AND9352/D

CMOS 16-BIT MICROCONTROLLER

LC88 SERIES CHAPTER 5 INSTRUCTIONS USER'S MANUAL

ON Semiconductor®

www.onsemi.com

APPLICATION NOTE

Microcontroller Business Unit
ON Semiconductor

December 2015 - Rev. 0

Publication Order Number :
AND9352/D

Contents

Contents

5. Instructions

5.1 Overview

The Xstormy16 instructions are classified as shown below. The supported instruction word lengths are word and its multiples. The number of possible operands is from 0 to 3.

- Xstormy16 Instruction Types

Type	Instruction	Operand	Operation
Data transfer	MOV, MOVF, MASK	2	op1←op2
PUSH, POP	PUSH, POP	1	SP⇔op1
SWAP	SWPN, SWPB, SWPW	1/2	op1⇔op1 or op1⇔op2
Logical operation	AND, OR, XOR, NOT	1/2	op1←f(op1, op2), op1←not(op1)
Arithmetic operation	ADD, ADC, SUB, SBC, INC, DEC	1/2	op1←f(op1, op2), op1←inc/dec(op1)
Logical shift	RRC, RLC, SHR, SHL	2	Shift/Rotate op1 by op2.
Arithmetic shift	ASR	2	Shift op1 by op2.
Bit manipulation	SET1, CLR1	2	Set/Clear bit op2 of op1.
Data conversion	CBW, REV	1	op1[15:8]←op1[7]
Conditional branch	BGE, BNC, BLT, BC, BGT, BHI, BLE, BLS, BPL, BNV, BMI, BV, BNZ, BZ	3	If test(op1-op2), then branch by op3.
Bit conditional branch	BN, BP	3	If test(bit op2 of op1), then branch by op3.
Flag conditional branch	BGE, BNC, BLT, BC, BGT, BHI, BLE, BLS, BPL, BNV, BMI, BV, BNZ, BZ	1	If flag, then branch by op1.
Unconditional branch	BR, JMP, JMPF	1/2	Branch by op1. Jump to op1(, 2).
Unconditional call	CALLR, CALL, CALLF, ICALLR, ICALL, ICALLF	1/2	Branch by op1. Jump to op1(, 2).
Return	IRET, RET	0	Return from subroutine call.
Multiplication/division	MUL, DIV, DIVLH, SDIV, SDIVLH	0	
System control	NOP, HALT, HOLD, HOLDX, BRK, RESET	0	Control system.

5.2 Addressing Modes

5.2.1 Overview

Xstormy16 addresses data on a 64KB bank basis. It can handle a maximum of 4GB of data (0 to FFFF_FFFFH). This 4GB data includes the programming ROM data.
The program counter (PC) can handle a linear 4GB addressing space (8000H to FFFF_FFFFH). However, only 16 MB space (8000H to 00FF_FFFFH) can be specified with absolute addresses.

5.2.2 Addressing (immediate)

	Addressing Mode	Description	Symbol
1	Immediate data	The data in the instruction code is the operand of the instruction.	#imm16, #imm8, #imm4, #imm2

5.2.3 Addressing (general-purpose register)

	Addressing Mode	Description	Symbol
2	Register direct (R0 to R15)	The general-purpose register designated by the data in the instruction code is the operand of the instruction.	Rd Rs
3	PSW register indirect (R0 to R15)	The general-purpose register designated by bits 15 to 12 of the PSW is the operand of the instruction.	Rx, RxL, RxH

5.2.4 Addressing (bit)

	Addressing Mode	Description	Symbol
4	Immediate data	The required bits are designated by the data in the instruction code.	#imm4
5	Register indirect	The required bits are designated by the contents of the general-purpose register specified by the data in the instruction code.	Rs

5.2.5 Addressing (shift amount)

	Addressing Mode	Description	Symbol
6	Immediate data	The shift or rotation amount is designated by the data in the instruction code.	#imm3, #imm4
7	Register indirect	The required shift or rotation amount is designated by the contents of the general-purpose register specified by the data in the instruction code.	Rs

5.2.6 Addressing (memory: 0 to 0FFFFH)

	Addressing Mode	Description	Symbol
8	SFR direct (7F00 to 7FFFH)	The result of adding 7F00H to the 8-bit data in the instruction code is regarded as an address and used to designate the operand (SFR) in memory.	m16
9	RAM direct (0000H to 00FFH)	The 8-bit data in the instruction code is regarded as an address and used to designate the operand in memory.	m16
10	Register indirect (0000 to FFFFH)	The contents of the general-purpose register specified by the data in the instruction code are regarded as an address*1 and used to designate the operand in memory.	(Rd) (Rs)
11	Post-increment register indirect (0000 to FFFFH)	The contents of the general-purpose register specified by the data in the instruction code are regarded as an address*1 and used to designate the operand in memory. Subsequently, the contents of this general-purpose register are incremented by 1 (byte access) or 2 (word access).	(Rd++) (Rs++)
12	Predecrement register indirect (0000 to FFFFH)	The contents of the general-purpose register specified by the data in the instruction code are decremented by 1 (byte access) or 2 (word access). This value is regarded as an address*1 and used to designate the operand in memory.	(--Rd) (--Rs)
13	Register indirect with offset (0000 to FFFFH)	The result*2 of adding the 12-bit singed offset data in the instruction code to the contents of the general-purpose register specified by the data in the instruction code is regarded as an address*1 and used to designate the operand in memory.	(Rd, ±n) (Rs, ±n)
14	Post-increment register indirect with offset (0000 to FFFFH)	The result*2 of adding the 12-bit singed offset data in the instruction code to the contents of the general-purpose register specified by the data in the instruction code is regarded as an address*1 and used to designate the operand in memory. Subsequently, the contents of this general-purpose register are incremented by 1 (byte access) or 2 (word access).	(Rd++, ±n) (Rs++, ±n)
15	Predecrement register indirect with offset (0000 to FFFFH)	The contents of the general-purpose register specified by the data in the instruction code are decremented by 1 (byte access) or 2 (word access). The result*2 of adding this value to the 12-bit signed offset data in the instruction code is regarded as an address*1 and used to designate the operand in memory.	(--Rd, ±n) (--Rs, ±n)

*1: When a word is accessed, the higher-order byte of the operand is designated if LSB of the address data is 1 and the lower-order byte if the LSB is 0.

*2: Any carry or borrow occurring as the result of the 16-bit arithmetic operation is ignored.

5.2.7 Addressing (memory: 0 to FFFF_FFFFH)

	Addressing Mode	Description	Symbol
16	Extended address register indirect (0000_0000 to FFFF_FFFFH)	The 32-bit address[*1] that contains the contents of the general-purpose register specified by the data in the instruction code in its lower-order 16 bit positions and the contents of the register R8 in its higher-order 16 bit positions is used to designate the operand in memory.	(Rd) (Rs)
17	Extended address post-increment register indirect (0000_0000 to FFFF_FFFFH)	The 32-bit address[*1] that contains the contents of the general-purpose register specified by the data in the instruction code in its lower-order 16 bit positions and the contents of the register R8 in its higher-order 16 bit positions is used to designate the operand in memory. Subsequently, the contents of this general-purpose register are incremented by 1 (byte access) or 2 (word access).	(Rd++) (Rs++)
18	Extended address pre-decrement register indirect (0000_0000 to FFFF_FFFFH)	The contents of the general-purpose register specified by the data in the instruction code are decremented by 1 (byte access) or 2 (word access). The 32-bit address[*1] that contains this value in its lower-order 16 bit positions and the contents of register R8 in its higher-order 16 bit positions is used to designate the operand in memory.	(--Rd) (--Rs)
19	Extended address register indirect with offset [*2] (0000_0000 to FFFF_FFFFH)	The operand in memory is designated by the 32-bit address[*1] of which the lower-order 16 bits are the contents of the result[*2] of adding the 12-bit signed offset data in the instruction code to the contents of the general-purpose register specified by the data in the instruction code and the higher-order 16 bits are the contents of the base register specified in the instruction code.	(Rb, Rd, ±n) (Rb, Rs, ±n)
20	Extended address post-increment register indirect with offset [*2] (0000_0000 to FFFF_FFFFH)	The operand in memory is designated by the 32-bit address[*1] of which the lower-order 16 bits are the contents of the result[*2] of adding the 12-bit signed offset data in the instruction code to the contents of the general-purpose register specified by the data in the instruction code and the higher-order 16 bits are the contents of the base register specified in the instruction code. Subsequently, the contents of this general-purpose register are incremented by 1 (byte access) or 2 (word access).	(Rb, Rd++, ±n) (Rb, Rs++, ±n)
21	Extended address pre-decrement register indirect with offset[*2] (0000_0000 to FFFF_FFFFH)	The contents of the general-purpose register specified by the data in the instruction code are decremented by 1 (byte access) or 2 (word access). The 32-bit address[*1] of which the lower-order 16 bits are the results[*2] of adding to this value the 12-bit signed offset data in the instruction code and the higher-order 16 bits are the contents of the base register specified in the instruction coded is used to designate the operand in memory.	(Rb, --Rd, ±n) (Rb, --Rs, ±n)

*1: When a word is accessed, the higher-order byte of the operand is designated if LSB of the address data is 1 and the lower-order byte if the LSB is 0.

*2: Any carry or borrow occurring as the result of the 16-bit arithmetic operation is ignored.

5.2.8 Addressing (program counter (PC))

	Addressing Mode	Description	Symbol
22	Direct absolute PC (00_0000 to FF_FFFFH)	The 24-bit data in the instruction code is used to designate the PC value directly.	a24
23	Register indirect absolute PC (0000_0000H to FFFF_FFFFH)	The PC value is designated directly by the concatenation of the contents of the two general-purpose registers (32-bit data) specified by the data in the instruction code.	(Rb, Rs)
24	Direct relative PC (0000_0000H to FFFF_FFFFH)	The PC value is designated by the current value of the PC plus the 8- or 12-bit signed data in the instruction code.	r8 r12
25	Register indirect relative PC (0000_0000H to FFFF_FFFFH)	The PC value is designated by the current value of the PC plus the contents of the general-purpose register specified by the data in the instruction code that is regarded as 16-bit signed data.	Rs

5.3 Coding Conventions

This chapter provides a description of a set of Xstormy16 instructions. The symbols used in the individual instruction descriptions are explained below.

[] : Indicates that the item(s) are optional.

Underscore : Underscores identifies argument descriptions.
These include immediate data, memory addresses, and labels.

#immD : The "#" in the first place denotes the keyword which indicates that the following argument is immediate data. "immD" following the "#" represents the immediate data. "D" indicates the allowable bit length of the immediate data.
The valid value range of "D" varies with each instruction.

Rd : The first "R" denotes the keyword which indicates that the argument is a general-purpose register. "d" following "R" represents the number of the general-purpose register. The valid value range of "d" varies with each instruction.
(Example: R0, r0, R7, r7, R8, r8, R13, r13)

Rs : The basic coding conventions are identical to those for the above Rd.
In this manual, the source of transfer operation is identified by Rs and the destination by Rd.

Rx : Denotes a general-purpose register that is designated indirectly by the value of bits 12 to 15 (N0 to N3) of the PSW. When coding, write Rx directly.

m16 : Denotes the target address. The value range of m16 is from 00H to FFH (0000H to 00FFH) when RAM is to be manipulated. When SFRs are to be manipulated, the value range is form 7F00H to 7FFFH.

() : Denote the contents on which the operation is to be performed.
(m16), for example, represents the contents of the designated RAM or SFR. (PC) represents the value of the program counter.

Hibyte : Denote the higher-order 8 bits of 16-bit data or general-purpose register.

Lobyte : Denote the lower-order 8 bits of 16-bit data or general-purpose register.

++ : Denotes post incrementing (incremented by 1 after the operation is performed).

—— : Denotes predecrementing (decremented by 1 before the operation is performed).

PC : Denotes the program counter.

SP : Denotes the stack pointer.

PSW : Denotes the program status word.

CY : Denotes the flag containing the carry/borrow from bit 15.

HC : Denotes the flag containing the carry/borrow from bit 3.

OV : Denotes the overflow flag.

Z8 : Denotes the zero flag for the lower-order 8 bits.

Z16 : Denotes the zero flag for the data.

P : Denotes the parity flag.

S : Denotes the sign flag.

& : Denotes the logical AND operator.

| : Denotes the logical OR operator.

^ : Denotes the exclusive OR operator.

5.4 Instruction Descriptions

ADC Rd, #imm4

Instruction code	[0 1 0 1 0 0 1 1][i3i2i1i0d3d2d1d0] 5300H
Argument	Rd = 4bit(R select), imm4 = 4bit(immediate data)
Word count	1
Cycle count	1
Function	(Rd) ← (Rd) + #imm4 + CY, (PC) ← (PC) +2
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the contents of the general-purpose register designated by Rd, immediate data desaignated by imm4, and the value of the carry flag (CY) and places the result in Rd.
The legitimate value range designated by Rd is from R0 to R15 and that by imm4 from 0 to Fh.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	HC	OV	P	S
		-	-	-	-	-	-	-				-	-
MOV.W	R0, #0x7FFF	7FFFh	-	-	-	0	0	0	-	-	-	1	0
MOV.W	R1, #0x8766	7FFFh	8766h	-	-	1	0	0	-	-	-	0	1
MOV.W	R2, #0xFFFF	7FFFh	8766h	FFFFh	-	2	0	0	-	-	-	0	1
MOV.W	R3, #0x3456	7FFFh	8766h	FFFFh	3456h	3	0	0	-	-	-	1	0
ADC	R0, #0x06	8005h	8766h	FFFFh	3456h	0	0	0	0	1	1	1	1
ADC	R1, #0x0A	8005h	8770h	FFFFh	3456h	1	0	0	0	1	0	1	1
ADC	R2, #0x01	8005h	8770h	0000h	3456h	2	1	1	1	1	0	0	0
ADC	R3, #0x0F	8005h	8770h	0000h	3466h	3	0	0	0	1	0	1	0

ADC Rd, #imm16

Instruction code	0 0 1 1 0 0 0 1][0 1 0 1 d3d2d1d0][i15 to i8][i7 to i0] 3150H
Argument	Rd = 4bit(R select), imm16 = 16bit(immediate data)
Word count	2
Cycle count	2
Function	(Rd) ← (Rd) + #imm16 + CY, (PC) ← (PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the contents of the general-purpose register designated by Rd, immediate data desaignated by imm16, and the value of the carry flag (CY) and places the result in Rd.
The legitimate value range designated by Rd is from R0 to R15 and that by imm16 from 0 to FFFFh.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	HC	OV	P	S
		-	-	-	-	-	-	-				-	-
MOV.W	R0, #0x7FFF	7FFFh	-	-	-	0	0	0	-	-	-	1	0
MOV.W	R1, #0x8766	7FFFh	8766h	-	-	1	0	0	-	-	-	0	1
MOV.W	R2, #0xFFFF	7FFFh	8766h	FFFFh	-	2	0	0	-	-	-	0	1
MOV.W	R3, #0x3456	7FFFh	8766h	FFFFh	3456h	3	0	0	-	-	-	1	0
ADC	R0, #0x00F6	80F5h	8766h	FFFFh	3456h	0	0	0	0	1	1	1	1
ADC	R1, #0xA987	80F5h	30EDh	FFFFh	3456h	1	0	0	1	0	1	0	0
ADC	R2, #0x0001	80F5h	30EDh	0001h	3456h	2	0	0	1	1	0	1	0
ADC	R3, #0x0055	80F5h	30EDh	0001h	34Ach	3	0	0	0	0	0	1	0

ADC Rd, Rs

Instruction code	[0 1 0 0 1 0 1 1][s3s2s1s0d3d2d1d0] 4B00H
Argument	Rd = 4bit(R select), Rs = 4bit(R select)
Word count	1
Cycle count	1
Function	(Rd) ← (Rd) + (Rs) + CY, (PC) ← (PC)+2
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the contents of general-purpose register designated by Rd, the contents of the general-purpose register designated by Rs, and the value of the carry flag (CY) and places the result in Rd. The legitimate value range designated by Rd is from R0 to R15 and that by Rs from R0 to R15.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	HC	OV	P	S
		-	-	-	-	-	-	-				-	-
MOV.W	R0, #0x789A	789Ah	-	-	-	0	0	0	-	-	-	0	0
MOV.W	R1, #0x8766	789Ah	8766h	-	-	1	0	0	-	-	-	0	1
MOV.W	R2, #0xFEDC	789Ah	8766h	FEDCh	-	2	0	0	-	-	-	0	1
MOV.W	R3, #0x3456	789Ah	8766h	FEDCh	3456h	3	0	0	-	-	-	1	0
ADC	R0, R1	0000h	8766h	FEDCh	3456h	0	1	1	1	1	0	0	0
ADC	R1, R2	0000h	8643h	FEDCh	3456h	1	0	0	1	1	0	0	1
ADC	R2, R3	0000h	8643h	3333h	3456h	2	0	0	1	1	0	0	0
ADC	R3, R0	0000h	8643h	3333h	3457h	3	0	0	0	0	0	0	0
ADC	R3, R2	0000h	8643h	3333h	678Ah	3	0	0	0	0	0	0	0
ADC	R3, R2	0000h	8643h	3333h	9ABDh	3	0	0	0	0	1	0	1

ADC Rx, #imm8

Instruction code	[0 1 0 1 1 0 1 1][i7i6i5i4i3i2i1i0] 5B00H
Argument	imm8 = 8bit(immediate data)
Word count	1
Cycle count	1
Function	(Rx) ← (Rx) + #imm8 + CY, (PC) ← (PC)+2
Affected flags	Z8, Z16, CY, HC, OV, P, S

[Description]

This instruction adds the contents of the general-purpose register Rx designated indirectly by bits 12 to 15 (N0 to N3) of the PSW, immediate data designated by imm8, and the value of the carry flag (CY) and places the result in Rx.

The legitimate value range designated by imm8 is from 0 to FF.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	HC	OV	P	S
		-	-	-	-	-	-	-				-	-
MOV.W	R3, #0x3456	-	-	-	3456h	3	0	0	-	-	-	1	0
MOV.W	R2, #0x0000	-	-	0000h	3456h	2	1	1	-	-	-	0	0
MOV.W	R1, #0x8766	-	8766h	0000h	3456h	1	0	0	-	-	-	0	1
MOV.W	R0, #0x7FFF	7FFFh	8766h	0000h	3456h	0	0	0	-	-	-	1	0
ADC	Rx, #0xF6	80F5h	8766h	0000h	3456h	0	0	0	0	1	1	1	1
INC	R1	80F5h	8767h	0000h	3456h	1	0	0	0	1	1	1	1
ADC	Rx, #0x99	80F5h	8800h	0000h	3456h	1	1	0	0	1	0	0	1
NOT	R2	80F5h	8800h	FFFFh	3456h	2	0	0	0	1	0	0	1
ADC	Rx, #0x01	80F5h	8800h	0000h	3456h	2	1	1	1	1	0	0	0
SWPB	R3	80F5h	8800h	0000h	5634h	3	0	0	1	1	0	1	0
ADC	Rx, #0x55	80F5h	8800h	0000h	568Ah	3	0	0	0	0	0	1	0

ADD Rd, #imm4

Instruction code	[0 1 0 1 0 0 0 1][i3i2i1i0d3d2d1d0] 5100H
Argument	Rd = 4bit(R select), imm4 = 4bit(immediate data)
Word count	1
Cycle count	1
Function	(Rd) ← (Rd) + #imm4, (PC)←(PC)+2
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the contents of the general-purpose register designated by Rd and immediate data designated by imm4 and places the result in Rd.
The legitimate value range designated by Rd is from R0 to R15 and that by imm4 from 0 to Fh.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	HC	OV	P	S
		-	-	-	-	-	-	-				-	-
MOV.W	R0, #0x7FFF	7FFFh	-	-	-	0	0	0	-	-	-	1	0
MOV.W	R1, #0x8766	7FFFh	8766h	-	-	1	0	0	-	-	-	0	1
MOV.W	R2, #0xFFFF	7FFFh	8766h	FFFFh	-	2	0	0	-	-	-	0	1
MOV.W	R3, #0x3456	7FFFh	8766h	FFFFh	3456h	3	0	0	-	-	-	1	0
ADD	R0, #0x06	8005h	8766h	FFFFH	3456h	0	0	0	0	1	1	1	1
ADD	R1, #0x0A	8005h	8770h	FFFFh	3456h	1	0	0	0	1	0	1	1
ADD	R2, #0x01	8005h	8770h	0000h	3456h	2	1	1	1	1	0	0	0
ADD	R3, #0x0F	8005h	8770h	0000h	3465h	3	0	0	0	1	0	1	0

ADD Rd, #imm16

Instruction code	[0 0 1 1 0 0 0 1][0 1 0 0 d3d2d1d0][i15 to i8][i7 to i0] 3140H
Argument	Rd = 4bit(R select), imm16 = 16bit(immediate data)
Word count	2
Cycle count	2
Function	(Rd) ← (Rd) + #imm16, (PC) ← (PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the contents of the general-purpose register designated by Rd and immediate data designated by imm16 and places the result in Rd. The legitimate value range designated by Rd is from R0 to R15 and that by imm16 from 0 to FFFFh.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	HC	OV	P	S
		-	-	-	-	-	-	-				-	-
MOV.W	R0, #0x7FFF	7FFFh	-	-	-	0	0	0	-	-	-	1	0
MOV.W	R1, #0x8766	7FFFh	8766h	-	-	1	0	0	-	-	-	0	1
MOV.W	R2, #0xFFFF	7FFFh	8766h	FFFFh	-	2	0	0	-	-	-	0	1
MOV.W	R3, #0x3456	7FFFh	8766h	FFFFh	3456h	3	0	0	-	-	-	1	0
ADD	R0, #0x00F6	80F5h	8766h	FFFFh	3456h	0	0	0	0	1	1	1	1
ADD	R1, #0xA987	80F5h	30EDh	FFFFh	3456h	1	0	0	1	0	1	0	0
ADD	R2, #0x0001	80F5h	30EDh	0000h	3456h	2	1	1	1	1	0	0	0
ADD	R3, #0x0055	80F5h	30EDh	0000h	34ABh	3	0	0	0	0	0	0	0

ADD Rd, Rs

Instruction code	[0 1 0 0 1 0 0 1][s3s2s1s0d3d2d1d0] 4900H
Argument	Rd = 4bit(R select), Rs = 4bit(R select)
Word count	1
Cycle count	1
Function	(Rd) ← (Rd) + (Rs), (PC) ← (PC)+2
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the contents of the general-purpose register designated by Rd and the general-purpose register designated by Rs and places the result in Rd.
The legitimate value range designated by Rd is from R0 to R15 and that by Rs from R0 to R15.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	HC	OV	P	S
		-	-	-	-	-	-	-				-	-
MOV.W	R0, #0x789A	789Ah	-	-	-	0	0	0	-	-	-	0	0
MOV.W	R1, #0x8766	789Ah	8766h	-	-	1	0	0	-	-	-	0	1
MOV.W	R2, #0xFEDC	789Ah	8766h	FEDCh	-	2	0	0	-	-	-	0	1
MOV.W	R3, #0x3456	789Ah	8766h	FEDCh	3456h	3	0	0	-	-	-	1	0
ADD	R0, R1	0000h	8766h	FEDCh	3456h	0	1	1	1	1	0	0	0
ADD	R1, R2	0000h	8642h	FEDCh	3456h	1	0	0	1	1	0	1	1
ADD	R2, R3	0000h	8642h	3332h	3456h	2	0	0	1	1	0	1	0
ADD	R3, R0	0000h	8642h	3332h	3456h	3	0	0	0	0	0	1	0
ADD	R3, R2	0000h	8642h	3332h	6788h	3	0	0	0	0	0	1	0
ADD	R3, R2	0000h	8642h	3332h	9ABAh	3	0	0	0	0	1	1	1

ADD Rx, #imm8

Instruction code	[0 1 0 1 1 0 0 1][i7i6i5i4i3i2i1i0] 5900H
Argument	imm8 = 8bit(immediate data)
Word count	1
Cycle count	1
Function	(Rx) ← (Rx) + #imm8, (PC) ← (PC)+2
Affected flags	Z8, Z16, CY, HC, OV, P, S

[Description]

This instruction adds the contents of the general-purpose register Rx designated indirectly by bits 12 to 15 (N0 to N3) of the PSW and immediate data designated by imm8 and places the result in Rx.
The legitimate value range designated by imm8 is from 0 to FFh.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	HC	OV	P	S
		-	-	-	-	-	-	-				-	-
MOV.W	R3, #0x3456	-	-	-	3456h	3	0	0	-	-	-	1	0
MOV.W	R2, #0x0000	-	-	0000h	3456h	2	1	1	-	-	-	0	0
MOV.W	R1, #0x8766	-	8766h	0000h	3456h	1	0	0	-	-	-	0	1
MOV.W	R0, #0x7FFF	7FFFh	8766h	0000h	3456h	0	0	0	-	-	-	1	0
ADD	Rx, #0xF6	80F5h	8766h	0000h	3456h	0	0	0	0	1	1	1	1
INC	R1	80F5h	8767h	0000h	3456h	1	0	0	0	1	1	1	1
ADD	Rx, #0x99	80F5h	8800h	0000h	3456h	1	1	0	0	1	0	0	1
NOT	R2	80F5h	8800h	FFFFh	3456h	2	0	0	0	1	0	0	1
ADD	Rx, #0x01	80F5h	8800h	0000h	3456h	2	1	1	1	1	0	0	0
SWPB	R3	80F5h	8800h	0000h	5634h	3	0	0	1	1	0	1	0
ADD	Rx, #0x55	80F5h	8800h	0000h	5689h	3	0	0	0	0	0	1	0

AND Rd, #imm16

Instruction code	[0 0 1 1 0 0 0 1][0 0 0 0 d3d2d1d0][i15 to i8][i7 to i0] 3100H
Argument	Rd = 4bit(R select), imm16 = 16bit(immediate data)
Word count	2
Cycle count	2
Function	(Rd) ← (Rd) & #imm16, (PC) ← (PC)+4
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction takes the AND of the contents of general-purpose register designated by Rd and immediate data designated by imm16 and places the result in Rd.
The legitimate value range designated by Rd is from R0 to R15 and that by imm16 from 0 to FFFFh.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	HC	OV	P	S
		-	-	-	-	-	-	-	-	-		-	-
MOV.W	R0, #0x5678	5678h	-	-	-	0	0	0	0	0	-	1	0
MOV.W	R1, #0x0000	5678h	0000h	-	-	1	1	1	0	0	-	0	1
MOV.W	R2, #0xFEDC	5678h	0000h	FEDCh	-	2	0	0	0	1	-	0	1
MOV.W	R3, #0x3456	5678h	0000h	FEDCh	3456h	3	0	0	1	0	-	1	0
AND	R0, #0xFFFF	5678h	0000h	FEDCh	3456h	0	0	0	0	0	1	1	1
AND	R1, #0x89AB	5678h	0000h	FEDCh	3456h	1	1	1	0	0	1	0	0
AND	R2, #0x9ABC	5678h	0000h	9A9Ch	3456h	2	0	0	0	1	0	1	0
AND	R3, #0x1234	5678h	0000h	9A9Ch	1014h	3	0	0	1	0	0	1	0

AND Rd, Rs

Instruction code	[0 1 0 0 0 0 0 0][s3s2s1s0d3d2d1d0] 4000H
Argument	Rd = 4bit(R select), Rs = 4bit(R select)
Word count	1
Cycle count	1
Function	(Rd) ← (Rd) & (Rs), (PC)←(PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction takes the AND of the contents of the general-purpose register designated by Rd and the general-purpose register designated by Rs and places the result in Rd.
The legitimate value range designated by Rd is from R0 to R15 and that by Rs from R0 to R15.

[Example]

	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
	-	-	-	-	-	-	-	-	-
MOV.W R0, #0x5678	5678h	-	-	-	0	0	0	0	0
MOV.W R1, #0x0000	5678h	0000h	-	-	1	1	1	0	0
MOV.W R2, #0x1200	5678h	0000h	1200h	-	2	1	0	0	0
MOV.W R3, #0xFFFF	5678h	0000h	1200h	FFFFh	3	0	0	0	1
AND R0, R3	5678h	0000h	1200h	FFFFh	0	0	0	0	0
AND R1, R3	5678h	0000h	1200h	FFFFh	1	1	1	0	0
AND R2, R3	5678h	0000h	1200h	FFFFh	2	1	0	0	0
AND R2, R0	5678h	0000h	1200h	FFFFh	2	1	0	0	0
MOV.W R0, #0x8118	8118h	0000h	1200h	FFFFh	0	0	0	0	1
MOV.W R1, #0x5678	8118h	5678h	1200h	FFFFh	1	0	0	0	0
MOV.W R3, #0x3456	8118h	5678h	3456h	FFFFh	2	0	0	1	0
AND R0, R3	8118h	5678h	3456h	FFFFh	0	0	0	0	1
AND R1, R3	8118h	5678h	3456h	FFFFh	1	0	0	0	0
AND R2, R3	8118h	5678h	3456h	FFFFh	2	0	0	1	0
AND R2, R0	8118h	5678h	0010h	FFFFh	2	0	0	1	0

AND Rx, #imm8

Instruction code	[0 1 0 0 0 0 0 1][i7i6i5i4i3i2i1i0] 4100H
Argument	imm8 = 8bit(immediate data)
Word count	1
Cycle count	1
Function	(Rx) ← (Rx) & 16bit data(Hibyte=00H, Lobyte=#imm8), (PC) ← (PC)+2
Affected flags	Z8, Z16, P, S

[Description]

This instruction takes the AND of the contents of the general-purpose register (Rx) designated indirectly by bits 12 to 15 (N0 to N3) of the PSW and 16-bit data (of which the higher-order 8 bits are 00H and the lower-order 8 bits are #imm8) and places the result in Rx.
The legitimate value range designated by imm8 is from 0 to FFh.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0000	-	-	-	0000h	3	1	1	0	0
MOV.W	R2, #0x0012	-	-	0012h	0000h	2	0	0	0	0
MOV.W	R1, #0x0000	-	0000h	0012h	0000h	1	1	1	0	0
MOV.W	R0, #0x5678	5678h	0000h	0012h	0000h	0	0	0	0	0
AND	Rx, #0x08	0008h	0000h	0012h	0000h	0	0	0	1	0
INC	R1	0008h	0001h	0012h	0000h	1	0	0	1	0
AND	Rx, #0x01	0008h	0001h	0012h	0000h	1	0	0	1	0
SWPB	R2	0008h	0001h	1200h	0000h	2	1	0	0	0
AND	Rx, #0x41	0008h	0001h	0000h	0000h	2	1	1	0	0
DEC	R3	0008h	0001h	0000h	FFFFh	3	0	0	0	1
AND	Rx, #0xFF	0008h	0001h	0000h	00FFh	3	0	0	0	0

ASR Rd, #imm4

Instruction code	[0 0 1 1 0 1 1 1][i3i2i1i0d3d2d1d0] 3700H
Argument	Rd = 4bit(R select), imm4 = 4bit(immediate data)
Word count	1
Cycle count	1
Function	(Rd) ← (Rd) arithmetic shift right #imm4 bit (CY) ← last shift bit, (PC) ← (PC)+2
Affected flags	Z8, Z16, CY, P, S, N0 to N3

[Description]

This instruction performs an arithmetic shift right of the contents of the general-purpose register designated by Rd by value (arithmetic shift amount) indicated by immediate data designated by imm4 and places the carryover bit out of the LSB in the carry flag (CY).
The legitimate value range designated by Rd is from R0 to R15 and that by imm4 from 0 to Fh.

[Example]

	R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	P	S
	-	-	-	-	-	-	-		-	-
MOV.W R0, #0xCDEF	CDEFh	-	-	-	0	0	0	-	0	1
MOV.W R1, #0x5432	CDEFh	5432h	-	-	1	0	0	-	0	0
MOV.W R2, #0x0000	CDEFh	5432h	0000h	-	2	1	1	-	0	0
MOV.W R3, #0x7654	CDEFh	5432h	0000h	7654h	3	0	0	-	0	0
CLR1 R14, #2	CDEFh	5432h	0000h	7654h	E	1	0	0	0	0
ASR R0, #0x02	F37Bh	5432h	0000h	7654h	0	0	0	1	0	1
ASR R1, #0x00	F37Bh	5432h	0000h	7654h	1	0	0	1	0	0
ASR R2, #0x04	F37Bh	5432h	0000h	7654h	2	1	1	0	0	0
ASR R3, #0x0B	F37Bh	5432h	0000h	000Eh	3	0	0	1	1	0

<Note>

During the execution of an arithmetic shift instruction, the MSB of Rd is regarded as the sign bit and its value remains unchanged during the shift operations. The value of the MSB is copied to the right bit position on each shift operation.

ASR Rd, Rs

Instruction code	[0 0 1 1 0 1 1 0][s3s2s1s0d3d2d1d0] 3600H
Argument	Rd = 4bit(R select), Rs = 4bit(R select)
Word count	1
Cycle count	1
Function	(Rd) ← (Rd) arithmetic shift right (Lower 4bit value of Rs) bit (CY) ← last shift bit, (PC) ← (PC)+2
Affected flags	Z8, Z16, CY, P, S, N0 to N3

[Description]

This instruction performs an arithmetic shift right of the contents of the general-purpose register designated by Rd by the value (arithmetic shift amount) indicated by the lower-order 4 bits of the general-purpose register designated by Rs and places the carryover bit out of the LSB in the carry flag (CY).
The legitimate value range indicated by Rd is from R0 to R15 and that by RsR0 to R15.

[Example]

	R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	P	S
	-	-	-	-	-	-	-		-	-
MOV.W R0, #0xCDEF	CDEFh	-	-	-	0	0	0	-	0	1
MOV.W R1, #0x5432	CDEFh	5432h	-	-	1	0	0	-	0	0
MOV.W R2, #0x0000	CDEFh	5432h	0000h	-	2	1	1	-	0	0
MOV.W R3, #0x7654	CDEFh	5432h	0000h	7654h	3	0	0	-	0	0
CLR1 R14, #2	CDEFh	5432h	0000h	7654h	E	1	0	0	0	0
ASR R0, R1	F37Bh	5432h	0000h	7654h	0	0	0	1	0	1
ASR R1, R2	F37Bh	5432h	0000h	7654h	1	0	0	1	0	0
ASR R2, R3	F37Bh	5432h	0000h	7654h	2	1	1	0	0	0
ASR R3, R0	F37Bh	5432h	0000h	000Eh	3	0	0	1	1	0

<Note>

During the execution of an arithmetic shift instruction, the MSB of Rd is regarded as the sign bit and its value remains unchanged during the shift operations. The value of the MSB is copied to the right bit position on each shift operation.

BC r8

Instruction code	[1 1 0 1 0 0 1 1][r7r6r5r4r3r2r1r0] D300H
Argument	r8 = 8bit(relative address, signed)
Word count	1
Cycle count	2 or 3
Function	If CY=1, then (PC) ← (PC)+2±(r8) If CY=0, then (PC) ← (PC)+2
Affected flags	

[Description]

This instruction adds the value of relative address designated by r8 + 2 to the program counter (PC) and places the result in the PC if the carry flag (CY) is 1. If CY is 0, 2 is added to the PC.
The legitimate value range of the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

```
        MOV.W   R2, #0x0002
        MOV.W   R3, 0xFFFF
        RRC     R2, #1
loop:
        BC      LA          ;; NOT JUMP LA
        RRC     R2, #1
        BC      LB          ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R2	R3	PSW
-	-	-	-
9002h	0002h	-	2020h
9006h	0002h	FFFFh	3040h
9008h	0001h	FFFFh	2020h
900Ah	0001h	FFFFh	2020h
900Ch	0000h	FFFFh	2007h
9014h	0000h	FFFFh	2007h
-	-	-	-
-	-	-	-
-	-	-	-
9016h	0000h	0000h	3007h
9018h	0000h	0000h	3007h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BC Rd, #imm8, r12

Instruction code	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][0 0 1 1 r11 to r8][r7 to r0] 20003000H
Argument	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of unsigned comparison is (Rd) < #imm8, then (PC)←(PC)+4±(r12) If result of unsigned comparison is (Rd) ≧#imm8, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting immediate data designated by imm8 from the contents of the general-purpose register designated by Rd is negative. If the result of the subtraction is nonnegative, 4 is added to the PC.

The legitimate value range designated by Rd is from R0 to R7, that by imm8 is from 0 to FFh, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

		PC	R0	R1	R2	R3	PSW
		-	-	-	-	-	-
	MOV.W R0, #0x0056	9002h	0056h	-	-	-	0000h
	MOV.W R1, #0x0012	9004h	0056h	0012h	-	-	1000h
	MOV.W R2, #0x0056	9006h	0056h	0012h	0056h	-	2000h
	MOV.W R3, #0xFFFF	900Ah	0056h	0012h	0056h	FFFFh	3040h
loop:							
	BC R0,#0x56, LA ;; NOT JUMP LA	900Eh	0056h	0012h	0056h	FFFFh	0003h
	BC R1,#0x56, LB ;; JUMP LB	9018h	0056h	0012h	0056h	FFFFh	106Ch
	BR loop	-	-	-	-	-	-
LA:							
	DEC R3	-	-	-	-	-	-
	BR loop	-	-	-	-	-	-
LB:							
	INC R3	901Ah	0056h	0012h	0056h	0000h	300Fh
	NOP	901Ch	0056h	0012h	0056h	0000h	300Fh

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BC Rd, Rs, r12

Instruction code	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][0 0 1 1 r11 to r8][r7 to r0] 0D003000H
Argument	Rd = 4bit(R select), Rs = 4bit(R select), r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of unsigned comparison is (Rd) < (Rs), then (PC)←(PC)+4±(r12) If result of unsigned comparison is (Rd)≧(Rs), then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting the contents of the general-purpose register designated by Rs from the contents of the general-purpose register designated by Rd is negative. If the result of the subtraction is nonnegative, 4 is added to the PC.

The legitimate value range designated by Rd is from R0 to R15, that by Rs is from R0 to R15, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0x5678
        MOV.W   R1, #0x1234
        MOV.W   R2, #0x5678
        MOV.W   R3, #0xFFFF
loop:
        BC      R0, R2, LA      ;; NOT JUMP LA
        BC      R1, R2, LB      ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9004h	5678h	-	-	-	0000h
9008h	5678h	1234h	-	-	1020h
900Ch	5678h	1234h	5678h	-	2000h
9010h	5678h	1234h	5678h	FFFFh	3040h
9014h	5678h	1234h	5678h	FFFFh	0003h
901Eh	5678h	1234h	5678h	FFFFh	106Ch
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	5678h	1234h	5678h	0000h	300Fh
9022h	5678h	1234h	5678h	0000h	300Fh

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BC Rx, #imm16, r8

Instruction code	[1 1 0 0 0 0 1 1][r7r6r5r4r3r2r1r0][i15 to i8][i7 to i0] C3000000H
Argument	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)
Word count	2
Cycle count	3 or 4
Function	If result of unsigned comparison is (Rx) < #imm16, then (PC)←(PC)+4±(r8) If result of unsigned comparison is (Rx)≧#imm16, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S

[Description]

This instruction adds the value of the relative address designated by r8 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting immediate data designated by imm16 from the contents of the general-purpose register Rx designated indirectly by bits 12 to 15 (N0 to N3) of the PSW is negative. If the result of the subtraction is nonnegative, 4 is added to the PC.

The legitimate value range designated by imm16 is from 0 to FFFFh, and that by the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

		PC	R0	R1	R2	R3	PSW
		-	-	-	-	-	-
	MOV.W R2, #0x5678	9004h	-	-	5678h	-	2000h
	MOV.W R3, #0xFFFF	9008h	-	-	5678h	FFFFh	3040h
loop:							
	MOV.W R0, #0x5678	900Ch	5678h	-	5678h	FFFFh	0000h
	BC Rx, #0x5678, LA ;; NOT JUMP LA	9010h	5678h	-	5678h	FFFFh	0003h
	MOV.W R1, #0x1234	9014h	5678h	1234h	5678h	FFFFh	1020h
	BC Rx, #0x5678, LB ;; JUMP LB	901Eh	5678h	1234h	5678h	FFFFh	106Ch
	BR loop	-	-	-	-	-	-
LA:							
	DEC R3	-	-	-	-	-	-
	BR loop	-	-	-	-	-	-
LB:							
	INC R3	9020h	5678h	1234h	5678h	0000h	300Fh
	NOP	9022h	5678h	1234h	5678h	0000h	300Fh

<Note>

This instruction takes 4 cycles to execute if the conditions are met.

BGE r8

Instruction code	[1 1 0 1 0 0 0 0][r7r6r5r4r3r2r1r0] D000H
Argument	r8 = 8bit(relative address, signed)
Word count	1
Cycle count	2 or 3
Function	If S ^ OV = 0, then (PC)←(PC)+2±(r8) If S ^ OV = 1, then (PC)←(PC)+2
Affected flags	

[Description]

This instruction adds the value of the relative address designated by r8 + 2 to the program counter (PC) and places the result in the PC if the result of the exclusive logical OR of the sign flag (S) and overflow flag (OV) is 0. If the result of the logical operation is 1, 2 is added to the PC.
The legitimate value range of the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

```
        MOV.W   R3,#0xFFFF
loop:
        BGE     LA          ;; NOT JUMP LA
        MOV.W   R3, #0x0000
        BGE     LB          ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R3	PSW
-	-	-
9004h	FFFFh	3040h
9006h	FFFFh	3040h
9008h	0000h	3003h
9010h	0000h	3003h
-	-	-
-	-	-
-	-	-
9012h	0001h	3020h
9014h	0001h	3020h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BGE Rd, #imm8, r12

Instruction code	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][0 0 0 0 r11 to r8][r7 to r0] 20000000H
Argument	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of signed comparison is (Rd)≧#imm8, then (PC)←(PC)+4±(r12) If result of signed comparison is (Rd) < #imm8, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting immediate data designated by imm8 from the contents (signed 16-bit data) of the general-purpose register designated by Rd is nonnegative. If the result of the subtraction is negative, 4 is added to the PC.

The legitimate value range designated by Rd is from R0 to R7, that by imm8 is from 0 to FFh, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

	PC	R0	R1	R2	R3	PSW
	-	-	-	-	-	-
MOV.W R0, #0xCDEF	9004h	CDEFh	-	-	-	0040h
MOV.W R1, #0x789A	9008h	CDEFh	789Ah	-	-	1000h
MOV.W R2, #0x1234	900Ch	CDEFh	789Ah	1234h	-	2020h
MOV.W R3, #0xFFFF	9010h	CDEFh	789Ah	1234h	FFFFh	3040h
loop:						
BGE R0, #0x12, LA ;; NOT JUMP LA	9014h	CDEFh	789Ah	1234h	FFFFh	0060h
BGE R1, #0x12, LB ;; JUMP LB	901Eh	CDEFh	789Ah	1234h	FFFFh	1000h
BR loop	-	-	-	-	-	-
LA:						
DEC R3	-	-	-	-	-	-
BR loop	-	-	-	-	-	-
LB:						
INC R3	9020h	CDEFh	789Ah	1234h	0000h	3003h
NOP	9022h	CDEFh	789Ah	1234h	0000h	3003h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BGE Rd, Rs, r12

Instruction code	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][0 0 0 0 r11 to r8][r7 to r0] 0D000000H
Argument	Rd = 4bit(R select), Rs = 4bit(R select), r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of signed comparison is (Rd)≧(Rs), then (PC)←(PC)+4±(r12) If result of signed comparison is (Rd) < (Rs), then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting the contents (signed 16-bit data) of the general-purpose register designated by Rs from the contents (signed 16-bit data) of the general-purpose register designated by Rd is nonnegative. If the result of the subtraction is negative, 4 is added to the PC.
The legitimate value range designated by Rd is from R0 to R15, that by Rs is from R0 to R15, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

			PC	R0	R1	R2	R3	PSW
			-	-	-	-	-	-
	MOV.W R0, #0xCDEF		9004h	CDEFh	-	-	-	0040h
	MOV.W R1, #0x789A		9008h	CDEFh	789Ah	-	-	1000h
	MOV.W R2, #0x1234		900Ch	CDEFh	789Ah	1234h	-	2020h
	MOV.W R3, #0xFFFF		9010h	CDEFh	789Ah	1234h	FFFFh	3040h
loop:								
	BGE R0, R2 , LA	;; NOT JUMP LA	9014h	CDEFh	789Ah	1234h	FFFFh	0040h
	BGE R1, R2 , LB	;; JUMP LB	901Eh	CDEFh	789Ah	1234h	FFFFh	1000h
	BR loop		-	-	-	-	-	-
LA:								
	DEC R3		-	-	-	-	-	-
	BR loop		-	-	-	-	-	-
LB:								
	INC R3		9020h	CDEFh	789Ah	1234h	0000h	3003h
	NOP		9022h	CDEFh	789Ah	1234h	0000h	3003h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BGE Rx, #imm16, r8

Instruction code	[1 1 0 0 0 0 0 0][r7r6r5r4r3r2r1r0][i15 to i8][i7 to i0] C0000000H
Argument	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)
Word count	2
Cycle count	3 or 4
Function	If result of signed comparison is (Rx)≧#imm16, then (PC)←(PC)+4±(r8) If result of signed comparison is (Rx) < #imm16, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S

[Description]

This instruction adds the value of the relative address designated by r8 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting immediate data designated by imm16 (signed 16-bit data) from the contents (signed 16-bit data) of the general-purpose register Rx designated indirectly by bits 12 to 15 (N0 to N3) of the PSW is nonnegative. If the result of the subtraction is negative, 4 is added to the PC.

The legitimate value range designated by imm16 is from 0 to FFFFh, and that by the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

		PC	R0	R1	R2	R3	PSW
		-	-	-	-	-	-
	MOV.W R2, #0x1234	9004h	-	-	1234h	-	2020h
	MOV.W R3, #0xFFFF	9008h	-	-	1234h	FFFFh	3040h
loop:							
	MOV.W R0, #0xCDEF	900Ch	CDEFh	-	1234h	FFFFh	0040h
	BGE Rx, #0x1234, LA ;; NOT JUMP LA	9010h	CDEFh	-	1234h	FFFFh	0040h
	MOV.W R1, #0x789A	9014h	CDEFh	789Ah	1234h	FFFFh	1000h
	BGE Rx, #0x1234, LB ;; JUMP LB	901Eh	CDEFh	789Ah	1234h	FFFFh	1000h
	BR loop	-	-	-	-	-	-
LA:							
	DEC R3	-	-	-	-	-	-
	BR loop	-	-	-	-	-	-
LB:							
	INC R3	9020h	CDEFh	789Ah	1234h	0000h	3003h
	NOP	9022h	CDEFh	789Ah	1234h	0000h	3003h

<Note>

This instruction takes 4 cycles to execute if the conditions are met.

BGT r8

Instruction code	[1 1 0 1 0 1 0 0][r7r6r5r4r3r2r1r0] D400H
Argument	r8 = 8bit(relative address, signed)
Word count	1
Cycle count	2 or 3
Function	If S ^ OV | Z16 = 0, then (PC)←(PC)+2±(r8) If S ^ OV | Z16 = 1, then (PC)←(PC)+2
Affected flags	

[Description]

This instruction adds the value of the relative address designated by r8 + 2 to the program counter (PC) and places the result in the PC if the result of the logical OR between the 16-bit operation flag (Z16) and the result of an exclusive logical OR of the sign flag (S) and overflow flag (OV) is 0. If the result of the logical operations is 1, 2 is added to the PC.

The legitimate value range of the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

```
        MOV.W   R3,#0xFFFF
loop:
        BGT     LA          ;; NOT JUMP LA
        MOV.W   R3, #0x1200
        BGT     LB          ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R3	PSW
-	-	-
9004h	FFFFh	3040h
9006h	FFFFh	3040h
900Ah	1200h	3001h
9012h	1200h	3001h
-	-	-
-	-	-
-	-	-
9014h	1201h	3020h
9016h	1201h	3020h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BGT Rd, #imm8, r12

Instruction code	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][0 1 0 0 r11 to r8][r7 to r0] 20004000H
Argument	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of signed comparison is (Rd) > #imm8, then (PC)←(PC)+4±(r12) If result of signed comparison is (Rd)≦#imm8, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting immediate data designated by imm8 from the contents (signed 16-bit data) of the general-purpose register designated by Rd is positive. If the result of the subtraction is nonpositive, 4 is added to the PC.

The legitimate value range designated by Rd is from R0 to R7, that by imm8 is from 0 to FFh, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0x0056
        MOV.W   R1, #0x7654
        MOV.W   R2, #0x0056
        MOV.W   R3, #0xFFFF
loop:
        BGT     R0, #0x56, LA   ;; NOT JUMP LA
        BGT     R1, #0x56, LB   ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9002h	0056h	-	-	-	0000h
9006h	0056h	7654h	-	-	1000h
9008h	0056h	7654h	0056h	-	2000h
900Ch	0056h	7654h	0056h	FFFFh	3040h
9010h	0056h	7654h	0056h	FFFFh	0003h
901Ah	0056h	7654h	0056h	FFFFh	1008h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
901Ch	0056h	7654h	0056h	0000h	300Bh
901Eh	0056h	7654h	0056h	0000h	300Bh

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BGT Rd, Rs, r12

Instruction code	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][0 1 0 0 r11 to r8][r7 to r0] 0D004000H
Argument	Rd = 4bit(R select), Rs = 4bit(R select), r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of signed comparison is (Rd) > (Rs), then (PC)←(PC)+4±(r12) If result of signed comparison is (Rd)≦(Rs), then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting the contents (signed 16-bit data) of the general-purpose register designated by Rs from the contents (signed 16-bit data) of the general-purpose register designated by Rd is positive. If the result of the subtraction is nonpositive, 4 is added to the PC.
The legitimate value range designated by Rd is from R0 to R15, that by Rs is from R0 to R15, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

			PC	R0	R1	R2	R3	PSW
			-	-	-	-	-	-
	MOV.W R0, #0x89AB		9004h	89ABh	-	-	-	0040h
	MOV.W R1, #0x789A		9008h	89ABh	789Ah	-	-	1000h
	MOV.W R2, #0x89AB		900Ch	89ABh	789Ah	89ABh	-	2040h
	MOV.W R3, #0xFFFF		9010h	89ABh	789Ah	89ABh	FFFFh	3040h
loop:								
	BGT R0, R2, LA	;; NOT JUMP LA	9014h	89ABh	789Ah	89ABh	FFFFh	0003h
	BGT R1, R2, LB	;; JUMP LB	901Eh	89ABh	789Ah	89ABh	FFFFh	107Ch
	BR loop		-	-	-	-	-	-
LA:								
	DEC R3		-	-	-	-	-	-
	BR loop		-	-	-	-	-	-
LB:								
	INC R3		9020h	89ABh	789Ah	89ABh	0000h	301Fh
	NOP		9022h	89ABh	789Ah	89ABh	0000h	301Fh

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BGT Rx, #imm16, r8

Instruction code	[1 1 0 0 0 1 0 0][r7r6r5r4r3r2r1r0][i15 to i8][i7 to i0] C4000000H
Argument	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)
Word count	2
Cycle count	3 or 4
Function	If result of signed comparison is (Rx) > #imm16, then (PC)←(PC)+4±(r8) If result of signed comparison is (Rx)≦ #imm16, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S

[Description]

This instruction adds the value of the relative address designated by r8 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting immediate data designated by imm16 (signed 16-bit data) from the contents (signed 16-bit data) of the general-purpose register Rx designated indirectly by bits 12 to 15 (N0 to N3) of the PSW is positive. If the result of the subtraction is nonpositive, 4 is added to the PC.

The legitimate value range designated by imm16 is from 0 to FFFFh, and that by the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

	PC	R0	R1	R2	R3	PSW
	-	-	-	-	-	-
MOV.W R2, #0x89AB	9004h	-	-	89ABh	-	2040h
MOV.W R3, #0xFFFF	9008h	-	-	89ABh	FFFFh	3040h
loop:						
MOV.W R0, #0x89AB	900Ch	89ABh	-	89ABh	FFFFh	0040h
BGT Rx,#0x89AB, LA ;; NOT JUMP LA	9010h	89ABh	-	89ABh	FFFFh	0003h
MOV.W R1, #0x789A	9014h	89ABh	789Ah	89ABh	FFFFh	1000h
BGT Rx,#0x89AB, LB ;; JUMP LB	901Eh	89ABh	789Ah	89ABh	FFFFh	107Ch
BR loop	-	-	-	-	-	-
LA:						
DEC R3	-	-	-	-	-	-
BR loop	-	-	-	-	-	-
LB:						
INC R3	9020h	89ABh	789Ah	89ABh	0000h	301Fh
NOP	9022h	89ABh	789Ah	89ABh	0000h	301Fh

<Note>

This instruction takes 4 cycles to execute if the conditions are met.

BHI r8

Instruction code	[1 1 0 1 0 1 0 1][r7r6r5r4r3r2r1r0] D500H
Argument	r8 = 8bit(relative address, signed)
Word count	1
Cycle count	2 or 3
Function	If CY | Z16 = 0, then (PC)←(PC)+2±(r8) If CY | Z16 = 1, then (PC)←(PC)+2
Affected flags	

[Description]

This instruction adds the value of the relative address designated by r8 + 2 to the program counter (PC) and places the result in the PC if the result of the logical OR of the carry flag (CY) and the 16-bit operation flag (Z16) is 0. If the result of the logical operation is 1, 2 is added to the PC.
The legitimate value range of the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

```
        MOV.W   R2, #0x0001
        MOV.W   R3,#0xFFFF
        RRC     R2, #1
loop:
        BHI     LA          ;; NOT JUMP LA
        RRC     R2, #1
        BHI     LB          ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R2	R3	PSW
-	-	-	-
9002h	0001h	-	2020h
9006h	0001h	FFFFh	3040h
9008h	0000h	FFFFh	2007h
900Ah	0000h	FFFFh	2007h
900Ch	8000h	FFFFh	2061h
9014h	8000h	FFFFh	2061h
-	-	-	-
-	-	-	-
-	-	-	-
9016h	8000h	0000h	3003h
9018h	8000h	0000h	3003h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BHI Rd, #imm8, r12

Instruction code	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][0 1 0 1 r11 to r8][r7 to r0] 20005000H
Argument	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of unsigned comparison is (Rd) > #imm8, then (PC)←(PC)+4±(r12) If result of unsigned comparison is (Rd)≦#imm8, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting immediate data designated by imm8 from the contents of the general-purpose register designated by Rd is positive. If the result of the subtraction is nonpositive, 4 is added to the PC.

