	0	1 0 0 1		0	0 d d	0	rd	0 0 1 1	0	rs	
	SUB.L ERs, @(d:16,ERd)		0	1	0	1 0 0 1		1	1 0 0 0	0	rd	0 0 1 1	0	rs	d
	SUB.L ERs, @(d:32,ERd)		0	1	0	1 0 0 1		1	1 0 0 1	1	rd	0 0 1 1	0	rs	d d
	SUB.L ERs, @(d:16,Rd,B)		0	1	0	1 0 0 1		1	1 0 1 0	0	rd	0 0 1 1	0	rs	d
	SUB.L ERs, @(d:16,Rd,W)		0	1	0	1 0 0 1		1	1 1 0 0	0	rd	0 0 1 1	0	rs	d
	SUB.L ERs, @(d:16,ERd,L)		0	1	0	1 0 0 1		1	1 1 1 0	0	rd	0 0 1 1	0	rs	d
	SUB.L ERs, @(d:32,Rd,B)		0	1	0	1 0 0 1		1	1 0 1 1	1	rd	0 0 1 1	0	rs	d d
	SUB.L ERs, @(d:32,Rd,W)		0	1	0	1 0 0 1		1	1 1 0 1	1	rd	0 0 1 1	0	rs	d d
	SUB.L ERs, @(d:32,ERd,L)		0	1	0	1 0 0 1		1	1 1 1 1	1	rd	0 0 1 1	0	rs	d d
	SUB.L ERs, @aa:16		0	1	0	1 0 0 1		0	1 0 0 0	0 0 0 0	0 0 1 1	0	rs	a	
	SUB.L ERs, @aa:32		0	1	0	1 0 0 1		0	1 0 0 0	1 0 0 0	0 0 1 1	0	rs	a a	
	SUB.L @ERs, ERd		0	1	0	1 0 1 0		0	0 0 0 0	0	rs	0 0 1 1	0	rd	
	SUB.L @ERs+, ERd		0	1	0	1 0 1 0		1	0 0 0 0	0	rs	0 0 1 1	0	rd	
	SUB.L @ERs-, ERd		0	1	0	1 0 1 0		1	0 1 0 0	0	rs	0 0 1 1	0	rd	
	SUB.L @+ERs, ERd		0	1	0	1 0 1 0		1	0 0 1 0	0	rs	0 0 1 1	0	rd	
	SUB.L @-ERs, ERd		0	1	0	1 0 1 0		1	0 1 1 0	0	rs	0 0 1 1	0	rd	
	SUB.L @(d:2,ERs), ERd		0	1	0	1 0 1 0		0	0 d d	0	rs	0 0 1 1	0	rd	
	SUB.L @(d:16,ERs), ERd		0	1	0	1 0 1 0		1	1 0 0 0	0	rs	0 0 1 1	0	rd	d
	SUB.L @(d:32,ERs), ERd		0	1	0	1 0 1 0		1	1 0 0 1	1	rs	0 0 1 1	0	rd	d d
	SUB.L @(d:16,Rs,B), ERd		0	1	0	1 0 1 0		1	1 0 1 0	0	rs	0 0 1 1	0	rd	d
	SUB.L @(d:16,Rs,W), ERd		0	1	0	1 0 1 0		1	1 1 0 0	0	rs	0 0 1 1	0	rd	d
	SUB.L @(d:16,ERs,L), ERd		0	1	0	1 0 1 0		1	1 1 1 0	0	rs	0 0 1 1	0	rd	d
	SUB.L @(d:32,Rs,B), ERd		0	1	0	1 0 1 0		1	1 0 1 1	1	rs	0 0 1 1	0	rd	d d
SUB.L @(d:32,Rs,W), ERd		0	1	0	1 0 1 0		1	1 1 1 0	1	rs	0 0 1 1	0	rd	d d	
SUB.L @(d:32,ERs,L), ERd		0	1	0	1 0 1 0		1	1 1 1 1	1	rs	0 0 1 1	0	rd	d d	



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension																								
		15	8	7	0	15	8	7	0		15	8	7	0																									
SUB	SUB.L @aa:16,ERd				0		1		0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	rd	a										
	SUB.L @aa:32,ERd				0		1		0	1	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	1	1	0	rd	a a									
	SUB.L @ERs,@ERd	0		1		0		0	0	1	0	0	0	6	9	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	0
	SUB.L @ERs,@ERd+	0		1		0		0	0	1	0	0	0	6	9	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs,@ERd-	0		1		0		0	0	1	0	0	0	6	9	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs,@+ERd	0		1		0		0	0	1	0	0	0	6	9	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs,@-ERd	0		1		0		0	0	1	0	0	0	6	9	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs,@(d:2,ERd)	0		1		0		0	0	1	0	0	0	6	9	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs,@(d:16,ERd)	0		1		0		0	0	1	0	0	0	6	9	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs,@(d:32,ERd)	0		1		0		0	0	1	0	0	0	6	9	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs,@(d:16,Rd,B)	0		1		0		0	0	1	0	0	0	6	9	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs,@(d:16,Rd,W)	0		1		0		0	0	1	0	0	0	6	9	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs,@(d:16,ERd,L)	0		1		0		0	0	1	0	0	0	6	9	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs,@(d:32,Rd,B)	0		1		0		0	0	1	0	0	0	6	9	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs,@(d:32,Rd,W)	0		1		0		0	0	1	0	0	0	6	9	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs,@(d:32,ERd,L)	0		1		0		0	0	1	0	0	0	6	9	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs,@aa:16	0		1		0		0	0	1	0	0	0	6	9	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs,@aa:32	0		1		0		0	0	1	0	0	0	6	9	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs+,@ERd	0		1		0		0	0	1	0	0	0	6	D	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs+,@ERd+	0		1		0		0	0	1	0	0	0	6	D	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs+,@ERd-	0		1		0		0	0	1	0	0	0	6	D	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs+,@+ERd	0		1		0		0	0	1	0	0	0	6	D	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs+,@-ERd	0		1		0		0	0	1	0	0	0	6	D	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs+,@(d:2,ERd)	0		1		0		0	0	1	0	0	0	6	D	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs+,@(d:16,ERd)	0		1		0		0	0	1	0	0	0	6	D	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs+,@(d:32,ERd)	0		1		0		0	0	1	0	0	0	6	D	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs+,@(d:16,Rd,B)	0		1		0		0	0	1	0	0	0	6	D	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
	SUB.L @ERs+,@(d:16,Rd,W)	0		1		0		0	0	1	0	0	0	6	D	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0	
SUB.L @ERs+,@(d:16,ERd,L)	0		1		0		0	0	1	0	0	0	6	D	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0		
SUB.L @ERs+,@(d:32,Rd,B)	0		1		0		0	0	1	0	0	0	6	D	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0		
SUB.L @ERs+,@(d:32,Rd,W)	0		1		0		0	0	1	0	0	0	6	D	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0		
SUB.L @ERs+,@(d:32,ERd,L)	0		1		0		0	0	1	0	0	0	6	D	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0		
SUB.L @ERs+,@aa:16	0		1		0		0	0	1	0	0	0	6	D	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0		
SUB.L @ERs+,@aa:32	0		1		0		0	0	1	0	0	0	6	D	0	0	rs	1	1	0	0	0	0	0	0	0	rd	0	0	0	1	1	0	0	0	0	0		

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
SUB	SUB.L @ERs, @ERd	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	0	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @ERs, @ERd+	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @ERs, @ERd-	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	0 1 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @ERs, @+ERd	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	0 0 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @ERs, @-ERd	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	0 1 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @ERs, @(d:2,ERd)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	0	0 0 d d	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @ERs, @(d:16,ERd)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	1 0 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @ERs, @(d:32,ERd)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	1 0 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @ERs, @(d:16,Rd.B)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	1 0 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @ERs, @(d:16,Rd.W)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @ERs, @(d:16,ERd.L)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @ERs, @(d:32,Rd.B)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	1 0 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @ERs, @(d:32,Rd.W)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @ERs, @(d:32,ERd.L)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	1	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @ERs, @aa:16	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	0	1 0 0 0	0 0 0 0	0 0 1 1	0 0 0 0	a	
	SUB.L @ERs, @aa:32	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0	0	1 0 0 0	1 0 0 0	0 0 1 1	0 0 0 0	a a	
	SUB.L @+ERs, @ERd	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	0	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @+ERs, @ERd+	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @+ERs, @ERd-	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1	0 1 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @+ERs, @+ERd	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1	0 0 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @+ERs, @-ERd	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1	0 1 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @+ERs, @(d:2,ERd)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	0	0 0 d d	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @+ERs, @(d:16,ERd)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1	1 0 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @+ERs, @(d:32,ERd)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1	1 0 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @+ERs, @(d:16,Rd.B)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1	1 0 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @+ERs, @(d:16,Rd.W)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @+ERs, @(d:16,ERd.L)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @+ERs, @(d:32,Rd.B)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1	1 0 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @+ERs, @(d:32,Rd.W)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @+ERs, @(d:32,ERd.L)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	1	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @+ERs, @aa:16	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	0	1 0 0 0	0 0 0 0	0 0 1 1	0 0 0 0	a	
	SUB.L @+ERs, @aa:32	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0	0	1 0 0 0	1 0 0 0	0 0 1 1	0 0 0 0	a a	
SUB.L @-ERs, @ERd	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0	0	0 0 0 0	0	rd	0 0 1 1	0 0 0 0		
SUB.L @-ERs, @ERd+	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0	1	0 0 0 0	0	rd	0 0 1 1	0 0 0 0		



Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
SUB	SUB.L @-ERs, @ERd	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @-ERs, @+ERd	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @-ERs, @-ERd	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @-ERs, @(d2,ERd)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		0 0 0 d	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @-ERs, @(d:16,ERd)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @-ERs, @(d:32,ERd)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @-ERs, @(d:16,Rd,B)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @-ERs, @(d:16,Rd,W)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @-ERs, @(d:16,ERd,L)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @-ERs, @(d:32,Rd,B)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @-ERs, @(d:32,Rd,W)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @-ERs, @(d:32,ERd,L)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @-ERs, @aa:16	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		0 1 0 0	0 0 0 0	0 0 1 1	0 0 0 0	a	
	SUB.L @-ERs, @aa:32	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		0 1 0 0	1 0 0 0	0 0 1 1	0 0 0 0	a a	
	SUB.L @(d2,ERs), @ERd	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d2,ERs), @ERd+	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d2,ERs), @ERd-	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 0 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d2,ERs), @+ERd	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d2,ERs), @-ERd	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 0 1 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d2,ERs), @(d2,ERd)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		0 0 d d	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d2,ERs), @(d:16,ERd)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d2,ERs), @(d:32,ERd)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d2,ERs), @(d:16,Rd,B)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d2,ERs), @(d:16,Rd,W)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d2,ERs), @(d:16,ERd,L)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d2,ERs), @(d:32,Rd,B)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d2,ERs), @(d:32,Rd,W)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d2,ERs), @(d:32,ERd,L)	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d2,ERs), @aa:16	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		0 1 0 0	0 0 0 0	0 0 1 1	0 0 0 0	a	
	SUB.L @(d2,ERs), @aa:32	0	1	0	0 1 d d	6	9	0	rs	1 1 0 0		0 1 0 0	1 0 0 0	0 0 1 1	0 0 0 0	a a	
SUB.L @(d:16,ERs), @ERd	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 0 1 1	0 0 0 0		
SUB.L @(d:16,ERs), @ERd+	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 0 1 1	0 0 0 0		
SUB.L @(d:16,ERs), @ERd-	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 0 1 1	0 0 0 0		
SUB.L @(d:16,ERs), @+ERd	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 0 1 1	0 0 0 0		

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension			
		15	8	7	0	15	8	7	0		15	8	7	0				
SUB	SUB.L @(d:16,ERs),@ERd	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	0 0 1 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:16,ERs),@(d:2,ERd)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	0	0 0 d d	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:16,ERs),@(d:16,ERd)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 0 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:16,ERs),@(d:32,ERd)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 0 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:16,ERs),@(d:16,Rd.B)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 0 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:16,ERs),@(d:16,Rd.W)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:16,ERs),@(d:16,ERd.L)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:16,ERs),@(d:32,Rd.B)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 0 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:16,ERs),@(d:32,Rd.W)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:16,ERs),@(d:32,ERd.L)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:16,ERs),@aa:16	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	0	1 0 0 0	0 0 0 0	0 0 1 1	0 0 0 0	a	
	SUB.L @(d:16,ERs),@aa:32	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	0	1 0 0 0	1 0 0 0	0 0 1 1	0 0 0 0	a a	
	SUB.L @(d:32,ERs),@ERd	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:32,ERs),@ERd+	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:32,ERs),@ERd-	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	0 1 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:32,ERs),@+ERd	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	0 0 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:32,ERs),@ERd	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	0 1 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:32,ERs),@(d:2,ERd)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0	0 d d d	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:32,ERs),@(d:16,ERd)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	1 0 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:32,ERs),@(d:32,ERd)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	1 0 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:32,ERs),@(d:16,Rd.B)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	1 0 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:32,ERs),@(d:16,Rd.W)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:32,ERs),@(d:16,ERd.L)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:32,ERs),@(d:32,Rd.B)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	1 0 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:32,ERs),@(d:32,Rd.W)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:32,ERs),@(d:32,ERd.L)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:32,ERs),@aa:16	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0	1 0 0 0	0 0 0 0	0 0 1 1	0 0 0 0	a	
	SUB.L @(d:32,ERs),@aa:32	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0	1 0 0 0	1 0 0 0	0 0 1 1	0 0 0 0	a a	
	SUB.L @(d:16,Rs.B),@ERd	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	0	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:16,Rs.B),@ERd+	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
SUB.L @(d:16,Rs.B),@ERd-	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1	0 1 0 0	0	rd	0 0 1 1	0 0 0 0		
SUB.L @(d:16,Rs.B),@+ERd	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1	0 0 1 0	0	rd	0 0 1 1	0 0 0 0		
SUB.L @(d:16,Rs.B),@ERd	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1	0 1 1 0	0	rd	0 0 1 1	0 0 0 0		
SUB.L @(d:16,Rs.B),@(d:2,ERd)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	0	0 d d d	0	rd	0 0 1 1	0 0 0 0		



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
SUB	SUB.L @(d:16,Rs.B),@(d:16,ERd)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:16,Rs.B),@(d:32,ERd)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:16,Rs.B),@(d:16,Rd.B)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:16,Rs.B),@(d:16,Rd.W)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:16,Rs.B),@(d:16,ERd.L)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:16,Rs.B),@(d:32,Rd.B)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:16,Rs.B),@(d:32,Rd.W)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:16,Rs.B),@(d:32,ERd.L)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:16,Rs.B),@aa:16	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 0 1 1	0 0 0 0	a	
	SUB.L @(d:16,Rs.B),@aa:32	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 0 1 1	0 0 0 0	a a	
	SUB.L @(d:16,Rs.W),@ERd	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:16,Rs.W),@ERd+	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:16,Rs.W),@ERd-	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:16,Rs.W),@+ERd	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:16,Rs.W),@-ERd	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:16,Rs.W),@(d:2,ERd)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:16,Rs.W),@(d:16,ERd)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:16,Rs.W),@(d:32,ERd)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:16,Rs.W),@(d:16,Rd.B)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:16,Rs.W),@(d:16,Rd.W)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:16,Rs.W),@(d:16,ERd.L)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:16,Rs.W),@(d:32,Rd.B)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:16,Rs.W),@(d:32,Rd.W)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:16,Rs.W),@(d:32,ERd.L)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:16,Rs.W),@aa:16	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 0 1 1	0 0 0 0	a	
	SUB.L @(d:16,Rs.W),@aa:32	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 0 1 1	0 0 0 0	a a	
	SUB.L @(d:16,ERs.L),@ERd	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:16,ERs.L),@ERd+	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
SUB.L @(d:16,ERs.L),@ERd-	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 0 1 1	0 0 0 0		
SUB.L @(d:16,ERs.L),@+ERd	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 0 1 1	0 0 0 0		
SUB.L @(d:16,ERs.L),@-ERd	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 0 1 1	0 0 0 0		
SUB.L @(d:16,ERs.L),@(d:2,ERd)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 0 1 1	0 0 0 0		
SUB.L @(d:16,ERs.L),@(d:16,ERd)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d	
SUB.L @(d:16,ERs.L),@(d:32,ERd)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d	

Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
SUB	SUB.L @(d:16,ERs.L),@(d:16,Rd,B)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:16,ERs.L),@(d:16,Rd,W)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:16,ERs.L),@(d:16,ERd,L)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:16,ERs.L),@(d:32,Rd,B)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:16,ERs.L),@(d:32,Rd,W)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:16,ERs.L),@(d:32,ERd,L)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:16,ERs.L),@aa:16	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	0	0 0 0 0	0 0 1 1	0 0 0 0	a
	SUB.L @(d:16,ERs.L),@aa:32	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	1	0 0 0 0	0 0 1 1	0 0 0 0	a a
	SUB.L @(d:32,Rs,B),@ERd	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:32,Rs,B),@ERd+	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:32,Rs,B),@ERd-	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:32,Rs,B),@+ERd	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:32,Rs,B),@-ERd	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:32,Rs,B),@(d:2,ERd)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:32,Rs,B),@(d:16,ERd)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:32,Rs,B),@(d:32,ERd)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:32,Rs,B),@(d:16,Rd,B)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:32,Rs,B),@(d:16,Rd,W)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:32,Rs,B),@(d:16,ERd,L)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:32,Rs,B),@(d:32,Rd,B)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:32,Rs,B),@(d:32,Rd,W)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:32,Rs,B),@(d:32,ERd,L)	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:32,Rs,B),@aa:16	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 0 1 1	0 0 0 0	a
	SUB.L @(d:32,Rs,B),@aa:32	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 0 1 1	0 0 0 0	a a
	SUB.L @(d:32,Rs,W),@ERd	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:32,Rs,W),@ERd+	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:32,Rs,W),@ERd-	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:32,Rs,W),@+ERd	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:32,Rs,W),@-ERd	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:32,Rs,W),@(d:2,ERd)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 0 1 1	0 0 0 0	
SUB.L @(d:32,Rs,W),@(d:16,ERd)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d	
SUB.L @(d:32,Rs,W),@(d:32,ERd)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d	
SUB.L @(d:32,Rs,W),@(d:16,Rd,B)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d	
SUB.L @(d:32,Rs,W),@(d:16,Rd,W)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d	



Instruction	Mnemonic	Opcode						EA Ex- tension	Opcode						EA Extension		
		15	8	7	0	15	8		7	0	15	8	7	0			
SUB	SUB.L @(d:32,Rs,W),@(d:16,ERd,L)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:32,Rs,W),@(d:32,Rd,B)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:32,Rs,W),@(d:32,Rd,W)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:32,Rs,W),@(d:32,ERd,L)	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:32,Rs,W),@aa:16	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 0 1 1	0 0 0 0	a
	SUB.L @(d:32,Rs,W),@aa:32	7	8	1	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 0 1 1	0 0 0 0	a a
	SUB.L @(d:32,ERs,L),@ERd	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:32,ERs,L),@ERd+	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:32,ERs,L),@ERd-	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:32,ERs,L),@+ERd	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:32,ERs,L),@-ERd	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:32,ERs,L),@(d:2,ERd)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @(d:32,ERs,L),@(d:16,ERd)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:32,ERs,L),@(d:32,ERd)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:32,ERs,L),@(d:16,Rd,B)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:32,ERs,L),@(d:16,Rd,W)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:32,ERs,L),@(d:16,ERd,L)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @(d:32,ERs,L),@(d:32,Rd,B)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:32,ERs,L),@(d:32,Rd,W)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:32,ERs,L),@(d:32,ERd,L)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d
	SUB.L @(d:32,ERs,L),@aa:16	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 0 1 1	0 0 0 0	a
	SUB.L @(d:32,ERs,L),@aa:32	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 0 1 1	0 0 0 0	a a
	SUB.L @aa:16,@ERd	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @aa:16,@ERd+	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 0 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @aa:16,@ERd-	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 1 0	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @aa:16,@+ERd	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 0 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @aa:16,@-ERd	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 1 1	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @aa:16,@(d:2,ERd)	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 0 d d	0	rd	0 0 1 1	0 0 0 0	
	SUB.L @aa:16,@(d:16,ERd)	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d
	SUB.L @aa:16,@(d:32,ERd)	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d
SUB.L @aa:16,@(d:16,Rd,B)	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d	
SUB.L @aa:16,@(d:16,Rd,W)	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d	
SUB.L @aa:16,@(d:16,ERd,L)	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d	
SUB.L @aa:16,@(d:32,Rd,B)	0	1	0	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d	

Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension				
		15	8	7	0	15	8	7	0		15	8	7	0					
SUB	SUB.L @aa:16,@(d:32,Rd,W)	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d			
	SUB.L @aa:16,@(d:32,ERd,L)	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d			
	SUB.L @aa:16,@aa:16	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 1 0 0	0 0 0 0	0 0 1 1	0 0 0 0	a				
	SUB.L @aa:16,@aa:32	0	1	0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 1 0 0	1 0 0 0	0 0 1 1	0 0 0 0	a a				
	SUB.L @aa:32,@+ERd	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 0 0 0	0	rd	0 0 1 1	0 0 0 0				
	SUB.L @aa:32,@ERd+	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 0 0	0	rd	0 0 1 1	0 0 0 0				
	SUB.L @aa:32,@ERd-	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 1 0	0	rd	0 0 1 1	0 0 0 0				
	SUB.L @aa:32,@+ERd	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 0 1	0	rd	0 0 1 1	0 0 0 0				
	SUB.L @aa:32,@-ERd	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 1 1	0	rd	0 0 1 1	0 0 0 0				
	SUB.L @aa:32,@(d:2,ERd)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 0 d d	0	rd	0 0 1 1	0 0 0 0				
	SUB.L @aa:32,@(d:16,ERd)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 0	0	rd	0 0 1 1	0 0 0 0	d			
	SUB.L @aa:32,@(d:32,ERd)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 0	1	rd	0 0 1 1	0 0 0 0	d d			
	SUB.L @aa:32,@(d:16,Rd,B)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 1	0	rd	0 0 1 1	0 0 0 0	d			
	SUB.L @aa:32,@(d:16,Rd,W)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 0	0	rd	0 0 1 1	0 0 0 0	d			
	SUB.L @aa:32,@(d:16,ERd,L)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 1	0	rd	0 0 1 1	0 0 0 0	d			
	SUB.L @aa:32,@(d:32,Rd,B)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 1	1	rd	0 0 1 1	0 0 0 0	d d			
	SUB.L @aa:32,@(d:32,Rd,W)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 0	1	rd	0 0 1 1	0 0 0 0	d d			
	SUB.L @aa:32,@(d:32,ERd,L)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 1	1	rd	0 0 1 1	0 0 0 0	d d			
SUB.L @aa:32,@aa:16	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 1 0 0	0 0 0 0	0 0 1 1	0 0 0 0	a					
SUB.L @aa:32,@aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 1 0 0	1 0 0 0	0 0 1 1	0 0 0 0	a a					
SUBS	SUBS #1,ERd											1	B	0 0 0 0	0	rd			
	SUBS #2,ERd											1	B	1 0 0 0	0	rd			
	SUBS #4,ERd											1	B	1 0 0 1	0	rd			
SUBX	SUBX.B #xx:8,Rd											B	rd	x x x x	x x x x				
	SUBX.B #xx:8,@ERd					7	D	0	rd	0 0 0 0	B	0 0 0 0	x x x x	x x x x					
	SUBX.B #xx:8,@ERd-	0	1	7	0 1 1 0	6	C	0	rd	1 0 0 0	B	0 0 0 0	x x x x	x x x x					
	SUBX.B Rs,Rd											1	E	rs	rd				
	SUBX.B Rs,@ERd											7	D	0	rd	0 0 0 0			
	SUBX.B Rs,@ERd-	0	1	7	0 1 1 0	6	C	0	rd	1 0 0 0	1	E	rs	0 0 0 0					
	SUBX.B @ERs,Rd											7	C	0	rs	0 0 0 0			
	SUBX.B @ERs-,Rd	0	1	7	0 1 1 0	6	C	0	rs	0 0 0 0	1	E	0 0 0 0	rd					
	SUBX.B @ERs,@ERd	0	1	7	0 1 0 0	6	8	0	rs	1 1 0 1	0 0 0 0	0	rd	0 0 1 1	0 0 0 0				
	SUBX.B @ERs-,@ERd-	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 1	1 0 1 0	0	rd	0 0 1 1	0 0 0 0				
SUBX.W #xx:16,Rd											0	1	5	0 0 0 1	7	9	0 0 1 1	rd	x