The legitimate value range designated by Rd is from R0 to R7, that by imm8 is from 0 to FFh, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

		PC	R0	R1	R2	R3	PSW
		-	-	-	-	-	-
	MOV.W R0, #0x0089	9002h	0089h	-	-	-	0020h
	MOV.W R1, #0x0098	9004h	0089h	0098h	-	-	1020h
	MOV.W R2, #0x0089	9006h	0089h	0098h	0089h	-	2020h
	MOV.W R3, #0xFFFF	900Ah	0089h	0098h	0089h	FFFFh	3040h
loop:							
	BHI R0,#0x89, LA ;; NOT JUMP LA	900Eh	0089h	0098h	0089h	FFFFh	0003h
	BHI R1,#0x89, LB ;; JUMP LB	9018h	0089h	0098h	0089h	FFFFh	1008h
	BR loop	-	-	-	-	-	-
LA:							
	DEC R3	-	-	-	-	-	-
	BR loop	-	-	-	-	-	-
LB:							
	INC R3	901Ah	0089h	0098h	0089h	0000h	300Bh
	NOP	901Ch	0089h	0098h	0089h	0000h	300Bh

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BHI Rd, Rs, r12

Instruction code	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][0 1 0 1 r11 to r8][r7 to r0] 0D005000H
Argument	Rd = 4bit(R select), Rs = 4bit(R select), r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of unsigned comparison is (Rd) > (Rs), then (PC)←(PC)+4±(r12) If result of unsigned comparison is (Rd)≦(Rs), then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting the contents of the general-purpose register designated by Rs from the contents of the general-purpose register designated by Rd is positive. If the result of the subtraction is nonpositive, 4 is added to the PC.

The legitimate value range designated by Rd is from R0 to R15, that by Rs is from R0 to R15, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0x89AB
        MOV.W   R1, #0x9876
        MOV.W   R2, #0x89AB
        MOV.W   R3, #0xFFFF
loop:
        BHI     R0, R2, LA      ;; NOT JUMP LA
        BHI     R1, R2, LB      ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9004h	89ABh	-	-	-	0040h
9008h	89ABh	9876h	-	-	1040h
900Ch	89ABh	9876h	89ABh	-	2040h
9010h	89ABh	9876h	89ABh	FFFFh	3040h
9014h	89ABh	9876h	89ABh	FFFFh	0003h
901Eh	89ABh	9876h	89ABh	FFFFh	1008h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	89ABh	9876h	89ABh	0000h	300Bh
9022h	89ABh	9876h	89ABh	0000h	300Bh

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BHI Rx, #imm16, r8

Instruction code	[1 1 0 0 0 1 0 1][r7r6r5r4r3r2r1r0][i15 to i8][i7 to i0] C5000000H
Argument	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)
Word count	2
Cycle count	3 or 4
Function	If result of unsigned comparison is (Rx) > #imm16, then (PC)←(PC)+4±(r8) If result of unsigned comparison is (Rx)≦#imm16, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S

[Description]

This instruction adds the value of the relative address designated by r8 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting immediate data designated by imm16 from the contents of the general-purpose register Rx designated indirectly by bits 12 to 15 (N0 to N3) of the PSW is positive. If the result of the subtraction is nonpositive, 4 is added to the PC.

The legitimate value range designated by imm16 is from 0 to FFFFh, and that by the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

	PC	R0	R1	R2	R3	PSW
	-	-	-	-	-	-
MOV.W R2, #0x89AB	9004h	-	-	89ABh	-	2040h
MOV.W R3, #0xFFFF	9008h	-	-	89ABh	FFFFh	3040h
loop:						
MOV.W R0, #0x89AB	900Ch	89ABh	-	89ABh	FFFFh	0040h
BHI Rx,#0x89AB, LA ;; NOT JUMP LA	9010h	89ABh	-	89ABh	FFFFh	0003h
MOV.W R1, #0x9876	9014h	89ABh	9876h	89ABh	FFFFh	1040h
BHI Rx,#0x89AB, LB ;; JUMP LB	901Eh	89ABh	9876h	89ABh	FFFFh	1008h
BR loop	-	-	-	-	-	-
LA:						
DEC R3	-	-	-	-	-	-
BR loop	-	-	-	-	-	-
LB:						
INC R3	9020h	89ABh	9876h	89ABh	0000h	300Bh
NOP	9022h	89ABh	9876h	89ABh	0000h	300Bh

<Note>

This instruction takes 4 cycles to execute if the conditions are met.

BLE r8

Instruction code	[1 1 0 1 0 1 1 0][r7r6r5r4r3r2r1r0] D600H
Argument	r8 = 8bit(relative address, signed)
Word count	1
Cycle count	2 or 3
Function	If S ^ OV | Z16 =1, then (PC)←(PC)+2±(r8) If S ^ OV | Z16 =0, then (PC)←(PC)+2
Affected flags	

[Description]

This instruction adds the value of the relative address designated by r8 + 2 to the program counter (PC) and places the result in the PC if the result of the logical OR between the 16-bit operation flag (Z16) and the result of exclusive logical OR of the sign flag (S) and the overflow flag (OV) is 1. If the result of the logical operations is 0, 2 is added to the PC.

The legitimate value range of the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

```
        MOV.W   R3, #0x1200
loop:
        BLE     LA          ;; NOT JUMP LA
        MOV.W   R3, #0x0000
        BLE     LB          ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R3	PSW
-	-	-
9004h	1200h	3001h
9006h	1200h	3001h
9008h	0000h	3003h
9010h	0000h	3003h
-	-	-
-	-	-
-	-	-
9012h	0001h	3020h
9014h	0001h	3020h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BLE Rd, #imm8, r12

Instruction code	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][0 1 1 0 r11 to r8][r7 to r0] 20006000H
Argument	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of signed comparison is (Rd)≦#imm8, then (PC)←(PC)+4±(r12) If result of signed comparison is (Rd) > #imm8, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting immediate data designated by imm8 from the contents (signed 16-bit data) of the general-purpose register designated by Rd is nonpositive. If the result of the subtraction is positive, 4 is added to the PC.

The legitimate value range designated by Rd is from R0 to R7, that by imm8 is from 0 to FFh, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0x00EF
        MOV.W   R1, #0x0098
        MOV.W   R2, #0x00CD
        MOV.W   R3, #0xFFFF
loop:
        BLE     R0,#0xCD, LA  ;; NOT JUMP LA
        BLE     R1,#0xCD, LB  ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9002h	00EFh	-	-	-	0020h
9004h	00EFh	0098h	-	-	1020h
9006h	00EFh	0098h	00CDh	-	2020h
900Ah	00EFh	0098h	00CDh	FFFFh	3040h
900Eh	00EFh	0098h	00CDh	FFFFh	0000h
9018h	00EFh	0098h	00CDh	FFFFh	106Ch
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
901Ah	00EFh	0098h	00CDh	0000h	300Fh
901Ch	00EFh	0098h	00CDh	0000h	300Fh

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BLE Rd, Rs, r12

Instruction code	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][0 1 1 0 r11 to r8][r7 to r0] 0D006000H
Argument	Rd = 4bit(R select), Rs = 4bit(R select), r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of signed comparison is (Rd)≦(Rs), then (PC)←(PC)+4±(r12) If result of signed comparison is (Rd) > (Rs), then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting the contents (signed 16-bit data) of the general-purpose register designated by Rs from the contents (signed 16-bit data) of the general-purpose register designated by Rd is nonpositive. If the result of the subtraction is positive, 4 is added to the PC.
The legitimate value range designated by Rd is from R0 to R15, that by Rs is from R0 to R15, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0x7654
        MOV.W   R1, #0x9876
        MOV.W   R2, #0xCDEF
        MOV.W   R3, #0xFFFF
loop:
        BLE     R0, R2, LA      ;; NOT JUMP LA
        BLE     R1, R2, LB      ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9004h	7654h	-	-	-	0000h
9008h	7654h	9876h	-	-	1040h
900Ch	7654h	9876h	CDEFh	-	2040h
9010h	7654h	9876h	CDEFh	FFFFh	3040h
9014h	7654h	9876h	CDEFh	FFFFh	007Ch
901Eh	7654h	9876h	CDEFh	FFFFh	104Ch
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	7654h	9876h	CDEFh	0000h	300Fh
9022h	7654h	9876h	CDEFh	0000h	300Fh

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BLE Rx, #imm16, r8

Instruction code	[1 1 0 0 0 1 1 0][r7r6r5r4r3r2r1r0][i15 to i8][i7 to i0] C6000000H
Argument	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)
Word count	2
Cycle count	3 or 4
Function	If result of signed comparison is (Rx)≦#imm16, then (PC)←(PC)+4±(r8) If result of signed comparison is (Rx) > #imm16, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S

[Description]

This instruction adds the value of the relative address designated by r8 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting immediate data designated by imm16 (signed 16-bit data) from the contents (signed 16-bit data) of the general-purpose register Rx designated indirectly by bits 12 to 15 (N0 to N3) of the PSW is nonpositive. If the result of the subtraction is positive, 4 is added to the PC.

The legitimate value range designated by imm16 is signed 16-bit data (-32768 to 32767), and that by the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

```
        MOV.W   R2, #0xCDEF
        MOV.W   R3, #0xFFFF
loop:
        MOV.W   R0, #0x7654
        BLE     Rx,#0xCDEF,LA ;; NOT JUMP LA
        MOV.W   R1, #0x9876
        BLE     Rx,#0xCDEF,LB ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9004h	-	-	CDEFh	-	2040h
9008h	-	-	CDEFh	FFFFh	3040h
900Ch	7654h	-	CDEFh	FFFFh	0000h
9010h	7654h	-	CDEFh	FFFFh	007Ch
9014h	7654h	9876h	CDEFh	FFFFh	105Ch
901Eh	7654h	9876h	CDEFh	FFFFh	104Ch
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	7654h	9876h	CDEFh	0000h	300Fh
9022h	7654h	9876h	CDEFh	0000h	300Fh

<Note>

This instruction takes 4 cycles to execute if the conditions are met.

BLS r8

Instruction code	[1 1 0 1 0 1 1 1][r7r6r5r4r3r2r1r0] D700H
Argument	r8 = 8bit(relative address, signed)
Word count	1
Cycle count	2 or 3
Function	If CY | Z16 = 1, then (PC)←(PC)+2±(r8) If CY | Z16 = 0, then (PC)←(PC)+2
Affected flags	

[Description]

This instruction adds the value of the relative address designated by r8 + 2 to the program counter (PC) and places the result in the PC if the result of the logical OR of the carry flag (CY) and the 16-bit operation flag (Z16) is 1. If the result of the logical operation is 0, 2 is added to the PC.
The legitimate value range of the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

```
        MOV.W   R2, #0x0002
        MOV.W   R3,#0xFFFFh
        RRC     R2, #1
loop:
        BLS     LA          ;; NOT JUMP LA
        RRC     R2, #1
        BLS     LB          ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R2	R3	PSW
-	-	-	-
9002h	0002h	-	2020h
9006h	0002h	FFFFh	3040h
9008h	0001h	FFFFh	2020h
900Ah	0001h	FFFFh	2020h
900Ch	0000h	FFFFh	2007h
9014h	0000h	FFFFh	2007h
-	-	-	-
-	-	-	-
-	-	-	-
9016h	0000h	0000h	3007h
9018h	0000h	0000h	3007h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BLS Rd, #imm8, r12

Instruction code	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][0 1 1 1 r11 to r8][r7 to r0] 20007000H
Argument	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of unsigned comparison is (Rd)≦#imm8, then (PC)←(PC)+4±(r12) If result of unsigned comparison is (Rd) > #imm8, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting immediate data designated by imm8 from the contents of the general-purpose register designated by Rd is nonpositive. If the result of the subtraction is positive, 4 is added to the PC.

The legitimate value range designated by Rd is from R0 to R7, that by imm8 is from 0 to FFh, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

		PC	R0	R1	R2	R3	PSW
		-	-	-	-	-	-
	MOV.W R0, #0x00FE	9002h	00FEh	-	-	-	0020h
	MOV.W R1, #0x0098	9004h	00FEh	0098h	-	-	1020h
	MOV.W R2, #0x00CD	9006h	00FEh	0098h	00CDh	-	2020h
	MOV.W R3, #0xFFFF	900Ah	00FEh	0098h	00CDh	FFFFh	3040h
loop:							
	BLS R0, #0xCD, LA ;; NOT JUMP LA	900Eh	00FEh	0098h	00CDh	FFFFh	0020h
	BLS R1, #0xCD, LB ;; JUMP LB	9018h	00FEh	0098h	00CDh	FFFFh	106Ch
	BR loop	-	-	-	-	-	-
LA:							
	DEC R3	-	-	-	-	-	-
	BR loop	-	-	-	-	-	-
LB:							
	INC R3	901Ah	00FEh	0098h	00CDh	0000h	300Fh
	NOP	901Ch	00FEh	0098h	00CDh	0000h	300Fh

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BLS Rd, Rs, r12

Instruction code	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][0 1 1 1 r11 to r8][r7 to r0] 0D007000H
Argument	Rd = 4bit(R select), Rs = 4bit(R select), r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of unsigned comparison is (Rd)≦(Rs), then (PC)←(PC)+4±(r12) If result of unsigned comparison is (Rd) > (Rs), then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting the contents of the general-purpose register designated by Rs from the contents of the general-purpose register designated by Rd is nonpositive. If the result of the subtraction is positive, 4 is added to the PC.

The legitimate value range designated by Rd is from R0 to R15, that by Rs is from R0 to R15, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0xFEDC
        MOV.W   R1, #0x9876
        MOV.W   R2, #0xCDEF
        MOV.W   R3, #0xFFFF
loop:
        BLS     R0, R2, LA      ;; NOT JUMP LA
        BLS     R1, R2, LB      ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9004h	FEDCh	-	-	-	0040h
9008h	FEDCh	9876h	-	-	1040h
900Ch	FEDCh	9876h	CDEFh	-	2040h
9010h	FEDCh	9876h	CDEFh	FFFFh	3040h
9014h	FEDCh	9876h	CDEFh	FFFFh	0008h
901Eh	FEDCh	9876h	CDEFh	FFFFh	104Ch
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	FEDCh	9876h	CDEFh	0000h	300Fh
9022h	FEDCh	9876h	CDEFh	0000h	300Fh

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BLS Rx, #imm16, r8

Instruction code	[1 1 0 0 0 1 1 1][r7r6r5r4r3r2r1r0][i15 to i8][i7 to i0] C7000000H
Argument	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)
Word count	2
Cycle count	3 or 4
Function	If result of unsigned comparison is (Rx)≦#imm16, then (PC)←(PC)+4±(r8) If result of unsigned comparison is (Rx) > #imm16, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S

[Description]

This instruction adds the value of the relative address designated by r8 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting immediate data designated by imm16 from the contents of the general-purpose register Rx designated indirectly by bits 12 to 15 (N0 to N3) of the PSW is nonpositive. If the result of the subtraction is positive, 4 is added to the PC.

The legitimate value range designated by imm16 is from 0 to FFFFh, and that by the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

		PC	R0	R1	R2	R3	PSW
		-	-	-	-	-	-
	MOV.W R2, #0xCDEF	9004h	-	-	CDEFh	-	2040h
	MOV.W R3, #0xFFFF	9008h	-	-	CDEFh	FFFFh	3040h
loop:							
	MOV.W R0, #0xFEDC	900Ch	FEDCh	-	CDEFh	FFFFh	0040h
	BLS Rx,#0xCDEF,LA ;; NOT JUMP LA	9010h	FEDCh	-	CDEFh	FFFFh	0008h
	MOV.W R1, #0x9876	9014h	FEDCh	9876h	CDEFh	FFFFh	1048h
	BLS Rx,#0xCDEF,LB ;; JUMP LB	901Eh	FEDCh	9876h	CDEFh	FFFFh	104Ch
	BR loop	-	-	-	-	-	-
LA:							
	DEC R3	-	-	-	-	-	-
	BR loop	-	-	-	-	-	-
LB:							
	INC R3	9020h	FEDCh	9876h	CDEFh	0000h	300Fh
	NOP	9022h	FEDCh	9876h	CDEFh	0000h	300Fh

<Note>

This instruction takes 4 cycles to execute if the conditions are met.

BLT r8

Instruction code	[1 1 0 1 0 0 1 0][r7r6r5r4r3r2r1r0] D200H
Argument	r8 = 8bit(relative address, signed)
Word count	1
Cycle count	2 or 3
Function	If S ^ OV = 1, then (PC)←(PC)+2±(r8) If S ^ OV = 0, then (PC)←(PC)+2
Affected flags	

[Description]

This instruction adds the value of the relative address designated by r8 + 2 to the program counter (PC) and places the result in the PC if the result of the exclusive logical OR of the sign flag (S) and the overflow flag (OV) is 1. If the result of the logical operation is 0, 2 is added to the PC.
The legitimate value range of the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

```
        MOV.W   R3, #0x0000
loop:
        BLT     LA          ;; NOT JUMP LA
        MOV.W   R3,#0xFFFF
        BLT     LB          ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R3	PSW
-	-	-
9002h	0000h	3003h
9004h	0000h	3003h
9008h	FFFFh	3040h
9010h	FFFFh	3040h
-	-	-
-	-	-
-	-	-
9012h	0000h	3003h
9014h	0000h	3003h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BLT Rd, #imm8, r12

Instruction code	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][0 0 1 0 r11 to r8][r7 to r0] 20002000H
Argument	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of signed comparison is (Rd) < #imm8, then (PC)←(PC)+4±(r12) If result of signed comparison is (Rd)≧#imm8, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting immediate data designated by imm8 from the contents (signed 16-bit data) of the general-purpose register designated by Rd is negative. If the result of the subtraction is nonnegative, 4 is added to the PC.

The legitimate value range designated by Rd is from R0 to R7, that by imm8 is from 0 to FFh, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0x0056
        MOV.W   R1, #0x0034
        MOV.W   R2, #0x0056
        MOV.W   R3, #0xFFFF
loop:
        BLT     R0, #0x56, LA   ;; NOT JUMP LA
        BLT     R1, #0x56, LB   ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9002h	0056h	-	-	-	0000h
9004h	0056h	0034h	-	-	1020h
9006h	0056h	0034h	0056h	-	2000h
900Ah	0056h	0034h	0056h	FFFFh	3040h
900Eh	0056h	0034h	0056h	FFFFh	0003h
9018h	0056h	0034h	0056h	FFFFh	104Ch
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
901Ah	0056h	0034h	0056h	0000h	300Fh
901Ch	0056h	0034h	0056h	0000h	300Fh

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BLT Rd, Rs, r12

Instruction code	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][0 0 1 0 r11 to r8][r7 to r0] 0D002000H
Argument	Rd = 4bit(R select), Rs = 4bit(R select), r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of signed comparison is (Rd) < (Rs), then (PC)←(PC)+4±(r12) If result of signed comparison is (Rd)≧(Rs), then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting the contents (signed 16-bit data) of the general-purpose register designated by Rs from the contents (signed 16-bit data) of the general-purpose register designated by Rd is negative. If the result of the subtraction is nonnegative, 4 is added to the PC.
The legitimate value range designated by Rd is from R0 to R15, that by Rs is from R0 to R15, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0x5678
        MOV.W   R1, #0xCDEF
        MOV.W   R2, #0x5678
        MOV.W   R3, #0xFFFF
loop:
        BLT     R0, R2, LA      ;; NOT JUMP LA
        BLT     R1, R2, LB      ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9004h	5678h	-	-	-	0000h
9008h	5678h	CDEFh	-	-	1040h
900Ch	5678h	CDEFh	5678h	-	2000h
9010h	5678h	CDEFh	5678h	FFFFh	3040h
9014h	5678h	CDEFh	5678h	FFFFh	0003h
901Eh	5678h	CDEFh	5678h	FFFFh	1010h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	5678h	CDEFh	5678h	0000h	3013h
9022h	5678h	CDEFh	5678h	0000h	3013h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BLT Rx, #imm16, r8

Instruction code	[1 1 0 0 0 0 1 0][r7r6r5r4r3r2r1r0][i15 to i8][i7 to i0] C2000000H
Argument	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)
Word count	2
Cycle count	3 or 4
Function	If result of signed comparison is (Rx) < #imm16, then (PC)←(PC)+4±(r8) If result of signed comparison is (Rx)≧#imm16, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S

[Description]

This instruction adds the value of the relative address designated by r8 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting immediate data designated by imm16 (signed 16-bit data) from the contents (signed 16-bit data) of the general-purpose register Rx designated indirectly by bits 12 to 15 (N0 to N3) of the PSW is negative. If the result of the subtraction is nonnegative, 4 is added to the PC.

The legitimate value range designated by imm16 is from 0 to FFFFh, and that by the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

		PC	R0	R1	R2	R3	PSW
		-	-	-	-	-	-
	MOV.W R2, #0xCDEF	9004h	-	-	CDEFh	-	2040h
	MOV.W R3, #0xFFFF	9008h	-	-	CDEFh	FFFFh	3040h
loop:							
	MOV.W R0, #0x5678	900Ch	5678h	-	CDEFh	FFFFh	0000h
	BLT Rx, #0x5678, LA ;; NOT JUMP LA	9010h	5678h	-	CDEFh	FFFFh	0003h
	MOV.W R1, #0xCDEF	9014h	5678h	CDEFh	CDEFh	FFFFh	1040h
	BLT Rx, #0x5678, LB ;; JUMP LB	901Eh	5678h	CDEFh	CDEFh	FFFFh	1010h
	BR loop	-	-	-	-	-	-
LA:							
	DEC R3	-	-	-	-	-	-
	BR loop	-	-	-	-	-	-
LB:							
	INC R3	9020h	5678h	CDEFh	CDEFh	0000h	3013h
	NOP	9022h	5678h	CDEFh	CDEFh	0000h	3013h

<Note>

This instruction takes 4 cycles to execute if the conditions are met.

BMI r8

Instruction code	[1 1 0 1 1 0 1 0][r7r6r5r4r3r2r1r0] DA00H
Argument	r8 = 8bit(relative address, signed)
Word count	1
Cycle count	2 or 3
Function	If S = 1, then (PC)←(PC)+2±(r8) If S = 0, then (PC)←(PC)+2
Affected flags	

[Description]

This instruction adds the value of the relative address designated by r8 + 2 to the program counter (PC) and places the result in the PC if the value of the sign flag (S) is 1. If the value of S is 0, 2 is added to the PC. The legitimate value range of the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

```
        MOV.W   R3, #0x000
loop:
        BMI     LA          ;; NOT JUMP LA
        MOV.W   R3,#0xFFFF
        BMI     LB          ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R3	PSW
-	-	-
9002h	0000h	3003h
9004h	0000h	3003h
9008h	FFFFh	3040h
9010h	FFFFh	3040h
-	-	-
-	-	-
-	-	-
9012h	0000h	3003h
9014h	0000h	3003h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BMI Rd, #imm8, r12

Instruction code	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][1 0 1 0 r11 to r8][r7 to r0] 2000A000H
Argument	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of (Rd) - #imm8 is S = 1, then (PC) ← (PC)+4±(r12) If result of (Rd) - #imm8 is S = 0, then (PC) ← (PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the sign flag (S) is set to 1 as the result of subtracting immediate data designated by imm8 from the contents of the general-purpose register designated by Rd. If S is set to 0 as the result of the subtraction, 4 is added to the PC.

The legitimate value range designated by Rd is from R0 to R7, that by imm8 is from 0 to FFh, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

			PC	R0	R1	R2	R3	PSW
			-	-	-	-	-	-
	MOV.W	R0, #0x9876	9004h	9876h	-	-	-	0040h
	MOV.W	R1, #0x5678	9008h	9876h	5678h	-	-	1000h
	MOV.W	R2, #0x0012	900Ah	9876h	5678h	0012h	-	2000h
	MOV.W	R3, #0xFFFF	900Eh	9876h	5678h	0012h	FFFFh	3040h
loop:								
	BMI	R1, #0x12, LA ;; NOT JUMP LA	9012h	9876h	5678h	0012h	FFFFh	1000h
	BMI	R0, #0x12, LB ;; JUMP LB	901Ch	9876h	5678h	0012h	FFFFh	0040h
	BR	loop	-	-	-	-	-	-
LA:								
	DEC	R3	-	-	-	-	-	-
	BR	loop	-	-	-	-	-	-
LB:								
	INC	R3	901Eh	9876h	5678h	0012h	0000h	3003h
	NOP		9020h	9876h	5678h	0012h	0000h	3003h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BMI Rd, Rs, r12

Instruction code	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][1 0 1 0 r11 to r8][r7 to r0] 0D00A000H
Argument	Rd = 4bit(R select), Rs = 4bit(R select), r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of (Rd) – (Rs) is S = 1, then (PC) ← (PC)+4±(r12) If result of (Rd) – (Rs) is S = 0, then (PC) ← (PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the sign flag (S) is set to 1 as the result of subtracting the contents of the general-purpose register designated by Rs from the contents of the general-purpose register designated by R. If S is set to 0 as the result of the subtraction, 4 is added to the PC.

The legitimate value range designated by Rd is from R0 to R15, that by Rs is from R0 to R15, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

			PC	R0	R1	R2	R3	PSW
			-	-	-	-	-	-
	MOV.W R0, #0x9876		9004h	9876h	-	-	-	0040h
	MOV.W R1, #0x5678		9008h	9876h	5678h	-	-	1000h
	MOV.W R2, #0x1234		900Ch	9876h	5678h	1234h	-	2020h
	MOV.W R3, #0xFFFF		9010h	9876h	5678h	1234h	FFFFh	3040h
loop:								
	BMI R1, R2, LA	;; NOT JUMP LA	9014h	9876h	5678h	1234h	FFFFh	1000h
	BMI R0, R2, LB	;; JUMP LB	901Eh	9876h	5678h	1234h	FFFFh	0060h
	BR loop		-	-	-	-	-	-
LA:								
	DEC R3		-	-	-	-	-	-
	BR loop		-	-	-	-	-	-
LB:								
	INC R3		9020h	9876h	5678h	1234h	0000h	3003h
	NOP		9022h	9876h	5678h	1234h	0000h	3003h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BMI Rx, #imm16, r8

Instruction code	[1 1 0 0 1 0 1 0][r7r6r5r4r3r2r1r0][i15 to i8][i7 to i0] CA000000H
Argument	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)
Word count	2
Cycle count	3 or 4
Function	If result of (Rx) – #imm16 is S = 1 , then (PC) ← (PC)+4±(r8) If result of (Rx) – #imm16 is S = 0 , then (PC) ← (PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S

[Description]

This instruction adds the value of the relative address designated by r8 + 4 to the program counter (PC) and places the result in the PC if the sign flag (S) is set to 1 as the result of subtracting immediate data designated by imm16 from the contents of the general-purpose register Rx designated indirectly by bits 12 to 15 (N0 to N3) of the PSW. If S is set to 0 as the result of the subtraction, 4 is added to the PC.
The legitimate value range designated by imm16 is from 0 to FFFFh, and that by the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

		PC	R0	R1	R2	R3	PSW
		-	-	-	-	-	-
	MOV.W R2, #0x1234	9004h	-	-	1234h	-	2020h
	MOV.W R3, #0xFFFF	9008h	-	-	1234h	FFFFh	3040h
loop:							
	MOV.W R1, #0x5678	900Ch	-	5678h	1234h	FFFFh	1000h
	BMI Rx, #0x1234, LA ;; NOT JUMP LA	9010h	-	5678h	1234h	FFFFh	1000h
	MOV.W R0, #0x9876	9014h	9876h	5678h	1234h	FFFFh	0040h
	BMI Rx, #0x1234, LB ;; JUMP LB	901Eh	9876h	5678h	1234h	FFFFh	0060h
	BR loop	-	-	-	-	-	-
LA:							
	DEC R3	-	-	-	-	-	-
	BR loop	-	-	-	-	-	-
LB:							
	INC R3	9020h	9876h	5678h	1234h	0000h	3003h
	NOP	9022h	9876h	5678h	1234h	0000h	3003h

<Note>

This instruction takes 4 cycles to execute if the conditions are met.

BN m16, #imm3, r12

Instruction code	[0 1 1 1 1 1 X 0][m7m6m5m4m3m2m1m0][0 i2i1i0 r11 to r8][r7 to r0] 7C00H(RAM), 7E00H(SFR)
Argument	m16 = 16bit(Lower 8bit valid for operation code), imm3 = 3bit(bit select) r12 = 12bit(relative address, signed)
Word count	2
Cycle count	3 or 4
Function	If (m16) of bit #imm3 = 0, then (PC)←(PC)+4±(r12) If (m16) of bit #imm3 = 1, then (PC)←(PC)+4
Affected flags	

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the bit designated by immediate data imm3 in the RAM (data memory) location or the SFR (one of the registers dedicated to control the internal peripheral functions) designated by m16 is 0. If the specified bit in the memory location m16 is 1, 4 is added to the PC.

The compiler generates the instruction code while regarding RAM or SFR as the destination of transfer according to the value of m16 (first operand data).

・ When specifying a RAM location, specify m16 with a value from 00H to FFH (0000H to 00FFH). It is disallowed to specify a RAM address not lower than 100H.

・ When specifying a SFR, specify m16 with a value from 7F00H to 7FFFH. The basic types of generated instruction code are 7C00H (RAM) and 7E00H (SFR), respectively, The lower-order 8 bits of m16 are reflected in the behavior of the instruction code.

The legitimate value range designated by imm3 is from 0 to 8h and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   0x50,#0x1221
        MOV.W   R3, #0xFFFF
loop:
        BN      0x50, #0, LA   ;; NOT JUMP LA
        BN      0x51, #0, LB   ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	RAM (51h)	RAM (50h)	R3
-	-	-	-
9004h	12h	21h	-
9008h	12h	21h	FFFFh
900Ch	12h	21h	FFFFh
9016h	12h	21h	FFFFh
-	-	-	-
-	-	-	-
-	-	-	-
9018h	12h	21h	0000h
901Ah	12h	21h	0000h

<Note>

This instruction takes 4 cycles to execute if the conditions are met.

BN Rd, #imm4, r12

Instruction code	[0 0 0 0 0 1 0 0][i3i2i1i0d3d2d1d0][0 0 0 0 r11 to r8][r7 to r0] 0400H
Argument	Rd = 4bit(R select),imm4 = 4bit(bit select),r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If (Rd) of bit #imm4 = 0, then (PC)←(PC)+4±(r12) If (Rd) of bit #imm4 = 1, then (PC)←(PC)+4
Affected flags	N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the bit of the general-purpose register designated by Rd designated by immediate data designated by imm4 is 0. If the specified bit of Rd is 1, 4 is added to the PC.
The legitimate value range designated by Rd is from R0 to R15, that by imm4 is from 0 to 0Fh, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0x0001
        MOV.W   R1, #0x1234
        MOV.W   R2, #0x0000
        MOV.W   R3, #0xFFFF
loop:
        BN      R0, #0, LA      ;; NOT JUMP LA
        BN      R1, #0, LB      ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9002h	0001h	-	-	-	0020h
9006h	0001h	1234h	-	-	1020h
9008h	0001h	1234h	0000h	-	2003h
900Ch	0001h	1234h	0000h	FFFFh	3040h
9010h	0001h	1234h	0000h	FFFFh	0040h
901Ah	0001h	1234h	0000h	FFFFh	1040h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
901Ch	0001h	1234h	0000h	0000h	3003h
901Eh	0001h	1234h	0000h	0000h	3003h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BN Rd, Rs, r12

Instruction code	[0 0 0 0 0 1 1 0][s3s2s1s0d3d2d1d0][0 0 0 0 r11 to r8][r7 to r0] 0600H
Argument	Rd = 4bit(R select), Rs = 4bit(bit select), r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If (Rd) of bit (Rs)&000Fh =0, then (PC)←(PC)+4±(r12) If (Rd) of bit (Rs)&000Fh =1, then (PC)←(PC)+4
Affected flags	N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the bit of the general-purpose register Rd that is designated by the lower-order 4 bits of the general-purpose register designated by Rs is 0. If the specified bit of Rd is 1, 4 is added to the PC. The legitimate value range designated by Rd is from R0 to R15, that by Rs from R0 to R15, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0x0001
        MOV.W   R1, #0x1234
        MOV.W   R2, #0x0000
        MOV.W   R3, #0xFFFF
loop:
        BN      R0, R2, LA      ;; NOT JUMP LA
        BN      R1, R2, LB      ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9002h	0001h	-	-	-	0020h
9006h	0001h	1234h	-	-	1020h
9008h	0001h	1234h	0000h	-	2003h
900Ch	0001h	1234h	0000h	FFFFh	3040h
9010h	0001h	1234h	0000h	FFFFh	0040h
901Ah	0001h	1234h	0000h	FFFFh	1040h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
901Ch	0001h	1234h	0000h	0000h	3003h
901Eh	0001h	1234h	0000h	0000h	3003h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BNC r8

Instruction code	[1 1 0 1 0 0 0 1][r7r6r5r4r3r2r1r0] D100H
Argument	r8 = 8bit(relative address, signed)
Word count	1
Cycle count	2 or 3
Function	If CY = 0, then (PC)←(PC)+2±(r8) If CY = 1, then (PC)←(PC)+2
Affected flags	

[Description]

This instruction adds the value of the relative address designated by r8 + 2 to the program counter (PC) and places the result in the PC if the value of the carry flag (CY) is 0. If the value of CY is 1, 2 is added to the PC.

The legitimate value range of the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

```
        MOV.W   R2, #0x0001
        MOV.W   R3, #0xFFFF
        RRC     R2, #1
loop:
        BNC     LA          ;; NOT JUMP LA
        RRC     R2, #1
        BNC     LB          ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R2	R3	PSW
-	-	-	-
9002h	0001h	-	2020h
9006h	0001h	FFFFh	3040h
9008h	0000h	FFFFh	2007h
900Ah	0000h	FFFFh	2007h
900Ch	8000h	FFFFh	2061h
9014h	8000h	FFFFh	2061h
-	-	-	-
-	-	-	-
-	-	-	-
9016h	8000h	0000h	3003h
9018h	8000h	0000h	3003h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BNC Rd, #imm8, r12

Instruction code	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][0 0 0 1 r11 to r8][r7 to r0] 20001000H
Argument	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of unsigned comparison is (Rd)≧#imm8, then (PC) ←(PC)+4±(r12) If result of unsigned comparison is (Rd) < #imm8, then (PC) ← (PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting immediate data designated by imm8 from the contents of the general-purpose register designated by Rd is nonnegative. If the result of the subtraction is negative, 4 is added to the PC.

The legitimate value range designated by Rd is from R0 to R7, that by imm8 is from 0 to FFh, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

			PC	R0	R1	R2	R3	PSW
			-	-	-	-	-	-
	MOV.W	R0, #0x0012	9002h	0012h	-	-	-	0000h
	MOV.W	R1, #0x00CD	9004h	0012h	00CDh	-	-	1020h
	MOV.W	R2, #0x0056	9006h	0012h	00CDh	0056h	-	2000h
	MOV.W	R3, #0xFFFF	900Ah	0012h	00CDh	0056h	FFFFh	3040h
loop:								
	BNC	R0, #0x56, LA ;; NOT JUMP LA	900Eh	0012h	00CDh	0056h	FFFFh	006Ch
	BNC	R1, #0x56, LB ;; JUMP LB	9018h	0012h	00CDh	0056h	FFFFh	1000h
	BR	loop	-	-	-	-	-	-
LA:								
	DEC	R3	-	-	-	-	-	-
	BR	loop	-	-	-	-	-	-
LB:								
	INC	R3	901Ah	0012h	00CDh	0056h	0000h	3003h
	NOP		901Ch	0012h	00CDh	0056h	0000h	3003h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BNC Rd, Rs, r12

Instruction code	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][0 0 0 1 r11 to r8][r7 to r0] 0D001000H
Argument	Rd = 4bit(R select), Rs = 4bit(R select), r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of unsigned comparison is (Rd)≧(Rs), then (PC) ←(PC)+4±(r12) If result of unsigned comparison is (Rd) < (Rs), then (PC) ← (PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting the contents of the general-purpose register designated by Rs from the contents of the general-purpose register designated by Rd is nonnegative. If the result of the subtraction is negative, 4 is added to the PC.

The legitimate value range designated by Rd is from R0 to R15, that by Rs is from R0 to R15, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0x1234
        MOV.W   R1, #0xCDEF
        MOV.W   R2, #0x5678
        MOV.W   R3, #0xFFFF
loop:
        BNC     R0, R2, LA      ;; NOT JUMP LA
        BNC     R1, R2, LB      ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9004h	1234h	-	-	-	0020h
9008h	1234h	CDEFh	-	-	1040h
900Ch	1234h	CDEFh	5678h	-	2000h
9010h	1234h	CDEFh	5678h	FFFFh	3040h
9014h	1234h	CDEFh	5678h	FFFFh	006Ch
901Eh	1234h	CDEFh	5678h	FFFFh	1010h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	1234h	CDEFh	5678h	0000h	3013h
9022h	1234h	CDEFh	5678h	0000h	3013h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BNC Rx, #imm16, r8

Instruction code	[1 1 0 0 0 0 0 1][r7r6r5r4r3r2r1r0][i15 to i8][i7 to i0] C1000000H
Argument	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)
Word count	2
Cycle count	3 or 4
Function	If result of unsigned comparison is (Rx)≧#imm16, then (PC) ←(PC)+4±(r8) If result of unsigned comparison is (Rx) < #imm16, then (PC) ←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S

[Description]

This instruction adds the value of the relative address designated by r8 + 4 to the program counter (PC) and places the result in the PC if the result of subtracting immediate data designated by imm16 from the contents of the general-purpose register Rx designated indirectly by bits 12 to 15 (N0 to N3) of the PSW is nonnegative. If the result of the subtraction is negative, 4 is added to the PC.

The legitimate value range designated by imm16 is from 0 to FFFFh, and that by the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

		PC	R0	R1	R2	R3	PSW
		-	-	-	-	-	-
	MOV.W R2, #0x5678	9004h	-	-	5678h	-	2000h
	MOV.W R3, #0xFFFF	9008h	-	-	5678h	FFFFh	3040h
loop:							
	MOV.W R0, #0x1234	900Ch	1234h	-	5678h	FFFFh	0020h
	BNC Rx, #0x5678, LA ;; NOT JUMP LA	9010h	1234h	-	5678h	FFFFh	006Ch
	MOV.W R1, #0xCDEF	9014h	1234h	CDEFh	5678h	FFFFh	104Ch
	BNC Rx, #0x5678, LB ;; JUMP LB	901Eh	1234h	CDEFh	5678h	FFFFh	1010h
	BR loop	-	-	-	-	-	-
LA:							
	DEC R3	-	-	-	-	-	-
	BR loop	-	-	-	-	-	-
LB:							
	INC R3	9020h	1234h	CDEFh	5678h	0000h	3013h
	NOP	9022h	1234h	CDEFh	5678h	0000h	3013h

<Note>

This instruction takes 4 cycles to execute if the conditions are met.

BNV r8

Instruction code	[1 1 0 1 1 0 0 1][r7r6r5r4r3r2r1r0] D900H
Argument	r8 = 8bit(relative address, signed)
Word count	1
Cycle count	2 or 3
Function	If OV = 0 then (PC)←(PC)+2±(r8) If OV = 1 then (PC)←(PC)+2
Affected flags	

[Description]

This instruction adds the value of the relative address designated by r8 + 2 to the program counter (PC) and places the result in the PC if the value of the overflow flag (OV) is 0. If the value of OV is 1, 2 is added to the PC.

The legitimate value range of the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

```
        MOV.W   R2, #0x789A
        MOV.W   R3, #0xFFFF
loop:
        ADD     R2, #0x2345
        BNV     LA          ;; NOT JUMP LA
        ADD     R2, #0x2345
        BNV     LB          ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R2	R3	PSW
-	-	-	-
9004h	789Ah	-	2000h
9008h	789Ah	FFFFh	3040h
900Ch	9BDFh	FFFFh	2050h
900Eh	9BDFh	FFFFh	2050h
9012h	BF24h	FFFFh	2068h
901Ah	BF24h	FFFFh	2068h
-	-	-	-
-	-	-	-
-	-	-	-
901Ch	BF24h	0000h	300Bh
901Eh	BF24h	0000h	300Bh

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BNV Rd, #imm8, r12

Instruction code	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][1 0 0 1 r11 to r8][r7 to r0] 20009000H
Argument	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of (Rd) - #imm8 is OV=0, then (PC)←(PC)+4±(r12) If result of (Rd) - #imm8 is OV=1, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the OV flag (OV) is set to 0 as the result of subtracting immediate data designated by imm8 from the contents of the general-purpose register designated by Rd. If OV is set to 1 as the result of the subtraction, 4 is added to the PC.

The legitimate value range designated by Rd is from R0 to R7, that by imm8 is from 0 to FFh, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

			PC	R0	R1	R2	R3	PSW
			-	-	-	-	-	-
	MOV.W	R0, #0x8000	9004h	8000h	-	-	-	0061h
	MOV.W	R1, #0x5678	9008h	8000h	5678h	-	-	1000h
	MOV.W	R2, #0x0012	900Ah	8000h	5678h	0012h	-	2000h
	MOV.W	R3, #0xFFFF	900Eh	8000h	5678h	0012h	FFFFh	3040h
loop:								
	BNV	R0, #0x12, LA ;; NOT JUMP LA	9012h	8000h	5678h	0012h	FFFFh	0038h
	BNV	R1, #0x12, LB ;; JUMP LB	901Ch	8000h	5678h	0012h	FFFFh	1000h
	BR	loop	-	-	-	-	-	-
LA:								
	DEC	R3	-	-	-	-	-	-
	BR	loop	-	-	-	-	-	-
LB:								
	INC	R3	901Eh	8000h	5678h	0012h	0000h	3003h
	NOP		9020h	8000h	5678h	0012h	0000h	3003h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BNV Rd, Rs, r12

Instruction code	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][1 0 0 1 r11 to r8][r7 to r0] 0D009000H
Argument	Rd = 4bit(R select), Rs = 4bit(R select), r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of (Rd) – (Rs) is OV=0, then (PC)←(PC)+4±(r12) If result of (Rd) – (Rs) is OV=1, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the value of the overflow flag (OV) is set to 0 as the result of subtracting the contents of the general-purpose register designated by Rs from the contents of the general-purpose register designated by Rd. If OV is set to 1 as the result of the subtraction, 4 is added to the PC.
The legitimate value range designated by Rd is from R0 to R15, that by Rs is from R0 to R15, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0x89AB
        MOV.W   R1, #0x5678
        MOV.W   R2, #0x1234
        MOV.W   R3, #0xFFFF
loop:
        BNV     R0, R2, LA      ;; NOT JUMP LA
        BNV     R1, R2, LB      ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9004h	89ABh	-	-	-	0040h
9008h	89ABh	5678h	-	-	1000h
900Ch	89ABh	5678h	1234h	-	2020h
9010h	89ABh	5678h	1234h	FFFFh	3040h
9014h	89ABh	5678h	1234h	FFFFh	0010h
901Eh	89ABh	5678h	1234h	FFFFh	1000h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	89ABh	5678h	1234h	0000h	3003h
9022h	89ABh	5678h	1234h	0000h	3003h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BNV Rx, #imm16, r8

Instruction code	[1 1 0 0 1 0 0 1][r7r6r5r4r3r2r1r0][i15 to i8][i7 to i0] C9000000H
Argument	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)
Word count	2
Cycle count	3 or 4
Function	If result of (Rx) - #imm16 is OV=0, then (PC)←(PC)+4±(r8) If result of (Rx) - #imm16 is OV=1, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S

[Description]

This instruction adds the value of the relative address designated by r8 + 4 to the program counter (PC) and places the result in the PC if the value of the overflow flag (OV) is set to 0 as result of subtracting immediate data designated by imm16 from the contents of the general-purpose register Rx designated indirectly by bits 12 to 15 (N0 to N3) of the PSW. If OV is set to 1 as the result of the subtraction, 4 is added to the PC.
The legitimate value range designated by imm16 is from 0 to FFFFh, and that by the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

		PC	R0	R1	R2	R3	PSW
		-	-	-	-	-	-
	MOV.W R2, #0x1234	9004h	-	-	1234h	-	2020h
	MOV.W R3, #0xFFFF	9008h	-	-	1234h	FFFFh	3040h
loop:							
	MOV.W R0, #0x8000	900Ch	8000h	-	1234h	FFFFh	0061h
	BNV Rx, #0x1234, LA ;; NOT JUMP LA	9010h	8000h	-	1234h	FFFFh	0038h
	MOV.W R1, #0x5678	9014h	8000h	5678h	1234h	FFFFh	1018h
	BNV Rx, #0x1234, LB ;; JUMP LB	901Eh	8000h	5678h	1234h	FFFFh	1000h
	BR Loop	-	-	-	-	-	-
LA:							
	DEC R3	-	-	-	-	-	-
	BR Loop	-	-	-	-	-	-
LB:							
	INC R3	9020h	8000h	5678h	1234h	0000h	3003h
	NOP	9022h	8000h	5678h	1234h	0000h	3003h

<Note>

This instruction takes 4 cycles to execute if the conditions are met.

BNZ r8

Instruction code	[1 1 0 1 1 1 0 1][r7r6r5r4r3r2r1r0] DD00H
Argument	r8 = 8bit(relative address, signed)
Word count	1
Cycle count	2 or 3
Function	If Z16 = 0, then (PC)←(PC)+2±(r8) If Z16 = 1, then (PC)←(PC)+2
Affected flags	

[Description]

This instruction adds the value of the relative address designated by r8 + 2 to the program counter (PC) and places the result in the PC if the value of the16-bit operation flag (Z16) is 0. If the value of Z16 is 1, 2 is added to the PC.

The legitimate value range of the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

```
        MOV.W   R3,
                #0x0000
loop:
        BNZ     LA          ;; NOT JUMP LA
        MOV.W   R3,#0x1234
        BNZ     LB          ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R3	PSW
-	-	-
9002h	0000h	3003h
9004h	0000h	3003h
9008h	1234h	3020h
9010h	1234h	3020h
-	-	-
-	-	-
-	-	-
9012h	1235h	3000h
9014h	1235h	3000h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BNZ Rd, #imm8, r12

Instruction code	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][1 1 0 1 r11 to r8][r7 to r0] 2000D000H
Argument	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of (Rd) - #imm8 is Z16 = 0, then (PC)←(PC)+4±(r12) If result of (Rd) - #imm8 is Z16 = 1, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the 16-bit operation flag (Z16) is set to 0 as the result of subtracting immediate data designated by imm8 from the contents of the general-purpose register designated by Rd. If Z16 is set to 1 as the result of the subtraction, 4 is added to the PC.

The legitimate value range designated by Rd is from R0 to R7, that by imm8 is from 0 to FFh, and that of the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0x0034
        MOV.W   R1, #0x8234
        MOV.W   R2, #0x0034
        MOV.W   R3, #0xFFFF
loop:
        BNZ     R0, #0x34,  ;; NOT JUMP LA
        BNZ     R1, #0x34,  ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9002h	0034h	-	-	-	0020h
9006h	0034h	8234h	-	-	1060h
9008h	0034h	8234h	0034h	-	2020h
900Ch	0034h	8234h	0034h	FFFFh	3040h
9010h	0034h	8234h	0034h	FFFFh	0003h
901Ah	0034h	8234h	0034h	FFFFh	1041h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
901Ch	0034h	8234h	0034h	0000h	3003h
901Eh	0034h	8234h	0034h	0000h	3003h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BNZ Rd, Rs, r12

Instruction code	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][1 1 0 1 r11 to r8][r7 to r0] 0D00D000H
Argument	Rd = 4bit(R select), Rs = 4bit(R select), r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of (Rd) - (Rs) is Z16 = 0, then (PC)←(PC)+4±(r12) If result of (Rd) - (Rs) is Z16 = 1, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the 16-bit operation flag (Z16) is set to 0 as the result of subtracting the contents of the general-purpose register designated by Rs from the contents of the general-purpose register designated by Rd. If Z16 is set to 1 as the result of the subtraction, 4 is added to the PC.
The legitimate value range designated by Rd is from R0 to R15, that by Rs is from R0 to R15, and that of the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0x1234
        MOV.W   R1, #0x8234
        MOV.W   R2, #0x1234
        MOV.W   R3, #0xFFFF
loop:
        BNZ     R0, R2, LA      ;; NOT JUMP LA
        BNZ     R1, R2, LB      ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9004h	1234h	-	-	-	0020h
9008h	1234h	8234h	-	-	1060h
900Ch	1234h	8234h	1234h	-	2020h
9010h	1234h	8234h	1234h	FFFFh	3040h
9014h	1234h	8234h	1234h	FFFFh	0003h
901Eh	1234h	8234h	1234h	FFFFh	1031h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	1234h	8234h	1234h	0000h	3013h
9022h	1234h	8234h	1234h	0000h	3013h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BNZ Rx, #imm16, r8

Instruction code	[1 1 0 0 1 1 0 1][r7r6r5r4r3r2r1r0][i15 to i8][i7 to i0] CD000000H
Argument	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)
Word count	2
Cycle count	3 or 4
Function	If result of (Rx) - #imm16 is Z16 =0, then (PC)←(PC)+4±(r8) If result of (Rx) - #imm16 is Z16 =1, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S

[Description]

This instruction adds the value of the relative address designated by r8 + 4 to the program counter (PC) and places the result in the PC if the 16-bit operation flag (Z16) is set to 0 as the result of subtracting immediate data designated by imm16 from the contents of the general-purpose register Rx designated indirectly by bits 12 to 15 (N0 to N3) of the PSW. If Z16 is set to 1 as the result of the subtraction, 4 is added to the PC.
The legitimate value range designated by imm16 is from 0 to FFFFh, and that of the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

		PC	R0	R1	R2	R3	PSW
		-	-	-	-	-	-
MOV.W	R2, #0x1234	9004h	-	-	1234h	-	2020h
MOV.W	R3, #0xFFFF	9008h	-	-	1234h	FFFFh	3040h
loop:							
MOV.W	R0, #0x1234	900Ch	1234h	-	1234h	FFFFh	0020h
BNZ	Rx, #0x1234, LA ;; NOT JUMP LA	9010h	1234h	-	1234h	FFFFh	0003h
MOV.W	R1, #0x8234	9014h	1234h	8234h	1234h	FFFFh	1060h
BNZ	Rx, #0x1234, LB ;; JUMP LB	901Eh	1234h	8234h	1234h	FFFFh	1031h
BR	loop	-	-	-	-	-	-
LA:							
DEC	R3	-	-	-	-	-	-
BR	loop	-	-	-	-	-	-
LB:							
INC	R3	9020h	1234h	8234h	1234h	0000h	3013h
NOP		9020h	1234h	8234h	1234h	0000h	3013h

<Note>

This instruction takes 4 cycles to execute if the conditions are met.

BNZ. B r8

Instruction code	[1 1 0 1 1 1 0 0][r7r6r5r4r3r2r1r0] DC00H
Argument	r8 = 8bit(relative address, signed)
Word count	1
Cycle count	2 or 3
Function	If Z8 = 0, then (PC)←(PC)+2±(r8) If Z8 = 1, then (PC)←(PC)+2
Affected flags	

[Description]

This instruction adds the value of the relative address designated by r8 + 2 to the program counter (PC) and places the result in the PC if the value of the 8-bit operation flag (Z8) is 0. If the value of Z8 is 1, 2 is added to the PC.

The legitimate value range of the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

```
        MOV.W   R3, #0x1200
loop:
        BNZ.B   LA          ;;NOT JUMP LA
        MOV.W   R3, #0x1234
        BNZ.B   LB          ;;JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R3	PSW
-	-	-
9004h	1200h	3001h
9006h	1200h	3001h
900Ah	1234h	3020h
9012h	1234h	3020h
-	-	-
-	-	-
-	-	-
9014h	1235h	3000h
9016h	1235h	3000h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BNZ. B Rd, #imm8, r12

Instruction code	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][1 1 0 0 r11 to r8][r7 to r0] 2000C000H
Argument	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of (Rd) - #imm8 is Z8 = 0, then (PC)←(PC)+4±(r12) If result of (Rd) - #imm8 is Z8 = 1, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the 8-bit operation flag (Z8) is set to 0 as the result of subtracting immediate data designated by imm8 from the contents of the general-purpose register designated by Rd. If Z8 is set to 1 as the result of the subtraction, 4 is added to the PC.

The legitimate value range designated by Rd is from R0 to R7, that by imm8 is from 0 to FFh, and that of the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0x5634
        MOV.W   R1, #0x8000
        MOV.W   R2, #0x1234
        MOV.W   R3, #0xFFFF
loop:
        BNZ.B   R0, #0x34, LA   ;; NOT JUMP LA
        BNZ.B   R1, #0x34, LB   ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9004h	5634h	-	-	-	0020h
9008h	5634h	8000h	-	-	1061h
900Ch	5634h	8000h	1234h	-	2020h
9010h	5634h	8000h	1234h	FFFFh	3040h
9014h	5634h	8000h	1234h	FFFFh	0001h
901Eh	5634h	8000h	1234h	FFFFh	1038h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	5634h	8000h	1234h	0000h	301Bh
9022h	5634h	8000h	1234h	0000h	301Bh

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BNZ. B Rd, Rs, r12

Instruction code	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][1 1 0 0 r11 to r8][r7 to r0] 0D00C000H
Argument	Rd = 4bit(R select), Rs = 4bit(R select), r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of (Rd) – (Rs) is Z8 = 0, then (PC)←(PC)+4±(r12) If result of (Rd) – (Rs) is Z8 = 1, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the 8-bit operation flag (Z8) is set to 0 as the result of subtracting the contents of the general-purpose register designated by Rs from the contents of the general-purpose register designated by Rd. If Z8 is set to 1 as the result of the subtraction, 4 is added to the PC.
The legitimate value range designated by Rd is from R0 to R15, that by Rs is from R0 to R15, and that of the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0x5634
        MOV.W   R1, #0x8000
        MOV.W   R2, #0x1234
        MOV.W   R3, #0xFFFF
loop:
        BNZ.B   R0, R2, LA      ;; NOT JUMP LA
        BNZ.B   R1, R2, LB      ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9004h	5634h	-	-	-	0020h
9008h	5634h	8000h	-	-	1061h
900Ch	5634h	8000h	1234h	-	2020h
9010h	5634h	8000h	1234h	FFFFh	3040h
9014h	5634h	8000h	1234h	FFFFh	0001h
901Eh	5634h	8000h	1234h	FFFFh	1038h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	5634h	8000h	1234h	0000h	301Bh
9022h	5634h	8000h	1234h	0000h	301Bh

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BNZ. B Rx, #imm16, r8

Instruction code	[1 1 0 0 1 1 0 0][r7r6r5r4r3r2r1r0][i15 to i8][i7 to i0] CC000000H
Argument	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)
Word count	2
Cycle count	3 or 4
Function	If result of (Rx) – #imm16 is Z8 = 0, then (PC)←(PC)+4±(r8) If result of (Rx) – #imm16 is Z8 = 1, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S

[Description]

This instruction adds the value of the relative address designated by r8 + 4 to the program counter (PC) and places the result in the PC if the 8-bit operation flag (Z8) is set to 0 as the result of subtracting immediate data designated by imm16 from the contents of the general-purpose register Rx designated indirectly by bits 12 to 15 (N0 to N3) of the PSW. If Z8 is set to 1 as the result of the subtraction, 4 is added to the PC.

The legitimate value range designated by imm16 is from 0 to FFFFh, and that of the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

			PC	R0	R1	R2	R3	PSW
			-	-	-	-	-	-
	MOV.W	R2, #0x1234	9004h	-	-	1234h	-	2020h
	MOV.W	R3, #0xFFFF	9008h	-	-	1234h	FFFFh	3040h
loop:								
	MOV.W	R0, #0x5634	900Ch	5634h	-	1234h	FFFFh	0020h
	BNZ.B	Rx, #0x1234, LA ;; NOT JUMP LA	9010h	5634h	-	1234h	FFFFh	0001h
	MOV.W	R1, #0x8000	9014h	5634h	8000h	1234h	FFFFh	1061h
	BNZ.B	Rx, #0x1234, LB ;; JUMP LB	901Eh	5634h	8000h	1234h	FFFFh	1038h
	BR	loop	-	-	-	-	-	-
LA:								
	DEC	R3	-	-	-	-	-	-
	BR	loop	-	-	-	-	-	-
LB:								
	INC	R3	9020h	5634h	8000h	1234h	0000h	301Bh
	NOP		9022h	5634h	8000h	1234h	0000h	301Bh

<Note>

This instruction takes 4 cycles to execute if the conditions are met.

BP m16, #imm3, r12

Instruction code	[0 1 1 1 1 1 X 1][m7m6m5m4m3m2m1m0][0 i2i1i0 r11 to r8][r7 to r0] 7D00H(RAM), 7F00H(SFR)
Argument	m16 = 16bit(Lower 8bit valid for operation code), imm3 = 3bit(bit select) r12 = 12bit(relative address, signed)
Word count	2
Cycle count	3 or 4
Function	if (m16) of bit #imm3 = 1, then (PC)←(PC)+4±(r12) if (m16) of bit #imm3 = 0, then (PC)←(PC)+4
Affected flags	

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC, if the bit indicated by immediate data designated by imm3 in the data memory location designated by m16, is 1. If the bit designated by m16 is 0, 4 is added to the PC.

The compiler generates the instruction code while regarding RAM or SFR as the destination of transfer according to the value of m16 (first operand data).

・ When specifying a RAM location, specify m16 with a value from 00H to FFH (0000H to 00FFH). It is disallowed to specify a RAM address not lower than 100H.

・ When specifying a SFR, specify m16 with a value from 7F00H to 7FFFH.

The basic types of generated instruction code are 7D00H (RAM) and 7F00H (SFR), respectively, The lower-order 8 bits of m16 are reflected in the behavior of the instruction code.

The legitimate value range designated by imm3 is from 0 to 8h and that of the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   0x50, #01221
        MOV.W   R3, #0FFFF
loop:
        BP      0x51, #0, LA   ;; NOT JUMP LA
        BP      0x50, #0, LB   ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	RAM (51h)	RAM (50h)	R3
-	-	-	-
9004h	12h	21h	-
9008h	12h	21h	FFFFh
900Ch	12h	21h	FFFFh
9016h	12h	21h	FFFFh
-	-	-	-
-	-	-	-
-	-	-	-
9018h	12h	21h	0000h
901Ah	12h	21h	0000h

<Note>

This instruction takes 4 cycles to execute if the conditions are met.

BP Rd, #imm4, r12

Instruction code	[0 0 0 0 0 1 0 1][i3i2i1i0d3d2d1d0][0 0 0 0 r11 to r8][r7 to r0] 0500H
Argument	Rd = 4bit(R select), imm4 = 4bit(bit select), r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If (Rd) of bit #imm4 = 1, then (PC)←(PC)+4±(r12) If (Rd) of bit #imm4 =0, then (PC)←(PC)+4
Affected flags	N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the bit, in the contents of the general-purpose register designated by Rd, that is specified by immediate data designated by imm4 is 1. If the bit specified in the contents of Rd is 0, 4 is added to the PC.

The legitimate value range designated by Rd is from R0 to R15, that by imm4 is from 0 to 0Fh, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0x0001
        MOV.W   R1, #0x1234
        MOV.W   R2, #0x0000
        MOV.W   R3, #0xFFFF
loop:
        BP      R1, #0, LA      ;; NOT JUMP LA
        BP      R0, #0, LB      ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9002h	0001h	-	-	-	0020h
9006h	0001h	1234h	-	-	1020h
9008h	0001h	1234h	0000h	-	2003h
900Ch	0001h	1234h	0000h	FFFFh	3040h
9010h	0001h	1234h	0000h	FFFFh	1040h
901Ah	0001h	1234h	0000h	FFFFh	0040h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
901Ch	0001h	1234h	0000h	0000h	3003h
901Eh	0001h	1234h	0000h	0000h	3003h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BP Rd, Rs, r12

Instruction code	[0 0 0 0 0 1 1 1][s3s2s1s0d3d2d1d0][0 0 0 0 r11 to r8][r7 to r0] 0700H
Argument	Rd = 4bit(R select), Rs = 4bit(bit select), r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If (Rd) of bit (Rs)&000Fh =1, then (PC)←(PC)+4±(r12) If (Rd) of bit (Rs)&000Fh =0, then (PC)←(PC)+4
Affected flags	N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the bit, in the contents of the general-purpose register designated by Rd, that is is specified by the lower-order 4 bits of the general-purpose register designated by Rs is 1. If the specified bit of Rd is 0, 4 is added to the PC.

The legitimate value range designated by Rd is from R0 to R15, that by Rs is from R0 to R15, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0x0001
        MOV.W   R1, #0x1234
        MOV.W   R2, #0x0000
        MOV.W   R3, #0xFFFF
loop:
        BP      R1, R2, LA      ;; NOT JUMP LA
        BP      R0, R2, LB      ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9002h	0001h	-	-	-	0020h
9006h	0001h	1234h	-	-	1020h
9008h	0001h	1234h	0000h	-	2003h
900Ch	0001h	1234h	0000h	FFFFh	3040h
9010h	0001h	1234h	0000h	FFFFh	1040h
901Ah	0001h	1234h	0000h	FFFFh	0040h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
901Ch	0001h	1234h	0000h	0000h	3003h
901Eh	0001h	1234h	0000h	0000h	3003h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BPL r8

Instruction code	[1 1 0 1 1 0 0 0][r7r6r5r4r3r2r1r0] D800H
Argument	r8 = 8bit(relative address, signed)
Word count	1
Cycle count	2 or 3
Function	If S = 0, then (PC)←(PC)+2±(r8) If S = 1, then (PC)←(PC)+2
Affected flags	

[Description]

This instruction adds the value of the relative address designated by r8 + 2 to the program counter (PC) and places the result in the PC if the value of the sign flag (S) is 0. If the value of S is 1, 2 is added to the PC. The legitimate value range of the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

```
        MOV.W   R3,
                #0xFFFF
loop:
        BPL     LA          ;; NOT JUMP LA
        MOV.W   R3, #0x0000
        BPL     LB          ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R3	PSW
-	-	-
9004h	FFFFh	3040h
9006h	FFFFh	3040h
9008h	0000h	3003h
9010h	0000h	3003h
-	-	-
-	-	-
-	-	-
9012h	0001h	3020h
9014h	0001h	3020h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BPL Rd, #imm8, r12

Instruction code	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][1 0 0 0 r11 to r8][r7 to r0] 20008000H
Argument	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of (Rd) - #imm8 is S=0, then (PC)←(PC)+4±(r12) If result of (Rd) - #imm8 is S=1, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the sign flag (S) is set to 0 as the result of subtracting immediate data designated by imm8 from the contents of the general-purpose register designated by Rd. If S is set to 1 as the result of the subtraction, 4 is added to the PC.

The legitimate value range designated by Rd is from R0 to R7, that by imm8 is from 0 to FFh, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

			PC	R0	R1	R2	R3	PSW
			-	-	-	-	-	-
	MOV.W	R0, #0x9876	9004h	9876h	-	-	-	0040h
	MOV.W	R1, #0x5678	9008h	9876h	5678h	-	-	1000h
	MOV.W	R2, #0x0012	900Ah	9876h	5678h	0012h	-	2000h
	MOV.W	R3, #0xFFFF	900Eh	9876h	5678h	0012h	FFFFh	3040h
loop:								
	BPL	R0, #0x12, LA ;; NOT JUMP LA	9012h	9876h	5678h	0012h	FFFFh	0040h
	BPL	R1, #0x12, LB ;; JUMP LB	901Ch	9876h	5678h	0012h	FFFFh	1000h
	BR	loop	-	-	-	-	-	-
LA:								
	DEC	R3	-	-	-	-	-	-
	BR	loop	-	-	-	-	-	-
LB:								
	INC	R3	901Eh	9876h	5678h	0012h	0000h	3003h
	NOP		9020h	9876h	5678h	0012h	0000h	3003h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BPL Rd, Rs, r12

Instruction code	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][1 0 0 0 r11 to r8][r7 to r0] 0D008000H
Argument	Rd = 4bit(R select), Rs = 4bit(R select), r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of (Rd) - (Rs) is S=0, then (PC)←(PC)+4±(r12) If result of (Rd) - (Rs) is S=1, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the sign flag (S) is set to 0 as the result of subtracting the contents of the general-purpose register designated by Rs from the contents of the general-purpose register designated by Rd. If S is set to 1 as the result of the subtraction, 4 is added to the PC.
The legitimate value range designated by Rd is from R0 to R15, that by Rs is from R0 to R15, and that by the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0x9876
        MOV.W   R1, #0x5678
        MOV.W   R2, #0x1234
        MOV.W   R3, #0xFFFF
loop:
        BPL     R0, R2, LA      ;; NOT JUMP LA
        BPL     R1, R2, LB      ;; JUMP LB
        BR      Loop
LA:
        DEC     R3
        BR      Loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9004h	9876h	-	-	-	0040h
9008h	9876h	5678h	-	-	1000h
900Ch	9876h	5678h	1234h	-	2020h
9010h	9876h	5678h	1234h	FFFFh	3040h
9014h	9876h	5678h	1234h	FFFFh	0060h
901Eh	9876h	5678h	1234h	FFFFh	1000h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	9876h	5678h	1234h	0000h	3003h
9022h	9876h	5678h	1234h	0000h	3003h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BPL Rx, #imm16, r8

Instruction code	[1 1 0 0 1 0 0 0][r7r6r5r4r3r2r1r0][i15 to i8][i7 to i0] C8000000H
Argument	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)
Word count	2
Cycle count	3 or 4
Function	If result of (Rx) - #imm16 is S=0, then (PC)←(PC)+4±(r8) If result of (Rx) - #imm16 is S=1, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S

[Description]

This instruction adds the value of the relative address designated by r8 + 4 to the program counter (PC) and places the result in the PC if the sign flag (S) is set to 0 as the result of subtracting immediate data designated by imm16 from the contents of the general-purpose register Rx designated indirectly by the value of bits 12 to 15 (N0 to N3) of the PSW. If S is set to 1 as the result of the subtraction, 4 is added to the PC.
The legitimate value range designated by imm16 is from 0 to FFFFh, and that by the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

			PC	R0	R1	R2	R3	PSW
			-	-	-	-	-	-
	MOV.W	R2, #0x1234	9004h	-	-	1234h	-	2020h
	MOV.W	R3, #0xFFFF	9008h	-	-	1234h	FFFFh	3040h
loop:								
	MOV.W	R0, #0x9876	900Ch	9876h	-	1234h	FFFFh	0040h
	BPL	Rx, #0x1234, LA ;; NOT JUMP LA	9010h	9876h	-	1234h	FFFFh	0060h
	MOV.W	R1, #0x5678	9014h	9876h	5678h	1234h	FFFFh	1000h
	BPL	Rx, #0x1234, LB ;; JUMP LB	901Eh	9876h	5678h	1234h	FFFFh	1000h
	BR	loop	-	-	-	-	-	-
LA:								
	DEC	R3	-	-	-	-	-	-
	BR	loop	-	-	-	-	-	-
LB:								
	INC	R3	9020h	9876h	5678h	1234h	0000h	3003h
	NOP		9022h	9876h	5678h	1234h	0000h	3003h

<Note>

This instruction takes 4 cycles to execute if the conditions are met.

BR r12

Instruction code	[0 0 0 1 r11r10r9r8][r7r6r5r4r3r2r1 0] 1000H
Argument	r12 = 12bit(relative address, signed)
Word count	1
Cycle count	2
Function	(PC)←(PC)+2±(r12)
Affected flags	

[Description]

This instruction adds the value of the relative address designated by r12 + 2 to the program counter (PC) and places the result in the PC.

The legitimate value range of the relative address designated by r12 is that of signed 12-bit data (-2078 to 2047).

[Example] The value of label LA is 9106H.

```
        MOV.W   R3, #0x0200
loop:
        BR      LA          ;; JUMP LA
        NOP
        NOP
          .
          .
          .
LA:
        INC     R3
        NOP
```

PC	R3	PSW
-	-	-
9004h	0200h	3021h
9106h	0200h	3021h
-	-	-
-	-	-
9108h	0201h	3000h
910Ah	0201h	3000h

BR Rs

Instruction code	[0 0 0 0 0 0 0 0][0 0 1 0 s3s2s1s0] 0020H
Argument	Rs = 4bit(relative address, signed)
Word count	1
Cycle count	2
Function	(PC)←(PC)+2±(Rs)
Affected flags	

[Description]

This instruction adds the value of the relative address (the contents of the general-purpose register designated by Rs) + 2 to the program counter (PC) and places the result in the PC

The legitimate value range odesignated by Rs is from R0 to R15, and that by the relative address (the contents of the general-purpose register designated by Rs) is that of signed 16-bit data (-32768 to 32767).

[Example] The value of label LA is 9106H.

```
        MOV.W   R3, #0x0100
loop:
        BR      R3          ;; JUMP LA
        NOP
        NOP
          .
          .
          .
LA:
        INC     R3
        NOP
```

PC	R3	PSW
-	-	-
9004h	0100h	3021h
9106h	0100h	3021h
-	-	-
-	-	-
9108h	0101h	3000h
910Ah	0101h	3000h

BRK

Instruction code	[0 0 0 0 0 0 0 0][0 0 0 0 0 1 0 1] 0005H
Argument	
Word count	1
Cycle count	1
Function	(PC)←(PC): This instruction sequence 1 time
Affected flags	

[Description]

This instruction halts the program counter (PC) while preserving the current CPU state. The halt state can be reset by generating an interrupt or reset.

BV r8

Instruction code	[1 1 0 1 1 0 1 1][r7r6r5r4r3r2r1r0] DB00H
Argument	r8 = 8bit(relative address, signed)
Word count	1
Cycle count	2 or 3
Function	If OV = 1, then (PC)←(PC)+2±(r8) If OV = 0, then (PC)←(PC)+2
Affected flags	

[Description]

This instruction adds the value of the relative address designated by r8 + 2 to the program counter (PC) and places the result in the PC if the value of the overflow flag (OS) is 1. If the value of OV is 0, 2 is added to the PC.

The legitimate value range of the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

```
        MOV.W   R2, #0x789A
        MOV.W   R3, #0xFFFF
loop:
        ADD     R2, #0x0234
        BV      LA          ;; NOT JUMP LA
        ADD     R2, #0x2345
        BV      LB          ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R2	R3	PSW
-	-	-	-
9004h	789Ah	-	2000h
9008h	789Ah	FFFFh	3040h
900Ch	7ACEh	FFFFh	2000h
900Eh	7ACEh	FFFFh	2000h
9012h	9E13h	FFFFh	2058h
901Ah	9E13h	FFFFh	2058h
-	-	-	-
-	-	-	-
-	-	-	-
901Ch	9E13h	0000h	301Bh
901Eh	9E13h	0000h	301Bh

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BV Rd, #imm8, r12

Instruction code	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][1 0 1 1 r11 to r8][r7 to r0] 2000B000H
Argument	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of (Rd) - #imm8 is OV = 1, then (PC) ←(PC)+4±(r12) If result of (Rd) - #imm8 is OV = 0, then (PC) ←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the overflow flag (OV) is set to 1 as the result of subtracting immediate data imm8 from the contents of the general-purpose register Rd. If OV is set to 0 as the result of the subtraction, 4 is added to the PC.

The legitimate value range of Rd is from R0 to R7, that of imm8 is from 0 to FFh, and that of the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0x8000
        MOV.W   R1, #0x5678
        MOV.W   R2, #0x0012
        MOV.W   R3, #0xFFFF
loop:
        BV      R1, #0x12, LA   ;; NOT JUMP LA
        BV      R0, #0x12, LB   ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9004h	8000h	-	-	-	0061h
9008h	8000h	5678h	-	-	1000h
900Ah	8000h	5678h	0012h	-	2000h
900Eh	8000h	5678h	0012h	FFFFh	3040h
9012h	8000h	5678h	0012h	FFFFh	1000h
901Ch	8000h	5678h	0012h	FFFFh	0038h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
901Eh	8000h	5678h	0012h	0000h	301Bh
9020h	8000h	5678h	0012h	0000h	301Bh

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BV Rd, Rs, r12

Instruction code	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][1 0 1 1 r11 to r8][r7 to r0] 0D00B000H
Argument	Rd = 4bit(R select), Rs = 4bit(R select), r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of (Rd) – (Rs) is OV = 1, then (PC) ←(PC)+4±(r12) If result of (Rd) – (Rs) is OV= 0, then (PC) ←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the overflow flag (OV) is set to 1 as the result of subtracting the contents of the general-purpose register Rs from the contents of the general-purpose register Rd. If OV is set to 0 as the result of the subtraction, 4 is added to the PC.

The legitimate value range of Rd is from R0 to R15, that of Rs is from R0 to R15, and that of the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0x89AB
        MOV.W   R1, #0x5678
        MOV.W   R2, #0x1234
        MOV.W   R3, #0xFFFF
loop:
        BV      R1, R2, LA      ;; NOT JUMP LA
        BV      R0, R2, LB      ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9004h	89ABh	-	-	-	0040h
9008h	89ABh	5678h	-	-	1000h
900Ch	89ABh	5678h	1234h	-	2020h
9010h	89ABh	5678h	1234h	FFFFh	3040h
9014h	89ABh	5678h	1234h	FFFFh	1000h
901Eh	89ABh	5678h	1234h	FFFFh	0010h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	89ABh	5678h	1234h	0000h	3013h
9022h	89ABh	5678h	1234h	0000h	3013h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BV Rx, #imm16, r8

Instruction code	[1 1 0 0 1 0 1 1][r7r6r5r4r3r2r1r0][i15 to i8][i7 to i0] CB000000H
Argument	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)
Word count	2
Cycle count	3 or 4
Function	If result of (Rx) - #imm16 is OV = 1, then (PC) ←(PC)+4±(r8) If result of (Rx) - #imm16 is OV = 0, then (PC) ←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S

[Description]

This instruction adds the value of the relative address designated by r8 + 4 to the program counter (PC) and places the result in the PC if the overflow flag (OV) is set to 1 as the result of subtracting immediate data imm16 from the contents of the general-purpose register Rx designated indirectly by bits 12 to 15 (N0 to N3) of the PSW. If OV is set to 0 as the result of the subtraction, 4 is added to the PC.
The legitimate value range of imm16 is from 0 to FFFFh, and that of the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

```
        MOV.W   R2, #0x1234
        MOV.W   R3, #0xFFFF
loop:
        MOV.W   R1, #0x5678
        BV      Rx, #0x1234, LA ;; NOT JUMP LA
        MOV.W   R0, #0x8000
        BV      Rx, #0x1234, LB ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9004h	-	-	1234h	-	2020h
9008h	-	-	1234h	FFFFh	3040h
900Ch	-	5678h	1234h	FFFFh	1000h
9010h	-	5678h	1234h	FFFFh	1000h
9014h	8000h	5678h	1234h	FFFFh	0061h
901Eh	8000h	5678h	1234h	FFFFh	0038h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	8000h	5678h	1234h	0000h	301Bh
9022h	8000h	5678h	1234h	0000h	301Bh

<Note>

This instruction takes 4 cycles to execute if the conditions are met.

BZ r8

Instruction code	[1 1 0 1 1 1 1 1][r7r6r5r4r3r2r1r0] DF00H
Argument	r8 = 8bit(relative address, signed)
Word count	1
Cycle count	2 or 3
Function	If Z16 = 1, then (PC)←(PC)+2±(r8) If Z16 = 0, then (PC)←(PC)+2
Affected flags	

[Description]

This instruction adds the value of the relative address designated by r8 + 2 to the program counter (PC) and places the result in the PC if the value of the 16-bit operation flag (Z16) is 1. If the value of Z16 is 0, 2 is added to the PC.

The legitimate value range of the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

```
        MOV.W   R3, #0x1234
loop:
        BZ      LA          ;; NOT JUMP LA
        MOV.W   R3, #0x0000
        BZ      LB          ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R3	PSW
-	-	-
9004h	1234h	3020h
9006h	1234h	3020h
9008h	0000h	3003h
9010h	0000h	3003h
-	-	-
-	-	-
-	-	-
9012h	0001h	3020h
9014h	0001h	3020h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BZ Rd, #imm8, r12

Instruction code	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][1 1 1 1 r11 to r8][r7 to r0] 2000F000H
Argument	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of (Rd) - #imm8 is Z16 = 1, then (PC)←(PC)+4±(r12) If result of (Rd) - #imm8 is Z16 = 0, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the 16-bit operation flag (Z16) is set to 1 as the result of subtracting immediate data imm8 from the contents of the general-purpose register Rd. If Z16 is set to 0 as the result of the subtraction, 4 is added to the PC.

The legitimate value range of Rd is from R0 to R7, that of imm8 is from 0 to FFh, and that of the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0x0034
        MOV.W   R1, #0x8234
        MOV.W   R2, #0x0034
        MOV.W   R3, #0xFFFF
loop:
        BZ      R1, #0x34, LA   ;; NOT JUMP LA
        BZ      R0, #0x34, LB   ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9002h	0034h	-	-	-	0020h
9006h	0034h	8234h	-	-	1060h
9008h	0034h	8234h	0034h	-	2020h
900Ch	0034h	8234h	0034h	FFFFh	3040h
9010h	0034h	8234h	0034h	FFFFh	1041h
901Ah	0034h	8234h	0034h	FFFFh	0003h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
901Ch	0034h	8234h	0034h	0000h	3003h
901Eh	0034h	8234h	0034h	0000h	3003h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BZ Rd, Rs, r12

Instruction code	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][1 1 1 1 r11 to r8][r7 to r0] 0D00F000H
Argument	Rd = 4bit(R select), Rs = 4bit(R select), r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of (Rd) – (Rs) is Z16 = 1, then (PC)←(PC)+4±(r12) If result of (Rd) – (Rs) is Z16 = 0, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the 16-bit operation flag (Z16) is set to 1 as the result of subtracting the contents of the general-purpose register Rs from the contents of the general-purpose register Rd. If Z16 is set to 0 as the result of the subtraction, 4 is added to the PC.

The legitimate value range of Rd is from R0 to R15, that of Rs is from R0 to R15, and that of the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0x1234
        MOV.W   R1, #0x8234
        MOV.W   R2, #0x1234
        MOV.W   R3, #0xFFFF
loop:
        BZ      R1, R2, LA      ;; NOT JUMP LA
        BZ      R0, R2, LB      ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9004h	1234h	-	-	-	0020h
9008h	1234h	8234h	-	-	1060h
900Ch	1234h	8234h	1234h	-	2020h
9010h	1234h	8234h	1234h	FFFFh	3040h
9014h	1234h	8234h	1234h	FFFFh	1031h
901Eh	1234h	8234h	1234h	FFFFh	0003h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	1234h	8234h	1234h	0000h	3003h
9022h	1234h	8234h	1234h	0000h	3003h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BZ Rx, #imm16, r8

Instruction code	[1 1 0 0 1 1 1 1][r7r6r5r4r3r2r1r0][i15 to i8][i7 to i0] CF000000H
Argument	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)
Word count	2
Cycle count	3 or 4
Function	If result of (Rx) - #imm16 is Z16 = 1, then (PC)←(PC)+4±(r8) If result of (Rx) - #imm16 is Z16 = 0, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S

[Description]

This instruction adds the value of the relative address designated by r8 + 4 to the program counter (PC) and places the result in the PC if the 16-bit operation flag (Z16) is set to 1 as the result of subtracting immediate data imm16 from the contents of the general-purpose register Rx designated indirectly by bits 12 to 15 (N0 to N3) of the PSW. If Z16 is set to 0 as the result of the subtraction, 4 is added to the PC.
The legitimate value range of imm16 is from 0 to FFFFh, and that of the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

		PC	R0	R1	R2	R3	PSW
		-	-	-	-	-	-
	MOV.W R2, #0x1234	9004h	-	-	1234h	-	2020h
	MOV.W R3, #0xFFFF	9008h	-	-	1234h	FFFFh	3040h
loop:							
	MOV.W R1, #0x8234	900Ch	-	8234h	1234h	FFFFh	1060h
	BZ Rx, #0x1234, LA ;; NOT JUMP LA	9010h	-	8234h	1234h	FFFFh	1031h
	MOV.W R0, #0x1234	9014h	1234h	8234h	1234h	FFFFh	0030h
	BZ Rx, #0x1234, LB ;; JUMP LB	901Eh	1234h	8234h	1234h	FFFFh	0003h
	BR loop	-	-	-	-	-	-
LA:							
	DEC R3	-	-	-	-	-	-
	BR loop	-	-	-	-	-	-
LB:							
	INC R3	9020h	1234h	8234h	1234h	0000h	3003h
	NOP	9020h	1234h	8234h	1234h	0000h	3003h

<Note>

This instruction takes 4 cycles to execute if the conditions are met.

BZ. B r8

Instruction code	[1 1 0 1 1 1 1 0][r7r6r5r4r3r2r1r0] DE00H
Argument	r8 = 8bit(relative address, signed)
Word count	1
Cycle count	2 or 3
Function	If Z8 = 1, then (PC)←(PC)+2± (r8) If Z8 = 0, then (PC)←(PC)+2
Affected flags	

[Description]

This instruction adds the value of the relative address designated by r8 + 2 to the program counter (PC) and places the result in the PC if the value of the 8-bit operation flag (Z8) is 1. If the value of Z8 is 0, 2 is added to the PC.

The legitimate value range of the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

```
        MOV.W   R3, #0x1234
loop:
        BZ.B    LA          ;; NOT JUMP LA
        MOV.W   R3, #0x1200
        BZ.B    LB          ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R3	PSW
-	-	-
9004h	1234h	3020h
9006h	1234h	3020h
900Ah	1200h	3001h
9012h	1200h	3001h
-	-	-
-	-	-
-	-	-
9014h	1201h	3020h
9016h	1201h	3020h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BZ. B Rd, #imm8, r12

Instruction code	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][1 1 1 0 r11 to r8][r7 to r0] 2000E000H
Argument	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of (Rd) - #imm8 is Z8 =1, then (PC)←(PC)+4±(r12) If result of (Rd) - #imm8 is Z8 =0, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the 8-bit operation flag (Z8) is set to 1 as the result of subtracting immediate data imm8 from the contents of the general-purpose register Rd. If Z8 is set to 0 as the result of the subtraction, 4 is added to the PC.

The legitimate value range of Rd is from R0 to R7, that of imm8 is from 0 to FFh, and that of the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

			PC	R0	R1	R2	R3	PSW
			-	-	-	-	-	-
	MOV.W	R0, #0x5634	9004h	5634h	-	-	-	0020h
	MOV.W	R1, #0x8000	9008h	5634h	8000h	-	-	1061h
	MOV.W	R2, #0x1234	900Ch	5634h	8000h	1234h	-	2020h
	MOV.W	R3, #0xFFFF	9010h	5634h	8000h	1234h	FFFFh	3040h
loop:								
	BZ.B	R1, #0x34, LA ;; NOT JUMP LA	9014h	5634h	8000h	1234h	FFFFh	1038h
	BZ.B	R0, #0x34, LB ;; JUMP LB	901Eh	5634h	8000h	1234h	FFFFh	0001h
	BR	loop	-	-	-	-	-	-
LA:								
	DEC	R3	-	-	-	-	-	-
	BR	loop	-	-	-	-	-	-
LB:								
	INC	R3	9020h	5634h	8000h	1234h	0000h	3003h
	NOP		9022h	5634h	8000h	1234h	0000h	3003h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BZ. B Rd, Rs, r12

Instruction code	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][1 1 1 0 r11 to r8][r7 to r0] 0D00E000H
Argument	Rd = 4bit(R select), Rs = 4bit(R select), r12 = 12bit(relative address, signed)
Word count	2
Cycle count	2 or 3
Function	If result of (Rd) – (Rs) is Z8 =1, then (PC)←(PC)+4±(r12) If result of (Rd) – (Rs) is Z8 =0, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S, N0 to N3

[Description]

This instruction adds the value of the relative address designated by r12 + 4 to the program counter (PC) and places the result in the PC if the 8-bit operation flag (Z8) is set to 1 as the result of subtracting the contents of the general-purpose register Rs from the contents of the general-purpose register Rd. If Z8 is set to 0 as the result of the subtraction, 4 is added to the PC.

The legitimate value range of Rd is from R0 to R15, that of Rs is from R0 to R15, and that of the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example]

```
        MOV.W   R0, #0x5634
        MOV.W   R1, #0x8000
        MOV.W   R2, #0x1234
        MOV.W   R3, #0xFFFF
loop:
        BZ.B    R1, R2, LA      ;; NOT JUMP LA
        BZ.B    R0, R2, LB      ;; JUMP LB
        BR      loop
LA:
        DEC     R3
        BR      loop
LB:
        INC     R3
        NOP
```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9004h	5634h	-	-	-	0020h
9008h	5634h	8000h	-	-	1061h
900Ch	5634h	8000h	1234h	-	2020h
9010h	5634h	8000h	1234h	FFFFh	3040h
9014h	5634h	8000h	1234h	FFFFh	1038h
901Eh	5634h	8000h	1234h	FFFFh	0001h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	5634h	8000h	1234h	0000h	3003h
9022h	5634h	8000h	1234h	0000h	3003h

<Note>

This instruction takes 3 cycles to execute if the conditions are met.

BZ. B Rx, #imm16, r8

Instruction code	[1 1 0 0 1 1 1 0][r7r6r5r4r3r2r1r0][i15 to i8][i7 to i0] CE000000H
Argument	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)
Word count	2
Cycle count	3 or 4
Function	If result of (Rx) - #imm16 is Z8 =1, then (PC)←(PC)+4±(r8) If result of (Rx) - #imm16 is Z8 =0, then (PC)←(PC)+4
Affected flags	Z8, Z16, CY, HC, OV, P, S

[Description]

This instruction adds the value of the relative address designated by r8 + 4 to the program counter (PC) and places the result in the PC if the 8-bit operation flag (Z8) is set to 1 as the result of subtracting immediate data imm16 from the contents of the general-purpose register Rx designated indirectly by bits 12 to 15 (N0 to N3) of the PSW. If Z8 is set to 0 as the result of the subtraction, 4 is added to the PC.
The legitimate value range of imm16 is from 0 to FFFFh, and that of the relative address designated by r8 is that of signed 8-bit data (-128 to 127).

[Example]

			PC	R0	R1	R2	R3	PSW
			-	-	-	-	-	-
	MOV.W	R2, #0x1234	9004h	-	-	1234h	-	2020h
	MOV.W	R3, #0xFFFF	9008h	-	-	1234h	FFFFh	3040h
loop:								
	MOV.W	R1, #0x8000	900Ch	-	8000h	1234h	FFFFh	1061h
	BZ.B	Rx, #0x1234, LA ;; NOT JUMP LA	9010h	-	8000h	1234h	FFFFh	1038h
	MOV.W	R0, #0x5634	9014h	5634h	8000h	1234h	FFFFh	0038h
	BZ.B	Rx, #0x1234, LB ;; JUMP LB	901Eh	5634h	8000h	1234h	FFFFh	0001h
	BR	loop	-	-	-	-	-	-
LA:								
	DEC	R3	-	-	-	-	-	-
	BR	loop	-	-	-	-	-	-
LB:								
	INC	R3	9020h	5634h	8000h	1234h	0000h	3003h
	NOP		9022h	5634h	8000h	1234h	0000h	3003h

<Note>

This instruction takes 4 cycles to execute if the conditions are met.

CALL Rb, Rs

Instruction code	[0 0 0 0 0 0 0 0][1 0 1 b0 s3s2s1s0] 00A0H
Argument	Rb = 1bit(absolute address), Rs = 4bit(absolute address)
Word count	1
Cycle count	4
Function	(SP)←(SP)+4: [SP+1, SP]←(PC&0000FFFFh), [SP+3, SP+2]←(PC&FFFF0000h) (PC)←(Rb<<16+Rs)
Affected flags	

[Description]

This instruction stores the address of the instruction following this CALL instruction (return address) in the data memory location (RAM) designated by the stack pointer (SP) and increments the SP. Finally, the instruction places the absolute address (of which the higher-order 16 bits are the contents of the general-purpose register designated by the base register (Rb) and the lower-order 16 bits are the contents of Rs) in the program counter (PC).
The legitimate values of Rb is R8 and R9, and the legitimate value range of Rs is from R0 to R15.

[Example] The value of label LA is 910AH.

			PC	RAM (00h)	RAM (01h)	RAM (02h)	RAM (03h)	R3	R8	PSW	SP
			-	-	-		-	-	-	-	-
	MOV.W R15, #0x0000		9004h	-	-		-	-	-	F003h	0000h
	MOV.W R3, #0x910A		9004h	-	-		-	910Ah	-	3060h	0000h
	MOV.W R8, #0x0000		9008h	-	-		-	910Ah	0000h	8003h	0000h
loop:											
	CALL R8, R3	;; CALL LA	910Ah	0Ah	90h	00h	00h	910Ah	0000h	8003h	0004h
	INC R3		9010h	0Ah	90h	00h	00h	910Ch	0000h	3060h	0000h
	NOP		9012h	0Ah	90h	00h	00h	910Ch	0000h	3060h	0000h
	.										
	.										
	.										
LA:											
	INC R3		910Ch	0Ah	90h	00h	00h	910Bh	0000h	3040h	0004h
	RET		900Eh	0Ah	90h	00h	00h	910Bh	0000h	3040h	0000h

CALLF a24

Instruction code	[0 0 0 0 0 0 0 1][a7a6a5a4a3a2a1a0][a23 to a16][a15 to a8] 0100H
Argument	a24 = 24bit(absolute address)
Word count	2
Cycle count	4
Function	(SP)←(SP)+4: [SP+1, SP]←(PC&0000FFFFh), [SP+3, SP+2]←(PC&FFFF0000h) (PC)←(a24)
Affected flags	

[Description]

This instruction stores the address of the instruction following this CALL instruction (return address) in the data memory location (RAM) designated by the stack pointer (SP) and increments the SP. Finally, the instruction places the absolute address (a24) in the program counter (PC).
The legitimate value range of a24 is from 0 to FF__FFFFh.

[Example] The value of label LA is 910AH.

			PC	RAM (00h)	RAM (01h)	RAM (02h)	RAM (03h)	R3	PSW	SP
			-	-	-	-	-	-	-	-
	MOV.W	R15, #0x0000	9004h	-	-	-	-	-	F003h	0000h
	MOV.W	R3, #0xFFFF	9008h	-	-	-	-	FFFFh	3040h	0000h
loop:										
	CALLF	LA ;; CALL LA	910Ah	0Ch	90h	00h	00h	FFFFh	3040h	0004h
	INC	R3	900Eh	0Ch	90h	00h	00h	0001h	3020h	0000h
	NOP		9010h	0Ch	90h	00h	00h	0001h	3020h	0000h
	.									
	.									
	.									
LA:										
	INC	R3	910Ch	0Ch	90h	00h	00h	0000h	3003h	0004h
	RET		900Ch	0Ch	90h	00h	00h	0000h	3003h	0000h

CALLR r12

Instruction code	[0 0 0 1 r11r10r9r8][r7r6r5r4r3r2r1 1] 1001H
Argument	r12 = 12bit(relative address, signed)
Word count	1
Cycle count	4
Function	(SP)←(SP)+4: [SP+1, SP]←(PC&0000FFFFh), [SP+3, SP+2]←(PC&FFFF0000h) (PC)←(PC)+2±(r12)
Affected flags	

[Description]

This instruction stores the address of the instruction following this CALL instruction (return address) in the data memory location (RAM) designated by the stack pointer (SP) and increments the SP. Finally, the instruction adds the value of the relative address (r12) + 2 to the program counter (PC) and places the result in the PC.

The legitimate value range of the relative address designated by r12 is that of signed 12-bit data (-2048 to 2047).

[Example] The value of label LA is 910AH.

			PC	RAM (00h)	RAM (01h)	RAM (02h)	RAM (03h)	R3	PSW	SP
			-	-	-	-	-	-	-	-
	MOV.W R15, #0x0000		9004h	-	-	-	-	-	F003h	0000h
	MOV.W R3, #0xFFFF		9008h	-	-	-	-	FFFFh	3040h	0000h
loop:										
	CALLR LA	;; CALL LA	910Ah	0Ah	90h	00h	00h	FFFFh	3040h	0004h
	INC R3		900Ch	0Ah	90h	00h	00h	0001h	3020h	0000h
	NOP		900Eh	0Ah	90h	00h	00h	0001h	3020h	0000h
	.									
	.									
	.									
LA:										
	INC R3		910Ch	0Ah	90h	00h	00h	0000h	3003h	0004h
	RET		900Ah	0Ah	90h	00h	00h	0000h	3003h	0000h

<Note>

The value of the relative address (r) is valid if it is in the value range of signed 12-bit data (-2048 to +2047).

CALLR Rs

Instruction code	[0 0 0 0 0 0 0 0][0 0 0 1 s3s2s1s0] 0010H
Argument	Rs = 4bit(relative address, signed)
Word count	1
Cycle count	4
Function	(SP)←(SP)+4: [SP+1, SP]←(PC&0000FFFFh), [SP+3, SP+2]←(PC&FFFF0000h) (PC)←(PC)+2±(Rs)
Affected flags	

[Description]

This instruction stores the address of the instruction following this CALL instruction (return address) in the data memory location (RAM) designated by the stack pointer (SP) and increments the SP. Finally the instruction adds the value of the relative address (the contents of the general-purpose register Rs) + 2 to the program counter (PC) and places the result in the PC.
The legitimate value range of Rs is from R0 to R15.

[Example] The value of label LA is 910AH.

```
        MOV.W   R15, #0x0000
        MOV.W   R3, #0x0100
loop:
        CALLR   R3              ;; CALL LA
        INC     R3
        NOP
          .
          .
          .
LA:
        INC     R3
        RET
```

PC	RAM (00h)	RAM (01h)	RAM (02h)	RAM (03h)	R3	PSW	SP
-	-	-	-	-	-	-	-
9004h	-	-	-	-	-	F003h	0000h
9008h	-	-	-	-	0100h	3021h	0000h
910Ah	0Ah	90h	00h	00h	0100h	3021h	0004h
900Ch	0Ah	90h	00h	00h	0102h	3000h	0000h
900Eh	0Ah	90h	00h	00h	0102h	3000h	0000h
910Ch	0Ah	90h	00h	00h	0101h	3000h	0004h
900Ah	0Ah	90h	00h	00h	0101h	3000h	0000h

<Note>

The value of the relative address (Rs) is valid if it is in the value range of signed 16-bit data (-32768 to +32767).

CBW Rd

Instruction code	[0 0 1 1 0 0 0 0][1 0 1 0 d3d2d1d0] 30A0H
Argument	Rd = 4bit(R select)
Word count	1
Cycle count	1
Function	if (Rd) of bit7 = 1, then Hibyte(Rd) = FFh if (Rd) of bit7 = 0, then Hibyte(Rd) = 00h
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction extends arithmetic 8-bit data into 16-bit data regarding bit 7 of the general-purpose register Rd as the sign bit.
The legitimate value range of Rd is from R0 to R15.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-
MOV.W	R0, #0x2345	2345h	-	-	-	0	0	0	0	0
MOV.W	R1, #0xFEDC	2345h	FEDCh	-	-	1	0	0	0	1
MOV.W	R2, #0x8888	2345h	FEDCh	8888h	-	2	0	0	0	1
MOV.W	R3, #0x5500	2345h	FEDCh	8888h	5500h	3	1	0	0	0
CBW	R0	0045h	FEDCh	8888h	5500h	0	0	0	1	0
CBW	R1	0045h	FFDCh	8888h	5500h	1	0	0	1	1
CBW	R2	0045h	FFDCh	FF88h	5500h	2	0	0	0	1
CBW	R3	0045h	FFDCh	FF88h	0000h	3	1	1	0	0

<Note>

The higher-order 8 bits are set to FFH if bit 7 of Rd is 1 and to 00H if bit 7 is 0.

CLR1 m16, #imm3

Instruction code	[1 1 1 X i2i1i0 0][m7m6m5m4m3m2m1m0] E000H(RAM), F000H(SFR)
Argument	m16 = 16bit(Lower 8bit valid for operation code), imm3 = 3bit(bit select)
Word count	1
Cycle count	2
Function	(m16) of bit #imm3 ← 0, (PC)←(PC)+2
Affected flags	Z8, Z16, P, S

[Description]

This instruction zero clears the bit designated by immediate data imm3 in the data memory location m16.
The compiler generates the instruction code while regarding RAM or SFR as the destination of transfer according to the value of m16 (first operand data).

- When specifying a RAM location, specify m16 with a value from 00H to FFH (0000H to 00FFH). It is disallowed to specify a RAM address not lower than 100H.
- When specifying a SFR, specify m16 with a value from 7F00H to 7FFFH.

The basic types of generated instruction code are E000H (RAM) and F000H (SFR), respectively, The lower-order 8 bits of m16 are reflected in the behavior of the instruction code.
The legitimate value range of imm3 is from 0 to 7h.

[Example]

		RAM (50h)	RAM (51h)	RAM (52h)	RAM (53h)	Z8	Z16	P	S
		-	-	-	-	-	-	-	-
MOV.B	0x50, #0xFF	FFh	-	-	-	0	0	0	1
MOV.B	0x51, #0x33	FFh	33h	-	-	0	0	0	0
MOV.B	0x52, #0x00	FFh	33h	00h	-	1	1	0	0
MOV.B	0x53, #0x54	FFh	33h	00h	54h	0	0	1	0
CLR1	0x50, #0x02	FBh	33h	00h	54h	0	0	1	1
CLR1	0x51, #0x00	FBh	32h	00h	54h	0	0	1	0
CLR1	0x52, #0x04	FBh	32h	00h	54h	1	1	0	0
CLR1	0x53, #0x04	FBh	32h	00h	44h	0	0	0	0

CLR1 Rd, #imm4

Instruction code	[0 0 0 0 1 0 0 0][i3i2i1i0d3d2d1d0] 0800H
Argument	Rd = 4bit(R select), imm4 = 4bit(bit select)
Word count	1
Cycle count	1
Function	(Rd) of bit #imm4 ← 0, (PC)←(PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction zero clears the bit designated by immediate data imm4 in the general-purpose register Rd. The legitimate value range of Rd is from R0 to R15 and that of im4 from 0 to F.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-
MOV.W	R0, #0xFFFF	FFFFh	-	-	-	0	0	0	0	1
MOV.W	R1, #0x0001	FFFFh	0001h	-	-	1	0	0	1	0
MOV.W	R2, #0x0000	FFFFh	0001h	0000h	-	2	1	1	0	0
MOV.W	R3, #0x7654	FFFFh	0001h	0000h	7654h	3	0	0	0	0
CLR1	R0, #0x01	FFFDh	0001h	0000h	7654h	0	0	0	1	1
CLR1	R1, #0x00	FFFDh	0000h	0000h	7654h	1	1	1	0	0
CLR1	R2, #0x04	FFFDh	0000h	0000h	7654h	2	1	1	0	0
CLR1	R3, #0x0D	FFFDh	0000h	0000h	5654h	3	0	0	1	0

CLR1 Rd, Rs

Instruction code	[0 0 0 0 1 0 1 0][s3s2s1s0d3d2d1d0] 0A00H
Argument	Rd = 4bit(R select), Rs = 4bit(bit select)
Word count	1
Cycle count	1
Function	(Rd) of bit (Rs)&000Fh ← 0, (PC)←(PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction zero clears the bit designated by the lower-order 4 bits of the general-purpose register Rs in the general-purpose register Rd.
The legitimate value range of Rd is from R0 to R15 and that of Rd from R0 to R15.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-
MOV.W	R0, #0xFFFF	FFFFh	-	-	-	0	0	0	0	1
MOV.W	R1, #0x0001	FFFFh	0001h	-	-	1	0	0	1	0
MOV.W	R2, #0x0000	FFFFh	0001h	0000h	-	2	1	1	0	0
MOV.W	R3, #0x7654	FFFFh	0001h	0000h	7654h	3	0	0	0	0
CLR1	R0, R1	FFFDh	0001h	0000h	7654h	0	0	0	1	1
CLR1	R1, R2	FFFDh	0000h	0000h	7654h	1	1	1	0	0
CLR1	R2, R3	FFFDh	0000h	0000h	7654h	2	1	1	0	0
CLR1	R3, R0	FFFDh	0000h	0000h	5654h	3	0	0	1	0

DEC Rd[, #imm2]

Instruction code	[0 0 1 1 0 0 0 0][0 1 i1i0d3d2d1d0] 3040H
Argument	Rd = 4bit(R select), imm2 = 2bit(immediate data)
Word count	1
Cycle count	1
Function	(Rd)←(Rd) - #imm2 – 1, (PC)←(PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction subtracts the value of immediate data imm2 + 1 from the contents of the general-purpose register Rd and places the result in Rd.
The legitimate value range of Rd is from R0 to R15 and that of imm2 from 0 to 3.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-
MOV.W	R0, #0x1234	1234h	-	-	-	0	0	0	1	0
MOV.W	R1, #0x0000	1234h	0000h	-	-	1	1	1	0	0
MOV.W	R2, #0x0003	1234h	0000h	0003h	-	2	0	0	0	0
MOV.W	R3, #0x8765	1234h	0000h	0003h	8765h	3	0	0	0	1
DEC	R0	1233h	0000h	0003h	8765h	0	0	0	0	0
DEC	R0, #0	1232h	0000h	0003h	8765h	0	0	0	1	0
DEC	R1, #1	1232h	FFFEh	0003h	8765h	1	0	0	1	1
DEC	R2, #2	1232h	FFFEh	0000h	8765h	2	1	1	0	0
DEC	R3, #3	1232h	FFFEh	0000h	8761h	3	0	0	1	1

<Note>

imm2 is assumed to be 0 if the immediate data (imm2) is omitted.