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension																	
		15	8	7	0	15	8	7	0		15	8	7	0																		
SUBX	SUBX.W #xx:16,@ERd					7	D	1	rd	0	0	0	1		7	9	0	0	1	1	0	0	0	0	x							
	SUBX.W #xx:16,@ERd-	0	1	5	0	1	1	0		6	D	0	rd	1	0	0	1		7	9	0	0	1	1	0	0	0	0	x			
	SUBX.W Rs,Rd					0	1	5	0	0	0	0	1		1	9	rs	rd														
	SUBX.W Rs,@ERd					7	D	1	rd	0	0	0	1		1	9	rs	rd	0	0	0	0										
	SUBX.W Rs,@ERd-	0	1	5	0	1	1	0		6	D	0	rd	1	0	0	1		1	9	rs	rd	0	0	0	0						
	SUBX.W @ERs,Rd					7	C	1	rs	0	0	0	1		1	9	0	0	0	0	rd											
	SUBX.W @ERs-,Rd	0	1	5	0	1	1	0		6	D	0	rs	0	0	0	1		1	9	0	0	0	0	rd							
	SUBX.W @ERs,@ERd	0	1	5	0	1	1	0		6	9	0	rs	1	1	0	1		0	0	0	0	0	rd	0	0	1	1	0	0	0	0
	SUBX.W @ERs-,@ERd-	0	1	5	0	1	1	0		6	D	0	rs	1	1	0	1		1	0	1	0	0	rd	0	0	1	1	0	0	0	0
	SUBXL #xx:32,ERd					0	1	0	0	0	0	0	1		7	A	0	0	1	1	0	rd	x	x								
	SUBXL #xx:32,@ERd	0	1	0	0	1	0	0		6	9	0	rd	1	0	0	1		7	A	0	0	1	1	0	0	0	0	x	x		
	SUBXL #xx:32,@ERd-	0	1	0	0	1	1	0		6	D	0	rd	1	0	0	1		7	A	0	0	1	1	0	0	0	0	x	x		
	SUBXL ERs,ERd					0	1	0	0	0	0	0	1		1	A	rs	rd														
	SUBXL ERs,@ERd	0	1	0	0	1	0	0		6	9	0	rd	1	0	0	1		1	A	rs	rd	0	0	0	0						
	SUBXL ERs,@ERd-	0	1	0	0	1	1	0		6	D	0	rd	1	0	0	1		1	A	rs	rd	0	0	0	0						
SUBXL @ERs,ERd	0	1	0	0	1	0	0		6	9	0	rs	0	0	0	1		1	A	1	0	0	0	0	rd							
SUBXL @ERs-,ERd	0	1	0	0	1	1	0		6	D	0	rs	0	0	0	1		1	A	1	0	0	0	0	rd							
SUBXL @ERs,@ERd	0	1	0	0	1	0	0		6	9	0	rs	1	1	0	1		0	0	0	0	0	rd	0	0	1	1	0	0	0	0	
SUBXL @ERs-,@ERd-	0	1	0	0	1	1	0		6	D	0	rs	1	1	0	1		1	0	1	0	0	rd	0	0	1	1	0	0	0	0	
TAS	TAS @ERd					0	1	E	0	0	0	0	0		7	B	0	rd	1	1	1	0	0									
TRAPA	TRAPA #x:2														5	7	0	0	x	x	0	0	0	0								
XOR	XOR.B #xx:8,Rd														D	rd	x	x	x	x	x	x	x	x								
	XOR.B #xx:8,@ERd					7	D	0	rd	0	0	0	0		D	0	0	0	0	x	x	x	x	x	x	x						
	XOR.B #xx:8,@ERd+	0	1	7	0	1	0	0		6	C	0	rd	1	0	0	0		D	0	0	0	0	x	x	x	x	x	x			
	XOR.B #xx:8,@ERd-	0	1	7	0	1	1	0		6	C	0	rd	1	0	0	0		D	0	0	0	0	x	x	x	x	x	x			
	XOR.B #xx:8,@+ERd	0	1	7	0	1	0	1		6	C	0	rd	1	0	0	0		D	0	0	0	0	x	x	x	x	x	x			
	XOR.B #xx:8,@-ERd	0	1	7	0	1	1	1		6	C	0	rd	1	0	0	0		D	0	0	0	0	x	x	x	x	x	x			
	XOR.B #xx:8,@(d:2),ERd	0	1	7	0	1	d	d		6	8	0	rd	1	0	0	0		D	0	0	0	0	x	x	x	x	x	x			
	XOR.B #xx:8,@(d:16),ERd	0	1	7	0	1	0	0		6	E	0	rd	1	0	0	0	d		D	0	0	0	0	x	x	x	x	x	x		
	XOR.B #xx:8,@(d:32),ERd	7	8	0	rd	0	1	0	0	6	A	0	0	1	0	1	0	0	0	d	d	D	0	0	0	0	x	x	x	x	x	x
	XOR.B #xx:8,@(d:16),Rd.B	0	1	7	0	1	0	1		6	E	0	rd	1	0	0	0	d		D	0	0	0	0	x	x	x	x	x	x		
	XOR.B #xx:8,@(d:16),Rd.W	0	1	7	0	1	1	0		6	E	0	rd	1	0	0	0	d		D	0	0	0	0	x	x	x	x	x	x		
XOR.B #xx:8,@(d:16),ERd.L	0	1	7	0	1	1	1		6	E	0	rd	1	0	0	0	d		D	0	0	0	0	x	x	x	x	x	x			
XOR.B #xx:8,@(d:32),Rd.B	7	8	0	rd	0	1	0	1	6	A	0	0	1	0	1	0	0	0	d	d	D	0	0	0	0	x	x	x	x	x	x	

Instruction	Mnemonic	Opcode						EA Ex- tension	Opcode						EA Extension		
		15	8	7	0	15	8		7	0	15	8	7	0			
XOR	XOR.B #xx:8,@(d:32,Rd,W)	7	8	0	rd	0 1 1 0	6	A	0 0 1 0	1 0 0 0	d d	D	0 0 0 0	x x x x	x x x x		
	XOR.B #xx:8,@(d:32,ERd,L)	7	8	0	rd	0 1 1 1	6	A	0 0 1 0	1 0 0 0	d d	D	0 0 0 0	x x x x	x x x x		
	XOR.B #xx:8,@aa:8						7	F	a a a a	a a a a		D	0 0 0 0	x x x x	x x x x		
	XOR.B #xx:8,@aa:16						6	A	0 0 0 1	1 0 0 0	a	D	0 0 0 0	x x x x	x x x x		
	XOR.B #xx:8,@aa:32						6	A	0 0 1 1	1 0 0 0	a a	D	0 0 0 0	x x x x	x x x x		
	XOR.B Rs,Rd												1	5	rs	rd	
	XOR.B Rs,@ERd						7	D	0 rd	0 0 0 0		1	5	rs	0 0 0 0		
	XOR.B Rs,@ERd+						0	1	7	1 0 0 1		1 0 0 0	0	rd	0 1 0 1	rs	
	XOR.B Rs,@ERd-						0	1	7	1 0 0 1		1 0 1 0	0	rd	0 1 0 1	rs	
	XOR.B Rs,@+ERd						0	1	7	1 0 0 1		1 0 0 1	0	rd	0 1 0 1	rs	
	XOR.B Rs,@-ERd						0	1	7	1 0 0 1		1 0 1 1	0	rd	0 1 0 1	rs	
	XOR.B Rs,@(d:2,ERd)						0	1	7	1 0 0 1		0 0 d d	0	rd	0 1 0 1	rs	
	XOR.B Rs,@(d:16,ERd)						0	1	7	1 0 0 1		1 1 0 0	0	rd	0 1 0 1	rs	d
	XOR.B Rs,@(d:32,ERd)						0	1	7	1 0 0 1		1 1 0 0	1	rd	0 1 0 1	rs	d d
	XOR.B Rs,@(d:16,Rd,B)						0	1	7	1 0 0 1		1 1 0 1	0	rd	0 1 0 1	rs	d
	XOR.B Rs,@(d:16,Rd,W)						0	1	7	1 0 0 1		1 1 1 0	0	rd	0 1 0 1	rs	d
	XOR.B Rs,@(d:16,ERd,L)						0	1	7	1 0 0 1		1 1 1 1	0	rd	0 1 0 1	rs	d
	XOR.B Rs,@(d:32,Rd,B)						0	1	7	1 0 0 1		1 1 0 1	1	rd	0 1 0 1	rs	d d
	XOR.B Rs,@(d:32,Rd,W)						0	1	7	1 0 0 1		1 1 1 0	1	rd	0 1 0 1	rs	d d
	XOR.B Rs,@(d:32,ERd,L)						0	1	7	1 0 0 1		1 1 1 1	1	rd	0 1 0 1	rs	d d
	XOR.B Rs,@aa:8						7	F	a a a a	a a a a		1	5	rs	0 0 0 0		
	XOR.B Rs,@aa:16						6	A	0 0 0 1	1 0 0 0	a	1	5	rs	0 0 0 0		
	XOR.B Rs,@aa:32						6	A	0 0 1 1	1 0 0 0	a a	1	5	rs	0 0 0 0		
	XOR.B @ERs,Rd						7	C	0 rs	0 0 0 0		1	5	0 0 0 0	rd		
	XOR.B @ERs+,Rd						0	1	7	1 0 1 0		1 0 0 0	0	rs	0 1 0 1	rd	
	XOR.B @ERs-,Rd						0	1	7	1 0 1 0		1 0 1 0	0	rs	0 1 0 1	rd	
	XOR.B @+ERs,Rd						0	1	7	1 0 1 0		1 0 0 1	0	rs	0 1 0 1	rd	
	XOR.B @-ERs,Rd						0	1	7	1 0 1 0		1 0 1 1	0	rs	0 1 0 1	rd	
	XOR.B @(d:2,ERs),Rd						0	1	7	1 0 1 0		0 0 d d	0	rs	0 1 0 1	rd	
	XOR.B @(d:16,ERs),Rd						0	1	7	1 0 1 0		1 1 0 0	0	rs	0 1 0 1	rd	d
	XOR.B @(d:32,ERs),Rd						0	1	7	1 0 1 0		1 1 0 0	1	rs	0 1 0 1	rd	d d
	XOR.B @(d:16,Rs,B),Rd						0	1	7	1 0 1 0		1 1 0 1	0	rs	0 1 0 1	rd	d
XOR.B @(d:16,Rs,W),Rd						0	1	7	1 0 1 0		1 1 1 0	0	rs	0 1 0 1	rd	d	
XOR.B @(d:16,ERs,L),Rd						0	1	7	1 0 1 0		1 1 1 1	0	rs	0 1 0 1	rd	d	



Instruction	Mnemonic	Opcode			Opcode			EA Extension	Opcode			EA Extension						
		15	8	7	0	15	8		7	0	15		8	7	0			
XOR	XOR.B @(d:32,Rs.B),Rd				0	1	7	1 0 1 0		1	1 0 1	1	rs	0 1 0 1	rd	d d		
	XOR.B @(d:32,Rs.W),Rd				0	1	7	1 0 1 0		1	1 1 0	1	rs	0 1 0 1	rd	d d		
	XOR.B @(d:32,ERs.L),Rd				0	1	7	1 0 1 0		1	1 1 1	1	rs	0 1 0 1	rd	d d		
	XOR.B @aa:8,Rd				7	E	a a a a	a a a a a			1	5		0 0 0 0	rd			
	XOR.B @aa:16,Rd				6	A	0 0 0 1	0 0 0 0 a			1	5		0 0 0 0	rd			
	XOR.B @aa:32,Rd				6	A	0 0 1 1	0 0 0 0 a a			1	5		0 0 0 0	rd			
	XOR.B @ERs,@ERd				7	C	0	rs 0 1 0 1			0 0 0 0	0	rd	0 1 0 1	0 0 0 0			
	XOR.B @ERs,@ERd+				7	C	0	rs 0 1 0 1			1 0 0 0	0	rd	0 1 0 1	0 0 0 0			
	XOR.B @ERs,@ERd-				7	C	0	rs 0 1 0 1			1 0 1 0	0	rd	0 1 0 1	0 0 0 0			
	XOR.B @ERs,@+ERd				7	C	0	rs 0 1 0 1			1 0 0 1	0	rd	0 1 0 1	0 0 0 0			
	XOR.B @ERs,@-ERd				7	C	0	rs 0 1 0 1			1 0 1 1	0	rd	0 1 0 1	0 0 0 0			
	XOR.B @ERs,@(d:2,ERd)				7	C	0	rs 0 1 0 1			0 0 d d	0	rd	0 1 0 1	0 0 0 0			
	XOR.B @ERs,@(d:16,ERd)				7	C	0	rs 0 1 0 1			1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d		
	XOR.B @ERs,@(d:32,ERd)				7	C	0	rs 0 1 0 1			1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d d		
	XOR.B @ERs,@(d:16,Rd.B)				7	C	0	rs 0 1 0 1			1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d		
	XOR.B @ERs,@(d:16,Rd.W)				7	C	0	rs 0 1 0 1			1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d		
	XOR.B @ERs,@(d:16,ERd.L)				7	C	0	rs 0 1 0 1			1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d		
	XOR.B @ERs,@(d:32,Rd.B)				7	C	0	rs 0 1 0 1			1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d		
	XOR.B @ERs,@(d:32,Rd.W)				7	C	0	rs 0 1 0 1			1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d		
	XOR.B @ERs,@(d:32,ERd.L)				7	C	0	rs 0 1 0 1			1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d		
	XOR.B @ERs,@aa:16				7	C	0	rs 0 1 0 1			0 1 0 0	0 0 0 0	0	0 1 0 1	0 0 0 0	a a		
	XOR.B @ERs,@aa:32				7	C	0	rs 0 1 0 1			0 1 0 0	1 0 0 0	0	0 1 0 1	0 0 0 0	a a		
	XOR.B @ERs+,@ERd		0	1	7	0	1 0 0 0	6	C	0	rs 1 1 0 0			0 0 0 0	0	rd	0 1 0 1	0 0 0 0
	XOR.B @ERs+,@ERd+		0	1	7	0	1 0 0 0	6	C	0	rs 1 1 0 0			1 0 0 0	0	rd	0 1 0 1	0 0 0 0
	XOR.B @ERs+,@ERd-		0	1	7	0	1 0 0 0	6	C	0	rs 1 1 0 0			1 0 1 0	0	rd	0 1 0 1	0 0 0 0
	XOR.B @ERs+,@+ERd		0	1	7	0	1 0 0 0	6	C	0	rs 1 1 0 0			1 0 0 1	0	rd	0 1 0 1	0 0 0 0
	XOR.B @ERs+,@-ERd		0	1	7	0	1 0 0 0	6	C	0	rs 1 1 0 0			1 0 1 1	0	rd	0 1 0 1	0 0 0 0
	XOR.B @ERs+,@(d:2,ERd)		0	1	7	0	1 0 0 0	6	C	0	rs 1 1 0 0			0 0 d d	0	rd	0 1 0 1	0 0 0 0
	XOR.B @ERs+,@(d:16,ERd)		0	1	7	0	1 0 0 0	6	C	0	rs 1 1 0 0			1 1 0 0	0	rd	0 1 0 1	0 0 0 0
	XOR.B @ERs+,@(d:32,ERd)		0	1	7	0	1 0 0 0	6	C	0	rs 1 1 0 0			1 1 0 0	1	rd	0 1 0 1	0 0 0 0
XOR.B @ERs+,@(d:16,Rd.B)		0	1	7	0	1 0 0 0	6	C	0	rs 1 1 0 0			1 1 0 1	0	rd	0 1 0 1	0 0 0 0	
XOR.B @ERs+,@(d:16,Rd.W)		0	1	7	0	1 0 0 0	6	C	0	rs 1 1 0 0			1 1 1 0	0	rd	0 1 0 1	0 0 0 0	
XOR.B @ERs+,@(d:16,ERd.L)		0	1	7	0	1 0 0 0	6	C	0	rs 1 1 0 0			1 1 1 1	0	rd	0 1 0 1	0 0 0 0	
XOR.B @ERs+,@(d:32,Rd.B)		0	1	7	0	1 0 0 0	6	C	0	rs 1 1 0 0			1 1 0 1	1	rd	0 1 0 1	0 0 0 0	

Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension		
		15	8	7	0	15	8		7	0	15	8	7	0			
XOR	XOR.B @ERs+, @(d:32,Rd,W)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0	1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d	
	XOR.B @ERs+, @(d:32,ERd,L)	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0	1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d	
	XOR.B @ERs+, @aa:16	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0	0 1 0 0	0	0 0 0 0	0 1 0 1	0 0 0 0	a	
	XOR.B @ERs+, @aa:32	0	1	7	0 1 0 0	6	C	0	rs	1 1 0 0	0 1 0 0	1	0 0 0 0	0 1 0 1	0 0 0 0	a a	
	XOR.B @ERs-, @ERd	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	0 0 0 0	0	rd	0 1 0 1	0 0 0 0		
	XOR.B @ERs-, @ERd+	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 0 0 0	0	rd	0 1 0 1	0 0 0 0		
	XOR.B @ERs-, @ERd-	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 0 1 0	0	rd	0 1 0 1	0 0 0 0		
	XOR.B @ERs-, @+ERd	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 0 0 1	0	rd	0 1 0 1	0 0 0 0		
	XOR.B @ERs-, @-ERd	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 0 1 1	0	rd	0 1 0 1	0 0 0 0		
	XOR.B @ERs-, @(d:2,ERd)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	0 0 d d	0	rd	0 1 0 1	0 0 0 0		
	XOR.B @ERs-, @(d:16,ERd)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d	
	XOR.B @ERs-, @(d:32,ERd)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d d	
	XOR.B @ERs-, @(d:16,Rd,B)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d	
	XOR.B @ERs-, @(d:16,Rd,W)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d	
	XOR.B @ERs-, @(d:16,ERd,L)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d	
	XOR.B @ERs-, @(d:32,Rd,B)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d	
	XOR.B @ERs-, @(d:32,Rd,W)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d	
	XOR.B @ERs-, @(d:32,ERd,L)	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d	
	XOR.B @ERs-, @aa:16	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	0 1 0 0	0	0 0 0 0	0 1 0 1	0 0 0 0	a	
	XOR.B @ERs-, @aa:32	0	1	7	0 1 1 0	6	C	0	rs	1 1 0 0	0 1 0 0	1	0 0 0 0	0 1 0 1	0 0 0 0	a a	
	XOR.B @+ERs, @ERd	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	0 0 0 0	0	rd	0 1 0 1	0 0 0 0		
	XOR.B @+ERs, @ERd+	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 0 0 0	0	rd	0 1 0 1	0 0 0 0		
	XOR.B @+ERs, @ERd-	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 0 1 0	0	rd	0 1 0 1	0 0 0 0		
	XOR.B @+ERs, @+ERd	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 0 0 1	0	rd	0 1 0 1	0 0 0 0		
	XOR.B @+ERs, @-ERd	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 0 1 1	0	rd	0 1 0 1	0 0 0 0		
	XOR.B @+ERs, @(d:2,ERd)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	0 0 d d	0	rd	0 1 0 1	0 0 0 0		
	XOR.B @+ERs, @(d:16,ERd)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d	
	XOR.B @+ERs, @(d:32,ERd)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d d	
	XOR.B @+ERs, @(d:16,Rd,B)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d	
	XOR.B @+ERs, @(d:16,Rd,W)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d	
	XOR.B @+ERs, @(d:16,ERd,L)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d	
	XOR.B @+ERs, @(d:32,Rd,B)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d	
XOR.B @+ERs, @(d:32,Rd,W)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d		
XOR.B @+ERs, @(d:32,ERd,L)	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d		



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
XOR	XOR.B @+ERs,@aa:16	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	0	0 1 0 0	0 0 0 0	0 1 0 1	0 0 0 0	a
	XOR.B @+ERs,@aa:32	0	1	7	0 1 0 1	6	C	0	rs	1 1 0 0	0	0 1 0 0	1 0 0 0	0 1 0 1	0 0 0 0	a a
	XOR.B @-ERs,@ERd	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	0	0 0 0 0	0 rd	0 1 0 1	0 0 0 0	
	XOR.B @-ERs,@ERd+	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	0	1 0 0 0	0 rd	0 1 0 1	0 0 0 0	
	XOR.B @-ERs,@ERd-	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	1	0 1 0 0	0 rd	0 1 0 1	0 0 0 0	
	XOR.B @-ERs,@+ERd	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	1	0 0 1 0	0 rd	0 1 0 1	0 0 0 0	
	XOR.B @-ERs,@-ERd	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	1	0 1 1 0	0 rd	0 1 0 1	0 0 0 0	
	XOR.B @-ERs,@(d:2,ERd)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	0	0 0 d d	0 rd	0 1 0 1	0 0 0 0	
	XOR.B @-ERs,@(d:16,ERd)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	1	1 0 0 0	0 rd	0 1 0 1	0 0 0 0	d
	XOR.B @-ERs,@(d:32,ERd)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	1	1 0 0 1	1 rd	0 1 0 1	0 0 0 0	d d
	XOR.B @-ERs,@(d:16,Rd,B)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	1	1 0 1 0	0 rd	0 1 0 1	0 0 0 0	d
	XOR.B @-ERs,@(d:16,Rd,W)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	1	1 1 0 0	0 rd	0 1 0 1	0 0 0 0	d
	XOR.B @-ERs,@(d:16,ERd,L)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	1	1 1 1 0	0 rd	0 1 0 1	0 0 0 0	d
	XOR.B @-ERs,@(d:32,Rd,B)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	1	1 0 1 1	1 rd	0 1 0 1	0 0 0 0	d d
	XOR.B @-ERs,@(d:32,Rd,W)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	1	1 1 0 1	1 rd	0 1 0 1	0 0 0 0	d d
	XOR.B @-ERs,@(d:32,ERd,L)	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	1	1 1 1 1	1 rd	0 1 0 1	0 0 0 0	d d
	XOR.B @-ERs,@aa:16	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	0	1 0 0 0	0 0 0 0	0 1 0 1	0 0 0 0	a
	XOR.B @-ERs,@aa:32	0	1	7	0 1 1 1	6	C	0	rs	1 1 0 0	0	1 0 0 0	1 0 0 0	0 1 0 1	0 0 0 0	a a
	XOR.B @(d:2,ERs),@ERd	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	0	0 0 0 0	0 rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:2,ERs),@ERd+	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	1	0 0 0 0	0 rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:2,ERs),@ERd-	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	1	0 1 0 0	0 rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:2,ERs),@+ERd	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	1	0 0 1 0	0 rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:2,ERs),@-ERd	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	1	0 1 1 0	0 rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:2,ERs),@(d:2,ERd)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	0	0 d d 0	0 rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:2,ERs),@(d:16,ERd)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	1	1 0 0 0	0 rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:2,ERs),@(d:32,ERd)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	1	1 0 0 1	1 rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:2,ERs),@(d:16,Rd,B)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	1	1 0 1 0	0 rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:2,ERs),@(d:16,Rd,W)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	1	1 1 0 0	0 rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:2,ERs),@(d:16,ERd,L)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	1	1 1 1 0	0 rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:2,ERs),@(d:32,Rd,B)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	1	1 0 1 1	1 rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:2,ERs),@(d:32,Rd,W)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	1	1 1 0 1	1 rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:2,ERs),@(d:32,ERd,L)	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	1	1 1 1 1	1 rd	0 1 0 1	0 0 0 0	d d
XOR.B @(d:2,ERs),@aa:16	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	0	1 0 0 0	0 0 0 0	0 1 0 1	0 0 0 0	a	
XOR.B @(d:2,ERs),@aa:32	0	1	7	0 1 d d	6	8	0	rs	1 1 0 0	0	1 0 0 0	1 0 0 0	0 1 0 1	0 0 0 0	a a	

Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
XOR	XOR.B @(d:16,ERs),@ERd	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:16,ERs),@ERd+	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:16,ERs),@ERd-	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:16,ERs),@+ERd	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:16,ERs),@-ERd	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:16,ERs),@(d:2,ERd)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:16,ERs),@(d:16,ERd)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:16,ERs),@(d:32,ERd)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:16,ERs),@(d:16,Rd,B)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:16,ERs),@(d:16,Rd,W)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:16,ERs),@(d:16,ERd,L)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:16,ERs),@(d:32,Rd,B)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:16,ERs),@(d:32,Rd,W)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:16,ERs),@(d:32,ERd,L)	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:16,ERs),@aa:16	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 1 0 1	0 0 0 0	a	
	XOR.B @(d:16,ERs),@aa:32	0	1	7	0 1 0 0	6	E	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 0 1	0 0 0 0	a a	
	XOR.B @(d:32,ERs),@ERd	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:32,ERs),@ERd+	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:32,ERs),@ERd-	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:32,ERs),@+ERd	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:32,ERs),@-ERd	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:32,ERs),@(d:2,ERd)	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:32,ERs),@(d:16,ERd)	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:32,ERs),@(d:32,ERd)	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:32,ERs),@(d:16,Rd,B)	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:32,ERs),@(d:16,Rd,W)	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:32,ERs),@(d:16,ERd,L)	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:32,ERs),@(d:32,Rd,B)	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:32,ERs),@(d:32,Rd,W)	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:32,ERs),@(d:32,ERd,L)	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:32,ERs),@aa:16	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	0 0 0 0	0 1 0 1	0 0 0 0	a	
	XOR.B @(d:32,ERs),@aa:32	7	8	0	rs	0 1 0 0	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	1 0 0 0	0 1 0 1	0 0 0 0	a a	
XOR.B @(d:16,Rs,B),@ERd	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 1 0 1	0 0 0 0		
XOR.B @(d:16,Rs,B),@ERd+	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 1 0 1	0 0 0 0		



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
XOR	XOR.B @(d:16,Rs.B),@ERd-	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:16,Rs.B),@+ERd	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:16,Rs.B),@-ERd	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:16,Rs.B),@(d:2,ERd)	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:16,Rs.B),@(d:16,ERd)	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:16,Rs.B),@(d:32,ERd)	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:16,Rs.B),@(d:16,Rd.B)	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:16,Rs.B),@(d:16,Rd.W)	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:16,Rs.B),@(d:16,ERd.L)	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:16,Rs.B),@(d:32,Rd.B)	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:16,Rs.B),@(d:32,Rd.W)	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:16,Rs.B),@(d:32,ERd.L)	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:16,Rs.B),@aa:16	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 1 0 1	0 0 0 0	a	
	XOR.B @(d:16,Rs.B),@aa:32	0	1	7	0 1 0 1	6	E	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 0 1	0 0 0 0	a a	
	XOR.B @(d:16,Rs.W),@ERd-	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:16,Rs.W),@ERd+	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:16,Rs.W),@ERd-	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:16,Rs.W),@+ERd	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:16,Rs.W),@-ERd	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:16,Rs.W),@(d:2,ERd)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:16,Rs.W),@(d:16,ERd)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:16,Rs.W),@(d:32,ERd)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:16,Rs.W),@(d:16,Rd.B)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:16,Rs.W),@(d:16,Rd.W)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d
XOR.B @(d:16,Rs.W),@(d:16,ERd.L)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d	
XOR.B @(d:16,Rs.W),@(d:32,Rd.B)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d	
XOR.B @(d:16,Rs.W),@(d:32,Rd.W)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d	
XOR.B @(d:16,Rs.W),@(d:32,ERd.L)	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d	
XOR.B @(d:16,Rs.W),@aa:16	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 1 0 1	0 0 0 0	a		
XOR.B @(d:16,Rs.W),@aa:32	0	1	7	0 1 1 0	6	E	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 0 1	0 0 0 0	a a		
XOR.B @(d:16,ERs.L),@ERd	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 1 0 1	0 0 0 0		
XOR.B @(d:16,ERs.L),@ERd+	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 1 0 1	0 0 0 0		
XOR.B @(d:16,ERs.L),@ERd-	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 0 1	0 0 0 0		
XOR.B @(d:16,ERs.L),@+ERd	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 0 1	0 0 0 0		

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension			
		15	8	7	0	15	8	7	0		15	8	7	0				
XOR	XOR.B @(d:16,ERs.L),@ERd	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1	0 1 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:16,ERs.L),@(d:2,ERd)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	0	0 d d	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:16,ERs.L),@(d:16,ERd)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1	1 0 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:16,ERs.L),@(d:32,ERd)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1	1 0 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:16,ERs.L),@(d:16,Rd.B)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1	1 0 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:16,ERs.L),@(d:16,Rd.W)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1	1 1 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:16,ERs.L),@(d:16,ERd.L)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1	1 1 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:16,ERs.L),@(d:32,Rd.B)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1	1 0 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:16,ERs.L),@(d:32,Rd.W)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1	1 1 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:16,ERs.L),@(d:32,ERd.L)	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	1	1 1 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:16,ERs.L),@aa:16	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	0	1 0 0	0	0 0 0 0	0 1 0 1	0 0 0 0	a
	XOR.B @(d:16,ERs.L),@aa:32	0	1	7	0 1 1 1	6	E	0	rs	1 1 0 0	d	0	1 0 0	1	0 0 0 0	0 1 0 1	0 0 0 0	a a
	XOR.B @(d:32,Rs.B),@ERd	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	0	0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:32,Rs.B),@ERd+	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:32,Rs.B),@ERd-	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	0 1 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:32,Rs.B),@+ERd	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	0 0 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:32,Rs.B),@-ERd	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	0 1 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:32,Rs.B),@(d:2,ERd)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	0	0 d d	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:32,Rs.B),@(d:16,ERd)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	1 0 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:32,Rs.B),@(d:32,ERd)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	1 0 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:32,Rs.B),@(d:16,Rd.B)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	1 0 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:32,Rs.B),@(d:16,Rd.W)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	1 1 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:32,Rs.B),@(d:16,ERd.L)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	1 1 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.B @(d:32,Rs.B),@(d:32,Rd.B)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	1 0 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:32,Rs.B),@(d:32,Rd.W)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	1 1 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:32,Rs.B),@(d:32,ERd.L)	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	1	1 1 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.B @(d:32,Rs.B),@aa:16	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	0	1 0 0	0	0 0 0 0	0 1 0 1	0 0 0 0	a
	XOR.B @(d:32,Rs.B),@aa:32	7	8	0	rs	0 1 0 1	6	A	0 0 1 0	1 1 0 0	d d	0	1 0 0	1	0 0 0 0	0 1 0 1	0 0 0 0	a a
	XOR.B @(d:32,Rs.W),@ERd	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	0	0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:32,Rs.W),@ERd+	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1	0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:32,Rs.W),@ERd-	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1	0 1 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @(d:32,Rs.W),@+ERd	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1	0 0 1	0	rd	0 1 0 1	0 0 0 0	
XOR.B @(d:32,Rs.W),@-ERd	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1	0 1 1	0	rd	0 1 0 1	0 0 0 0		
XOR.B @(d:32,Rs.W),@(d:2,ERd)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	0	0 d d	0	rd	0 1 0 1	0 0 0 0		



Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension			
		15	8	7	0	15	8		7	0	15	8	7	0				
XOR	XOR.B @(d:32.Rs.W),@(d:16.ERd)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d	
	XOR.B @(d:32.Rs.W),@(d:32.ERd)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d d	
	XOR.B @(d:32.Rs.W),@(d:16.Rd.B)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d	
	XOR.B @(d:32.Rs.W),@(d:16.Rd.W)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d	
	XOR.B @(d:32.Rs.W),@(d:16.ERd.L)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d	
	XOR.B @(d:32.Rs.W),@(d:32.Rd.B)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d	
	XOR.B @(d:32.Rs.W),@(d:32.Rd.W)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d	
	XOR.B @(d:32.Rs.W),@(d:32.ERd.L)	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d	
	XOR.B @(d:32.Rs.W),@aa:16	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	0 0 0 0	0 1 0 1	0 0 0 0	a		
	XOR.B @(d:32.Rs.W),@aa:32	7	8	0	rs	0 1 1 0	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	1 0 0 0	0 1 0 1	0 0 0 0	a a		
	XOR.B @(d:32.ERs.L),@ERd	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 1 0 1	0 0 0 0		
	XOR.B @(d:32.ERs.L),@ERd+	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 1 0 1	0 0 0 0		
	XOR.B @(d:32.ERs.L),@ERd-	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 1 0 1	0 0 0 0		
	XOR.B @(d:32.ERs.L),@+ERd	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 1 0 1	0 0 0 0		
	XOR.B @(d:32.ERs.L),@-ERd	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 1 0 1	0 0 0 0		
	XOR.B @(d:32.ERs.L),@(d:2.ERd)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 1 0 1	0 0 0 0		
	XOR.B @(d:32.ERs.L),@(d:16.ERd)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d	
	XOR.B @(d:32.ERs.L),@(d:32.ERd)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d d	
	XOR.B @(d:32.ERs.L),@(d:16.Rd.B)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d	
	XOR.B @(d:32.ERs.L),@(d:16.Rd.W)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d	
	XOR.B @(d:32.ERs.L),@(d:16.ERd.L)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d	
	XOR.B @(d:32.ERs.L),@(d:32.Rd.B)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d	
	XOR.B @(d:32.ERs.L),@(d:32.Rd.W)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d	
	XOR.B @(d:32.ERs.L),@(d:32.ERd.L)	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d	
	XOR.B @(d:32.ERs.L),@aa:16	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	0 0 0 0	0 1 0 1	0 0 0 0	a		
	XOR.B @(d:32.ERs.L),@aa:32	7	8	0	rs	0 1 1 1	6	A	0 0 1 0	1 1 0 0	d d	0 1 0 0	1 0 0 0	0 1 0 1	0 0 0 0	a a		
	XOR.B @aa:16,@ERd						6	A	0 0 0 1	0 1 0 1	a		0 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.B @aa:16,@ERd+						6	A	0 0 0 1	0 1 0 1	a		1 0 0 0	0	rd	0 1 0 1	0 0 0 0	
XOR.B @aa:16,@ERd-						6	A	0 0 0 1	0 1 0 1	a		1 0 1 0	0	rd	0 1 0 1	0 0 0 0		
XOR.B @aa:16,@+ERd						6	A	0 0 0 1	0 1 0 1	a		1 0 0 1	0	rd	0 1 0 1	0 0 0 0		
XOR.B @aa:16,@-ERd						6	A	0 0 0 1	0 1 0 1	a		1 0 1 1	0	rd	0 1 0 1	0 0 0 0		
XOR.B @aa:16,@(d:2.ERd)						6	A	0 0 0 1	0 1 0 1	a		0 0 d d	0	rd	0 1 0 1	0 0 0 0		
XOR.B @aa:16,@(d:16.ERd)						6	A	0 0 0 1	0 1 0 1	a		1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d	
XOR.B @aa:16,@(d:32.ERd)						6	A	0 0 0 1	0 1 0 1	a		1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d d	

Instruction	Mnemonic	Opcode				EA Ex- tension	Opcode				EA Extension				
		15	8	7	0		15	8	7	0					
XOR	XOR.B @aa:16,@(d:16,Rd,B)			6	A	0 0 0 0 1	0 1 0 1 1	a	1 1 0 1 1	0	rd	0 1 0 1 1	0 0 0 0 0	d	
	XOR.B @aa:16,@(d:16,Rd,W)			6	A	0 0 0 0 1	0 1 0 1 1	a	1 1 1 1 0	0	rd	0 1 0 1 1	0 0 0 0 0	d	
	XOR.B @aa:16,@(d:16,ERd,L)			6	A	0 0 0 0 1	0 1 0 1 1	a	1 1 1 1 1	0	rd	0 1 0 1 1	0 0 0 0 0	d	
	XOR.B @aa:16,@(d:32,Rd,B)			6	A	0 0 0 0	0 1 0 1 1	a	1 1 0 1 1	1	rd	0 1 0 1 1	0 0 0 0 0	d	d
	XOR.B @aa:16,@(d:32,Rd,W)			6	A	0 0 0 0 1	0 1 0 1 1	a	1 1 1 1 0	1	rd	0 1 0 1 1	0 0 0 0 0	d	d
	XOR.B @aa:16,@(d:32,ERd,L)			6	A	0 0 0 0 1	0 1 0 1 1	a	1 1 1 1 1	1	rd	0 1 0 1 1	0 0 0 0 0	d	d
	XOR.B @aa:16,@aa:16			6	A	0 0 0 0 1	0 1 0 1 1	a	0 1 0 0 0	0	0 0 0 0 0	0 1 0 1 1	0 0 0 0 0	a	
	XOR.B @aa:16,@aa:32			6	A	0 0 0 0 1	0 1 0 1 1	a	0 1 0 0 0	1	0 0 0 0 0	0 1 0 1 1	0 0 0 0 0	a	a
	XOR.B @aa:32,@ERd			6	A	0 0 0 1 1	0 1 0 1 1	a	0 0 0 0 0	0	rd	0 1 0 1 1	0 0 0 0 0		
	XOR.B @aa:32,@ERd+			6	A	0 0 0 1 1	0 1 0 1 1	a	1 0 0 0 0	0	rd	0 1 0 1 1	0 0 0 0 0		
	XOR.B @aa:32,@ERd-			6	A	0 0 0 1 1	0 1 0 1 1	a	1 0 1 0 0	0	rd	0 1 0 1 1	0 0 0 0 0		
	XOR.B @aa:32,@+ERd			6	A	0 0 0 1 1	0 1 0 1 1	a	1 0 0 0 1	0	rd	0 1 0 1 1	0 0 0 0 0		
	XOR.B @aa:32,@-ERd			6	A	0 0 0 1 1	0 1 0 1 1	a	1 0 1 1 1	0	rd	0 1 0 1 1	0 0 0 0 0		
	XOR.B @aa:32,@(d:2,ERd)			6	A	0 0 0 1 1	0 1 0 1 1	a	0 0 d d	0	rd	0 1 0 1 1	0 0 0 0 0		
	XOR.B @aa:32,@(d:16,ERd)			6	A	0 0 0 1 1	0 1 0 1 1	a	1 1 0 0 0	0	rd	0 1 0 1 1	0 0 0 0 0	d	
	XOR.B @aa:32,@(d:32,ERd)			6	A	0 0 0 1 1	0 1 0 1 1	a	1 1 0 0 1	1	rd	0 1 0 1 1	0 0 0 0 0	d	d
	XOR.B @aa:32,@(d:16,Rd,B)			6	A	0 0 0 1 1	0 1 0 1 1	a	1 1 0 1 1	0	rd	0 1 0 1 1	0 0 0 0 0	d	
	XOR.B @aa:32,@(d:16,Rd,W)			6	A	0 0 0 1 1	0 1 0 1 1	a	1 1 1 1 0	0	rd	0 1 0 1 1	0 0 0 0 0	d	
	XOR.B @aa:32,@(d:16,ERd,L)			6	A	0 0 0 1 1	0 1 0 1 1	a	1 1 1 1 1	0	rd	0 1 0 1 1	0 0 0 0 0	d	
	XOR.B @aa:32,@(d:32,Rd,B)			6	A	0 0 0 1 1	0 1 0 1 1	a	1 1 0 1 1	1	rd	0 1 0 1 1	0 0 0 0 0	d	d
	XOR.B @aa:32,@(d:32,Rd,W)			6	A	0 0 0 1 1	0 1 0 1 1	a	1 1 1 1 0	1	rd	0 1 0 1 1	0 0 0 0 0	d	d
	XOR.B @aa:32,@(d:32,ERd,L)			6	A	0 0 0 1 1	0 1 0 1 1	a	1 1 1 1 1	1	rd	0 1 0 1 1	0 0 0 0 0	d	d
	XOR.B @aa:32,@aa:16			6	A	0 0 0 1 1	0 1 0 1 1	a	0 1 0 0 0	0	0 0 0 0 0	0 1 0 1 1	0 0 0 0 0	a	
	XOR.B @aa:32,@aa:32			6	A	0 0 0 1 1	0 1 0 1 1	a	0 1 0 0 0	1	0 0 0 0 0	0 1 0 1 1	0 0 0 0 0	a	a
	XOR.W #xx:16,Rd									7		9	0 1 0 1 1	rd	x
	XOR.W #xx:16,@ERd			0	1	5	1 1 1 1 0		0 0 0 0 0	0	rd	0 1 0 1 1	0 0 0 0 0		x
	XOR.W #xx:16,@ERd+			0	1	5	1 1 1 1 0		1 0 0 0 0	0	rd	0 1 0 1 1	0 0 0 0 0		x
	XOR.W #xx:16,@ERd-			0	1	5	1 1 1 1 0		1 0 1 0 0	0	rd	0 1 0 1 1	0 0 0 0 0		x
	XOR.W #xx:16,@+ERd			0	1	5	1 1 1 1 0		1 0 0 1 0	0	rd	0 1 0 1 1	0 0 0 0 0		x
	XOR.W #xx:16,@-ERd			0	1	5	1 1 1 1 0		1 0 1 1 0	0	rd	0 1 0 1 1	0 0 0 0 0		x
	XOR.W #xx:16,@(d:2,ERd)			0	1	5	1 1 1 1 0		0 0 d d	0	rd	0 1 0 1 1	0 0 0 0 0		x
	XOR.W #xx:16,@(d:16,ERd)			0	1	5	1 1 1 1 0		1 1 0 0 0	0	rd	0 1 0 1 1	0 0 0 0 0	d	x
XOR.W #xx:16,@(d:32,ERd)			0	1	5	1 1 1 1 0		1 1 0 0 1	1	rd	0 1 0 1 1	0 0 0 0 0	d	d	
XOR.W #xx:16,@(d:16,Rd,B)			0	1	5	1 1 1 1 0		1 1 0 1 0	0	rd	0 1 0 1 1	0 0 0 0 0	d	x	

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0		15	8	7	0		
XOR	XOR.W #xoc:16,@(d:16,Rd,W)	0	1	5	1 1 1 0	1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d	x
	XOR.W #xoc:16,@(d:16,ERd,L)	0	1	5	1 1 1 0	1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d	x
	XOR.W #xoc:16,@(d:32,Rd,B)	0	1	5	1 1 1 0	1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d	d
	XOR.W #xoc:16,@(d:32,Rd,W)	0	1	5	1 1 1 0	1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d	d
	XOR.W #xoc:16,@(d:32,ERd,L)	0	1	5	1 1 1 0	1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d	d
	XOR.W #xoc:16,@aa:16	0	1	5	1 1 1 0	0 1 0 0	0	0 0 0 0	0 1 0 1	0 0 0 0	a	x
	XOR.W #xoc:16,@aa:32	0	1	5	1 1 1 0	0 1 0 0	1	0 0 0 0	0 1 0 1	0 0 0 0	a	a
	XOR.W Rs,Rd						6	5	rs	rd		
	XOR.W Rs,@ERd		7	D	1	rd	0 0 0 0	6	5	rs	0 0 0 0	
	XOR.W Rs,@ERd+		0	1	5	1 0 0 1	1 0 0 0	0	rd	0 1 0 1	rs	
	XOR.W Rs,@ERd-		0	1	5	1 0 0 1	1 0 1 0	0	rd	0 1 0 1	rs	
	XOR.W Rs,@+ERd		0	1	5	1 0 0 1	1 0 0 1	0	rd	0 1 0 1	rs	
	XOR.W Rs,@-ERd		0	1	5	1 0 0 1	1 0 1 1	0	rd	0 1 0 1	rs	
	XOR.W Rs,@(d:2,ERd)		0	1	5	1 0 0 1	0 0 d d	0	rd	0 1 0 1	rs	
	XOR.W Rs,@(d:16,ERd)		0	1	5	1 0 0 1	1 1 0 0	0	rd	0 1 0 1	rs	d
	XOR.W Rs,@(d:32,ERd)		0	1	5	1 0 0 1	1 1 0 0	1	rd	0 1 0 1	rs	d
	XOR.W Rs,@(d:16,Rd,B)		0	1	5	1 0 0 1	1 1 0 1	0	rd	0 1 0 1	rs	d
	XOR.W Rs,@(d:16,Rd,W)		0	1	5	1 0 0 1	1 1 1 0	0	rd	0 1 0 1	rs	d
	XOR.W Rs,@(d:16,ERd,L)		0	1	5	1 0 0 1	1 1 1 1	0	rd	0 1 0 1	rs	d
	XOR.W Rs,@(d:32,Rd,B)		0	1	5	1 0 0 1	1 1 0 1	1	rd	0 1 0 1	rs	d
	XOR.W Rs,@(d:32,Rd,W)		0	1	5	1 0 0 1	1 1 1 0	1	rd	0 1 0 1	rs	d
	XOR.W Rs,@(d:32,ERd,L)		0	1	5	1 0 0 1	1 1 1 1	1	rd	0 1 0 1	rs	d
	XOR.W Rs,@aa:16		6	B	0 0 0 1	1 0 0 0	a	6	5	rs	0 0 0 0	
	XOR.W Rs,@aa:32		6	B	0 0 1 1	1 0 0 0	a	a	6	5	rs	0 0 0 0
	XOR.W @ERs,Rd		7	C	1	rs	0 0 0 0	6	5	0 0 0 0	rd	
	XOR.W @ERs+,Rd		0	1	5	1 0 1 0	1 0 0 0	0	rs	0 1 0 1	rd	
	XOR.W @ERs-,Rd		0	1	5	1 0 1 0	1 0 1 0	0	rs	0 1 0 1	rd	
	XOR.W @+ERs,Rd		0	1	5	1 0 1 0	1 0 0 1	0	rs	0 1 0 1	rd	
	XOR.W @-ERs,Rd		0	1	5	1 0 1 0	1 0 1 1	0	rs	0 1 0 1	rd	
	XOR.W @(d:2,ERs),Rd		0	1	5	1 0 1 0	0 0 d d	0	rs	0 1 0 1	rd	
	XOR.W @(d:16,ERs),Rd		0	1	5	1 0 1 0	1 1 0 0	0	rs	0 1 0 1	rd	d
	XOR.W @(d:32,ERs),Rd		0	1	5	1 0 1 0	1 1 0 0	1	rs	0 1 0 1	rd	d
XOR.W @(d:16,Rs,B),Rd		0	1	5	1 0 1 0	1 1 0 1	0	rs	0 1 0 1	rd	d	
XOR.W @(d:16,Rs,W),Rd		0	1	5	1 0 1 0	1 1 1 0	0	rs	0 1 0 1	rd	d	