DIV

Instruction code	[0 0 0 0 0 0 0 0][1 1 0 0 0 0 0 0] 00C0H
Argument	
Word count	1
Cycle count	18 cycles
Function	(R0: quotient)…(R1: remainder)←(R0)÷(R2), (PC)←(PC)+2
Affected flags	Z8, Z16, P, S CY, HC, OV, and N0 to N3 all cleared

[Description]

This instruction places the result of dividing the contents of the general-purpose register R0 by the contents of R2 and places the quotient in R0 and the remainder in R1.
No valid result is guaranteed if the value of R2 is 0.

[Example]

```
MOV.W   R0, #0x89AB
MOV.W   R1, #0x5678
MOV.W   R2, #0x1234
MOV.W   R3, #0xDEF0
DIV
```

R0	R1	R2	R3	PSW
-	-	-	-	-
89ABh	-	-	-	0040h
89ABh	5678h	-	-	1000h
89ABh	5678h	1234h	-	2020h
89ABh	5678h	1234h	DEF0h	3040h
0007h	0A3Fh	1234h	DEF0h	0020h

<Note>

The flags (Z8, Z16, P, and S) are affected by the contents of R0 (quotient).

DIVLH

Instruction code	[0 0 0 0 0 0 0 0][1 1 1 0 0 0 0 0] 00E0H
Argument	
Word count	1
Cycle count	18 cycles
Function	(R0: quotient)…(R1: remainder)←(R1<<16+R0)÷(R2), (PC)←(PC)+2
Affected flags	Z8, Z16, P, S CY, HC, OV, and N0 to N3 all cleared

[Description]

This instruction places the result of dividing unsigned 32-bit data (R1<<16 + R0) by the contents of R2 in R0 and the remainder in R1.
No valid result is guaranteed if the value of R2 is 0 or R1≧R2.

[Example]

```
MOV.W   R0, #0x89AB
MOV.W   R1, #0x5678
MOV.W   R2, #0x1234
MOV.W   R3, #0xDEF0
DIVLH
MOV.W   R0, #0xFFFF
MOV.W   R1, #0x2FFF
MOV.W   R2, #0x0000
DIVLH
MOV.W   R0, #0x5555
DIVLH
```

R0	R1	R2	R3	PSW
-	-	-	-	-
89ABh	-	-	-	0040h
89ABh	0567h	-	-	1020h
89ABh	0567h	1234h	-	2020h
89ABh	0567h	1234h	DEF0h	3040h
4C01h	0777h	1234h	DEF0h	0000h
FFFFh	0777h	1234h	DEF0h	0040h
FFFFh	2FFFh	1234h	DEF0h	1020h
FFFFh	2FFFh	F000h	DEF0h	2041h
3333h	2FFFh	F000h	DEF0h	0000h
5555h	2FFFh	F000h	DEF0h	0000h
3332h	7555h	F000h	DEF0h	0020h

<Note>

The flags (Z8, Z16, P, and S) are affected by the contents of R0 (quotient).

HALT

Instruction code	[0 0 0 0 0 0 0 0][0 0 0 0 1 0 0 0]	0008H
Argument		
Word count	1	
Cycle count	1	
Function	HALT mode, (PC)←(PC)+2	
Affected flags		

[Description]

The CPU enters the HALT mode after executing the HALT instruction.

HOLD

Instruction code	[0 0 0 0 0 0 0 0][0 0 0 0 1 0 1 0]	000AH
Argument		
Word count	1	
Cycle count	1	
Function	HOLD mode, (PC)←(PC)+2	
Affected flags		

[Description]

The CPU enters the HOLD mode after executing the HOLD instruction.

HOLDX

Instruction code	[0 0 0 0 0 0 0 0][0 0 0 0 1 0 1 1]	000BH
Argument		
Word count	1	
Cycle count	1	
Function	HOLDX mode, (PC)←(PC)+2	
Affected flags		

[Description]

The CPU enters the HOLDX mode after executing the HOLDX instruction.

ICALL Rb, Rs

Instruction code	[0 0 0 0 0 0 0 0][0 1 1 b0 s3s2s1s0] 0060H
Argument	Rb = 1bit(absolute address), Rs = 4bit(absolute address)
Word count	1
Cycle count	4
Function	(SP)←(SP)+6: [SP+1, SP]←(PC&0000FFFFh), [SP+3, SP+2]←(PC&FFFF0000h), [SP+5, SP+4]←(PSW) (PC)←(Rb<<16+Rs)
Affected flags	

[Description]

This instruction stores the address of the instruction following this ICALL instruction (return address) and the contents of the program status word (PSW) in the data memory locations (RAM) designated by the stack pointer (SP) and increments the SP. Finally, the instruction places the absolute address (of which the higher-order 16 bits are the contents of the general-purpose register designated by the base register (Rb) and the lower-order 16 bits are the contents of Rs) in the program counter (PC).

The legitimate values of Rb is R8 and R9, and the legitimate value range of Rs is from R0 to R15.

[Example] The value of label LA is 910AH.

	PC	RAM (00h)	RAM (01h)	RAM (02h)	RAM (03h)	RAM (04h)	RAM (05h)	R3	R8	PSW	SP
	-	-	-	-	-	-	-		-	-	-
MOV.W R15, #0x0000	9004h	-	-	-	-	-	-		-	F003h	0000h
MOV.W R3, #0x910A	9008h							910Ah		3060h	0000h
MOV.W R8, #0x0000	900Ch	-	-	-	-	-	-	910Ah	0000h	8003h	0000h
loop:											
ICALL R8, R3;; CALL LA	910Ah	0Eh	90h	00h	00h	03h	80h	910Ah	0000h	8003h	0006h
INC R3	900Ch	0Eh	90h	00h	00h	03h	80h	910Ch	0000h	3060h	0000h
NOP	900Eh	0Eh	90h	00h	00h	03h	80h	910Ch	0000h	3060h	0000h
.											
.											
.											
LA:											
INC R3	910Ch	0Eh	90h	00h	00h	03h	80h	910Bh	0000h	3040h	0006h
IRET	900Ah	0Eh	90h	00h	00h	03h	80h	910Bh	0000h	8003h	0000h

ICALLF a24

Instruction code	[0 0 0 0 0 0 1 1][a7a6a5a4a3a2a1a0][a23 to a16][a15 to a8] 0300H
Argument	a24 = 24bit(absolute address)
Word count	2
Cycle count	4
Function	(SP)←(SP)+6: [SP+1, SP]←(PC&0000FFFFh), [SP+3, SP+2]←(PC&FFFF0000h), [SP+6, SP+5]←(PSW) (PC)←(a24)
Affected flags	

[Description]

This instruction stores the address of the instruction following this ICALL instruction (return address) and the contents of the program status word (PSW) in the data memory locations (RAM) designated by the stack pointer (SP) and increments the SP. Finally, the instruction places the absolute address (a24) in the program counter (PC).

The legitimate value range of a24 is from 0 to FF__FFFFh.

[Example] The value of label LA is 910AH.

		PC	RAM (00h)	RAM (01h)	RAM (02h)	RAM (03h)	RAM (04h)	RAM (05h)	R3	PSW	SP
		-	-	-	-	-	-	-	-	-	-
MOV.W	R15, #0x0000	9004h	-	-	-	-	-	-	-	F003h	0000h
MOV.W	R3, #0xFFFF	9008h	-	-	-	-	-	-	FFFFh	3040h	0000h
loop:											
ICALLF	LA;; CALL LA	910Ah	0Ch	90h	00h	00h	40h	30h	FFFFh	3040h	0006h
INC	R3	900Eh	0Ch	90h	00h	00h	40h	30h	0001h	3020h	0000h
NOP		9010h	0Ch	90h	00h	00h	40h	30h	0001h	3020h	0000h
.											
.											
.											
LA:											
INC	R3	910Ch	0Ch	90h	00h	00h	40h	30h	0000h	3003h	0006h
IRET		900Ch	0Ch	90h	00h	00h	40h	30h	0000h	3040h	0000h

ICALLR Rs

Instruction code	[0 0 0 0 0 0 0 0][0 0 1 1 s3s2s1s0] 0030H
Argument	Rs = 4bit(relative address, signed)
Word count	1
Cycle count	4
Function	(SP)←(SP)+6: [SP+1, SP]←(PC&0000FFFFh), [SP+3, SP+2]←(PC&FFFF0000h), [SP+5, SP+4]←(PSW) (PC)←(PC)+2±(Rs)
Affected flags	

[Description]

This instruction stores the address of the instruction following this ICALL instruction (return address) and the contents of the program status word (PSW) in the data memory locations (RAM) designated by the stack pointer (SP) and increments the SP. Finally the instruction adds the value of the relative address (the contents of the general-purpose register Rs) + 2 to the program counter (PC) and places the result in the PC.
The legitimate value range of Rs is from R0 to R15 and that of the relative address (the contents of the general-purpose register Rs) is that of signed 16-bit data (-32768 to + 32767).

[Example] The value of label LA is 910AH.

	PC	RAM (00h)	RAM (01h)	RAM (02h)	RAM (03h)	RAM (04h)	RAM (05h)	R3	PSW	SP
	-	-	-	-	-	-	-	-	-	-
MOV.W R15, #0x0000	9004h	-	-	-	-	-	-	-	F003h	0000h
MOV.W R3, #0x910A	9008h	-	-	-	-	-	-	910Ah	3060h	0000h
loop:										
ICALLR R3;; CALL LA	910Ah	0Ah	90h	00h	00h	60h	30h	910Ah	3060h	0006h
INC R3	900Ch	0Ah	90h	00h	00h	60h	30h	910Ch	3060h	0000h
NOP	900Eh	0Ah	90h	00h	00h	60h	30h	910Ch	3060h	0000h
.										
.										
.										
LA:										
INC R3	910Ch	0Ah	90h	00h	00h	60h	30h	910Bh	3040h	0006h
IRET	900Ah	0Ah	90h	00h	00h	60h	30h	910Bh	3060h	0000h

INC Rd[, #imm2]

Instruction code	[0 0 1 1 0 0 0 0][0 0 i1i0d3d2d1d0] 3000H
Argument	Rd = 4bit(R select), imm2 = 2bit(immediate data)
Word count	1
Cycle count	1
Function	(Rd)←(Rd) + #imm2 + 1, (PC)←(PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction adds the value of immediate data imm2 + 1 to the contents of the general-purpose register Rd and places the result in Rd.
The legitimate value range of Rd is from R0 to R15 and that of imm2 from 0 to 3.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-
MOV.W	R0, #0x1234	1234h	-	-	-	0	0	0	1	0
MOV.W	R1, #0x0000	1234h	0000h	-	-	1	1	1	0	0
MOV.W	R2, #0xFFFD	1234h	0000h	FFFDh	-	2	0	0	1	1
MOV.W	R3, #0x8765	1234h	0000h	FFFDh	8765h	3	0	0	0	1
INC	R0	1235h	0000h	FFFDh	8765h	0	0	0	0	0
INC	R0, #0	1236h	0000h	FFFDh	8765h	0	0	0	0	0
INC	R1, #1	1236h	0002h	FFFDh	8765h	1	0	0	1	0
INC	R2, #2	1236h	0002h	0000h	8765h	2	1	1	0	0
INC	R3, #3	1236h	0002h	0000h	8769h	3	0	0	0	1

<Note>

imm2 is assumed to be 0 if the immediate data (imm2) is omitted.

IRET

Instruction code	[0 0 0 0 0 0 0 0][0 0 0 0 0 0 1 0] 0002H
Argument	
Word count	1
Cycle count	3
Function	(SP)←(SP)-6: (PC)←(PC-3<<24+SP-4<<16+SP-5<<8+SP-6) (PSW)←(SP-1<<8, SP-2)
Affected flags	

[Description]

This instruction decrements the value of the stack pointer (SP), places the contents of the data memory locations (RAM) designated by SP in the program counter (PC) and program status word (PSW), then resumes the execution of the interrupt acceptance function that has been disabled when the interrupt is accepted.

[Example] The value of label LA is 910AH.

	PC	RAM (00h)	RAM (01h)	RAM (02h)	RAM (03h)	RAM (04h)	RAM (05h)	R3	PSW	SP
	-	-	-	-	-	-	-	-	-	-
MOV.W R15, #0x0000	9004h	-	-	-	-	-	-	-	F003h	0000h
MOV.W R3, #0xFFFF	9008h	-	-	-	-	-	-	FFFFh	3040h	0000h
loop:										
ICALLF LA;; CALL LA	910Ah	0Ch	90h	00h	00h	40h	30h	FFFFh	3040h	0006h
INC R3	900Eh	0Ch	90h	00h	00h	40h	30h	0001h	3020h	0000h
NOP	9010h	0Ch	90h	00h	00h	40h	30h	0001h	3020h	0000h
.										
.										
.										
LA:										
INC R3	910Ch	0Ch	90h	00h	00h	40h	30h	0000h	3003h	0006h
IRET	900Ch	0Ch	90h	00h	00h	40h	30h	0000h	3040h	0000h

JMP Rb, Rs

Instruction code	[0 0 0 0 0 0 0 0][0 1 0 b0 s3s2s1s0] 0040H
Argument	Rb = 1bit(absolute address), Rs = 4bit(absolute address)
Word count	1
Cycle count	2
Function	(PC)←(Rb<<16+Rs)
Affected flags	

[Description]

This instruction places the absolute address (of which the higher-order 16 bits are the contents of the general-purpose register designated by the base register (Rb) and the lower-order 16 bits are the contents of Rs) in the program counter (PC).

The legitimate values of Rb is R8 and R9, and the legitimate value range of Rs is from R0 to R15.

[Example] The value of label LA is 9106H.

```
                                                    PC      R3      R8      PSW
                                                    |       -       -       -
        MOV.W   R3, #0x9106                         9004h   9106h   -       3060h
        MOV.W   R8, #0x0000                         9004h   9106h   0000h   8003h
loop:                                               |       |       |       |
        JMP     R8, R3          ;; JUMP LA          9106h   9106h   0000h   8003h
        NOP                                         -       -       -       -
        NOP                                         -       -       -       -
          .                                         |       |       |       |
          .                                         |       |       |       |
          .                                         |       |       |       |
LA:                                                 |       |       |       |
        INC     R3                                  9108h   9107h   0000h   3060h
        NOP                                         910Ah   9107h   0000h   3060h
```

JMPF a24

Instruction code	[0 0 0 0 0 0 1 0][a7a6a5a4a3a2a1a0][a23 to a16][a15 to a8]	0200H
Argument	a24 = 24bit(absolute address)	
Word count	2	
Cycle count	3	
Function	(PC)←(a24)	
Affected flags		

[Description]

This instruction places the absolute address a24 in the program counter (PC).
The legitimate value range of a24 is from 0 to FF__FFFFh.

[Example] The value of label LA is 9106H.

```
                                          PC      R3      PSW
                                          -       -       -
        MOV.W   R3,#0xFFFF                9004h   FFFFh   3040h
loop:                                     
        JMPF    LA      ;; JUMP LA        9106h   FFFFh   3040h
        NOP                               -       -       -
        NOP                               -       -       -
          .                               
          .                               
          .                               
LA:                                       
        INC     R3                        9108h   0000h   3003h
        NOP                               910Ah   0000h   3003h
```

MASK Rd, #imm16

Instruction code	[0 0 1 1 0 0 0 0][1 1 1 0 d3d2d1d0][i15 to i8][i7 to i0] 30E0H
Argument	Rd = 4bit(R select), imm16 = 16bit(immediate data)
Word count	2
Cycle count	4
Function	(Rd)←{(Rd) &~ #imm16} | { (Rx) & #imm16}, (PC)←(PC)+4
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers (overwrites), to Rd, only such bits of the general-purpose register (Rx) designated indirectly by bits 12 to 15 (N0 to N3) of the PSW that the value of the corresponding bits of immediate data imm16 is 1.

The legitimate value range of Rd is from R0 to R15 and that of imm16 is from 0 to FFFF.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0000	-	-	-	0000h	3	1	1	0	0
MOV.W	R0, #0x5555	5555h	-	-	0000h	0	0	0	0	0
MASK	R3, #0xFFFF	5555h	-	-	5555h	3	0	0	0	0
MOV.W	R1, #0x1200	5555h	1200h	-	5555h	1	1	0	0	0
MASK	R3, #0xFFFF	5555h	1200h	-	1200h	3	1	0	0	0
SWPB	R1	5555h	0012h	-	1200h	1	0	0	0	0
MASK	R3, #0xFF00	5555h	0012h	-	0000h	3	1	1	0	0
MOV.W	R0, #0x6789	6789h	0012h	-	0000h	0	0	0	0	0
MASK	R2, #0x1234	6789h	0012h	0200h	0000h	2	1	0	1	0
NOT	R0	9876h	0012h	0200h	0000h	0	0	0	0	1
MASK	R2, #0xEDCB	9876h	0012h	8A42h	0000h	2	0	0	1	1

MASK Rd, Rs

Instruction code	[0 0 1 1 0 0 1 1][s3s2s1s0d3d2d1d0] 3300H
Argument	Rd = 4bit(R select), Rs = 4bit(R select)
Word count	1
Cycle count	3
Function	(Rd)←{(Rd) & ~(Rs)} | {(Rx) & (Rs)}, (PC)←(PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers (overwrites), to Rd, only such bits of the general-purpose register (Rx) designated indirectly by bits 12 to 15 (N0 to N3) of the PSW that the value of the corresponding bits of the general-purpose register Rs is 1.
The legitimate value range of Rd is from R0 to R15 and that of Rs is from R0 to R15.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0000	-	-	-	0000h	3	1	1	0	0
MOV.W	R2, #0xFFFF	-	-	FFFFh	0000h	2	0	0	0	1
MOV.W	R0, #0x5555	5555h	-	FFFFh	0000h	0	0	0	0	0
MASK	R3, R2	5555h	-	FFFFh	5555h	3	0	0	0	0
MOV.W	R1, #0x1200	5555h	1200h	FFFFh	5555h	1	1	0	0	0
MASK	R3, R2	5555h	1200h	FFFFh	1200h	3	1	0	0	0
MOV.W	R2, #0xFF00	5555h	1200h	FF00h	1200h	2	1	0	0	0
SWPB	R1	5555h	0012h	FF00h	1200h	1	0	0	0	0
MASK	R3, R2	5555h	0012h	FF00h	0000h	3	1	1	0	0
MOV.W	R2, #0x1234	5555h	0012h	1234h	0000h	2	0	0	1	0
MOV.W	R0, #0x6789	6789h	0012h	1234h	0000h	0	0	0	0	0
MASK	R3, R2	6789h	0012h	1234h	0200h	3	1	0	1	0
NOT	R2	6789h	0012h	EDCBh	0200h	2	0	0	1	1
NOT	R0	9876h	0012h	EDCBh	0200h	0	0	0	0	1
MASK	R3, R2	9876h	0012h	EDCBh	8A42h	3	0	0	1	1

MOV Rd, Rs

Instruction code	[0 1 0 0 0 1 1 0][s3s2s1s0d3d2d1d0] 4600H
Argument	Rd = 4bit(R select), Rs = 4bit(R select)
Word count	1
Cycle count	1
Function	(Rd)←(Rs), (PC)←(PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers the contents of the general-purpose register Rs to the general-purpose register Rd. The legitimate value range of Rd is from R0 to R15 and that of Rs is from R0 to R15.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-
MOV.W	R0, #0x5555	5555h	-	-	-	0	0	0	0	0
MOV	R3, R0	5555h	-	-	5555h	3	0	0	0	0
MOV.W	R1, #0x1200	5555h	1200h	-	5555h	1	1	0	0	0
MOV	R3, R1	5555h	1200h	-	1200h	3	1	0	0	0
MOV.W	R2, #0x0000	5555h	1200h	0000h	1200h	2	1	1	0	0
MOV	R3, R2	5555h	1200h	0000h	0000h	3	1	1	0	0
MOV.W	R0, #0x5634	5634h	1200h	0000h	0000h	0	0	0	1	0
MOV	R3, R0	5634h	1200h	0000h	5634h	3	0	0	1	0
MOV.W	R1, #0x8118	5634h	8118h	0000h	5634h	1	0	0	0	1
MOV	R3, R1	5634h	8118h	0000h	8118h	3	0	0	0	1
MOV.W	R2, #0x5555	5634h	8118h	5555h	8118h	2	0	0	0	0
MOV	R3, R2	5634h	8118h	5555h	5555h	3	0	0	0	0

MOV. B (Rd), Rs

Instruction code	[0 1 1 1 0 0 1 0][d3d2d1d0 0 s2s1s0] 7200H
Argument	Rd = 4bit(R select), Rs = 3bit(R select)
Word count	1
Cycle count	2 or 3
Function	[Rd] ← Lobyte (Rs), (PC)←(PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers the lower-order 8 bits of the general-purpose register Rs to the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the contents of Rd.

The legitimate value range of Rd is from R0 to R15 and that of Rs is from R0 to R7.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50,#0x6666	66h	66h	-	-	-	-	-	0	0	0	0
MOV.W	R3, #0x0050	66h	66h	-	-	-	0050h	3	0	0	0	0
MOV.W	R0, #0x5555	66h	66h	5555h	-	-	0050h	0	0	0	0	0
MOV.B	(R3), R0	55h	66h	5555h	-	-	0050h	0	0	0	0	0
MOV.W	R1, #0x1200	55h	66h	5555h	1200h	-	0050h	1	1	0	0	0
MOV.B	(R3), R1	00h	66h	5555h	1200h	-	0050h	1	1	1	0	0
MOV.W	R2, #0x0000	00h	66h	0055h	1200h	0000h	0050h	2	1	1	0	0
MOV.B	(R3), R2	00h	66h	0055h	1200h	0000h	0050h	2	1	1	0	0
MOV.W	R0, #0x5634	00h	66h	5634h	1200h	0000h	0050h	0	0	0	1	0
MOV.B	(R3), R0	34h	66h	5634h	1200h	0000h	0050h	0	0	0	1	0
MOV.W	R1, #0x1881	34h	66h	5634h	1881h	0000h	0050h	1	0	0	0	0
MOV.B	(R3), R1	34h	66h	5634h	1881h	0000h	0050h	1	0	0	0	1
MOV.W	R2, #0x5555	34h	66h	5634h	1881h	5555h	0050h	2	0	0	0	0
MOV.B	(R3), R2	81h	66h	5634h	1881h	5555h	0050h	2	0	0	0	0

<Note>

This instruction takes 3 cycles to transfer the contents of Rs to the program memory (ROM).
However, no data can actually be transferred to ROM.

MOV. B (--Rd), Rs

Instruction code	[0 1 1 0 1 0 1 0][d3d2d1d0 0 s2s1s0] 6A00H
Argument	Rd = 4bit(R select), Rs = 3bit(R select)
Word count	1
Cycle count	2 or 3
Function	(Rd)←(Rd)-1, [Rd] ← Lobyte (Rs), (PC)←(PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction firstly subtracts 1 from the contents of the general-purpose register Rd. Subsequently, it transfers the lower-order 8 bits of the general-purpose register Rs to the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the contents of Rd.
The legitimate value range of Rd is from R0 to R15 and that of Rs is from R0 to R7.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50, #0x6666	66h	66h	-	-	-	-	-	0	0	0	0
MOV.W	R3, #0x0051	66h	66h	-	-	-	0051h	3	0	0	1	0
MOV.W	R0, #0x5555	66h	66h	5555h	-	-	0051h	0	0	0	0	0
MOV.B	(--R3), R0	55h	66h	5555h	-	-	0050h	0	0	0	0	0
INC	R3	55h	66h	5555h	-	-	0051h	3	0	0	1	0
MOV.W	R1, #0x1200	55h	66h	5555h	1200h	-	0051h	1	1	0	0	0
MOV.B	(--R3), R1	00h	66h	5555h	1200h	-	0050h	1	1	1	0	0
INC	R3	00h	66h	5555h	1200h	-	0051h	3	0	0	1	0
MOV.W	R2, #0x0000	00h	66h	5555h	1200h	0000h	0051h	2	1	1	0	0
MOV.B	(--R3), R2	00h	66h	5555h	1200h	0000h	0050h	2	1	1	0	0
INC	R3	00h	66h	5555h	1200h	0000h	0051h	3	0	0	1	0
MOV.W	R0, #0x5634	00h	66h	5634h	1200h	0000h	0051h	0	0	0	1	0
MOV.B	(--R3), R0	34h	66h	5634h	1200h	0000h	0050h	0	0	0	1	0
INC	R3	34h	66h	5634h	1200h	0000h	0051h	3	0	0	1	0
MOV.W	R1, #0x1881	34h	66h	5634h	1881h	0000h	0051h	1	0	0	0	0
MOV.B	(--R3), R1	81h	66h	5634h	1881h	0000h	0050h	1	0	0	0	1
INC	R3	81h	66h	5634h	1881h	0000h	0051h	3	0	0	1	0
MOV.W	R2, #0x5555	81h	66h	5634h	1881h	5555h	0051h	2	0	0	0	0
MOV.B	(--R3), R2	55h	66h	5634h	1881h	5555h	0050h	2	0	0	0	0
INC	R3	55h	66h	5634h	1881h	5555h	0051h	3	0	0	1	0

<Note>

This instruction takes 3 cycles to transfer the contents of Rs to the program memory (ROM).
However, no data can actually be transferred to ROM.

MOV. B (Rd, ±n), Rs

Instruction code	[0 1 1 1 0 0 1 0][d3d2d1d0 1 s2s1s0][0 0 0 0 n11 to n8][n7 to n0] 7208H
Argument	Rd = 4bit(R select), n = 12bit(signed), Rs = 3bit(R select)
Word count	2
Cycle count	3 or 4
Function	[(Rd±n)&FFFFh]←Lobyte (Rs), (PC)←(PC)+4
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers the lower-order 8 bits of the general-purpose register Rs to the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the address derived by the arithmetic operation＊1 performed on the contents of Rd and n.

The legitimate value range of Rd is from R0 to R15, that of Rs is from R0 to R7, and that of n is that of signed 12-bit data (-2048 to 2047).

＊1: Any carry or borrow resulting from a 16-bit arithmetic operation is ignored.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50,#0x6666	66h	66h	-	-	-	-	-	0	0	0	0
MOV.W	R3, #0x0000	66h	66h	-	-	-	0000h	3	1	1	0	0
MOV.W	R0, #0x5555	66h	66h	5555h	-	-	0000h	0	0	0	0	0
MOV.B	(R3,0x50), R0	55h	66h	5555h	-	-	0000h	0	0	0	0	0
MOV.W	R1, #0x1200	55h	66h	5555h	1200h	-	0000h	1	1	0	0	0
MOV.B	(R3,0x50), R1	00h	66h	5555h	1200h	-	0000h	1	1	1	0	0
MOV.W	R2, #0x0000	00h	66h	5555h	1200h	0000h	0000h	2	1	1	0	0
MOV.B	(R3,0x50), R2	00h	66h	5555h	1200h	0000h	0000h	2	1	1	0	0
MOV.W	R0, #0x5634	00h	66h	5634h	1200h	0000h	0000h	0	0	0	1	0
MOV.B	(R3,0x50), R0	34h	66h	5634h	1200h	0000h	0000h	0	0	0	1	0
MOV.W	R1, #0x1881	34h	66h	5634h	1881h	0000h	0000h	1	0	0	0	0
MOV.B	(R3,0x50), R1	81h	66h	5634h	1881h	0000h	0000h	1	0	0	0	1
MOV.W	R2, #0x5555	81h	66h	5634h	1881h	5555h	0000h	2	0	0	0	0
MOV.B	(R3,0x50), R2	55h	66h	5634h	1881h	5555h	0000h	2	0	0	0	0

<Note>

This instruction takes 4 cycles to transfer the contents of Rs to the program memory (ROM).
However, no data can actually be transferred to ROM.

MOV. B (--Rd, ±n), Rs

Instruction code	[0 1 1 0 1 0 1 0][d3d2d1d0 1 s2s1s0][0 0 0 0 n11 to n8][n7 to n0] 6A08H
Argument	Rd = 4bit(R select), n = 12bit(signed), Rs = 3bit(R select)
Word count	2
Cycle count	3 or 4
Function	(Rd)←(Rd)-1, [(Rd±n)&FFFFh] ← Lobyte(Rs), (PC)←(PC)+4
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction firstly subtracts 1 from the contents of the general-purpose register Rd. Subsequently, it transfers the lower-order 8 bits of the general-purpose register Rs to the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the address derived by the arithmetic operation*1 performed on the contents of Rd and n.

The legitimate value range of Rd is from R0 to R15, that of Rs is from R0 to R7, and that of n is that of 12-bit data (-2048 to 2047).

*1: Any carry or borrow resulting from a 16-bit arithmetic operation is ignored.

[Example]

	RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
	-	-	-	-	-	-	-	-	-	-	-
MOV.W 0x50, #0x6666	66h	66h	-	-	-	-	-	0	0	0	0
MOV.W R3, #0x0001	66h	66h	-	-	-	0001h	3	0	0	1	0
MOV.W R0, #0x5555	66h	66h	5555h	-	-	0001h	0	0	0	0	0
MOV.B (--R3,0x50), R0	55h	66h	5555h	-	-	0000h	0	0	0	0	0
INC R3	55h	66h	5555h	-	-	0001h	3	0	0	1	0
MOV.W R1, #0x1200	55h	66h	5555h	1200h	-	0001h	1	1	0	0	0
MOV.B (--R3,0x50), R1	00h	66h	5555h	1200h	-	0000h	1	1	1	0	0
INC R3	00h	66h	5555h	1200h	-	0001h	3	0	0	1	0
MOV.W R2, #0x0000	00h	66h	5555h	1200h	0000h	0001h	2	1	1	0	0
MOV.B (--R3,0x50), R2	00h	66h	5555h	1200h	0000h	0000h	2	1	1	0	0
INC R3	00h	66h	5555h	1200h	0000h	0001h	3	0	0	1	0
MOV.W R0, #0x5634	00h	66h	5634h	1200h	0000h	0001h	0	0	0	1	0
MOV.B (--R3,0x50), R0	34h	66h	5634h	1200h	0000h	0000h	0	0	0	1	0
INC R3	34h	66h	5634h	1200h	0000h	0001h	3	0	0	1	0
MOV.W R1, #0x1881	34h	66h	5634h	1881h	0000h	0001h	1	0	0	0	0
MOV.B (--R3,0x50), R1	81h	66h	5634h	1881h	0000h	0000h	1	0	0	0	1
INC R3	81h	66h	5634h	1881h	0000h	0001h	3	0	0	1	0
MOV.W R2, #0x5555	81h	66h	5634h	1881h	5555h	0001h	2	0	0	0	0
MOV.B (--R3, 0x50), R2	55h	66h	5634h	1881h	5555h	0000h	2	0	0	0	0
INC R3	55h	66h	5634h	1881h	5555h	0001h	3	0	0	1	0

<Note>

This instruction takes 4 cycles to transfer the contents of Rs to the program memory (ROM).
However, no data can actually be transferred to ROM.

MOV. B (Rd+ +), Rs

Instruction code	[0 1 1 0 0 0 1 0][d3d2d1d0 0 s2s1s0] 6200H
Argument	Rd = 4bit(R select), Rs = 3bit(R select)
Word count	1
Cycle count	2 or 3
Function	[Rd] ← Lobyte(Rs), (Rd)←(Rd)+1, (PC)←(PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers the lower-order 8 bits of the general-purpose register Rs to the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the general-purpose register Rd. Subsequently, the instruction increments the contents of Rd by +1.
The legitimate value range of Rd is from R0 to R15 and that of Rs is from R0 to R7.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50,#0x6666	66h	66h	-	-	-	-	-	0	0	0	0
MOV.W	R3, #0x0050	66h	66h	-	-	-	0050h	3	0	0	0	0
MOV.W	R0, #0x5555	66h	66h	5555h	-	-	0050h	0	0	0	0	0
MOV.B	(R3++), R0	55h	66h	5555h	-	-	0051h	0	0	0	0	0
DEC	R3	55h	66h	5555h	-	-	0050h	3	0	0	0	0
MOV.W	R1, #0x1200	55h	66h	5555h	1200h	-	0050h	1	1	0	0	0
MOV.B	(R3++), R1	00h	66h	5555h	1200h	-	0051h	1	1	1	0	0
DEC	R3	00h	66h	5555h	1200h	-	0050h	3	0	0	0	0
MOV.W	R2, #0x0000	00h	66h	5555h	1200h	0000h	0050h	2	1	1	0	0
MOV.B	(R3++), R2	00h	66h	5555h	1200h	0000h	0051h	2	1	1	0	0
DEC	R3	00h	66h	5555h	1200h	0000h	0050h	3	0	0	0	0
MOV.W	R0, #0x5634	00h	66h	5634h	1200h	0000h	0050h	0	0	0	1	0
MOV.B	(R3++), R0	34h	66h	5634h	1200h	0000h	0051h	0	0	0	1	0
DEC	R3	34h	66h	5634h	1200h	0000h	0050h	3	0	0	0	0
MOV.W	R1, #0x1881	34h	66h	5634h	1881h	0000h	0050h	1	0	0	0	0
MOV.B	(R3++), R1	81h	66h	5634h	1881h	0000h	0051h	1	0	0	0	1
DEC	R3	81h	66h	5634h	1881h	0000h	0050h	3	0	0	0	0
MOV.W	R2, #0x5555	81h	66h	5634h	1881h	5555h	0050h	2	0	0	0	0
MOV.B	(R3++), R2	55h	66h	5634h	1881h	5555h	0051h	2	0	0	0	0
DEC	R3	55h	66h	5634h	1881h	5555h	0050h	2	0	0	0	0

<Note>

This instruction takes 3 cycles to transfer the contents of Rs to the program memory (ROM).
However, no data can actually be transferred to ROM.

MOV. B (Rd + +, ±n), Rs

Instruction code	[0 1 1 0 0 0 1 0][d3d2d1d0 1 s2s1s0][0 0 0 0 n11 to n8][n7 to n0] 6208H
Argument	Rd = 4bit(R select), n = 12bit(signed), Rs = 3bit(R select)
Word count	2
Cycle count	3 or 4
Function	[(Rd±n)&FFFFh] ← Lobyte(Rs), (Rd)←(Rd)+1, (PC)←(PC)+4
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers the lower-order 8 bits of the general-purpose register Rs to the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the address derived by the arithmetic operation *1 performed on the contents of Rd and n. Subsequently, the instruction increments the contents of Rd by +1.

The legitimate value range of Rd is from R0 to R15, that of Rs is from R0 to R7, and that of n is that of signed 12-bit data (-2048 to 2047).

*1: Any carry or borrow resulting from a 16-bit arithmetic operation is ignored..

[Example]

	RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
	-	-	-	-	-	-	-	-	-	-	-
MOV.W 0x50, #0x6666	66h	66h	-	-	-	-	-	0	0	0	0
MOV.W R3, #0x0000	66h	66h	-	-	-	0000h	3	1	1	0	0
MOV.W R0, #0x5555	66h	66h	5555h	-	-	0000h	0	0	0	0	0
MOV.B (R3++, 0x50), R0	55h	66h	5555h	-	-	0001h	0	0	0	0	0
DEC R3	55h	66h	5555h	-	-	0000h	3	1	1	0	0
MOV.W R1, #0x1200	55h	66h	5555h	1200h	-	0000h	1	1	0	0	0
MOV.B (R3++, 0x50), R1	00h	66h	5555h	1200h	-	0001h	1	1	1	0	0
DEC R3	00h	66h	5555h	1200h	-	0000h	3	1	1	0	0
MOV.W R2, #0x0000	00h	66h	5555h	1200h	0000h	0000h	2	1	1	0	0
MOV.B (R3++, 0x50), R2	00h	66h	5555h	1200h	0000h	0001h	2	1	1	0	0
DEC R3	00h	66h	5555h	1200h	0000h	0000h	3	1	1	0	0
MOV.W R0, #0x5634	00h	66h	5634h	1200h	0000h	0000h	0	0	0	1	0
MOV.B (R3++, 0x50), R0	34h	66h	5634h	1200h	0000h	0001h	0	0	0	1	0
DEC R3	34h	66h	5634h	1200h	0000h	0000h	3	1	1	0	0
MOV.W R1, #0x1881	34h	66h	5634h	1881h	0000h	0000h	1	0	0	0	0
MOV.B (R3++, 0x50), R1	81h	66h	5634h	1881h	0000h	0001h	1	0	0	0	1
DEC R3	81h	66h	5634h	1881h	0000h	0000h	3	1	1	0	0
MOV.W R2, #0x5555	81h	66h	5634h	1881h	5555h	0000h	2	0	0	0	0
MOV.B (R3++, 0x50), R2	55h	66h	5634h	1881h	5555h	0001h	2	0	0	0	0
DEC R3	55h	66h	5634h	1881h	5555h	0000h	3	1	1	0	0

<Note>

This instruction takes 4 cycles to transfer the contents of Rs to the program memory (ROM).
However, no data can actually be transferred to ROM.

MOV. B m16, #imm16

Instruction code	[0 1 1 1 1 0 X 0][m7m6m5m4m3m2m1m0][i15 to i8][i7 to i0] 7800H(RAM), 7A00H(SFR)
Argument	m16 = 16bit (lower 8bit valid for operation code) imm16 = 16bit (immediate data)
Word count	2
Cycle count	2
Function	(m16)←Lobyte #imm16, (PC)←(PC)+4
Affected flags	Z8, Z16, P, S

[Description]

This instruction transfers the lower-order 8 bits of immediate data imm16 to the data memory (RAM) location or SFR (one of the registers dedicated to control the internal peripheral devices) addressed by m16.

The compiler generates the instruction code while regarding RAM or SFR as the destination of transfer according to the value of m16 (first operand data).

・ When specifying a RAM location, specify m16 with a value from 00H to FFH (0000H to 00FFH). It is disallowed to specify a RAM address not lower than 100H.

・ When specifying a SFR, specify m16 with a value from 7F00H to 7FFFH.

The basic types of generated instruction code are 7800H (RAM) and 7A00H (SFR), respectively, The lower-order 8 bits of m16 are reflected in the behavior of the instruction code.

imm16 (second operand data) may be 16-bit data. Since this instruction is a byte transfer instruction, however, the higher-order 8 bits of imm16 is irrelevant to the actual behavior of the instruction. The MOV. W instruction should be used to handle 16-bit data.

[Example]

	RAM (50h)	Z8	Z16	P	S
	-	-	-	-	-
MOV.B 0x50, #0x55	55h	0	0	0	0
MOV.B 0x50, #0x00	00h	1	1	0	0
MOV.B 0x50, #0x34	34h	0	0	1	0
MOV.B 0x50, #0x81	81h	0	0	0	1
MOV.B 0x50, #0x55	55h	0	0	0	0

MOV. B m16, Rs

Instruction code	[1 0 X 1 s2s1s0 0][m7m6m5m4m3m2m1m0] 9000H(RAM), B000H(SFR)
Argument	m16 = 16bit(Lower 8bit valid for operation code), Rs = 3bit(R select)
Word count	1
Cycle count	1
Function	(m16) ← Lobyte(Rs), (PC)←(PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers the lower-order 8 bits of the general-purpose register Rs to the data memory (RAM) location or SFR (one of the registers dedicated to control the internal peripheral devices) addressed by m16.

The compiler generates the instruction code while regarding RAM or SFR as the destination of transfer according to the value of m16 (first operand data).

・ When specifying a RAM location, specify m16 with a value from 00H to FFH (0000H to 00FFH). It is disallowed to specify a RAM address not lower than 100H.

・ When specifying a SFR, specify m16 with a value from 7F00H to 7FFFH.

The basic types of generated instruction code are 9000H (RAM) and B000H (SFR), respectively, The lower-order 8 bits of m16 are reflected in the behavior of the instruction code.

Rs (second operand data) may be 16-bit data. Since this instruction is a byte transfer instruction, however, the higher-order 8 bits of Rs is irrelevant to the actual behavior of the instruction. The MOV. W instruction should be used to handle 16-bit data.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	R0, #0x5555	-	-	5555h	-	-	-	0	0	0	0	0
MOV.W	R3, #0x5555	-	-	5555h	-	-	5555h	3	0	0	0	0
MOV.B	0x50, R0	55h	-	5555h	-	-	5555h	0	0	0	0	0
MOV.W	R1, #0x1200	55h	-	5555h	1200h	-	5555h	1	1	0	0	0
MOV.W	R3, #0x6666	55h	-	5555h	1200h	-	6666h	3	0	0	0	0
MOV.B	0x50, R1	00h	-	5555h	1200h	-	6666h	1	1	1	0	0
MOV.W	R2, #0x0000	00h	-	5555h	1200h	0000h	6666h	2	1	1	0	0
MOV.W	R3, #0x3333	00h	-	5555h	1200h	0000h	3333h	3	0	0	0	0
MOV.B	0x50, R2	00h	-	5555h	1200h	0000h	3333h	2	1	1	0	0
MOV.W	R0, #0x5634	00h	-	5634h	1200h	0000h	3333h	0	0	0	1	0
MOV.W	R3, #0x6655	00h	-	5634h	1200h	0000h	6655h	3	0	0	0	0
MOV.B	0x50, R0	34h	-	5634h	1200h	0000h	6655h	0	0	0	1	0
MOV.W	R1, #0x1881	34h	-	5634h	1881h	0000h	6655h	1	0	0	0	0
MOV.W	R3, #0x3366	34h	-	5634h	1881h	0000h	3366h	3	0	0	0	0
MOV.B	0x50, R1	81h	-	5634h	1881h	0000h	3366h	1	0	0	0	1
MOV.W	R2, #0x5555	81h	-	5634h	1881h	5555h	3366h	2	0	0	0	0
MOV.W	R3, #0x6355	81h	-	5634h	1881h	5555h	6355h	3	0	0	0	0
MOV.B	0x50, R2	55h	-	5634h	1881h	5555h	6355h	2	0	0	0	0

MOV. B Rd, (Rs)

Instruction code	[0 1 1 1 0 0 0 0][s3s2s1s0 0 d2d1d0] 7000H
Argument	Rd = 3bit(R select), Rs = 4bit(R select)
Word count	1
Cycle count	2 or 3
Function	(Rd) ← Lobyte [Rs], (PC)←(PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers the contents of the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by Rs to the lower-order 8 bit positions of Rd.
The legitimate value range of Rd is from R0 to R15 and that of Rs is from R0 to R7.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0050	-	-	-	-	-	0050h	3	0	0	0	0
MOV.W	0x50, #0x5555	55h	55h	-	-	-	0050h	3	0	0	0	0
MOV.B	R0, (R3)	55h	55h	0055h	-	-	0050h	0	0	0	0	0
MOV.W	0x50, #0x1200	00h	12h	0055h	-	-	0050h	0	1	0	0	0
MOV.B	R1, (R3)	00h	12h	0055h	0000h	-	0050h	1	1	1	0	0
MOV.W	0x50, #0x0000	00h	00h	0055h	0000h	-	0050h	1	1	1	0	0
MOV.B	R2, (R3)	00h	00h	0055h	0000h	0000h	0050h	2	1	1	0	0
MOV.W	0x50, #0x5634	34h	56h	0055h	0000h	0000h	0050h	2	0	0	1	0
MOV.B	R0, (R3)	34h	56h	0034h	0000h	0000h	0050h	0	0	0	1	0
MOV.W	0x50, #0x1881	81h	18h	0034h	0000h	0000h	0050h	0	0	0	0	0
MOV.B	R1, (R3)	81h	18h	0034h	0081h	0000h	0050h	1	0	0	0	1
MOV.W	0x50, #0x5555	55h	55h	0034h	0081h	0000h	0050h	1	0	0	0	0
MOV.B	R2, (R3)	55h	55h	0034h	0081h	0055h	0050h	2	0	0	0	0

<Note>

The higher-order 8 bits of Rd are loaded with 00H.

This instruction takes 3 cycles to transfer the contents of program memory (ROM) to Rd.

MOV. B Rd, (--Rs)

Instruction code	[0 1 1 0 1 0 0 0][s3s2s1s0 0 d2d1d0] 6800H
Argument	Rd = 3bit(R select), Rs = 4bit(R select)
Word count	1
Cycle count	2 or 3
Function	(Rs) ← (Rs) – 1, (Rd) ← Lobyte[Rs], (PC) ← (PC) + 2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction firstly subtracts 1 from the contents of the general-purpose register Rs. Subsequently, the instruction transfers the contents of the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by Rs to the lower-order 8 bit positions of Rd.
The legitimate value range of Rd is from R0 to R15 and that of Rs is from R0 to R7.

[Example]

	RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
	-	-	-	-	-	-	-	-	-	-	-
MOV.W R3, #0x0051	-	-	-	-	-	0051h	3	0	0	1	0
MOV.W 0x50, #0x5555	55h	55h	-	-	-	0051h	3	0	0	0	0
MOV.B R0, (--R3)	55h	55h	0055h	-	-	0050h	0	0	0	0	0
INC R3	55h	55h	0055h	-	-	0051h	3	0	0	1	0
MOV.W 0x50, #0x1200	00h	12h	0055h	-	-	0051h	3	1	0	0	0
MOV.B R1, (--R3)	00h	12h	0055h	0000h	-	0050h	1	1	1	0	0
INC R3	00h	12h	0055h	0000h	-	0051h	3	0	0	1	0
MOV.W 0x50, #0x0000	00h	00h	0055h	0000h	-	0051h	3	1	1	0	0
MOV.B R2, (--R3)	00h	00h	0055h	0000h	0000h	0050h	2	1	1	0	0
INC R3	00h	00h	0055h	0000h	0000h	0051h	3	0	0	1	0
MOV.W 0x50, #0x5634	34h	56h	0055h	0000h	0000h	0051h	3	0	0	1	0
MOV.B R0, (--R3)	34h	56h	0034h	0000h	0000h	0050h	0	0	0	1	0
INC R3	34h	56h	0034h	0000h	0000h	0051h	3	0	0	1	0
MOV.W 0x50, #0x1881	81h	18h	0034h	0000h	0000h	0051h	3	0	0	0	0
MOV.B R1, (--R3)	81h	18h	0034h	0081h	0000h	0050h	1	0	0	0	1
INC R3	81h	18h	0034h	0081h	0000h	0051h	3	0	0	1	0
MOV.W 0x50, #0x5555	55h	55h	0034h	0081h	0000h	0051h	3	0	0	0	0
MOV.B R2, (--R3)	55h	55h	0034h	0081h	0055h	0050h	2	0	0	0	0
INC R3	55h	55h	0034h	0081h	0055h	0051h	3	0	0	1	0

<Note>

The higher-order 8 bits of Rd are loaded with 00H.

This instruction takes 3 cycles to transfer the contents of program memory (ROM) to Rd.

MOV. B Rd, (Rs, ±n)

Instruction code	[0 1 1 1 0 0 0 0][s3s2s1s0 1 d2d1d0][0 0 0 0 n11 to n8][n7 to n0] 7008H
Argument	Rd = 3bit(R select), Rs = 4bit(R select), n = 12bit(signed)
Word count	2
Cycle count	3 or 4
Function	(Rd) ← Lobyte [(Rs±n)&FFFFh], (PC)←(PC)+4
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers the contents of the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the address derived by the arithmetic operation *1 performed on the contents of Rs and n to the lower-order 8 bit positions of Rd.

The legitimate value range of Rd is from R0 to R7, that of Rs is from 0 to R15, and that of n is that of signed 12-bit data (-2048 to 2047).

*1: Any carry or borrow resulting from a 16-bit arithmetic operation is ignored..

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0000	-	-	-	-	-	0000h	3	1	1	0	0
MOV.W	0x50, #0x5555	55h	55h	-	-	-	0000h	3	0	0	0	0
MOV.B	R0, (R3, 0x50)	55h	55h	0055h	-	-	0000h	0	0	0	0	0
MOV.W	0x50, #0x1200	00h	12h	0055h	-	-	0000h	0	1	0	0	0
MOV.B	R1, (R3, 0x50)	00h	12h	0055h	0000h	-	0000h	1	1	1	0	0
MOV.W	0x50, #0x0000	00h	00h	0055h	0000h	-	0000h	1	1	1	0	0
MOV.B	R2, (R3, 0x50)	00h	00h	0055h	0000h	0000h	0000h	2	1	1	0	0
MOV.W	0x50, #0x5634	34h	56h	0055h	0000h	0000h	0000h	2	0	0	1	0
MOV.B	R0, (R3, 0x50)	34h	56h	0034h	0000h	0000h	0000h	0	0	0	1	0
MOV.W	0x50, #0x1881	18h	81h	0034h	0000h	0000h	0000h	0	0	0	0	0
MOV.B	R1, (R3, 0x50)	18h	81h	0034h	0081h	0000h	0000h	1	0	0	0	1
MOV.W	0x50, #0x5555	55h	55h	0034h	0081h	0000h	0000h	1	0	0	0	0
MOV.B	R2, (R3, 0x50)	55h	55h	0034h	0081h	0055h	0000h	2	0	0	0	0

<Note>

The higher-order 8 bits of Rd are loaded with 00H.

This instruction takes 4 cycles to transfer the contents of program memory (ROM) to Rd.

MOV. B Rd, (--Rs, ±n)

Instruction code	[0 1 1 0 1 0 0 0][s3s2s1s0 1 d2d1d0][0 0 0 0 n11 to n8][n7 to n0]	6808H
Argument	Rd = 3bit(R select), Rs = 4bit(R select), n = 12bit(signed)	
Word count	2	
Cycle count	3 or 4	
Function	(Rs)←(Rs)-1, (Rd)←Lobyte [(Rs±n)&FFFFh], (PC)←(PC)+4	
Affected flags	Z8, Z16, P, S, N0 to N3	

[Description]

This instruction firstly subtracts 1 from the contents of the general-purpose register Rs.
Subsequently, the instruction transfers the contents of the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the address derived by the arithmetic operation＊1 performed on the contents of Rs and n to the lower-order 8 bit positions of Rd.
The legitimate value range of Rd is from R0 to R7, that of Rs is from R0 to R15, and that of n is that of signed 12-bit data (-2048 to 2047).

*1: Any carry or borrow resulting from a 16-bit arithmetic operation is ignored..

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0001	-	-	-	-	-	0001h	3	0	0	1	0
MOV.W	0x50, #0x5555	55h	55h	-	-	-	0001h	3	0	0	0	0
MOV.B	R0,(--R3, 0x50)	55h	55h	0055h	-	-	0000h	0	0	0	0	0
INC	R3	55h	55h	0055h	-	-	0001h	3	0	0	1	0
MOV.W	0x50, #0x1200	00h	12h	0055h	-	-	0001h	3	1	0	0	0
MOV.B	R1,(--R3, 0x50)	00h	12h	0055h	0000h	-	0000h	1	1	1	0	0
INC	R3	00h	12h	0055h	0000h	-	0001h	3	0	0	1	0
MOV.W	0x50, #0x0000	00h	00h	0055h	0000h	-	0001h	3	1	1	0	0
MOV.B	R2,(--R3, 0x50)	00h	00h	0055h	0000h	0000h	0000h	2	1	1	0	0
INC	R3	00h	00h	0055h	0000h	0000h	0001h	3	0	0	1	0
MOV.W	0x50, #0x5634	34h	56h	0055h	0000h	0000h	0001h	3	0	0	1	0
MOV.B	R0,(--R3, 0x50)	34h	56h	0034h	0000h	0000h	0000h	0	0	0	1	0
INC	R3	34h	56h	0034h	0000h	0000h	0001h	3	0	0	1	0
MOV.W	0x50, #0x1881	18h	81h	0034h	0000h	0000h	0001h	3	0	0	0	0
MOV.B	R1,(--R3, 0x50)	18h	81h	0034h	0081h	0000h	0000h	1	0	0	0	1
INC	R3	18h	81h	0034h	0081h	0000h	0001h	3	0	0	1	0
MOV.W	0x50, #0x5555	55h	55h	0034h	0081h	0000h	0001h	3	0	0	0	0
MOV.B	R2,(--R3, 0x50)	55h	55h	0034h	0081h	0055h	0000h	2	0	0	0	0
INC	R3	55h	55h	0034h	0081h	0055h	0001h	3	0	0	1	0

<Note>

The higher-order 8 bits of Rd are loaded with 00H.
This instruction takes 4 cycles to transfer the contents of program memory (ROM) to Rd.

MOV. B Rd, (Rs + +)

Instruction code	[0 1 1 0 0 0 0 0][s3s2s1s0 0 d2d1d0] 6000H
Argument	Rd = 3bit(R select), Rs = 4bit(R select)
Word count	1
Cycle count	2 or 3
Function	(Rd) ← Lobyte [Rs], (Rs) ← (Rs) + 1, (PC)←(PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers the contents of the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by Rs to the lower-order 8 bit positions of Rd. Subsequently, the instruction increments the contents of Rs by 1.

The legitimate value range of Rd is from R0 to R7 and that of Rs is from R0 to R15.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0050	-	-	-	-	-	0050h	3	0	0	0	0
MOV.W	0x50, #0x5555	55h	55h	-	-	-	0050h	3	0	0	0	0
MOV.B	R0, (R3++)	55h	55h	0055h	-	-	0051h	0	0	0	0	0
DEC	R3	55h	55h	0055h	-	-	0050h	3	0	0	0	0
MOV.W	0x50, #0x1200	00h	12h	0055h	-	-	0050h	3	1	0	0	0
MOV.B	R1, (R3++)	00h	12h	0055h	0000h	-	0051h	1	1	1	0	0
DEC	R3	00h	12h	0055h	0000h	-	0050h	3	0	0	0	0
MOV.W	0x50, #0x0000	00h	00h	0055h	0000h	-	0050h	3	1	1	0	0
MOV.B	R2, (R3++)	00h	00h	0055h	0000h	0000h	0051h	2	1	1	0	0
DEC	R3	00h	00h	0055h	0000h	0000h	0050h	3	0	0	0	0
MOV.W	0x50, #0x5634	34h	56h	0055h	0000h	0000h	0050h	3	0	0	1	0
MOV.B	R0, (R3++)	34h	56h	0034h	0000h	0000h	0051h	0	0	0	1	0
DEC	R3	34h	56h	0034h	0000h	0000h	0050h	3	0	0	0	0
MOV.W	0x50, #0x1881	81h	18h	0034h	0000h	0000h	0050h	3	0	0	0	0
MOV.B	R1, (R3++)	81h	18h	0034h	0081h	0000h	0051h	1	0	0	0	1
DEC	R3	81h	18h	0034h	0081h	0000h	0050h	3	0	0	0	0
MOV.W	0x50, #0x5555	55h	55h	0034h	0081h	0000h	0050h	3	0	0	0	0
MOV.B	R2, (R3++)	55h	55h	0034h	0081h	0055h	0051h	2	0	0	0	0
DEC	R3	55h	55h	0034h	0081h	0055h	0050h	3	0	0	0	0

<Note>

The higher-order 8 bits of Rd are loaded with 00H.

This instruction takes 3 cycles to transfer the contents of program memory (ROM) to Rd.

MOV. B Rd, (Rs + +, ±n)

Instruction code	[0 1 1 0 0 0 0 0][s3s2s1s0 1 d2d1d0][0 0 0 0 n11 to n8][n7 to n0]	6008H
Argument	Rd = 3bit(R select), Rs = 4bit(R select), n = 12bit(signed)	
Word count	2	
Cycle count	3 or 4	
Function	(Rd) ← Lobyte[(Rs±n)&FFFFh], (Rs) ← (Rs) + 1, (PC)←(PC)+4	
Affected flags	Z8, Z16, P, S, N0 to N3	

[Description]

This instruction transfers the contents of the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the address derived by the arithmetic operation *1 performed on the contents of Rs and n to the lower-order 8 bit positions of Rd. Subsequently, the instruction increments the contents of Rs by 1.

The legitimate value range of Rd is from R0 to R7, that of Rs is from 0 to R15, and that of n is that of signed 12-bit data (-2048 to 2047).

*1: Any carry or borrow resulting from a 16-bit arithmetic operation is ignored..

[Example]

	RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
	-	-	-	-	-	-	-	-	-	-	-
MOV.W R3, #0x0000	-	-	-	-	-	0000h	3	1	1	0	0
MOV.W 0x50, #0x5555	55h	55h	-	-	-	0000h	3	0	0	0	0
MOV.B R0, (R3++, 0x50)	55h	55h	0055h	-	-	0001h	0	0	0	0	0
DEC R3	55h	55h	0055h	-	-	0000h	3	1	1	0	0
MOV.W 0x50, #0x1200	00h	12h	0055h	-	-	0000h	3	1	0	0	0
MOV.B R1, (R3++, 0x50)	00h	12h	0055h	0000h	-	0001h	1	1	1	0	0
DEC R3	00h	12h	0055h	0000h	-	0000h	3	1	1	0	0
MOV.W 0x50, #0x0000	00h	00h	0055h	0000h	-	0000h	3	1	1	0	0
MOV.B R2, (R3++, 0x50)	00h	00h	0055h	0000h	0000h	0001h	2	1	1	0	0
DEC R3	00h	00h	0055h	0000h	0000h	0000h	3	1	1	0	0
MOV.W 0x50, #0x5634	34h	56h	0055h	0000h	0000h	0000h	3	0	0	1	0
MOV.B R0, (R3++, 0x50)	34h	56h	0034h	0000h	0000h	0001h	0	0	0	1	0
DEC R3	34h	56h	0034h	0000h	0000h	0000h	3	1	1	0	0
MOV.W 0x50, #0x1881	81h	18h	0034h	0000h	0000h	0000h	3	0	0	0	0
MOV.B R1, (R3++, 0x50)	81h	18h	0034h	0081h	0000h	0001h	1	0	0	0	1
DEC R3	81h	18h	0034h	0081h	0000h	0000h	3	1	1	0	0
MOV.W 0x50, #0x5555	55h	55h	0034h	0081h	0000h	0000h	3	0	0	0	0
MOV.B R2, (R3++, 0x50)	55h	55h	0034h	0081h	0055h	0001h	2	0	0	0	0
DEC R3	55h	55h	0034h	0081h	0055h	0000h	3	1	1	0	0

<Note>

The higher-order 8 bits of Rd are loaded with 00H.

This instruction takes 4 cycles to transfer the contents of program memory (ROM) to Rd.

MOV. B Rd, m16

Instruction code	[1 0 X 0 d2d1d0 0][m7m6m5m4m3m2m1m0] 8000H(RAM), A000H(SFR)
Argument	Rd = 3bit(R select), m16 = 16bit(Lower 8bit valid for operation code)
Word count	1
Cycle count	1
Function	(Rd) ← Lobyte (m16), (PC) ← (PC) + 2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers the contents of data memory location designated by m16 to the lower-order 8 bit positions of the general-purpose register Rd.

The compiler generates the instruction code while regarding RAM or SFR as the destination of transfer according to the value of m16 (second operand data).

・ When specifying a RAM location, specify m16 with a value from 00H to FFH (0000H to 00FFH). It is disallowed to specify a RAM address not lower than 100H.

・ When specifying a SFR, specify m16 with a value from 7F00H to 7FFFH.

The basic types of generated instruction code are 8000H (RAM) and A000H (SFR), respectively, The lower-order 8 bits of m16 are reflected in the behavior of the instruction code.

Rd (first operand data) may be 16-bit data. Since this instruction is a byte transfer instruction, however, the higher-order 8 bits of Rd is irrelevant to the actual behavior of the instruction. The MOV. W instruction should be used to handle 16-bit data.

The legitimate value range of Rd is from R0 to R7.

[Example]

	RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
	-	-	-	-	-	-	-	-	-	-	-
MOV.W 0x50, #0x3C55	55h	3Ch	-	-	-	-	-	0	0	0	0
MOV.W r3, #0x5555	55h	3Ch	-	-	-	5555h	3	0	0	0	0
MOV.B r0, 0x50	55h	3Ch	0055h	-	-	5555h	0	0	0	0	0
MOV.B 0x50, #0x00	00h	3Ch	0055h	-	-	5555h	0	1	1	0	0
MOV.W r3, #0x6666	00h	3Ch	0055h	-	-	6666h	3	0	0	0	0
MOV.B r1, 0x50	00h	3Ch	0055h	0000h	-	6666h	1	1	1	0	0
MOV.B 0x50, #0x34	34h	3Ch	0055h	0000h	-	6666h	1	0	0	1	0
MOV.W R3, #0x3333	34h	3Ch	0055h	0000h	-	3333h	3	0	0	0	0
MOV.B R2, 0x50	34h	3Ch	0055h	0000h	0034h	3333h	2	0	0	1	0
MOV.B 0x50, #0x81	81h	3Ch	0055h	0000h	0034h	3333h	2	0	0	0	1
MOV.W R3, #0x5555	81h	3Ch	0055h	0000h	0034h	5555h	3	0	0	0	0
MOV.B R0, 0x50	81h	3Ch	0081h	0000h	0034h	5555h	0	0	0	0	1
MOV.B 0x50, #0x55	55h	3Ch	0081h	0000h	0034h	5555h	0	0	0	0	0
MOV.W R3, #0x6355	55h	3Ch	0081h	0000h	0034h	6355h	3	0	0	0	0
MOV.B R1, 0x50	55h	3Ch	0081h	0055h	0034h	6355h	1	0	0	0	0

MOV. B Rd, RxH

Instruction code	[0 0 1 1 0 0 0 0][1 1 0 1 d3d2d1d0] 30D0H
Argument	Rd = 4bit(R select)
Word count	1
Cycle count	1
Function	(Rd) ← { Hibyte(Rx) | Lobyte(Rd) }, (PC)←(PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers the higher-order 8 bits (RxH) of the general-purpose register designated indirectly by bits 12 to 15 (N0 to N3) of the PSW to the higher-order 8 bit positions of Rd.
The legitimate value range of Rd is from R0 to R15.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0xFFFF	-	-	-	-	-	FFFFh	3	0	0	0	1
MOV.W	R0, #0x0000	-	-	0000h	-	-	FFFFh	0	1	1	0	0
MOV.W	0x50, #0x6666	66h	66h	0000h	-	-	FFFFh	0	0	0	0	0
MOV.B	R3, Rxh	66h	66h	0000h	-	-	00FFH	3	0	0	0	0
SWPB	R3	66h	66h	0000h	-	-	FF00h	3	1	0	0	1
MOV.W	R1, #0x0012	66h	66h	0000h	0012h	-	FF00h	1	0	0	0	0
MOV.B	R3, Rxh	66h	66h	0000h	0012h	-	0000h	3	1	1	0	0
MOV.W	R2, #0x8967	66h	66h	0000h	0012h	8967h	0000h	2	0	0	0	1
MOV.B	R3, Rxh	66h	66h	0000h	0012h	8967h	8900h	3	0	0	1	1
SWPB	R3	66h	66h	0000h	0012h	8967h	0089h	3	0	0	1	0
MOV.W	R0, #0x5634	66h	66h	5634h	0012h	8967h	0089h	0	0	0	1	0
MOV.B	R3, Rxh	66h	66h	5634h	0012h	8967h	5689h	3	0	0	1	0

MOV. B Rd, RxL

Instruction code	[0 0 1 1 0 0 0 0][1 1 0 0 d3d2d1d0] 30C0H
Argument	Rd = 4bit(R select)
Word count	1
Cycle count	1
Function	(Rd) ← { Lobyte(Rx) | Hibyte(Rd) }, (PC)←(PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers the lower-order 8 bits (RxL) of the general-purpose register designated indirectly by bits 12 to 15 (N0 to N3) of the PSW to the lower-order 8 bit positions of Rd.
The legitimate value range of Rd is from R0 to R15.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0xFFFF	-	-	-	-	-	FFFFh	3	0	0	0	1
MOV.W	R0, #0x0000	-	-	0000h	-	-	FFFFh	0	1	1	0	0
MOV.W	0x50, #0x6666	66h	66h	0000h	-	-	FFFFh	0	0	0	0	0
MOV.B	R3, Rxl	66h	66h	0000h	-	-	FF00H	3	1	0	0	1
SWPB	R3	66h	66h	0000h	-	-	00FFh	3	0	0	0	0
MOV.W	R1, #0x1200	66h	66h	0000h	1200h	-	00FFh	1	1	0	0	0
MOV.B	R3, Rxl	66h	66h	0000h	1200h	-	0000h	3	1	1	0	0
MOV.W	R2, #0x6789	66h	66h	0000h	1200h	6789h	0000h	2	0	0	0	0
MOV.B	R3, Rxl	66h	66h	0000h	1200h	6789h	0089h	3	0	0	1	0
SWPB	R3	66h	66h	0000h	1200h	6789h	8900h	3	1	0	1	1
MOV.W	R0, #0x3456	66h	66h	3456h	1200h	6789h	8900h	0	0	0	1	0
MOV.B	R3, Rxl	66h	66h	3456h	1200h	6789h	8956h	3	0	0	1	1

MOV[.W] (Rd), Rs

Instruction code	[0 1 1 1 0 0 1 1][d3d2d1d0 0 s2s1s0] 7300H
Argument	Rd = 4bit(R select), Rs = 3bit(R select)
Word count	1
Cycle count	2 or 3

Function	if (Rd) = even data : [Rd] ← Lobyte (Rs), [Rd+1] ← Hibyte (Rs)
	if (Rd) = odd data : [Rd] ← Hibyte (Rs), [Rd-1] ← Lobyte(Rs)
	(PC)←(PC)+2

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

If the data memory (RAM) location, special function register (SFR), or program memory (ROM) location addressed by the contents of Rd is at an even address, the instruction transfers the contents of the lower-order 8 bits of the general-purpose register Rs to [Rd] and the contents of the higher-order 8 bits of Rs to [Rd + 1]. In the case of an odd address, the instruction transfers the contents of the higher-order 8 bits of Rs to [Rd] and the contents of the lower-order 8 bits of Rs to [Rd – 1].
The legitimate value range designated by Rd is from R0 to R15 and that by Rs is from R0 to R7.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50, #0x6666	66h	66h	-	-	-	-	-	0	0	0	0
MOV.W	R3, #0x0050	66h	66h	-	-	-	0050h	3	0	0	0	0
MOV.W	R0, #0x5555	66h	66h	5555h	-	-	0050h	0	0	0	0	0
MOV.W	(R3), R0	55h	55h	5555h	-	-	0050h	0	0	0	0	0
MOV.W	R1, #0x1200	55h	55h	5555h	1200h	-	0050h	1	1	0	0	0
MOV.W	(R3), R1	00h	12h	5555h	1200h	-	0050h	1	1	0	0	0
MOV.W	R2, #0x0000	00h	12h	5555h	1200h	0000h	0050h	2	1	1	0	0
MOV.W	(R3), R2	00h	00h	5555h	1200h	0000h	0050h	2	1	1	0	0
MOV.W	R0, #0x5634	00h	00h	5634h	1200h	0000h	0050h	0	0	0	1	0
MOV.W	(R3), R0	34h	56h	5634h	1200h	0000h	0050h	0	0	0	1	0
MOV.W	R1, #0x8118	34h	56h	5634h	8118h	0000h	0050h	1	0	0	0	1
MOV.W	(R3), R1	34h	56h	5555h	1200h	0000h	0050h	1	0	0	0	1
MOV.W	R2, #0x5555	34h	56h	5634h	1200h	0000h	0050h	2	0	0	0	0
MOV.W	(R3), R2	34h	56h	5634h	1200h	0000h	0050h	2	0	0	0	0

<Note>

This instruction takes 3 cycles to transfer the contents of Rs to program memory (ROM). However, no data can actually be transferred to ROM.

MOV[.W] (--Rd), Rs

Instruction code	[0 1 1 0 1 0 1 1][d3d2d1d0 0 s2s1s0] 6B00H
Argument	Rd = 4bit(R select), Rs = 3bit(R select)
Word count	1
Cycle count	2 or 3

Function	(Rd)←(Rd)-2
	if (Rd) =even data : [Rd]←Lobyte(Rs), [Rd+1]←Hibyte(Rs)
	if (Rd) =odd data : [Rd]←Hibyte(Rs), [Rd-1]←Lobyte(Rs)
	(PC)←(PC)+2

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction firstly subtracts 2 from the contents of the general-purpose register designated by Rd. Subsequently, if the data memory (RAM) location, special function register (SFR), or program memory (ROM) location addressed by the contents of Rd is at an even address, the instruction transfers the contents of the lower-order 8 bits of the general-purpose register Rs to [Rd] and the contents of the higher-order 8 bits of Rs to [Rd + 1]. In the case of an odd address, the instruction transfers the contents of the higher-order 8 bits of Rs to [Rd] and the contents of the lower-order 8 bits of Rs to [Rd – 1].
The legitimate value range designated by Rd is from R0 to R15 and that by Rs is from R0 to R7.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50, #0x6666	66h	66h	-	-	-	-	-	0	0	0	0
MOV.W	R3, #0x0052	66h	66h	-	-	-	0052h	3	0	0	1	0
MOV.W	R0, #0x5555	66h	66h	5555h	-	-	0052h	0	0	0	0	0
MOV.W	(--R3), R0	55h	55h	5555h	-	-	0050h	0	0	0	0	0
INC	R3, #1	55h	55h	5555h	-	-	0052h	3	0	0	1	0
MOV.W	R1, #0x1200	55h	55h	5555h	1200h	-	0052h	1	1	0	0	0
MOV.W	(--R3), R1	00h	12h	5555h	1200h	-	0050h	1	1	0	0	0
INC	R3, #1	00h	12h	5555h	1200h	-	0052h	3	0	0	1	0
MOV.W	R2, #0x0000	00h	12h	5555h	1200h	0000h	0052h	2	1	1	0	0
MOV.W	(--R3), R2	00h	00h	5555h	1200h	0000h	0050h	2	1	1	0	0
INC	R3, #1	00h	00h	5555h	1200h	0000h	0052h	3	0	0	1	0
MOV.W	R0, #0x5634	00h	00h	5634h	1200h	0000h	0052h	0	0	0	1	0
MOV.W	(--R3), R0	34h	56h	5634h	1200h	0000h	0050h	0	0	0	1	0
INC	R3, #1	34h	56h	5634h	1200h	0000h	0052h	3	0	0	1	0
MOV.W	R1, #0x8118	34h	56h	5634h	8118h	0000h	0052h	1	0	0	0	1
MOV.W	(--R3), R1	18h	81h	5634h	8118h	0000h	0050h	1	0	0	0	1

<Note>

This instruction takes 3 cycles to transfer the contents of Rs to program memory (ROM). However, no data can actually be transferred to ROM.

MOV[.W] (Rd, ±n), Rs

Instruction code	[0 1 1 1 0 0 1 1][d3d2d1d0 1 s2s1s0][0 0 0 0 n11 to n8][n7 to n0] 7308H
Argument	Rd = 4bit(R select), n = 12bit(signed), Rs = 3bit(R select)
Word count	2
Cycle count	3 or 4

Function	if (Rd±n) = even data :
	[(Rd±n)&FFFFh]←Lobyte(Rs), [(Rd±n+1)&FFFFh]←Hibyte(Rs)
	if (Rd±n) = odd data :
	[(Rd±n)&FFFFh]←Hibyte(Rs), [(Rd±n-1)&FFFFh]←Lobyte(Rs)
	(PC)←(PC)+4

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

If the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the address derived by the arithmetic operation*1 performed on the contents of the general-purpose register designated by Rd and n is at an even address, the instruction transfers the contents of the lower-order 8 bits of the general-purpose register Rs to [(Rd±n)&FFFFh] and the higher-order 8 bits of Rs to [(Rd±n+1)&FFFFh]. In the case of an odd address, the instruction transfers the contents of the higher-order 8 bits of Rs to [(Rd±n)&FFFFh] and the lower-order 8 bits of Rs to [(Rd±n-1)&FFFFh].
The legitimate value range designated by Rd is from R0 to R15, that by Rs is from R0 to R7, and that by n is that of signed 12-bit data (-2048 to 2047).

*1: Any carry or borrow resulting from a 16-bit arithmetic operation is ignored.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50, #0x6666	66h	66h	-	-	-	-	-	0	0	0	0
MOV.W	R3, #0x0000	66h	66h	-	-	-	0000h	3	1	1	0	0
MOV.W	R0, #0x5555	66h	66h	5555h	-	-	0000h	0	0	0	0	0
MOV.W	(R3, 0x50), R0	55h	55h	5555h	-	-	0000h	0	0	0	0	0
MOV.W	R1, #0x1200	55h	55h	5555h	1200h	-	0000h	1	1	0	0	0
MOV.W	(R3, 0x50), R1	00h	12h	5555h	1200h	-	0000h	1	1	0	0	0
MOV.W	R2, #0x0000	00h	12h	5555h	1200h	0000h	0000h	2	1	1	0	0
MOV.W	(R3, 0x50), R2	00h	00h	5555h	1200h	0000h	0000h	2	1	1	0	0
MOV.W	R0, #0x5634	00h	00h	5634h	1200h	0000h	0000h	0	0	0	1	0
MOV.W	(R3, 0x50), R0	34h	56h	5634h	1200h	0000h	0000h	0	0	0	1	0
MOV.W	R1, #0x8118	34h	56h	5634h	8118h	0000h	0000h	1	0	0	0	1
MOV.W	(R3, 0x50), R1	18h	81h	5634h	8118h	0000h	0000h	1	0	0	0	1
MOV.W	R2, #0x5555	18h	81h	5634h	8118h	5555h	0000h	2	0	0	0	0
MOV.W	(R3, 0x50), R2	55h	55h	5634h	8118h	5555h	0000h	2	0	0	0	0

<Note>

This instruction takes 4 cycles to transfer the contents of Rs to program memory (ROM). However, no data can actually be transferred to ROM.

MOV[.W] (--Rd, ±n), Rs

Instruction code	[0 1 1 0 1 0 1 1][d3d2d1d0 1 s2s1s0][0 0 0 0 n11 to n8][n7 to n0] 6B08H
Argument	Rd = 4bit(R select), n = 12bit(signed), Rs = 3bit(R select)
Word count	2
Cycle count	3 or 4

Function	(Rd)←(Rd)-2
	if (Rd±n) = even data :
	[(Rd±n)&FFFFh]←Lobyte(Rs), [(Rd±n+1)&FFFFh]←Hibyte(Rs)
	if (Rd±n) = odd data :
	[(Rd±n)&FFFFh]←Hibyte(Rs), [(Rd±n-1)&FFFFh]←Lobyte(Rs)
	(PC)←(PC)+4

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction firstly subtracts 2 from the contents of the general-purpose register designated by Rd. Subsequently, if the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the address derived by the arithmetic operation*1 performed on the contents of Rd and n is at an even address, the instruction transfers the contents of the lower-order 8 bits of the general-purpose register Rs to [(Rd±n)&FFFFh] and the higher-order 8 bits of Rs to [(Rd±n+1)&FFFFh]. In the case of an odd address, the instruction transfers the contents of the higher-order 8 bits of Rs to [(Rd±n)&FFFFh] and the lower-order 8 bits of Rs to [(Rd±n-1)&FFFFh].

The legitimate value range designated by Rd is from R0 to R15, that by Rs is from R0 to R7, and that by n is that of signed 12-bit data (-2048 to 2047).

*1: Any carry or borrow resulting from a 16-bit arithmetic operation is ignored.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50, #0x6666	66h	66h	-	-	-	-	-	0	0	0	0
MOV.W	R3, #0x0002	66h	66h	-	-	-	0002h	3	0	0	1	0
MOV.W	R0, #0x5555	66h	66h	5555h	-	-	0002h	0	0	0	0	0
MOV.W	(--R3, 0x50), R0	55h	55h	5555h	-	-	0000h	0	0	0	0	0
INC	R3, #1	55h	55h	5555h	-	-	0002h	3	0	0	1	0
MOV.W	R1, #0x1200	55h	55h	5555h	1200h	-	0002h	1	1	0	0	0
MOV.W	(--R3, 0x50), R1	00h	12h	5555h	1200h	-	0000h	1	1	0	0	0
INC	R3, #1	00h	12h	5555h	1200h	-	0002h	3	0	0	1	0
MOV.W	R2, #0x0000	00h	12h	5555h	1200h	0000h	0002h	2	1	1	0	0
MOV.W	(--R3, 0x50), R2	00h	00h	5555h	1200h	0000h	0000h	2	1	1	0	0
INC	R3, #1	00h	00h	5555h	1200h	0000h	0002h	3	0	0	1	0
MOV.W	R0, #0x5634	00h	00h	5634h	1200h	0000h	0002h	0	0	0	1	0
MOV.W	(--R3, 0x50), R0	34h	56h	5634h	1200h	0000h	0000h	0	0	0	1	0
INC	R3, #1	34h	56h	5634h	1200h	0000h	0002h	3	0	0	1	0
MOV.W	R1, #0x8118	34h	56h	5634h	8118h	0000h	0002h	1	0	0	0	1
MOV.W	(--R3, 0x50), R1	18h	81h	5634h	8118h	0000h	0000h	1	0	0	0	1

<Note>

This instruction takes 4 cycles to transfer the contents of Rs to program memory (ROM). However, no data can actually be transferred to ROM.

MOV[.W] (Rd++), Rs

Instruction code	[0 1 1 0 0 0 1 1][d3d2d1d0 0 s2s1s0] 6300H
Argument	Rd = 4bit(R select), Rs = 3bit(R select)
Word count	1
Cycle count	2 or 3

Function	if (Rd) = even data : [Rd]←Lobyte(Rs), [Rd+1]←Hibyte (Rs)
	if (Rd) = odd data : [Rd]←Hibyte(Rs), [Rd-1]←Lobyte(Rs)
	(Rd)← (Rd)+2, (PC)←(PC)+2

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

If the data memory (RAM) location, special function register (SFR), or program memory (ROM) location addressed by the contents of the general-purpose register designated by Rd is at an even address, the instruction transfers the contents of the lower-order 8 bits of the general-purpose register Rs to [Rd] and the contents of the higher-order 8 bits of Rs to [Rd + 1]. In the case of an odd address, the instruction transfers the contents of the higher-order 8 bits of Rs to [Rd] and the contents of the lower-order 8 bits of Rs to [Rd – 1]. Subsequently, the instruction increments the contents of Rd by 2.
The legitimate value range designated by Rd is from R0 to R15 and that by Rs is from R0 to R7.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50, #0x6666	66h	66h	-	-	-	-	-	0	0	0	0
MOV.W	R3, #0x0050	66h	66h	-	-	-	0050h	3	0	0	0	0
MOV.W	R0, #0x5555	66h	66h	5555h	-	-	0050h	0	0	0	0	0
MOV.W	(R3++), R0	55h	55h	5555h	-	-	0052h	0	0	0	0	0
DEC	R3, #1	55h	55h	5555h	-	-	0050h	3	0	0	0	0
MOV.W	R1, #0x1200	55h	55h	5555h	1200h	-	0050h	1	1	0	0	0
MOV.W	(R3++), R1	00h	12h	5555h	1200h	-	0052h	1	1	0	0	0
DEC	R3, #1	00h	12h	5555h	1200h	-	0050h	3	0	0	0	0
MOV.W	R2, #0x0000	00h	12h	5555h	1200h	0000h	0050h	2	1	1	0	0
MOV.W	(R3++), R2	00h	00h	5555h	1200h	0000h	0052h	2	1	1	0	0
DEC	R3, #1	00h	00h	5555h	1200h	0000h	0050h	3	0	0	0	0
MOV.W	R0, #0x5634	00h	00h	5634h	1200h	0000h	0050h	0	0	0	1	0
MOV.W	(R3++), R0	34h	56h	5634h	1200h	0000h	0052h	0	0	0	1	0
DEC	R3, #1	34h	56h	5634h	1200h	0000h	0050h	3	0	0	0	0
MOV.W	R1, #0x8118	34h	56h	5634h	8118h	0000h	0050h	1	0	0	0	1
MOV.W	(R3++), R1	18h	81h	5634h	8118h	0000h	0052h	1	0	0	0	1
DEC	R3, #1	18h	81h	5634h	8118h	0000h	0050h	3	0	0	0	0
MOV.W	R2, #0x5555	18h	81h	5634h	8118h	5555h	0050h	2	0	0	0	0
MOV.W	(R3++), R2	55h	55h	5634h	8118h	5555h	0052h	2	0	0	0	0
DEC	R3, #1	55h	55h	5634h	8118h	5555h	0050h	3	0	0	0	0

<Note>

This instruction takes 3 cycles to transfer the contents of Rs to program memory (ROM). However, no data can actually be transferred to ROM.

MOV[.W] (Rd++, ±n), Rs

Instruction code	[0 1 1 0 0 0 1 1][d3d2d1d0 1 s2s1s0][0 0 0 0 n11 to n8][n7 to n0] 6308H
Argument	Rd = 4bit(R select), n = 12bit(signed), Rs = 3bit(R select)
Word count	2
Cycle count	3 or 4

Function	if (Rd±n) =even data :
	[(Rd±n)&FFFFh]←Lobyte(Rs), [(Rd±n+1)&FFFFh]←Hibyte(Rs)
	if (Rd±n) =odd data :
	[(Rd±n)&FFFFh]←Hibyte(Rs), [(Rd±n-1)&FFFFh]←Lobyte(Rs)
	(Rd) ← (Rd)+2, (PC)←(PC)+4

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

If the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the address derived by the arithmetic operation*1 performed on the contents of the general-purpose register designated by Rd and n is at an even address, the instruction transfers the contents of the lower-order 8 bits of the general-purpose register Rs to [(Rd±n)&FFFFh] and the higher-order 8 bits of Rs to [(Rd±n+1)&FFFFh]. In the case of an odd address, the instruction transfers the contents of the higher-order 8 bits of Rs to [(Rd±n)&FFFFh] and the lower-order 8 bits of Rs to [(Rd±n-1)&FFFFh]. Subsequently, the instruction increments the contents of Rd by 2.

The legitimate value range designated by Rd is from R0 to R15, that by Rs is from R0 to R7, and that by n is that of signed 12-bit data (-2048 to 2047).

*1: Any carry or borrow resulting from a 16-bit arithmetic operation is ignored.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50, #0x6666	66h	66h	-	-	-	-	-	0	0	0	0
MOV.W	R3, #0x0000	66h	66h	-	-	-	0000h	3	1	1	0	0
MOV.W	R0, #0x5555	66h	66h	5555h	-	-	0000h	0	0	0	0	0
MOV.W	(R3++, 0x50), R0	55h	55h	5555h	-	-	0002h	0	0	0	0	0
DEC	R3, #1	55h	55h	5555h	-	-	0000h	3	1	1	0	0
MOV.W	R1, #0x1200	55h	55h	5555h	1200h	-	0000h	1	1	0	0	0
MOV.W	(R3++, 0x50), R1	00h	12h	5555h	1200h	-	0002h	1	1	0	0	0
DEC	R3, #1	00h	12h	5555h	1200h	-	0000h	3	1	1	0	0
MOV.W	R2, #0x0000	00h	12h	5555h	1200h	0000h	0000h	2	1	1	0	0
MOV.W	(R3++, 0x50), R2	00h	00h	5555h	1200h	0000h	0002h	2	1	1	0	0
DEC	R3, #1	00h	00h	5555h	1200h	0000h	0000h	3	1	1	0	0
MOV.W	R0, #0x5634	00h	00h	5634h	1200h	0000h	0000h	0	0	0	1	0
MOV.W	(R3++, 0x50), R0	34h	56h	5634h	1200h	0000h	0002h	0	0	0	1	0
DEC	R3, #1	34h	56h	5634h	1200h	0000h	0000h	3	1	1	0	0
MOV.W	R1, #0x	34h	56h	5634h	8118h	0000h	0000h	1	0	0	0	1
MOV.W	(R3++, 0x50), R1	18h	81h	5634h	8118h	0000h	0002h	1	0	0	0	1

<Note>

This instruction takes 4 cycles to transfer the contents of Rs to program memory (ROM). However, no data can actually be transferred to ROM.

MOV[.W] m16, #imm16

Instruction code	[0 1 1 1 1 0 X 1][m7m6m5m4m3m2m1m0][i15 to i8][i7 to i0] 7900H(RAM), 7B00H(SFR)
Argument	m16 = 16bit (lower 8bit valid for operation code) imm16 = 16bit (immediate data)
Word count	2
Cycle count	2
Function	if "m16" is even: (m16+1)←Hibyte(imm16), (m16)←Lobyte(imm16) if "m16" is odd: (m16)←Hibyte(imm16), (m16-1)←Lobyte(imm16) (PC)←(PC)+4
Affected flags	Z8, Z16, P, S

[Description]

This instruction transfers 16-bit immediate data imm16 to 2-byte data memory (RAM) location or SFR (one of the registers dedicated to control the internal peripheral devices) addressed by m16.

The 2-byte destination address is determined according to the following rules:
・If m16 is an even number, the higher-order 8 bits of imm16 is transferred to the odd address (m16+1) and the lower-order 8 bits to the even address (m16).
・If m16 is an odd number, the higher-order 8 bits of imm16 is transferred to the odd address (m16) and the lower-order 8 bits to the even address (m16-1).

The compiler generates the instruction code while regarding RAM or SFR as the destination of transfer according to the value of m16 (first operand data).
・ When specifying a RAM location, specify m16 with a value from 00H to FFH (0000H to 00FFH). It is disallowed to specify a RAM address not lower than 100H.
・ When specifying a SFR, specify m16 with a value from 7F00H to 7FFFH.

The basic types of generated instruction code are 7900H (RAM) and 7B00H (SFR), respectively. The lower-order 8 bits of m16 are reflected in the behavior of the instruction code.

[Example]

		RAM (50h)	RAM (51h)	Z8	Z16	P	S
		-	-	-	-	-	-
MOV.W	0x50, #0x5555	55h	55h	0	0	0	0
MOV.W	0x50, #0x1200	00h	12h	1	0	0	0
MOV.W	0x50, #0x0000	00h	00h	1	1	0	0
MOV.W	0x50, #0x3456	56h	34h	0	0	1	0
MOV.W	0x50, #0x8118	18h	81h	0	0	0	1
MOV.W	0x50, #0x5555	55h	55h	0	0	0	0

MOV[.W] m16, Rs

Instruction code	[1 0 X 1 s2s1s0 1][m7m6m5m4m3m2m1m0] 9100H(RAM), B100H(SFR)
Argument	m16 = 16bit (lower 8bit valid for operation code) Rs = 3bit (R select)
Word count	1
Cycle count	1
Function	if m16 is even: (m16+1)←Hibyte(Rs), (m16)←Lobyte(Rs) if m16 is odd: (m16)←Hibyte(Rs), (m16-1)←Lobyte(Rs) (PC)←(PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers the contents (16 bits) of the general-purpose register designated by Rs to 2-byte data memory (RAM) location or SFR (one of the registers dedicated to control the internal peripheral devices) addressed by m16. The legitimate value range designated by Rs is from R0 to R7.

The 2-byte destination address is determined according to the following rules:
・If m16 is an even number, the higher-order 8 bits of Rs are transferred to the odd address (m16+1) and the lower-order 8 bits to the even address (m16).
If m16 is an odd number, the higher-order 8 bits of Rs are transferred to the odd address (m16) and the lower-order 8 bits to the even address (m16-1).

The compiler generates the instruction code while regarding RAM or SFR as the destination of transfer according to the value of m16 (first operand data).
・ When specifying a RAM location, specify m16 with a value from 00H to FFH (0000H to 00FFH). It is disallowed to specify a RAM address not lower than 100H.
・ When specifying a SFR, specify m16 with a value from 7F00H to 7FFFH.

The basic types of generated instruction code are 9100H (RAM) and B100H (SFR), respectively. The lower-order 8 bits of m16 are reflected in the behavior of the instruction code.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	R0, #0x5555	-	-	5555h	-	-	-	0	0	0	0	0
MOV.W	R3, #0x3333	-	-	5555h	-	-	3333h	3	0	0	0	0
MOV.W	0x50, R0	55h	55h	5555h	-	-	3333h	0	0	0	0	0
MOV.W	R1, #0x1200	55h	55h	5555h	1200h	-	3333h	1	1	0	0	0
MOV.W	R3, #0x7777	55h	55h	5555h	1200h	-	7777h	3	0	0	0	1
MOV.W	0x50, R1	00h	12h	5555h	1200h	-	7777h	1	1	0	0	0
MOV.W	R2, #0x0000	00h	12h	5555h	1200h	0000h	7777h	2	1	1	0	0
MOV.W	R3, #0x3333	00h	12h	5555h	1200h	0000h	3333h	3	0	0	0	0
MOV.W	0x50, R2	00h	00h	5555h	1200h	0000h	3333h	2	1	1	0	0

MOV[.W] Rd, #imm8

Instruction code	[0 0 1 0 d2d1d0 1][i7i6i5i4i3i2i1i0] 2100H
Argument	Rd = 3bit(R select), imm8 = 8bit(immediate data)
Word count	1
Cycle count	1
Function	(Rd) ←16bit data(Hibyte=00H, Lobyte=#imm8), (PC)←(PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers immediate data designated by imm8 to the general-purpose register designated by Rd.

The legitimate value range designated by Rd is from R0 to R7 and that by imm8 is from 0 to FFh.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-
MOV.W	R0, #0x55	0055h	-	-	-	0	0	0	0	0
MOV.W	R1, #0x00	0055h	0000h	-	-	1	1	1	0	0
MOV.W	R2, #0x34	0055h	0000h	0034h	-	2	0	0	1	0
MOV.W	R3, #0x8118	0055h	0000h	0034h	8118h	3	0	0	0	1
MOV.W	R0, #0xFF	00FFh	0000h	0034h	8118h	0	0	0	0	0
MOV.W	R1, #0x33	00FFh	0033h	0034h	8118h	1	0	0	0	0

<Note>

The higher-order 8 bits of Rd are loaded with 00H.

MOV[.W] Rd, #imm16

Instruction code	[0 0 1 1 0 0 0 1][0 0 1 1 d3d2d1d0][i15 to i8][i7 to i0] 3130H
Argument	Rd = 4bit(R select), imm16 = 16bit(immediate data)
Word count	2
Cycle count	2
Function	(Rd)←#imm16, (PC)←(PC)+4
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers immediate data designated by imm16 to the general-purpose register designated by Rd.

The legitimate value range designated by Rd is from R0 to R15 and that by imm16 is from 0 to FFFF.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-
MOV.W	R0, #0x5555	5555h	-	-	-	0	0	0	0	0
MOV.W	R1, #0x1200	5555h	1200h	-	-	1	1	0	0	0
MOV.W	R2, #0x0000	5555h	1200h	0000h	-	2	1	1	0	0
MOV.W	R3, #0x5634	5555h	1200h	0000h	5634h	3	0	0	1	0
MOV.W	R0, #0x8118	8118h	1200h	0000h	5634h	0	0	0	0	1
MOV.W	R1, #0x00FF	8118h	00FFh	0000h	5634h	1	0	0	0	0
MOV.W	R2, #0x5555	8118h	00FFh	5555h	5634h	2	0	0	0	0

MOV[.W] Rd, (Rs)

Instruction code	[0 1 1 1 0 0 0 1][s3s2s1s0 0 d2d1d0] 7100H
Argument	Rd = 3bit(R select), Rs = 4bit(R select)
Word count	1
Cycle count	2 or 3

Function	if (Rs) =even data : Hibyte(Rd)←[Rs+1], Lobyte(Rd)← [Rs]
	if (Rs) =odd data : Hibyte(Rd)←[Rs], Lobyte [Rs-1]
	(PC)←(PC)+2

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

If the data memory (RAM) location, special function register (SFR), or program memory (ROM) location addressed by the contents of the general-purpose register designated by Rs is at an even address, the instruction transfers the contents to the lower-order 8-bit positions of the general-purpose register Rd and the contents of [Rs + 1] to the higher-order 8-bit positions of Rd. In the case of an odd address, the instruction transfers contents of [Rs] to the higher-order 8-bit positions of Rd and the contents of [Rs - 1] to the lower-order 8-bit positions of Rd.

The legitimate value range designated by Rd is from R0 to R7 and that by Rs is from R0 to R15.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0050	-	-	-	-	-	0050h	3	0	0	0	0
MOV.W	0x50, #0x5555	55h	55h	-	-	-	0050h	3	0	0	0	0
MOV.W	R0, (R3)	55h	55h	5555h	-	-	0050h	0	0	0	0	0
MOV.W	0x50, #0x1200	00h	12h	5555h	-	-	0050h	0	1	0	0	0
MOV.W	R1, (R3)	00h	12h	5555h	1200h	-	0050h	1	1	0	0	0
MOV.W	0x50, #0x0000	00h	00h	5555h	1200h	-	0050h	1	1	1	0	0
MOV.W	R2, (R3)	00h	00h	5555h	1200h	0000h	0050h	2	1	1	0	0
MOV.W	0x50, #0x5634	34h	56h	5555h	1200h	0000h	0050h	2	0	0	1	0
MOV.W	R0, (R3)	34h	56h	5634h	1200h	0000h	0050h	0	0	0	1	0
MOV.W	0x50, #0x8118	18h	81h	5634h	1200h	0000h	0050h	0	0	0	0	1
MOV.W	R1, (R3)	18h	81h	5634h	8118h	0000h	0050h	1	0	0	0	1
MOV.W	0x50, #0x5555	55h	55h	5634h	8118h	0000h	0050h	1	0	0	0	0
MOV.W	R2, (R3)	55h	55h	5634h	8118h	5555h	0050h	2	0	0	0	0

<Note>

The instruction takes 3 cycles to transfer the contents of program memory (ROM) to Rd.

MOV[.W] Rd, (--Rs)

Instruction code	[0 1 1 0 1 0 0 1][s3s2s1s0 0 d2d1d0] 6900H
Argument	Rd = 3bit(R select), Rs = 4bit(R select)
Word count	1
Cycle count	2 or 3

Function	(Rs)←(Rs)-2
	if (Rs) = even data : Hibyte(Rd)←[Rs+1], Lobyte(Rd)← [Rs]
	if (Rs) = odd data : Hibyte(Rd)←[Rs], Lobyte(Rd)← [Rs-1]
	(PC)←(PC)+2

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction firstly subtracts 2 from the contents of the general-purpose register designated by Rs. Subsequently, if the data memory (RAM) location, special function register (SFR), or program memory (ROM) location addressed by the contents of the general-purpose register Rs is at an even address, the instruction transfers the contents to the lower-order 8-bit positions of the general-purpose register Rd and the contents of [Rs + 1] to the higher-order 8-bit positions of Rd. In the case of an odd address, the instruction transfers contents of [Rs] to the higher-order 8-bit positions of Rd and the contents of [Rs - 1] to the lower-order 8-bit positions of Rd.

The legitimate value range designated by Rd is from R0 to R7 and that by Rs is from R0 to R15.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0052	-	-	-	-	-	0052h	3	0	0	1	0
MOV.W	0x50, #0x5555	55h	55h	-	-	-	0052h	3	0	0	0	0
MOV.W	R0, (--R3)	55h	55h	5555h	-	-	0050h	0	0	0	0	0
INC	R3, #1	55h	55h	5555h	-	-	0052h	3	0	0	1	0
MOV.W	0x50, #0x1200	00h	12h	5555h	-	-	0052h	3	1	0	0	0
MOV.W	R1, (--R3)	00h	12h	5555h	1200h	-	0050h	1	1	0	0	0
INC	R3, #1	00h	12h	5555h	1200h	-	0052h	3	0	0	1	0
MOV.W	0x50, #0x0000	00h	00h	5555h	1200h	-	0052h	3	1	1	0	0
MOV.W	R2, (--R3)	00h	00h	5555h	1200h	0000h	0050h	2	1	1	0	0
INC	R3, #1	00h	00h	5555h	1200h	0000h	0052h	3	0	0	1	0
MOV.W	0x50, #0x5634	34h	56h	5555h	1200h	0000h	0052h	3	0	0	1	0
MOV.W	R0, (--R3)	34h	56h	5634h	1200h	0000h	0050h	0	0	0	1	0
INC	R3, #1	34h	56h	5634h	1200h	0000h	0052h	3	0	0	1	0
MOV.W	0x50, #0x8118	18h	81h	5634h	1200h	0000h	0052h	3	0	0	0	1
MOV.W	R1, (--R3)	18h	81h	5634h	8118h	0000h	0050h	1	0	0	0	1
INC	R3, #1	18h	81h	5634h	8118h	0000h	0052h	3	0	0	1	0
MOV.W	0x50, #0x5555	55h	55h	5634h	8118h	0000h	0052h	3	0	0	0	0
MOV.W	R2, (--R3)	55h	55h	5634h	8118h	5555h	0050h	2	0	0	0	0
INC	R3, #1	55h	55h	5634h	8118h	5555h	0052h	3	0	0	1	0

<Note>

The instruction takes 3 cycles to transfer the contents of program memory (ROM) to Rd.

MOV[.W] Rd, Rs, ±n)

Instruction code	[0 1 1 1 0 0 0 1][s3s2s1s0 1 d2d1d0][0 0 0 0 n11 to n8][n7 to n0] 7108H
Argument	Rd = 3bit(R select), Rs = 4bit(R select), n = 12bit(signed)
Word count	2
Cycle count	3 or 4

Function	if (Rs±n) =even data :
	Hibyte(Rd)←[(Rs±n+1)&FFFFh], Lobyte(Rd)← [(Rs±n)&FFFFh]
	if (Rs±n) =odd data :
	Hibyte(Rd)←[(Rs±n)&FFFFh], Lobyte(Rd)←[(Rs±n-1)&FFFFh]
	(PC)←(PC)+4

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

If the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the address derived by the arithmetic operation *1 performed on the contents of the general-purpose register designated by Rs and n is at an even address, the instruction transfers the contents to the lower-order 8-bit positions of the general-purpose register Rd and the contents of [(Rs±n+1)&FFFFh] to the higher-order 8-bit positions of Rd. In the case of an odd address, the instruction transfers contents of [(Rs ±n)&FFFFh] to the higher-order 8-bit positions of Rd and the contents of [(Rs±n-1)&FFFFh] to the lower-order 8-bit positions of Rd.

The legitimate value range designated by Rd is from R0 to R7, that by Rs is from R0 to R15, and that by n is that of signed 12-bit data (-2048 to 2047).

*1: Any carry or borrow resulting from a 16-bit arithmetic operation is ignored.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0000	-	-	-	-	-	0000h	3	1	1	0	0
MOV.W	0x50, #0x5555	55h	55h	-	-	-	0000h	3	0	0	0	0
MOV.W	R0, (R3, 0x50)	55h	55h	5555h	-	-	0000h	0	0	0	0	0
MOV.W	0x50, #0x1200	00h	12h	5555h	-	-	0000h	0	1	0	0	0
MOV.W	R1, (R3, 0x50)	00h	12h	5555h	1200h	-	0000h	1	1	0	0	0
MOV.W	0x50, #0x0000	00h	00h	5555h	1200h	-	0000h	1	1	1	0	0
MOV.W	R2, (R3, 0x50)	00h	00h	5555h	1200h	0000h	0000h	2	1	1	0	0
MOV.W	0x50, #0x5634	34h	56h	5555h	1200h	0000h	0000h	2	0	0	1	0
MOV.W	R0, (R3, 0x50)	34h	56h	5634h	1200h	0000h	0000h	0	0	0	1	0
MOV.W	0x50, #0x8118	18h	81h	5634h	1200h	0000h	0000h	0	0	0	0	1
MOV.W	R1, (R3, 0x50)	18h	81h	5634h	8118h	0000h	0000h	1	0	0	0	1
MOV.W	0x50, #0x5555	55h	55h	5634h	8118h	0000h	0000h	1	0	0	0	0
MOV.W	R2, (R3, 0x50)	55h	55h	5634h	8118h	5555h	0000h	2	0	0	0	0

<Note>

This instruction takes 4 cycles to transfer the contents of program memory (ROM) to Rd.

MOV[.W] Rd, (--Rs, ±n)

Instruction code	[0 1 1 0 1 0 0 1][s3s2s1s0 1 d2d1d0][0 0 0 0 n11 to n8][n7 to n0] 6908H
Argument	Rd = 3bit(R select), Rs = 4bit(R select), n = 12bit(signed)
Word count	2
Cycle count	3 or 4

Function	(Rs)←(Rs)-2
	if (Rs±n) =even data :
	Hibyte(Rd)←[(Rs±n+1)&FFFFh], Lobyte(Rd)← [(Rs±n)&FFFFh]
	if (Rs±n) =odd data :
	Hibyte(Rd)←[(Rs±n)&FFFFh], Lobyte(Rd) ← [(Rs±n-1)&FFFFh]
	((PC)←(PC)+4

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction firstly subtracts 2 from the contents of the general-purpose register designated by Rs. Subsequently, if the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the address derived by the arithmetic operation*1 performed on the contents of the general-purpose register Rs and n is at an even address, the instruction transfers the contents to the lower-order 8-bit positions of the general-purpose register Rd and the contents of [(Rs±n+1)&FFFFh] to the higher-order 8-bit positions of Rd. In the case of an odd address, the instruction transfers contents of [(Rs±n)&FFFFh] to the higher-order 8-bit positions of Rd and the contents of [(Rs±n-1)&FFFFh] to the lower-order 8-bit positions of Rd.

The legitimate value range designated by Rd is from R0 to R7, that by Rs is from R0 to R15, and that by n is that of signed 12-bit data (-2048 to 2047).

*1: Any carry or borrow resulting from a 16-bit arithmetic operation is ignored.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0002	-	-	-	-	-	0002h	3	0	0	1	0
MOV.W	0x50, #0x5555	55h	55h	-	-	-	0002h	3	0	0	0	0
MOV.W	R0, (--R3, 0x50)	55h	55h	5555h	-	-	0000h	0	0	0	0	0
INC	R3, #1	55h	55h	5555h	-	-	0002h	3	0	0	1	0
MOV.W	0x50, #0x1200	00h	12h	5555h	-	-	0002h	3	1	0	0	0
MOV.W	R1, (--R3, 0x50)	00h	12h	5555h	1200h	-	0000h	1	1	0	0	0
INC	R3, #1	00h	12h	5555h	1200h	-	0002h	3	0	0	1	0
MOV.W	0x50, #0x0000	00h	00h	5555h	1200h	-	0002h	3	1	1	0	0
MOV.W	R2, (--R3, 0x50)	00h	00h	5555h	1200h	0000h	0000h	2	1	1	0	0
INC	R3, #1	00h	00h	5555h	1200h	0000h	0002h	3	0	0	1	0
MOV.W	0x50, #0x5634	34h	56h	5555h	1200h	0000h	0002h	3	0	0	1	0
MOV.W	R0, (--R3, 0x50)	34h	56h	5634h	1200h	0000h	0000h	0	0	0	1	0
INC	R3, #1	34h	56h	5634h	1200h	0000h	0002h	3	0	0	1	0
MOV.W	0x50, #0x8118	18h	81h	5634h	1200h	0000h	0002h	3	0	0	0	1
MOV.W	R1, (--R3, 0x50)	18h	81h	5634h	8118h	0000h	0000h	1	0	0	0	1

<Note>

This instruction takes 4 cycles to transfer the contents of program memory (ROM) to Rd.

MOV[.W] Rd, (Rs__++)

Instruction code	[0 1 1 0 0 0 0 1][s3s2s1s0 0 d2d1d0] 6100H
Argument	Rd = 3bit(R select), Rs = 4bit(R select)
Word count	1
Cycle count	2 or 3

Function	If (Rs) = even data : Hibyte(Rd)←[Rs+1], Lobyte(Rd)← [Rs]
	If (Rs) = odd data : Hibyte(Rd)←[Rs], Lobyte(Rd)←[Rs-1]
	(Rs)←(Rs)+2, (PC)←(PC)+2

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

If the data memory (RAM) location, special function register (SFR), or program memory (ROM) location addressed by the contents of the general-purpose register designated by Rs is an even address, the instruction transfers the contents to the lower-order 8-bit positions of the general-purpose register Rd and the contents of [Rs + 1] to the higher-order 8-bit positions of Rd. In the case of an odd address, the instruction transfers contents of [Rs] to the higher-order 8-bit positions of Rd and the contents of [Rs - 1] to the lower-order 8-bit positions of Rd. Subsequently, the instruction increments the contents of Rs by 2.
The legitimate value range designated by Rd is from R0 to R7 and that by Rs is from R0 to R15.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0050	-	-	-	-	-	0050h	3	0	0	0	0
MOV.W	0x50, #0x5555	55h	55h	-	-	-	0050h	3	0	0	0	0
MOV.W	R0, (R3++)	55h	55h	5555h	-	-	0052h	0	0	0	0	0
DEC	R3, #1	55h	55h	5555h	-	-	0050h	3	0	0	0	0
MOV.W	0x50, #0x1200	00h	12h	5555h	-	-	0050h	3	1	0	0	0
MOV.W	R1, (R3++)	00h	12h	5555h	1200h	-	0052h	1	1	0	0	0
DEC	R3, #1	00h	12h	5555h	1200h	-	0050h	3	0	0	0	0
MOV.W	0x50, #0x0000	00h	00h	5555h	1200h	-	0050h	3	1	1	0	0
MOV.W	R2, (R3++)	00h	00h	5555h	1200h	0000h	0052h	2	1	1	0	0
DEC	R3, #1	00h	00h	5555h	1200h	0000h	0050h	3	0	0	0	0
MOV.W	0x50, #0x5634	34h	56h	5555h	1200h	0000h	0050h	3	0	0	1	0
MOV.W	R0, (R3++)	34h	56h	5634h	1200h	0000h	0052h	0	0	0	1	0
DEC	R3, #1	34h	56h	5634h	1200h	0000h	0050h	3	0	0	0	0
MOV.W	0x50, #0x8118	18h	81h	5634h	1200h	0000h	0050h	3	0	0	0	1
MOV.W	R1, (R3++)	18h	81h	5634h	8118h	0000h	0052h	1	0	0	0	1
DEC	R3, #1	18h	81h	5634h	8118h	0000h	0050h	3	0	0	0	0
MOV.W	0x50, #0x5555	55h	55h	5634h	8118h	0000h	0050h	3	0	0	0	0
MOV.W	R2, (R3++)	55h	55h	5634h	8118h	5555h	0052h	2	0	0	0	0
DEC	R3, #1	55h	55h	5634h	8118h	5555h	0050h	3	0	0	0	0

<Note>

The instruction takes 3 cycles to transfer the contents of program memory (ROM) to Rd.

MOV[.W] Rd, (Rs++, ±n)

Instruction code	[0 1 1 0 0 0 0 1][s3s2s1s0 1 d2d1d0][0 0 0 0 n11 to n8][n7 to n0] 6108H
Argument	Rd = 3bit(R select), Rs = 4bit(R select), n = 12bit(signed)
Word count	2
Cycle count	3 or 4

Function	if (Rs±n) =even data :
	Hibyte(Rd)←[(Rs±n+1)&FFFFh], Lobyte(Rd)←[(Rs±n)&FFFFh]
	if (Rs±n) =odd data :
	Hibyte(Rd)←[(Rs±n)&FFFFh], Lobyte(Rd)←[(Rs±n-1)&FFFFh]
	(Rs)←(Rs)+2, (PC)←(PC)+4

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

If the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the address derived by the arithmetic operation*1 performed on the contents of the general-purpose register designated by Rs and n is at an even address, the instruction transfers the contents to the lower-order 8-bit positions of the general-purpose register Rd and the contents of [(Rs±n+1)&FFFFh] to the higher-order 8-bit positions of Rd. In the case of an odd address, the instruction transfers contents of [(Rs ±n)&FFFFh] to the higher-order 8-bit positions of Rd and the contents of [(Rs±n-1)&FFFFh] to the lower-order 8-bit positions of Rd. Subsequently, the instruction increments the contents of Rs by 2.

The legitimate value range designated by Rd is from R0 to R7, that by Rs is from R0 to R15, and that by n is that of signed 12-bit data (-2048 to 2047).

*1: Any carry or borrow resulting from a 16-bit arithmetic operation is ignored.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0000	-	-	-	-	-	0000h	3	1	1	0	0
MOV.W	0x50, #0x5555	55h	55h	-	-	-	0000h	3	0	0	0	0
MOV.W	R0, (R3++, 0x50)	55h	55h	5555h	-	-	0002h	0	0	0	0	0
DEC	R3, #1	55h	55h	5555h	-	-	0000h	3	1	1	0	0
MOV.W	0x50, #0x1200	00h	12h	5555h	-	-	0000h	3	1	0	0	0
MOV.W	R1, (R3++, 0x50)	00h	12h	5555h	1200h	-	0002h	1	1	0	0	0
DEC	R3, #1	00h	12h	5555h	1200h	-	0000h	3	1	1	0	0
MOV.W	0x50, #0x0000	00h	00h	5555h	1200h	-	0000h	3	1	1	0	0
MOV.W	R2, (R3++, 0x50)	00h	00h	5555h	1200h	0000h	0002h	2	1	1	0	0
DEC	R3, #1	00h	00h	5555h	1200h	0000h	0000h	3	1	1	0	0
MOV.W	0x50, #0x5634	34h	56h	5555h	1200h	0000h	0000h	3	0	0	1	0
MOV.W	R0, (R3++, 0x50)	34h	56h	5634h	1200h	0000h	0002h	0	0	0	1	0
DEC	R3, #1	34h	56h	5634h	1200h	0000h	0000h	3	1	1	0	0
MOV.W	0x50, #0x8118	18h	81h	5634h	1200h	0000h	0000h	3	0	0	0	1
MOV.W	R1, (R3++, 0x50)	18h	81h	5634h	8118h	0000h	0002h	1	0	0	0	1
DEC	R3, #1	18h	81h	5634h	8118h	0000h	0000h	3	1	1	0	0
MOV.W	0x50, #0x5555	55h	55h	5634h	8118h	0000h	0000h	3	0	0	0	0
MOV.W	R2, (R3++, 0x50)	55h	55h	5634h	8118h	5555h	0002h	2	0	0	0	0
DEC	R3, #1	55h	55h	5634h	8118h	5555h	0000h	3	1	1	0	0

<Note>

This instruction takes 4 cycles to transfer the contents to program memory (ROM) to Rd.

MOV[.W] Rd, m16

Instruction code	[1 0 X 0 d2d1d0 1][m7m6m5m4m3m2m1m0] 8100H(RAM), A100H(SFR)
Argument	Rd = 3bit(R select), m16 = 16bit(Lower 8bit valid for operation code)
Word count	1
Cycle count	1
Function	if "m16" is even : Hibyte(Rd)←(m16+1), Lobyte(Rd)←(m16) if "m16" is odd : Hibyte(Rd)←(m16), Lobyte(Rd)←(m16-1) (PC)←(PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers the contents of 2-byte data memory (RAM) location or SFR (one of the registers dedicated to control the internal peripheral devices) designated by m16 to the lower-order 8-bit positions of the general-purpose register designated by Rd. The legitimate value range designated by Rd is from R0 to R7. The 2-byte destination address is determined according to the following rules:

・If m16 is an even number, the contents of the odd address (m16+1) are transferred to the higher-order 8-bit positions of Rd and those of the even address (m16) to the lower-order 8-bit positions of Rd.

・If m16 is an odd number, the contents of the odd address (m16) are transferred to the higher-order 8-bit positions of Rd and those of the even address (m16–1) to the lower-order 8-bit positions of Rd.

The compiler generates the instruction code while regarding RAM or SFR as the destination of transfer according to the value of m16 (second operand data).

・ When specifying a RAM location, specify m16 with a value from 00H to FFH (0000H to 00FFH). It is disallowed to specify a RAM address not lower than 100H.

・ When specifying a SFR, specify m16 with a value from 7F00H to 7FFFH.

The basic types of generated instruction code are 8100H (RAM) and A100H (SFR), respectively. The lower-order 8 bits of m16 are reflected in the behavior of the instruction code.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50, #0x05555	55h	55h	-	-	-	-	-	0	0	0	0
MOV.W	R3, #05555	55h	55h	-	-	-	5555h	3	0	0	0	0
MOV.W	R0, 0x50	55h	55h	5555h	-	-	5555h	0	0	0	0	0
MOV.W	0x50, #0x1200	00h	12h	5555h	-	-	5555h	0	1	0	0	0
MOV.W	R3, #06666	00h	12h	5555h	-	-	6666h	3	0	0	0	0
MOV.W	R1, 0x50	00h	12h	5555h	1200h	-	6666h	1	1	0	0	0
MOV.W	0x50, #0x0000	00h	00h	5555h	1200h	-	6666h	1	1	1	0	0
MOV.W	R3, #0x3333	00h	00h	5555h	1200h	-	3333h	3	0	0	0	0
MOV.W	R2, 0x50	00h	00h	5555h	1200h	0000h	3333h	2	1	1	0	0
MOV.W	0x50, #0x3456	56h	34h	5555h	1200h	0000h	3333h	2	0	0	1	0
MOV.W	R3, #0x6655	56h	34h	5555h	1200h	0000h	6655h	3	0	0	0	0
MOV.W	R0, 0x50	56h	34h	3456h	1200h	0000h	6655h	0	0	0	1	0
MOV.W	0x50, #0x8118	18h	81h	3456h	1200h	0000h	6655h	0	0	0	0	1
MOV.W	R3, #0x3366	18h	81h	3456h	1200h	0000h	3366h	3	0	0	0	0
MOV.W	R1, 0x50	18h	81h	3456h	8118h	0000h	3366h	1	0	0	0	1
MOV.W	0x50, #0x5555	55h	55h	3456h	8118h	0000h	3366h	1	0	0	0	0
MOV.W	R3, #0x6355	55h	55h	3456h	8118h	0000h	6355h	3	0	0	0	0
MOV.W	R2, 0x50	55h	55h	3456h	8118h	5555h	6355h	2	0	0	0	0

MOV[.W] Rx, #imm8

Instruction code	[0 1 0 0 0 1 1 1][i7i6i5i4i3i2i1i0] 4700H
Argument	imm8 = 8bit(immediate data)
Word count	1
Cycle count	1
Function	(Rx)←16bit data(Hibyte=00H, Lobyte=#imm8), (PC)←(PC)+2
Affected flags	Z8, Z16, P, S

[Description]

This instruction transfers immediate data designated by imm8 to the general-purpose register Rx designated indirectly by the value of bits 12 to 15 (N0 to N3) of the PSW.
The legitimate value range designated by imm8 is from 0 to FF.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x3456	-	-	-	-	-	3456h	3	0	0	1	0
MOV.W	R2, #0x2222	-	-	-	-	2222h	3456h	2	0	0	0	0
MOV.W	R1, #0x1111	-	-	-	1111h	2222h	3456h	1	0	0	0	0
MOV.W	R0, #0x0000	-	-	0000h	1111h	2222h	3456h	0	1	1	0	0
MOV.W	0x50, #0x6666	66h	66h	0000h	1111h	2222h	3456h	0	0	0	0	0
MOV.W	Rx, #0x55	66h	66h	0055h	1111h	2222h	3456h	0	0	0	0	0
DEC	R1	66h	66h	0055h	1110h	2222h	3456h	1	0	0	1	0
MOV.W	Rx, #0x00	66h	66h	0055h	0000h	2222h	3456h	1	1	1	0	0
INC	R2	66h	66h	0055h	0000h	2223h	3456h	2	0	0	1	0
MOV.W	Rx, #0x34	66h	66h	0055h	0000h	0034h	3456h	2	0	0	1	0
SWPB	R3	66h	66h	0055h	0000h	0034h	5634h	3	0	0	1	0
MOV.W	Rx, #0x81	66h	66h	0055h	0000h	0034h	0081h	3	0	0	0	0
MOV.W	0x50, #0x8118	18h	81h	0055h	0000h	0034h	0081h	3	0	0	0	1
MOV.W	Rx, #0xFF	18h	81h	0055h	0000h	0034h	00FFh	3	0	0	0	0
NOT	R0	18h	81h	FFAAh	0000h	0034h	00FFh	0	0	0	0	1
MOV.W	Rx, #0x55	18h	81h	0055h	0000h	0034h	00FFh	0	0	0	0	0

<Note>

The higher-order 8-bit positions of Rx are loaded with 00H.

MOVF.B (Rb, Rd, ±n), Rs

Instruction code	[0 1 1 1 0 1 1 0][d3d2d1d0 1 s2s1s0][0 b2b1b0 n11 to n8][n7 to n0] 7608H
Argument	Rb = 3bit(Rb select), Rd = 4bit(R select), n = 12bit(signed), Rs = 3bit(R select)
Word count	2
Cycle count	3 or 4
Function	[Rb<<16+Rd±n]←Lobyte (Rs), (PC)←(PC)+4
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers the contents of the lower-order 8 bits of the general-purpose register designated by Rs to the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the 32-bit address of which the higher-order 16 bits are the contents of the base register (Rb) and the lower-order 16 bits are the result of the arithmetic operation*1 performed on the contents of Rd and n. The legitimate value range designated by Rd is from R0 to R15, that by Rb is from R8 to R13, that by Rs is from R0 to R7, and that by n is that of signed 12-bit data (-2048 to 2047).

*1: Any carry or borrow resulting from the arithmetic operation performed on the lower-order 16 bits is reflected in the higher-order 16 bits.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	R8	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50, #0x6666	66h	66h	-	-	-	-	-	-	0	0	0	0
MOV.W	R3, #0x0000	66h	66h	-	-	-	0000h	-	3	1	1	0	0
MOV.W	R0, #0x5555	66h	66h	5555h	-	-	0000h	-	0	0	0	0	0
MOV.W	R8, #0x0000	66h	66h	5555h	-	-	0000h	0000h	8	1	1	0	0
MOVF.B	(R8, R3, 0x50), R0	55h	66h	5555h	-	-	0000h	0000h	0	0	0	0	0
MOV.W	R1, #0x1200	55h	66h	5555h	1200h	-	0000h	0000h	1	1	0	0	0
MOVF.B	(R8, R3, 0x50), R1	00h	66h	5555h	1200h	-	0000h	0000h	1	1	1	0	0
MOV.W	R2, #0x0000	00h	66h	5555h	1200h	0000h	0000h	0000h	2	1	1	0	0
MOVF.B	(R8, R3, 0x50), R2	00h	66h	5555h	1200h	0000h	0000h	0000h	2	1	1	0	0
MOV.W	R0, #0x5634	00h	66h	5634h	1200h	0000h	0000h	0000h	0	0	0	1	0
MOVF.B	(R8, R3, 0x50), R0	34h	66h	5634h	1200h	0000h	0000h	0000h	0	0	0	1	0
MOV.W	R1, #0x1881	34h	66h	5634h	1881h	0000h	0000h	0000h	1	0	0	0	0
MOVF.B	(R8, R3, 0x50), R1	81h	66h	5634h	1881h	0000h	0000h	0000h	1	0	0	0	1
MOV.W	R2, #0x5555	81h	66h	5634h	1881h	5555h	0000h	0000h	2	0	0	0	0
MOVF.B	(R8, R3, 0x50), R2	55h	66h	5634h	1881h	5555h	0000h	0000h	2	0	0	0	0

<Note>

This instruction takes 4 cycles to transfer the contents of Rs to program memory (ROM). However, no data can actually be transferred to ROM.

MOVF.B (Rb, --Rd, ±n), Rs

Instruction code	[0 1 1 0 1 1 1 0][d3d2d1d0 1 s2s1s0][0 b2b1b0 n11 to n8][n7 to n0] 6E08H
Argument	Rb = 3bit(Rb select), Rd = 4bit(R select), n = 12bit(signed), Rs = 3bit(R select)
Word count	2
Cycle count	3 or 4
Function	(Rd)←(Rd)-1, if Borrow : (Rb)←(Rb)-1 [Rb<<16+Rd±n]←Lobyte(Rs) (PC)←(PC)+4
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction firstly subtracts 1 from the contents of the general-purpose register Rd. Rd is decremented if a borrow occurs as the result of the subtraction performed on Rd.

Subsequently, the instruction transfers the contents of the lower-order 8 bits of the general-purpose register Rs to the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the 32-bit address of which the higher-order 16 bits are the contents of the base register (Rb) and the lower-order 16 bits are the result of the arithmetic operation*1 performed on the contents of Rd and n.

The legitimate value range designated by Rd is from R0 to R15, that by Rb is from R8 to R13, that by Rs is from R0 to R7, and that by n is that of signed 12-bit data (-2048 to 2047).

*1: Any carry or borrow resulting from the arithmetic operation performed on the lower-order 16 bits is reflected in the higher-order 16 bits.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	R8	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50, #0x6666	66h	66h	-	-	-	-	-	-	0	0	0	0
MOV.W	R3, #0x0001	66h	66h	-	-	-	0001h	-	3	0	0	1	0
MOV.W	R0, #0x5555	66h	66h	5555h	-	-	0001h	-	0	0	0	0	0
MOV.W	R8, #0x0000	66h	66h	5555h			0001h	0000h	8	1	1	0	0
MOVF.B	(R8, --R3, 0x50), R0	55h	66h	5555h	-	-	0000h	0000h	0	0	0	0	0
INC	R3	55h	66h	5555h	-	-	0001h	0000h	3	0	0	1	0
MOV.W	R1, #0x1200	55h	66h	5555h	1200h	-	0001h	0000h	1	1	0	0	0
MOVF.B	(R8, --R3, 0x50), R1	00h	66h	5555h	1200h	-	0000h	0000h	1	1	1	0	0
INC	R3	00h	66h	5555h	1200h	-	0001h	0000h	3	0	0	1	0
MOV.W	R2, #0x0000	00h	66h	5555h	1200h	0000h	0001h	0000h	2	1	1	0	0
MOVF.B	(R8, --R3, 0x50), R2	00h	66h	5555h	1200h	0000h	0000h	0000h	2	1	1	0	0
INC	R3	00h	66h	5555h	1200h	0000h	0001h	0000h	3	0	0	1	0
MOV.W	R0, #0x5634	00h	66h	5634h	1200h	0000h	0001h	0000h	0	0	0	1	0
MOVF.B	(R8, --R3, 0x50), R0	34h	66h	5634h	1200h	0000h	0000h	0000h	0	0	0	1	0
INC	R3	34h	66h	5634h	1200h	0000h	0001h	0000h	3	0	0	1	0
MOV.W	R1, #0x1881	34h	66h	5634h	1881h	0000h	0001h	0000h	1	0	0	0	0
MOVF.B	(R8, --R3, 0x50), R1	81h	66h	5634h	1881h	0000h	0000h	0000h	1	0	0	0	1
INC	R3	81h	66h	5634h	1881h	0000h	0001h	0000h	3	0	0	1	0
MOV.W	R2, #0x5555	81h	66h	5634h	1881h	5555h	0001h	0000h	2	0	0	0	0
MOVF.B	(R8, --R3, 0x50), R2	55h	66h	5634h	1881h	5555h	0000h	0000h	2	0	0	0	0
INC	R3	55h	66h	5634h	1881h	5555h	0001h	0000h	3	0	0	1	0

<Note>

This instruction takes 4 cycles to transfer the contents of Rs to program memory (ROM). However, no data can actually be transferred to ROM.

MOVF.B (Rb, Rd++, ±n), Rs

Instruction code	[0 1 1 0 0 1 1 0][d3d2d1d0 1 s2s1s0][0 b2b1b0 n11 to n8][n7 to n0] 6608H
Argument	Rb = 3bit(Rb select), Rd = 4bit(R select), n = 12bit(signed), Rs = 3bit(R select)
Word count	2
Cycle count	3 or 4
Function	[Rb<<16+Rd±n] ← Lobyte (Rs) (Rd)←(Rd)+1, if Carry : (Rb)←(Rb)+1 (PC)←(PC)+4
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers the contents of the lower-order 8 bits of the general-purpose register designated by Rs to the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the 32-bit address of which the higher-order 16 bits are the contents of the base register (Rb) and the lower-order 16 bits are the result of the arithmetic operation*1 performed on the contents of Rd and n. Subsequently, the contents of Rd are incremented by 1. Rb is incremented if a carry occurs as the result of the addition performed on Rd.

The legitimate value range designated by Rd is from R0 to R15, that by Rb is from R8 to R13, that by Rs is from R0 to R7, and that by n is that of signed 12-bit data (-2048 to 2047).

*1: Any carry or borrow resulting from the arithmetic operation performed on the lower-order 16 bits is reflected in the higher-order 16 bits.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	R8	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50, #0x6666	66h	66h	-	-	-	-	-	-	0	0	0	0
MOV.W	R3, #0x0000	66h	66h	-	-	-	0000h	-	3	1	1	0	0
MOV.W	R0, #0x5555	66h	66h	5555h	-	-	0000h	-	0	0	0	0	0
MOVF.B	(R8, R3++, 0x50), R0	55h	66h	5555h	-	-	0001h	0000h	0	0	0	0	0
DEC	R3	55h	66h	5555h	-	-	0000h	0000h	3	1	1	0	0
MOV.W	R1, #0x1200	55h	66h	5555h	1200h	-	0000h	0000h	1	1	0	0	0
MOVF.B	(R8, R3++, 0x50), R1	00h	66h	5555h	1200h	-	0001h	0000h	1	1	1	0	0
DEC	R3	00h	66h	5555h	1200h	-	0000h	0000h	3	1	1	0	0
MOV.W	R2, #0x0000	00h	66h	5555h	1200h	0000h	0000h	0000h	2	1	1	0	0
MOVF.B	(R8, R3++, 0x50), R2	00h	66h	5555h	1200h	0000h	0001h	0000h	2	1	1	0	0
DEC	R3	00h	66h	5555h	1200h	0000h	0000h	0000h	3	1	1	0	0
MOV.W	R0, #0x5634	00h	66h	5634h	1200h	0000h	0000h	0000h	0	0	0	1	0
MOVF.B	(R8, R3++, 0x50), R0	34h	66h	5634h	1200h	0000h	0001h	0000h	0	0	0	1	0
DEC	R3	34h	66h	5634h	1200h	0000h	0000h	0000h	3	1	1	0	0
MOV.W	R1, #0x1881	34h	66h	5634h	1881h	0000h	0000h	0000h	1	0	0	0	0
MOVF.B	(R8, R3++, 0x50), R1	81h	66h	5634h	1881h	0000h	0001h	0000h	1	0	0	0	1
DEC	R3	81h	66h	5634h	1881h	0000h	0000h	0000h	3	1	1	0	0
MOV.W	R2, #0x5555	81h	66h	5634h	1881h	5555h	0000h	0000h	2	0	0	0	0
MOVF.B	(R8, R3++, 0x50), R2	55h	66h	5634h	1881h	5555h	0001h	0000h	2	0	0	0	0
DEC	R3	55h	66h	5634h	1881h	5555h	0000h	0000h	3	1	1	0	0

<Note>

This instruction takes 4 cycles to transfer the contents of Rs to program memory (ROM). However, no data can actually be transferred to ROM.

MOVF.B (Rd), Rs

Instruction code	[0 1 1 1 0 1 1 0][d3d2d1d0 0 s2s1s0] 7600H
Argument	Rd = 4bit(R select), Rs = 3bit(R select)
Word count	1
Cycle count	2 or 3
Function	[R8<<16+Rd]←Lobyte(Rs), (PC)←(PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers the contents of the lower-order 8 bits of the general-purpose register designated by Rs to the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the 32-bit address of which the higher-order 16 bits are the contents of R8 (Rb0) and the lower-order 16 bits are the contents of Rd.
The legitimate value range designated by Rd is from R0 to R15 and that by Rs is from R0 to R7.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50, #0x6666	66h	66h	-	-	-	-	-	0	0	0	0
MOV.W	R3, #0x0050	66h	66h	-	-	-	0050h	3	0	0	0	0
MOV.W	R0, #0x5555	66h	66h	5555h	-	-	0050h	0	0	0	0	0
MOVF.B	(R3), R0	55h	66h	5555h	-	-	0050h	0	0	0	0	0
MOV.W	R1, #0x1200	55h	66h	5555h	1200h	-	0050h	1	1	0	0	0
MOVF.B	(R3), R1	00h	66h	5555h	1200h	-	0050h	1	1	1	0	0
MOV.W	R2, #0x0000	00h	66h	0055h	1200h	0000h	0050h	2	1	1	0	0
MOVF.B	(R3), R2	00h	66h	0055h	1200h	0000h	0050h	2	1	1	0	0
MOV.W	R0, #0x5634	00h	66h	5634h	1200h	0000h	0050h	0	0	0	1	0
MOVF.B	(R3), R0	34h	66h	5634h	1200h	0000h	0050h	0	0	0	1	0
MOV.W	R1, #0x1881	34h	66h	5634h	1881h	0000h	0050h	1	0	0	0	0
MOVF.B	(R3), R1	34h	66h	5634h	1881h	0000h	0050h	1	0	0	0	1
MOV.W	R2, #0x5555	34h	66h	5634h	1881h	5555h	0050h	2	0	0	0	0
MOVF.B	(R3), R2	81h	66h	5634h	1881h	5555h	0050h	2	0	0	0	0

<Note>

In this case, Rb0 refers to R8.
This instruction takes 3 cycles to transfer the contents of Rs to program memory (ROM). However, no data can actually be transferred to ROM.

MOVF.B (--Rd), Rs

Instruction code	[0 1 1 0 1 1 1 0][d3d2d1d0 0 s2s1s0] 6E00H
Argument	Rd = 4bit(R select), Rs = 3bit(R select)
Word count	1
Cycle count	2 or 3
Function	(Rd)←(Rd)-1, if Borrow : (R8)←(R8)-1 [R8<<16+Rd]←Lobyte(Rs) (PC)←(PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction firstly subtracts 1 from the contents of the general-purpose register designated by Rd. R8 is decremented if a borrow occurs as the result of the subtraction performed on Rd.

Subsequently, the instruction transfers the contents of the lower-order 8 bits of the general-purpose register Rs to the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the 32-bit address of which the higher-order 16 bits are the contents of R8 (Rb0) and the lower-order 16 bits are the contents of Rd.

The legitimate value range designated by Rd is from R0 to R15 and that by Rs is from R0 to R7

.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50, #0x6666	66h	66h	-	-	-	-	-	0	0	0	0
MOV.W	R3, #0x0051	66h	66h	-	-	-	0051h	3	0	0	1	0
MOV.W	R0, #0x5555	66h	66h	5555h	-	-	0051h	0	0	0	0	0
MOVF.B	(--R3), R0	55h	66h	5555h	-	-	0050h	0	0	0	0	0
MOV.W	R3	55h	66h	5555h	-	-	0051h	3	0	0	1	0
MOVF.B	R1, #0x1200	55h	66h	5555h	1200h	-	0051h	1	1	0	0	0
MOV.W	(--R3), R1	00h	66h	5555h	1200h	-	0050h	1	1	1	0	0
MOVF.B	R3	00h	66h	5555h	1200h	-	0051h	3	0	0	1	0
MOV.W	R2, #0x0000	00h	66h	5555h	1200h	0000h	0051h	2	1	1	0	0
MOVF.B	(--R3), R2	00h	66h	5555h	1200h	0000h	0050h	2	1	1	0	0
MOV.W	R3	00h	66h	5555h	1200h	0000h	0051h	3	0	0	1	0
MOVF.B	R0, #0x5634	00h	66h	5634h	1200h	0000h	0051h	0	0	0	1	0
MOV.W	(--R3), R0	34h	66h	5634h	1200h	0000h	0050h	0	0	0	1	0
MOVF.B	R3	34h	66h	5634h	1200h	0000h	0051h	3	0	0	1	0
MOV.W	R1, #0x1881	34h	66h	5634h	1881h	0000h	0051h	1	0	0	0	0
MOVF.B	(--R3), R1	81h	66h	5634h	1881h	0000h	0050h	1	0	0	0	1

<Note>

In this case, Rb0 refers to R8.

This instruction takes 3 cycles to transfer the contents of Rs to program memory (ROM). However, no data can actually be transferred to ROM.

MOVF.B (Rd++), Rs

Instruction code	[0 1 1 0 0 1 1 0][d3d2d1d0 0 s2s1s0] 6600H
Argument	Rd = 4bit(R select), Rs = 3bit(R select)
Word count	1
Cycle count	2 or 3
Function	[R8<<16+Rd]←Lobyte(Rs) (Rd)←(Rd)+1, if Carry : (R8)←(R8)+1 (PC)←(PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers the contents of the lower-order 8 bits of the general-purpose register designated by Rs to the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the 32-bit address of which the higher-order 16 bits are the contents of R8 (Rb0) and the lower-order 16 bits are the contents of Rd. Subsequently, the instruction adds 1 to the contents of Rd. R8 is incremented if a carry occurs as the result of the addition performed on Rd.
The legitimate value range designated by Rd is from R0 to R15 and that by Rs is from R0 to R7.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50, #0x6666	66h	66h	-	-	-	-	-	0	0	0	0
MOV.W	R3, #0x0050	66h	66h	-	-	-	0050h	3	0	0	0	0
MOV.W	R0, #0x5555	66h	66h	5555h	-	-	0050h	0	0	0	0	0
MOVF.B	(R3++), R0	55h	66h	5555h	-	-	0051h	0	0	0	0	0
DEC	R3	55h	66h	5555h	-	-	0050h	3	0	0	0	0
MOV.W	R1, #0x1200	55h	66h	5555h	1200h	-	0050h	1	1	0	0	0
MOVF.B	(R3++), R1	00h	66h	5555h	1200h	-	0051h	1	1	1	0	0
DEC	R3	00h	66h	5555h	1200h	-	0050h	3	0	0	0	0
MOV.W	R2, #0x0000	00h	66h	5555h	1200h	0000h	0050h	2	1	1	0	0
MOVF.B	(R3++), R2	00h	66h	5555h	1200h	0000h	0051h	2	1	1	0	0
DEC	R3	00h	66h	5555h	1200h	0000h	0050h	3	0	0	0	0
MOV.W	R0, #0x5634	00h	66h	5634h	1200h	0000h	0050h	0	0	0	1	0
MOVF.B	(R3++), R0	34h	66h	5634h	1200h	0000h	0051h	0	0	0	1	0
DEC	R3	34h	66h	5634h	1200h	0000h	0050h	3	0	0	0	0
MOV.W	R1, #0x1881	34h	66h	5634h	1881h	0000h	0050h	1	0	0	0	0
MOVF.B	(R3++), R1	81h	66h	5634h	1881h	0000h	0051h	1	0	0	0	1
DEC	R3	81h	66h	5634h	1881h	0000h	0050h	3	0	0	0	0
MOV.W	R2, #0x5555	81h	66h	5634h	1881h	5555h	0050h	2	0	0	0	0
MOVF.B	(R3++), R2	55h	66h	5634h	1881h	5555h	0051h	2	0	0	0	0
DEC	R3	55h	66h	5634h	1881h	5555h	0050h	3	0	0	0	0

<Note>

In this case, Rb0 refers to R8.
This instruction takes 3 cycles to transfer the contents of Rs to program memory (ROM). However, no data can actually be transferred to ROM.

MOVF.B Rd, (Rb, Rs, ±n)

Instruction code	[0 1 1 1 0 1 0 0][s3s2s1s0 1 d2d1d0][0 b2b1b0 n11 to n8][n7 to n0] 7408H
Argument	Rd = 3bit(R select), Rb = 3bit(Rb select), Rs = 4bit(R select), n = 12bit(signed)
Word count	2
Cycle count	3 or 4
Function	Lobyte (Rd)← [Rb<<16+Rs±n], (PC)←(PC)+4
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers the contents of the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the 32-bit address of which the higher-order 16 bits are the contents of the general-purpose register designated by the base register (Rb) and the lower-order 16 bits are the result of the arithmetic operation*1 performed on the contents of Rs and n, to the lower-order 8-bit positions of the general-purpose register Rd.

The legitimate value range designated by Rd is from R0 to R7, that by Rb is from R8 to R13, that by Rs is from R0 to R15, and that by n is that of signed 12-bit data (-2048 to 2047).

*1: Any carry or borrow resulting from the arithmetic operation performed on the lower-order 16 bits is reflected in the higher-order 16 bits.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	R8	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0000	-	-	-	-	-	0000h	-	3	1	1	0	0
MOV.W	0x50, #0x5555	55h	55h	-	-	-	0000h	-	3	0	0	0	0
MOVF.B	R0, (R8, R3, 0x50)	55h	55h	0055h	-	-	0000h	0000h	0	0	0	0	0
MOV.W	0x50, #0x1200	00h	12h	0055h	-	-	0000h	0000h	0	1	0	0	0
MOVF.B	R1, (R8, R3, 0x50)	00h	12h	0055h	0000h	-	0000h	0000h	1	1	1	0	0
MOV.W	0x50, #0x0000	00h	00h	0055h	0000h	-	0000h	0000h	1	1	1	0	0
MOVF.B	R2, (R8, R3, 0x50)	00h	00h	0055h	0000h	0000h	0000h	0000h	2	1	1	0	0
MOV.W	0x50, #0x5634	34h	56h	0055h	0000h	0000h	0000h	0000h	2	0	0	1	0
MOVF.B	R0, (R8, R3, 0x50)	34h	56h	0034h	0000h	0000h	0000h	0000h	0	0	0	1	0
MOV.W	0x50, #0x1881	81h	18h	0034h	0000h	0000h	0000h	0000h	0	0	0	0	0
MOVF.B	R1, (R8, R3, 0x50)	81h	18h	0034h	0081h	0000h	0000h	0000h	1	0	0	0	1
MOV.W	0x50, #0x5555	55h	55h	0034h	0081h	0000h	0000h	0000h	1	0	0	0	0
MOVF.B	R2, (R8, R3, 0x50)	55h	55h	0034h	0081h	0055h	0000h	0000h	2	0	0	0	0

<Note>

The higher-order 8 bits of Rd are loaded with 00H.

This instruction takes 4 cycles to transfer the contents of program memory (ROM) to Rd.

MOVF.B Rd, (Rb, --Rs,±n)

Instruction code	[0 1 1 0 1 1 0 0][s3s2s1s0 1 d2d1d0][0 b2b1b0 n11 to n8][n7 to n0] 6C08H
Argument	Rd = 3bit(R select), Rb = 3bit(Rb select), Rs = 4bit(R select), n = 12bit(signed)
Word count	2
Cycle count	3 or 4
Function	(Rs)←(Rs)-1, if Borrow : (Rb)←(Rb)-1 Lobyte (Rd)← [Rb<<16+Rs±n] (PC)←(PC)+4
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction firstly subtracts 1 from the contents of the general-purpose register designated by Rs. Rb is decremented if a borrow occurs as the result of the subtraction performed on Rs.

Subsequently, the instruction transfers the contents of the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the 32-bit address, of which the higher-order 16 bits are the contents of the base register (Rb) and the lower-order 16 bits are the result of the arithmetic operation*1 performed on the contents of Rs and n, to the lower-order 8-bit positions of the general-purpose register Rd.

The legitimate value range designated by Rd is from R0 to R7, that by Rb is from R8 to R13, that by Rs is from R0 to R15, and that by n is that of signed 12-bit data (-2048 to 2047).

*1: Any carry or borrow resulting from the arithmetic operation performed on the lower-order 16 bits is reflected in the higher-order 16 bits.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	R8	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0001	-	-	-	-	-	0001h	-	3	0	0	1	0
MOV.W	0x50, #0x5555	55h	55h	-	-	-	0001h	-	3	0	0	0	0
MOVF.B	R0, (R8, --R3, 0x50)	55h	55h	0055h	-	-	0000h	0000h	0	0	0	0	0
INC	R3	55h	55h	0055h	-	-	0001h	0000h	3	0	0	1	0
MOV.W	0x50, #0x1200	00h	12h	0055h	-	-	0001h	0000h	3	1	0	0	0
MOVF.B	R1, (R8, --R3, 0x50)	00h	12h	0055h	0000h	-	0000h	0000h	1	1	1	0	0
INC	R3	00h	12h	0055h	0000h	-	0001h	0000h	3	0	0	1	0
MOV.W	0x50, #0x0000	00h	00h	0055h	0000h	-	0001h	0000h	3	1	1	0	0
MOVF.B	R2, (R8, --R3, 0x50)	00h	00h	0055h	0000h	0000h	0000h	0000h	2	1	1	0	0
INC	R3	00h	00h	0055h	0000h	0000h	0001h	0000h	3	0	0	1	0
MOV.W	0x50, #0x5634	34h	56h	0055h	0000h	0000h	0001h	0000h	3	0	0	1	0
MOVF.B	R0, (R8, --R3, 0x50)	34h	56h	0034h	0000h	0000h	0000h	0000h	0	0	0	1	0
INC	R3	34h	56h	0034h	0000h	0000h	0001h	0000h	3	0	0	1	0
MOV.W	0x50, #0x1881	81h	18h	0034h	0000h	0000h	0001h	0000h	3	0	0	0	0
MOVF.B	R1, (R8, --R3, 0x50)	81h	18h	0034h	0081h	0000h	0000h	0000h	1	0	0	0	1
INC	R3	81h	18h	0034h	0081h	0000h	0001h	0000h	3	0	0	1	0
MOV.W	0x50, #0x5555	55h	55h	0034h	0081h	0000h	0001h	0000h	3	0	0	0	0
MOVF.B	R2, (R8, --R3, 0x50)	55h	55h	0034h	0081h	0055h	0000h	0000h	2	0	0	0	0
INC	R3	55h	55h	0034h	0081h	0055h	0001h	0000h	3	0	0	1	0

<Note>

The higher-order 8 bits of Rd are loaded with 00H.

This instruction takes 4 cycles to transfer the contents of program memory (ROM) to Rd.

MOVF.B Rd, (Rb, Rs++, ±n)

Instruction code	[0 1 1 0 0 1 0 0][s3s2s1s0 1 d2d1d0][0 b2b1b0 n11 to n8][n7 to n0] 6408H
Argument	Rd = 3bit(R select), Rb = 3bit(Rb select), Rs = 4bit(R select), n = 12bit(signed)
Word count	2
Cycle count	3 or 4
Function	Lobyte (Rd)← [Rb<<16+Rs±n] (Rs)←(Rs)+1, if Carry : (Rb)←(Rb)+1 (PC)←(PC)+4
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers the contents of the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the 32-bit address, of which the higher-order 16 bits are the contents of the general-purpose register designated by the base register (Rb) and the lower-order 16 bits are the result of the arithmetic operation*1 performed on the contents of Rs and n, to the lower-order 8-bit positions of the general-purpose register Rd. Subsequently, the instruction adds 1 to the contents of Rs. Rb is incremented if a carry occurs as the result of the addition performed on Rs.

The legitimate value range designated by Rd is from R0 to R7, that by Rb is from R8 to R13, that by Rs is from R0 to R15, and that by n is that of signed 12-bit data (-2048 to 2047).

*1: Any carry or borrow resulting from the arithmetic operation performed on the lower-order 16 bits is reflected in the higher-order 16 bits.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	R8	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0000	-	-	-	-	-	0000h	-	3	1	1	0	0
MOV.W	0x50, #0x5555	55h	55h	-	-	-	0000h	-	3	0	0	0	0
MOVF.B	R0, (R8, R3++, 0x50)	55h	55h	0055h	-	-	0001h	0000h	0	0	0	0	0
DEC	R3	55h	55h	0055h	-	-	0000h	0000h	3	1	1	0	0
MOV.W	0x50, #0x1200	00h	12h	0055h	-	-	0000h	0000h	3	1	0	0	0
MOVF.B	R1, (R8, R3++, 0x50)	00h	12h	0055h	0000h	-	0001h	0000h	1	1	1	0	0
DEC	R3	00h	12h	0055h	0000h	-	0000h	0000h	3	1	1	0	0
MOV.W	0x50, #0x0000	00h	00h	0055h	0000h	-	0000h	0000h	3	1	1	0	0
MOVF.B	R2, (R8, R3++, 0x50)	00h	00h	0055h	0000h	0000h	0001h	0000h	2	1	1	0	0
DEC	R3	00h	00h	0055h	0000h	0000h	0000h	0000h	3	1	1	0	0
MOV.W	0x50, #0x5634	34h	56h	0055h	0000h	0000h	0000h	0000h	3	0	0	1	0
MOVF.B	R0, (R8, R3++, 0x50)	34h	56h	0034h	0000h	0000h	0001h	0000h	0	0	0	1	0
DEC	R3	34h	56h	0034h	0000h	0000h	0000h	0000h	3	1	1	0	0
MOV.W	0x50, #0x1881	81h	18h	0034h	0000h	0000h	0000h	0000h	3	0	0	0	0
MOVF.B	R1, (R8, R3++, 0x50)	81h	18h	0034h	0081h	0000h	0001h	0000h	1	0	0	0	1
DEC	R3	81h	18h	0034h	0081h	0000h	0000h	0000h	3	1	1	0	0
MOV.W	0x50, #0x5555	55h	55h	0034h	0081h	0000h	0000h	0000h	3	0	0	0	0
MOVF.B	R2, (R8, R3++, 0x50)	55h	55h	0034h	0081h	0055h	0001h	0000h	2	0	0	0	0
DEC	R3	55h	55h	0034h	0081h	0055h	0000h	0000h	3	1	1	0	0

<Note>

The higher-order 8 bits of Rd are loaded with 00H.

This instruction takes 4 cycles to transfer the contents of program memory (ROM) to Rd.

MOVF.B Rd, (Rs)

Instruction code	[0 1 1 1 0 1 0 0][s3s2s1s0 0 d2d1d0] 7400H
Argument	Rd = 3bit(R select), Rs = 4bit(R select)
Word count	1
Cycle count	2 or 3
Function	Lobyte (Rd)← [R8<<16+Rs], (PC)←(PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers the contents of the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the 32-bit address, of which the higher-order 16 bits are the contents of R8 (Rb0) and the lower-order 16 bits are the contents of the general-purpose register designated by Rs, to the lower-order 8-bit positions of the general-purpose register Rd.
The legitimate value range designated by Rd is from R0 to R7 and that by Rs is from R0 to R15.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0050	-	-	-	-	-	0050h	3	0	0	0	0
MOV.W	0x50, #0x5555	55h	55h	-	-	-	0050h	3	0	0	0	0
MOVF.B	R0, (R3)	55h	55h	0055h	-	-	0050h	0	0	0	0	0
MOV.W	0x50, #0x1200	00h	12h	0055h	-	-	0050h	0	1	0	0	0
MOVF.B	R1, (R3)	00h	12h	0055h	0000h	-	0050h	1	1	1	0	0
MOV.W	0x50, #0x0000	00h	00h	0055h	0000h	-	0050h	1	1	1	0	0
MOVF.B	R2, (R3)	00h	00h	0055h	0000h	0000h	0050h	2	1	1	0	0
MOV.W	0x50, #0x5634	34h	56h	0055h	0000h	0000h	0050h	2	0	0	1	0
MOVF.B	R0, (R3)	34h	56h	0034h	0000h	0000h	0050h	0	0	0	1	0
MOV.W	0x50, #0x1881	81h	18h	0034h	0000h	0000h	0050h	0	0	0	0	0
MOVF.B	R1, (R3)	81h	18h	0034h	0081h	0000h	0050h	1	0	0	0	1
MOV.W	0x50, #0x5555	55h	55h	0034h	0081h	0000h	0050h	1	0	0	0	0
MOVF.B	R2, (R3)	55h	55h	0034h	0081h	0055h	0050h	2	0	0	0	0

<Note>

In this case, Rb0 refers to R8.

The higher-order 8 bits of Rd are loaded with 00H.

The instruction takes 3 cycles to transfer the contents of program memory (ROM) to Rd.

MOVF.B Rd, (--Rs)

Instruction code	[0 1 1 0 1 1 0 0][s3s2s1s0 0 d2d1d0] 6C00H
Argument	Rd = 3bit(R select), Rs = 4bit(R select)
Word count	1
Cycle count	2 or 3
Function	(Rs)←(Rs)-1, if Borrow : (R8)←(R8)-1 Lobyte (Rd)← [R8<<16+Rs] (PC)←(PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction firstly subtracts 1 from the contents of the general-purpose register designated by Rs. R8 is decremented if a borrow occurs as the result of the subtraction performed on Rs.

Subsequently, the instruction transfers the contents of the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the 32-bit address, of which the higher-order 16 bits are the contents of R8 (Rb0) and the lower-order 16 bits are the contents of Rs, to the lower-order 8-bit positions of the general-purpose register Rd.

The legitimate value range designated by Rd is from R0 to R7 and that by Rs is from R0 to R15.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0051	-	-	-	-	-	0051h	3	0	0	1	0
MOV.W	0x50, #0x5555	55h	55h	-	-	-	0051h	3	0	0	0	0
MOVF.B	R0, (--R3)	55h	55h	0055h	-	-	0050h	0	0	0	0	0
INC	R3	55h	55h	0055h	-	-	0051h	3	0	0	1	0
MOV.W	0x50, #0x1200	12h	00h	0055h	-	-	0051h	3	1	0	0	0
MOVF.B	R1, (--R3)	12h	00h	0055h	0000h	-	0050h	1	1	1	0	0
INC	R3	12h	00h	0055h	0000h	-	0051h	3	0	0	1	0
MOV.W	0x50, #0x0000	00h	00h	0055h	0000h	-	0051h	3	1	1	0	0
MOVF.B	R2, (--R3)	00h	00h	0055h	0000h	0000h	0050h	2	1	1	0	0
INC	R3	00h	00h	0055h	0000h	0000h	0051h	3	0	0	1	0
MOV.W	0x50, #0x5634	56h	34h	0055h	0000h	0000h	0051h	3	0	0	1	0
MOVF.B	R0, (--R3)	56h	34h	0034h	0000h	0000h	0050h	0	0	0	1	0
INC	R3	56h	34h	0034h	0000h	0000h	0051h	3	0	0	1	0
MOV.W	0x50, #0x1881	81h	18h	0034h	0000h	0000h	0051h	3	0	0	0	0
MOVF.B	R1, (--R3)	81h	18h	0034h	0081h	0000h	0050h	1	0	0	0	1

<Note>

In this case, Rb0 refers to R8.

The higher-order 8 bits of Rd are loaded with 00H.

The instruction takes 3 cycles to transfer the contents of program memory (ROM) to Rd.

MOVF.B Rd, (Rs++)

Instruction code	[0 1 1 0 0 1 0 0][s3s2s1s0 0 d2d1d0] 6400H
Argument	Rd = 3bit(R select), Rs = 4bit(R select)
Word count	1
Cycle count	2 or 3
Function	Lobyte (Rd) ← [R8<<16+Rs] (Rs)←(Rs)+1, if Carry : (R8)←(R8)+1 (PC) ← (PC)+2
Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction transfers the contents of the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the 32-bit address, of which the higher-order 16 bits are the contents of R8 (Rb0) and the lower-order 16 bits are the contents of the general-purpose register designated by Rs, to the lower-order 8-bit positions of the general-purpose register Rd. Subsequently, the instruction adds 1 to the contents of Rs. R8 is incremented if a carry occurs as the result of the addition performed on Rs. The legitimate value range designated by Rd is from R0 to R7 and that by Rs is from R0 to R15.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0050	-	-	-	-	-	0050h	3	0	0	0	0
MOV.W	0x50, #0x5555	55h	55h	-	-	-	0050h	3	0	0	0	0
MOVF.B	R0, (R3++)	55h	55h	0055h	-	-	0051h	0	0	0	0	0
DEC	R3	55h	55h	0055h	-	-	0050h	3	0	0	0	0
MOV.W	0x50, #0x1200	00h	12h	0055h	-	-	0050h	3	1	0	0	0
MOVF.B	R1, (R3++)	00h	12h	0055h	0000h	-	0051h	1	1	1	0	0
DEC	R3	00h	12h	0055h	0000h	-	0050h	3	0	0	0	0
MOV.W	0x50, #0x0000	00h	00h	0055h	0000h	-	0050h	3	1	1	0	0
MOVF.B	R2, (R3++)	00h	00h	0055h	0000h	0000h	0051h	2	1	1	0	0
DEC	R3	00h	00h	0055h	0000h	0000h	0050h	3	0	0	0	0
MOV.W	0x50, #0x5634	34h	56h	0055h	0000h	0000h	0050h	3	0	0	1	0
MOVF.B	R0, (R3++)	34h	56h	0034h	0000h	0000h	0051h	0	0	0	1	0
DEC	R3	34h	56h	0034h	0000h	0000h	0050h	3	0	0	0	0
MOV.W	0x50, #0x1881	81h	18h	0034h	0000h	0000h	0050h	3	0	0	0	1
MOVF.B	R1, (R3++)	81h	18h	0034h	0081h	0000h	0051h	1	0	0	0	1

<Note>

In this case, Rb0 refers to R8.

The higher-order 8 bits of Rd are loaded with 00H.

The instruction takes 3 cycles to transfer the contents of program memory (ROM) to Rd.

MOVF[.W] (Rb, Rd, ±n), Rs

Instruction code	[0 1 1 1 0 1 1 1][d3d2d1d0 1 s2s1s0][0 b2b1b0 n11 to n8][n7 to n0] 7708H
Argument	Rb = 3bit(Rb select), Rd = 4bit(R select), n = 12bit(signed), Rs = 3bit(R select)
Word count	2
Cycle count	3 or 4

Function	if (Rd±n) = even data : [Rb<<16+Rd±n+1] ←Hibyte(Rs),
	[Rb<<16+Rd±n] ←Lobyte (Rs)
	if (Rd±n) =odd data : [Rb<<16+Rd±n] ←Hibyte(Rs),
	[Rb<<16+Rd±n-1] ←Lobyte (Rs)
	(PC)←(PC)+4

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

If the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the 32-bit address, of which the higher-order 16 bits are the contents of the general-purpose register designated by the base register (Rb) and the lower-order 16 bits are the result of the arithmetic operation*1 performed on the contents of Rd and n, is at an even address, the instruction transfers the contents of the lower-order 8 bits of Rs to [Rb<<16+Rd±n] and the higher-order 8 bits of Rs to [Rb<<16+Rd±n+1].

In the case of an odd address, the instruction transfers the contents of the higher-order 8 bits of Rs to [Rb<<16+Rd±n] and the lower-order 8 bits of Rs to [Rb<<16+Rd±n-1].

The legitimate value range designated by Rd is from R0 to R15, that by Rb is from R8 to R13, that by Rs is from R0 to R7, and that by n is that of signed 12-bit data (-2048 to 2047).

*1: Any carry or borrow resulting from the arithmetic operation performed on the lower-order 16 bits is reflected in the higher-order 16 bits.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	R8	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50, #0x6666	66h	66h	-	-	-	-	-	-	0	0	0	0
MOV.W	R3, #0x0000	66h	66h	-	-	-	0000h	-	3	1	1	0	0
MOV.W	R0, #0x5555	66h	66h	5555h	-	-	0000h	-	0	0	0	0	0
MOVF.W	(R8, R3, 0x50), R0	55h	55h	5555h	-	-	0000h	0000h	0	0	0	0	0
MOV.W	R1, #0x1200	55h	55h	5555h	1200h	-	0000h	0000h	1	1	0	0	0
MOVF.W	(R8, R3, 0x50), R1	00h	12h	5555h	1200h	-	0000h	0000h	1	1	0	0	0
MOV.W	R2, #0x0000	00h	12h	5555h	1200h	0000h	0000h	0000h	2	1	1	0	0
MOVF.W	(R8, R3, 0x50), R2	00h	00h	5555h	1200h	0000h	0000h	0000h	2	1	1	0	0
MOV.W	R0, #0x5634	00h	00h	5634h	1200h	0000h	0000h	0000h	0	0	0	1	0
MOVF.W	(R8, R3, 0x50), R0	34h	56h	5634h	1200h	0000h	0000h	0000h	0	0	0	1	0
MOV.W	R1, #0x8118	34h	56h	5634h	8118h	0000h	0000h	0000h	1	0	0	0	1
MOVF.W	(R8, R3, 0x50), R1	18h	81h	5634h	8118h	0000h	0000h	0000h	1	0	0	0	1
MOV.W	R2, #0x5555	18h	81h	5634h	8118h	5555h	0000h	0000h	2	0	0	0	0
MOVF.W	(R8, R3, 0x50), R2	55h	55h	5634h	8118h	5555h	0000h	0000h	2	0	0	0	0

<Note>

This instruction takes 4 cycles to transfer the contents of Rs to program memory (ROM). However, no data can actually be transferred to ROM.

MOVF[.W] (Rb, --Rd, ±n), Rs

Instruction code	[0 1 1 0 1 1 1 1][d3d2d1d0 1 s2s1s0][0 b2b1b0 n11 to n8][n7 to n0] 6F08H
Argument	Rb = 3bit(Rb select), Rd = 4bit(R select), n = 12bit(signed), Rs = 3bit(R select)
Word count	2
Cycle count	3 or 4

Function	(Rd)←(Rd)-2, if Borrow : (Rb)←(Rb)-1
	if (Rd±n) = even data : [Rb<<16+Rd±n+1] ←Hibyte(Rs),
	[Rb<<16+Rd±n] ←Lobyte(Rs)
	if (Rd±n) = odd data : [Rb<<16+Rd±n] ←Hibyte(Rs),
	[Rb<<16+Rd±n-1] ←Lobyte(Rs)
	(PC)←(PC)+4

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction firstly subtracts 2 from the contents of the general-purpose register designated by Rd. Rb is decremented if a borrow occurs as the result of the subtraction performed on Rd.

Subsequently, if the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the 32-bit address, of which the higher-order 16 bits are the contents of the base register (Rb) and the lower-order 16 bits are the result of the arithmetic operation*1 performed on the contents of Rd and n, is at an even address, the instruction transfers the contents of the lower-order 8 bits of Rs to [Rb<<16+Rd±n] and the higher-order 8 bits of Rs to [Rb<<16+Rd±n+1].

In the case of an odd address, the instruction transfers the contents of the higher-order 8 bits of Rs to [Rb<<16+Rd±n] and the lower-order 8 bits of Rs to [Rb<<16+Rd±n-1].

The legitimate value range designated by Rd is from R0 to R15, that by Rb is from R8 to R13, that by Rs is from R0 to R7, and that by n is that of signed 12-bit data (-2048 to 2047).

*1: Any carry or borrow resulting from the arithmetic operation performed on the lower-order 16 bits is reflected in the higher-order 16 bits.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	R8	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50, #0x6666	66h	66h	-	-	-	-	-	-	0	0	0	0
MOV.W	R3, #0x0002	66h	66h	-	-	-	0002h	-	3	0	0	1	0
MOV.W	R0, #0x5555	66h	66h	5555h	-	-	0002h	-	0	0	0	0	0
MOVF.W	(R8, --R3, 0x50), R0	55h	55h	5555h	-	-	0000h	0000h	0	0	0	0	0
INC	R3, #1	55h	55h	5555h	-	-	0002h	0000h	3	0	0	1	0
MOV.W	R1, #0x1200	55h	55h	5555h	1200h	-	0002h	0000h	1	1	0	0	0
MOVF.W	(R8, --R3, 0x50), R1	00h	12h	5555h	1200h	-	0000h	0000h	1	1	0	0	0
INC	R3, #1	00h	12h	5555h	1200h	-	0002h	0000h	3	0	0	1	0
MOV.W	R2, #0x0000	00h	12h	5555h	1200h	0000h	0002h	0000h	2	1	1	0	0
MOVF.W	(R8, --R3, 0x50), R2	00h	00h	5555h	1200h	0000h	0000h	0000h	2	1	1	0	0
INC	R3, #1	00h	00h	5555h	1200h	0000h	0002h	0000h	3	0	0	1	0
MOV.W	R0, #0x5634	00h	00h	5634h	1200h	0000h	0002h	0000h	0	0	0	1	0
MOVF.W	(R8, --R3, 0x50), R0	34h	56h	5634h	1200h	0000h	0000h	0000h	0	0	0	1	0

<Note>

This instruction takes 4 cycles to transfer the contents of Rs to program memory (ROM). However, no data can actually be transferred to ROM.

MOVF[.W] (Rb, Rd++, ±n), Rs

Instruction code	[0 1 1 0 0 1 1 1][d3d2d1d0 1 s2s1s0][0 b2b1b0 n11 to n8][n7 to n0] 6708H
Argument	Rb = 3bit(Rb select), Rd = 4bit(R select), n = 12bit(signed), Rs = 3bit(R select)
Word count	2
Cycle count	3 or 4

Function	if (Rd±n) = even data : [Rb<<16+Rd±n+1] ←Hibyte (Rs),
	[Rb<<16+Rd±n] ←Lobyte(Rs)
	if (Rd±n) = odd data : [Rb<<16+Rd±n] ←Hibyte(Rs),
	[Rb<<16+Rd±n-1] ←Lobyte(Rs)
	(Rd)←(Rd)+2, if Carry : (Rb)←(Rb)+1
	(PC)←(PC)+4

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

If the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the 32-bit address, of which the higher-order 16 bits are the contents of the general-purpose register designated by the base register (Rb) and the lower-order 16 bits are the result of the arithmetic operation*1 performed on the contents of Rd and n, is at an even address, the instruction transfers the contents of the lower-order 8 bits of Rs to [Rb<<16+Rd±n] and the higher-order 8 bits of Rs to [Rb<<16+Rd±n+1].

In the case of an odd address, the instruction transfers the contents of the higher-order 8 bits of Rs to [Rb<<16+Rd±n] and the lower-order 8 bits of Rs to [Rb<<16+Rd±n-1]. Subsequently, the instruction adds 2 to the contents of Rd. Rb is incremented if a carry occurs as the result of the addition performed on Rd.

The legitimate value range designated by Rd is from R0 to R15, that by Rb is from R8 to R13, that by Rs is from R0 to R7, and that by n is that of signed 12-bit data (-2048 to 2047).

*1: Any carry or borrow resulting from the arithmetic operation performed on the lower-order 16 bits is reflected in the higher-order 16 bits.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	R8	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50, #0x6666	66h	66h	-	-	-	-	-	-	0	0	0	0
MOV.W	R3, #0x0000	66h	66h	-	-	-	0000h	-	3	1	1	0	0
MOV.W	R0, #0x5555	66h	66h	5555h	-	-	0000h	-	0	0	0	0	0
MOVF.W	(R8, R3++, 0x50), R0	55h	55h	5555h	-	-	0002h	0000h	0	0	0	0	0
DEC	R3, #1	55h	55h	5555h	-	-	0000h	0000h	3	1	1	0	0
MOV.W	R1, #0x1200	55h	55h	5555h	1200h	-	0000h	0000h	1	1	0	0	0
MOVF.W	(R8, R3++, 0x50), R1	00h	12h	5555h	1200h	-	0002h	0000h	1	1	0	0	0
DEC	R3, #1	00h	12h	5555h	1200h	-	0000h	0000h	3	1	1	0	0
MOV.W	R2, #0x0000	00h	12h	5555h	1200h	0000h	0000h	0000h	2	1	1	0	0
MOVF.W	(R8, R3++, 0x50), R2	00h	00h	5555h	1200h	0000h	0002h	0000h	2	1	1	0	0
DEC	R3, #1	00h	00h	5555h	1200h	0000h	0000h	0000h	3	1	1	0	0
MOV.W	R0, #0x5634	00h	00h	5634h	1200h	0000h	0000h	0000h	0	0	0	1	0
MOVF.W	(R8, R3++, 0x50), R0	34h	56h	5634h	1200h	0000h	0002h	0000h	0	0	0	1	0

<Note>

This instruction takes 4 cycles to transfer the contents of Rs to program memory (ROM). However, no data can actually be transferred to ROM.

MOVF[.W] (Rd), Rs

Instruction code	[0 1 1 1 0 1 1 1][d3d2d1d0 0 s2s1s0] 7700H
Argument	Rd = 4bit(R select), Rs = 3bit(R select)
Word count	1
Cycle count	2 or 3

Function	if (Rd) = even data : [R8<<16+Rd+1] ←Hibyte(Rs),
	[R8<<16+Rd] ←Lobyte(Rs),
	if (Rd) = odd data : [R8<<16+Rd] ←Hibyte(Rs),
	[R8<<16+Rd-1] ←Lobyte(Rs)
	(PC)←(PC)+2

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

If the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the 32-bit address, of which the higher-order 16 bits are the contents of R8 (Rb0) and the lower-order 16 bits are the contents of the general-purpose register designated by Rd, is at an even address, the instruction transfers the contents of the lower-order 8 bits of the general-purpose register Rs to [R8<<16+Rd] and the contents of the higher-order 8 bits of Rs to [R8<<16+Rd+1]. In the case of an odd address, the instruction transfers the contents of the higher-order 8 bits to [R8<<16+Rd] and the contents of the lower-order 8 bits of Rs to [R8<<16+Rd-1].

The legitimate value range designated by Rd is from R0 to R15 and that by Rs is from R0 to R7.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50, #0x6666	66h	66h	-	-	-	-	-	0	0	0	0
MOV.W	R3, #0x0050	66h	66h	-	-	-	0050h	3	0	0	0	0
MOV.W	R0, #0x5555	66h	66h	5555h	-	-	0050h	0	0	0	0	0
MOVF.W	(R3), R0	55h	55h	5555h	-	-	0050h	0	0	0	0	0
MOV.W	R1, #0x1200	55h	55h	5555h	1200h	-	0050h	1	1	0	0	0
MOVF.W	(R3), R1	00h	12h	5555h	1200h	-	0050h	1	1	0	0	0
MOV.W	R2, #0x0000	00h	12h	5555h	1200h	0000h	0050h	2	1	1	0	0
MOVF.W	(R3), R2	00h	00h	5555h	1200h	0000h	0050h	2	1	1	0	0
MOV.W	R0, #0x5634	00h	00h	5634h	1200h	0000h	0050h	0	0	0	1	0
MOVF.W	(R3), R0	34h	56h	5634h	1200h	0000h	0050h	0	0	0	1	0
MOV.W	R1, #0x8118	34h	56h	5634h	8118h	0000h	0050h	1	0	0	0	1
MOVF.W	(R3), R1	34h	56h	5555h	1200h	0000h	0050h	1	0	0	0	1
MOV.W	R2, #0x5555	34h	56h	5634h	1200h	0000h	0050h	2	0	0	0	0
MOVF.W	(R3), R2	34h	56h	5634h	1200h	0000h	0050h	2	0	0	0	0

<Note>

In this case, Rb0 refers to R8.

This instruction takes 3 cycles to transfer the contents of Rs to program memory (ROM). However, no data can actually be transferred to ROM.

MOVF[.W] (--Rd), Rs

Instruction code	[0 1 1 0 1 1 1 1][d3d2d1d0 0 s2s1s0] 6F00H
Argument	Rd = 4bit(R select), Rs = 3bit(R select)
Word count	1
Cycle count	2 or 3

Function	(Rd)←(Rd)-2, if Borrow : (R8)←(R8)-1
	if (Rd) = even data : [R8<<16+Rd+1] ←Hibyte(Rs),
	[R8<<16+Rd] ←Lobyte(Rs)
	if (Rd) = odd data : [R8<<16+Rd] ←Hibyte(Rs),
	[R8<<16+Rd-1] ←Lobyte(Rs)
	(PC)←(PC)+2

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction firstly subtracts 2 from the contents of the general-purpose register designated by Rd. R8 is decremented if a borrow occurs as the result of the subtraction performed on Rd.

Subsequently, if the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by designated by the 32-bit address, of which the higher-order 16 bits are the contents of R8 (Rb0) and the lower-order 16 bits are the contents of the general-purpose register Rd, is at an even address, the instruction transfers the contents of the lower-order 8 bits of the general-purpose register Rs to [R8<<16+Rd] and the contents of the higher-order 8 bits of Rs to [R8<<16+Rd+1]. In the case of an odd address, the instruction transfers the contents of the higher-order 8 bits to [R8<<16+Rd] and the contents of the lower-order 8 bits of Rs to [R8<<16+Rd-1].

The legitimate value range designated by Rd is from R0 to R15 and that by Rs is from R0 to R7.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50, #0x6666	66h	66h	-	-	-	-	-	0	0	0	0
MOV.W	R3, #0x0052	66h	66h	-	-	-	0052h	3	0	0	1	0
MOV.W	R0, #0x5555	66h	66h	5555h	-	-	0052h	0	0	0	0	0
MOVF.W	(--R3), R0	55h	55h	5555h	-	-	0050h	0	0	0	0	0
INC	R3, #1	55h	55h	5555h	-	-	0052h	3	0	0	1	0
MOV.W	R1, #0x1200	55h	55h	5555h	1200h	-	0052h	1	1	0	0	0
MOVF.W	(--R3), R1	00h	12h	5555h	1200h	-	0050h	1	1	0	0	0
INC	R3, #1	00h	12h	5555h	1200h	-	0052h	3	0	0	1	0
MOV.W	R2, #0x0000	00h	12h	5555h	1200h	0000h	0052h	2	1	1	0	0
MOVF.W	(--R3), R2	00h	00h	5555h	1200h	0000h	0050h	2	1	1	0	0
INC	R3, #1	00h	00h	5555h	1200h	0000h	0052h	3	0	0	1	0
MOV.W	R0, #0x5634	00h	00h	5634h	1200h	0000h	0052h	0	0	0	1	0
MOVF.W	(--R3), R0	34h	56h	5634h	1200h	0000h	0050h	0	0	0	1	0

<Note>

In this case, Rb0 refers to R8.

This instruction takes 3 cycles to transfer the contents of Rs to program memory (ROM). However, no data can actually be transferred to ROM.

MOVF[.W] (Rd++), Rs

Instruction code	[0 1 1 0 0 1 1 1][d3d2d1d0 0 s2s1s0] 6700H
Argument	Rd = 4bit(R select), Rs = 3bit(R select)
Word count	1
Cycle count	2 or 3

Function	if (Rd) = even data : [R8<<16+Rd+1] ←Hibyte(Rs),
	[R8<<16+Rd] ←Lobyte(Rs)
	if (Rd) = odd data : [R8<<16+Rd] ←Hibyte(Rs),
	[R8<<16+Rd-1] ←Lobyte(Rs)
	(Rd)←(Rd)+2, if Carry : (R8)←(R8)+1
	(PC)←(PC)+2

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

If the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the 32-bit address, of which the higher-order 16 bits are the contents of R8 (Rb0) and the lower-order 16 bits are the contents of the general-purpose register designated by Rd, is at an even address, the instruction transfers the contents of the lower-order 8 bits of the general-purpose register Rs to [R8<<16+Rd] and the contents of the higher-order 8 bits of Rs to [R8<<16+Rd+1]. In the case of an odd address, the instruction transfers the contents of the higher-order 8 bits of Rs to [R8<<16+Rd] and the contents of the lower-order 8 bits of Rs to [R8<<16+Rd-1]. Subsequently, the instruction adds 2 to the contents of Rd. R8 is incremented if a carry occurs as the result of the addition performed on Rd.
The legitimate value range designated by Rd is from R0 to R15 and that by Rs is from R0 to R7.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50, #0x6666	66h	66h	-	-	-	-	-	0	0	0	0
MOV.W	R3, #0x0050	66h	66h	-	-	-	0050h	3	0	0	0	0
MOV.W	R0, #0x5555	66h	66h	5555h	-	-	0050h	0	0	0	0	0
MOVF.W	(R3++), R0	55h	55h	5555h	-	-	0052h	0	0	0	0	0
DEC	R3, #1	55h	55h	5555h	-	-	0050h	3	0	0	0	0
MOV.W	R1, #0x1200	55h	55h	5555h	1200h	-	0050h	1	1	0	0	0
MOVF.W	(R3++), R1	00h	12h	5555h	1200h	-	0052h	1	1	0	0	0
DEC	R3, #1	00h	12h	5555h	1200h	-	0050h	3	0	0	0	0
MOV.W	R2, #0x0000	00h	12h	5555h	1200h	0000h	0050h	2	1	1	0	0
MOVF.W	(R3++), R2	00h	00h	5555h	1200h	0000h	0052h	2	1	1	0	0
DEC	R3, #1	00h	00h	5555h	1200h	0000h	0050h	3	0	0	0	0
MOV.W	R0, #0x5634	00h	00h	5634h	1200h	0000h	0050h	0	0	0	1	0
MOVF.W	(R3++), R0	34h	56h	5634h	1200h	0000h	0052h	0	0	0	1	0

<Note>

In this case, Rb0 refers to R8.

This instruction takes 3 cycles to transfer the contents of Rs to program memory (ROM). However, no data can actually be transferred to ROM.

MOVF[.W] Rd, (Rb, Rs, ±n)

Instruction code	[0 1 1 1 0 1 0 1][s3s2s1s0 1 d2d1d0][0 b2b1b0 n11 to n8][n7 to n0] 7508H
Argument	Rd = 3bit(R select), Rb = 3bit(Rb select), Rs = 4bit(R select), n = 12bit(signed)
Word count	2
Cycle count	3 or 4

Function	if (Rs±n)=even data:
	Hibyte(Rd)←[Rb<<16+Rs±n+1], Lobyte(Rd)←[Rb<<16+Rs±n]
	if (Rs±n)=odd data:
	Hibyte(Rd)←[Rb<<16+Rs±n], Lobyte(Rd)←[Rb<<16+Rs±n-1]
	(PC)←(PC)+4

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

If the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the 32-bit address, of which the higher-order 16 bits are the contents of the general-purpose register designated by the base register (Rb) and the lower-order 16 bits are the result of the arithmetic operation*1 performed on the contents of Rs and n, is at an even address, the instruction transfers the contents to the lower-order 8-bit positions of the general-purpose register Rd and the contents of [Rb<<16+Rs±n+1] to the higher-order 8-bit positions of Rd. In the case of an odd address, the instruction transfers the contents of [Rb<<16+Rs±n] to the higher-order 8-bit positions of Rd and the contents of [Rb<<16+Rs±n-1] to the lower-order 8-bit positions of Rd.

The legitimate value range designated by Rd is from R0 to R7, that by Rb is from R8 to R13, that by Rs is from R0 to R15, and that by n is that of signed 12-bit data (-2048 to 2047).

*1: Any carry or borrow resulting from the arithmetic operation performed on the lower-order 16 bits is reflected in the higher-order 16 bits.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	R8	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0000	-	-	-	-	-	0000h	-	3	1	1	0	0
MOV.W	0x50, #0x5555	55h	55h	-	-	-	0000h	-	3	0	0	0	0
MOVF.W	R0, (R8, R3, 0x50)	55h	55h	5555h	-	-	0000h	0000h	0	0	0	0	0
MOV.W	0x50, #0x1200	00h	12h	5555h	-	-	0000h	0000h	0	1	0	0	0
MOVF.W	R1, (R8, R3, 0x50)	00h	12h	5555h	1200h	-	0000h	0000h	1	1	0	0	0
MOV.W	0x50, #0x0000	00h	00h	5555h	1200h	-	0000h	0000h	1	1	1	0	0
MOVF.W	R2, (R8, R3, 0x50)	00h	00h	5555h	1200h	0000h	0000h	0000h	2	1	1	0	0
MOV.W	0x50, #0x5634	34h	56h	5555h	1200h	0000h	0000h	0000h	2	0	0	1	0
MOVF.W	R0, (R8, R3, 0x50)	34h	56h	5634h	1200h	0000h	0000h	0000h	0	0	0	1	0
MOV.W	0x50, #0x8118	18h	81h	5634h	1200h	0000h	0000h	0000h	0	0	0	0	1
MOVF.W	R1, (R8, R3, 0x50)	18h	81h	5634h	8118h	0000h	0000h	0000h	1	0	0	0	1
MOV.W	0x50, #0x5555	55h	55h	5634h	8118h	0000h	0000h	0000h	1	0	0	0	0
MOVF.W	R2, (R8, R3, 0x50)	55h	55h	5634h	8118h	5555h	0000h	0000h	2	0	0	0	0

<Note>

This instruction takes 4 cycles to transfer the contents of program memory (ROM) to Rd.

MOVF[.W] Rd, (Rb, --Rs, ±n)

Instruction code	[0 1 1 0 1 1 0 1][s3s2s1s0 1 d2d1d0][0 b2b1b0 n11 to n8][n7 to n0] 6D08H
Argument	Rd = 3bit(R select), Rb = 3bit(Rb select), Rs = 4bit(R select), n = 12bit(signed)
Word count	2
Cycle count	3 or 4

Function	(Rs)←(Rs)-2, if Borrow : (Rb)←(Rb)-1
	If (Rs±n) =even data:
	Hibyte(Rd)←[Rb<<16+Rs±n+1], Lobyte(Rd)←[Rb<<16+Rs±n]
	If (Rs±n) =odd data:
	Hibyte(Rd)←[Rb<<16+Rs±n], Lobyte(Rd)←[Rb<<16+Rs±n-1]
	(PC)←(PC)+4

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction firstly subtracts 2 from the contents of the general-purpose register designated by Rs. Rb is decremented if a borrow occurs as the result of the subtraction performed on Rs.

Subsequently, if the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the 32-bit address, of which the higher-order 16 bits are the contents of the base register (Rb) and the lower-order 16 bits are the result of the arithmetic operation*1 performed on the contents of Rs and n, is at an even address, the instruction transfers the contents to the lower-order 8-bit positions of the general-purpose register Rd and the contents of [Rb<<16+Rs±n+1] to the higher-order 8-bit positions of Rd. In the case of an odd address, the instruction transfers the contents of [Rb<<16+Rs±n] to the higher-order 8-bit positions of Rd and the contents of [Rb<<16+Rs±n-1] to the lower-order 8-bit positions of Rd.

The legitimate value range designated by Rd is from R0 to R7, that by Rb is from R8 to R13, that by Rs is from R0 to R15, and that by n is that of signed 12-bit data (-2048 to 2047).

*1: Any carry or borrow resulting from the arithmetic operation performed on the lower-order 16 bits is reflected in the higher-order 16 bits.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	R8	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0002	-	-	-	-	-	0002h	-	3	0	0	1	0
MOV.W	0x50, #0x5555	55h	55h	-	-	-	0002h	-	3	0	0	0	0
MOVF.W	R0,(R8,--R3,0x50)	55h	55h	5555h	-	-	0000h	0000h	0	0	0	0	0
INC	R3, #1	55h	55h	5555h	-	-	0002h	0000h	3	0	0	1	0
MOV.W	0x50, #0x1200	00h	12h	5555h	-	-	0002h	0000h	3	1	0	0	0
MOVF.W	R1,(R8,--R3,0x50)	00h	12h	5555h	1200h	-	0000h	0000h	1	1	0	0	0
INC	R3, #1	00h	12h	5555h	1200h	-	0002h	0000h	3	0	0	1	0
MOV.W	0x50, #0x0000	00h	00h	5555h	1200h	-	0002h	0000h	3	1	1	0	0
MOVF.W	R2,(R8,--R3,0x50)	00h	00h	5555h	1200h	0000h	0000h	0000h	2	1	1	0	0
INC	R3, #1	00h	00h	5555h	1200h	0000h	0002h	0000h	3	0	0	1	0
MOV.W	0x50, #0x5634	34h	56h	5555h	1200h	0000h	0002h	0000h	3	0	0	1	0
MOVF.W	R0,(R8,--R3,0x50)	34h	56h	5634h	1200h	0000h	0000h	0000h	0	0	0	1	0
INC	R3, #1	34h	56h	5634h	1200h	0000h	0002h	0000h	3	0	0	1	0
MOV.W	0x50, #0x8118	18h	81h	5634h	1200h	0000h	0002h	0000h	3	0	0	0	1
MOVF.W	R1,(R8,--R3,0x50)	18h	81h	5634h	8118h	0000h	0000h	0000h	1	0	0	0	1

<Note>

This instruction takes 4 cycles to transfer the contents of program memory (ROM) to Rd.

MOVF[.W] Rd, (Rb, Rs++, ±n)

Instruction code	[0 1 1 0 0 1 0 1][s3s2s1s0 1 d2d1d0][0 b2b1b0 n11 to n8][n7 to n0] 6508H
Argument	Rd = 3bit(R select), Rb = 3bit(Rb select), Rs = 4bit(R select), n = 12bit(signed)
Word count	2
Cycle count	3 or 4

Function	if (Rs±n) =even data:
	Hibyte(Rd)←[Rb<<16+Rs±n+1], Lobyte(Rd)←[Rb<<16+Rs±n]
	if (Rs±n) =odd data:
	Hibyte(Rd)←[Rb<<16+Rs±n], Lobyte(Rd)←[Rb<<16+Rs±n-1]
	(Rs)←(Rs)+1, if Carry : (Rb)←(Rb)+1
	(PC)←(PC)+4

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

If the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the 32-bit address, of which the higher-order 16 bits are the contents of the general-purpose register designated by the base register (Rb) and the lower-order 16 bits are the result of the arithmetic operation*1 performed on the contents of the general-purpose register Rs and n, is at an even address, the instruction transfers the contents to the lower-order 8-bit positions of the general-purpose register Rd and the contents of [Rb<<16+Rs±n+1] to the higher-order 8-bit positions of Rd. In the case of an odd address, the instruction transfers the contents of [Rb<<16+Rs ± n] to the higher-order 8-bit positions of the general-purpose register Rd and the contents of [Rb<<16+Rs±n-1] to the lower-order 8-bit positions of Rd. Subsequently, the instruction adds 2 to the contents of Rs. Rb is incremented if a carry occurs as the result of the addition performed on Rs.

The legitimate value range designated by Rd is from R0 to R7, that by Rb is from R8 to R13, that by Rs is from R0 to R15, and that by n is that of signed 12-bit data (-2048 to 2047).

*1: Any carry or borrow resulting from the arithmetic operation performed on the lower-order 16 bits is reflected in the higher-order 16 bits.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	R8	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0000	-	-	-	-	-	0000h	-	3	1	1	0	0
MOV.W	0x50, #0x5555	55h	55h	-	-	-	0000h	-	3	0	0	0	0
MOVF.W	R0,(R8,R3++, 0x50)	55h	55h	5555h	-	-	0002h	0000h	0	0	0	0	0
DEC	R3, #1	55h	55h	5555h	-	-	0000h	0000h	3	1	1	0	0
MOV.W	0x50, #0x1200	00h	12h	5555h	-	-	0000h	0000h	3	1	0	0	0
MOVF.W	R1,(R8,R3++, 0x50)	00h	12h	5555h	1200h	-	0002h	0000h	1	1	0	0	0
DEC	R3, #1	00h	12h	5555h	1200h	-	0000h	0000h	3	1	1	0	0
MOV.W	0x50, #0x0000	00h	00h	5555h	1200h	-	0000h	0000h	3	1	1	0	0
MOVF.W	R2,(R8,R3++, 0x50)	00h	00h	5555h	1200h	0000h	0002h	0000h	2	1	1	0	0
DEC	R3, #1	00h	00h	5555h	1200h	0000h	0000h	0000h	3	1	1	0	0
MOV.W	0x50, #0x5634	34h	56h	5555h	1200h	0000h	0000h	0000h	3	0	0	1	0
MOVF.W	R0,(R8,R3++, 0x50)	34h	56h	5634h	1200h	0000h	0002h	0000h	0	0	0	1	0
DEC	R3, #1	34h	56h	5634h	1200h	0000h	0000h	0000h	3	1	1	0	0
MOV.W	0x50, #0x8118	18h	81h	5634h	1200h	0000h	0000h	0000h	3	0	0	0	1
MOVF.W	R1,(R8,R3++, 0x50)	18h	81h	5634h	8118h	0000h	0002h	0000h	1	0	0	0	1

<Note>

This instruction takes 4 cycles to transfer the contents of program memory (ROM) to Rd.

MOVF[.W] Rd, (Rs)

Instruction code	[0 1 1 1 0 1 0 1][s3s2s1s0 0 d2d1d0] 7500H
Argument	Rd = 3bit(R select), Rs = 4bit(R select)
Word count	1
Cycle count	2 or 3

Function	if (Rs) = even data :
	Hibyte(Rd)←[R8<<16+Rs+1], Lobyte(Rd)←[R8<<16+Rs]
	if (Rs) = odd data :
	Hibyte(Rd)←[R8<<16+Rs], Lobyte(Rd)←[R8<<16+Rs-1]
	(PC)←(PC)+2

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

If the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the 32-bit address, of which the higher-order 16 bits are the contents of R8 (Rb0) and the lower-order 16 bits are the contents of the general-purpose register designated by Rs, is at an even address, the instruction transfers the contents to the lower-order 8-bit positions of the general-purpose register Rd and the contents of [R8<<16+Rs+1] to the higher-order 8-bit positions of Rd. In the case of an odd address, the instruction transfers the contents of [R8<<16+Rs] to the higher-order 8-bit positions of Rd and the contents of [R8<<16+Rs-1] to the lower-order 8-bit positions of Rd.
The legitimate value range designated by Rd is from R0 to R7 and that by Rs is from R0 to R15.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0050	-	-	-	-	-	0050h	3	0	0	0	0
MOV.W	0x50, #0x5555	55h	55h	-	-	-	0050h	3	0	0	0	0
MOVF.W	R0, (R3)	55h	55h	5555h	-	-	0050h	0	0	0	0	0
MOV.W	0x50, #0x1200	12h	00h	5555h	-	-	0050h	0	1	0	0	0
MOVF.W	R1, (R3)	12h	00h	5555h	1200h	-	0050h	1	1	0	0	0
MOV.W	0x50, #0x0000	00h	00h	5555h	1200h	-	0050h	1	1	1	0	0
MOVF.W	R2, (R3)	00h	00h	5555h	1200h	0000h	0050h	2	1	1	0	0
MOV.W	0x50, #0x5634	56h	34h	5555h	1200h	0000h	0050h	2	0	0	1	0
MOVF.W	R0, (R3)	56h	34h	5634h	1200h	0000h	0050h	0	0	0	1	0
MOV.W	0x50, #0x8118	81h	18h	5634h	1200h	0000h	0050h	0	0	0	0	1
MOVF.W	R1, (R3)	81h	18h	5634h	8118h	0000h	0050h	1	0	0	0	1
MOV.W	0x50, #0x5555	55h	55h	5634h	8118h	0000h	0050h	1	0	0	0	0
MOVF.W	R2, (R3)	55h	55h	5634h	8118h	5555h	0050h	2	0	0	0	0

<Note>

In this case, Rb0 refers to R8.

The instruction takes 3 cycles to transfer the contents of program memory (ROM) to Rd.

MOVF[.W] Rd, (--Rs)

Instruction code	[0 1 1 0 1 1 0 1][s3s2s1s0 0 d2d1d0] 6D00H
Argument	Rd = 3bit(R select), Rs = 4bit(R select)
Word count	1
Cycle count	2 or 3

Function	(Rs)←(Rs)-2, if Borrow : (R8)←(R8)-1
	if (Rs) = even data :
	Hibyte(Rd)←[R8<<16+Rs+1], Lobyte(Rd)←[R8<<16+Rs]
	if (Rs) = odd data :
	Hibyte(Rd)←[R8<<16+Rs], Lobyte(Rd)←[R8<<16+Rs-1]
	(PC)←(PC)+2

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

This instruction firstly subtracts 2 from the contents of the general-purpose register designated by Rs. R8 is decremented if a borrow occurs as the result of the subtraction performed on Rs.

Subsequently, if the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the 32-bit address, of which the higher-order 16 bits are the contents of R8 (Rb0) and the lower-order 16 bits are the contents of the general-purpose register Rs, is at an even address, the instruction transfers the contents to the lower-order 8-bit positions of the general-purpose register Rd and the contents of [R8<<16+Rs+1] to the higher-order 8-bit positions of Rd. In the case of an odd address, the instruction transfers the contents of [R8<<16+Rs] to the higher-order 8-bit positions of Rd and the contents of [R8<<16+Rs-1] to the lower-order 8-bit positions of Rd.

The legitimate value range designated by Rd is from R0 to R7 and that by Rs is from R0 to R15.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0052	-	-	-	-	-	0052h	3	0	0	1	0
MOV.W	0x50, #0x5555	55h	55h	-	-	-	0052h	3	0	0	0	0
MOVF.W	R0, (--R3)	55h	55h	5555h	-	-	0050h	0	0	0	0	0
INC	R3, #1	55h	55h	5555h	-	-	0052h	3	0	0	1	0
MOV.W	0x50, #0x1200	12h	00h	5555h	-	-	0052h	3	1	0	0	0
MOVF.W	R1, (--R3)	12h	00h	5555h	1200h	-	0050h	1	1	0	0	0
INC	R3, #1	12h	00h	5555h	1200h	-	0052h	3	0	0	1	0
MOV.W	0x50, #0x0000	00h	00h	5555h	1200h	-	0052h	3	1	1	0	0
MOVF.W	R2, (--R3)	00h	00h	5555h	1200h	0000h	0050h	2	1	1	0	0
INC	R3, #1	00h	00h	5555h	1200h	0000h	0052h	3	0	0	1	0
MOV.W	0x50, #0x5634	56h	34h	5555h	1200h	0000h	0052h	3	0	0	1	0
MOVF.W	R0, (--R3)	56h	34h	5634h	1200h	0000h	0050h	0	0	0	1	0
INC	R3, #1	56h	34h	5634h	1200h	0000h	0052h	3	0	0	1	0
MOV.W	0x50, #0x8118	81h	18h	5634h	1200h	0000h	0052h	3	0	0	0	1
MOVF.W	R1, (--R3)	81h	18h	5634h	8118h	0000h	0050h	1	0	0	0	1

<Note>

In this case, Rb0 refers to R8.

The instruction takes 3 cycles to transfer the contents of program memory (ROM) to Rd.

MOVF[.W] Rd, (Rs++)

Instruction code	[0 1 1 0 0 1 0 1][s3s2s1s0 0 d2d1d0] 6500H
Argument	Rd = 3bit(R select), Rs = 4bit(R select)
Word count	1
Cycle count	2 or 3

Function	if (Rs) = even data :
	Hibyte(Rd)←[R8<<16+Rs+1], Lobyte(Rd)←[R8<<16+Rs]
	if (Rs) = odd data :
	Hibyte (Rd)←[R8<<16+Rs], Lobyte(Rd)←[R8<<16+Rs-1]
	(Rs)←(Rs)+2, if Carry : (R8)←(R8)+1
	(PC)←(PC)+2

Affected flags	Z8, Z16, P, S, N0 to N3

[Description]

If the data memory (RAM) location, special function register (SFR), or program memory (ROM) location designated by the 32-bit address, of which the higher-order 16 bits are the contents of R8 (Rb0) and the lower-order 16 bits are the contents of the general-purpose register designated by Rs, is at an even address, the instruction transfers the contents to the lower-order 8-bit positions of the general-purpose register Rd and the contents of [R8<<16+Rs+1] to the higher-order 8-bit positions of Rd. In the case of an odd address, the instruction transfers the contents of [R8<<16+Rs] to the higher-order 8-bit positions of Rd and the contents of [R8<<16+Rs-1] to the lower-order 8-bit positions of Rd.

Subsequently, the instruction adds 2 to the contents of the general-purpose register Rs. R8 is incremented if a carry occurs as the result of the addition performed on Rs.

The legitimate value range designated by Rd is from R0 to R7 and that by Rs is from R0 to R15.

[Example]

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	R3, #0x0050	-	-	-	-	-	0050h	3	0	0	0	0
MOV.W	0x50, #0x5555	55h	55h	-	-	-	0050h	3	0	0	0	0
MOVF.W	R0, (R3++)	55h	55h	5555h	-	-	0052h	0	0	0	0	0
DEC	R3, #1	55h	55h	5555h	-	-	0050h	3	0	0	0	0
MOV.W	0x50, #0x1200	12h	00h	5555h	-	-	0050h	3	1	0	0	0
MOVF.W	R1, (R3++)	12h	00h	5555h	1200h	-	0052h	1	1	0	0	0
DEC	R3, #1	12h	00h	5555h	1200h	-	0050h	3	0	0	0	0
MOV.W	0x50, #0x0000	00h	00h	5555h	1200h	-	0050h	3	1	1	0	0
MOVF.W	R2, (R3++)	00h	00h	5555h	1200h	0000h	0052h	2	1	1	0	0
DEC	R3, #1	00h	00h	5555h	1200h	0000h	0050h	3	0	0	0	0
MOV.W	0x50, #0x5634	56h	34h	5555h	1200h	0000h	0050h	3	0	0	1	0
MOVF.W	R0, (R3++)	56h	34h	5634h	1200h	0000h	0052h	0	0	0	1	0
DEC	R3, #1	56h	34h	5634h	1200h	0000h	0050h	3	0	0	0	0
MOV.W	0x50, #0x8118	81h	18h	5634h	1200h	0000h	0050h	3	0	0	0	1
MOVF.W	R1, (R3++)	81h	18h	5634h	8118h	0000h	0052h	1	0	0	0	1

<Note>

In this case, Rb0 refers to R8.

The instruction takes 3 cycles to transfer the contents of program memory (ROM) to Rd.

MUL

Instruction code	[0 0 0 0 0 0 0 0][1 1 0 1 0 0 0 0] 00D0H
Argument	
Word count	1
Cycle count	4 or 18 cycles
Function	(R0)× (R2)=Result(32bit), R1= Result >>16, R0= Result &FFFFh, (PC)←(PC) +2
Affected flags	Z8, Z16, P, S CY, HC, OV, and N3-N0 all cleared.

[Description]

This instruction places the higher-order 16 bits of the result of multiplications performed on the contents of the general-purpose registers R0 and R2 in R1 and the lower-order 16 bits of the result in R0.

[Example]

	R0	R1	R2	R3	PSW
	-	-	-	-	-
MOV.W R0,#0X48D0	48D0h	-	-	-	0020h
MOV.W R1,#0X5678	48D0h	5678h	-	-	1000h
MOV.W R2,#0X4000	48D0h	5678h	4000h	-	2021h
MOV.W R3,#0XDEF0	48D0h	5678h	4000h	DEF0h	3040h
MUL	0000h	1234h	4000h	DEF0h	0003h

<Note>

The flags (Z8, Z16, P, and S) are affected by R0 (lower-order 16-bit result).

Please refer to the datasheet of each product for the cycle count.

NOP

Instruction code	[0 0 0 0 0 0 0 0][0 0 0 0 0 0 0 0]	0000H
Argument		
Word count	1	
Cycle count	1	
Function	(PC)←(PC)+2	
Affected flags		

[Description]

This instruction consumes one system clock and does nothing.

NOT Rd

Instruction code	[0 0 1 1 0 0 0 0][1 0 1 1 d3d2d1d0] 30B0H
Argument	Rd = 4bit(R select)
Word count	1
Cycle count	1
Function	(Rd)← ~(Rd), (PC)←(PC)+2
Affected flags	Z8,Z16,P,S,N0 to N3

[Description]

This instruction inverts the contents of the general-purpose register designated by Rd.
The legitimate value range designated by Rd is from R0 to R15.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-
MOV.W	R0,#0X5678	5678h	-	-	-	0	0	0	0	0
MOV.W	R1,#0X0000	5678h	0000h	-	-	1	1	1	0	0
MOV.W	R2,#0XFFFF	5678h	0000h	FFFFh	-	2	0	0	0	1
MOV.W	R3,#0X3456	5678h	0000h	FFFFh	3456h	3	0	0	1	0
NOT	R0	A987h	0000h	FFFFh	3456h	0	0	0	0	1
NOT	R1	A987h	FFFFh	FFFFh	3456h	1	0	0	0	1
NOT	R2	A987h	FFFFh	0000h	3456h	2	1	1	0	0
NOT	R3	A987h	FFFFh	0000h	CBA9h	3	0	0	1	1

OR Rd, Rs

Instruction code	[0 1 0 0 0 0 1 0][s3s2s1s 0 d3d2d1d0] 4200H
Argument	Rd = 4bit(R select),Rs = 4bit(R select)
Word count	1
Cycle count	1
Function	(Rd)←(Rd) | (Rs), (PC)←(PC)+2
Affected flags	Z8,Z16,P,S,N0 to N3

[Description]

This instruction takes the OR of the contents of the general-purpose registers designated by Rd and designated by Rs and places the result in Rd.
The legitimate value range designated by Rd is from R0 to R15 and that by Rs is from R0 to R15.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-
MOV.W	R0,#0X5678	5678h	-	-	-	0	0	0	0	0
MOV.W	R1,#0X0000	5678h	0000h	-	-	1	1	1	0	0
MOV.W	R2,#0XFEDC	5678h	0000h	FEDCh	-	2	0	0	0	1
MOV.W	R3,#0X3456	5678h	0000h	FEDCh	3456h	3	0	0	1	0
OR	R0,R1	5678h	0000h	FEDCh	3456h	0	0	0	0	0
OR	R1,R2	5678h	FEDCh	FEDCh	3456h	1	0	0	0	1
OR	R2,R3	5678h	FEDCh	FEDEh	3456h	2	0	0	1	1
OR	R3,R0	5678h	FEDCh	FEDEh	767Eh	3	0	0	1	0

OR Rd, #imm16

Instruction code	[0 0 1 1 0 0 0 1][0 0 0 1 d3d2d1d0][i15 to i8][i7 to i0] 3110H
Argument	Rd = 4bit(R select),imm16 = 16bit(immediate data)
Word count	2
Cycle count	2
Function	(Rd)← (Rd) | #imm16, (PC)←(PC)+4
Affected flags	Z8,Z16,P,S,N0 to N3

[Description]

This instruction takes the OR of the contents of the general-purpose register designated by Rd and immediate data designated by imm16 and places the result in Rd.
The legitimate value range designated by Rd is from R0 to R15 and that by imm16 is from 0 to FF.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-
MOV.W	R0,#0X5678	5678h	-	-	-	0	0	0	0	0
MOV.W	R1,#0X0000	5678h	0000h	-	-	1	1	1	0	0
MOV.W	R2,#0XFEDC	5678h	0000h	FEDCh	-	2	0	0	0	1
MOV.W	R3,#0X3456	5678h	0000h	FEDCh	3456h	3	0	0	1	0
OR	R0,#0X3456	767Eh	0000h	FEDCh	3456h	0	0	0	1	0
OR	R1,#0X0066	767Eh	0066h	FEDCh	3456h	1	0	0	0	0
OR	R2,#0X0123	767Eh	0066h	FFFFh	3456h	2	0	0	0	1
OR	R3,#0X7F00	767Eh	0066h	FFFFh	7F56h	3	0	0	1	0

OR Rx, #imm8

Instruction code	[0 1 0 0 0 0 1 1][i7i6i5i4i3i2i1i0] 4300H
Argument	imm8 = 8bit(immediate data)
Word count	1
Cycle count	1
Function	(Rx)← (Rx) | 16bit data(Hibyte = 00H, Lobyte = #imm8), (PC)←(PC)+2
Affected flags	Z8,Z16,P,S

[Description]

This instruction takes the OR of the contents of the general-purpose register Rx designated indirectly by the value of bits 12 to 15 (N0 to N3) of the PSW and the 16-bit data, of which the higher-order 8 bits are 00h and the lower-order 8 bits are immediate data designated by imm8, and places the result in Rx.
The legitimate value range designated by imm8 is from 0 to FF.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-
MOV.W	R3,#0X3456	-	-	-	3456h	3	0	0	1	0
MOV.W	R2,#0XFEDC	-	-	FEDCh	3456h	2	0	0	0	1
MOV.W	R1,#0X0000	-	0000h	FEDCh	3456h	1	1	1	0	0
MOV.W	R0,#0X5678	5678h	0000h	FEDCh	3456h	0	0	0	0	0
OR	Rx,#0X78	5678h	0000h	FEDCh	3456h	0	0	0	0	0
INC	R1	5678h	0001h	FEDCh	3456h	1	0	0	1	0
OR	Rx,#0X66	5678h	0067h	FEDCh	3456h	1	0	0	1	0
SWPB	R2	5678h	0067h	DCFEh	3456h	2	0	0	0	1
OR	Rx,#0X01	5678h	0067h	DCFFh	3456h	2	0	0	1	1
DEC	R3	5678h	0067h	DCFFh	3455h	3	0	0	1	0
OR	Rx,#0XAA	5678h	0067h	DCFFh	34FFh	3	0	0	1	0

POP PSW

Instruction code	[0 0 0 0 0 0 0 0][1 0 0 1 1 1 1 0] 009EH
Argument	
Word count	1
Cycle count	1
Function	(SP)←(SP)-2 Hibyte(PSW)←[SP+1], Lobyte(PSW)← [SP], (PC)←(PC)+2
Affected flags	

[Description]

This instruction decrements the stack pointer (SP) by 2 and transfers the contents of the data memory (RAM) location designated by SP to the program status word (PSW).

[Example]

		RAM (50h)	RAM (51h)	RAM (52h)	RAM (53h)	R0	R1	R2	R3	PSW	SP
		-	-	-	-	-	-	-	-	-	-
MOV.W	R15,#0X0050	-	-	-	-	-	-	-	-	F000h	0050h
MOV.W	R0,#0X5555	-	-	-	-	5555h	-	-	-	0000h	0050h
PUSH	R0	55h	55h	-	-	5555h	-	-	-	0000h	0052h
MOV.W	R1,#0X0000	55h	55h	-	-	5555h	0000h	-	-	1003h	0052h
PUSH	R1	55h	55h	00h	00h	5555h	0000h	-	-	1003h	0054h
POP	PSW	55h	55h	00h	00h	5555h	0000h	-	-	0000h	0052h
POP	PSW	55h	55h	00h	00h	5555h	0000h	-	-	5555h	0050h
MOV.W	R2,#0X1200	55h	55h	00h	00h	5555h	0000h	1200h	-	2515h	0050h
PUSH	R2	00h	12h	00h	00h	5555h	0000h	1200h	-	2515h	0052h
MOV.W	R3,#0X3456	00h	12h	00h	00h	5555h	0000h	1200h	3456h	3534h	0052h
PUSH	R3	00h	12h	56h	34h	5555h	0000h	1200h	3456h	3534h	0054h
POP	PSW	00h	12h	56h	34h	5555h	0000h	1200h	3456h	3456h	0052h
POP	PSW	00h	12h	56h	34h	5555h	0000h	1200h	3456h	1200h	0050h
MOV.W	R0,#0X8118	00h	12h	56h	34h	8118h	0000h	1200h	3456h	0240h	0050h
PUSH	R0	18h	81h	56h	34h	8118h	0000h	1200h	3456h	0240h	0052h
MOV.W	R1,#0X5555	18h	81h	56h	34h	8118h	5555h	1200h	3456h	1200h	0052h
PUSH	R1	18h	81h	55h	55h	8118h	5555h	1200h	3456h	1200h	0054h
POP	PSW	18h	81h	55h	55h	8118h	5555h	1200h	3456h	1000h	0052h
POP	PSW	18h	81h	55h	55h	8118h	5555h	1200h	3456h	1200h	0050h

POP Rs

Instruction code	[0 0 0 0 0 0 0 0][1 0 0 1 s3s2s1s0] 0090H
Argument	Rs = 4bit(R select)
Word count	1
Cycle count	1
Function	(SP)←(SP)-2, Hibyte(Rs)←[SP+1], Lobyte(Rs)← [SP], (PC)←(PC)+2
Affected flags	

[Description]

This instruction decrements the stack pointer (SP) by 2 and transfers the contents of the data memory (RAM) location designated by SP to the general-purpose register designated by Rs.
The legitimate value range designated by Rs is from R0 to R15.

[Example]

		RAM (50h)	RAM (51h)	RAM (52h)	RAM (53h)	R0	R1	R2	R3	SP
		-	-	-	-	-	-	-	-	-
MOV.W	R15,#0X0050	-	-	-	-	-	-	-	-	0050h
MOV.W	R0,#0X5555	-	-	-	-	5555h	-	-	-	0050h
PUSH	R0	55h	55h	-	-	5555h	-	-	-	0052h
MOV.W	R1,#0X0000	55h	55h	-	-	5555h	0000h	-	-	0052h
PUSH	R1	55h	55h	00h	00h	5555h	0000h	-	-	0054h
POP	R2	55h	55h	00h	00h	5555h	0000h	0000h	-	0052h
POP	R3	55h	55h	00h	00h	5555h	0000h	0000h	5555h	0050h
MOV.W	R3,#0X1200	55h	55h	00h	00h	5555h	0000h	0000h	1200h	0050h
PUSH	R3	00h	12h	00h	00h	5555h	0000h	0000h	1200h	0052h
MOV.W	R2,#0X3456	00h	12h	00h	00h	5555h	0000h	3456h	1200h	0052h
PUSH	R2	00h	12h	56h	34h	5555h	0000h	3456h	1200h	0054h
POP	R1	00h	12h	56h	34h	5555h	3456h	3456h	1200h	0052h
POP	R0	00h	12h	56h	34h	1200h	3456h	3456h	1200h	0050h
MOV.W	R0,#0X8118	00h	12h	56h	34h	8118h	3456h	3456h	1200h	0050h
PUSH	R0	18h	81h	56h	34h	8118h	3456h	3456h	1200h	0052h
MOV.W	R1,#0X5555	18h	81h	56h	34h	8118h	5555h	3456h	1200h	0052h
PUSH	R1	18h	81h	55h	55h	8118h	5555h	3456h	1200h	0054h
POP	R2	18h	81h	55h	55h	8118h	5555h	5555h	1200h	0052h
POP	R3	18h	81h	55h	55h	8118h	5555h	5555h	8118h	0050h

PUSH PSW

Instruction code	[0 0 0 0 0 0 0 0][1 0 0 0 1 1 1 0] 008EH
Argument	
Word count	1
Cycle count	1
Function	[SP+1]←Hibyte(PSW), [SP]←Lobyte(PSW), (SP)←(SP)+2,(PC)←(PC)+2
Affected flags	

[Description]

This instruction transfers the contents of the program status word (PSW) to the data memory (RAM) location designated by the stack pointer (SP), then increments the SP by 2.

[Example]

		RAM (50h)	RAM (51h)	RAM (52h)	RAM (53h)	R0	R1	R2	R3	PSW	SP
		-	-	-	-	-	-	-	-	-	-
MOV.W	R15,#0X0050	-	-	-	-	-	-	-	-	F000h	0050h
MOV.W	R0,#0X5555	-	-	-	-	5555h	-	-	-	0000h	0050h
PUSH	PSW	00h	00h	-	-	5555h	-	-	-	0000h	0052h
MOV.W	R1,#0X0000	00h	00h	-	-	5555h	0000h	-	-	1003h	0052h
PUSH	PSW	00h	00h	03h	10h	5555h	0000h	-	-	1003h	0054h
POP	R0	00h	00h	03h	10h	1003h	0000h	-	-	1003h	0052h
POP	R1	00h	00h	03h	10h	1003h	0000h	-	-	1003h	0050h
MOV.W	R2,#0X1200	00h	00h	03h	10h	1003h	0000h	1200h	-	2001h	0050h
PUSH	PSW	01h	20h	03h	10h	1003h	0000h	1200h	-	2001h	0052h
MOV.W	R3,#0X3456	01h	20h	03h	10h	1003h	0000h	1200h	3456h	3020h	0052h
PUSH	PSW	01h	20h	20h	30h	1003h	0000h	1200h	3456h	3020h	0054h
POP	R2	01h	20h	20h	30h	1003h	0000h	3020h	3456h	3020h	0052h
POP	R3	01h	20h	20h	30h	1003h	0000h	3020h	2001h	3020h	0050h
MOV.W	R0,#0X8118	01h	20h	20h	30h	8118h	0000h	3020h	2001h	0040h	0050h
PUSH	PSW	40h	00h	20h	30h	8118h	0000h	3020h	2001h	0040h	0052h
MOV.W	R1,#0X5555	40h	00h	20h	30h	8118h	5555h	3020h	2001h	1000h	0052h
PUSH	PSW	40h	00h	00h	10h	8118h	5555h	3020h	2001h	1000h	0054h
POP	R0	40h	00h	00h	10h	1000h	5555h	3020h	2001h	1000h	0052h
POP	R1	40h	00h	00h	10h	1000h	0040h	3020h	2001h	1000h	0050h

PUSH Rs

Instruction code	[0 0 0 0 0 0 0 0][1 0 0 0 s3s2s1s0] 0080H
Argument	Rs = 4bit(R select)
Word count	1
Cycle count	1
Function	[SP+1]←Hibyte(Rs), [SP]←Lobyte(Rs), (SP)←(SP)+2,(PC)←(PC)+2
Affected flags	

[Description]

This instruction transfers the contents of the general-purpose register designated by Rs to the data memory (RAM) location designated by the stack pointer (SP), then increments the SP by 2.
The legitimate value range designated by Rs is from R0 to R15.