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension																			
		15	8	7	0	15	8	7	0		15	8	7	0																				
XOR	XOR.W @(d:16,ERs),Rd				0	1		5	1	0	1	0	1	1	1	1	0	rs	0	1	0	1	rd	d										
	XOR.W @(d:32,Rs.B),Rd				0	1		5	1	0	1	0	1	1	1	0	1	1	rs	0	1	0	1	rd	d	d								
	XOR.W @(d:32,Rs.W),Rd				0	1		5	1	0	1	0	1	1	1	1	0	1	rs	0	1	0	1	rd	d	d								
	XOR.W @(d:32,ERs.L),Rd				0	1		5	1	0	1	0	1	1	1	1	1	1	rs	0	1	0	1	rd	d	d								
	XOR.W @aa:16,Rd				6	B		0	0	0	0	1	0	0	0	0	0	a	6	5	0	0	0	0	rd									
	XOR.W @aa:32,Rd				6	B		0	0	1	1	0	0	0	0	0	a	a	6	5	0	0	0	0	rd									
	XOR.W @ERs,@ERd				7	C		1	rs	0	1	0	1	0	0	0	0	0	rd	0	1	0	1	0	0	0								
	XOR.W @ERs,@ERd+				7	C		1	rs	0	1	0	1	1	0	0	0	0	rd	0	1	0	1	0	0	0								
	XOR.W @ERs,@ERd-				7	C		1	rs	0	1	0	1	1	0	1	0	0	rd	0	1	0	1	0	0	0								
	XOR.W @ERs,@+ERd				7	C		1	rs	0	1	0	1	1	0	0	1	0	rd	0	1	0	1	0	0	0								
	XOR.W @ERs,@-ERd				7	C		1	rs	0	1	0	1	1	0	1	1	0	rd	0	1	0	1	0	0	0								
	XOR.W @ERs,@(d:2,ERd)				7	C		1	rs	0	1	0	1	0	0	d	d	0	rd	0	1	0	1	0	0	0								
	XOR.W @ERs,@(d:16,ERd)				7	C		1	rs	0	1	0	1	1	1	0	0	0	rd	0	1	0	1	0	0	0	d							
	XOR.W @ERs,@(d:32,ERd)				7	C		1	rs	0	1	0	1	1	1	0	0	1	rd	0	1	0	1	0	0	0	d	d						
	XOR.W @ERs,@(d:16,Rd.B)				7	C		1	rs	0	1	0	1	1	1	0	1	0	rd	0	1	0	1	0	0	0	d							
	XOR.W @ERs,@(d:16,Rd.W)				7	C		1	rs	0	1	0	1	1	1	1	0	0	rd	0	1	0	1	0	0	0	d							
	XOR.W @ERs,@(d:16,ERd.L)				7	C		1	rs	0	1	0	1	1	1	1	1	0	rd	0	1	0	1	0	0	0	d							
	XOR.W @ERs,@(d:32,Rd.B)				7	C		1	rs	0	1	0	1	1	1	0	1	1	rd	0	1	0	1	0	0	0	d	d						
	XOR.W @ERs,@(d:32,Rd.W)				7	C		1	rs	0	1	0	1	1	1	1	0	1	rd	0	1	0	1	0	0	0	d	d						
	XOR.W @ERs,@(d:32,ERd.L)				7	C		1	rs	0	1	0	1	1	1	1	1	1	rd	0	1	0	1	0	0	0	d	d						
	XOR.W @ERs,@aa:16				7	C		1	rs	0	1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	a						
	XOR.W @ERs,@aa:32				7	C		1	rs	0	1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	a	a					
	XOR.W @ERs+,@ERd	0		1		5		0	1	0	0	0	6	D	0	rs	1	1	0	0	0	0	0	0	rd	0	1	0	1	0	0	0	0	
	XOR.W @ERs+,@ERd+	0		1		5		0	1	0	0	0	6	D	0	rs	1	1	0	0	1	0	0	0	rd	0	1	0	1	0	0	0	0	
	XOR.W @ERs+,@ERd-	0		1		5		0	1	0	0	0	6	D	0	rs	1	1	0	0	1	0	0	0	rd	0	1	0	1	0	0	0	0	
	XOR.W @ERs+,@+ERd	0		1		5		0	1	0	0	0	6	D	0	rs	1	1	0	0	1	0	0	0	rd	0	1	0	1	0	0	0	0	
	XOR.W @ERs+,@-ERd	0		1		5		0	1	0	0	0	6	D	0	rs	1	1	0	0	1	0	0	0	rd	0	1	0	1	0	0	0	0	
	XOR.W @ERs+,@(d:2,ERd)	0		1		5		0	1	0	0	0	6	D	0	rs	1	1	0	0	0	0	d	d	0	rd	0	1	0	1	0	0	0	0
	XOR.W @ERs+,@(d:16,ERd)	0		1		5		0	1	0	0	0	6	D	0	rs	1	1	0	0	0	0	0	rd	0	1	0	1	0	0	0	0	d	
	XOR.W @ERs+,@(d:32,ERd)	0		1		5		0	1	0	0	0	6	D	0	rs	1	1	0	0	1	0	0	0	rd	0	1	0	1	0	0	0	d	d
	XOR.W @ERs+,@(d:16,Rd.B)	0		1		5		0	1	0	0	0	6	D	0	rs	1	1	0	0	0	0	0	rd	0	1	0	1	0	0	0	0	d	
	XOR.W @ERs+,@(d:16,Rd.W)	0		1		5		0	1	0	0	0	6	D	0	rs	1	1	0	0	0	0	0	rd	0	1	0	1	0	0	0	0	d	
XOR.W @ERs+,@(d:16,ERd.L)	0		1		5		0	1	0	0	0	6	D	0	rs	1	1	0	0	0	0	0	rd	0	1	0	1	0	0	0	0	d		
XOR.W @ERs+,@(d:32,Rd.B)	0		1		5		0	1	0	0	0	6	D	0	rs	1	1	0	0	1	0	0	0	rd	0	1	0	1	0	0	0	d	d	



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
XOR	XOR.W @ERs+,@(d:32,Rd,W)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0		1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @ERs+,@(d:32,ERd,L)	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0		1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @ERs+,@aa:16	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0		0 1 0 0	0	0 0 0 0	0 1 0 1	0 0 0 0	a
	XOR.W @ERs+,@aa:32	0	1	5	0 1 0 0	6	D	0	rs	1 1 0 0		0 1 0 0	1	0 0 0 0	0 1 0 1	0 0 0 0	a a
	XOR.W @ERs-,@ERd	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		0 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @ERs-,@ERd+	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @ERs-,@ERd-	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 0 1 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @ERs-,@+ERd	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 0 0 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @ERs-,@-ERd	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 0 1 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @ERs-,@(d:2,ERd)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		0 0 d d	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @ERs-,@(d:16,ERd)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @ERs-,@(d:32,ERd)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @ERs-,@(d:16,Rd,B)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @ERs-,@(d:16,Rd,W)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @ERs-,@(d:16,ERd,L)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @ERs-,@(d:32,Rd,B)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @ERs-,@(d:32,Rd,W)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @ERs-,@(d:32,ERd,L)	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @ERs-,@aa:16	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		0 1 0 0	0	0 0 0 0	0 1 0 1	0 0 0 0	a
	XOR.W @ERs-,@aa:32	0	1	5	0 1 1 0	6	D	0	rs	1 1 0 0		0 1 0 0	1	0 0 0 0	0 1 0 1	0 0 0 0	a a
	XOR.W @+ERs,@ERd	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		0 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @+ERs,@ERd+	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @+ERs,@ERd-	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 0 1 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @+ERs,@+ERd	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 0 0 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @+ERs,@-ERd	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 0 1 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @+ERs,@(d:2,ERd)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		0 0 d d	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @+ERs,@(d:16,ERd)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @+ERs,@(d:32,ERd)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d d
XOR.W @+ERs,@(d:16,Rd,B)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d	
XOR.W @+ERs,@(d:16,Rd,W)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d	
XOR.W @+ERs,@(d:16,ERd,L)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d	
XOR.W @+ERs,@(d:32,Rd,B)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d	
XOR.W @+ERs,@(d:32,Rd,W)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d	
XOR.W @+ERs,@(d:32,ERd,L)	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d	

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
XOR	XOR.W @+ERs,@aa:16	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		0 1 0 0	0 0 0 0	0 1 0 1	0 0 0 0	a
	XOR.W @+ERs,@aa:32	0	1	5	0 1 0 1	6	D	0	rs	1 1 0 0		0 1 0 0	1 0 0 0	0 1 0 1	0 0 0 0	a a
	XOR.W @-ERs,@ERd	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		0 0 0 0	0 rd	0 1 0 1	0 0 0 0	
	XOR.W @-ERs,@ERd+	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 0 0	0 rd	0 1 0 1	0 0 0 0	
	XOR.W @-ERs,@ERd-	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 1 0	0 rd	0 1 0 1	0 0 0 0	
	XOR.W @-ERs,@+ERd	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 0 1	0 rd	0 1 0 1	0 0 0 0	
	XOR.W @-ERs,@-ERd	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 1 1	0 rd	0 1 0 1	0 0 0 0	
	XOR.W @-ERs,@(d:2,ERd)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		0 0 d d	0 rd	0 1 0 1	0 0 0 0	
	XOR.W @-ERs,@(d:16,ERd)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 0	0 rd	0 1 0 1	0 0 0 0	d
	XOR.W @-ERs,@(d:32,ERd)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 0	1 rd	0 1 0 1	0 0 0 0	d d
	XOR.W @-ERs,@(d:16,Rd.B)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 1	0 rd	0 1 0 1	0 0 0 0	d
	XOR.W @-ERs,@(d:16,Rd.W)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 0	0 rd	0 1 0 1	0 0 0 0	d
	XOR.W @-ERs,@(d:16,ERd.L)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 1	0 rd	0 1 0 1	0 0 0 0	d
	XOR.W @-ERs,@(d:32,Rd.B)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 1	1 rd	0 1 0 1	0 0 0 0	d d
	XOR.W @-ERs,@(d:32,Rd.W)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 0	1 rd	0 1 0 1	0 0 0 0	d d
	XOR.W @-ERs,@(d:32,ERd.L)	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 1	1 rd	0 1 0 1	0 0 0 0	d d
	XOR.W @-ERs,@aa:16	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		0 1 0 0	0 0 0 0	0 1 0 1	0 0 0 0	a
	XOR.W @-ERs,@aa:32	0	1	5	0 1 1 1	6	D	0	rs	1 1 0 0		0 1 0 0	1 0 0 0	0 1 0 1	0 0 0 0	a a
	XOR.W @(d:2,ERs),@ERd	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		0 0 0 0	0 rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:2,ERs),@ERd+	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 0 0 0	0 rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:2,ERs),@ERd-	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 0 1 0	0 rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:2,ERs),@+ERd	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 0 0 1	0 rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:2,ERs),@-ERd	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 0 1 1	0 rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:2,ERs),@(d:2,ERd)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		0 0 d d	0 rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:2,ERs),@(d:16,ERd)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 0	0 rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:2,ERs),@(d:32,ERd)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 0	1 rd	0 1 0 1	0 0 0 0	d d
	XOR.W @(d:2,ERs),@(d:16,Rd.B)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 1	0 rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:2,ERs),@(d:16,Rd.W)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 1 0	0 rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:2,ERs),@(d:16,ERd.L)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 1 1	0 rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:2,ERs),@(d:32,Rd.B)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 0 1	1 rd	0 1 0 1	0 0 0 0	d d
	XOR.W @(d:2,ERs),@(d:32,Rd.W)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 1 0	1 rd	0 1 0 1	0 0 0 0	d d
	XOR.W @(d:2,ERs),@(d:32,ERd.L)	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		1 1 1 1	1 rd	0 1 0 1	0 0 0 0	d d
XOR.W @(d:2,ERs),@aa:16	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		0 1 0 0	0 0 0 0	0 1 0 1	0 0 0 0	a	
XOR.W @(d:2,ERs),@aa:32	0	1	5	0 1 d d	6	9	0	rs	1 1 0 0		0 1 0 0	1 0 0 0	0 1 0 1	0 0 0 0	a a	



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension			
		15	8	7	0	15	8	7	0		15	8	7	0				
XOR	XOR.W @(d:16,ERs),@ERd	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	0	0 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:16,ERs),@ERd+	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	0 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:16,ERs),@ERd-	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	0 1 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:16,ERs),@+ERd	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	0 0 1 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:16,ERs),@-ERd	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	0 1 1 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:16,ERs),@(d:2,ERd)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	0	0 d d 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:16,ERs),@(d:16,ERd)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 0 0 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:16,ERs),@(d:32,ERd)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 0 0 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @(d:16,ERs),@(d:16,Rd,B)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 0 1 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:16,ERs),@(d:16,Rd,W)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:16,ERs),@(d:16,ERd,L)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:16,ERs),@(d:32,Rd,B)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 0 1 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @(d:16,ERs),@(d:32,Rd,W)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @(d:16,ERs),@(d:32,ERd,L)	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	1	1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @(d:16,ERs),@aa:16	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	0	1 0 0 0	0 0 0 0	0 1 0 1	0 0 0 0	a	
	XOR.W @(d:16,ERs),@aa:32	0	1	5	0 1 0 0	6	F	0	rs	1 1 0 0	d	0	1 0 0 0	1 0 0 0	0 1 0 1	0 0 0 0	a a	
	XOR.W @(d:32,ERs),@ERd	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0	0 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:32,ERs),@ERd+	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	0 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:32,ERs),@ERd-	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	0 1 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:32,ERs),@+ERd	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	0 0 1 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:32,ERs),@-ERd	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	0 1 1 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:32,ERs),@(d:2,ERd)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0	0 d d 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:32,ERs),@(d:16,ERd)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	1 0 0 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:32,ERs),@(d:32,ERd)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	1 0 0 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @(d:32,ERs),@(d:16,Rd,B)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	1 0 1 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:32,ERs),@(d:16,Rd,W)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:32,ERs),@(d:16,ERd,L)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:32,ERs),@(d:32,Rd,B)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	1 0 1 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @(d:32,ERs),@(d:32,Rd,W)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @(d:32,ERs),@(d:32,ERd,L)	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1	1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @(d:32,ERs),@aa:16	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0	1 0 0 0	0 0 0 0	0 1 0 1	0 0 0 0	a	
	XOR.W @(d:32,ERs),@aa:32	7	8	0	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0	1 0 0 0	1 0 0 0	0 1 0 1	0 0 0 0	a a	
XOR.W @(d:16,Rs,B),@ERd	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	0	0 0 0 0	0	rd	0 1 0 1	0 0 0 0		
XOR.W @(d:16,Rs,B),@ERd+	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1	0 0 0 0	0	rd	0 1 0 1	0 0 0 0		

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
XOR	XOR.W @(d:16,Rs.B),@ERd-	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:16,Rs.B),@+ERd	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:16,Rs.B),@-ERd	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:16,Rs.B),@(d:2,ERd)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:16,Rs.B),@(d:16,ERd)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:16,Rs.B),@(d:32,ERd)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @(d:16,Rs.B),@(d:16,Rd.B)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:16,Rs.B),@(d:16,Rd.W)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:16,Rs.B),@(d:16,ERd.L)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:16,Rs.B),@(d:32,Rd.B)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @(d:16,Rs.B),@(d:32,Rd.W)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @(d:16,Rs.B),@(d:32,ERd.L)	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @(d:16,Rs.B),@aa:16	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 1 0 1	0 0 0 0	a	
	XOR.W @(d:16,Rs.B),@aa:32	0	1	5	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 0 1	0 0 0 0	a a	
	XOR.W @(d:16,Rs.W),@ERd	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:16,Rs.W),@ERd+	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:16,Rs.W),@ERd-	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:16,Rs.W),@+ERd	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:16,Rs.W),@-ERd	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:16,Rs.W),@(d:2,ERd)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:16,Rs.W),@(d:16,ERd)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:16,Rs.W),@(d:32,ERd)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @(d:16,Rs.W),@(d:16,Rd.B)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:16,Rs.W),@(d:16,Rd.W)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:16,Rs.W),@(d:16,ERd.L)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:16,Rs.W),@(d:32,Rd.B)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @(d:16,Rs.W),@(d:32,Rd.W)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @(d:16,Rs.W),@(d:32,ERd.L)	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @(d:16,Rs.W),@aa:16	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 1 0 1	0 0 0 0	a	
	XOR.W @(d:16,Rs.W),@aa:32	0	1	5	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 0 1	0 0 0 0	a a	
	XOR.W @(d:16,ERs.L),@ERd	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:16,ERs.L),@ERd+	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 1 0 1	0 0 0 0	
XOR.W @(d:16,ERs.L),@ERd-	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 0 1	0 0 0 0		
XOR.W @(d:16,ERs.L),@+ERd	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 0 1	0 0 0 0		



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
XOR	XOR.W @(d:16,ERs.L),@ERd	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:16,ERs.L),@(d:2,ERd)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:16,ERs.L),@(d:16,ERd)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:16,ERs.L),@(d:32,ERd)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @(d:16,ERs.L),@(d:16,Rd.B)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:16,ERs.L),@(d:16,Rd.W)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:16,ERs.L),@(d:16,ERd.L)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:16,ERs.L),@(d:32,Rd.B)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @(d:16,ERs.L),@(d:32,Rd.W)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @(d:16,ERs.L),@(d:32,ERd.L)	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.W @(d:16,ERs.L),@aa:16	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 1 0 1	0 0 0 0	a	
	XOR.W @(d:16,ERs.L),@aa:32	0	1	5	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 0 1	0 0 0 0	a a	
	XOR.W @(d:32,Rs.B),@ERd	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:32,Rs.B),@ERd+	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:32,Rs.B),@ERd-	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:32,Rs.B),@+ERd	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:32,Rs.B),@-ERd	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:32,Rs.B),@(d:2,ERd)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 1 0 1	0 0 0 0	
	XOR.W @(d:32,Rs.B),@(d:16,ERd)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.W @(d:32,Rs.B),@(d:32,ERd)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d d
XOR.W @(d:32,Rs.B),@(d:16,Rd.B)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d	
XOR.W @(d:32,Rs.B),@(d:16,Rd.W)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d	
XOR.W @(d:32,Rs.B),@(d:16,ERd.L)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d	
XOR.W @(d:32,Rs.B),@(d:32,Rd.B)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d	
XOR.W @(d:32,Rs.B),@(d:32,Rd.W)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d	
XOR.W @(d:32,Rs.B),@(d:32,ERd.L)	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d	
XOR.W @(d:32,Rs.B),@aa:16	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0 0 0 0	0 1 0 1	0 0 0 0	a		
XOR.W @(d:32,Rs.B),@aa:32	7	8	0	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1 0 0 0	0 1 0 1	0 0 0 0	a a		
XOR.W @(d:32,Rs.W),@ERd	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 1 0 1	0 0 0 0		
XOR.W @(d:32,Rs.W),@ERd+	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 1 0 1	0 0 0 0		
XOR.W @(d:32,Rs.W),@ERd-	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 1 0 1	0 0 0 0		
XOR.W @(d:32,Rs.W),@+ERd	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 1 0 1	0 0 0 0		
XOR.W @(d:32,Rs.W),@-ERd	7	8	0	rs	0 1 1 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 1 0 1	0 0 0 0		

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension																								
		15	8	7	0	15	8	7	0		15	8	7	0																									
XOR	XOR.W @(d:32.Rs.W),@ERd	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	0	1	1	0	rd	0	1	0	1	0	0	0	0	0			
	XOR.W @(d:32.Rs.W),@(d:2,ERd)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	0	0	d	d	0	rd	0	1	0	1	0	0	0	0	0			
	XOR.W @(d:32.Rs.W),@(d:16,ERd)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	0	0	rd	0	1	0	1	0	0	0	0	0	d		
	XOR.W @(d:32.Rs.W),@(d:32,ERd)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	0	1	rd	0	1	0	1	0	0	0	0	0	d	d	
	XOR.W @(d:32.Rs.W),@(d:16,Rd.B)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	1	0	rd	0	1	0	1	0	0	0	0	0	d		
	XOR.W @(d:32.Rs.W),@(d:16,Rd.W)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	0	0	rd	0	1	0	1	0	0	0	0	0	d		
	XOR.W @(d:32.Rs.W),@(d:16,ERd.L)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	1	0	rd	0	1	0	1	0	0	0	0	0	d		
	XOR.W @(d:32.Rs.W),@(d:32,Rd.B)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	1	1	rd	0	1	0	1	0	0	0	0	0	d	d	
	XOR.W @(d:32.Rs.W),@(d:32,Rd.W)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	0	1	rd	0	1	0	1	0	0	0	0	0	d	d	
	XOR.W @(d:32.Rs.W),@(d:32,ERd.L)	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	1	1	rd	0	1	0	1	0	0	0	0	0	d	d	
	XOR.W @(d:32.Rs.W),@aa:16	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	0	1	0	0	0	0	rd	0	1	0	1	0	0	0	0	0	a	
	XOR.W @(d:32.Rs.W),@aa:32	7	8	0	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	0	1	0	0	1	0	rd	0	1	0	1	0	0	0	0	0	a	a
	XOR.W @(d:32,ERs.L),@ERd	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	0	0	0	0	0	rd	0	1	0	1	0	0	0	0	0			
	XOR.W @(d:32,ERs.L),@ERd+	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	0	0	0	0	rd	0	1	0	1	0	0	0	0	0			
	XOR.W @(d:32,ERs.L),@ERd-	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	0	1	0	0	rd	0	1	0	1	0	0	0	0	0			
	XOR.W @(d:32,ERs.L),@+ERd	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	0	0	1	0	rd	0	1	0	1	0	0	0	0	0			
	XOR.W @(d:32,ERs.L),@ERd	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	0	1	1	0	rd	0	1	0	1	0	0	0	0	0			
	XOR.W @(d:32,ERs.L),@(d:2,ERd)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	0	0	d	d	0	rd	0	1	0	1	0	0	0	0	0			
	XOR.W @(d:32,ERs.L),@(d:16,ERd)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	0	0	rd	0	1	0	1	0	0	0	0	0	d		
	XOR.W @(d:32,ERs.L),@(d:32,ERd)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	0	1	rd	0	1	0	1	0	0	0	0	0	d	d	
	XOR.W @(d:32,ERs.L),@(d:16,Rd.B)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	1	0	rd	0	1	0	1	0	0	0	0	0	d		
	XOR.W @(d:32,ERs.L),@(d:16,Rd.W)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	0	0	rd	0	1	0	1	0	0	0	0	0	d		
	XOR.W @(d:32,ERs.L),@(d:16,ERd.L)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	1	0	rd	0	1	0	1	0	0	0	0	0	d		
	XOR.W @(d:32,ERs.L),@(d:32,Rd.B)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	1	1	rd	0	1	0	1	0	0	0	0	0	d	d	
	XOR.W @(d:32,ERs.L),@(d:32,Rd.W)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	0	1	rd	0	1	0	1	0	0	0	0	0	d	d	
	XOR.W @(d:32,ERs.L),@(d:32,ERd.L)	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	1	1	rd	0	1	0	1	0	0	0	0	0	d	d	
	XOR.W @(d:32,ERs.L),@aa:16	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	0	1	0	0	0	0	rd	0	1	0	1	0	0	0	0	0	a	
	XOR.W @(d:32,ERs.L),@aa:32	7	8	0	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	0	1	0	0	1	0	rd	0	1	0	1	0	0	0	0	0	a	a
	XOR.W @aa:16,@ERd									6	B	0	0	0	1	0	1	0	1	a		0	0	0	0	0	rd	0	1	0	1	0	0	0	0	0			
	XOR.W @aa:16,@ERd+									6	B	0	0	0	1	0	1	0	1	a		1	0	0	0	0	rd	0	1	0	1	0	0	0	0	0			
	XOR.W @aa:16,@ERd-									6	B	0	0	0	1	0	1	0	1	a		1	0	1	0	0	rd	0	1	0	1	0	0	0	0	0			
	XOR.W @aa:16,@+ERd									6	B	0	0	0	1	0	1	0	1	a		1	0	0	1	0	rd	0	1	0	1	0	0	0	0	0			