[Example]

		RAM (50h)	RAM (51h)	RAM (52h)	RAM (53h)	R0	R1	R2	R3	SP
		-	-	-	-	-	-	-	-	-
MOV.W	R15,#0X0050	-	-	-	-	-	-	-	-	0050h
MOV.W	R0,#0X5555	-	-	-	-	5555h	-	-	-	0050h
PUSH	R0	55h	55h	-	-	5555h	-	-	-	0052h
MOV.W	R1,#0X0000	55h	55h	-	-	5555h	0000h	-	-	0052h
PUSH	R1	55h	55h	00h	00h	5555h	0000h	-	-	0054h
POP	R2	55h	55h	00h	00h	5555h	0000h	0000h	-	0052h
POP	R3	55h	55h	00h	00h	5555h	0000h	0000h	5555h	0050h
MOV.W	R3,#0X1200	55h	55h	00h	00h	5555h	0000h	0000h	1200h	0050h
PUSH	R3	00h	12h	00h	00h	5555h	0000h	0000h	1200h	0052h
MOV.W	R2,#0X3456	00h	12h	00h	00h	5555h	0000h	3456h	1200h	0052h
PUSH	R2	00h	12h	56h	34h	5555h	0000h	3456h	1200h	0054h
POP	R1	00h	12h	56h	34h	5555h	3456h	3456h	1200h	0052h
POP	R0	00h	12h	56h	34h	1200h	3456h	3456h	1200h	0050h
MOV.W	R0,#0X8118	00h	12h	56h	34h	8118h	3456h	3456h	1200h	0050h
PUSH	R0	18h	81h	56h	34h	8118h	3456h	3456h	1200h	0052h
MOV.W	R1,#0X5555	18h	81h	56h	34h	8118h	5555h	3456h	1200h	0052h
PUSH	R1	18h	81h	55h	55h	8118h	5555h	3456h	1200h	0054h
POP	R2	18h	81h	55h	55h	8118h	5555h	5555h	1200h	0052h
POP	R3	18h	81h	55h	55h	8118h	5555h	5555h	8118h	0050h

RESET

Instruction code	[0 0 0 0 0 0 0 0][0 0 0 0 1 1 1 1] 000FH
Argument	
Word count	1
Cycle count	1
Function	Initialize
Affected flags	

[Description]

The CPU is initialized as the result of executing the RESET instruction.

RET

Instruction code	[0 0 0 0 0 0 0 0][0 0 0 0 0 0 1 1] 0003H
Argument	
Word count	1
Cycle count	3
Function	(PC)←(SP-1<<24+SP-2<<16+SP-3<<8+SP-4), (SP)←(SP)-4
Affected flags	

[Description]

This instruction decrements the stack pointer (SP) and places the contents of the data memory (RAM) location designated by SP to the program counter (PC).

[Example] The value of label LA is 910AH.

		PC	RAM (00h)	RAM (01h)	RAM (02h)	RAM (03h)	R3	PSW	SP
		-	-	-	-	-	-	-	-
	MOV.W R15,#0X0000	9004h	-	-	-	-	-	F003h	0000h
	MOV.W R3,#0XFFFF	9008h	-	-	-	-	FFFFh	3040h	0000h
loop:									
	CALLF LA ;; CALL LA	910Ah	0Ch	90h	00h	00h	FFFFh	3040h	0004h
	INC R3	900Eh	0Ch	90h	00h	00h	0001h	3020h	0000h
	NOP	9010h	0Ch	90h	00h	00h	0001h	3020h	0000h
	.								
	.								
	.								
LA:									
	INC R3	910Ch	0Ch	90h	00h	00h	0000h	3003h	0004h
	RET	900Ch	0Ch	90h	00h	00h	0000h	3003h	0000h

REV Rd

Instruction code	[0 0 1 1 0 0 0 0][1 1 1 1 d3d2d1d0] 30F0H
Argument	Rd = 4bit(R select)
Word count	1
Cycle count	1
Function	(Rd)←mirror(Rd), (PC)←(PC)+2
Affected flags	Z8,Z16,P,S,N0 to N3

[Description]

This instruction swaps the contents (exchanges the MSB and LSB sides) of the general-purpose register designated by Rd.
The legitimate value range designated by Rd is from R0 to R15.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-
MOV.W	R0,#0XCDEF	CDEFh	-	-	-	0	0	0	0	1
MOV.W	R1,#0X0000	CDEFh	0000h	-	-	1	1	1	0	0
MOV.W	R2,#0X8888	CDEFh	0000h	8888h	-	2	0	0	0	1
MOV.W	R3,#0X5500	CDEFh	0000h	8888h	5500h	3	1	0	0	0
REV	R0	F7B3h	0000h	8888h	5500h	0	0	0	0	1
REV	R1	F7B3h	0000h	8888h	5500h	1	1	1	0	0
REV	R2	F7B3h	0000h	1111h	5500h	2	0	0	0	0
REV	R3	F7B3h	0000h	1111h	00AAh	3	0	0	0	0

RLC Rd, #imm4

Instruction code	[0 0 1 1 1 0 1 1][i3i2i1i0d3d2d1d0] 3B00H
Argument	Rd = 4bit(R select),imm4 = 4bit(immediate data)
Word count	1
Cycle count	1
Function	(Rd)←(Rd) rotate left #imm4 bit through carry (PC)←(PC)+2
Affected flags	Z8,Z16,CY,P,S,N0 to N3

[Description]

This instruction rotates the contents of the general-purpose register Rd through the carry flag (CY) (17-bit space) to the left by the amount (rotate amount) designated by immediate data imm4.
The legitimate value range designated by Rd is from R0 to R15 and that by imm4 is from 0 to F.

[Example]

	R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	P	S
	-	-	-	-	-	-	-		-	-
MOV.W R0,#0XBA98	BA98h	-	-	-	0	0	0	-	0	1
MOV.W R1,#0XF123	BA98h	F123h	-	-	1	0	0	-	0	1
MOV.W R2,#0X0000	BA98h	F123h	0000h	-	2	1	1	-	0	0
MOV.W R3,#0X8761	BA98h	F123h	0000h	8761h	3	0	0	-	1	1
CLR1 R14,#2	BA98h	F123h	0000h	8761h	E	0	0	0	0	0
RLC R0,#0X03	D4C2h	F123h	0000h	8761h	0	0	0	1	1	1
RLC R1,#0X00	D4C2h	F123h	0000h	8761h	1	0	0	1	0	1
RLC R2,#0X01	D4C2h	F123h	0001h	8761h	2	0	0	0	1	0
RLC R3,#0X02	D4C2h	F123h	0001h	1D85h	3	0	0	0	1	0

RLC Rd, Rs

Instruction code	[0 0 1 1 1 0 1 0][s3s2s1s0d3d2d1d0] 3A00H
Argument	Rd = 4bit(R select),Rs = 4bit(R select)
Word count	1
Cycle count	1
Function	(Rd)←(Rd) rotate left (Rs)&000Fh bit through carry (PC)←(PC)+2
Affected flags	Z8,Z16,CY,P,S,N0 to N3

[Description]

This instruction rotates the contents of the general-purpose register Rd through the carry flag (CY) (17-bit space) to the left by the amount (rotate amount) of the lower-order 4 bits of the general-purpose register designated by Rs.

The legitimate value range designated by Rd is from R0 to R15 and that by Rs is from R0 to R15.

[Example]

	R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	P	S
	-	-	-	-	-	-	-		-	-
MOV.W R0,#0XBA98	BA98h	-	-	-	0	0	0	-	0	1
MOV.W R1,#0XF123	BA98h	F123h	-	-	1	0	0	-	0	1
MOV.W R2,#0X0000	BA98h	F123h	0000h	-	2	1	1	-	0	0
MOV.W R3,#0X8761	BA98h	F123h	0000h	8761h	3	0	0	-	1	1
CLR1 R14,#2	BA98h	F123h	0000h	8761h	E	0	0	0	0	0
RLC R0,R1	D4C2h	F123h	0000h	8761h	0	0	0	1	1	1
RLC R1,R2	D4C2h	F123h	0000h	8761h	1	0	0	1	0	1
RLC R2,R3	D4C2h	F123h	0001h	8761h	2	0	0	0	1	0
RLC R3,R0	D4C2h	F123h	0001h	1D85h	3	0	0	0	1	0

RRC Rd, #imm4

Instruction code	[0 0 1 1 1 0 0 1][i3i2i1i0d3d2d1d0] 3900H
Argument	Rd = 4bit(R select),imm4 = 4bit(immediate data)
Word count	1
Cycle count	1
Function	(Rd)←(Rd) rotate right #imm4 bit through carry (PC)←(PC)+2
Affected flags	Z8,Z16,CY,P,S,N0 to N3

[Description]

This instruction rotates the contents of the general-purpose register Rd through the carry flag (CY) (17-bit space) to the right by the amount (rotate amount) designated by immediate data imm4.
The legitimate value range designated by Rd is from R0 to R15 and that by imm4 is from 0 to F.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	P	S
		-	-	-	-	-	-	-		-	-
MOV.W	R0,#0X1234	1234h	-	-	-	0	0	0	-	1	0
MOV.W	R1,#0XF123	1234h	F123h	-	-	1	0	0	-	0	1
MOV.W	R2,#0X0000	1234h	F123h	0000h	-	2	1	1	-	0	0
MOV.W	R3,#0X8761	1234h	F123h	0000h	8761h	3	0	0	-	1	1
CLR1	R14,#2	1234h	F123h	0000h	8761h	E	0	0	0	0	0
RRC	R0,#0X03	0246h	F123h	0000h	8761h	0	0	0	1	0	0
RRC	R1,#0X00	0246h	F123h	0000h	8761h	1	0	0	1	0	1
RRC	R2,#0X01	0246h	F123h	8000h	8761h	2	1	0	0	1	1
RRC	R3,#0X06	0246h	F123h	8000h	0A1Dh	3	0	0	1	0	0

RRC Rd, Rs

Instruction code	[0 0 1 1 1 0 0 0][s3s2s1s0d3d2d1d0] 3800H
Argument	Rd = 4bit(R select),Rs = 4bit(R select)
Word count	1
Cycle count	1
Function	(Rd)←(Rd) rotate right (Rs)&000Fh bit through carry (PC)←(PC)+2
Affected flags	Z8,Z16,CY,P,S,N0 to N3

[Description]

This instruction rotates the contents of the general-purpose register Rd through the carry flag (CY) (17-bit space) to the right by the amount (rotate amount) of the lower-order 4 bits of the general-purpose register designated by Rs.

The legitimate value range designated by Rd is from R0 to R15 and that by Rs is from R0 to R15.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	P	S
		-	-	-	-	-	-	-		-	-
MOV.W	R0,#0X1234	1234h	-	-	-	0	0	0	-	1	0
MOV.W	R1,#0XF123	1234h	F123h	-	-	1	0	0	-	0	1
MOV.W	R2,#0X0000	1234h	F123h	0000h	-	2	1	1	-	0	0
MOV.W	R3,#0X8761	1234h	F123h	0000h	8761h	3	0	0	-	1	1
CLR1	R14,#2	1234h	F123h	0000h	8761h	E	0	0	0	0	0
RRC	R0,R1	0246h	F123h	0000h	8761h	0	0	0	1	0	0
RRC	R1,R2	0246h	F123h	0000h	8761h	1	0	0	1	0	1
RRC	R2,R3	0246h	F123h	8000h	8761h	2	1	0	0	1	1
RRC	R3,R0	0246h	F123h	8000h	0A1Dh	3	0	0	1	0	0

SBC Rd, #imm4

Instruction code	[0 1 0 1 0 1 1 1][i3i2i1i0d3d2d1d0] 5700H
Argument	Rd = 4bit(R select),imm4 = 4bit(immediate data)
Word count	1
Cycle count	1
Function	(Rd)←(Rd) - #imm4 – CY, (PC)←(PC)+2
Affected flags	Z8,Z16,CY,HC,OV,P,S,N0 to N3

[Description]

This instruction subtracts immediate data designated by imm4 and the value of the carry flag (CY) from the contents of the general-purpose register designated by Rd and places the result in Rd.
The legitimate value range designated by Rd is from R0 to R15 and that by imm4 is from 0 to F.

[Example]

	R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	HC	OV	P	S
	-	-	-	-	-	-	-				-	-
MOV.W R0,#0X0034	0034h	-	-	-	0	0	0	-	-	-	1	0
MOV.W R1,#0X0001	0034h	0001h	-	-	1	0	0	-	-	-	1	0
MOV.W R2,#0XBA98	0034h	0001h	BA98h	-	2	0	0	-	-	-	0	1
MOV.W R3,#0X3456	0034h	0001h	BA98h	3456h	3	0	0	-	-	-	1	0
SBC R0,#0X4	0030h	0001h	BA98h	3456h	0	0	0	0	0	0	0	0
SBC R1,#0XF	0030h	FFF2h	BA98h	3456h	1	0	0	1	1	0	1	1
SBC R2,#0X8	0030h	FFF2h	BA8Fh	3456h	2	0	0	0	1	0	0	1
SBC R3,#0X1	0030h	FFF2h	BA8Fh	3455h	3	0	0	0	0	0	1	0

SBC Rd, #imm16

Instruction code	[0 0 1 1 0 0 0 1][0 1 1 1 d3d2d1d0][i15 to i8][i7 to i0] 3170H
Argument	Rd = 4bit(R select),imm16 = 16bit(immediate data)
Word count	2
Cycle count	2
Function	(Rd)←(Rd) - #imm16 – CY, (PC)←(PC)+4
Affected flags	Z8,Z16,CY,HC,OV,P,S,N0 to N3

[Description]

This instruction subtracts immediate data designated by imm16 and the value of the carry flag (CY) from the contents of the general-purpose register designated by Rd and places the result in Rd.
The legitimate value range designated by Rd is from R0 to R15 and that by imm16 is from 0 to FFFF.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	HC	OV	P	S
		-	-	-	-	-	-	-				-	-
MOV.W	R0,#0X1234	1234h	-	-	-	0	0	0	-	-	-	1	0
MOV.W	R1,#0X0001	1234h	0001h	-	-	1	0	0	-	-	-	1	0
MOV.W	R2,#0XBA98	1234h	0001h	BA98h	-	2	0	0	-	-	-	0	1
MOV.W	R3,#0X8765	1234h	0001h	BA98h	8765h	3	0	0	-	-	-	0	1
SBC	R0,#0X1234	0000h	0001h	BA98h	8765h	0	1	1	0	0	0	0	0
SBC	R1,#0XFFFF	0000h	0002h	BA98h	8765h	1	0	0	1	1	0	1	0
SBC	R2,#0X9898	0000h	0002h	21FFh	8765h	2	0	0	0	1	0	0	0
SBC	R3,#0X5678	0000h	0002h	21FFh	30EDh	3	0	0	0	1	1	0	0

SBC Rx, #imm8

Instruction code	[0 1 0 1 1 1 1 1][i7i6i5i4i3i2i1i0] 5F00H
Argument	imm8 = 8bit(immediate data)
Word count	1
Cycle count	1
Function	(Rx)←(Rx) - #imm8 - CY,(PC)←(PC)+2
Affected flags	Z8,Z16,CY,HC,OV,P,S

[Description]

This instruction subtracts immediate data designated by imm8 and the value of the carry flag (CY) from the contents of the general-purpose register Rx designated indirectly by bits 12 to 15 (N0 to N3) of the PSW and places the result in Rx.
The legitimate value range designated by imm8 is from 0 to FF.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	HC	OV	P	S
		-	-	-	-	-	-	-				-	-
MOV.W	R3,#0X3456	-	-	-	3456h	3	0	0	-	-	-	1	0
MOV.W	R2,#0XFFFF	-	-	FFFFh	3456h	2	0	0	-	-	-	0	1
MOV.W	R1,#0X7654	-	7654h	FFFFh	3456h	1	0	0	-	-	-	0	0
MOV.W	R0,#0X8000	8000h	7654h	FFFFh	3456h	0	1	0	-	-	-	1	1
SBC	Rx,#0XF6	7F0Ah	7654h	FFFFh	3456h	0	0	0	0	1	1	1	0
INC	R1	7F0Ah	7655h	FFFFh	3456h	1	0	0	0	1	1	1	0
SBC	Rx,#0X99	7F0Ah	75BCh	FFFFh	3456h	1	0	0	0	1	0	0	0
NOT	R2	7F0Ah	75BCh	0000h	3456h	2	1	1	0	1	0	0	0
SBC	Rx,#0X01	7F0Ah	75BCh	FFFFh	3456h	2	0	0	1	1	0	0	1
SWPB	R3	7F0Ah	75BCh	FFFFh	5634h	3	0	0	1	1	0	1	0
SBC	Rx,#0X55	7F0Ah	75BCh	FFFFh	55DEh	3	0	0	0	1	0	0	0

SBC Rd, Rs

Instruction code	[0 1 0 0 1 1 1 1][s3s2s1s0d3d2d1d0] 4F00H
Argument	Rd = 4bit(R select),Rs = 4bit(R select)
Word count	1
Cycle count	1
Function	(Rd)←(Rd) - (Rs) – CY, (PC)←(PC)+2
Affected flags	Z8,Z16,CY,HC,OV,P,S,N0 to N3

[Description]

This instruction subtracts the contents of the general-purpose register designated by Rs and the value of the carry flag (CY) from the contents of the general-purpose register designated by Rd and places the result in Rd.

The legitimate value range designated by Rd is from R0 to R15 and that by Rs is from R0 to R15.

[Example]

	R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	HC	OV	P	S
	-	-	-	-	-	-	-				-	-
MOV.W R0,#0X1234	1234h	-	-	-	0	0	0	-	-	-	1	0
MOV.W R1,#0X1234	1234h	1234h	-	-	1	0	0	-	-	-	1	0
MOV.W R2,#0X89AB	1234h	1234h	89ABh	-	2	0	0	-	-	-	0	1
MOV.W R3,#0X3456	1234h	1234h	89ABh	3456h	3	0	0	-	-	-	1	0
SBC R0,R1	0000h	1234h	89ABh	3456h	0	1	1	0	0	0	0	0
SBC R1,R2	0000h	8889h	89ABh	3456h	1	0	0	1	1	1	1	1
SBC R2,R3	0000h	8889h	5554h	3456h	2	0	0	0	0	1	1	0
SBC R3,R0	0000h	8889h	5554h	3456h	3	0	0	0	0	0	1	0
SBC R3,R2	0000h	8889h	5554h	DF02h	3	0	0	1	0	0	0	1
SBC R3,R2	0000h	8889h	5554h	89ADh	3	0	0	0	1	0	0	1

SDIV

Instruction code	[0 0 0 0 0 0 0 0][1 1 0 0 1 0 0 0] 00C8H
Argument	
Word count	1
Cycle count	18 to 19 cycles
Function	(R0 : quotient)…(R1 : remainder)←(R0)÷(R2)(signed division), (PC)←(PC)+2
Affected flags	Z8,Z16,P,S,CY(equal to S) HC, OV, and N3 to N0 all cleared.

[Description]

This instruction places the result of dividing the contents (signed 16-bit data) of the general-purpose register R0 by the contents (signed 16-bit data) of the general-purpose register R2 in R0 and the remainder of the division in R1.
No valid result is guaranteed if the value of R2 is 0.

[Example]

	R0	R1	R2	R3	PSW
	-	-	-	-	-
MOV.W R0,#0X89AB	89ABh	-	-	-	0040h
MOV.W R1,#0X5678	89ABh	5678h	-	-	1000h
MOV.W R2,#0X1234	89ABh	5678h	1234h	-	2020h
MOV.W R3,#0XDEF0	89ABh	5678h	1234h	DEF0h	3040h
SDIV	FFFAh	E6E3h	1234h	DEF0h	0044h
MOV.W R0,#0X8000	8000h	E6E3h	1234h	DEF0h	0065h
MOV.W R2,#0X0002	8000h	E6E3h	0002h	DEF0h	2024h
SDIV	C000h	0000h	0002h	DEF0h	0045h
MOV.W R0,#0XFFFF	FFFFh	0000h	0002h	DEF0h	0044h
SDIV	0000h	FFFFh	0002h	DEF0h	0003h

<Note>

The cycle count of this instruction is variable.
The sign of the remainder is identical to that of the dividend.
The flags (Z8, Z16, P, and S) are affected by R0 (quotient).

SDIVLH

Instruction code	[0 0 0 0 0 0 0 0][1 1 1 0 1 0 0 0] 00E8H
Argument	
Word count	1
Cycle count	18 to 19 cycles
Function	(R0 : quotient)…(R1 : remainder)←(R1<<16+R0)÷(R2)(signed division), (PC)←(PC)+2
Affected flags	Z8,Z16,P,S,CY(equal to S) HC, OV, and N3 to N0 all cleared.

[Description]

This instruction places the result of dividing signed 32-bit data (R1<<16+R0) by R2 (signed 16-bit data) in R0 and the remainder of the division in R1.

No valid result is guaranteed if the value of R2 is 0 or the quotient (R0) exceeds the value range of 8000h (-32768) to 7FFFh (32767).

[Example]

		R0	R1	R2	R3	PSW
		-	-	-	-	-
MOV.W	R0,#0X0A9F	0A9Fh	-	-	-	0000h
MOV.W	R1,#0X3AB0	0A9Fh	3AB0h	-	-	1020h
MOV.W	R2,#0X8001	0A9Fh	3AB0h	8001h	-	2040h
MOV.W	R3,#0XDEF0	0A9Fh	3AB0h	8001h	DEF0h	3040h
SDIVLH		8A9Fh	0000h	8001h	DEF0h	0064h
MOV.W	R0,#0X0AA0	0AA0h	0777h	8001h	DEF0h	0004h
MOV.W	R1,#0X3AB0	0AA0h	3AB0h	8001h	DEF0h	1024h
MOV.W	R2,#0X8001	0AA0h	3AB0h	8001h	DEF0h	2044h
SDIVLH		8A9Fh	0001h	8001h	DEF0h	0064h
MOV.W	R0,#0XF560	F560h	0001h	8001h	DEF0h	0044h
MOV.W	R1,#0XC54F	F560h	C54Fh	8001h	DEF0h	1064h
MOV.W	R2,#0X7FFF	F560h	C54Fh	7FFFh	DEF0h	2024h
SDIVLH		8A9Fh	FFFFh	7FFFh	DEF0h	0064h

<Note>

The cycle count of this instruction is variable.

The sign of the remainder is identical to that of the dividend.

The flags (Z8, Z16, P, and S) are affected by R0 (quotient).

SET1 m16, #imm3

Instruction code	[1 1 1 X i2i1i0 1][m7m6m5m4m3m2m1m0] E100H(RAM),F100H(SFR)
Argument	m16 = 16bit(Lower 8bit valid for operation code),imm3 = 3bit(bit select)
Word count	1
Cycle count	2
Function	(m16)←(m16) of bit #imm3 ←1,(PC)←(PC)+2
Affected flags	Z8,Z16,P,S,N0 to N3

[Description]

This instruction sets the bit, in the 2-byte RAM (data memory) location or SFR (one of the registers dedicated to control the internal peripheral devices) addressed by m16, that is designated by immediate data designated by imm3, to 1.
The legitimate value range designated by imm3 is from 0 to 8.

The compiler generates the instruction code while regarding RAM or SFR as the destination of transfer according to the value of m16 (first operand data).
・ When specifying a RAM location, specify m16 with a value from 00H to FFH (0000H to 00FFH). It is disallowed to specify a RAM address not lower than 100H.
・ When specifying a SFR, specify m16 with a value from 7F00H to 7FFFH.
The basic types of generated instruction code are E100H (RAM) and F100H (SFR), respectively, The lower-order 8 bits of m16 are reflected in the behavior of the instruction code.

[Example]

	RAM (50h)	RAM (51h)	RAM (52h)	RAM (53h)	Z8	Z16	P	S
	-	-	-	-	-	-	-	-
MOV.B 0X50,#0XFF	FFh	-	-	-	0	0	0	1
MOV.B 0X51,#0X32	FFh	32h	-	-	0	0	1	0
MOV.B 0X52,#0X00	FFh	32h	00h	-	1	1	0	0
MOV.B 0X53,#0X54	FFh	32h	00h	54h	0	0	1	0
SET1 0X50,#0X02	FFh	32h	00h	54h	0	0	0	1
SET1 0X51,#0X00	FFh	33h	00h	54h	0	0	0	0
SET1 0X52,#0X04	FFh	33h	10h	54h	0	0	1	0
SET1 0X53,#0X07	FFh	33h	10h	D4h	0	0	0	1

SET1 Rd, #imm4

Instruction code	[0 0 0 0 1 0 0 1][i3i2i1i0d3d2d1d0] 0900H
Argument	Rd = 4bit(R select),imm4 = 4bit(bit select)
Word count	1
Cycle count	1
Function	(Rd)←(Rd) of bit #imm4←1,(PC)←(PC)+2
Affected flags	Z8,Z16,P,S,N0 to N3

[Description]

This instruction sets the bit of the general-purpose register designated by Rd that is designated by immediate data designated by imm4 to 1.

The legitimate value range designated by Rd is from R0 to R15 and that by imm4 is from 0 to F.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-
MOV.W	R0,#0X7FFF	7FFFh	-	-	-	0	0	0	1	0
MOV.W	R1,#0X5432	7FFFh	5432h	-	-	1	0	0	0	0
MOV.W	R2,#0X0000	7FFFh	5432h	0000h	-	2	1	1	0	0
MOV.W	R3,#0X7654	7FFFh	5432h	0000h	7654h	3	0	0	0	0
SET1	R0,#0X02	7FFFh	5432h	0000h	7654h	0	0	0	1	0
SET1	R1,#0X00	7FFFh	5433h	0000h	7654h	1	0	0	1	0
SET1	R2,#0X04	7FFFh	5433h	0010h	7654h	2	0	0	1	0
SET1	R3,#0X0F	7FFFh	5433h	0010h	F654h	3	0	0	1	1

SET1 Rd, Rs

Instruction code	[0 0 0 0 1 0 1 1][s3s2s1s0d3d2d1d0] 0B00H
Argument	Rd = 4bit(R select),Rs = 4bit(bit select)
Word count	1
Cycle count	1
Function	(Rd)←(Rd) of bit (Rs)&000Fh ←1,(PC)←(PC)+2
Affected flags	Z8,Z16,P,S,N0 to N3

[Description]

This instruction sets the bit, in the general-purpose register designated by Rd, that is designated by the lower-order 4 bits of the general-purpose register designated by Rs, to 1.
The legitimate value range designated by Rd is from R0 to R15 and that by Rs is from R0 to R15.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-
MOV.W	R0,#0X7FFF	7FFFh	-	-	-	0	0	0	1	0
MOV.W	R1,#0X5432	7FFFh	5432h	-	-	1	0	0	0	0
MOV.W	R2,#0X0000	7FFFh	5432h	0000h	-	2	1	1	0	0
MOV.W	R3,#0X7654	7FFFh	5432h	0000h	7654h	3	0	0	0	0
SET1	R0,R1	7FFFh	5432h	0000h	7654h	0	0	0	1	0
SET1	R1,R2	7FFFh	5433h	0000h	7654h	1	0	0	1	0
SET1	R2,R3	7FFFh	5433h	0010h	7654h	2	0	0	1	0
SET1	R3,R0	7FFFh	5433h	0010h	F654h	3	0	0	1	1

SHL Rd, #imm4

Instruction code	[0 0 1 1 1 1 1 1][i3i2i1i0d3d2d1d0] 3F00H
Argument	Rd = 4bit(R select),imm4 = 4bit(immediate data)
Word count	1
Cycle count	1
Function	(Rd)←(Rd) logical shift left #imm4 bit (CY)←last shift bit, (PC)←(PC)+2
Affected flags	Z8,Z16,CY,P,S,N0 to N3

[Description]

This instruction shifts the contents of the general-purpose register designated by Rd to the left by the amount (shift amount) of immediate data designated by imm4. Finally, the instruction places the overflow bit out of the MSB in the carry flag (CY).

The legitimate value range designated by Rd is from R0 to R15 and that by imm4 is from 0 to F.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	P	S
		-	-	-	-	-	-	-		-	-
MOV.W	R0,#0XCDEF	CDEFh	-	-	-	0	0	0	-	0	1
MOV.W	R1,#0X5432	CDEFh	5432h	-	-	1	0	0	-	0	0
MOV.W	R2,#0X0000	CDEFh	5432h	0000h	-	2	1	1	-	0	0
MOV.W	R3,#0X8761	CDEFh	5432h	0000h	8761h	3	0	0	-	1	1
CLR1	R14,#2	CDEFh	5432h	0000h	8761h	E	0	0	0	0	0
SHL	R0,#0X02	37BCh	5432h	0000h	8761h	0	0	0	1	0	0
SHL	R1,#0X00	37BCh	5432h	0000h	8761h	1	0	0	1	0	0
SHL	R2,#0X01	37BCh	5432h	0000h	8761h	2	1	1	0	0	0
SHL	R3,#0X0C	37BCh	5432h	0000h	1000h	3	1	0	0	1	0

<Note>

The contents of Rd are shifted to the left and are padded with 0s from the LSB side.

SHL Rd, Rs

Instruction code	[0 0 1 1 1 1 1 0][s3s2s1s0d3d2d1d0] 3E00H
Argument	Rd = 4bit(R select),Rs = 4bit(R select)
Word count	1
Cycle count	1
Function	(Rd)←(Rd) logical shift left (Rs)&000Fh bit (CY)←last shift bit,(PC)←(PC)+2
Affected flags	Z8,Z16,CY,P,S,N0 to N3

[Description]

This instruction shifts the contents of the general-purpose register designated by Rd to the left by the amount (shift amount) of the lower-order 4 bits of the general-purpose register designated by Rs. Finally, the instruction places the overflow bit out of the MSB in the carry flag (CY).
The legitimate value range designated by Rd is from R0 to R15 and that of Rs is from R0 to R15.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	P	S
		-	-	-	-	-	-	-		-	-
MOV.W	R0,#0XCDEF	CDEFh	-	-	-	0	0	0	-	0	1
MOV.W	R1,#0X5432	CDEFh	5432h	-	-	1	0	0	-	0	0
MOV.W	R2,#0X0000	CDEFh	5432h	0000h	-	2	1	1	-	0	0
MOV.W	R3,#0X8761	CDEFh	5432h	0000h	8761h	3	0	0	-	1	1
CLR1	R14,#2	CDEFh	5432h	0000h	8761h	E	0	0	0	0	0
SHL	R0,R1	37BCh	5432h	0000h	8761h	0	0	0	1	0	0
SHL	R1,R2	37BCh	5432h	0000h	8761h	1	0	0	1	0	0
SHL	R2,R3	37BCh	5432h	0000h	8761h	2	1	1	0	0	0
SHL	R3,R0	37BCh	5432h	0000h	1000h	3	1	0	0	1	0

<Note>

The contents of Rd are shifted to the left and are padded with 0s from the LSB side.

SHR Rd, #imm4

Instruction code	[0 0 1 1 1 1 0 1][i3i2i1i0d3d2d1d0] 3D00H
Argument	Rd = 4bit(R select),imm4 = 4bit(immediate data)
Word count	1
Cycle count	1
Function	(Rd)←(Rd) logical shift right #imm4 bit (CY)←last shift bit,(PC)←(PC)+2
Affected flags	Z8,Z16,CY,P,S,N0 to N3

[Description]

This instruction shifts the contents of the general-purpose register designated by Rd to the right by the amount (shift amount) of immediate data designated by imm4. Finally, the instruction places the overflow bit out of the LSB in the carry flag (CY).

The legitimate value range designated by Rd is from R0 to R15 and that by imm4 is from 0 to F.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	P	S
		-	-	-	-	-	-	-		-	-
MOV.W	R0,#0XFEDC	FEDCh	-	-	-	0	0	0	-	0	1
MOV.W	R1,#0X9BDF	FEDCh	9BDFh	-	-	1	0	0	-	0	1
MOV.W	R2,#0X0000	FEDCh	9BDFh	0000h	-	2	1	1	-	0	0
MOV.W	R3,#0X8761	FEDCh	9BDFh	0000h	8761h	3	0	0	-	1	1
CLR1	R14,#2	FEDCh	9BDFh	0000h	8761h	E	0	0	0	0	0
SHR	R0,#0X0F	0001h	9BDFh	0000h	8761h	0	0	0	1	1	0
SHR	R1,#0X00	0001h	9BDFh	0000h	8761h	1	0	0	1	0	1
SHR	R2,#0X01	0001h	9BDFh	0000h	8761h	2	1	1	0	0	0
SHR	R3,#0X01	0001h	9BDFh	0000h	43B0h	3	0	0	1	0	0

<Note>

The contents of Rd are shifted to the right and are padded with 0s from the MSB side.

SHR Rd, Rs

Instruction code	[0 0 1 1 1 1 0 0][s3s2s1s0d3d2d1d0] 3C00H
Argument	Rd = 4bit(R select),Rs = 4bit(R select)
Word count	1
Cycle count	1
Function	(Rd)←(Rd) logical shift right (Rs)&000Fh bit (CY)←last shift bit,(PC)←(PC)+2
Affected flags	Z8,Z16,CY,P,S,N0 to N3

[Description]

This instruction shifts the contents of the general-purpose register designated by Rd to the right by the amount (shift amount) of the lower-order 4 bits of the general-purpose register designated by Rs. Finally, the instruction places the overflow bit out of the LSB in the carry flag (CY).
The legitimate value range designated by Rd is from R0 to R15 and that by Rs is from R0 to R15.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	P	S
		-	-	-	-	-	-	-		-	-
MOV.W	R0,#0XFEDC	FEDCh	-	-	-	0	0	0	-	0	1
MOV.W	R1,#0X9BDF	FEDCh	9BDFh	-	-	1	0	0	-	0	1
MOV.W	R2,#0X0000	FEDCh	9BDFh	0000h	-	2	1	1	-	0	0
MOV.W	R3,#0X8761	FEDCh	9BDFh	0000h	8761h	3	0	0	-	1	1
CLR1	R14,#2	FEDCh	9BDFh	0000h	8761h	E	0	0	0	0	0
SHR	R0,R1	0001h	9BDFh	0000h	8761h	0	0	0	1	1	0
SHR	R1,R2	0001h	9BDFh	0000h	8761h	1	0	0	1	0	1
SHR	R2,R3	0001h	9BDFh	0000h	8761h	2	1	1	0	0	0
SHR	R3,R0	0001h	9BDFh	0000h	43B0h	3	0	0	1	0	0

<Note>

The contents of Rd are shifted to the right and are padded with 0s from the MSB side.

SUB Rd, #imm4

Instruction code	[0 1 0 1 0 1 0 1][i3i2i1i0d3d2d1d0] 5500H
Argument	Rd = 4bit(R select),imm4 = 4bit(immediate data)
Word count	1
Cycle count	1
Function	(Rd)←(Rd) - #imm4,(PC)←(PC)+2
Affected flags	Z8,Z16,CY,HC,OV,P,S,N0 to N3

[Description]

This instruction subtracts immediate data designated by imm4 from the contents of the general-purpose register designated by Rd and places the result in Rd.
The legitimate value range designated by Rd is from R0 to R15 and that by imm4 is from 0 to F.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	HC	OV	P	S
		-	-	-	-	-	-	-				-	-
MOV.W	R0,#0X0034	0034h	-	-	-	0	0	0	-	-	-	1	0
MOV.W	R1,#0X0001	0034h	0001h	-	-	1	0	0	-	-	-	1	0
MOV.W	R2,#0XBA98	0034h	0001h	BA98h	-	2	0	0	-	-	-	0	1
MOV.W	R3,#0X3456	0034h	0001h	BA98h	3456h	3	0	0	-	-	-	1	0
SUB	R0,#0X4	0030h	0001h	BA98h	3456h	0	0	0	0	0	0	0	0
SUB	R1,#0XF	0030h	FFF2h	BA98h	3456h	1	0	0	1	1	0	1	1
SUB	R2,#0X8	0030h	FFF2h	BA90h	3456h	2	0	0	0	0	0	1	1
SUB	R3,#0X1	0030h	FFF2h	BA90h	3455h	3	0	0	0	0	0	1	0

SUB Rd, #imm16

Instruction code	[0 0 1 1 0 0 0 1][0 1 1 0 d3d2d1d0][i15 to i8][i7 to i0] 3160H
Argument	Rd = 4bit(R select),imm16 = 16bit(immediate data)
Word count	2
Cycle count	2
Function	(Rd)←(Rd) - #imm16,(PC)←(PC)+4
Affected flags	Z8,Z16,CY,HC,OV,P,S,N0 to N3

[Description]

This instruction subtracts immediate data designated by imm16 from the contents of the general-purpose register designated by Rd and places the result in Rd.
The legitimate value range designated by Rd is from R0 to R15 and that by imm16 is from 0 to FFFF.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	HC	OV	P	S
		-	-	-	-	-	-	-				-	-
MOV.W	R0,#0X1234	1234h	-	-	-	0	0	0	-	-	-	1	0
MOV.W	R1,#0X0001	1234h	0001h	-	-	1	0	0	-	-	-	1	0
MOV.W	R2,#0XBA98	1234h	0001h	BA98h	-	2	0	0	-	-	-	0	1
MOV.W	R3,#0X8765	1234h	0001h	BA98h	8765h	3	0	0	-	-	-	0	1
SUB	R0,#0X1234	0000h	0001h	BA98h	8765h	0	1	1	0	0	0	0	0
SUB	R1,#0XFFFF	0000h	0002h	BA98h	8765h	1	0	0	1	1	0	1	0
SUB	R2,#0X9898	0000h	0002h	2200h	8765h	2	1	0	0	0	0	0	0
SUB	R3,#0X5678	0000h	0002h	2200h	30EDh	3	0	0	0	1	1	0	0

SUB Rx, #imm8

Instruction code	[0 1 0 1 1 1 0 1][i7i6i5i4i3i2i1i0] 5D00H
Argument	imm8 = 8bit(immediate data)
Word count	1
Cycle count	1
Function	(Rx)←(Rx) - #imm8,(PC)←(PC)+2
Affected flags	Z8,Z16,CY,HC,OV,P,S

[Description]

This instruction subtracts immediate data designated by imm8 from the contents of the general-purpose register Rx designated indirectly by the value of bits 12 to 15 (N0 to N3) of the PSW and places the result in Rx.

The legitimate value range designated by imm8 is from 0 to FF.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	HC	OV	P	S
		-	-	-	-	-	-	-				-	-
MOV.W	R3,#0X3456	-	-	-	3456h	3	0	0	-	-	-	1	0
MOV.W	R2,#0XFFFF	-	-	FFFFh	3456h	2	0	0	-	-	-	0	1
MOV.W	R1,#0X7654	-	7654h	FFFFh	3456h	1	0	0	-	-	-	0	0
MOV.W	R0,#0X8000	8000h	7654h	FFFFh	3456h	0	1	0	-	-	-	1	1
SUB	Rx,#0XF6	7F0Ah	7654h	FFFFh	3456h	0	0	0	0	1	1	1	0
INC	R1	7F0Ah	7655h	FFFFh	3456h	1	0	0	0	1	1	1	0
SUB	Rx,#0X99	7F0Ah	75BCh	FFFFh	3456h	1	0	0	0	1	0	0	0
NOT	R2	7F0Ah	75BCh	0000h	3456h	2	1	1	0	1	0	0	0
SUB	Rx,#0X01	7F0Ah	75BCh	FFFFh	3456h	2	0	0	1	1	0	0	1
SWPB	R3	7F0Ah	75BCh	FFFFh	5634h	3	0	0	1	1	0	1	0
SUB	Rx,#0X55	7F0Ah	75BCh	FFFFh	55DFh	3	0	0	0	0	1	0	0

SUB Rd, Rs

Instruction code	[0 1 0 0 1 1 0 1][s3s2s1s0d3d2d1d0] 4D00H
Argument	Rd = 4bit(R select),Rs = 4bit(R select)
Word count	1
Cycle count	1
Function	(Rd)← (Rd) - (Rs),(PC)←(PC)+2
Affected flags	Z8,Z16,CY,HC,OV,P,S,N0 to N3

[Description]

This instruction subtracts the contents of the general-purpose register designated by Rs from the contents of the general-purpose register designated by Rd and places the result in Rd.
The legitimate value range designated by Rd is from R0 to R15 and that by Rs is from R0 to R15.

[Example]

	R0	R1	R2	R3	N3 to N0	Z8	Z16	CY	HC	OV	P	S
	-	-	-	-	-	-	-				-	-
MOV.W R0,#0X1234	1234h	-	-	-	0	0	0	-	-	-	1	0
MOV.W R1,#0X1234	1234h	1234h	-	-	1	0	0	-	-	-	1	0
MOV.W R2,#0X89AB	1234h	1234h	89ABh	-	2	0	0	-	-	-	0	1
MOV.W R3,#0X3456	1234h	1234h	89ABh	3456h	3	0	0	-	-	-	1	0
SUB R0,R1	0000h	1234h	89ABh	3456h	0	1	1	0	0	0	0	0
SUB R1,R2	0000h	8889h	89ABh	3456h	1	0	0	1	1	1	1	1
SUB R2,R3	0000h	8889h	5555h	3456h	2	0	0	0	0	1	0	0
SUB R3,R0	0000h	8889h	5555h	3456h	3	0	0	0	0	0	1	0
SUB R3,R2	0000h	8889h	5555h	DF01h	3	0	0	1	0	0	0	1
SUB R3,R2	0000h	8889h	5555h	89ACh	3	0	0	0	1	0	1	1

SWPB Rd

Instruction code	[0 0 1 1 0 0 0 0][1 0 0 0 d3d2d1d0] 3080H
Argument	Rd = 4bit(R select)
Word count	1
Cycle count	1
Function	Hibyte(Rd) ⇔ Lobyte(Rd) (PC)←(PC)+2
Affected flags	Z8,Z16,P,S,N0 to N3

[Description]

This instruction swaps the higher-order 8 bits of the general-purpose register designated by Rd with its lower-order 8 bits.

The legitimate value range designated by Rd is from R0 to R15.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-
MOV.W	R0,#0X5678	5678h	-	-	-	0	0	0	0	0
SWPB	R0	7856h	-	-	-	0	0	0	0	0
MOV.W	R1,#0X0000	7856h	0000h	-	-	1	1	1	0	0
SWPB	R1	7856h	0000h	-	-	1	1	1	0	0
MOV.W	R2,#0X1200	7856h	0000h	1200h	-	2	1	0	0	0
SWPB	R2	7856h	0000h	0012h	-	2	0	0	0	0
MOV.W	R3,#0X3456	7856h	0000h	0012h	3456h	3	0	0	1	0
SWPB	R3	7856h	0000h	0012h	5634h	3	0	0	1	0
MOV.W	R0,#0X8118	8118h	0000h	0012h	5634h	0	0	0	0	1
SWPB	R0	1881h	0000h	0012h	5634h	0	0	0	0	0
MOV.W	R1,#0X5678	1881h	5678h	0012h	5634h	1	0	0	0	0
SWPB	R1	1881h	7856h	0012h	5634h	1	0	0	0	0

SWPN Rd

Instruction code	[0 0 1 1 0 0 0 0][1 0 0 1 d3d2d1d0] 3090H
Argument	Rd = 4bit(R select)
Word count	1
Cycle count	1
Function	Hibyte(Rd)←Hibyte(Rd), Lobyte(Rd)←(Rd)&000Fh<<4+(Rd)&00F0h>>4 (PC)←(PC)+2
Affected flags	Z8,Z16,P,S,N0 to N3

[Description]

This instruction swaps between the higher- and lower-order 4 bits of the lower-order 8 bits of the general-purpose register designated by Rd.
The legitimate value range designated by Rd is from R0 to R15.

[Example]

	R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
	-	-	-	-	-	-	-	-	-
MOV.W R0,#0X5678	5678h	-	-	-	0	0	0	0	0
SWPN R0	5687h	-	-	-	0	0	0	0	0
MOV.W R1,#0X0000	5687h	0000h	-	-	1	1	1	0	0
SWPN R1	5687h	0000h	-	-	1	1	1	0	0
MOV.W R2,#0X1200	5687h	0000h	1200h	-	2	1	0	0	0
SWPN R2	5687h	0000h	1200h	-	2	1	0	0	0
MOV.W R3,#0X3456	5687h	0000h	1200h	3456h	3	0	0	1	0
SWPN R3	5687h	0000h	1200h	3465h	3	0	0	1	0
MOV.W R0,#0X8118	8118h	0000h	1200h	3465h	0	0	0	0	1
SWPN R0	8181h	0000h	1200h	3465h	0	0	0	0	1
MOV.W R1,#0X5678	8181h	5678h	1200h	3465h	1	0	0	0	0
SWPN R1	8181h	5687h	1200h	3465h	1	0	0	0	0

<Note>

The value of the higher-order 8 bits of Rd remains unchanged.

SWPW Rd, Rs

Instruction code	[0 0 1 1 0 0 1 0][s3s2s1s0d3d2d1d0] 3200H
Argument	Rd = 4bit(R select),Rs = 4bit(R select)
Word count	1
Cycle count	2
Function	(Rd)⇔(Rs) exchange,(PC)←(PC)+2
Affected flags	Z8,Z16,P,S,N0 to N3

[Description]

This instruction swaps the contents of the general-purpose register designated by Rd with the contents of the general-purpose register designated by Rs.
The legitimate value range designated by Rd is from R0 to R15 and that by Rs is from R0 to R15.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-
MOV.W	R0,#0X5678	5678h	-	-	-	0	0	0	0	0
MOV.W	R1,#0X0000	5678h	0000h	-	-	1	1	1	0	0
MOV.W	R2,#0X1200	5678h	0000h	1200h	-	2	1	0	0	0
MOV.W	R3,#0X3456	5678h	0000h	1200h	3456h	3	0	0	1	0
SWPW	R0,R1	0000h	5678h	1200h	3456h	0	1	1	0	0
SWPW	R1,R2	0000h	1200h	5678h	3456h	1	1	0	0	0
SWPW	R2,R3	0000h	1200h	3456h	5678h	2	0	0	1	0
MOV.W	R0,#0X8118	8118h	1200h	3456h	5678h	0	0	0	0	1
MOV.W	R1,#0X5678	8118h	5678h	3456h	5678h	1	0	0	0	0
SWPW	R3,R0	5678h	5678h	3456h	8118h	3	0	0	0	1
SWPW	R0,R1	5678h	5678h	3456h	8118h	0	0	0	0	0

XOR Rd, Rs

Instruction code	[0 1 0 0 0 1 0 0][s3s2s1s0d3d2d1d0] 4400H
Argument	Rd = 4bit(R select),Rs = 4bit(R select)
Word count	1
Cycle count	1
Function	(Rd)←(Rd) ^ (Rs),(PC)←(PC)+2
Affected flags	Z8,Z16,P,S,N0 to N3

[Description]

This instruction takes the exclusive OR of the contents of the general-purpose register designated by Rd and the contents of the general-purpose register designated by Rs and places the result in Rd.
The legitimate value range designated by Rd is from R0 to R15 and that by Rs is from R0 to R15.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-
MOV.W	R0,#0X5678	5678h	-	-	-	0	0	0	0	0
MOV.W	R1,#0X0000	5678h	0000h	-	-	1	1	1	0	0
MOV.W	R2,#0XFEDC	5678h	0000h	FEDCh	-	2	0	0	0	1
MOV.W	R3,#0X3456	5678h	0000h	FEDCh	3456h	3	0	0	1	0
XOR	R0,R1	5678h	0000h	FEDCh	3456h	0	0	0	0	0
XOR	R1,R2	5678h	FEDCh	FEDCh	3456h	1	0	0	0	1
XOR	R2,R3	5678h	FEDCh	CA8Ah	3456h	2	0	0	1	1
XOR	R3,R0	5678h	FEDCh	CA8Ah	622Eh	3	0	0	1	0

XOR Rd, #imm16

Instruction code	[0 0 1 1 0 0 0 1][0 0 1 0 d3d2d1d0][i15 to i8][i7 to i0] 3120H
Argument	Rd = 4bit(R select),imm16 = 16bit(immediate data)
Word count	2
Cycle count	2
Function	(Rd)←(Rd) ^ #imm16,(PC)←(PC)+4
Affected flags	Z8,Z16,P,S,N0 to N3

[Description]

This instruction takes the exclusive OR of the contents of the general-purpose register designated by Rd and immediate data designated by imm16 and places the result in Rd.
The legitimate value range designated by Rd is from R0 to R15 and that by imm16 is from 0 to FFFF.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-
MOV.W	R0,#0X5678	5678h	-	-	-	0	0	0	0	0
MOV.W	R1,#0X0000	5678h	0000h	-	-	1	1	1	0	0
MOV.W	R2,#0XFEDC	5678h	0000h	FEDCh	-	2	0	0	0	1
MOV.W	R3,#0X3456	5678h	0000h	FEDCh	3456h	3	0	0	1	0
XOR	R0,#0X0078	5600h	0000h	FEDCh	3456h	0	1	0	0	0
XOR	R1,#0X0000	5600h	0000h	FEDCh	3456h	1	1	1	0	0
XOR	R2,#0X0012	5600h	0000h	FECEh	3456h	2	0	0	0	1
XOR	R3,#0XFFFF	5600h	0000h	FECEh	CBA9h	3	0	0	1	1

XOR Rx, #imm8

Instruction code	[0 1 0 0 0 1 0 1][i7i6i5i4i3i2i1i0] 4500H
Argument	imm8 = 8bit(immediate data)
Word count	1
Cycle count	1
Function	(Rx)← (Rx) ^ 16bit data(Hibyte=00H, Lobyte=#imm8),(PC)←(PC)+2
Affected flags	Z8,Z16,P,S

[Description]

This instruction takes the exclusive OR of the contents of the general-purpose register Rx designated indirectly by the value of bits 12 to 15 (N0 to N3) of the PSW and the 16-bit data of which the higher-order 8 bits are 00h and the lower-order 8 bits are immediate data designated by imm8 and places the result in Rx.
The legitimate value range designated by imm8 is from 0 to FF.

[Example]

		R0	R1	R2	R3	N3 to N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-
MOV.W	R3,#0X3456	-	-	-	3456h	3	0	0	1	0
MOV.W	R2,#0XFEDC	-	-	FEDCh	3456h	2	0	0	0	1
MOV.W	R1,#0X0001	-	0001h	FEDCh	3456h	1	0	0	1	0
MOV.W	R0,#0X5678	5678h	0001h	FEDCh	3456h	0	0	0	0	0
XOR	Rx,#0X78	5600h	0001h	FEDCh	3456h	0	1	0	0	0
DEC	R1	5600h	0000h	FEDCh	3456h	1	1	1	0	0
XOR	Rx,#0X00	5600h	0000h	FEDCh	3456h	1	1	1	0	0
SWPB	R2	5600h	0000h	DCFEh	3456h	2	0	0	0	1
XOR	Rx,#0X01	5600h	0000h	DCFFh	3456h	2	0	0	1	1
DEC	R3	5600h	0000h	DCFFh	3455h	3	0	0	1	0
XOR	Rx,#0XFF	5600h	0000h	DCFFh	34AAh	3	0	0	1	0

Instructions

Important Note

This document is designed to provide the reader with accurate information in easily understandable form regarding the device features and the correct device implementation procedures.
The sample configurations included in the various descriptions are intended for reference only and should not be directly incorporated in user product configurations.
ON Semiconductor shall bear no responsibility for obligations concerning patent infringements, safety or other legal disputes arising from prototypes or actual products created using the information contained herein.

LC88 SERIES CHAPTER 5 INSTRUCTIONS USER'S MANUAL

Rev. 0 December 18, 2015

Microcontroller Business Unit
ON Semiconductor