Instruction	Mnemonic	Opcode				EA Ex- tension	Opcode				EA Extension	
		15	8	7	0		15	8	7	0		
XOR	XOR.W @aa:16, @-ERd	6	B	0 0 0 0 1	0 1 0 1 0	a	1 0 0 1 1	0	rd	0 1 0 1 1	0 0 0 0 0	
	XOR.W @aa:16, @(d:2,ERd)	6	B	0 0 0 0 1	0 1 0 1 0	a	0 0 0 d d	0	rd	0 1 0 1 1	0 0 0 0 0	
	XOR.W @aa:16, @(d:16,ERd)	6	B	0 0 0 0 1	0 1 0 1 0	a	1 1 0 0 0	0	rd	0 1 0 1 1	0 0 0 0 0	d
	XOR.W @aa:16, @(d:32,ERd)	6	B	0 0 0 0 1	0 1 0 1 0	a	1 1 0 0 0	1	rd	0 1 0 1 1	0 0 0 0 0	d d
	XOR.W @aa:16, @(d:16,Rd,B)	6	B	0 0 0 0 1	0 1 0 1 0	a	1 1 0 0 1	0	rd	0 1 0 1 1	0 0 0 0 0	d
	XOR.W @aa:16, @(d:16,Rd,W)	6	B	0 0 0 0 1	0 1 0 1 0	a	1 1 1 0 0	0	rd	0 1 0 1 1	0 0 0 0 0	d
	XOR.W @aa:16, @(d:16,ERd,L)	6	B	0 0 0 0 1	0 1 0 1 0	a	1 1 1 1 0	0	rd	0 1 0 1 1	0 0 0 0 0	d
	XOR.W @aa:16, @(d:32,Rd,B)	6	B	0 0 0 0 1	0 1 0 1 0	a	1 1 0 0 1	1	rd	0 1 0 1 1	0 0 0 0 0	d d
	XOR.W @aa:16, @(d:32,Rd,W)	6	B	0 0 0 0 1	0 1 0 1 0	a	1 1 1 0 1	1	rd	0 1 0 1 1	0 0 0 0 0	d d
	XOR.W @aa:16, @(d:32,ERd,L)	6	B	0 0 0 0 1	0 1 0 1 0	a	1 1 1 1 1	1	rd	0 1 0 1 1	0 0 0 0 0	d d
	XOR.W @aa:16, @aa:16	6	B	0 0 0 0 1	0 1 0 1 0	a	0 1 0 0 0	0 0 0 0 0		0 1 0 1 1	0 0 0 0 0	a
	XOR.W @aa:16, @aa:32	6	B	0 0 0 0 1	0 1 0 1 0	a	0 1 0 0 0	1 0 0 0 0		0 1 0 1 1	0 0 0 0 0	a a
	XOR.W @aa:32, @ERd	6	B	0 0 0 1 1	0 1 0 1 0	a a	0 0 0 0 0	0	rd	0 1 0 1 1	0 0 0 0 0	
	XOR.W @aa:32, @ERd+	6	B	0 0 0 1 1	0 1 0 1 0	a a	1 0 0 0 0	0	rd	0 1 0 1 1	0 0 0 0 0	
	XOR.W @aa:32, @ERd-	6	B	0 0 0 1 1	0 1 0 1 0	a a	1 0 1 0 0	0	rd	0 1 0 1 1	0 0 0 0 0	
	XOR.W @aa:32, @+ERd	6	B	0 0 0 1 1	0 1 0 1 0	a a	1 0 0 1 0	0	rd	0 1 0 1 1	0 0 0 0 0	
	XOR.W @aa:32, @-ERd	6	B	0 0 0 1 1	0 1 0 1 0	a a	1 0 1 1 0	0	rd	0 1 0 1 1	0 0 0 0 0	
	XOR.W @aa:32, @(d:2,ERd)	6	B	0 0 0 1 1	0 1 0 1 0	a a	0 0 d d	0	rd	0 1 0 1 1	0 0 0 0 0	
	XOR.W @aa:32, @(d:16,ERd)	6	B	0 0 0 1 1	0 1 0 1 0	a a	1 1 0 0 0	0	rd	0 1 0 1 1	0 0 0 0 0	d
	XOR.W @aa:32, @(d:32,ERd)	6	B	0 0 0 1 1	0 1 0 1 0	a a	1 1 0 0 0	1	rd	0 1 0 1 1	0 0 0 0 0	d d
XOR.W @aa:32, @(d:16,Rd,B)	6	B	0 0 0 1 1	0 1 0 1 0	a a	1 1 0 0 1	0	rd	0 1 0 1 1	0 0 0 0 0	d	
XOR.W @aa:32, @(d:16,Rd,W)	6	B	0 0 0 1 1	0 1 0 1 0	a a	1 1 1 0 0	0	rd	0 1 0 1 1	0 0 0 0 0	d	
XOR.W @aa:32, @(d:16,ERd,L)	6	B	0 0 0 1 1	0 1 0 1 0	a a	1 1 1 1 0	0	rd	0 1 0 1 1	0 0 0 0 0	d	
XOR.W @aa:32, @(d:32,Rd,B)	6	B	0 0 0 1 1	0 1 0 1 0	a a	1 1 0 0 1	1	rd	0 1 0 1 1	0 0 0 0 0	d d	
XOR.W @aa:32, @(d:32,Rd,W)	6	B	0 0 0 1 1	0 1 0 1 0	a a	1 1 1 0 1	1	rd	0 1 0 1 1	0 0 0 0 0	d d	
XOR.W @aa:32, @(d:32,ERd,L)	6	B	0 0 0 1 1	0 1 0 1 0	a a	1 1 1 1 1	1	rd	0 1 0 1 1	0 0 0 0 0	d d	
XOR.W @aa:32, @aa:16	6	B	0 0 0 1 1	0 1 0 1 0	a a	0 1 0 0 0	0 0 0 0 0		0 1 0 1 1	0 0 0 0 0	a	
XOR.W @aa:32, @aa:32	6	B	0 0 0 1 1	0 1 0 1 0	a a	0 1 0 0 0	1 0 0 0 0		0 1 0 1 1	0 0 0 0 0	a a	
XOR.L #xx:16,ERd							7	A	0 1 0 1 1	1	rd	x
XOR.L #xx:32,ERd							7	A	0 1 0 1 1	0	rd	x x
XOR.L #xx:16, @ERd		0	1	0	1 1 1 1 0		0 0 0 0 0	0	rd	0 1 0 1 1	0 0 0 0 0	x
XOR.L #xx:16, @ERd+		0	1	0	1 1 1 1 0		1 0 0 0 0	0	rd	0 1 0 1 1	0 0 0 0 0	x
XOR.L #xx:16, @ERd-		0	1	0	1 1 1 1 0		1 0 1 0 0	0	rd	0 1 0 1 1	0 0 0 0 0	x
XOR.L #xx:16, @+ERd		0	1	0	1 1 1 1 0		1 0 0 1 0	0	rd	0 1 0 1 1	0 0 0 0 0	x

Instruction	Mnemonic	Opcode				EA Extension	Opcode				EA Extension						
		15	8	7	0		15	8	7	0							
XOR	XORL #xx:16, @-ERd				0	1	0	1 1 1 0	1 0 1 1	0	rd	0 1 0 1	0 0 0 0			x	
	XORL #xx:16, @(d:2,ERd)				0	1	0	1 1 1 0	0 0 d d	0	rd	0 1 0 1	0 0 0 0			x	
	XORL #xx:16, @(d:16,ERd)				0	1	0	1 1 1 0	1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d		x	
	XORL #xx:16, @(d:32,ERd)				0	1	0	1 1 1 0	1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d	d	x	
	XORL #xx:16, @(d:16,Rd.B)				0	1	0	1 1 1 0	1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d		x	
	XORL #xx:16, @(d:16,Rd.W)				0	1	0	1 1 1 0	1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d		x	
	XORL #xx:16, @(d:16,ERd.L)				0	1	0	1 1 1 0	1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d		x	
	XORL #xx:16, @(d:32,Rd.B)				0	1	0	1 1 1 0	1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d	d	x	
	XORL #xx:16, @(d:32,Rd.W)				0	1	0	1 1 1 0	1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d	d	x	
	XORL #xx:16, @(d:32,ERd.L)				0	1	0	1 1 1 0	1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d	d	x	
	XORL #xx:16, @aa:16				0	1	0	1 1 1 0	0 1 0 0	0	0 0 0 0	0 1 0 1	0 0 0 0	a		x	
	XORL #xx:16, @aa:32				0	1	0	1 1 1 0	0 1 0 0	1	0 0 0 0	0 1 0 1	0 0 0 0	a	a	x	
	XORL #xx:32, @ERd				0	1	0	1 1 1 0	0 0 0 0	0	rd	0 1 0 1	1 0 0 0			x	x
	XORL #xx:32, @ERd+				0	1	0	1 1 1 0	1 0 0 0	0	rd	0 1 0 1	1 0 0 0			x	x
	XORL #xx:32, @ERd-				0	1	0	1 1 1 0	1 0 1 0	0	rd	0 1 0 1	1 0 0 0			x	x
	XORL #xx:32, @+ERd				0	1	0	1 1 1 0	1 0 0 1	0	rd	0 1 0 1	1 0 0 0			x	x
	XORL #xx:32, @-ERd				0	1	0	1 1 1 0	1 0 1 1	0	rd	0 1 0 1	1 0 0 0			x	x
	XORL #xx:32, @(d:2,ERd)				0	1	0	1 1 1 0	0 0 d d	0	rd	0 1 0 1	1 0 0 0			x	x
	XORL #xx:32, @(d:16,ERd)				0	1	0	1 1 1 0	1 1 0 0	0	rd	0 1 0 1	1 0 0 0	d		x	x
	XORL #xx:32, @(d:32,ERd)				0	1	0	1 1 1 0	1 1 0 0	1	rd	0 1 0 1	1 0 0 0	d	d	x	x
	XORL #xx:32, @(d:16,Rd.B)				0	1	0	1 1 1 0	1 1 0 1	0	rd	0 1 0 1	1 0 0 0	d		x	x
	XORL #xx:32, @(d:16,Rd.W)				0	1	0	1 1 1 0	1 1 1 0	0	rd	0 1 0 1	1 0 0 0	d		x	x
	XORL #xx:32, @(d:16,ERd.L)				0	1	0	1 1 1 0	1 1 1 1	0	rd	0 1 0 1	1 0 0 0	d		x	x
	XORL #xx:32, @(d:32,Rd.B)				0	1	0	1 1 1 0	1 1 0 1	1	rd	0 1 0 1	1 0 0 0	d	d	x	x
	XORL #xx:32, @(d:32,Rd.W)				0	1	0	1 1 1 0	1 1 1 0	1	rd	0 1 0 1	1 0 0 0	d	d	x	x
	XORL #xx:32, @(d:32,ERd.L)				0	1	0	1 1 1 0	1 1 1 1	1	rd	0 1 0 1	1 0 0 0	d	d	x	x
	XORL #xx:32, @aa:16				0	1	0	1 1 1 0	0 1 0 0	0	0 0 0 0	0 1 0 1	1 0 0 0	a		x	x
	XORL #xx:32, @aa:32				0	1	0	1 1 1 0	0 1 0 0	1	0 0 0 0	0 1 0 1	1 0 0 0	a	a	x	x
	XORL ERs, ERd				0	1	F	0 0 0 0		6		5	0	rs	0	rd	
	XORL ERs, @ERd				0	1	0	1 0 0 1		0 0 0 0	0	rd	0 1 0 1	0	rs		
	XORL ERs, @ERd+				0	1	0	1 0 0 1		1 0 0 0	0	rd	0 1 0 1	0	rs		
	XORL ERs, @ERd-				0	1	0	1 0 0 1		1 0 1 0	0	rd	0 1 0 1	0	rs		
XORL ERs, @+ERd				0	1	0	1 0 0 1		1 0 0 1	0	rd	0 1 0 1	0	rs			
XORL ERs, @-ERd				0	1	0	1 0 0 1		1 0 1 1	0	rd	0 1 0 1	0	rs			



Instruction	Mnemonic	Opcode			Opcode			EA Extension	Opcode			EA Extension																	
		15	8	7	0	15	8		7	0	15		8	7	0														
XOR	XOR.L ERs, @(d:2,ERd)				0	1	0	1 0 0 1		0	0	d	d	0	rd	0	1	0	1	0	rs								
	XOR.L ERs, @(d:16,ERd)				0	1	0	1 0 0 1		1	1	0	0	0	rd	0	1	0	1	0	rs	d							
	XOR.L ERs, @(d:32,ERd)				0	1	0	1 0 0 1		1	1	0	0	1	rd	0	1	0	1	0	rs	d	d						
	XOR.L ERs, @(d:16,Rd,B)				0	1	0	1 0 0 1		1	1	0	1	0	rd	0	1	0	1	0	rs	d							
	XOR.L ERs, @(d:16,Rd,W)				0	1	0	1 0 0 1		1	1	1	0	0	rd	0	1	0	1	0	rs	d	d						
	XOR.L ERs, @(d:16,ERd,L)				0	1	0	1 0 0 1		1	1	1	1	0	rd	0	1	0	1	0	rs	d							
	XOR.L ERs, @(d:32,Rd,B)				0	1	0	1 0 0 1		1	1	0	1	1	rd	0	1	0	1	0	rs	d	d						
	XOR.L ERs, @(d:32,Rd,W)				0	1	0	1 0 0 1		1	1	1	0	1	rd	0	1	0	1	0	rs	d	d						
	XOR.L ERs, @(d:32,ERd,L)				0	1	0	1 0 0 1		1	1	1	1	1	rd	0	1	0	1	0	rs	d	d						
	XOR.L ERs, @aa:16				0	1	0	1 0 0 1		0	1	0	0	0	0	0	0	1	0	1	0	rs	a						
	XOR.L ERs, @aa:32				0	1	0	1 0 0 1		0	1	0	0	1	0	0	0	0	1	0	1	0	rs	a	a				
	XOR.L @ERs,ERd				0	1	0	1 0 1 0		0	0	0	0	0	rs	0	1	0	1	0	rd								
	XOR.L @ERs+,ERd				0	1	0	1 0 1 0		1	0	0	0	0	rs	0	1	0	1	0	rd								
	XOR.L @ERs-,ERd				0	1	0	1 0 1 0		1	0	1	0	0	rs	0	1	0	1	0	rd								
	XOR.L @+ERs,ERd				0	1	0	1 0 1 0		1	0	0	1	0	rs	0	1	0	1	0	rd								
	XOR.L @-ERs,ERd				0	1	0	1 0 1 0		1	0	1	1	0	rs	0	1	0	1	0	rd								
	XOR.L @(d:2,ERs),ERd				0	1	0	1 0 1 0		0	0	d	d	0	rs	0	1	0	1	0	rd								
	XOR.L @(d:16,ERs),ERd				0	1	0	1 0 1 0		1	1	0	0	0	rs	0	1	0	1	0	rd	d							
	XOR.L @(d:32,ERs),ERd				0	1	0	1 0 1 0		1	1	0	0	1	rs	0	1	0	1	0	rd	d	d						
	XOR.L @(d:16,Rs,B),ERd				0	1	0	1 0 1 0		1	1	0	1	0	rs	0	1	0	1	0	rd	d							
	XOR.L @(d:16,Rs,W),ERd				0	1	0	1 0 1 0		1	1	1	0	0	rs	0	1	0	1	0	rd	d							
	XOR.L @(d:16,ERs,L),ERd				0	1	0	1 0 1 0		1	1	1	1	0	rs	0	1	0	1	0	rd	d							
	XOR.L @(d:32,Rs,B),ERd				0	1	0	1 0 1 0		1	1	0	1	1	rs	0	1	0	1	0	rd	d	d						
	XOR.L @(d:32,Rs,W),ERd				0	1	0	1 0 1 0		1	1	1	0	1	rs	0	1	0	1	0	rd	d	d						
	XOR.L @(d:32,ERs,L),ERd				0	1	0	1 0 1 0		1	1	1	1	1	rs	0	1	0	1	0	rd	d	d						
	XOR.L @aa:16,ERd				0	1	0	1 0 1 0		0	1	0	0	0	0	0	0	0	1	0	1	0	rd	a					
	XOR.L @aa:32,ERd				0	1	0	1 0 1 0		0	1	0	0	1	0	0	0	0	0	1	0	1	0	rd	a	a			
	XOR.L @ERs, @ERd	0		1		0		0 1 0 0	6	9	0		rs	1	1	0	0	0	0	0	0	0	0	0	0				
XOR.L @ERs, @ERd+	0		1		0		0 1 0 0	6	9	0		rs	1	1	0	1	0	0	0	0	0	0	0	0					
XOR.L @ERs, @ERd-	0		1		0		0 1 0 0	6	9	0		rs	1	1	0	0	1	0	0	0	0	0	0	0					
XOR.L @ERs, @+ERd	0		1		0		0 1 0 0	6	9	0		rs	1	1	0	0	1	0	0	0	0	0	0	0					
XOR.L @ERs, @-ERd	0		1		0		0 1 0 0	6	9	0		rs	1	1	0	0	1	1	0	0	0	0	0	0					
XOR.L @ERs, @(d:2,ERd)	0		1		0		0 1 0 0	6	9	0		rs	1	1	0	0	0	d	d	0	rd	0	1	0	0	0	0		
XOR.L @ERs, @(d:16,ERd)	0		1		0		0 1 0 0	6	9	0		rs	1	1	0	0	1	1	0	0	rd	0	1	0	1	0	0	0	d

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension			
		15	8	7	0	15	8	7	0		15	8	7	0				
XOR	XOR.L @ERs, @(d:32,ERd)	0	1	0	0 1 0 0	6	9	0	rs	1 1 0 0		1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d	d
	XOR.L @ERs, @(d:16,Rd.B)	0	1	0	0 1 0 0	6	9	0	rs	1 1 0 0		1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d	
	XOR.L @ERs, @(d:16,Rd.W)	0	1	0	0 1 0 0	6	9	0	rs	1 1 0 0		1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d	
	XOR.L @ERs, @(d:16,ERd.L)	0	1	0	0 1 0 0	6	9	0	rs	1 1 0 0		1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d	
	XOR.L @ERs, @(d:32,Rd.B)	0	1	0	0 1 0 0	6	9	0	rs	1 1 0 0		1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d	d
	XOR.L @ERs, @(d:32,Rd.W)	0	1	0	0 1 0 0	6	9	0	rs	1 1 0 0		1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d	d
	XOR.L @ERs, @(d:32,ERd.L)	0	1	0	0 1 0 0	6	9	0	rs	1 1 0 0		1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d	d
	XOR.L @ERs, @aa:16	0	1	0	0 1 0 0	6	9	0	rs	1 1 0 0		0 1 0 0	0	0 0 0 0	0 1 0 1	0 0 0 0	a	
	XOR.L @ERs, @aa:32	0	1	0	0 1 0 0	6	9	0	rs	1 1 0 0		0 1 0 0	1	0 0 0 0	0 1 0 1	0 0 0 0	a	a
	XOR.L @ERs, @ERd	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0		0 0 0 0	0	rd	0 1 0 1	0 0 0 0		
	XOR.L @ERs+, @ERd+	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0		1 0 0 0	0	rd	0 1 0 1	0 0 0 0		
	XOR.L @ERs+, @ERd-	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0		1 0 1 0	0	rd	0 1 0 1	0 0 0 0		
	XOR.L @ERs+, @+ERd	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0		1 0 0 1	0	rd	0 1 0 1	0 0 0 0		
	XOR.L @ERs+, @-ERd	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0		1 0 1 1	0	rd	0 1 0 1	0 0 0 0		
	XOR.L @ERs+, @(d:2,ERd)	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0		0 0 d d	0	rd	0 1 0 1	0 0 0 0		
	XOR.L @ERs+, @(d:16,ERd)	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0		1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d	
	XOR.L @ERs+, @(d:32,ERd)	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0		1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d	d
	XOR.L @ERs+, @(d:16,Rd.B)	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0		1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d	
	XOR.L @ERs+, @(d:16,Rd.W)	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0		1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d	
	XOR.L @ERs+, @(d:16,ERd.L)	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0		1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d	
	XOR.L @ERs+, @(d:32,Rd.B)	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0		1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d	d
	XOR.L @ERs+, @(d:32,Rd.W)	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0		1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d	d
	XOR.L @ERs+, @(d:32,ERd.L)	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0		1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d	d
	XOR.L @ERs+, @aa:16	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0		0 1 0 0	0	0 0 0 0	0 1 0 1	0 0 0 0	a	
	XOR.L @ERs+, @aa:32	0	1	0	0 1 0 0	6	D	0	rs	1 1 0 0		0 1 0 0	1	0 0 0 0	0 1 0 1	0 0 0 0	a	a
	XOR.L @ERs-, @ERd	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		0 0 0 0	0	rd	0 1 0 1	0 0 0 0		
	XOR.L @ERs-, @ERd+	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 0 0 0	0	rd	0 1 0 1	0 0 0 0		
	XOR.L @ERs-, @ERd-	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 0 1 0	0	rd	0 1 0 1	0 0 0 0		
	XOR.L @ERs-, @+ERd	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 0 0 1	0	rd	0 1 0 1	0 0 0 0		
	XOR.L @ERs-, @-ERd	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 0 1 1	0	rd	0 1 0 1	0 0 0 0		
	XOR.L @ERs-, @(d:2,ERd)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		0 0 d d	0	rd	0 1 0 1	0 0 0 0		
	XOR.L @ERs-, @(d:16,ERd)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d	
XOR.L @ERs-, @(d:32,ERd)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d	d	
XOR.L @ERs-, @(d:16,Rd.B)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d		



Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
XOR	XOR.L @ERs, @(d:16,Rd,W)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.L @ERs, @(d:16,ERd,L)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.L @ERs, @(d:32,Rd,B)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @ERs, @(d:32,Rd,W)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @ERs, @(d:32,ERd,L)	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @ERs, @aa:16	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		0 1 0 0	0 0 0 0	0 1 0 1	0 0 0 0	a	
	XOR.L @ERs, @aa:32	0	1	0	0 1 1 0	6	D	0	rs	1 1 0 0		0 1 0 0	1 0 0 0	0 1 0 1	0 0 0 0	a a	
	XOR.L @+ERs, @ERd	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		0 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @+ERs, @ERd+	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @+ERs, @ERd-	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 0 1 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @+ERs, @+ERd	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 0 0 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @+ERs, @-ERd	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 0 1 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @+ERs, @(d:2,ERd)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		0 0 d d	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @+ERs, @(d:16,ERd)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.L @+ERs, @(d:32,ERd)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @+ERs, @(d:16,Rd,B)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.L @+ERs, @(d:16,Rd,W)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.L @+ERs, @(d:16,ERd,L)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.L @+ERs, @(d:32,Rd,B)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @+ERs, @(d:32,Rd,W)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @+ERs, @(d:32,ERd,L)	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @+ERs, @aa:16	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		0 1 0 0	0 0 0 0	0 1 0 1	0 0 0 0	a	
	XOR.L @+ERs, @aa:32	0	1	0	0 1 0 1	6	D	0	rs	1 1 0 0		0 1 0 0	1 0 0 0	0 1 0 1	0 0 0 0	a a	
	XOR.L @-ERs, @ERd	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		0 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @-ERs, @ERd+	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @-ERs, @ERd-	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 1 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @-ERs, @+ERd	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 0 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @-ERs, @-ERd	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 0 1 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @-ERs, @(d:2,ERd)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		0 0 d d	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @-ERs, @(d:16,ERd)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.L @-ERs, @(d:32,ERd)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @-ERs, @(d:16,Rd,B)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d
XOR.L @-ERs, @(d:16,Rd,W)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d	
XOR.L @-ERs, @(d:16,ERd,L)	0	1	0	0 1 1 1	6	D	0	rs	1 1 0 0		1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d	

Instruction	Mnemonic	Opcode						EA Extension	Opcode						EA Extension																					
		15	8	7	0	15	8		7	0	15	8	7	0																						
XOR	XOR.L @-ERs, @(d:32,Rd,B)	0	1	0	0	1	1	1	6	D	0	rs	1	1	0	0	1	1	0	1	1	0	1	rd	0	1	0	1	0	0	0	0	0	d	d	
	XOR.L @-ERs, @(d:32,Rd,W)	0	1	0	0	1	1	1	6	D	0	rs	1	1	0	0	1	1	1	0	1	1	0	1	rd	0	1	0	1	0	0	0	0	0	d	d
	XOR.L @-ERs, @(d:32,ERd,L)	0	1	0	0	1	1	1	6	D	0	rs	1	1	0	0	1	1	1	1	1	0	0	rd	0	1	0	1	0	0	0	0	0	d	d	
	XOR.L @-ERs, @aa:16	0	1	0	0	1	1	1	6	D	0	rs	1	1	0	0	0	1	0	0	0	0	0	rd	0	1	0	1	0	0	0	0	0	a		
	XOR.L @-ERs, @aa:32	0	1	0	0	1	1	1	6	D	0	rs	1	1	0	0	0	1	0	0	1	0	0	0	rd	0	1	0	1	0	0	0	0	a	a	
	XOR.L @(d:2,ERs), @+ERd	0	1	0	0	1	d	d	6	9	0	rs	1	1	0	0	0	0	0	0	rd	0	1	0	1	0	0	0	0	0	0	0	0			
	XOR.L @(d:2,ERs), @ERd+	0	1	0	0	1	d	d	6	9	0	rs	1	1	0	0	1	0	0	0	rd	0	1	0	1	0	0	0	0	0	0	0				
	XOR.L @(d:2,ERs), @ERd-	0	1	0	0	1	d	d	6	9	0	rs	1	1	0	0	1	0	1	0	rd	0	1	0	1	0	0	0	0	0	0	0				
	XOR.L @(d:2,ERs), @+ERd	0	1	0	0	1	d	d	6	9	0	rs	1	1	0	0	1	0	1	0	rd	0	1	0	1	0	0	0	0	0	0	0				
	XOR.L @(d:2,ERs), @-ERd	0	1	0	0	1	d	d	6	9	0	rs	1	1	0	0	1	0	1	1	rd	0	1	0	1	0	0	0	0	0	0	0				
	XOR.L @(d:2,ERs), @(d:2,ERd)	0	1	0	0	1	d	d	6	9	0	rs	1	1	0	0	0	0	d	d	rd	0	1	0	1	0	0	0	0	0	0					
	XOR.L @(d:2,ERs), @(d:16,ERd)	0	1	0	0	1	d	d	6	9	0	rs	1	1	0	0	1	1	0	0	rd	0	1	0	1	0	0	0	0	0	0	d				
	XOR.L @(d:2,ERs), @(d:32,ERd)	0	1	0	0	1	d	d	6	9	0	rs	1	1	0	0	1	1	0	0	rd	0	1	0	1	0	0	0	0	0	d	d				
	XOR.L @(d:2,ERs), @(d:16,Rd,B)	0	1	0	0	1	d	d	6	9	0	rs	1	1	0	0	1	1	0	1	rd	0	1	0	1	0	0	0	0	0	d					
	XOR.L @(d:2,ERs), @(d:16,Rd,W)	0	1	0	0	1	d	d	6	9	0	rs	1	1	0	0	1	1	1	0	rd	0	1	0	1	0	0	0	0	0	d					
	XOR.L @(d:2,ERs), @(d:16,ERd,L)	0	1	0	0	1	d	d	6	9	0	rs	1	1	0	0	1	1	1	1	rd	0	1	0	1	0	0	0	0	d						
	XOR.L @(d:2,ERs), @(d:32,Rd,B)	0	1	0	0	1	d	d	6	9	0	rs	1	1	0	0	1	1	0	1	rd	0	1	0	1	0	0	0	0	d	d					
	XOR.L @(d:2,ERs), @(d:32,Rd,W)	0	1	0	0	1	d	d	6	9	0	rs	1	1	0	0	1	1	1	0	rd	0	1	0	1	0	0	0	0	d	d					
	XOR.L @(d:2,ERs), @(d:32,ERd,L)	0	1	0	0	1	d	d	6	9	0	rs	1	1	0	0	1	1	1	1	rd	0	1	0	1	0	0	0	0	d	d					
	XOR.L @(d:2,ERs), @aa:16	0	1	0	0	1	d	d	6	9	0	rs	1	1	0	0	0	1	0	0	rd	0	1	0	1	0	0	0	0	a						
	XOR.L @(d:2,ERs), @aa:32	0	1	0	0	1	d	d	6	9	0	rs	1	1	0	0	0	1	0	0	rd	0	1	0	1	0	0	0	0	a	a					
	XOR.L @(d:16,ERs), @ERd	0	1	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	0	0	0	rd	0	1	0	1	0	0	0	0							
	XOR.L @(d:16,ERs), @ERd+	0	1	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	0	0	rd	0	1	0	1	0	0	0	0							
	XOR.L @(d:16,ERs), @ERd-	0	1	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	0	1	rd	0	1	0	1	0	0	0	0							
	XOR.L @(d:16,ERs), @+ERd	0	1	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	0	0	rd	0	1	0	1	0	0	0	0							
	XOR.L @(d:16,ERs), @-ERd	0	1	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	0	1	rd	0	1	0	1	0	0	0	0							
	XOR.L @(d:16,ERs), @(d:2,ERd)	0	1	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	0	0	d	rd	0	1	0	1	0	0	0	0							
	XOR.L @(d:16,ERs), @(d:16,ERd)	0	1	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	1	0	rd	0	1	0	1	0	0	0	0	d						
	XOR.L @(d:16,ERs), @(d:32,ERd)	0	1	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	1	0	rd	0	1	0	1	0	0	0	0	d	d					
	XOR.L @(d:16,ERs), @(d:16,Rd,B)	0	1	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	1	0	rd	0	1	0	1	0	0	0	0	d						
	XOR.L @(d:16,ERs), @(d:16,Rd,W)	0	1	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	1	1	rd	0	1	0	1	0	0	0	0	d						
	XOR.L @(d:16,ERs), @(d:16,ERd,L)	0	1	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	1	1	rd	0	1	0	1	0	0	0	0	d						
XOR.L @(d:16,ERs), @(d:32,Rd,B)	0	1	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	1	0	rd	0	1	0	1	0	0	0	0	d	d						
XOR.L @(d:16,ERs), @(d:32,Rd,W)	0	1	0	0	1	0	0	6	F	0	rs	1	1	0	0	d	1	1	1	rd	0	1	0	1	0	0	0	0	d	d						



Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
XOR	XOR.L @(d:16,ERs),@(d:32,ERdL)	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @(d:16,ERs),@aa:16	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	0	0 0 0 0	0 1 0 1	0 0 0 0	a
	XOR.L @(d:16,ERs),@aa:32	0	1	0	0 1 0 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	1	0 0 0 0	0 1 0 1	0 0 0 0	a a
	XOR.L @(d:32,ERs),@ERd	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:32,ERs),@ERd+	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:32,ERs),@ERd-	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:32,ERs),@+ERd	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:32,ERs),@-ERd	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:32,ERs),@(d:2,ERd)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:32,ERs),@(d:16,ERd)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.L @(d:32,ERs),@(d:32,ERd)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @(d:32,ERs),@(d:16,Rd,B)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.L @(d:32,ERs),@(d:16,Rd,W)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.L @(d:32,ERs),@(d:16,ERdL)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.L @(d:32,ERs),@(d:32,Rd,B)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @(d:32,ERs),@(d:32,Rd,W)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @(d:32,ERs),@(d:32,ERdL)	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @(d:32,ERs),@aa:16	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 1 0 1	0 0 0 0	a
	XOR.L @(d:32,ERs),@aa:32	7	8	1	rs	0 1 0 0	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 1 0 1	0 0 0 0	a a
	XOR.L @(d:16,Rs,B),@ERd	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:16,Rs,B),@ERd+	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:16,Rs,B),@ERd-	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 1 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:16,Rs,B),@+ERd	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 0 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:16,Rs,B),@-ERd	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 0 1 1	0	rd	0 1 0 1	0 0 0 0	
XOR.L @(d:16,Rs,B),@(d:2,ERd)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 0 d d	0	rd	0 1 0 1	0 0 0 0		
XOR.L @(d:16,Rs,B),@(d:16,ERd)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d	
XOR.L @(d:16,Rs,B),@(d:32,ERd)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d d	
XOR.L @(d:16,Rs,B),@(d:16,Rd,B)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d	
XOR.L @(d:16,Rs,B),@(d:16,Rd,W)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d	
XOR.L @(d:16,Rs,B),@(d:16,ERdL)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d	
XOR.L @(d:16,Rs,B),@(d:32,Rd,B)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d	
XOR.L @(d:16,Rs,B),@(d:32,Rd,W)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d	
XOR.L @(d:16,Rs,B),@(d:32,ERdL)	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d	
XOR.L @(d:16,Rs,B),@aa:16	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	0	0 0 0 0	0 1 0 1	0 0 0 0	a	

Instruction	Mnemonic	Opcode				Opcode				EA Ex- tension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
XOR	XOR.L @(d:16,Rs.B),@aa:32	0	1	0	0 1 0 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 0 1	0 0 0 0	a a
	XOR.L @(d:16,Rs.W),@ERd	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 0 0 0	0 rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:16,Rs.W),@ERd+	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 0 0	0 rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:16,Rs.W),@ERd-	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 1 0	0 rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:16,Rs.W),@+ERd	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 0 1	0 rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:16,Rs.W),@-ERd	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 0 1 1	0 rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:16,Rs.W),@(d:2,ERd)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 0 d d	0 rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:16,Rs.W),@(d:16,ERd)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	0 rd	0 1 0 1	0 0 0 0	d
	XOR.L @(d:16,Rs.W),@(d:32,ERd)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 0	1 rd	0 1 0 1	0 0 0 0	d d
	XOR.L @(d:16,Rs.W),@(d:16,Rd.B)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	0 rd	0 1 0 1	0 0 0 0	d
	XOR.L @(d:16,Rs.W),@(d:16,Rd.W)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	0 rd	0 1 0 1	0 0 0 0	d
	XOR.L @(d:16,Rs.W),@(d:16,ERd.L)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	0 rd	0 1 0 1	0 0 0 0	d
	XOR.L @(d:16,Rs.W),@(d:32,Rd.B)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 0 1	1 rd	0 1 0 1	0 0 0 0	d d
	XOR.L @(d:16,Rs.W),@(d:32,Rd.W)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 0	1 rd	0 1 0 1	0 0 0 0	d d
	XOR.L @(d:16,Rs.W),@(d:32,ERd.L)	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	1 1 1 1	1 rd	0 1 0 1	0 0 0 0	d d
	XOR.L @(d:16,Rs.W),@aa:16	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 1 0 1	0 0 0 0	a
	XOR.L @(d:16,Rs.W),@aa:32	0	1	0	0 1 1 0	6	F	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 0 1	0 0 0 0	a a
	XOR.L @(d:16,ERs.L),@ERd	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 0 0 0	0 rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:16,ERs.L),@ERd+	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 0 0	0 rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:16,ERs.L),@ERd-	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 1 0	0 rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:16,ERs.L),@+ERd	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 0 1	0 rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:16,ERs.L),@-ERd	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 0 1 1	0 rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:16,ERs.L),@(d:2,ERd)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 0 d d	0 rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:16,ERs.L),@(d:16,ERd)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	0 rd	0 1 0 1	0 0 0 0	d
	XOR.L @(d:16,ERs.L),@(d:32,ERd)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 0	1 rd	0 1 0 1	0 0 0 0	d d
	XOR.L @(d:16,ERs.L),@(d:16,Rd.B)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	0 rd	0 1 0 1	0 0 0 0	d
	XOR.L @(d:16,ERs.L),@(d:16,Rd.W)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	0 rd	0 1 0 1	0 0 0 0	d
	XOR.L @(d:16,ERs.L),@(d:16,ERd.L)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	0 rd	0 1 0 1	0 0 0 0	d
	XOR.L @(d:16,ERs.L),@(d:32,Rd.B)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 0 1	1 rd	0 1 0 1	0 0 0 0	d d
	XOR.L @(d:16,ERs.L),@(d:32,Rd.W)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 0	1 rd	0 1 0 1	0 0 0 0	d d
	XOR.L @(d:16,ERs.L),@(d:32,ERd.L)	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	1 1 1 1	1 rd	0 1 0 1	0 0 0 0	d d
	XOR.L @(d:16,ERs.L),@aa:16	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	0 0 0 0	0 1 0 1	0 0 0 0	a
XOR.L @(d:16,ERs.L),@aa:32	0	1	0	0 1 1 1	6	F	0	rs	1 1 0 0	d	0 1 0 0	1 0 0 0	0 1 0 1	0 0 0 0	a a	
XOR.L @(d:32,Rs.B),@ERd	7	8	1	rs	0 1 0 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 0 0	0 rd	0 1 0 1	0 0 0 0		



Instruction	Mnemonic	Opcode						Opcode						EA Extension	Opcode						EA Extension																	
		15	8	7	0			15	8	7	0				15	8	7	0																				
XOR	XOR.L @(d:32,Rs.B),@ERd+	7	8	1	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	d	d	1	0	0	0	0	rd	0	1	0	1	0	0	0	0	0		
	XOR.L @(d:32,Rs.B),@ERd-	7	8	1	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	d	d	1	0	1	0	0	rd	0	1	0	1	0	0	0	0			
	XOR.L @(d:32,Rs.B),@+ERd	7	8	1	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	d	d	1	0	0	1	0	rd	0	1	0	1	0	0	0	0			
	XOR.L @(d:32,Rs.B),@-ERd	7	8	1	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	d	d	1	0	1	1	0	rd	0	1	0	1	0	0	0	0			
	XOR.L @(d:32,Rs.B),@(d:2,ERd)	7	8	1	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	d	d	0	0	0	d	0	rd	0	1	0	1	0	0	0	0			
	XOR.L @(d:32,Rs.B),@(d:16,ERd)	7	8	1	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	0	0	rd	0	1	0	1	0	0	0	0	d		
	XOR.L @(d:32,Rs.B),@(d:32,ERd)	7	8	1	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	0	1	rd	0	1	0	1	0	0	0	0	d	d	
	XOR.L @(d:32,Rs.B),@(d:16,Rd,B)	7	8	1	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	1	0	rd	0	1	0	1	0	0	0	0	d		
	XOR.L @(d:32,Rs.B),@(d:16,Rd,W)	7	8	1	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	0	0	rd	0	1	0	1	0	0	0	0	d		
	XOR.L @(d:32,Rs.B),@(d:16,ERd,L)	7	8	1	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	1	0	rd	0	1	0	1	0	0	0	0	d		
	XOR.L @(d:32,Rs.B),@(d:32,Rd,B)	7	8	1	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	1	1	rd	0	1	0	1	0	0	0	0	d	d	
	XOR.L @(d:32,Rs.B),@(d:32,Rd,W)	7	8	1	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	0	1	rd	0	1	0	1	0	0	0	0	d	d	
	XOR.L @(d:32,Rs.B),@(d:32,ERd,L)	7	8	1	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	1	1	rd	0	1	0	1	0	0	0	0	d	d	
	XOR.L @(d:32,Rs.B),@aa:16	7	8	1	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	d	d	0	1	0	0	0	0	rd	0	1	0	1	0	0	0	0	a	
	XOR.L @(d:32,Rs.B),@aa:32	7	8	1	rs	0	1	0	1	6	B	0	0	1	0	1	1	0	0	d	d	0	1	0	0	1	0	rd	0	1	0	1	0	0	0	0	a	a
	XOR.L @(d:32,Rs.W),@ERd	7	8	1	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	0	0	0	0	0	rd	0	1	0	1	0	0	0	0			
	XOR.L @(d:32,Rs.W),@ERd+	7	8	1	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	0	0	0	0	rd	0	1	0	1	0	0	0	0			
	XOR.L @(d:32,Rs.W),@ERd-	7	8	1	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	0	1	0	0	rd	0	1	0	1	0	0	0	0			
	XOR.L @(d:32,Rs.W),@+ERd	7	8	1	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	0	0	1	0	rd	0	1	0	1	0	0	0	0			
	XOR.L @(d:32,Rs.W),@-ERd	7	8	1	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	0	1	1	0	rd	0	1	0	1	0	0	0	0			
	XOR.L @(d:32,Rs.W),@(d:2,ERd)	7	8	1	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	0	0	d	d	0	rd	0	1	0	1	0	0	0	0			
	XOR.L @(d:32,Rs.W),@(d:16,ERd)	7	8	1	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	0	0	rd	0	1	0	1	0	0	0	0	d		
	XOR.L @(d:32,Rs.W),@(d:32,ERd)	7	8	1	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	0	1	rd	0	1	0	1	0	0	0	0	d	d	
	XOR.L @(d:32,Rs.W),@(d:16,Rd,B)	7	8	1	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	1	0	rd	0	1	0	1	0	0	0	0	d		
	XOR.L @(d:32,Rs.W),@(d:16,Rd,W)	7	8	1	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	0	0	rd	0	1	0	1	0	0	0	0	d		
	XOR.L @(d:32,Rs.W),@(d:16,ERd,L)	7	8	1	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	1	0	rd	0	1	0	1	0	0	0	0	d		
	XOR.L @(d:32,Rs.W),@(d:32,Rd,B)	7	8	1	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	0	1	1	rd	0	1	0	1	0	0	0	0	d	d	
	XOR.L @(d:32,Rs.W),@(d:32,Rd,W)	7	8	1	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	0	1	rd	0	1	0	1	0	0	0	0	d	d	
	XOR.L @(d:32,Rs.W),@(d:32,ERd,L)	7	8	1	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	1	1	1	1	1	rd	0	1	0	1	0	0	0	0	d	d	
	XOR.L @(d:32,Rs.W),@aa:16	7	8	1	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	0	1	0	0	0	0	rd	0	1	0	1	0	0	0	0	a	
XOR.L @(d:32,Rs.W),@aa:32	7	8	1	rs	0	1	1	0	6	B	0	0	1	0	1	1	0	0	d	d	0	1	0	0	1	0	rd	0	1	0	1	0	0	0	0	a	a	
XOR.L @(d:32,ERs.L),@ERd	7	8	1	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	0	0	0	0	0	rd	0	1	0	1	0	0	0	0				
XOR.L @(d:32,ERs.L),@ERd+	7	8	1	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	0	0	0	0	rd	0	1	0	1	0	0	0	0				
XOR.L @(d:32,ERs.L),@ERd-	7	8	1	rs	0	1	1	1	6	B	0	0	1	0	1	1	0	0	d	d	1	0	1	0	0	rd	0	1	0	1	0	0	0	0				

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension		
		15	8	7	0	15	8	7	0		15	8	7	0			
XOR	XOR.L @(d:32,ERs.L),@+ERd	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 0 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:32,ERs.L),@-ERd	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 0 1 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:32,ERs.L),@(d:2,ERd)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 0 d d	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @(d:32,ERs.L),@(d:16,ERd)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.L @(d:32,ERs.L),@(d:32,ERd)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @(d:32,ERs.L),@(d:16,Rd.B)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.L @(d:32,ERs.L),@(d:16,Rd.W)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.L @(d:32,ERs.L),@(d:16,ERd.L)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.L @(d:32,ERs.L),@(d:32,Rd.B)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @(d:32,ERs.L),@(d:32,Rd.W)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @(d:32,ERs.L),@(d:32,ERd.L)	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @(d:32,ERs.L),@aa:16	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	0	0 0 0 0	0 1 0 1	0 0 0 0	a
	XOR.L @(d:32,ERs.L),@aa:32	7	8	1	rs	0 1 1 1	6	B	0 0 1 0	1 1 0 0	d d	0 1 0 0	1	0 0 0 0	0 1 0 1	0 0 0 0	a a
	XOR.L @aa:16,@ERd	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @aa:16,@ERd+	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @aa:16,@ERd-	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 1 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @aa:16,@+ERd	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 0 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @aa:16,@-ERd	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 0 1 1	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @aa:16,@(d:2,ERd)	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 0 d d	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @aa:16,@(d:16,ERd)	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.L @aa:16,@(d:32,ERd)	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @aa:16,@(d:16,Rd.B)	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.L @aa:16,@(d:16,Rd.W)	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.L @aa:16,@(d:16,ERd.L)	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.L @aa:16,@(d:32,Rd.B)	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @aa:16,@(d:32,Rd.W)	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @aa:16,@(d:32,ERd.L)	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @aa:16,@aa:16	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 1 0 0	0	0 0 0 0	0 1 0 1	0 0 0 0	a
	XOR.L @aa:16,@aa:32	0	1		0	0 1 0 0	6	B	0 0 0 0	1 1 0 0	a	0 1 0 0	1	0 0 0 0	0 1 0 1	0 0 0 0	a a
	XOR.L @aa:32,@ERd	0	1		0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @aa:32,@ERd+	0	1		0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 0 0	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @aa:32,@ERd-	0	1		0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 1 0	0	rd	0 1 0 1	0 0 0 0	
XOR.L @aa:32,@+ERd	0	1		0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 0 1	0	rd	0 1 0 1	0 0 0 0		
XOR.L @aa:32,@-ERd	0	1		0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 0 1 1	0	rd	0 1 0 1	0 0 0 0		

Instruction	Mnemonic	Opcode				Opcode				EA Extension	Opcode				EA Extension	
		15	8	7	0	15	8	7	0		15	8	7	0		
XOR	XOR.L @aa:32,@(d:2,ERd)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 0 d d	0	rd	0 1 0 1	0 0 0 0	
	XOR.L @aa:32,@(d:16,ERd)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.L @aa:32,@(d:32,ERd)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @aa:32,@(d:16,Rd,B)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.L @aa:32,@(d:16,Rd,W)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 0	0	rd	0 1 0 1	0 0 0 0	d
	XOR.L @aa:32,@(d:16,ERd,L)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 1	0	rd	0 1 0 1	0 0 0 0	d
	XOR.L @aa:32,@(d:32,Rd,B)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 0 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @aa:32,@(d:32,Rd,W)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 0	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @aa:32,@(d:32,ERd,L)	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	1 1 1 1	1	rd	0 1 0 1	0 0 0 0	d d
	XOR.L @aa:32,@aa:16	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 1 0 0	0	0 0 0 0	0 1 0 1	0 0 0 0	a
XOR.L @aa:32,@aa:32	0	1	0	0 1 0 0	6	B	0 0 1 0	1 1 0 0	a a	0 1 0 0	1	0 0 0 0	0 1 0 1	0 0 0 0	a a	
XORC	XORC #xx:8,CCR											0	5	x x x x	x x x x	
	XORC #xx:8,EXR											0	5	x x x x	x x x x	

Notes: The unit of the EA extension is word (16 bits).

Register specification is as follows:

* The instruction codes for the H8S/2600 and H8S/2000 CPUs are also executable.

**Address Register
32-Bit Index Register
32-Bit Register**

**16-Bit Index Register
16-Bit Register**

**8-Bit Index Register
8-Bit Register**

Register Field	General Register	Register Field	General Register	Register Field	General Register
000	ER0	0000	R0	0000	R0H
001	ER1	0001	R1	0001	R1H
:	:	:	:	:	:
111	ER7	0111	R7	0111	R7H
		1000	E0	1000	R0L
		1001	E1	1001	R1L
		:	:	:	:
		1111	E7	1111	R7L

2.5 Condition Code Modification

Table 2.3 shows the effect of each CPU instruction on the condition code. The notation used in the table is below.

$m = 31$: For longword operands
15	: For word operands
7	: For byte operands
S_i	: The i -th bit in the source operand
D_i	: The i -th bit in the destination operand
R_i	: The i -th bit in the result
D_n	: The specified bit in the destination operand
—	: Not affected
\updownarrow	: Modified according to the result of the instruction (see definition)
0	: Always cleared to 0
1	: Always set to 1
*	: Undetermined (no guaranteed value)
Z'	: Z flag before instruction execution
C'	: C flag before instruction execution

Table 2.3 Condition Code Modification

Instruction	H	N	Z	V	C	Definition
ADD	\updownarrow	\updownarrow	\updownarrow	\updownarrow	\updownarrow	$H = S_{m-4} \cdot D_{m-4} + D_{m-4} \cdot \overline{R_{m-4}} + S_{m-4} \cdot \overline{R_{m-4}}$ $N = R_m$ $Z = \overline{R_m} \cdot \overline{R_{m-1}} \cdot \dots \cdot \overline{R_0}$ $V = S_m \cdot D_m \cdot \overline{R_m} + \overline{S_m} \cdot \overline{D_m} \cdot R_m$ $C = S_m \cdot D_m + D_m \cdot \overline{R_m} + S_m \cdot \overline{R_m}$
ADDS	—	—	—	—	—	
ADDX	\updownarrow	\updownarrow	\updownarrow	\updownarrow	\updownarrow	$H = S_{m-4} \cdot D_{m-4} + D_{m-4} \cdot \overline{R_{m-4}} + S_{m-4} \cdot \overline{R_{m-4}}$ $N = R_m$ $Z = Z' \cdot \overline{R_m} \cdot \dots \cdot \overline{R_0}$ $V = S_m \cdot D_m \cdot \overline{R_m} + \overline{S_m} \cdot \overline{D_m} \cdot R_m$ $C = S_m \cdot D_m + D_m \cdot \overline{R_m} + S_m \cdot \overline{R_m}$
AND	—	\updownarrow	\updownarrow	0	—	$N = R_m$ $Z = \overline{R_m} \cdot \overline{R_{m-1}} \cdot \dots \cdot \overline{R_0}$

Instruction	H	N	Z	V	C	Definition
ANDC	↑	↑	↑	↑	↑	Stores the corresponding bits of the result. No flags change when the operand is EXR.
BAND	—	—	—	—	↓	$C = C' \cdot Dn$
Bcc	—	—	—	—	—	
BCLR	—	—	—	—	—	
BCLR/EQ	—	—	—	—	—	
BCLR/NE	—	—	—	—	—	
BFLD	—	—	—	—	—	
BEST	—	—	—	—	—	
BIAND	—	—	—	—	↓	$C = C' \cdot \overline{Dn}$
BILD	—	—	—	—	↓	$C = \overline{Dn}$
BIOR	—	—	—	—	↓	$C = C' + \overline{Dn}$
BIST	—	—	—	—	—	
BISTZ	—	—	—	—	—	
BIXOR	—	—	—	—	↓	$C = C' \cdot Dn + \overline{C'} \cdot \overline{Dn}$
BLD	—	—	—	—	↓	$C = Dn$
BNOT	—	—	—	—	—	
BOR	—	—	—	—	↓	$C = C' + Dn$
BRA	—	—	—	—	—	
BRA/S	—	—	—	—	—	
BRA/BC	—	—	—	—	—	
BRA/BS	—	—	—	—	—	
BSET	—	—	—	—	—	
BSET/EQ	—	—	—	—	—	
BSET/NE	—	—	—	—	—	
BSR	—	—	—	—	—	
BSR/BC	—	—	—	—	—	
BSR/BS	—	—	—	—	—	
BST	—	—	—	—	—	
BSTZ	—	—	—	—	—	
BTST	—	—	↑	—	—	$Z = \overline{Dn}$
BXOR	—	—	—	—	↓	$C = C' \cdot \overline{Dn} + \overline{C'} \cdot Dn$

Instruction	H	N	Z	V	C	Definition
CLRMAC*	—	—	—	—	—	
CMP	↓	↓	↓	↓	↓	$H = \overline{S_{m-4}} \cdot \overline{D_{m-4}} + \overline{D_{m-4}} \cdot R_{m-4} + S_{m-4} \cdot R_{m-4}$ $N = R_m$ $Z = \overline{R_m} \cdot \overline{R_{m-1}} \cdot \dots \cdot \overline{R_0}$ $V = \overline{S_m} \cdot D_m \cdot \overline{R_m} + S_m \cdot \overline{D_m} \cdot R_m$ $C = S_m \cdot \overline{D_m} + \overline{D_m} \cdot R_m + S_m \cdot R_m$
DAA	*	↓	↓	*	↓	$N = R_m$ $Z = \overline{R_m} \cdot \overline{R_{m-1}} \cdot \dots \cdot \overline{R_0}$ <p>C: Decimal addition carry</p>
DAS	*	↓	↓	*	—	$N = R_m$ $Z = \overline{R_m} \cdot \overline{R_{m-1}} \cdot \dots \cdot \overline{R_0}$ <p>C: Decimal subtraction borrow</p>
DEC	—	↓	↓	↓	—	$N = R_m$ $Z = \overline{R_m} \cdot \overline{R_{m-1}} \cdot \dots \cdot \overline{R_0}$ $V = D_m \cdot \overline{R_m}$
DIVS	—	↓	↓	—	—	$N = S_m \cdot \overline{D_m} + \overline{S_m} \cdot D_m$ $Z = \overline{S_m} \cdot \overline{S_{m-1}} \cdot \dots \cdot \overline{S_0}$
DIVU	—	↓	↓	—	—	$N = S_m$ $Z = \overline{S_m} \cdot \overline{S_{m-1}} \cdot \dots \cdot \overline{S_0}$
DIVXS	—	↓	↓	—	—	$N = S_m \cdot \overline{D_m} + \overline{S_m} \cdot D_m$ $Z = \overline{S_m} \cdot \overline{S_{m-1}} \cdot \dots \cdot \overline{S_0}$
DIVXU	—	↓	↓	—	—	$N = S_m$ $Z = \overline{S_m} \cdot \overline{S_{m-1}} \cdot \dots \cdot \overline{S_0}$
EEPMOV	—	—	—	—	—	
EXTS	—	↓	↓	0	—	$N = R_m$ $Z = \overline{R_m} \cdot \overline{R_{m-1}} \cdot \dots \cdot \overline{R_0}$
EXTU	—	0	↓	0	—	$Z = \overline{R_m} \cdot \overline{R_{m-1}} \cdot \dots \cdot \overline{R_0}$
INC	—	↓	↓	↓	—	$N = R_m$ $Z = \overline{R_m} \cdot \overline{R_{m-1}} \cdot \dots \cdot \overline{R_0}$ $V = \overline{D_m} \cdot R_m$
JMP	—	—	—	—	—	
JSR	—	—	—	—	—	

Instruction	H	N	Z	V	C	Definition
LDC	↑	↑	↑	↑	↑	Stores the corresponding bits of the result. No flags change when the operand is EXR.
LDM	—	—	—	—	—	
LDMAC*	—	—	—	—	—	
MAC*	—	—	—	—	—	
MOV	—	↑	↑	0	—	$N = Rm$ $Z = \overline{Rm} \cdot \overline{Rm-1} \cdot \dots \cdot \overline{R0}$
MOVA	—	—	—	—	—	
MOVFPPE	—	↑	↑	0	—	$N = Rm$ $Z = \overline{Rm} \cdot \overline{Rm-1} \cdot \dots \cdot \overline{R0}$
MOVMD	—	—	—	—	—	
MOVSD	—	—	—	—	—	
MOVTPE	—	↑	↑	0	—	$N = Rm$ $Z = \overline{Rm} \cdot \overline{Rm-1} \cdot \dots \cdot \overline{R0}$
MULS	—	↑	↑	—	—	$N = Rm$ $Z = \overline{Rm} \cdot \overline{Rm-1} \cdot \dots \cdot \overline{R0}$
MULS/U*	—	↑	↑	—	—	$N = Rm$ $Z = \overline{Rm} \cdot \overline{Rm-1} \cdot \dots \cdot \overline{R0}$
MULU	—	—	—	—	—	
MULU/U*	—	—	—	—	—	
MULXS	—	↑	↑	—	—	$N = R2m$ $Z = \overline{R2m} \cdot \overline{R2m-1} \cdot \dots \cdot \overline{R0}$
MULXU	—	—	—	—	—	
NEG	↑	↑	↑	↑	↑	$H = Dm-4 + Rm-4$ $N = Rm$ $Z = \overline{Rm} \cdot \overline{Rm-1} \cdot \dots \cdot \overline{R0}$ $V = Dm \cdot Rm$ $C = Dm + Rm$
NOP	—	—	—	—	—	
NOT	—	↑	↑	0	—	$N = Rm$ $Z = \overline{Rm} \cdot \overline{Rm-1} \cdot \dots \cdot \overline{R0}$
OR	—	↑	↑	0	—	$N = Rm$ $Z = \overline{Rm} \cdot \overline{Rm-1} \cdot \dots \cdot \overline{R0}$

Instruction	H	N	Z	V	C	Definition
ORC	↑	↑	↑	↑	↑	Stores the corresponding bits of the result. No flags change when the operand is EXR.
POP	—	↑	↑	0	—	$N = Rm$ $Z = \overline{Rm} \cdot \overline{Rm-1} \cdot \dots \cdot \overline{R0}$
PUSH	—	↑	↑	0	—	$N = Rm$ $Z = \overline{Rm} \cdot \overline{Rm-1} \cdot \dots \cdot \overline{R0}$
ROTL	—	↑	↑	0	↑	$N = Rm$ $Z = \overline{Rm} \cdot \overline{Rm-1} \cdot \dots \cdot \overline{R0}$ $C = Dm$ (1-bit shift), $C = Dm-1$ (2-bit shift)
ROTR	—	↑	↑	0	↑	$N = Rm$ $Z = \overline{Rm} \cdot \overline{Rm-1} \cdot \dots \cdot \overline{R0}$ $C = D0$ (1-bit shift), $C = D1$ (2-bit shift)
ROTXL	—	↑	↑	0	↑	$N = Rm$ $Z = \overline{Rm} \cdot \overline{Rm-1} \cdot \dots \cdot \overline{R0}$ $C = Dm$ (1-bit shift), $C = Dm-1$ (2-bit shift)
ROTXR	—	↑	↑	0	↑	$N = Rm$ $Z = \overline{Rm} \cdot \overline{Rm-1} \cdot \dots \cdot \overline{R0}$ $C = D0$ (1-bit shift), $C = D1$ (2-bit shift)
RTE	↑	↑	↑	↑	↑	Stores the corresponding bits of the result.
RTE/L	↑	↑	↑	↑	↑	Stores the corresponding bits of the result.
RTS	—	—	—	—	—	
RTS/L	—	—	—	—	—	
SHAL	—	↑	↑	↑	↑	$N = Rm$ $Z = \overline{Rm} \cdot \overline{Rm-1} \cdot \dots \cdot \overline{R0}$ $V = \overline{Dm} \cdot \overline{Dm-1} + \overline{Dm} \cdot \overline{Dm-1}$ (1-bit shift) $V = \overline{Dm} \cdot \overline{Dm-1} \cdot \overline{Dm-2} + \overline{Dm} \cdot \overline{Dm-1} \cdot \overline{Dm-2}$ (2-bit shift) $C = Dm$ (1-bit shift), $C = Dm-1$ (2-bit shift)
SHAR	—	↑	↑	0	↑	$N = Rm$ $Z = \overline{Rm} \cdot \overline{Rm-1} \cdot \dots \cdot \overline{R0}$ $C = D0$ (1-bit shift), $C = D1$ (2-bit shift)
SHLL	—	↑	↑	0	↑	$N = Rm$ $Z = \overline{Rm} \cdot \overline{Rm-1} \cdot \dots \cdot \overline{R0}$ $C = Dm$ (1-bit shift), $C = Dm-n+1$ (n-bit shift)

Instruction	H	N	Z	V	C	Definition
SHLR	—	↓	↓	0	↓	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \dots \cdot \overline{R0}$ C = D0 (1-bit shift), C = Dn-1 (n-bit shift)
SLEEP	—	—	—	—	—	
STC	—	—	—	—	—	
STM	—	—	—	—	—	
STMAC*	—	↓	↓	↓	—	N = 1 if a MAC instruction resulted in a negative value in the MAC register Z = 1 if a MAC instruction resulted in a zero value in the MAC register V = 1 if a MAC instruction resulted in an overflow
SUB	↓	↓	↓	↓	↓	H = $S_{m-4} \cdot \overline{D_{m-4}} + \overline{D_{m-4}} \cdot R_{m-4} + S_{m-4} \cdot R_{m-4}$ N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \dots \cdot \overline{R0}$ V = $\overline{Sm} \cdot Dm \cdot \overline{Rm} + Sm \cdot \overline{Dm} \cdot Rm$ C = $Sm \cdot \overline{Dm} + \overline{Dm} \cdot Rm + Sm \cdot Rm$
SUBS	—	—	—	—	—	
SUBX	↓	↓	↓	↓	↓	H = $S_{m-4} \cdot \overline{D_{m-4}} + \overline{D_{m-4}} \cdot R_{m-4} + S_{m-4} \cdot R_{m-4}$ N = Rm Z = $Z' \cdot \overline{Rm} \cdot \dots \cdot \overline{R0}$ V = $\overline{Sm} \cdot Dm \cdot \overline{Rm} + Sm \cdot \overline{Dm} \cdot Rm$ C = $Sm \cdot \overline{Dm} + \overline{Dm} \cdot Rm + Sm \cdot Rm$
TAS	—	↓	↓	0	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \dots \cdot \overline{R0}$
TRAPA	—	—	—	—	—	
XOR	—	↓	↓	0	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \dots \cdot \overline{R0}$
XORC	↓	↓	↓	↓	↓	Stores the corresponding bits of the result. No flags change when the operand is EXR.

Note: * Available only when the multiplier is supported.

Section 3 Processing States

3.1 Overview

The CPU has five main processing states: the reset state, exception handling state, program execution state, bus-released state, and power-down state.

Figure 3.1 shows a diagram of the processing states. Figure 3.2 indicates the state transitions.

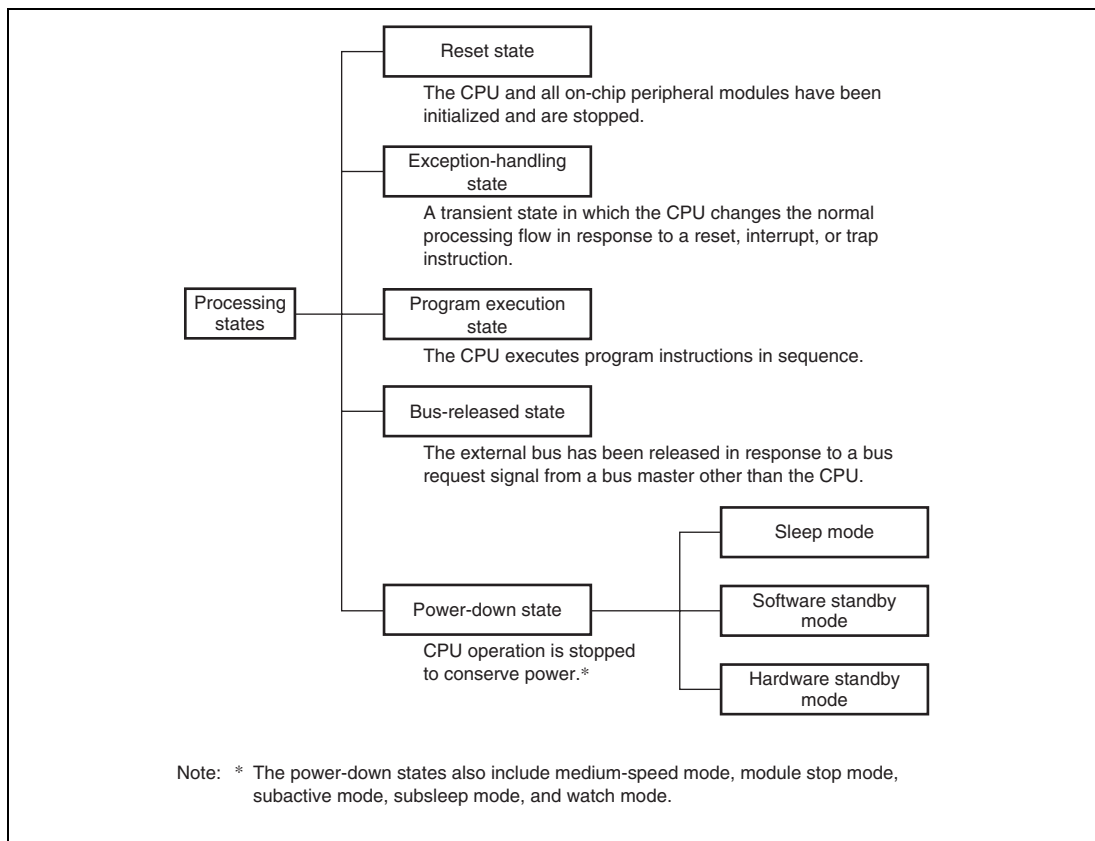


Figure 3.1 Processing States

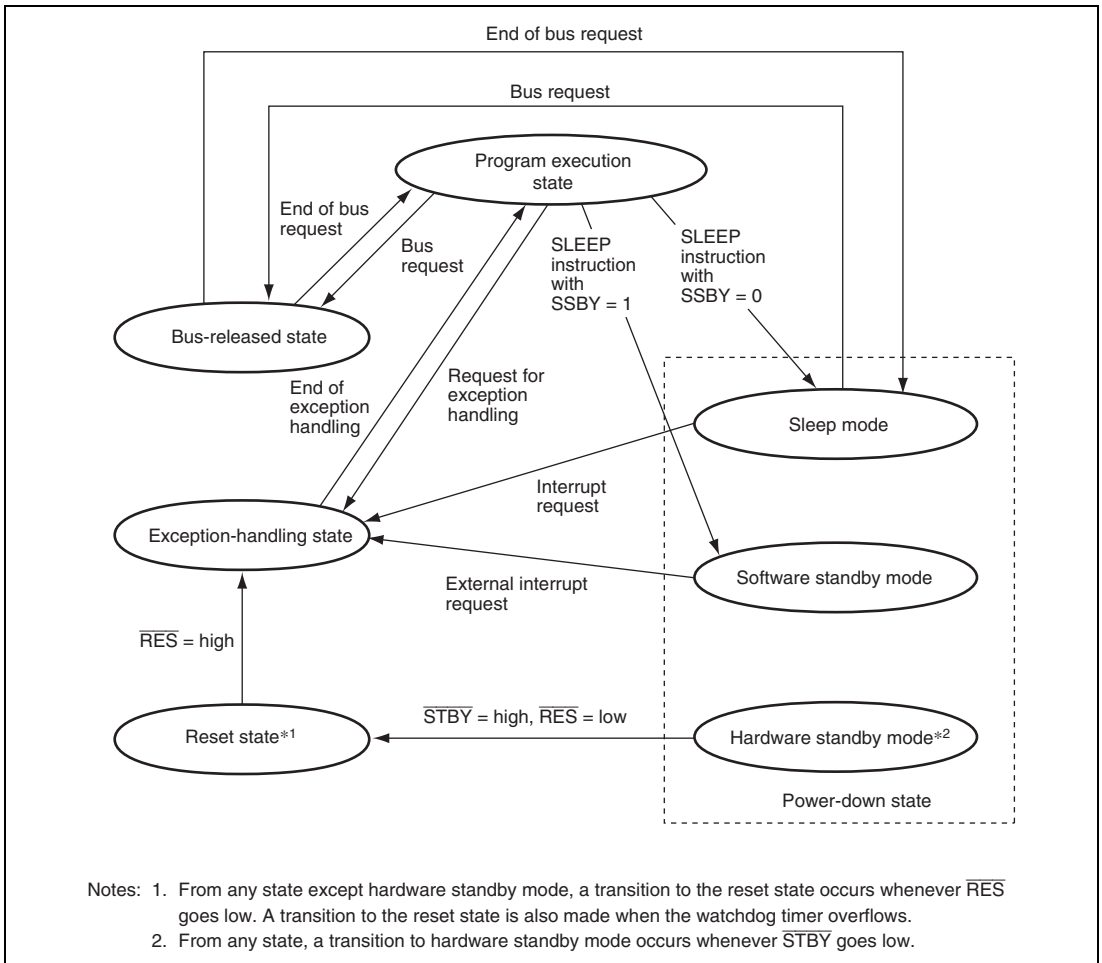


Figure 3.2 State Transitions

3.2 Reset State

When the \overline{RES} input goes low all current processing stops and the CPU enters the reset state. In the reset state, all interrupts are disabled.

Reset exception handling starts when the \overline{RES} pin changes from low to high.

The reset state can also be entered by a watchdog timer overflow. For details, see the hardware manual for the corresponding product.

3.3 Exception-Handling State


The exception-handling state is a transient state that occurs when the CPU alters the normal processing flow due to a reset, interrupt, or trap instruction. The CPU fetches a start address (vector) from the exception vector table and branches to that start address.

3.3.1 Types of Exception Handling and Their Priority

Exception handling is performed for traces, resets, interrupts, and trap instructions. Table 3.1 indicates the types of exception handling and their priority. Trap instruction exception handling is always accepted in the program execution state.

Exception handling and the stack structure differ according to the interrupt control mode set in SYSCR. For details, see the hardware manual for the corresponding product.

Table 3.1 Exception Handling Types and Priority

Priority	Type of Exception	Detection Timing	Start of Exception Handling
 High	Reset	Synchronized with clock	Exception handling starts when RES changes from low to high or a watchdog timer overflow occurs.
	Illegal instruction	When illegal instruction is detected	When an illegal instruction is detected, current execution stops and exception handling starts.
	Trace	End of instruction execution or end of exception-handling sequence	When the T bit is set to 1, the trace starts at the end of the current instruction or current exception-handling sequence.
	Interrupt	End of instruction execution or end of exception-handling sequence* ¹	When an interrupt is requested, exception handling starts at the end of the current instruction or current exception-handling sequence.
	Trap instruction	When TRAPA instruction is executed	Exception handling starts when a trap (TRAPA) instruction is executed* ² .
Low			

- Notes: 1. Interrupts are not detected at the end of the ANDC, ORC, XORC, and LDC instructions, or immediately after reset exception handling.
 2. Trap instruction exception handling is always accepted in the program execution state.

3.3.2 Reset Exception Handling

After the $\overline{\text{RES}}$ pin has gone low and the reset state has been entered, reset exception handling starts when the $\overline{\text{RES}}$ pin goes high again.

When reset exception handling starts, the CPU fetches a start address (vector) from the exception vector table and starts program execution from that start address. All interrupts, including NMI, are disabled during reset exception handling and after it ends.

3.3.3 Illegal Instruction

There are two types of illegal instructions: illegal general instructions and illegal slot instructions. These illegal instructions must not be executed.

When the CPU detects the execution of an undefined instruction, illegal general instruction exception handling starts. At this time, detection is not performed for fields that do not affect the instruction definition, such as the EA extension and register fields. If an illegal general instruction consists of multiple words, it may be detected during execution of the illegal instruction (2nd word or later). Therefore, the operation results for the execution of an undefined instruction are not guaranteed.

When an instruction which consists of multiple words or modifies the PC contents (branch, branch with bit condition, TRAPA, or RTE instruction) is executed as a delay slot instruction (placed immediately after a delayed branch instruction), illegal slot instruction exception handling starts.

When the exception handling for an illegal general instruction or an illegal slot instruction starts, the CPU refers to the SP contents (ER7) and pushes the PC and CCR contents onto the stack. Next, the CPU fetches the start address (vector) from the exception vector table and execution branches to that start address to execute the program. The PC contents to be saved vary depending on the execution state.

The contents of general registers including SP after the execution of an illegal general instruction or illegal slot instruction are not guaranteed. The RTE instruction should not be used when exiting the routine of the illegal instruction exception handling.

3.3.4 Trace

Trace mode is entered when the T bit in EXR is set to 1. When trace mode is established, trace exception handling starts at the end of each instruction.

At the end of a trace exception-handling sequence, the T bit of EXR is cleared to 0 and trace mode is cleared. The T bit saved on the stack retains its value of 1, and when the RTE instruction is executed to return from the trace exception-handling routine, trace mode is entered again. Trace exception-handling is not executed at the end of the RTE instruction.

3.3.5 Interrupt Exception Handling and Trap Instruction Exception Handling

When interrupt or trap-instruction exception handling begins, the CPU refers to the SP contents (ER7) and pushes the PC and CCR contents onto the stack. Next, the CPU specifies the I bit in CCR again. Then the CPU fetches a start address (vector) from the exception vector table and program execution starts from that start address.

Figure 3.3 shows the stack after exception handling ends.

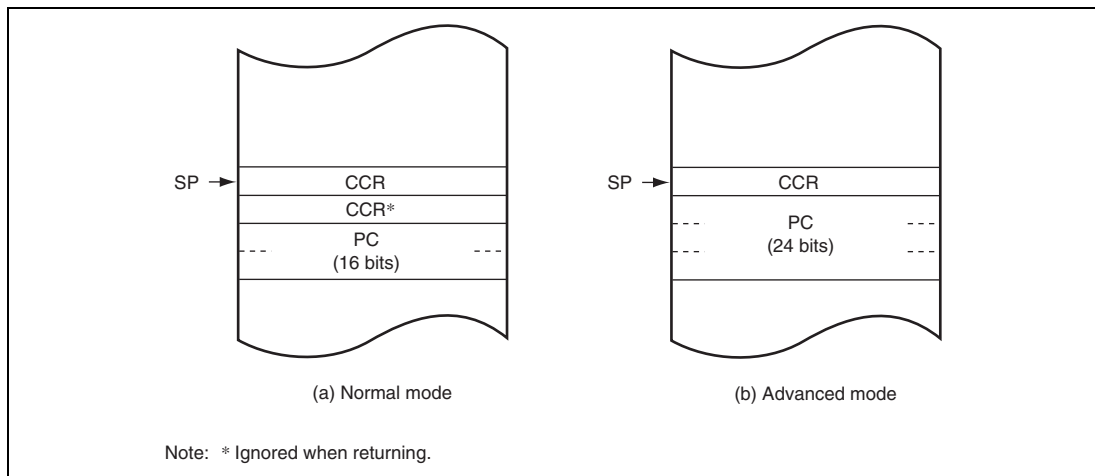


Figure 3.3 Stack Structure after Exception Handling (Example)

3.4 Program Execution State

In this state the CPU executes program instructions in sequence.

3.5 Bus-Released State

In this state, the bus has been released in response to a bus request from a bus master other than the CPU. While the bus is released, the CPU halts except for internal operations.

3.6 Power-Down State

The power-down state includes both modes in which the CPU stops operating and modes in which the CPU does not stop. There are three modes in which the CPU stops operating: sleep mode, software standby mode, and hardware standby mode.

There are also two other power-down modes: medium-speed mode and module stop mode. In medium-speed mode, the CPU operates on a medium-speed clock. Module stop mode permits halting of the operation of individual modules, other than the CPU. For details, see the hardware manual for the corresponding product.

3.6.1 Sleep Mode

A transition to sleep mode is made if the SLEEP instruction is executed while the SSBY bit in the standby control register (SBYCR) is cleared to 0. In sleep mode, CPU operations stop immediately after execution of the SLEEP instruction. The contents of CPU registers are retained.

3.6.2 Software Standby Mode

A transition to software standby mode is made if the SLEEP instruction is executed while the SSBY bit in SBYCR is set to 1. In software standby mode, the CPU, clock pulse generator, and all on-chip peripheral modules stop operating. The on-chip peripheral modules are reset, but as long as a specified voltage is supplied, the contents of CPU registers and on-chip RAM are retained. The I/O ports also remain in their existing states.

3.6.3 Hardware Standby Mode

A transition to hardware standby mode is made when the $\overline{\text{STBY}}$ pin goes low. In hardware standby mode, all functions enter the reset state and all operations stop. As long as a specified voltage is supplied, the on-chip RAM contents are retained.

Section 4 Basic Timing

The H8SX CPU is basically driven by a system clock, denoted by the symbol ϕ . The period from one rising edge of ϕ to the next is referred to as a "state." The memory cycle or bus cycle consists of one or two states. Different methods are used to access on-chip memory, on-chip peripheral modules, and the external address space. For details, see the hardware manual for the corresponding product.

4.1 On-Chip Memory (ROM, RAM) Access Timing

On-chip memory is accessed in one state. The data bus width is 32 bits, permitting byte, word, and longword accesses. Figure 4.1 shows the on-chip memory access cycle.

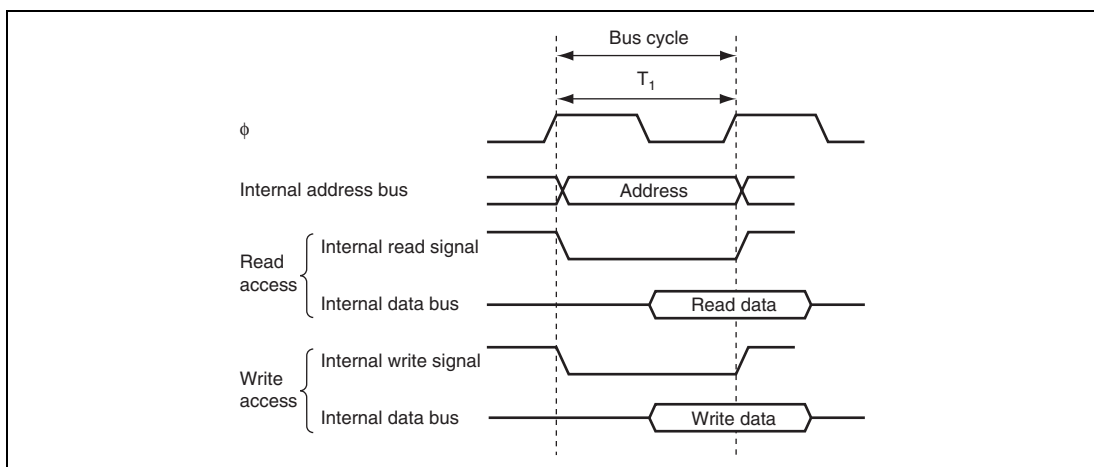


Figure 4.1 On-Chip Memory Access Cycle

4.2 On-Chip Peripheral Module Access Timing

The on-chip peripheral modules are accessed in two states. The data bus width is either 8 or 16 bits, depending on the particular on-chip register being accessed. Figure 4.2 shows the access timing for the on-chip peripheral modules.

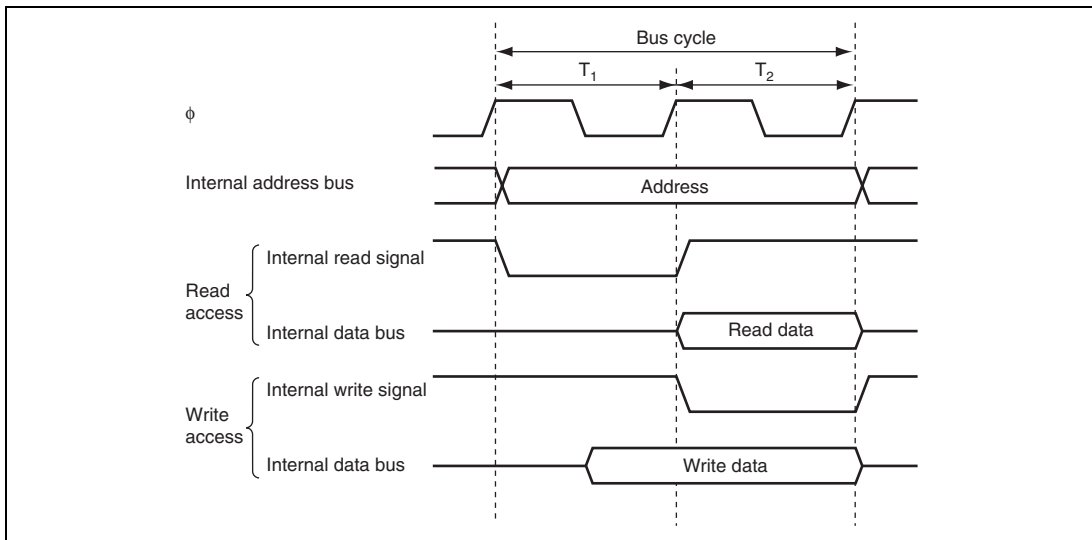


Figure 4.2 On-Chip Peripheral Module Access Timing

4.3 External Address Space Access Timing

The external address space is accessed in two or three states. The data bus width is 8, 16, or 32 bits. In a three-state access, wait states can be inserted.

4.4 Bus Arbitration Timing

The CPU releases the bus when it has been requested by other bus masters. The bus is released between bus cycles. In sleep mode, bus release is synchronized with a clock signal.

The bus should not be released under the following conditions:

- Between accesses which are divided into multiple accesses because of word or longword data access
- Between accesses which are divided into multiple accesses because of stack manipulation
- Between reading transfer data and writing it due to execution of a memory-memory transfer instruction, a block data transfer instruction, or a TAS instruction

Note however that the bus can be released between writing transfer data and reading the next transfer data due to execution of a block data transfer instruction.

- Between reading the target data and writing it due to execution of a bit manipulation instruction or operation instruction to memory.

Even in a case where the BSET/EQ or CMP instruction does not perform write, the period equivalent to the write state is regarded as a write access.

The number of states for execution reaches its maximum number without involving a transition of bus right when ADDX/SUBX.L #xx:32, <ea>. When the address for execution is odd and the instruction-fetching mode is 16 bits in this case, the bus is not released within a period of nine cycles.

Main Revisions and Additions in this Edition

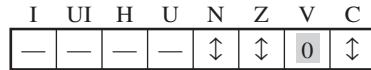
Item	Page	Revision (See Manual for Details)
Section 1. CPU	13	Deleted
1.5.2 Program Counter (PC)		The length of all CPU instructions is 16 bits (one word) or a multiple of 16 bits, so the least significant bit is ignored. (When the instruction code is fetched, the least significant bit is regarded as 0)
1.8.5 Register Indirect with Post-Increment, Pre-Decrement, Pre-Increment, or Post-Decrement —@ERn+, @-ERn, @+ERn, or @ERn-	40	Added Example 3: XOR.B R0L, @ER0+ When the value in ER0 before execution is H'00001234, location H'00001234 is read, the read data is EORed with H'35, and the result is written to location H'00001234. The value in ER0 after execution is H'00001235.
Section 2 Instruction Descriptions	76	Added
2.2.12 BFLD Bit Field Load		The immediate data that specifies a bit field is limited to consecutive bits with the value 1.
[8] Note		
2.2.13 BFST Bit Field STore	78	Added
[8] Note		The immediate data that specifies a bit field is limited to consecutive bits with the value 1.
2.2.25 BRA/BC BRAnch if Bit Cleared	100	Amended
[4] Description		...The displacement is a signed 8-bit or 16-bit value. The signed 8- or 16-bit displacement is capable of specifying branch destination addresses in the range from -128 to +126 bytes or from -32768 to +32766 bytes relative to the address that follows this instruction.
2.2.26 BRA/BS BRAnch if Bit Set	102	Amended
[4] Description		...The displacement is a signed 8-bit or 16-bit value. The signed 8- or 16-bit displacement is capable of specifying branch destination addresses in the range from -128 to +126 bytes or from -32768 to +32766 bytes relative to the address that follows this instruction.

Item				Page	Revision (See Manual for Details)								
2.2.39	DAA	Decimal Adjust	127	Amended									
Add					<table border="1"> <thead> <tr> <th>C Flag before Adjustment</th> <th>Upper 4 Bits before Adjustment</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0 to 2</td> </tr> <tr> <td>1</td> <td>0 to 2</td> </tr> <tr> <td>1</td> <td>0 to 3</td> </tr> </tbody> </table>	C Flag before Adjustment	Upper 4 Bits before Adjustment	1	0 to 2	1	0 to 2	1	0 to 3
C Flag before Adjustment	Upper 4 Bits before Adjustment												
1	0 to 2												
1	0 to 2												
1	0 to 3												
2.2.44	2.2.44	DIVS	DIVide	137	Added								
as Signed					The N flag is set to 1 when the signs of the dividend and divisor are different and cleared to 0 when they are the same. Accordingly, the N flag will be set to 1 in some cases where the quotient is 0.								
[8] Note													
2.2.45	DIVU	DIVide as	139	Added									
Unsigned					The N flag is set to 1 when the signs of the dividend and divisor are different and cleared to 0 when they are the same. Accordingly, the N flag will be set to 1 in some cases where the quotient is 0.								
[8] Note													
2.2.46	DIVXS	DIVide eXtend	140	Amended									
as Signed					This instruction divides the lower 16 or 32 bits of the contents of the destination register (Rd) by the source operand and stores the 16- or 32-bit result in the destination register....								
2.2.47	DIVXU	DIVide eXtend	142	Amended									
as Unsigned					This instruction divides the lower 16 or 32 bits of the contents of the destination register (Rd) by the source operand and stores the 16- or 32-bit result in the destination register....								
2.2.66	MOVA	MOVE effective	180	Amended									
Address					<ul style="list-style-type: none"> Example 								
[4] Description					<pre>MOVA/L @(H'FFFFFF00, R0L,B), ER1 : : : MOVA/L @(H'FFFFFF00, R1L,B), ER1</pre>								
[8] Notes			182		<ul style="list-style-type: none"> Example <pre>MOVAW @(H'FFFF8000, R1L,B), ER1</pre>								

Item **Page Revision (See Manual for Details)**

2.2.98 SHAR SHift Arithmetic Right 236 Amended

[5] Condition Code



- H: Previous value remains unchanged.
- N: Set to 1 if the result is negative; otherwise cleared to 0.
- Z: Set to 1 if the result is zero; otherwise cleared to 0.
- V: Always cleared to 0.
- C: Stores the last bit shifted out.

2.2.99 SHLL (1) SHift Logical Left 238 Amended

[1] Assembly-Language Format
 [4] Description
 [6] Available General Registers

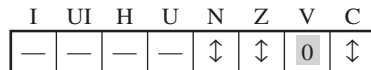
- [1] SHLL.Sz Rn, Rd
- [4] ...The number of bits to be shifted is specified by the immediate data or the contents of a general register Rn in which the lower 5 bits are valid. If a general register Rn is used for specifying the number of bits to be shifted, it should be specified as an 8-bit register.

239 [6]

Usage	General Register
Number of shifted bits	R0L to R7L, R0H to R7H
:	:

2.2.100 SHLL (2) SHift Logical Left 240 Amended

[5] Condition Code

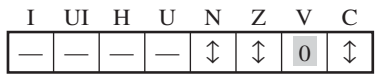


- H: Previous value remains unchanged.
- N: Set to 1 if the result is negative; otherwise cleared to 0.
- Z: Set to 1 if the result is zero; otherwise cleared to 0.
- V: Always cleared to 0.
- C: Stores the last bit shifted out.

Item		Page	Revision (See Manual for Details)
2.2.101 SHLR (1) Logical Right	Shift	242	Amended
[1] Assembly-Language Format			[1] SHLR.Sz Rn, Rd
[4] Description			[4] The number of bits to be shifted is specified by the immediate data or the contents of a general register Rn in which the lower 5 bits are valid. If a general register Rn is used for specifying the number of bits, it should be specified as an 8-bit register.
[6] Available General Registers			

Usage	General Register
Number of shifted bits	R0L to R7L, R0H to R7H

2.2.102 SHLR (2) Logical Right	Shift	244	Amended
[5] Condition Code			



- H: Previous value remains unchanged.
- N: Set to 1 if the result is negative; otherwise cleared to 0.
- Z: Set to 1 if the result is zero; otherwise cleared to 0.
- V: Always cleared to 0.
- C: Stores the last bit shifted out.

2.3 List of Instruction Set Arithmetic operations (69)	399	Amended
---	-----	---------

Instruction	Mnemonic	Size	Instruction Length	Min.	16-Bit Instruction Fetch	Number of Execution States ¹
CMP	CMP.L @-ERs, @(d:16,ERd)	L	4	6	7	7

Item

Page Revision (See Manual for Details)

Arithmetic operations (92)

422 Amended

Instruction	Mnemonic	Size	Instruction Length	Min. 16-Bit Instruction Fetch	Number of Execution States ¹⁾
SUB	SUB.W @ERs-,@(d:16,ERd)	W	4	6	7

Shift (1)

579 Added

Instruction	Mnemonic	...
SHLL	SHLL.B @ERd-	...

Branch (4)

360 Added

Instruction	Mnemonic	...
RTS/L	RTS/L (ERn-ERn+2)	...
	RTS/L (ERn-ERn+3)	...

Table 2.1 List of Instruction Set

634 Amended

Notes: 1. Values given in the table above are number of states for execution when the following conditions hold.

Number of states for execution can change according to the conditions of instruction execution.

In addition, the number of states may be reduced depending on the product.

Item**Page Revision (See Manual for Details)**

2.4	List of Instruction Codes	635 to 889	<ul style="list-style-type: none">• The BRA/BC, BRA/BS, BRA/S instructions have been moved to follow the BOR instruction. The MOV/MD and MOVSD instructions have been moved before the MOVTPC instruction.• Mnemonic DD.B @ERs+,Rd, ADD.B @ERs-,Rd, ADD.B @+ERs,Rd, ADD.B @-ERs,Rd, ADD.B @(d:2,ERs),Rd, ADD.B @(d:16,ERs),Rd, ADD.B @(d:32,ERs),Rd, ADD.B @(d:16,Rs.B),Rd, ADD.B @(d:16,Rs.W),Rd, ADD.B @(d:16,ERs.L),Rd, ADD.B @(d:32,Rs.B),Rd, ADD.B @(d:32,Rs.W),Rd, ADD.B @(d:32,ERs.L),Rd rs in the opcode (3 to 0) of the above mnemonic has been corrected to rd.• Mnemonic BAND #xx:3,@ERd, BIAND #xx:3,@ERd, BILD #xx:3,@ERd, BIOR #xx:3,@ERd, BIXOR #xx:3,@ERd, BLD #xx:3,@ERd, BOR #xx:3,@ERd, BTST #xx:3,@ERd, BTST Rn,@ERd, BXOR #xx:3,@ERd, STC.W CCR,@(d:32,ERd), STC.W EXR,@(d:32,ERd) rs in the opcode (6 to 4) of the above mnemonic has been corrected to rd.
-----	---------------------------	------------------	---

Item **Page** **Revision (See Manual for Details)**

2.4 List of Instruction Codes 751 • MOV.L #xx:16,@aa:16

EA Extension

a

=S

• MOVA/B.L@(d:16,@(d:16,ERs.L).B),ERd

EA Extension

d

d

2.5 Condition Code Modification 894 Amended

Instruction	H	N	Z	V	C
SHAR	—	↕	↕	0	↕

4.4 Bus Arbitration Timing 905 Added

The number of states for execution reaches its maximum number without involving a transition of bus right when ADDX/SUBX.L #xx:32, <ea>. When the address for execution is odd and the instruction-fetching mode is 16 bits in this case, the bus is not released within a period of nine cycles.

**Renesas 32-Bit CISC Microcomputer
Software Manual
H8SX Family**

Publication Date: 1st Edition, Feb 2003

Rev.4.00, Sep. 18, 2008

Published by: Sales Strategic Planning Div.
Renesas Technology Corp.

Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan



RENESAS SALES OFFICES

<http://www.renesas.com>

Refer to "<http://www.renesas.com/en/network>" for the latest and detailed information.

Renesas Technology America, Inc.

450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.

Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7858/7898

Renesas Technology Hong Kong Ltd.

7th Floor, North Tower, World Finance Centre, Harbour City, Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2377-3473

Renesas Technology Taiwan Co., Ltd.

10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 3518-3399

Renesas Technology Singapore Pte. Ltd.

1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.

Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd

Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510



H8SX Family Software Manual



Renesas Technology Corp.

2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan