

# AND9352/D

---

CMOS 16-BIT MICROCONTROLLER

## LC88 シリーズ 第 5 章 命令 ユーザーズマニュアル



**ON Semiconductor®**

[www.onsemi.jp](http://www.onsemi.jp)

---

### APPLICATION NOTE

オン・セミコンダクター  
マイクロコントローラービジネスユニット

ON Semiconductor and the ON logo are registered trademarks of Semiconductor Components Industries, LLC (SCILLC) or its subsidiaries in the United States and/or other countries. SCILLC owns the rights to a number of patents, trademarks, copyrights, trade secrets, and other intellectual property. A listing of SCILLC's product/patent coverage may be accessed at [www.onsemi.com/site/pdf/Patent-Marking.pdf](http://www.onsemi.com/site/pdf/Patent-Marking.pdf). SCILLC reserves the right to make changes without further notice to any products herein. SCILLC makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does SCILLC assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation special, consequential or incidental damages. "Typical" parameters which may be provided in SCILLC data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. SCILLC does not convey any license under its patent rights nor the rights of others. SCILLC products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the SCILLC product could create a situation where personal injury or death may occur. Should Buyer purchase or use SCILLC products for any such unintended or unauthorized application, Buyer shall indemnify and hold SCILLC and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that SCILLC was negligent regarding the design or manufacture of the part. SCILLC is an Equal Opportunity/Affirmative Action Employer. This literature is subject to all applicable copyright laws and is not for resale in any manner.

(参考訳)

ON Semiconductor及びONのロゴは、Semiconductor Components Industries, LLC (SCILLC) 若しくはその子会社の米国及び/または他の国における登録商標です。SCILLCは特許、商標、著作権、トレードシークレット(営業秘密)と他の知的所有権に対する権利を保有します。SCILLCの製品/特許の適用対象リストについては、以下のリンクからご覧いただけます。[www.onsemi.com/site/pdf/Patent-Marking.pdf](http://www.onsemi.com/site/pdf/Patent-Marking.pdf)。SCILLCは通告なしで、本書記載の製品の変更を行うことがあります。SCILLCは、いかなる特定の目的での製品の適合性について保証しておらず、また、お客様の製品において回路の応用や使用から生じた責任、特に、直接的、間接的、偶発的な損害に対して、いかなる責任も負うことはできません。SCILLCデータシートや仕様書に示される可能性のある「標準的」パラメータは、アプリケーションによっては異なることもあり、実際の性能も時間の経過により変化する可能性があります。「標準的」パラメータを含むすべての動作パラメータは、ご使用になるアプリケーションに応じて、お客様の専門技術者において十分検証されるようお願い致します。SCILLCは、その特許権やその他の権利の下、いかなるライセンスも許諾しません。SCILLC製品は、人体への外科的移植を目的とするシステムへの使用、生命維持を目的としたアプリケーション、また、SCILLC製品の不具合による死傷等の事故が起こり得るようなアプリケーションなどへの使用を意図した設計はされておらず、また、これらを使用対象としておりません。お客様が、このような意図されたものではない、許可されていないアプリケーション用にSCILLC製品を購入または使用した場合、たとえ、SCILLCがその部品の設計または製造に関して過失があったと主張されたとしても、そのような意図せぬ使用、また未許可の使用に関連した死傷等から、直接、又は間接的に生じるすべてのクレーム、費用、損害、経費、および弁護士料などを、お客様の責任において補償をお願いいたします。また、SCILLCとその役員、従業員、子会社、関連会社、代理店に対して、いかなる損害も与えないものとします。

SCILLCは雇用機会均等/差別撤廃雇用主です。この資料は適用されるあらゆる著作権法の対象となっており、いかなる方法によっても再販することはできません。

## 5 命令

### 5-1 概要

Xstormy16の命令は、下表の様に分類されます。命令語長は、ワード(16ビット)の整数倍です。命令のオペランド数は0から3です。

- 命令の分類

機能	命令	Operand	動作
データ転送	MOV, MOVF, MASK	2	$op1 \leftarrow op2$
PUSH,POP	PUSH, POP	1	$SP \leftrightarrow op1$
SWAP	SWPN, SWPB, SWPW	1/2	$op1 \leftrightarrow op1$ または $op1 \leftrightarrow op2$
論理演算	AND, OR, XOR, NOT	1/2	$op1 \leftarrow f(op1, op2), op1 \leftarrow \text{not}(op1)$
算術演算	ADD, ADC, SUB, SBC, INC, DEC	1/2	$op1 \leftarrow f(op1, op2), op1 \leftarrow \text{inc/dec}(op1)$
論理シフト	RRC, RLC, SHR, SHL	2	Shift/Rotate $op1$ by $op2$ .
算術シフト	ASR	2	Shift $op1$ by $op2$ .
ビット操作	SET1, CLR1	2	Set/Clear bit $op2$ of $op1$ .
データ変換	CBW, REV	1	$op1[15:8] \leftarrow op1[7]$
条件分岐	BGE, BNC, BLT, BC, BGT, BHI, BLE, BLS, BPL, BNV, BMI, BV, BNZ, BZ	3	If test( $op1-op2$ ), then branch by $op3$ .
ビット条件分岐	BN, BP	3	If test(bit $op2$ of $op1$ ), then branch by $op3$ .
フラグ条件分岐	BGE, BNC, BLT, BC, BGT, BHI, BLE, BLS, BPL, BNV, BMI, BV, BNZ, BZ	1	If flag, then branch by $op1$ .
無条件分岐	BR, JMP, JMPF	1/2	Branch by $op1$ . Jump to $op1(,2)$ .
無条件コール	CALLR, CALL, CALLF, ICALLR, ICALL, ICALLF	1/2	Branch by $op1$ . Jump to $op1(,2)$ .
リターン	IRET, RET	0	サブルーチンからの復帰。
乗算・除算	MUL, DIV, DIVLH, SDIV, SDIVLH	0	
システム制御	NOP, HALT, HOLD, HOLDX, BRK, RESET	0	システム制御。

## 5-2 アドレッシングモード

### 5-2-1 概要

Xstormy16では、データのアドレッシングは64KBバンク単位で行われ、最大4GB(0~FFFF\_FFFFH)のデータを扱えます。この4GBにはプログラム用のROMデータも含まれます。

プログラムカウンタ(PC)のアドレッシングでは、リニアな4GB(8000H~FFFF\_FFFFH)の空間を扱えます。ただし、絶対アドレスの指定ができるのは16MB(8000H~00FF\_FFFFH)の空間です。

### 5-2-2 アドレッシング(イミディエイト)

	アドレッシングモード	説明	記号
1	イミディエイトデータ	命令コード内のデータがオペランドとなります。	#imm16,#imm8, #imm4,#imm2

### 5-2-3 アドレッシング(汎用レジスタ)

	アドレッシングモード	説明	記号
2	レジスタ直接 (R0~R15)	命令コード内のデータで指定された汎用レジスタがオペランドとなります。	Rd Rs
3	PSW レジスタ間接 (R0~R15)	PSW の bit15~12 で指定された汎用レジスタがオペランドとなります。	Rx, RxL, RxH

### 5-2-4 アドレッシング(ビット指定)

	アドレッシングモード	説明	記号
4	イミディエイトデータ	命令コード内のデータでビットを指定します。	#imm4
5	レジスタ間接	命令コード内のデータで指定された汎用レジスタの内容でビットを指定します。	Rs

### 5-2-5 アドレッシング(シフト量指定)

	アドレッシングモード	説明	記号
6	イミディエイトデータ	命令コード内のデータでシフト・ローテート量を指定します。	#imm3, #imm4
7	レジスタ間接	命令コード内のデータで指定された汎用レジスタの内容でシフト・ローテート量を指定します。	Rs

## 5-2-6 アドレッシング (メモリ: 0~0FFFFH)

	アドレッシングモード	説明	記号
8	SFR 直接アドレス (7F00~7FFFH)	命令コード内の 8 ビットのデータに 7F00H を加えた値をアドレスとして、メモリ上のオペランド(SFR)を指定します。	m16
9	RAM 直接アドレス (0000H~00FFFH)	命令コード内の 8 ビットのデータ値をアドレスとして、メモリ上のオペランドを指定します。	m16
10	レジスタ間接 (0000~FFFFH)	命令コード内のデータで指定された汎用レジスタの内容をアドレス <sup>*1</sup> として、メモリ上のオペランドを指定します。	(Rd) (Rs)
11	ポストインクリメント・レジスタ間接 (0000~FFFFH)	命令コード内のデータで指定された汎用レジスタの内容をアドレス <sup>*1</sup> として、メモリ上のオペランドを指定します。 その後、前記汎用レジスタの内容が+1 (バイトアクセス) または+2 (ワードアクセス) されます。	(Rd++) (Rs++)
12	プリデクリメント・レジスタ間接 (0000~FFFFH)	命令コード内のデータで指定された汎用レジスタの内容は、1 (バイトアクセス) または 2 (ワードアクセス) を減算されます。この値をアドレス <sup>*1</sup> として、メモリ上のオペランドを指定します。	(--Rd) (--Rs)
13	オフセット付きレジスタ間接 (0000~FFFFH)	命令コード内のデータで指定された汎用レジスタの内容に、命令コード内の 12 ビット符号付きオフセットデータを加えた値 <sup>*2</sup> をアドレス <sup>*1</sup> として、メモリ上のオペランドを指定します。	(Rd,±n) (Rs,±n)
14	オフセット付きポストインクリメント・レジスタ間接 (0000~FFFFH)	命令コード内のデータで指定された汎用レジスタの内容に、命令コード内の 12 ビット符号付きオフセットデータを加えた値 <sup>*2</sup> をアドレス <sup>*1</sup> として、メモリ上のオペランドを指定します。その後、前記汎用レジスタの内容が+1 (バイトアクセス) または+2 (ワードアクセス) されます。	(Rd++,±n) (Rs++,±n)
15	オフセット付きプリデクリメント・レジスタ間接 (0000~FFFFH)	命令コード内のデータで指定された汎用レジスタの内容は、1 (バイトアクセス) または 2 (ワードアクセス) を減算されます。この値に、命令コード内の 12 ビット符号付きオフセットデータを加えた値 <sup>*2</sup> をアドレス <sup>*1</sup> として、メモリ上のオペランドを指定します。	(--Rd,±n) (--Rs,±n)

(注) \*1:ワードアクセスのとき、アドレスのLSBを1としてオペランドの上位バイト、0としてオペランドの下部バイトを指定します。

\*2:16ビット演算結果のキャリー・ボローは無視されます。

## 5-2-7 アドレッシング (メモリ: 0 ~ FFFF\_FFFFH)

	アドレッシングモード	説明	記号
16	アドレス拡張・レジスタ間接 (0000_0000 ~ FFFF_FFFFH)	命令コード内のデータで指定された汎用レジスタの内容を下位 16 ビットとし、レジスタ R8 の内容を上位 16 ビットとした 32 ビットのアドレス <sup>*1</sup> として、メモリ上のオペランドを指定します。	(Rd) (Rs)
17	アドレス拡張・ポストインクリメント・レジスタ間接 (0000_0000 ~ FFFF_FFFFH)	命令コード内のデータで指定された汎用レジスタの内容を下位 16 ビットとし、レジスタ R8 の内容を上位 16 ビットとした 32 ビットのアドレス <sup>*1</sup> として、メモリ上のオペランドを指定します。その後、前記汎用レジスタの内容が+1 (バイトアクセス) または+2 (ワードアクセス) されます。	(Rd++) (Rs++)
18	アドレス拡張・プリデクリメント・レジスタ間接 (0000_0000 ~ FFFF_FFFFH)	命令コード内のデータで指定された汎用レジスタの内容は、1 (バイトアクセス) または 2 (ワードアクセス) を減算されます。この値を下位 16 ビットとし、レジスタ R8 の内容を上位 16 ビットとした 32 ビットアドレス <sup>*1</sup> で、メモリ上のオペランドを指定します。	(--Rd) (--Rs)
19	アドレス拡張・オフセット付きレジスタ間接 <sup>*2</sup> (0000_0000 ~ FFFF_FFFFH)	命令コード内のデータで指定された汎用レジスタの内容に、命令コード内の 12 ビット符号付きオフセットデータを加えた値 <sup>*2</sup> を下位 16 ビットとし、命令コード内で指定されたベースレジスタの内容を上位 16 ビットとした 32 ビットのアドレス <sup>*1</sup> で、メモリ上のオペランドを指定します。	(Rb,Rd,±n) (Rb,Rs,±n)
20	アドレス拡張・オフセット付きポストインクリメント・レジスタ間接 <sup>*2</sup> (0000_0000 ~ FFFF_FFFFH)	命令コード内のデータで指定された汎用レジスタの内容に、命令コード内の 12 ビット符号付きオフセットデータを加えた値 <sup>*2</sup> を下位 16 ビットとし、命令コード内で指定されたベースレジスタの内容を上位 16 ビットとした 32 ビットのアドレス <sup>*1</sup> で、メモリ上のオペランドを指定します。 その後、前記汎用レジスタの内容が+1 (バイトアクセス) または+2 (ワードアクセス) されます。	(Rb,Rd++,±n) (Rb,Rs++,±n)
21	アドレス拡張・オフセット付きプリデクリメント・レジスタ間接 <sup>*2</sup> (0000_0000 ~ FFFF_FFFFH)	命令コード内のデータで指定された汎用レジスタの内容は、1 (バイトアクセス) または 2 (ワードアクセス) を減算されます。この値に、命令コード内の 12 ビット符号付きオフセットデータを加えた値 <sup>*2</sup> を下位 16 ビットとし、命令コード内で指定されたベースレジスタの内容を上位 16 ビットとした 32 ビットのアドレス <sup>*1</sup> で、メモリ上のオペランドを指定します。	(Rb,--Rd,±n) (Rb,--Rs,±n)

(注) \*1:ワードアクセスのとき、アドレスのLSBを1としてオペランドの上位バイト、0としてオペランドの下位バイトを指定します。

\*2:16ビット演算結果のキャリー・ボローは無視されます。

## 5-2-8 アドレッシング (プログラムカウンタ PC)

	アドレッシングモード	説明	記号
22	直接絶対 PC (00_0000～ FF_FFFFH)	命令コード内の 24 ビットのデータで PC 値を直接指定します。	a24
23	レジスタ間接絶対 PC (0000_0000H～ FFFF_FFFFH)	命令コード内のデータで指定された 2 つの汎用レジスタの内容を結合した 32 ビットのデータで PC 値を直接指定します。	(Rb, Rs)
24	直接相対 PC (0000_0000H～ FFFF_FFFFH)	命令コード内の 8 ビットまたは 12 ビットの符号付きデータを PC 値に加えたもので PC の値を指定します。	r8 r12
25	レジスタ間接相対 PC (0000_0000H～ FFFF_FFFFH)	命令コード内のデータで指定された汎用レジスタの内容を 16 ビットの符号付きデータとして、PC 値に加えたもので PC 値を指定します。	Rs

## 5-3 記号の説明

本章では各命令を個別に説明します。個別説明に先立ち、本章で使用する記号等の説明を行います。

<u>[]</u>	:省略可能であることを示します。
<u>下線</u>	:引数を記述する部分であることを示します。 イミディエイトデータ、メモリアドレス、ラベル等を記述します。
# <u>immD</u>	:最初の“#”は、引数がイミディエイトデータであることを明示するキーワードです。“#”に続く“immD”には、イミディエイトデータを記述します。また、“D”は、指定可能なイミディエイトデータのビット長を示しています。 なお、“D”の値の範囲は命令毎に異なります。
<u>Rd</u>	:最初の“R”は、引数が汎用レジスタであることを明示するキーワードです。 “R”に続く“d”には、汎用レジスタの番号を記述します。なお、“d”の値の範囲は命令毎に異なります。 (記述例 R0, r0, R7, r7, R8, r8, R13, r13)
<u>Rs</u>	:表記の基本スタイルは、上記 Rd と同じです。 本マニュアルでは、転送元を Rs と表記し、転送先を Rd と表記します。
<u>Rx</u>	:PSWの bit12～15 (N0～N3)の値によって間接指定される汎用レジスタを示します。コーディングの際は、そのまま Rx と記述してください。
<u>m16</u>	:操作対象のアドレスを示します。RAM を操作対象とする場合の m16 の範囲は、00H～FFH (0000H～00FFH)になります。また、SFRを操作対象とする場合の m16 の範囲は、7F00H～7FFFH になります。
<u>()</u>	:操作対象の内容を示します。 例えば(m16)と表記されている場合は、指定した RAM または SFR の内容ということになります。 また、(PC)と表記されている場合は、プログラムカウンタの値ということになります。
<u>Hibyte</u>	:16ビット長のデータや汎用レジスタなどの上位 8 ビットを示します。
<u>Lobyte</u>	:16ビット長のデータや汎用レジスタなどの下位 8 ビットを示します。
<u>++</u>	:ポストインクリメント(事後に+1の処理を行う)
<u>——</u>	:プリデクリメント(事前に-1の処理を行う)
<u>PC</u>	:プログラムカウンタ。
<u>SP</u>	:スタックポインタ。
<u>PSW</u>	:プログラムステータスワード。
<u>CY</u>	:ビット15からのキャリー・ボローフラグ。
<u>HC</u>	:ビット3からのキャリー・ボローフラグ
<u>OV</u>	:オーバーフローフラグ。
<u>Z8</u>	:下位8ビットのゼロフラグ。
<u>Z16</u>	:データのゼロフラグ。
<u>P</u>	:パリティフラグ。
<u>S</u>	:符号(サイン)フラグ。
<u>&amp;</u>	:論理AND演算。
<u> </u>	:論理OR演算。
<u>^</u>	:論理EOR(排他的OR)演算。



## 5-4 各命令の個別説明

## ADC Rd, #imm4

命令コード	[0 1 0 1 0 0 1 1][i3i2i1i0d3d2d1d0]	5300H
引数	Rd = 4bit(R select), imm4 = 4bit(immediate data)	
ワード数	1	
サイクル数	1	
機能	(Rd) ← (Rd) + #imm4 + CY, (PC) ← (PC)+2	
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0～N3	

## 【説明】

Rdで指定する汎用レジスタの内容と、imm4で指定するイミディエイトデータと、キャリーフラグ(CY)を加算し、その結果をRdに転送します。

なお、Rdで指定できる範囲はR0～R15、imm4で指定できる範囲は0～Fhになっています。

## 【例】

	R0	R1	R2	R3	N3～ 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

## 命令

### ADC Rd, #imm16

命令コード	0 0 1 1 0 0 0 1][0 1 0 1 d3d2d1d0][i15 ~ i8][i7 ~ i0]	3150H
引数	Rd = 4bit(R select), imm16 = 16bit(immediate data)	
ワード数	2	
サイクル数	2	
機能	(Rd) ← (Rd) + #imm16 + CY, (PC) ← (PC)+4	
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0～N3	

#### 【説明】

Rdで指定する汎用レジスタの内容と、imm16で指定するイミディエイトデータと、キャリーフラグ(CY)を加算し、その結果をRdに転送します。

なお、Rdで指定できる範囲はR0～R15、imm16で指定できる範囲は0～FFFFhになっています。

#### 【例】

	R0	R1	R2	R3	N3～ 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  $R_d$ ,  $R_s$ 

命令コード	[0 1 0 0 1 0 1 1][s3s2s1s0d3d2d1d0]	4B00H
引数	$R_d = 4\text{bit}(R \text{ select}), R_s = 4\text{bit}(R \text{ select})$	
ワード数	1	
サイクル数	1	
機能	$(R_d) \leftarrow (R_d) + (R_s) + CY, (PC) \leftarrow (PC) + 2$	
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0~N3	

## 【説明】

$R_d$ で指定する汎用レジスタの内容と、 $R_s$ で指定する汎用レジスタの内容と、キャリーフラグ(CY)を加算し、その結果を $R_d$ に転送します。

なお、 $R_d$ で指定できる範囲はR0~R15、 $R_s$ で指定できる範囲はR0~R15になっています。

## 【例】

		R0	R1	R2	R3	N3~ 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

## 命令

### ADC Rx, #imm8

命令コード	[0 1 0 1 1 0 1 1][i7i6i5i4i3i2i1i0]5B00H
引数	imm8 = 8bit(immediate data)
ワード数	1
サイクル数	1
機能	$(R_x) \leftarrow (R_x) + \#imm8 + CY, (PC) \leftarrow (PC) + 2$
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S

#### 【説明】

PSW の bit12～15 (N0～N3)の値によって間接指定される汎用レジスタ(Rx)の内容と、imm8で指定するイミディエイトデータと、キャリーフラグ(CY)を加算し、その結果をRxに転送します。

なお、imm8で指定できる範囲は0～FFになっています。

#### 【例】

	R0	R1	R2	R3	N3～ 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

命令コード	[0 1 0 1 0 0 0 1][i3i2i1i0d3d2d1d0] 5100H
引数	Rd = 4bit(R select), imm4 = 4bit(immediate data)
ワード数	1
サイクル数	1
機能	(Rd) ← (Rd) + #imm4, (PC) ← (PC) + 2
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0～N3

## 【説明】

Rdで指定する汎用レジスタの内容と、imm4で指定するイミディエイトデータを加算し、その結果を、Rdに転送します。

なお、Rdで指定できる範囲はR0～R15、imm4で指定できる範囲は0～Fhになっています。

## 【例】

	R0	R1	R2	R3	N3～ 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

## 命令

### ADD Rd, #imm16

命令コード	[0 0 1 1 0 0 0 1][0 1 0 0 d3d2d1d0][i15 ~ i8][i7 ~ i0]	3140H
引数	Rd = 4bit(R select), imm16 = 16bit(immediate data)	
ワード数	2	
サイクル数	2	
機能	(Rd) ← (Rd) + #imm16, (PC) ← (PC)+4	
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0～N3	

#### 【説明】

Rdで指定する汎用レジスタの内容と、imm16で指定するイミディエイトデータを加算し、その結果を、Rdに転送します。

なお、Rdで指定できる範囲はR0～R15、imm16で指定できる範囲は0～FFFFhになっています。

#### 【例】

	R0	R1	R2	R3	N3～ 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

命令コード	[0 1 0 0 1 0 0 1][s3s2s1s0d3d2d1d0] 4900H
引数	Rd = 4bit(R select), Rs = 4bit(R select)
ワード数	1
サイクル数	1
機能	(Rd) ← (Rd) + (Rs), (PC) ← (PC)+2
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0～N3

## 【説明】

Rdで指定する汎用レジスタの内容と、Rsで指定する汎用レジスタの内容を加算し、その結果をRdに転送します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15になっています。

## 【例】

		R0	R1	R2	R3	N3～ 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

## 命 令

### ADD Rx, #imm8

命令コード	[0 1 0 1 1 0 0 1][i7i6i5i4i3i2i1i0]	5900H
引数	imm8 = 8bit(immediate data)	
ワード数	1	
サイクル数	1	
機能	(Rx) ← (Rx) + #imm8, (PC) ← (PC)+2	
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S	

#### 【説明】

PSW の bit12～15 (N0～N3)の値によって間接指定される汎用レジスタ(Rx)の内容と、imm8で指定するイミディエイトデータを加算し、その結果をRxに転送します。

なお、imm8で指定できる範囲は0～FFhになっています。

#### 【例】

	R0	R1	R2	R3	N3～ 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

命令コード	[0 0 1 1 0 0 0 1][0 0 0 0 d3d2d1d0][i15 ~ i8][i7 ~ i0]	3100H
引数	Rd = 4bit(R select), imm16 = 16bit(immediate data)	
ワード数	2	
サイクル数	2	
機能	(Rd) ← (Rd) & #imm16, (PC) ← (PC)+4	
影響を受けるフラグ	Z8,Z16,P,S,N0～N3	

## 【説明】

Rdで指定する汎用レジスタの内容と、imm16で指定するイミディエイトデータの論理 AND 演算を行い、その結果をRdに転送します。

なお、Rdで指定できる範囲はR0～R15、imm16で指定できる範囲は0～FFFFhになっています。

## 【例】

	R0	R1	R2	R3	N3～ 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

## 命令

### AND Rd, Rs

命令コード	[0 1 0 0 0 0 0][s3s2s1s0d3d2d1d0]4000H
引数	Rd = 4bit(R select), Rs = 4bit(R select)
ワード数	1
サイクル数	1
機能	(Rd) ← (Rd) & (Rs), (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容と、Rsで指定する汎用レジスタの内容の論理 AND 演算を行い、その結果をRdに転送します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15になっています。

#### 【例】

	R0	R1	R2	R3	N3～ 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

命令コード	[0 1 0 0 0 0 1][i7i6i5i4i3i2i1i0]	4100H
引数	imm8 = 8bit(immediate data)	
ワード数	1	
サイクル数	1	
機能	(Rx) ← (Rx) & 16bit data(Hibyte=00H, Lobyte=#imm8), (PC) ← (PC)+2	
影響を受けるフラグ	Z8,Z16,P,S	

## 【説明】

PSW の bit12～15 (N0～N3) の値によって間接指定される汎用レジスタ(Rx)の内容と、16ビットデータ(上位 8 ビットは 00H, 下位 8 ビットは #imm8)との間で論理 AND 演算を行い、その結果を Rx に転送します。  
 なお、imm8 で指定できる範囲は 0～FFh になっています。

## 【例】

	R0	R1	R2	R3	N3～ 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

## 命 令

### ASR Rd, #imm4

命令コード	[0 0 1 1 0 1 1 1][i3i2i1i0d3d2d1d0]3700H
引数	Rd = 4bit(R select), imm4 = 4bit(immediate data)
ワード数	1
サイクル数	1
機能	(Rd) ← (Rd) arithmetic shift right #imm4 bit (CY) ← last shift bit, (PC) ← (PC)+2
影響を受けるフラグ	Z8,Z16,CY,P,S,N0～N3

#### 【説明】

imm4で指定するイミディエイトデータが示す値(算術シフト量)で、Rdで指定する汎用レジスタの内容を、右に算術シフトします。そして、最後にLSBからの溢れビットをキャリーフラグ(CY)に転送します。

なお、Rdで指定できる範囲はR0～R15、imm4で指定できる範囲は0～Fhになっています。

#### 【例】

		R0	R1	R2	R3	N3～ 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

#### <注意事項>

算術シフトは、Rdの内容のMSBを符号ビットとして、MSBは変わらず、シフトする度にMSBの値を1bit右にコピーします。

ASR Rd, Rs

命令コード	[0 0 1 1 0 1 1 0 ][s3s2s1s0d3d2d1d0]	3600H
引数	Rd = 4bit(R select), Rs = 4bit(R select)	
ワード数	1	
サイクル数	1	
機能	(Rd) ← (Rd) arithmetic shift right (Lower 4bit value of Rs) bit (CY) ← last shift bit, (PC) ← (PC)+2	
影響を受けるフラグ	Z8,Z16,CY,P,S,N0～N3	

## 【説明】

Rsで指定する汎用レジスタの内容の下位4bitが示す値(算術シフト量)で、Rdで指定する汎用レジスタの内容を右に算術シフトします。そして、最後にLSBからの溢れビットをキャリーフラグ(CY)に転送します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15になっています。

## 【例】

		R0	R1	R2	R3	N3～ 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

## &lt;注意事項&gt;

算術シフトは、Rdの内容のMSBを符号ビットとして、MSBは変わらず、シフトする度にMSBの値を1bit右にコピーします。

## 命令

### BC r8

命令コード	[1 1 0 1 0 0 1 1][r7r6r5r4r3r2r1r0]	D300H
引数	r8 = 8bit(relative address, signed)	
ワード数	1	
サイクル数	2 or 3	
機能	If CY=1, then (PC) ← (PC)+2±(r8) If CY=0, then (PC) ← (PC)+2	
影響を受けるフラグ		

#### 【説明】

キャリーフラグ(CY)が1ならば、r8で指定する相対アドレス+2を、プログラムカウンタ(PC)に加算し、その結果をPCに転送します。CYが0ならば、PCに2を加算します。

なお、r8で指定する相対アドレスが指定できる範囲は、符号付き8bitデータ(−128～127)になっています。

#### 【例】

```
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

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。

## BC Rd, #imm8, r12

命令コード	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][0 0 1 1 r11 ~ r8][r7 ~ r0] 20003000H
引数	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8, Z16, CY, HC, OV, P, S, N0 ~ N3

## 【説明】

Rdで指定する汎用レジスタの内容と、imm8で指定するイミディエイトデータの引き算結果が、負ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。引き算結果が負でなければ、PCに4を加算します。

なお、Rdで指定できる範囲はR0~R7、imm8で指定できる範囲は0~FFh、r12で指定する相対アドレスが指定できる範囲は、符号付き12bitデータ(−2048~2047)になっています。

## 【例】

```

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

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命令

### BC Rd, Rs, r12

命令コード	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][0 0 1 1 r11 ~ r8][r7 ~ r0] 0D003000H
引数	Rd = 4bit(R select),Rs = 4bit(R select),r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容と、Rsで指定する汎用レジスタの内容の引き算結果が、負ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。引き算結果が負でなければ、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15、r12で指定する相対アドレスが指定できる範囲は、符号付12bitデータ(−2048～2047)になっています。

#### 【例】

	PC	R0	R1	R2	R3	PSW
	-	-	-	-	-	-
MOV.W R0,#0x5678	9004h	5678h	-	-	-	0000h
MOV.W R1,#0x1234	9008h	5678h	1234h	-	-	1020h
MOV.W R2,#0x5678	900Ch	5678h	1234h	5678h	-	2000h
MOV.W R3,#0xFFFF	9010h	5678h	1234h	5678h	FFFFh	3040h
loop:						
BC R0,R2,LA ;; NOT JUMP LA	9014h	5678h	1234h	5678h	FFFFh	0003h
BC R1,R2,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

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。



## BC Rx, #imm16, r8

命令コード	[1 1 0 0 0 1 1][r7r6r5r4r3r2r1r0][i15 ~ i8][i7 ~ i0] C3000000H
引数	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)
ワード数	2
サイクル数	3 or 4
機能	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
影響を受けるフラグ	Z8, Z16, CY, HC, OV, P, S

## 【説明】

PSW の bit12～15 (N0～N3) の値によって間接指定される汎用レジスタ(Rx) の内容と、imm16で指定するイミディエイトデータの引き算結果が、負ならば、r8で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。引き算結果が負でなければ、PCに4を加算します。

なお、imm16で指定できる範囲は0～FFFFh、r8で指定する相対アドレスが指定できる範囲は、符号付8bitデータ(－128～127)になっています。

## 【例】

```

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

```

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

## &lt;注意事項&gt;

条件が一致した場合、この命令は4サイクルとなります。

## 命令

### BGE r8

命令コード	[1 1 0 1 0 0 0 0][r7r6r5r4r3r2r1r0]	D000H
引数	r8 = 8bit(relative address,signed)	
ワード数	1	
サイクル数	2 or 3	
機能	If S ^ OV = 0, then (PC) ← (PC) + 2 ± (r8) If S ^ OV = 1, then (PC) ← (PC) + 2	
影響を受けるフラグ		

#### 【説明】

サインフラグ(S)とオーバーフローフラグ(OV)で排他的論理和をした結果が0ならば、r8で指定する相対アドレス+2をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算結果が1ならば、PCに2を加算します。

なお、r8で指定する相対アドレスが指定できる範囲は、符号付8bitデータ(−128～127)になっています。

#### 【例】

```
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

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。

## BGE Rd, #imm8, r12

命令コード	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][0 0 0 0 r11 ~ r8][r7 ~ r0] 20000000H
引数	Rd = 3bit(R select),imm8 = 8bit(immediate data) r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	If result of signed comparison is (Rd) $\geq$ #imm8, then (PC) $\leftarrow$ (PC)+4 $\pm$ (r12) If result of signed comparison is (Rd) < #imm8, then (PC) $\leftarrow$ (PC)+4
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0~N3

## 【説明】

Rdで指定する汎用レジスタの内容(符号付き16bitデータ)とimm8で指定するイミディエイトデータの引き算結果が負でなければ、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。引き算結果が負ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0~R7、imm8で指定できる範囲は0~FFh、r12で指定する相対アドレスが指定できる範囲は、符号付12bitデータ(−2048~2047)になっています。

## 【例】

```

MOV.W R0,#0xCDEF
MOV.W R1,#0x789A
MOV.W R2,#0x1234
MOV.W R3,#0xFFFF
loop:
  BGE R0,#0x12,LA ;; NOT JUMP LA
  BGE R1,#0x12,LB ;; JUMP LB
  BR  loop
LA:
  DEC R3
  BR  loop
LB:
  INC R3
  NOP

```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9004h	CDEFh	-	-	-	0040h
9008h	CDEFh	789Ah	-	-	1000h
900Ch	CDEFh	789Ah	1234h	-	2020h
9010h	CDEFh	789Ah	1234h	FFFFh	3040h
9014h	CDEFh	789Ah	1234h	FFFFh	0060h
901Eh	CDEFh	789Ah	1234h	FFFFh	1000h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	CDEFh	789Ah	1234h	0000h	3003h
9022h	CDEFh	789Ah	1234h	0000h	3003h

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命令

### BGE Rd, Rs, r12

命令コード	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][0 0 0 0 r11 ~ r8][r7 ~ r0] 0D000000H
引数	Rd = 4bit(R select),Rs = 4bit(R select),r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	If result of signed comparison is (Rd) $\geq$ (Rs), then (PC) $\leftarrow$ (PC)+4 $\pm$ (r12) If result of signed comparison is (Rd) < (Rs), then (PC) $\leftarrow$ (PC)+4
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0~N3

#### 【説明】

Rdで指定する汎用レジスタの内容(符号付き16bitデータ)と、Rsで指定する汎用レジスタの内容(符号付き16bitデータ)の引き算結果が負でなければ、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。引き算結果が負ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0~R15、Rsで指定できる範囲はR0~R15、r12で指定する相対アドレスが指定できる範囲は、符号付12bitデータ(−2048~2047)になっています。

#### 【例】

```

MOV.W R0,#0xCDEF
MOV.W R1,#0x789A
MOV.W R2,#0x1234
MOV.W R3,#0xFFFF
loop:
    BGE    R0,R2 ,LA    ;; NOT JUMP LA
    BGE    R1,R2 ,LB    ;; JUMP LB
    BR     loop
LA:
    DEC    R3
    BR     loop
LB:
    INC    R3
    NOP

```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9004h	CDEFh	-	-	-	0040h
9008h	CDEFh	789Ah	-	-	1000h
900Ch	CDEFh	789Ah	1234h	-	2020h
9010h	CDEFh	789Ah	1234h	FFFFh	3040h
9014h	CDEFh	789Ah	1234h	FFFFh	0040h
901Eh	CDEFh	789Ah	1234h	FFFFh	1000h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	CDEFh	789Ah	1234h	0000h	3003h
9022h	CDEFh	789Ah	1234h	0000h	3003h

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。

## BGE Rx, #imm16, r8

命令コード	[1 1 0 0 0 0 0 0][r7r6r5r4r3r2r1r0][i15 ~ i8][i7 ~ i0]	C0000000H
引数	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)	
ワード数	2	
サイクル数	3 or 4	
機能	If result of signed comparison is (Rx) $\geq$ #imm16, then (PC) $\leftarrow$ (PC)+4 $\pm$ (r8) If result of signed comparison is (Rx) < #imm16, then (PC) $\leftarrow$ (PC)+4	
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S	

## 【説明】

PSW の bit12～15 (N0～N3)の値によって間接指定される汎用レジスタ(Rx)の内容(符号付き16bitデータ)と、imm16で指定するイミディエイトデータ(符号付き16bitデータ)の引き算結果が負でなければ、r8で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。引き算結果が負ならば、PCに4を加算します。

なお、imm16で指定できる範囲は0～FFFFh、r8で指定する相対アドレスが指定できる範囲は、符号付き8bitデータ(－128～127)になっています。

## 【例】

```

MOV.W R2,#0x1234
MOV.W R3,#0xFFFF
loop:
MOV.W R0,#0xCDEF
BGE Rx,#0x1234,LA ;; NOT JUMP LA
MOV.W R1,#0x789A
BGE 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	CDEFh	-	1234h	FFFFh	0040h
9010h	CDEFh	-	1234h	FFFFh	0040h
9014h	CDEFh	789Ah	1234h	FFFFh	1000h
901Eh	CDEFh	789Ah	1234h	FFFFh	1000h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	CDEFh	789Ah	1234h	0000h	3003h
9022h	CDEFh	789Ah	1234h	0000h	3003h

## &lt;注意事項&gt;

条件が一致した場合、この命令は4サイクルとなります。

## 命 令

### BGT r8

命令コード	[1 1 0 1 0 1 0 0][r7r6r5r4r3r2r1r0]	D400H
引数	r8 = 8bit(relative address,signed)	
ワード数	1	
サイクル数	2 or 3	
機能	If S ^ OV   Z16 = 0, then (PC)←(PC)+2±(r8) If S ^ OV   Z16 = 1, then (PC)←(PC)+2	
影響を受けるフラグ		

#### 【説明】

サインフラグ(S)とオーバーフローフラグ(OV)で排他的論理和をとり、その結果と16bit演算フラグ(Z16)の論理和をした結果が0ならば、r8で指定する相対アドレス+2をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算結果が1ならば、PCに2を加算します。

なお、r8で指定する相対アドレスが指定できる範囲は、符号付き8bitデータ(−128～127)になっています。

#### 【例】

```
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

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。

## BGT Rd, #imm8, r12

命令コード	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][0 1 0 0 r11 ~ r8][r7 ~ r0] 20004000H
引数	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8, Z16, CY, HC, OV, P, S, N0 ~ N3

## 【説明】

Rdで指定する汎用レジスタの内容(符号付き16bitデータ)と、imm8で指定するイミディエイトデータの引き算結果が正ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。引き算結果が正でなければ、PCに4を加算します。

なお、Rdで指定できる範囲はR0~R7、imm8で指定できる範囲は0~FFh、r12で指定する相対アドレスが指定できる範囲は、符号付き12bitデータ(−2048~2047)になっています。

## 【例】

```

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

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命令

### BGT Rd, Rs, r12

命令コード	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][0 1 0 0 r11 ~ r8][r7 ~ r0] 0D004000H
引数	Rd = 4bit(R select),Rs = 4bit(R select),r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容(符号付き16bitデータ)と、Rsで指定する汎用レジスタの内容(符号付き16bitデータ)の引き算結果が正ならば、r12で指定した相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。引き算結果が正でなければ、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15、r12で指定する相対アドレスが指定できる範囲は、符号付き12bitデータ(−2048～2047)になっています。

#### 【例】

		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

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。



## BGT Rx, #imm16, r8

命令コード	[1 1 0 0 0 1 0 0][r7r6r5r4r3r2r1r0][i15 ~ i8][i7 ~ i0]	C4000000H
引数	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)	
ワード数	2	
サイクル数	3 or 4	
機能	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	
影響を受けるフラグ	Z8, Z16, CY, HC, OV, P, S	

## 【説明】

PSW の bit12～15 (N0～N3) の値によって間接指定される汎用レジスタ (Rx) の内容 (符号付き 16bit データ) と、imm16 で指定するイミディエイトデータ (符号付き 16bit データ) の引き算結果が正ならば、r12 で指定する相対アドレス + 4 をプログラムカウンタ (PC) に加算し、その結果を PC に転送します。引き算結果が正でなければ、PC に 4 を加算します。

なお、imm16 で指定できる範囲は 0～FFFFh、r8 で指定する相対アドレスが指定できる範囲は、符号付き 8 bit データ (−128～127) になっています。

## 【例】

```

MOV.W R2, #0x89AB
MOV.W R3, #0xFFFF
loop:
MOV.W R0, #0x89AB
BGT Rx, #0x89AB, LA ;; NOT JUMP LA
MOV.W R1, #0x789A
BGT Rx, #0x89AB, LB ;; JUMP LB
BR loop
LA:
DEC R3
BR loop
LB:
INC R3
NOP

```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9004h	-	-	89ABh	-	2040h
9008h	-	-	89ABh	FFFFh	3040h
900Ch	89ABh	-	89ABh	FFFFh	0040h
9010h	89ABh	-	89ABh	FFFFh	0003h
9014h	89ABh	789Ah	89ABh	FFFFh	1000h
901Eh	89ABh	789Ah	89ABh	FFFFh	107Ch
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	89ABh	789Ah	89ABh	0000h	301Fh
9022h	89ABh	789Ah	89ABh	0000h	301Fh

## &lt;注意事項&gt;

条件が一致した場合、この命令は 4 サイクルとなります。

## 命令

### BHI r8

命令コード	[1 1 0 1 0 1 0 1][r7r6r5r4r3r2r1r0]	D500H
引数	r8 = 8bit(relative address,signed)	
ワード数	1	
サイクル数	2 or 3	
機能	If CY   Z16 = 0, then (PC)←(PC)+2±(r8) If CY   Z16 = 1, then (PC)←(PC)+2	
影響を受けるフラグ		

#### 【説明】

キャリーフラグ(CY)と16bit演算フラグ(Z16)で論理和をした結果が0ならば、r8で指定する相対アドレス+2をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算結果が1ならば、PCに2を加算します。

なお、r8で指定する相対アドレスが指定できる範囲は、符号付き8bitデータ(−128～127)になっています。

#### 【例】

```
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

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。

## BHI Rd, #imm8, r12

命令コード	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][0 1 0 1 r11 ~ r8][r7 ~ r0] 20005000H
引数	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S, N0～N3

## 【説明】

Rdで指定する汎用レジスタの内容と、imm8で指定するイミディエイトデータの引き算結果が正ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。引き算結果が正でなければ、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R7、imm8で指定できる範囲は0～FFh、r12で指定する相対アドレスが指定できる範囲は、符号付き12bitデータ(−2048～2047)になっています。

## 【例】

```

MOV.W R0,#0x0089
MOV.W R1,#0x0098
MOV.W R2,#0x0089
MOV.W R3,#0xFFFF
loop:
  BHI    R0,#0x89,LA ;; NOT JUMP LA
  BHI    R1,#0x89,LB ;; JUMP LB
  BR     loop
LA:
  DEC    R3
  BR     loop
LB:
  INC    R3
  NOP

```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9002h	0089h	-	-	-	0020h
9004h	0089h	0098h	-	-	1020h
9006h	0089h	0098h	0089h	-	2020h
900Ah	0089h	0098h	0089h	FFFFh	3040h
900Eh	0089h	0098h	0089h	FFFFh	0003h
9018h	0089h	0098h	0089h	FFFFh	1008h
-	-	-	-	-	-
-	-	-	-	-	-
901Ah	0089h	0098h	0089h	0000h	300Bh
901Ch	0089h	0098h	0089h	0000h	300Bh

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命令

### BHI Rd, Rs, r12

命令コード	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][0 1 0 1 r11 ~ r8][r7 ~ r0] 0D005000H
引数	Rd = 4bit(R select),Rs = 4bit(R select),r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S, N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容と、Rsで指定する汎用レジスタの内容の引き算結果が正ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。引き算結果が正でなければ、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15、r12で指定する相対アドレスが指定できる範囲は、符号付き12bitデータ(－2048～2047)になっています。

#### 【例】

```

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

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。

## BHI Rx, #imm16, r8

命令コード	[1 1 0 0 0 1 0 1][r7r6r5r4r3r2r1r0][i15 ~ i8][i7 ~ i0]	C5000000H
引数	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)	
ワード数	2	
サイクル数	3 or 4	
機能	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	
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S	

## 【説明】

PSW の bit12～15 (N0～N3)の値によって間接指定される汎用レジスタ(Rx)の内容と、imm16で指定するイミディエイトデータの引き算結果が正ならば、r8で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。引き算結果が正でなければ、PCに4を加算します。

なお、imm16で指定できる範囲は0～FFFFh、r8で指定する相対アドレスが指定できる範囲は、符号付き8 bitデータ(－128～127)になっています。

## 【例】

```

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

```

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

## &lt;注意事項&gt;

条件が一致した場合、この命令は4サイクルとなります。

## 命令

### BLE r8

命令コード	[1 1 0 1 0 1 1 0][r7r6r5r4r3r2r1r0]	D600H
引数	r8 = 8bit(relative address,signed)	
ワード数	1	
サイクル数	2 or 3	
機能	If S ^ OV   Z16 =1,then (PC)←(PC)+2±(r8) If S ^ OV   Z16 =0,then (PC)←(PC)+2	
影響を受けるフラグ		

#### 【説明】

サインフラグ(S)とオーバーフローフラグ(OV)で排他的論理和をとり、その結果と16bit演算フラグ(Z16)の論理和をした結果が1ならば、r12で指定する相対アドレス+2をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算結果が0ならば、PCに2を加算します。

なお、r8で指定する相対アドレスが指定できる範囲は、符号付き8bitデータ(−128〜127)になっています。

#### 【例】

```
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

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。

## BLE Rd, #imm8, r12

命令コード	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][0 1 1 0 r11 ~ r8][r7 ~ r0] 20006000H
引数	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
ワード数	2
サイクル数	2 or 3
機能	If result of signed comparison is (Rd) $\leq$ #imm8, then (PC) $\leftarrow$ (PC) + 4 $\pm$ (r12) If result of signed comparison is (Rd) $>$ #imm8, then (PC) $\leftarrow$ (PC) + 4
影響を受けるフラグ	Z8, Z16, CY, HC, OV, P, S, N0 ~ N3

## 【説明】

Rdで指定する汎用レジスタの内容(符号付き16bitデータ)と、imm8で指定するイミディエイトデータの引き算結果が正でなければ、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。引き算結果が正ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R7、imm8で指定できる範囲は0～FFh、r12で指定する相対アドレスが指定できる範囲は、符号付き12bitデータ(−2048～2047)になっています。

## 【例】

	PC	R0	R1	R2	R3	PSW
	-	-	-	-	-	-
MOV.W R0,#0x00EF	9002h	00EFh	-	-	-	0020h
MOV.W R1,#0x0098	9004h	00EFh	0098h	-	-	1020h
MOV.W R2,#0x00CD	9006h	00EFh	0098h	00CDh	-	2020h
MOV.W R3,#0xFFFF	900Ah	00EFh	0098h	00CDh	FFFFh	3040h
loop:						
BLE R0,#0xCD,LA ;; NOT JUMP LA	900Eh	00EFh	0098h	00CDh	FFFFh	0000h
BLE R1,#0xCD,LB ;; JUMP LB	9018h	00EFh	0098h	00CDh	FFFFh	106Ch
BR loop	-	-	-	-	-	-
LA:						
DEC R3	-	-	-	-	-	-
BR loop	-	-	-	-	-	-
LB:						
INC R3	901Ah	00EFh	0098h	00CDh	0000h	300Fh
NOP	901Ch	00EFh	0098h	00CDh	0000h	300Fh

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。。

## 命令

### BLE Rd, Rs, r12

命令コード	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][0 1 1 0 r11 ~ r8][r7 ~ r0] 0D006000H
引数	Rd = 4bit(R select),Rs = 4bit(R select),r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	If result of signed comparison is (Rd) $\leq$ (Rs), then (PC) $\leftarrow$ (PC)+4 $\pm$ (r12) If result of signed comparison is (Rd) > (Rs), then (PC) $\leftarrow$ (PC)+4
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S, N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容(符号付き16bitデータ)と、Rsで指定する汎用レジスタの内容(符号付き16bitデータ)の引き算結果が正でなければ、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。引き算結果が正ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15、r12で指定する相対アドレスが指定できる範囲は、符号付き12bitデータ(－2048～2047)になっています。

#### 【例】

	PC	R0	R1	R2	R3	PSW
	-	-	-	-	-	-
MOV.W R0,#0x7654	9004h	7654h	-	-	-	0000h
MOV.W R1,#0x9876	9008h	7654h	9876h	-	-	1040h
MOV.W R2,#0xCDEF	900Ch	7654h	9876h	CDEFh	-	2040h
MOV.W R3,#0xFFFF	9010h	7654h	9876h	CDEFh	FFFFh	3040h
loop:						
BLE R0,R2,LA ;; NOT JUMP LA	9014h	7654h	9876h	CDEFh	FFFFh	007Ch
BLE R1,R2,LB ;; JUMP LB	901Eh	7654h	9876h	CDEFh	FFFFh	104Ch
BR loop	-	-	-	-	-	-
LA:						
DEC R3	-	-	-	-	-	-
BR loop	-	-	-	-	-	-
LB:						
INC R3	9020h	7654h	9876h	CDEFh	0000h	300Fh
NOP	9022h	7654h	9876h	CDEFh	0000h	300Fh

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。



## BLE Rx, #imm16, r8

命令コード	[1 1 0 0 0 1 1 0][r7r6r5r4r3r2r1r0][i15 ~ i8][i7 ~ i0]	C6000000H
引数	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)	
ワード数	2	
サイクル数	3 or 4	
機能	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	
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S	

## 【説明】

PSW の bit12～15 (N0～N3)の値によって間接指定される汎用レジスタ(Rx)の内容(符号付き16bitデータ)と、imm16で指定するイミディエイトデータ(符号付き16bitデータ)の引き算結果が正でなければ、r8で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。引き算結果が正ならば、PCに4を加算します。

なお、imm16で指定できる範囲は符号付16bitデータ(−32768～32767)、r8で指定する相対アドレスが指定できる範囲は、符号付き8bitデータ(−128～127)になっています。

## 【例】

```

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

## &lt;注意事項&gt;

条件が一致した場合、この命令は4サイクルとなります。

## 命令

### BLS r8

命令コード	[1 1 0 1 0 1 1 1][r7r6r5r4r3r2r1r0]	D700H
引数	r8 = 8bit(relative address,signed)	
ワード数	1	
サイクル数	2 or 3	
機能	If CY   Z16 = 1, then (PC)←(PC)+2±(r8) If CY   Z16 = 0, then (PC)←(PC)+2	
影響を受けるフラグ		

#### 【説明】

キャリーフラグ(CY)と16bit演算フラグ(Z16)で論理和をした結果が1ならば、r8で指定する相対アドレス+2をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算結果が0ならば、PCに2を加算します。

なお、r8で指定する相対アドレスが指定できる範囲は、符号付き8bitデータ(−128～127)になっています。

#### 【例】

```
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

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。

## BLS Rd, #imm8, r12

命令コード	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][0 1 1 1 r11 ~ r8][r7 ~ r0] 20007000H
引数	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
ワード数	2
サイクル数	2 or 3
機能	If result of unsigned comparison is (Rd) $\leq$ #imm8, then (PC) $\leftarrow$ (PC)+4 $\pm$ (r12) If result of unsigned comparison is (Rd) $>$ #imm8, then (PC) $\leftarrow$ (PC)+4
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0~N3

## 【説明】

Rdで指定する汎用レジスタの内容と、imm8で指定するイミディエイトデータの引き算結果が正でなければ、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。引き算結果が正ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0~R7、imm8で指定できる範囲は0~FFh、r12で指定する相対アドレスが指定できる範囲は符号付12bitデータ(−2048~2047)になっています。

## 【例】

```

MOV.W  R0,#0x00FE
MOV.W  R1,#0x0098
MOV.W  R2,#0x00CD
MOV.W  R3,#0xFFFF
loop:
  BLS    R0,#0xCD,LA  ;; NOT JUMP LA
  BLS    R1,#0xCD,LB  ;; JUMP LB
  BR     loop
LA:
  DEC    R3
  BR     loop
LB:
  INC    R3
  NOP

```

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

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命令

### BLS Rd, Rs, r12

命令コード	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][0 1 1 1 r11 ~ r8][r7 ~ r0] 0D007000H
引数	Rd = 4bit(R select),Rs = 4bit(R select),r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	If result of unsigned comparison is (Rd) $\leq$ (Rs), then (PC) $\leftarrow$ (PC)+4 $\pm$ (r12) If result of unsigned comparison is (Rd) > (Rs), then (PC) $\leftarrow$ (PC)+4
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S, N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容と、Rsで指定する汎用レジスタの内容の引き算結果が正でなければ、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。引き算結果が正ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15、r12で指定する相対アドレスが指定できる範囲は、符号付き12bitデータ(－2048～2047)になっています。

#### 【例】

```

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

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。

## BLS Rx, #imm16, r8

命令コード	[1 1 0 0 0 1 1 1][r7r6r5r4r3r2r1r0][i15 ~ i8][i7 ~ i0]	C7000000H
引数	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)	
ワード数	2	
サイクル数	3 or 4	
機能	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	
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S	

## 【説明】

PSW の bit12～15 (N0～N3)の値によって間接指定される汎用レジスタ(Rx)の内容と、imm16で指定するイミディエイトデータの引き算結果が正でなければ、r8で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。引き算結果が正ならば、PCに4を加算します。  
 なお、imm16で指定できる範囲は0～FFFFh、r8で指定する相対アドレスが指定できる範囲は、符号付き8bitデータ(−128～127)になっています。

## 【例】

```

MOV.W  R2,#0xCDEF
MOV.W  R3,#0xFFFF
loop:
MOV.W  R0,#0xFEDC
BLS     Rx,#0xCDEF,LA ;; NOT JUMP LA
MOV.W  R1,#0x9876
BLS     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	FEDCh	-	CDEFh	FFFFh	0040h
9010h	FEDCh	-	CDEFh	FFFFh	0008h
9014h	FEDCh	9876h	CDEFh	FFFFh	1048h
901Eh	FEDCh	9876h	CDEFh	FFFFh	104Ch
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	FEDCh	9876h	CDEFh	0000h	300Fh
9022h	FEDCh	9876h	CDEFh	0000h	300Fh

## &lt;注意事項&gt;

条件が一致した場合、この命令は4サイクルとなります。

## 命 令

### BLT    r8

命令コード	[1 1 0 1 0 0 1 0][r7r6r5r4r3r2r1r0]	D200H
引数	r8 = 8bit(relative address,signed)	
ワード数	1	
サイクル数	2 or 3	
機能	If S ^ OV = 1, then (PC)←(PC)+2±(r8) If S ^ OV = 0, then (PC)←(PC)+2	
影響を受けるフラグ		

#### 【説明】

サインフラグ(S)とオーバーフローフラグ(OV)で排他的論理和をした結果が1ならば、r8で指定する相対アドレス+2をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算結果が0ならば、PCに2を加算します。

なお、r8で指定する相対アドレスが指定できる範囲は、符号付き8bitデータ(−128～127)になっています。

#### 【例】

```
                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

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。

BLT Rd, #imm8, r12

命令コード	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][0 0 1 0 r11 ~ r8][r7 ~ r0] 20002000H
引数	Rd = 3bit(R select),imm8 = 8bit(immediate data) r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0～N3

## 【説明】

Rdで指定する汎用レジスタの内容(符号付き16bitデータ)と、imm8で指定するイミディエイトデータの引き算結果が負ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。引き算結果が負でなければ、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R7、imm8で指定できる範囲は0～FFh、r12で指定する相対アドレスが指定できる範囲は、符号付き12bitデータ(－2048～2047)になっています。

## 【例】

```

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

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命令

### BLT Rd, Rs, r12

命令コード	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][0 0 1 0 r11 ~ r8][r7 ~ r0] 0D002000H
引数	Rd = 4bit(R select),Rs = 4bit(R select),r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容(符号付き16bitデータ)と、Rsで指定する汎用レジスタの内容(符号付き16bitデータ)の引き算結果が負ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。引き算結果が負でなければ、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15、r12で指定する相対アドレスが指定できる範囲は、符号付き12bitデータ(－2048～2047)になっています。

#### 【例】

```

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

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。



## BLT Rx, #imm16, r8

命令コード	[1 1 0 0 0 1 0][r7r6r5r4r3r2r1r0][i15 ~ i8][i7 ~ i0]	C2000000H
引数	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)	
ワード数	2	
サイクル数	3 or 4	
機能	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	
影響を受けるフラグ	Z8, Z16, CY, HC, OV, P, S	

## 【説明】

PSW の bit12～15 (N0～N3) の値によって間接指定される汎用レジスタ(Rx)の内容(符号付き16bitデータ)と、imm16で指定するイミディエイトデータ(符号付き16bitデータ)の引き算結果が負ならば、r8で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。引き算結果が負でなければ、PCに4を加算します。

なお、imm16で指定できる範囲は0～FFFFh、r8で指定する相対アドレスが指定できる範囲は、符号付き8bitデータ(－128～127)になっています。

## 【例】

```

MOV.W  R2, #0xCDEF
MOV.W  R3, #0xFFFF
loop:
MOV.W  R0, #0x5678
BLT    Rx, #0x5678, LA  ;; NOT JUMP LA
MOV.W  R1, #0xCDEF
BLT    Rx, #0x5678, 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	5678h	-	CDEFh	FFFFh	0000h
9010h	5678h	-	CDEFh	FFFFh	0003h
9014h	5678h	CDEFh	CDEFh	FFFFh	1040h
901Eh	5678h	CDEFh	CDEFh	FFFFh	1010h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	5678h	CDEFh	CDEFh	0000h	3013h
9022h	5678h	CDEFh	CDEFh	0000h	3013h

## &lt;注意事項&gt;

条件が一致した場合、この命令は4サイクルとなります。

## 命令

### BMI r8

命令コード	[1 1 0 1 1 0 1 0][r7r6r5r4r3r2r1r0]	DA00H
引数	r8 = 8bit(relative address,signed)	
ワード数	1	
サイクル数	2 or 3	
機能	If S = 1,then (PC)←(PC)+2±(r8) If S = 0,then (PC)←(PC)+2	
影響を受けるフラグ		

#### 【説明】

サインフラグ(S)が1ならば、r8で指定する相対アドレス+2をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。Sが0ならば、PCに2を加算します。

なお、r8で指定する相対アドレスが指定できる範囲は、符号付き8bitデータ(−128～127)になっています。

#### 【例】

```
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

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。

## BMI Rd, #imm8, r12

命令コード	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][1 0 1 0 r11 ~ r8][r7 ~ r0] 2000A000H
引数	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0～N3

## 【説明】

Rdで指定する汎用レジスタの内容と、imm8で指定するイミディエイトデータの引き算結果でサインフラグ(S)が1ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算の結果、Sが0ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R7、imm8で指定できる範囲は0～FFh、r12で指定する相対アドレスが指定できる範囲は、符号付き12bitデータ(−2048～2047)になっています。

## 【例】

	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

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命令

### BMI Rd, Rs, r12

命令コード	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][1 0 1 0 r11 ~ r8][r7 ~ r0] 0D00A000H
引数	Rd = 4bit(R select),Rs = 4bit(R select),r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容と、Rsで指定する汎用レジスタの内容の引き算結果でサインフラグ(S)が1ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算の結果、Sが0ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15、r12で指定する相対アドレスが指定できる範囲は、符号付き12bitデータ(－2048～2047)になっています。

#### 【例】

```
MOV.W R0,#0x9876
MOV.W R1,#0x5678
MOV.W R2,#0x1234
MOV.W R3,#0xFFFF
loop:
    BMI    R1,R2,LA    ;; NOT JUMP LA
    BMI    R0,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	1000h
901Eh	9876h	5678h	1234h	FFFFh	0060h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	9876h	5678h	1234h	0000h	3003h
9022h	9876h	5678h	1234h	0000h	3003h

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。

## BMI Rx, #imm16, r8

命令コード	[1 1 0 0 1 0 1 0][r7r6r5r4r3r2r1r0][i15 ~ i8][i7 ~ i0] CA000000H
引数	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)
ワード数	2
サイクル数	3 or 4
機能	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
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S

## 【説明】

PSW の bit12～15 (N0～N3)の値によって間接指定される汎用レジスタ(Rx)の内容と、imm16で指定するイミディエイトデータの引き算結果でサインフラグ(S)が1ならば、r8で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算の結果、Sが0ならば、PCに4を加算します。なお、imm16で指定できる範囲は0～FFFFh、なお、r8で指定する相対アドレスが指定できる範囲は、符号付き8bitデータ(−128～127)になっています。

## 【例】

```

MOV.W  R2,#0x1234
MOV.W  R3,#0xFFFF
loop:
MOV.W  R1,#0x5678
BMI    Rx,#0x1234,LA  ;; NOT JUMP LA
MOV.W  R0,#0x9876
BMI    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	9876h	5678h	1234h	FFFFh	0040h
901Eh	9876h	5678h	1234h	FFFFh	0060h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	9876h	5678h	1234h	0000h	3003h
9022h	9876h	5678h	1234h	0000h	3003h

## &lt; 注意事項 &gt;

条件が一致した場合、この命令は4サイクルとなります。

## 命令

### BN m16, #imm3, r12

命令コード	[0 1 1 1 1 X 0][m7m6m5m4m3m2m1m0][0 i2i1i0 r11 ~ r8][r7 ~ r0] 7C00H(RAM), 7E00H(SFR)
引数	m16 = 16bit(Lower 8bit valid for operation code), imm3 = 3bit(bit select) r12 = 12bit(relative address, signed)
ワード数	2
サイクル数	3 or 4
機能	If (m16) of bit #imm3 = 0, then (PC) ← (PC) + 4 ± (r12) If (m16) of bit #imm3 = 1, then (PC) ← (PC) + 4
影響を受けるフラグ	

#### 【説明】

m16によって指定するRAM(データメモリ)またはSFR(内蔵する周辺機能を制御する専用のレジスタ群)の内容のimm3で指定するイミディエイトデータが示すbitが0の場合、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。m16の内容の指定されるbitが1の場合、PCに4を加算します。

m16(第1オペランドデータ)の値によって、コンパイラは転送先がRAMであるのかSFRであるのかを判断して命令コードを生成します。

- RAMを指定する場合には、m16の値は00H~FFH(0000H~00FFH)の範囲で指定してください。この命令で100H番地以上のRAMを指定することはできません。

- SFRを指定する場合には、m16の値は7F00H~7FFFHの範囲で指定してください。なお、生成される命令コードの基本系は、それぞれ7C00H(RAM), 7E00H(SFR)になり、命令コードの動作に反映されるのは、m16の下位8ビットになります。

なお、imm3で指定できる範囲は0~8h、r12で指定する相対アドレスが指定できる範囲は符号付12bitデータ(-2048~2047)になっています。

#### 【例】

```

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

#### <注意事項>

条件が一致した場合、この命令は4サイクルとなります。

BN Rd, #imm4, r12

命令コード	[0 0 0 0 1 0 0][i3i2i1i0d3d2d1d0][0 0 0 0 r11 ~ r8][r7 ~ r0] 0400H
引数	Rd = 4bit(R select),imm4 = 4bit(bit select),r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	If (Rd) of bit #imm4 = 0, then (PC)←(PC)+4±(r12) If (Rd) of bit #imm4 = 1, then (PC)←(PC)+4
影響を受けるフラグ	N0～N3

## 【説明】

Rdで指定する汎用レジスタの内容のimm4で指定するイミディエイトデータが示すbitが0の場合、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。Rdの内容の指定されるbitが1の場合、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R15、imm4で指定できる範囲は0～Fh、r12で指定する相対アドレスが指定できる範囲は符号付12bitデータ(−2048～2047)になっています。

## 【例】

```

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

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命令

### BN Rd, Rs, r12

命令コード	[0 0 0 0 1 1 0][s3s2s1s0d3d2d1d0][0 0 0 0 r11 ~ r8][r7 ~ r0] 0600H
引数	Rd = 4bit(R select),Rs = 4bit(bit select),r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	If (Rd) of bit (Rs)&000Fh =0, then (PC)←(PC)+4±(r12) If (Rd) of bit (Rs)&000Fh =1, then (PC)←(PC)+4
影響を受けるフラグ	N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容で、Rsで指定する汎用レジスタの内容の下位4bitが示すbitが0の場合、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。Rdの内容の指定されるbitが1の場合、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15、r12で指定する相対アドレスが指定できる範囲は符号付12bitデータ(−2048～2047)になっています。

#### 【例】

```

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

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。



BNC r8

命令コード	[1 1 0 1 0 0 0 1][r7r6r5r4r3r2r1r0]	D100H
引数	r8 = 8bit(relative address,signed)	
ワード数	1	
サイクル数	2 or 3	
機能	If CY = 0, then (PC)←(PC)+2±(r8) If CY = 1, then (PC)←(PC)+2	
影響を受けるフラグ		

## 【説明】

キャリーフラグ(CY)が0ならば、r8で指定する相対アドレス+2をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。CYが1ならば、PCに2を加算します。

なお、r8で指定する相対アドレスが指定できる範囲は符号付12bitデータ(−128～127)になっています。

## 【例】

```

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

## &lt; 注意事項 &gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命令

### BNC Rd, #imm8, r12

命令コード	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][0 0 0 1 r11 ~ r8][r7 ~ r0] 20001000H
引数	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
ワード数	2
サイクル数	2 or 3
機能	If result of unsigned comparison is (Rd) $\geq$ #imm8, then (PC) $\leftarrow$ (PC)+4 $\pm$ (r12) If result of unsigned comparison is (Rd) < #imm8, then (PC) $\leftarrow$ (PC)+4
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S, N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容と、imm8で指定するイミディエイトデータの引き算結果が負でなければ、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。引き算結果が負ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R7、imm8で指定できる範囲は0～FFh、r12で指定する相対アドレスが指定できる範囲は符号付12bitデータ(－2048～2047)になっています。

#### 【例】

	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

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。

BNC Rd, Rs, r12

命令コード	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][0 0 0 1 r11 ~ r8][r7 ~ r0] 0D001000H
引数	Rd = 4bit(R select),Rs = 4bit(R select),r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	If result of unsigned comparison is (Rd) $\geq$ (Rs), then (PC) $\leftarrow$ (PC)+4 $\pm$ (r12) If result of unsigned comparison is (Rd) < (Rs), then (PC) $\leftarrow$ (PC)+4
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S, N0~N3

## 【説明】

Rdで指定する汎用レジスタの内容とRsで指定する汎用レジスタの内容の引き算結果が負でなければ、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。引き算結果が負ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0~R15、Rsで指定できる範囲はR0~R15、r12で指定する相対アドレスが指定できる範囲は符号付12bitデータ(−2048~2047)になっています。

## 【例】

```

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

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命令

### BNC Rx, #imm16, r8

命令コード	[1 1 0 0 0 0 1][r7r6r5r4r3r2r1r0][i15 ~ i8][i7 ~ i0]C1000000H
引数	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)
ワード数	2
サイクル数	3 or 4
機能	If result of unsigned comparison is (Rx) $\geq$ #imm16, then (PC) $\leftarrow$ (PC)+4 $\pm$ (r8) If result of unsigned comparison is (Rx) < #imm16, then (PC) $\leftarrow$ (PC)+4
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S

#### 【説明】

PSW の bit12～15 (N0～N3)の値によって間接指定される汎用レジスタ(Rx)の内容とimm16で指定するイミディエイトデータの引き算結果が負でなければ、r8で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。引き算結果が負ならば、PCに4を加算します。

なお、imm16で指定できる範囲は0～FFFFh、r8で指定する相対アドレスが指定できる範囲は符号付8bitデータ(－128～127)になっています。

#### 【例】

```

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

```

PC	R0	R1	R2	R3	PSW
-	-	-	-	-	-
9004h	-	-	5678h	-	2000h
9008h	-	-	5678h	FFFFh	3040h
900Ch	1234h	-	5678h	FFFFh	0020h
9010h	1234h	-	5678h	FFFFh	006Ch
9014h	1234h	CDEFh	5678h	FFFFh	104Ch
901Eh	1234h	CDEFh	5678h	FFFFh	1010h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	1234h	CDEFh	5678h	0000h	3013h
9022h	1234h	CDEFh	5678h	0000h	3013h

#### <注意事項>

条件が一致した場合、この命令は4サイクルとなります。

BNV r8

命令コード	[1 1 0 1 1 0 0 1][r7r6r5r4r3r2r1r0]	D900H
引数	r8 = 8bit(relative address,signed)	
ワード数	1	
サイクル数	2 or 3	
機能	If OV = 0 then (PC)←(PC)+2±(r8) If OV = 1 then (PC)←(PC)+2	
影響を受けるフラグ		

## 【説明】

オーバーフローフラグ (OV) が0ならば、r8で指定する相対アドレス+2をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。OVが1ならば、PCに2を加算します。

なお、r8で指定する相対アドレスが指定できる範囲は符号付8bitデータ(−128～127)になっています。

## 【例】

```

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

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命令

### BNV Rd, #imm8, r12

命令コード	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][1 0 0 1 r11 ~ r8][r7 ~ r0] 20009000H
引数	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S, N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容と、imm8で指定するイミディエイトデータの引き算結果でオーバーフローフラグ(OV)が0ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算の結果、OVが1ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R7、imm8で指定できる範囲は0～FFh、r12で指定する相対アドレスが指定できる範囲は符号付12bitデータ(－2048～2047)になっています。

#### 【例】

	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

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。

BNV Rd, Rs, r12

命令コード	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][1 0 0 1 r11 ~ r8][r7 ~ r0] 0D009000H
引数	Rd = 4bit(R select),Rs = 4bit(R select),r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S, N0～N3

## 【説明】

Rdで指定する汎用レジスタの内容と、Rsで指定する汎用レジスタの内容の引き算結果でオーバーフローフラグ(OV)が0ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算の結果、OVが1ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15、r12で指定する相対アドレスが指定できる範囲は符号付12bitデータ(－2048～2047)になっています。

## 【例】

```

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

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命 令

### BNV Rx, #imm16, r8

命令コード	[1 1 0 0 1 0 0 1][r7r6r5r4r3r2r1r0][i15 ~ i8][i7 ~ i0]	C9000000H
引数	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)	
ワード数	2	
サイクル数	3 or 4	
機能	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	
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S	

#### 【説明】

PSW の bit12～15 (N0～N3)の値によって間接指定される汎用レジスタ(Rx)の内容と、imm16で指定するイミディエイトデータの引き算結果で、オーバーフローフラグ(OV)が0ならば、r8で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算の結果、OVが1ならば、PCに4を加算します。

なお、imm16で指定できる範囲は0～FFFF、r8で指定する相対アドレスが指定できる範囲は符号付8bitデータ(−128～127)になっています。

#### 【例】

```

MOV.W  R2,#0x1234
MOV.W  R3,#0xFFFF
loop:
MOV.W  R0,#0x8000
BNV     Rx,#0x1234,LA  ;; NOT JUMP LA
MOV.W  R1,#0x5678
BNV     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	8000h	-	1234h	FFFFh	0061h
9010h	8000h	-	1234h	FFFFh	0038h
9014h	8000h	5678h	1234h	FFFFh	1018h
901Eh	8000h	5678h	1234h	FFFFh	1000h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	8000h	5678h	1234h	0000h	3003h
9022h	8000h	5678h	1234h	0000h	3003h

#### <注意事項>

条件が一致した場合、この命令は4サイクルとなります。



BNZ r8

命令コード	[1 1 0 1 1 0 1][r7r6r5r4r3r2r1r0]	DD00H
引数	r8 = 8bit(relative address,signed)	
ワード数	1	
サイクル数	2 or 3	
機能	If Z16 = 0, then (PC)←(PC)+2±(r8) If Z16 = 1, then (PC)←(PC)+2	
影響を受けるフラグ		

## 【説明】

16bit演算フラグ(Z16)が0ならば、r8で指定する相対アドレス+2をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。Z16が1ならば、PCに2を加算します。

なお、r8で指定する相対アドレスが指定できる範囲は符号付8bitデータ(−128～127)になっています。

## 【例】

```

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

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命令

### BNZ Rd, #imm8, r12

命令コード	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][1 1 0 1 r11 ~ r8][r7 ~ r0] 2000D000H
引数	Rd = 3bit(R select),imm8 = 8bit(immediate data) r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容と、imm8で指定するイミディエイトデータの引き算結果で16bit演算フラグ(Z16)が0ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算の結果、Z16が1ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R7、imm8で指定できる範囲は0～FFh、r12で指定する相対アドレスが指定できる範囲は符号付12bitデータ(－2048～2047)になっています。

#### 【例】

	PC	R0	R1	R2	R3	PSW
	-	-	-	-	-	-
MOV.W R0,#0x0034	9002h	0034h	-	-	-	0020h
MOV.W R1,#0x8234	9006h	0034h	8234h	-	-	1060h
MOV.W R2,#0x0034	9008h	0034h	8234h	0034h	-	2020h
MOV.W R3,#0xFFFF	900Ch	0034h	8234h	0034h	FFFFh	3040h
loop:						
BNZ R0,#0x34,LA ;; NOT JUMP LA	9010h	0034h	8234h	0034h	FFFFh	0003h
BNZ R1,#0x34,LB ;; JUMP LB	901Ah	0034h	8234h	0034h	FFFFh	1041h
BR loop	-	-	-	-	-	-
LA:						
DEC R3	-	-	-	-	-	-
BR loop	-	-	-	-	-	-
LB:						
INC R3	901Ch	0034h	8234h	0034h	0000h	3003h
NOP	901Eh	0034h	8234h	0034h	0000h	3003h

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。

## BNZ Rd, Rs, r12

命令コード	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][1 1 0 1 r11 ~ r8][r7 ~ r0] 0D00D000H
引数	Rd = 4bit(R select),Rs = 4bit(R select),r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S, N0～N3

## 【説明】

Rdで指定する汎用レジスタの内容と、Rsで指定する汎用レジスタの内容の引き算結果で16bit演算フラグ(Z16)が0ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算の結果、Z16が1ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15、r12で指定する相対アドレスが指定できる範囲は符号付12bitデータ(−2048～2047)になっています。

## 【例】

```

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

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命令

### BNZ Rx, #imm16, r8

命令コード	[1 1 0 0 1 1 0 1][r7r6r5r4r3r2r1r0][i15 ~ i8][i7 ~ i0]	CD000000H
引数	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)	
ワード数	2	
サイクル数	3 or 4	
機能	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	
影響を受けるフラグ	Z8, Z16, CY, HC, OV, P, S	

#### 【説明】

PSW の bit12～15 (N0～N3)の値によって間接指定される汎用レジスタ(Rx)の内容と、imm16で指定するイミディエイトデータの引き算結果で16bit演算フラグ(Z16)が0ならば、r8で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算の結果、Z16が1ならば、PCに4を加算します。

なお、imm16で指定できる範囲は0～FFFFh、r8で指定する相対アドレスが指定できる範囲は符号付8bitデータ(－128～127)になっています。

#### 【例】

```

MOV.W  R2, #0x1234
MOV.W  R3, #0xFFFF
loop:
MOV.W  R0, #0x1234
BNZ    Rx, #0x1234, LA  ;; NOT JUMP LA
MOV.W  R1, #0x8234
BNZ    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	1234h	-	1234h	FFFFh	0020h
9010h	1234h	-	1234h	FFFFh	0003h
9014h	1234h	8234h	1234h	FFFFh	1060h
901Eh	1234h	8234h	1234h	FFFFh	1031h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	1234h	8234h	1234h	0000h	3013h
9020h	1234h	8234h	1234h	0000h	3013h

#### <注意事項>

条件が一致した場合、この命令は4サイクルとなります。

BNZ. B    r8

命令コード	[1 1 0 1 1 0 0][r7r6r5r4r3r2r1r0]	DC00H
引数	r8 = 8bit(relative address,signed)	
ワード数	1	
サイクル数	2 or 3	
機能	If Z8 = 0, then (PC)←(PC)+2±(r8) If Z8 = 1, then (PC)←(PC)+2	
影響を受けるフラグ		

## 【説明】

8bit演算フラグ(Z8)が0ならば、r8で指定する相対アドレス+2をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。Z8が1ならば、PCに2を加算します。

なお、r8で指定する相対アドレスが指定できる範囲は符号付8bitデータ(−128～127)になっています。

## 【例】

```

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

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命 令

### BNZ. B    Rd, #imm8, r12

命令コード	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][1 1 0 0 r11 ~ r8][r7 ~ r0] 2000C000H
引数	Rd = 3bit(R select),imm8 = 8bit(immediate data) r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S, N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容と、imm8で指定するイミディエイトデータの引き算結果で8bit演算フラグ(Z8)が0ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算の結果、Z8が1ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R7、imm8で指定できる範囲は0～FFh、r12で指定する相対アドレスが指定できる範囲は符号付12bitデータ(－2048～2047)になっています。

#### 【例】

	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:						
BNZ.B    R0,#0x34,LA    ;; NOT JUMP LA	9014h	5634h	8000h	1234h	FFFFh	0001h
BNZ.B    R1,#0x34,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

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。

BNZ. B Rd, Rs, r12

命令コード	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][1 1 0 0 r11 ~ r8][r7 ~ r0] 0D00C000H
引数	Rd = 4bit(R select),Rs = 4bit(R select),r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S, N0～N3

## 【説明】

Rdで指定する汎用レジスタの内容と、Rsで指定する汎用レジスタの内容の引き算結果で、8bit演算フラグ (Z8) が0ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算の結果、Z8が1ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15、r12で指定する相対アドレスが指定できる範囲は符号付12bitデータ(−2048～2047)になっています。

## 【例】

```

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

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命 令

### BNZ.B Rx, #imm16, r8

命令コード	[1 1 0 0 1 1 0 0][r7r6r5r4r3r2r1r0][i15 ~ i8][i7 ~ i0]	CC000000H
引数	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)	
ワード数	2	
サイクル数	3 or 4	
機能	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	
影響を受けるフラグ	Z8, Z16, CY, HC, OV, P, S	

#### 【説明】

PSW の bit12～15 (N0～N3)の値によって間接指定される汎用レジスタ(Rx)の内容と、imm16で指定するイミディエイトデータの引き算結果で、8bit演算フラグ(Z8)が0ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算の結果、Z8が1ならば、PCに4を加算します。

なお、imm16で指定できる範囲は0～FFFFh、r8で指定する相対アドレスが指定できる範囲は符号付8bitデータ(−128～127)になっています。

#### 【例】

```

MOV.W  R2, #0x1234
MOV.W  R3, #0xFFFF
loop:
MOV.W  R0, #0x5634
BNZ.B  Rx, #0x1234, LA  ;; NOT JUMP LA
MOV.W  R1, #0x8000
BNZ.B  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	5634h	-	1234h	FFFFh	0020h
9010h	5634h	-	1234h	FFFFh	0001h
9014h	5634h	8000h	1234h	FFFFh	1061h
901Eh	5634h	8000h	1234h	FFFFh	1038h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	5634h	8000h	1234h	0000h	301Bh
9022h	5634h	8000h	1234h	0000h	301Bh

#### <注意事項>

条件が一致した場合、この命令は4サイクルとなります。



BP m16, #imm3, r12

命令コード	[0 1 1 1 1 X 1][m7m6m5m4m3m2m1m0][0 i2i1i0 r11 ~ r8][r7 ~ r0] 7D00H(RAM),7F00H(SFR)
引数	m16 = 16bit(Lower 8bit valid for operation code),imm3 = 3bit(bit select) r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	3 or 4
機能	if (m16) of bit #imm3 = 1, then (PC)←(PC)+4±(r12) if (m16) of bit #imm3 = 0, then (PC)←(PC)+4
影響を受けるフラグ	

## 【説明】

m16で指定するデータメモリの内容のimm3で指定するイミディエイトデータが示すbitが1の場合、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。m16の内容の指定されるbitが0の場合、PCに4を加算します。

m16(第1オペランドデータ)の値によって、コンパイラは転送先がRAMであるのかSFRであるのかを判断して命令コードを生成します。

・RAMを指定する場合には、m16の値は00H~FFH(0000H~00FFH)の範囲で指定してください。この命令で100H番地以上のRAMを指定することはできません。

・SFRを指定する場合には、m16の値は7F00H~7FFFHの範囲で指定してください。なお、生成される命令コードの基本系は、それぞれ7D00H(RAM), 7F00H(SFR)になり、命令コードの動作に反映されるのは、m16の下位8ビットになります。

なお、imm3で指定できる範囲は0~8h、r12で指定する相対アドレスが指定できる範囲は符号付12bitデータ(−2048~2047)になっています。

## 【例】

		PC	RAM (51h)	RAM (50h)	R3
	MOV.W	0x50,#01221	-	-	-
	MOV.W	R3,#0FFFF	9004h	12h	21h
loop:			9008h	12h	21h
	BP	0x51,#0,LA ;; NOT JUMP LA			
	BP	0x50,#0,LB ;; JUMP LB	900Ch	12h	21h
	BR	loop	9016h	12h	21h
LA:			-	-	-
	DEC	R3			
	BR	loop	-	-	-
LB:					
	INC	R3	9018h	12h	21h
	NOP		901Ah	12h	21h

## &lt;注意事項&gt;

条件が一致した場合、この命令は4サイクルとなります。

## 命令

### BP Rd, #imm4, r12

命令コード	[0 0 0 0 1 0 1][i3i2i1i0d3d2d1d0][0 0 0 0 r11 ~ r8][r7 ~ r0] 0500H
引数	Rd = 4bit(R select), imm4 = 4bit(bit select), r12 = 12bit(relative address, signed)
ワード数	2
サイクル数	2 or 3
機能	If (Rd) of bit #imm4 = 1, then (PC) ← (PC) + 4 ± (r12) If (Rd) of bit #imm4 = 0, then (PC) ← (PC) + 4
影響を受けるフラグ	N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容のimm4で指定するイミディエイトデータが示すbitが1の場合、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。Rdの内容の指定されるbitが0の場合、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R15、imm4で指定できる範囲は0～Fh、r12で指定する相対アドレスが指定できる範囲は符号付12bitデータ(−2048～2047)になっています。

#### 【例】

```

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

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。

BP Rd, Rs, r12

命令コード	[0 0 0 0 1 1 1][s3s2s1s0d3d2d1d0][0 0 0 0 r11 ~ r8][r7 ~ r0] 0700H
引数	Rd = 4bit(R select),Rs = 4bit(bit select),r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	If (Rd) of bit (Rs)&000Fh =1, then (PC)←(PC)+4±(r12) If (Rd) of bit (Rs)&000Fh =0, then (PC)←(PC)+4
影響を受けるフラグ	N0～N3

## 【説明】

Rdで指定する汎用レジスタの内容のRsで指定する汎用レジスタの内容の下位4bitが示すbitが1の場合、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。Rdの内容の指定されるbitが0の場合、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15、r12で指定する相対アドレスが指定できる範囲は符号付12bitデータ(−2048～2047)になっています。

## 【例】

```

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

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命令

### BPL r8

命令コード	[1 1 0 1 1 0 0 0][r7r6r5r4r3r2r1r0]	D800H
引数	r8 = 8bit(relative address,signed)	
ワード数	1	
サイクル数	2 or 3	
機能	If S = 0, then (PC)←(PC)+2±(r8) If S = 1, then (PC)←(PC)+2	
影響を受けるフラグ		

#### 【説明】

サインフラグ(S)が0ならば、r8で指定する相対アドレス+2をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。Sが1ならば、PCに2を加算します。

なお、r8で指定する相対アドレスが指定できる範囲は符号付8bitデータ(−128～127)になっています。

#### 【例】

```
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

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。

## BPL Rd, #imm8, r12

命令コード	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][1 0 0 0 r11 ~ r8][r7 ~ r0] 20008000H
引数	Rd = 3bit(R select),imm8 = 8bit(immediate data) r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0～N3

## 【説明】

Rdで指定する汎用レジスタの内容と、imm8で指定するイミディエイトデータの引き算結果で、サインフラグ(S)が0ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算の結果、Sが1ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R7、imm8で指定できる範囲は0～FFh、r12で指定する相対アドレスが指定できる範囲は符号付12bitデータ(−2048～2047)になっています。

## 【例】

	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

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命令

### BPL Rd, Rs, r12

命令コード	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][1 0 0 0 r11 ~ r8][r7 ~ r0] 0D008000H
引数	Rd = 4bit(R select),Rs = 4bit(R select),r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S, N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容と、Rsで指定する汎用レジスタの内容の引き算結果で、サインフラグ(S)が0ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算の結果、Sが1ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15、r12で指定する相対アドレスが指定できる範囲は符号付12bitデータ(−2048～2047)になっています。

#### 【例】

```

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

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。

## BPL Rx, #imm16, r8

命令コード	[1 1 0 0 1 0 0 0][r7r6r5r4r3r2r1r0][i15 ~ i8][i7 ~ i0]	C8000000H
引数	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)	
ワード数	2	
サイクル数	3 or 4	
機能	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	
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S	

## 【説明】

PSW の bit12～15 (N0～N3) の値によって間接指定される汎用レジスタ(Rx) の内容と, imm16 で指定したイミディエイトデータの引き算結果でサインフラグ(S) が 0 ならば, r8 で指定した相対アドレス + 4 をプログラムカウンタ(PC) に加算し, その結果を PC に転送します。演算の結果, S が 1 ならば, PC に 4 を加算します。  
 なお, imm16 で指定できる範囲は 0～FFFFh, r8 で指定する相対アドレスが指定できる範囲は符号付 8bit データ(－128～127) になっています。

## 【例】

```

MOV.W  R2,#0x1234
MOV.W  R3,#0xFFFF
loop:
MOV.W  R0,#0x9876
BPL    Rx,#0x1234,LA  ;; NOT JUMP LA
MOV.W  R1,#0x5678
BPL    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	9876h	-	1234h	FFFFh	0040h
9010h	9876h	-	1234h	FFFFh	0060h
9014h	9876h	5678h	1234h	FFFFh	1000h
901Eh	9876h	5678h	1234h	FFFFh	1000h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	9876h	5678h	1234h	0000h	3003h
9022h	9876h	5678h	1234h	0000h	3003h

## &lt; 注意事項 &gt;

条件が一致した場合、この命令は 4 サイクルとなります。

## 命 令

### BR r12

命令コード	[0 0 0 1 r11r10r9r8][r7r6r5r4r3r2r1 0]	1000H
引数	r12 = 12bit(relative address,signed)	
ワード数	1	
サイクル数	2	
機能	(PC)←(PC)+2±(r12)	
影響を受けるフラグ		

#### 【説明】

r12で指定する相対アドレス+2を、プログラムカウンタ(PC)に加算し、その結果をPCに転送します。  
なお、r12で指定する相対アドレスが指定できる範囲は符号付12bitデータ(−2078〜2047)になっています。

【例】ラベルLAの値は、9106Hです。

```
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

命令コード	[0 0 0 0 0 0 0 0][0 0 1 0 s3s2s1s0]	0020H
引数	Rs = 4bit(relative address,signed)	
ワード数	1	
サイクル数	2	
機能	(PC)←(PC)+2±(Rs)	
影響を受けるフラグ		

## 【説明】

相対アドレス(Rsで指定する汎用レジスタの内容)+2を、プログラムカウンタ(PC)に加算し、その結果をPCに転送します。

なお、Rsで指定できる範囲はR0～R15、相対アドレス(Rsで指定する汎用レジスタの内容)で指定できる範囲は、符号付き16bitデータ(−32768～32767)になっています。

【例】ラベルLAの値は、9106Hです。

```

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

## 命 令

### BRK

命令コード	[0 0 0 0 0 0 0 0][0 0 0 0 0 1 0 1]	0005H
引数		
ワード数	1	
サイクル数	1	
機能	(PC)←(PC) : This instruction sequence 1 time	
影響を受けるフラグ		

#### 【説明】

それまでの状態を保持したまま、プログラムカウンタ(PC)が停止します。割り込み、又はリセットで解除が可能です。

BV r8

命令コード	[1 1 0 1 1 0 1 1][r7r6r5r4r3r2r1r0]	DB00H
引数	r8 = 8bit(relative address,signed)	
ワード数	1	
サイクル数	2 or 3	
機能	If OV = 1, then (PC)←(PC)+2±(r8) If OV = 0, then (PC)←(PC)+2	
影響を受けるフラグ		

## 【説明】

オーバーフローフラグ (OV) が1ならば、r8で指定する相対アドレス+2を、プログラムカウンタ(PC)に加算し、その結果をPCに転送します。OVが0ならば、PCに2を加算します。

なお、r8で指定する相対アドレスが指定できる範囲は、符号付き8bitデータ(−128～127)になっています。

## 【例】

```

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

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命令

### BV Rd, #imm8, r12

命令コード	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][1 0 1 1 r11 ~ r8][r7 ~ r0] 2000B000H
引数	Rd = 3bit(R select), imm8 = 8bit(immediate data) r12 = 12bit(relative address, signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8, Z16, CY, HC, OV, P, S, N0 ~ N3

#### 【説明】

Rdで指定する汎用レジスタの内容と、imm8で指定するイミディエイトデータの引き算結果でオーバーフローフラグ(OV)が1ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算の結果、OVが0ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R7、imm8で指定できる範囲は0～FFh、r12で指定する相対アドレスが指定できる範囲は、符号付12bitデータ(−2048～2047)になっています。

#### 【例】

	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:						
BV R1, #0x12, LA ;; NOT JUMP LA	9012h	8000h	5678h	0012h	FFFFh	1000h
BV R0, #0x12, LB ;; JUMP LB	901Ch	8000h	5678h	0012h	FFFFh	0038h
BR loop	-	-	-	-	-	-
LA:						
DEC R3	-	-	-	-	-	-
BR loop	-	-	-	-	-	-
LB:						
INC R3	901Eh	8000h	5678h	0012h	0000h	301Bh
NOP	9020h	8000h	5678h	0012h	0000h	301Bh

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。

BV Rd, Rs, r12

命令コード	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][1 0 1 1 r11 ~ r8][r7 ~ r0] 0D00B000H
引数	Rd = 4bit(R select),Rs = 4bit(R select),r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0～N3

## 【説明】

Rdで指定する汎用レジスタの内容と、Rsで指定する汎用レジスタの内容の引き算結果でオーバーフローフラグ(OV)が1ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算の結果、OVが0ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15、r12で指定する相対アドレスが指定できる範囲は符号付12bitデータ(－2048～2047)になっています。

## 【例】

```

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

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命令

### BV Rx, #imm16, r8

命令コード	[1 1 0 0 1 0 1 1][r7r6r5r4r3r2r1r0][i15 ~ i8][i7 ~ i0]	CB000000H
引数	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)	
ワード数	2	
サイクル数	3 or 4	
機能	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	
影響を受けるフラグ	Z8, Z16, CY, HC, OV, P, S	

#### 【説明】

PSW の bit12～15 (N0～N3) の値によって間接指定される汎用レジスタ(Rx) の内容と、imm16で指定したイミディエイトデータの引き算結果でオーバーフローフラグ(OV)が1ならば、r8で指定した相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算の結果、OVが0ならば、PCに4を加算します。

なお、imm16で指定できる範囲は0～FFFFh、r8で指定する相対アドレスが指定できる範囲は符号付8bitデータ(－128～127)になっています。

#### 【例】

```

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

#### <注意事項>

条件が一致した場合、この命令は4サイクルとなります。

BZ r8

命令コード	[1 1 0 1 1 1 1 1][r7r6r5r4r3r2r1r0]	DF00H
引数	r8 = 8bit(relative address,signed)	
ワード数	1	
サイクル数	2 or 3	
機能	If Z16 = 1, then (PC)←(PC)+2±(r8) If Z16 = 0, then (PC)←(PC)+2	
影響を受けるフラグ		

## 【説明】

16bit演算フラグ(Z16)が1ならば、r8で指定した相対アドレス+2をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。Z16が0ならば、PCに2を加算します。

なお、r8で指定する相対アドレスが指定できる範囲は符号付8bitデータ(−128～127)になっています。

## 【例】

```

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

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命令

### BZ Rd, #imm8, r12

命令コード	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][1 1 1 r11 ~ r8][r7 ~ r0] 2000F000H
引数	Rd = 3bit(R select),imm8 = 8bit(immediate data) r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容と、imm8で指定するイミディエイトデータの引き算結果で16bit演算フラグ(Z16)が1ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算の結果、Z16が0ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R7、imm8で指定できる範囲は0～FFh、r12で指定する相対アドレスが指定できる範囲は符号付12bitデータ(－2048～2047)になっています。

#### 【例】

	PC	R0	R1	R2	R3	PSW
	-	-	-	-	-	-
MOV.W R0,#0x0034	9002h	0034h	-	-	-	0020h
MOV.W R1,#0x8234	9006h	0034h	8234h	-	-	1060h
MOV.W R2,#0x0034	9008h	0034h	8234h	0034h	-	2020h
MOV.W R3,#0xFFFF	900Ch	0034h	8234h	0034h	FFFFh	3040h
loop:						
BZ R1,#0x34,LA ;; NOT JUMP LA	9010h	0034h	8234h	0034h	FFFFh	1041h
BZ R0,#0x34,LB ;; JUMP LB	901Ah	0034h	8234h	0034h	FFFFh	0003h
BR loop	-	-	-	-	-	-
LA:						
DEC R3	-	-	-	-	-	-
BR loop	-	-	-	-	-	-
LB:						
INC R3	901Ch	0034h	8234h	0034h	0000h	3003h
NOP	901Eh	0034h	8234h	0034h	0000h	3003h

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。



BZ Rd, Rs, r12

命令コード	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][1 1 1 1 r11 ~ r8][r7 ~ r0] 0D00F000H
引数	Rd = 4bit(R select),Rs = 4bit(R select),r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0～N3

## 【説明】

Rdで指定する汎用レジスタの内容と、Rsで指定する汎用レジスタの内容の引き算結果で16bit演算フラグ (Z16) が1ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算の結果、Z16が0ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15、r12で指定する相対アドレスが指定できる範囲は符号付12bitデータ(－2048～2047)になっています。

## 【例】

```

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

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命 令

### BZ Rx, #imm16, r8

命令コード	[1 1 0 0 1 1 1 1][r7r6r5r4r3r2r1r0][i15 ~ i8][i7 ~ i0]	CF000000H
引数	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)	
ワード数	2	
サイクル数	3 or 4	
機能	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	
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S	

#### 【説明】

PSW の bit12～15 (N0～N3)の値によって間接指定される汎用レジスタ(Rx)の内容と、imm16で指定するイミディエイトデータの引き算結果で16bit演算フラグ(Z16)が1ならば、r8で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算の結果、Z16が0ならば、PCに4を加算します。

なお、imm16で指定できる範囲は0～FFFFh、r8で指定する相対アドレスが指定できる範囲は符号付8bitデータ(－128～127)になっています。

#### 【例】

```

MOV.W  R2,#0x1234
MOV.W  R3,#0xFFFF
loop:
MOV.W  R1,#0x8234
BZ     Rx,#0x1234,LA  ;; NOT JUMP LA
MOV.W  R0,#0x1234
BZ     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	-	8234h	1234h	FFFFh	1060h
9010h	-	8234h	1234h	FFFFh	1031h
9014h	1234h	8234h	1234h	FFFFh	0030h
901Eh	1234h	8234h	1234h	FFFFh	0003h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	1234h	8234h	1234h	0000h	3003h
9020h	1234h	8234h	1234h	0000h	3003h

#### <注意事項>

条件が一致した場合、この命令は4サイクルとなります。

BZ. B r8

命令コード	[1 1 0 1 1 1 0][r7r6r5r4r3r2r1r0]	DE00H
引数	r8 = 8bit(relative address,signed)	
ワード数	1	
サイクル数	2 or 3	
機能	If Z8 = 1, then (PC)←(PC)+2± (r8) If Z8 = 0, then (PC)←(PC)+2	
影響を受けるフラグ		

## 【説明】

8bit演算フラグ(Z8)が1ならば、r8で指定する相対アドレス+2をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。Z8が0ならば、PCに2を加算します。

なお、r8で指定する相対アドレスが指定できる範囲は符号付8bitデータ(−128～127)になっています。

## 【例】

```

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

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命令

### BZ. B Rd, #imm8, r12

命令コード	[0 0 1 0 d2d1d0 0][i7i6i5i4i3i2i1i0][1 1 1 0 r11 ~ r8][r7 ~ r0] 2000E000H
引数	Rd = 3bit(R select),imm8 = 8bit(immediate data) r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容と、imm8で指定するイミディエイトデータの引き算結果で8bit演算フラグ(Z8)が1ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算の結果、Z8が0ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R7、imm8で指定できる範囲は0～FFh、r12で指定する相対アドレスが指定できる範囲は符号付12bitデータ(－2048～2047)になっています。

#### 【例】

	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

#### <注意事項>

条件が一致した場合、この命令は3サイクルとなります。

BZ. B Rd, Rs, r12

命令コード	[0 0 0 0 1 1 0 1][s3s2s1s0d3d2d1d0][1 1 1 0 r11 ~ r8][r7 ~ r0] 0D00E000H
引数	Rd = 4bit(R select),Rs = 4bit(R select),r12 = 12bit(relative address,signed)
ワード数	2
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S, N0～N3

## 【説明】

Rdで指定する汎用レジスタの内容と、Rsで指定する汎用レジスタの内容の引き算結果で、8bit演算フラグ (Z8) が1ならば、r12で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算の結果、Z8が0ならば、PCに4を加算します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15、r12で指定する相対アドレスが指定できる範囲は符号付12bitデータ(－2048～2047)になっています。

## 【例】

```

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

## &lt;注意事項&gt;

条件が一致した場合、この命令は3サイクルとなります。

## 命 令

### BZ. B Rx, #imm16, r8

命令コード	[1 1 0 0 1 1 1 0][r7r6r5r4r3r2r1r0][i15 ~ i8][i7 ~ i0]	CE000000H
引数	imm16 = 16bit(immediate data), r8 = 8bit(relative address, signed)	
ワード数	2	
サイクル数	3 or 4	
機能	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	
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S	

#### 【説明】

PSW の bit12～15 (N0～N3)の値によって間接指定される汎用レジスタ(Rx)の内容と、imm16で指定するイミディエイトデータの引き算結果で8bit演算フラグ(Z8)が1ならば、r8で指定する相対アドレス+4をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。演算の結果、Z8が0ならば、PCに4を加算します。

なお、imm16で指定できる範囲は0～FFFFh、r8で指定する相対アドレスが指定できる範囲は符号付8bitデータ(－128～127)になっています。

#### 【例】

```

MOV.W  R2,#0x1234
MOV.W  R3,#0xFFFF
loop:
MOV.W  R1,#0x8000
BZ.B   Rx,#0x1234,LA  ;; NOT JUMP LA
MOV.W  R0,#0x5634
BZ.B   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	-	8000h	1234h	FFFFh	1061h
9010h	-	8000h	1234h	FFFFh	1038h
9014h	5634h	8000h	1234h	FFFFh	0038h
901Eh	5634h	8000h	1234h	FFFFh	0001h
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
9020h	5634h	8000h	1234h	0000h	3003h
9022h	5634h	8000h	1234h	0000h	3003h

#### <注意事項>

条件が一致した場合、この命令は4サイクルとなります。

## CALL Rb, Rs

命令コード	[0 0 0 0 0 0 0 0][1 0 1 b0 s3s2s1s0] 00A0H
引数	Rb = 1bit(absolute address),Rs = 4bit(absolute address)
ワード数	1
サイクル数	4
機能	(SP)←(SP)+4 : [SP+1,SP]←(PC&0000FFFFh), [SP+3,SP+2]←(PC&FFFF0000h) (PC)←(Rb<<16+Rs)
影響を受けるフラグ	

## 【説明】

CALL命令の次の命令番地(戻り番地)をスタックポインタ(SP)で指定されるデータメモリ(RAM)に格納し、さらにSPをインクリメントします。最後に絶対アドレス(ベースレジスタ(Rb)で指定される汎用レジスタの内容を上位16bit、Rsの内容を下位16bitとした32bit)をプログラムカウンタ(PC)に転送します。  
なお、Rbで指定できる範囲はR8・R9、Rsの指定できる範囲はR0～R15になっています。

【例】ラベルLAの値は、910AHです。

	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

## 命令

### CALLF a24

命令コード	[0 0 0 0 0 0 1][a7a6a5a4a3a2a1a0][a23 ~ a16][a15 ~ a8]0100H
引数	a24 = 24bit(absolute address)
ワード数	2
サイクル数	4
機能	(SP)←(SP)+4 : [SP+1,SP]←(PC&0000FFFFh), [SP+3,SP+2]←(PC&FFFF0000h) (PC)←(a24)
影響を受けるフラグ	

#### 【説明】

CALLF命令の次の命令番地(戻り番地)をスタックポインタ(SP)で指定されるデータメモリ(RAM)に格納し、さらにSPをインクリメントします。最後に絶対アドレス(a24)をプログラムカウンタ(PC)に転送します。なお、a24で指定できる範囲は0～FF\_FFFFhになっています。

【例】ラベルLAの値は、910AHです。

	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

命令コード	[0 0 0 1 r11r10r9r8][r7r6r5r4r3r2r1 1]1001H
引数	r12 = 12bit(relative address,signed)
ワード数	1
サイクル数	4
機能	(SP)←(SP)+4 : [SP+1,SP]←(PC&0000FFFFh), [SP+3,SP+2]←(PC&FFFF0000h) (PC)←(PC)+2±(r12)
影響を受けるフラグ	

## 【説明】

CALLR命令の次の命令番地(戻り番地)をスタックポインタ(SP)で指定されるデータメモリ(RAM)に格納し、さらにSPをインクリメントします。最後に相対アドレス(r12) + 2をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。

なお、r12で指定する相対アドレスが指定できる範囲は符号付き12bitデータ(−2048～2047)になっています。

【例】ラベルLAの値は、910AHです。

	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

## &lt;注意事項&gt;

この場合の相対アドレス(r)は、符号付き12bitデータの範囲内(−2048～+2047)で有効です。

## 命令

### CALLR Rs

命令コード	[0 0 0 0 0 0 0 0][0 0 0 1 s3s2s1s0]0010H
引数	Rs = 4bit(relative address,signed)
ワード数	1
サイクル数	4
機能	(SP)←(SP)+4 : [SP+1,SP]←(PC&0000FFFFh), [SP+3,SP+2]←(PC&FFFF0000h) (PC)←(PC)+2±(Rs)
影響を受けるフラグ	

#### 【説明】

CALLR命令の次の命令番地(戻り番地)をスタックポインタ(SP)で指定されるデータメモリ(RAM)に格納し、さらにSPをインクリメントします。最後に相対アドレス(Rsで指定する汎用レジスタの内容)+2をプログラムカウンタ(PC)に加算し、その結果をPCに転送します。

なお、Rsの指定できる範囲はR0～R15になっています。

【例】ラベルLAの値は、910AHです。

	PC	RAM (00h)	RAM (01h)	RAM (02h)	RAM (03h)	R3	PSW	SP
	-	-	-	-	-	-	-	-
MOV.W R15,#0x0000	9004h	-	-	-	-	-	F003h	0000h
MOV.W R3,#0x0100	9008h	-	-	-	-	0100h	3021h	0000h
loop:								
CALLR R3 ;; CALL LA	910Ah	0Ah	90h	00h	00h	0100h	3021h	0004h
INC R3	900Ch	0Ah	90h	00h	00h	0102h	3000h	0000h
NOP	900Eh	0Ah	90h	00h	00h	0102h	3000h	0000h
.								
.								
.								
LA:								
INC R3	910Ch	0Ah	90h	00h	00h	0101h	3000h	0004h
RET	900Ah	0Ah	90h	00h	00h	0101h	3000h	0000h

#### <注意事項>

この場合の相対アドレス(Rs)は、符号付き16bitデータの範囲内(−32768～+32767)で有効です。

## CBW Rd

命令コード	[0 0 1 1 0 0 0 0][1 0 1 0 d3d2d1d0] 30A0H
引数	Rd = 4bit(R select)
ワード数	1
サイクル数	1
機能	if (Rd) of bit7 = 1, then Hbyte(Rd) = FFh if (Rd) of bit7 = 0, then Hbyte(Rd) = 00h
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

## 【説明】

Rdで指定する汎用レジスタの内容のbit7を符号ビットとして、算術型8bitデータを16bitデータに拡張します。

なお、Rdで指定できる範囲はR0～R15になっています。

## 【例】

	R0	R1	R2	R3	N3～ 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

## &lt; 注意事項 &gt;

Rdの内容のbit7が“1”なら上位8bitを“FFH”に、bit7が“0”なら上位8bitを“00H”にします。

## 命令

### CLR1 m16, #imm3

命令コード	[1 1 1 X i2i1i0 0][m7m6m5m4m3m2m1m0] E000H(RAM),F000H(SFR)
引数	m16 = 16bit(Lower 8bit valid for operation code),imm3 = 3bit(bit select)
ワード数	1
サイクル数	2
機能	(m16) of bit #imm3 ← 0, (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S

#### 【説明】

m16で指定するデータメモリの内容のimm3で指定するイミディエイトデータが示すbitを0にクリアします。  
m16(第1オペランドデータ)の値によって、コンパイラは転送先がRAMであるのかSFRであるのかを判断して命令コードを生成します。

・RAMを指定する場合には、m16の値は00H～FFH(0000H～00FFH)の範囲で指定してください。この命令で100H番地以上のRAMを指定することはできません。

・SFRを指定する場合には、m16の値は7F00H～7FFFHの範囲で指定してください。なお、生成される命令コードの基本系は、それぞれE000H(RAM), F000H(SFR)になり、命令コードの動作に反映されるのは、m16の下位8ビットになります。

なお、imm3で指定できる範囲は0～7hになっています。

#### 【例】

	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

命令コード	[0 0 0 0 1 0 0 0][i3i2i1i0d3d2d1d0]0800H
引数	Rd = 4bit(R select),imm4 = 4bit(bit select)
ワード数	1
サイクル数	1
機能	(Rd) of bit #imm4 $\leftarrow$ 0 , (PC) $\leftarrow$ (PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

## 【説明】

Rdで指定する汎用レジスタの内容の、imm4で指定するイミディエイトデータが示すbitを0にクリアします。  
 なお、Rdで指定できる範囲はR0～R15、imm4で指定できる範囲は0～Fになっています。

## 【例】

	R0	R1	R2	R3	N3～ 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

## 命 令

### CLR1 Rd, Rs

命令コード	[0 0 0 0 1 0 1 0][s3s2s1s0d3d2d1d0]0A00H
引数	Rd = 4bit(R select),Rs = 4bit(bit select)
ワード数	1
サイクル数	1
機能	(Rd) of bit (Rs)&000Fh ← 0 , (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容で、Rsで指定する汎用レジスタの内容の下位4bitが示すbitを0にクリアします。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15になっています。

#### 【例】

	R0	R1	R2	R3	N3～ 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]

命令コード	[0 0 1 1 0 0 0 0][0 1 i1i0d3d2d1d0]	3040H
引数	Rd = 4bit(R select), imm2 = 2bit(immediate data)	
ワード数	1	
サイクル数	1	
機能	(Rd)←(Rd) - #imm2 - 1, (PC)←(PC)+2	
影響を受けるフラグ	Z8,Z16,P,S,N0～N3	

## 【説明】

Rdで指定する汎用レジスタの内容からimm2で指定するイミディエイトデータ+1の値を減算し、その結果をRdに転送します。

なお、Rdで指定できる範囲はR0～R15、imm2で指定できる範囲は0～3になっています。

## 【例】

	R0	R1	R2	R3	N3～ 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

## &lt;注意事項&gt;

イミディエイトデータ(imm2)を省略した場合は、“imm2=0”となります。

## 命 令

### DIV

命令コード	[0 0 0 0 0 0 0 0][1 1 0 0 0 0 0 0]00C0H
引数	
ワード数	1
サイクル数	18 cycles
機能	(R0 : quotient)...(R1 : remainder) $\leftarrow$ (R0) $\div$ (R2) , (PC) $\leftarrow$ (PC)+2
影響を受けるフラグ	Z8,Z16,P,S CY,HC,OV, N0～N3 は全てクリアされる。

#### 【説明】

汎用レジスタR0とR2の除算結果をR0に、剰余をR1に転送します。

R2が“0”の場合、有効な結果は保証されません。

#### 【例】

```
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

#### <注意事項>

フラグ(Z8, Z16, P, S)は、R0(商)に影響されます。





## 命 令

### HALT

命令コード	[0 0 0 0 0 0 0 0][0 0 0 0 1 0 0 0]0008H
引数	
ワード数	1
サイクル数	1
機能	HALT mode,(PC)←(PC)+2
影響を受けるフラグ	

#### 【説明】

HALT命令実行後、HALTモードに移行します。

## HOLD

命令コード	[0 0 0 0 0 0 0 0][0 0 0 0 1 0 1 0]	000AH
引数		
ワード数	1	
サイクル数	1	
機能	HOLD mode,(PC)←(PC)+2	
影響を受けるフラグ		

## 【説明】

HOLD命令実行後、HOLDモードに移行します。

## 命 令

### HOLDX

命令コード	[0 0 0 0 0 0 0 0][0 0 0 0 1 0 1 1]	000BH
引数		
ワード数	1	
サイクル数	1	
機能	HOLDX mode,(PC) $\leftarrow$ (PC)+2	
影響を受けるフラグ		

#### 【説明】

HOLDX命令実行後、HOLDXモードに移行します。

ICALL R<sub>b</sub>, R<sub>s</sub>

命令コード	[0 0 0 0 0 0 0 0][0 1 1 b0 s3s2s1s0] 0060H
引数	Rb = 1bit(absolute address), Rs = 4bit(absolute address)
ワード数	1
サイクル数	4
機能	(SP)←(SP)+6 : [SP+1,SP]←(PC&0000FFFFh), [SP+3,SP+2]←(PC&FFFF0000h),[SP+5,SP+4]←(PSW) (PC)←(Rb<<16+Rs)
影響を受けるフラグ	

## 【説明】

ICALL命令の次の命令番地(戻り番地)とプログラムステータスワード(PSW)の内容を、スタックポインタ(S P)で指定されるデータメモリ(RAM)に格納し、さらにSPをインクリメントします。最後に絶対アドレス(Rbで指定される汎用レジスタの内容を上位16bit、Rsの内容を下位16bitとした32bit)をプログラムカウンタ(PC)に転送します。

なお、Rbで指定できる範囲はR8・R9、Rsで指定できる範囲はR0～R15になっています。

## 【例】ラベルLAの値は、910AHです。

	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

## 命 令

### ICALLF    a24

命令コード	[0 0 0 0 0 1 1][a7a6a5a4a3a2a1a0][a23 ~ a16][a15 ~ a8]	0300H
引数	a24 = 24bit(absolute address)	
ワード数	2	
サイクル数	4	
機能	(SP)←(SP)+6 : [SP+1,SP]←(PC&0000FFFFh), [SP+3,SP+2]←(PC&FFFF0000h),[SP+6,SP+5]←(PSW) (PC)←(a24)	
影響を受けるフラグ		

#### 【説明】

ICALLF命令の次の命令番地(戻り番地)とプログラムステータスワード(PSW)の内容を、スタックポインタ(SP)で指定されるデータメモリ(RAM)に格納し、さらにSPをインクリメントします。最後に絶対アドレス(a24)をプログラムカウンタ(PC)に転送します。

なお、a24で指定できる範囲は0～FF\_FFFFになっています。

#### 【例】ラベルLAの値は、910AHです。

	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

命令コード	[0 0 0 0 0 0 0 0][0 0 1 1 s3s2s1s0]0030H
引数	Rs = 4bit(relative address,signed)
ワード数	1
サイクル数	4
機能	(SP)←(SP)+6 : [SP+1,SP]←(PC&0000FFFFh), [SP+3,SP+2]←(PC&FFFF0000h),[SP+5,SP+4]←(PSW) (PC)←(PC)+2±(Rs)
影響を受けるフラグ	

## 【説明】

ICALLR命令の次の命令番地(戻り番地)とプログラムステータスワード(PSW)の内容を、スタックポインタ(SP)で指定されるデータメモリ(RAM)に格納し、さらにSPをインクリメントします。最後に相対アドレス(Rsで指定される汎用レジスタの内容)+2をPCに加算し、その結果をプログラムカウンタ(PC)に転送します。なお、Rsで指定できる範囲はR0～R15、相対アドレス(Rsで指定する汎用レジスタの内容)で指定できる範囲は、符号付16bitデータ(−32768～+32767)になっています。

【例】ラベルLAの値は、910AHです。

	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

## 命 令

### INC Rd[, #imm2]

命令コード	[0 0 1 1 0 0 0 0][0 0 i1i0d3d2d1d0]3000H
引数	Rd = 4bit(R select),imm2 = 2bit(immediate data)
ワード数	1
サイクル数	1
機能	(Rd)←(Rd) + #imm2 + 1 , (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容とimm2で指定するイミディエイトデータ+1の値を加算し、その結果をRdに転送します。

なお、Rdで指定できる範囲はR0～R15、imm2で指定できる範囲は0～3になっています。

#### 【例】

	R0	R1	R2	R3	N3～ 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,#0xFFFFD	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

#### <注意事項>

イミディエイトデータ(imm2)を省略した場合は、“imm2=0”となります。



## IRET

命令コード	[0 0 0 0 0 0 0 0][0 0 0 0 0 0 1 0]0002H
引数	
ワード数	1
サイクル数	3
機能	(SP)←(SP)-6 : (PC)←(PC-3<<24+SP-4<<16+SP-5<<8+SP-6) (PSW)←(SP-1<<8,SP-2)
影響を受けるフラグ	

## 【説明】

スタックポインタ(SP)をデクリメントし、さらにSPで指定されるデータメモリ(RAM)の内容をプログラムカウンタ(PC)、プログラムステータスワード(PSW)に転送し、割り込み受付時に禁止された割り込み受付機能の動作を再開します。

【例】ラベルLAの値は、910AHです。

	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

## 命 令

### JMP R<sub>b</sub>, R<sub>s</sub>

命令コード	[0 0 0 0 0 0 0 0][0 1 0 b <sub>0</sub> s <sub>3</sub> s <sub>2</sub> s <sub>1</sub> s <sub>0</sub> ]	0040H
引数	R <sub>b</sub> = 1bit(absolute address), R <sub>s</sub> = 4bit(absolute address)	
ワード数	1	
サイクル数	2	
機能	(PC)←(R <sub>b</sub> <<16+R <sub>s</sub> )	
影響を受けるフラグ		

#### 【説明】

絶対アドレス(ベースレジスタ(R<sub>b</sub>))で指定する汎用レジスタの内容を上位16bit、R<sub>s</sub>の内容を下位16bitとした32bitをプログラムカウンタ(PC)に転送します。

なお、R<sub>b</sub>で指定できる範囲はR8・R9、R<sub>s</sub>で指定できる範囲はR0～R15になっています。

【例】ラベルLAの値は、9106Hです。

```
MOV.W  R3,#0x9106
MOV.W  R8,#0x0000
loop:
  JMP   R8,R3      ;; JUMP LA
  NOP
  NOP
  .
  .
  .
LA:
  INC   R3
  NOP
```

PC	R3	R8	PSW
	-	-	-
9004h	9106h	-	3060h
9004h	9106h	0000h	8003h
9106h	9106h	0000h	8003h
-	-	-	-
-	-	-	-
9108h	9107h	0000h	3060h
910Ah	9107h	0000h	3060h

JMPF a24

命令コード	[0 0 0 0 0 1 0][a7a6a5a4a3a2a1a0][a23 ~ a16][a15 ~ a8]	0200H
引数	a24 = 24bit(absolute address)	
ワード数	2	
サイクル数	3	
機能	(PC)←(a24)	
影響を受けるフラグ		

## 【説明】

a24で指定する絶対アドレスを、プログラムカウンタ(PC)に転送します。

なお、a24で指定できる範囲は0～FF\_FFFFになっています。

【例】ラベルLAの値は、9106Hです。

```

MOV.W   R3,#0xFFFF
loop:
JMPF     LA      ;; JUMP LA
NOP
NOP
.
.
.
LA:
INC      R3
NOP

```

PC	R3	PSW
-	-	-
9004h	FFFFh	3040h
9106h	FFFFh	3040h
-	-	-
-	-	-
9108h	0000h	3003h
910Ah	0000h	3003h

## 命令

### MASK Rd, #imm16

命令コード	[0 0 1 1 0 0 0 0][1 1 1 0 d3d2d1d0][i15 ~ i8][i7 ~ i0]	30E0H
引数	Rd = 4bit(R select), imm16 = 16bit(immediate data)	
ワード数	2	
サイクル数	4	
機能	(Rd) ← {(Rd) & ~ #imm16}   { (Rx) & #imm16} , (PC) ← (PC) + 4	
影響を受けるフラグ	Z8, Z16, P, S, N0 ~ N3	

#### 【説明】

PSW の bit12～15 (N0～N3) の値によって間接指定される汎用レジスタ(Rx)の内容を、imm16で指定するイミディエイトデータが“1”のbitに対応するbitのみRdに転送(上書き)します。

なお、Rdで指定できる範囲はR0～R15、imm16で指定できる範囲は0～FFFFになっています。

#### 【例】

	R0	R1	R2	R3	N3～ 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

命令コード	[0 0 1 1 0 0 1 1][s3s2s1s0d3d2d1d0]	3300H
引数	Rd = 4bit(R select), Rs = 4bit(R select)	
ワード数	1	
サイクル数	3	
機能	(Rd)←{(Rd) & ~(Rs)}   {(Rx) & (Rs)} , (PC)←(PC)+2	
影響を受けるフラグ	Z8,Z16,P,S, N0～N3	

## 【説明】

PSW の bit12～15 (N0～N3) の値によって間接指定される汎用レジスタ (Rx) の内容を、Rs の内容が“1”の bit に対応する bit のみ Rd に転送 (上書き) します。

なお、Rd で指定できる範囲は R0～R15、Rs で指定できる範囲は R0～R15 になっています。

## 【例】

	R0	R1	R2	R3	N3～ 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

## 命令

### MOV Rd, Rs

命令コード	[0 1 0 0 0 1 1 0][s3s2s1s0d3d2d1d0]	4600H
引数	Rd = 4bit(R select), Rs = 4bit(R select)	
ワード数	1	
サイクル数	1	
機能	(Rd)←(Rs) , (PC)←(PC)+2	
影響を受けるフラグ	Z8,Z16,P,S,N0～N3	

#### 【説明】

Rsで指定する汎用レジスタの内容を、Rdで指定する汎用レジスタに転送します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15になっています。

#### 【例】

	R0	R1	R2	R3	N3～ 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

命令コード	[0 1 1 1 0 0 1 0][d3d2d1d0 0 s2s1s0] 7200H
引数	Rd = 4bit(R select), Rs = 3bit(R select)
ワード数	1
サイクル数	2 or 3
機能	[Rd] ← Lobyte (Rs) , (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S, N0～N3

## 【説明】

Rsで指定する汎用レジスタの下位8bitの内容を、Rdの内容によってアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)に転送します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R7になっています。

## 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

## &lt;注意事項&gt;

Rsの内容をプログラムメモリ(ROM)に転送すると、3サイクル命令になります。  
但し、実際には“ROM”に転送することは出来ません。

## 命令

### MOV. B (—Rd), Rs

命令コード	[0 1 1 0 1 0 1 0][d3d2d1d0 0 s2s1s0] 6A00H
引数	Rd = 4bit(R select), Rs = 3bit(R select)
ワード数	1
サイクル数	2 or 3
機能	(Rd)←(Rd)-1, [Rd] ← Lobyte (Rs) , (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

最初に、Rdで指定する汎用レジスタの内容を減算(－1)します。

その後、Rsで指定する汎用レジスタの下位8bitの内容を、Rdの内容によってアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)に転送します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R7になっています。

#### 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

#### <注意事項>

Rsの内容をプログラムメモリ(ROM)に転送すると、3サイクル命令になります。  
但し、実際には“ROM”に転送することは出来ません。



MOV. B (R<sub>d</sub>, ±n), R<sub>s</sub>

命令コード	[0 1 1 1 0 0 1 0][d3d2d1d0 1 s2s1s0][0 0 0 0 n11 ~ n8][n7 ~ n0] 7208H
引数	Rd = 4bit(R select), n = 12bit(signed), Rs = 3bit(R select)
ワード数	2
サイクル数	3 or 4
機能	[(Rd±n)&FFFFh]←Lobyte (Rs), (PC)←(PC)+4
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

## 【説明】

Rsで指定する汎用レジスタの下位8bitの内容を、Rdの内容とnの演算結果\*1によってアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)に転送します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R7、nで指定できる範囲は符号付12bitデータ(－2048～2047)になっています。

\*1:16ビット演算結果のキャリー・ボローは、無視されます。

## 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

## &lt;注意事項&gt;

Rsの内容をプログラムメモリ(ROM)に転送すると、4サイクル命令になります。  
但し、実際には“ROM”に転送することは出来ません。

## 命 令

### MOV. B (—Rd, ±n), Rs

命令コード	[0 1 1 0 1 0 1 0][d3d2d1d0 1 s2s1s0][0 0 0 0 n11 ~ n8][n7 ~ n0] 6A08H
引数	Rd = 4bit(R select), n = 12bit(signed), Rs = 3bit(R select)
ワード数	2
サイクル数	3 or 4
機能	(Rd)←(Rd)-1, [(Rd±n)&FFFFh] ← Lobyte(Rs), (PC)←(PC)+4
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

最初に、Rdで指定する汎用レジスタの内容を、減算(－1)します。その後、Rsの下位8bitの内容を、Rdの内容とnの演算結果\*<sup>1</sup>によってアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、又はプログラムメモリ(ROM)に転送します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R7、nで指定できる範囲は符号付12bitデータ(－2048～2047)になっています。

\*1:16ビット演算結果のキャリー・ボローは、無視されます。

#### 【例】

	RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

#### <注意事項>

Rsの内容をプログラムメモリ(ROM)に転送すると、4サイクル命令になります。  
但し、実際には“ROM”に転送することは出来ません。

## MOV. B (Rd++), Rs

命令コード	[0 1 1 0 0 0 1 0][d3d2d1d0 0 s2s1s0] 6200H
引数	Rd = 4bit(R select),Rs = 3bit(R select)
ワード数	1
サイクル数	2 or 3
機能	[Rd] ← Lobyte(Rs) , (Rd)←(Rd)+1 , (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

## 【説明】

Rsで指定する汎用レジスタの下位8bitの内容を、Rdの内容によってアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)に転送します。その後、Rdの内容を+1します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R7になっています。

## 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

## &lt;注意事項&gt;

Rsの内容をプログラムメモリ(ROM)に転送すると、3サイクル命令になります。  
但し、実際には“ROM”に転送することは出来ません。

## 命令

### MOV. B (Rd++, ±n), Rs

命令コード	[0 1 1 0 0 0 1 0][d3d2d1d0 1 s2s1s0][0 0 0 0 n11 ~ n8][n7 ~ n0] 6208H
引数	Rd = 4bit(R select), n = 12bit(signed), Rs = 3bit(R select)
ワード数	2
サイクル数	3 or 4
機能	[(Rd±n)&FFFFh] ← Lobyte(Rs), (Rd)←(Rd)+1, (PC)←(PC)+4
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

Rsで指定する汎用レジスタの下位8bitの内容を、Rdで指定する汎用レジスタの内容とnの演算結果\*1によってアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)に転送します。その後、Rdの内容を+1します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R7、nで指定できる範囲は符号付12bitデータ(−2048～2047)になっています。

\*1:16ビット演算結果のキャリー・ボローは、無視されます。

#### 【例】

	RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

#### <注意事項>

Rsの内容をプログラムメモリ(ROM)に転送すると、4サイクル命令になります。  
但し、実際には“ROM”に転送することは出来ません。

MOV. B m16, #imm16

命令コード	[0 1 1 1 0 X 0][m7m6m5m4m3m2m1m0][i15 ~ i8][i7 ~ i0] 7800H(RAM), 7A00H(SFR)
引数	m16 = 16bit (lower 8bit valid for operation code) imm16 = 16bit (immediate data)
ワード数	2
サイクル数	2
機能	(m16)←Lobyte #imm16, (PC)←(PC)+4
影響を受けるフラグ	Z8,Z16,P,S

## 【説明】

imm16で指定するイミディエイトデータの下位8bitを、m16でアドレス指定するRAM(データメモリ)またはSFR(内蔵する周辺機能を制御する専用レジスタ群)に転送します。

m16(第1オペランドデータ)の値によって、コンパイラは転送先がRAMであるのかSFRであるのかを判断して命令コードを生成します。

- RAMを指定する場合には、m16の値は00H~FFH(0000H~00FFH)の範囲で指定してください。この命令で100H番地以上のRAMを指定することはできません。
- SFRを指定する場合には、m16の値は7F00H~7FFFHの範囲で指定してください。

なお、生成される命令コードの基本系は、それぞれ7800H(RAM), 7A00H(SFR)になり、命令コードの動作に反映されるのは、m16の下位8ビットになります。

imm16(第2オペランドデータ)には、16ビットデータを指定することができますが、この命令はバイト転送を行う命令ですから、imm16の上位8ビットは実際の動作には無関係です。16ビットデータを扱いたい場合には、MOV. W命令を使用してください。

## 【例】

	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

## 命令

### MOV. B m16, Rs

命令コード	[1 0 X 1 s2s1s0 0][m7m6m5m4m3m2m1m0] 9000H(RAM),B000H(SFR)
引数	m16 = 16bit(Lower 8bit valid for operation code),Rs = 3bit(R select)
ワード数	1
サイクル数	1
機能	(m16) ← Lobyte(Rs) , (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

Rsで指定する汎用レジスタの下位8bitの内容を、m16でアドレス指定する RAM(データメモリ)または SFR (内蔵する周辺機能を制御する専用レジスタ群)に転送します。

m16(第1オペランドデータ)の値によって、コンパイラは転送先が RAM であるのか SFR であるのかを判断して命令コードを生成します。

- RAM を指定する場合には、m16の値は 00H～FFH(0000H～00FFH)の範囲で指定してください。この命令で 100H 番地以上の RAM を指定することはできません。
- SFR を指定する場合には、m16の値は 7F00H～7FFFH の範囲で指定してください。

なお、生成される命令コードの基本系は、それぞれ 9000H(RAM), B000H(SFR)になり、命令コードの動作に反映されるのは、m16の下位 8ビットになります。

Rs(第2オペランドデータ)には、16ビットデータを指定することができますが、この命令はバイト転送を行う命令ですから、Rsの上位8ビットは実際の動作には無関係です。16ビットデータを扱いたい場合には、MOV. W命令を使用してください。

#### 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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)

命令コード	[0 1 1 1 0 0 0 0][s3s2s1s0 0 d2d1d0]	7000H
引数	Rd = 3bit(R select),Rs = 4bit(R select)	
ワード数	1	
サイクル数	2 or 3	
機能	(Rd) ← Lobyte [Rs] , (PC)←(PC)+2	
影響を受けるフラグ	Z8,Z16,P,S,N0～N3	

## 【説明】

Rsの内容によってアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)の内容を、Rdの下位8bitに転送します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R7になっています。

## 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

## &lt;注意事項&gt;

Rdの上位8bitには、“00H”が入ります。

プログラムメモリ(ROM)の内容をRdに転送すると、3サイクル命令となります。

## 命 令

### MOV. B Rd, (—Rs)

命令コード	[0 1 1 0 1 0 0 0][s3s2s1s0 0 d2d1d0] 6800H
引数	Rd = 3bit(R select),Rs = 4bit(R select)
ワード数	1
サイクル数	2 or 3
機能	(Rs) ← (Rs) − 1, (Rd) ← Lbyte[Rs], (PC) ← (PC) + 2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

最初に、Rsで指定する汎用レジスタの内容を、減算(−1)します。その後、Rsの内容によってアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)の内容を、Rdの下位8 bitに転送します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R7になっています。

#### 【例】

	RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

#### <注意事項>

Rdの上位8bitには、“00H”が入ります。

プログラムメモリ(ROM)の内容をRdに転送すると、3サイクル命令となります。



MOV. B  $R_d, (R_s, \pm n)$ 

命令コード	[0 1 1 1 0 0 0 0][s3s2s1s0 1 d2d1d0][0 0 0 0 n11 ~ n8][n7 ~ n0] 7008H
引数	$R_d = 3\text{bit}(R\text{ select}), R_s = 4\text{bit}(R\text{ select}), n = 12\text{bit}(\text{signed})$
ワード数	2
サイクル数	3 or 4
機能	$(R_d) \leftarrow \text{Lobyte} [(R_s \pm n) \& \text{FFFFh}], (PC) \leftarrow (PC) + 4$
影響を受けるフラグ	Z8, Z16, P, S, N0 ~ N3

## 【説明】

汎用レジスタ $R_s$ の内容と $n$ の演算結果\*<sup>1</sup>によってアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)の内容を、 $R_d$ の下位8bitに転送します。

なお、 $R_d$ で指定できる範囲は $R_0 \sim R_7$ 、 $R_s$ で指定できる範囲は $R_0 \sim R_{15}$ 、 $n$ で指定できる範囲は符号付12bitデータ(−2048～2047)になっています。

\*1:16ビット演算結果のキャリー・ボローは、無視されます。

## 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3~ N0	Z8	Z16	P	S
		-	-	-	-	-	-	-	-	-	-	-
MOV.W	$R_3, \#0x0000$	-	-	-	-	-	0000h	3	1	1	0	0
MOV.W	$0x50, \#0x5555$	55h	55h	-	-	-	0000h	3	0	0	0	0
MOV.B	$R_0, (R_3, 0x50)$	55h	55h	0055h	-	-	0000h	0	0	0	0	0
MOV.W	$0x50, \#0x1200$	00h	12h	0055h	-	-	0000h	0	1	0	0	0
MOV.B	$R_1, (R_3, 0x50)$	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	$R_2, (R_3, 0x50)$	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	$R_0, (R_3, 0x50)$	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	$R_1, (R_3, 0x50)$	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	$R_2, (R_3, 0x50)$	55h	55h	0034h	0081h	0055h	0000h	2	0	0	0	0

## &lt; 注意事項 &gt;

$R_d$ の上位8bitには、“00H”が入ります。

プログラムメモリ(ROM)の内容を $R_d$ に転送すると、4サイクル命令となります。

## 命 令

### MOV. B Rd, (—Rs, ±n)

命令コード	[0 1 1 0 1 0 0 0][s3s2s1s0 1 d2d1d0][0 0 0 0 n11 ~ n8][n7 ~ n0] 6808H
引数	Rd = 3bit(R select),Rs = 4bit(R select),n = 12bit(signed)
ワード数	2
サイクル数	3 or 4
機能	(Rs)←(Rs)-1 , (Rd)←Lobyte [(Rs±n)&FFFFh] , (PC)←(PC)+4
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

最初に、Rsで指定する汎用レジスタの内容を、減算(－1)します。

その後、Rsの内容とnの演算結果\*<sup>1</sup>によってアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)の内容をRdの下位8bitに転送します。

なお、Rdで指定できる範囲はR0～R7、Rsで指定できる範囲はR0～R15、nで指定できる範囲は符号付12bitデータ(－2048～2047)になっています。

\*1:16ビット演算結果のキャリー・ボローは、無視されます。

#### 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

#### <注意事項>

Rdの上位8bitには、“00H”が入ります。

プログラムメモリ(ROM)の内容をRdに転送すると、4サイクル命令となります。

MOV. B Rd, (Rs++)

命令コード	[0 1 1 0 0 0 0 0][s3s2s1s0 0 d2d1d0]	6000H
引数	Rd = 3bit(R select),Rs = 4bit(R select)	
ワード数	1	
サイクル数	2 or 3	
機能	(Rd) ← Lobyte [Rs] , (Rs) ← (Rs) + 1 , (PC)←(PC)+2	
影響を受けるフラグ	Z8,Z16,P,S,N0～N3	

## 【説明】

Rsで指定する汎用レジスタの内容によってアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)の内容を、Rdの下位8bitに転送します。その後、Rsの内容を+1します。

なお、Rdで指定できる範囲はR0～R7、Rsで指定できる範囲はR0～R15になっています。

## 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

## &lt; 注意事項 &gt;

Rdの上位8bitには、“00H”が入ります。

プログラムメモリ(ROM)の内容をRdに転送すると、3サイクル命令となります。

## 命令

### MOV. B Rd, (Rs++, ±n)

命令コード	[0 1 1 0 0 0 0 0][s3s2s1s0 1 d2d1d0][0 0 0 0 n11 ~ n8][n7 ~ n0] 6008H
引数	Rd = 3bit(R select),Rs = 4bit(R select),n = 12bit(signed)
ワード数	2
サイクル数	3 or 4
機能	(Rd) ← Lobyte[(Rs±n)&FFFFh], (Rs) ← (Rs) + 1, (PC)←(PC)+4
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

Rsで指定する汎用レジスタの内容とnの演算結果\*<sup>1</sup>によってアドレス指定されるデータメモリ(RAM), 特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)の内容を、Rdの下位8bitに転送します。その後、Rsの内容を+1します。

なお、Rdで指定できる範囲はR0～R7、Rsで指定できる範囲はR0～R15、nで指定できる範囲は符号付12bitデータ(－2048～2047)になっています。

\*1:16ビット演算結果のキャリー・ボローは、無視されます。

#### 【例】

	RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

#### <注意事項>

Rdの上位8bitには、“00H”が入ります。

プログラムメモリ(ROM)の内容をRdに転送すると、4サイクル命令となります。

## MOV. B Rd, m16

命令コード	[1 0 X 0 d2d1d0 0][m7m6m5m4m3m2m1m0]	8000H(RAM),A000H(SFR)
引数	Rd = 3bit(R select),m16 = 16bit(Lower 8bit valid for operation code)	
ワード数	1	
サイクル数	1	
機能	(Rd) ← Lobyte (m16), (PC) ← (PC) + 2	
影響を受けるフラグ	Z8,Z16,P,S,N0～N3	

## 【説明】

m16で指定するデータメモリの内容を、Rdで指定する汎用レジスタの下位8bitに転送します。

m16(第2オペランドデータ)の値によって、コンパイラは転送先が RAM であるのか SFR であるのかを判断して命令コードを生成します。

- ・RAM を指定する場合には、m16の値は 00H～FFH(0000H～00FFH)の範囲で指定してください。この命令で 100H 番地以上の RAM を指定することはできません。
- ・SFR を指定する場合には、m16の値は 7F00H～7FFFH の範囲で指定してください。

なお、生成される命令コードの基本系は、それぞれ 8000H(RAM), A000H(SFR)になり、命令コードの動作に反映されるのは、m16の下位 8 ビットになります。

Rd(第1オペランドデータ)には、16ビットデータを指定することができますが、この命令はバイト転送を行う命令ですから、Rdの上位8ビットは実際の動作には無関係です。16ビットデータを扱いたい場合には、MOV. W命令を使用してください。

なお、Rdで指定できる範囲はR0～R7になっています。

## 【例】

	RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

## 命令

### MOV. B Rd, RxH

命令コード	[0 0 1 1 0 0 0 0][1 1 0 1 d3d2d1d0] 30D0H
引数	Rd = 4bit(R select)
ワード数	1
サイクル数	1
機能	(Rd) ← { Hbyte(Rx)   Lbyte(Rd) }, (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

PSW の bit12～15 (N0～N3)の値によって間接指定される汎用レジスタの上位8bit (RxH) の内容を、Rd の上位8bitに転送します。

なお、Rdで指定できる範囲はR0～R15になっています。

#### 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

命令コード	[0 0 1 1 0 0 0 0][1 1 0 0 d3d2d1d0]	30C0H
引数	Rd = 4bit(R select)	
ワード数	1	
サイクル数	1	
機能	(Rd) ← { Lobyte(Rx)   Hibyte(Rd) }, (PC)←(PC)+2	
影響を受けるフラグ	Z8,Z16,P,S,N0～N3	

## 【説明】

PSW の bit12～15 (N0～N3)の値によって間接指定される汎用レジスタの下位8bit (RxL) の内容を、Rd の下位8bitに転送します。

なお、Rdで指定できる範囲はR0～R15になっています。

## 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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,Rx1	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,Rx1	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,Rx1	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,Rx1	66h	66h	3456h	1200h	6789h	8956h	3	0	0	1	1

## 命 令

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

命令コード	[0 1 1 1 0 0 1 1][d3d2d1d0 0 s2s1s0]7300H
引数	Rd = 4bit(R select),Rs = 3bit(R select)
ワード数	1
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

Rdの内容によってアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)が、偶数番地の場合、Rsの下位8bitの内容を[Rd]に、Rsの上位8bitの内容を[Rd+1]に転送します。奇数番地の場合、Rsの上位8bitの内容を[Rd]に、Rsの下位8bitの内容を[Rd-1]に転送します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R7になっています。

#### 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

#### <注意事項>

Rsの内容をプログラムメモリ(ROM)に転送すると、3サイクル命令となります。但し、実際には“ROM”に転送することは出来ません。



## MOV[. W] (—Rd), Rs

命令コード	[0 1 1 0 1 0 1 1][d3d2d1d0 0 s2s1s0] 6B00H
引数	Rd = 4bit(R select), Rs = 3bit(R select)
ワード数	1
サイクル数	2 or 3
機能	(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
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

## 【説明】

最初に、Rdで指定する汎用レジスタの内容を減算(－2)します。

その後、Rdの内容によってアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)が、偶数番地の場合、Rsの下位8bitの内容を[Rd]に、Rsの上位8bitの内容を[Rd+1]に転送します。奇数番地の場合、Rsの上位8bitの内容を[Rd]に、Rsの下位8bitの内容を[Rd－1]に転送します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R7になっています。

## 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

## &lt;注意事項&gt;

Rsの内容をプログラムメモリ(ROM)に転送すると、3サイクル命令となります。但し、実際には“ROM”に転送することは出来ません。

## 命令

### MOV[. W] (Rd, ±n), Rs

命令コード	[0 1 1 1 0 0 1 1][d3d2d1d0 1 s2s1s0][0 0 0 0 n11 ~ n8][n7 ~ n0] 7308H
引数	Rd = 4bit(R select), n = 12bit(signed), Rs = 3bit(R select)
ワード数	2
サイクル数	3 or 4
機能	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
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容とnの演算結果\*1によってアドレス指定されるデータメモリ(RAM), 特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)が偶数番地の場合、Rsの下位8bitの内容を[(Rd±n)&FFFFh]に、Rsの上位8bitを[(Rd±n+1)&FFFFh]に転送します。奇数番地の場合、Rsの上位8bitの内容を[(Rd±n)&FFFFh]に、Rsの下位8bitを[(Rd±n-1)&FFFFh]に転送します。

なお、Rdで指定できる範囲はR0～R15、nで指定できる範囲は符号付12bitデータ(−2048～2047)、Rsで指定できる範囲はR0～R7になっています。

※1:16ビット演算結果のキャリー・ボローは、無視されます。

#### 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

#### <注意事項>

Rsの内容をプログラムメモリ(ROM)に転送すると、4サイクル命令となります。但し、実際には“ROM”に転送することは出来ません。

## MOV[. W] (—Rd, ±n), Rs

命令コード	[0 1 1 0 1 0 1 1][d3d2d1d0 1 s2s1s0][0 0 0 0 n11 ~ n8][n7 ~ n0] 6B08H
引数	Rd = 4bit(R select), n = 12bit(signed), Rs = 3bit(R select)
ワード数	2
サイクル数	3 or 4
機能	(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
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

## 【説明】

最初に、Rdによって指定される汎用レジスタの内容を減算(−2)します。

その後、Rdの内容とnの演算結果\*1によってアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)が偶数番地の場合、Rsの下位8bitの内容を[(Rd±n)&FFFFh]に、Rsの上位8bitを[(Rd±n+1)&FFFFh]に転送します。奇数番地の場合、Rsの上位8bitの内容を[(Rd±n)&FFFFh]に、Rsの下位8bitを[(Rd±n−1)&FFFFh]に転送します。

なお、Rdで指定できる範囲はR0～R15、nで指定できる範囲は符号付12bitデータ(−2048～2047)、Rsで指定できる範囲はR0～R7になっています。

※1:16ビット演算結果のキャリー・ボローは、無視されます。

## 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

## &lt;注意事項&gt;

Rsの内容をプログラムメモリ(ROM)に転送すると、4サイクル命令となります。但し、実際には“ROM”に転送することは出来ません。

## 命 令

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

命令コード	[0 1 1 0 0 0 1 1][d3d2d1d0 0 s2s1s0] 6300H
引数	Rd = 4bit(R select),Rs = 3bit(R select)
ワード数	1
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容によってアドレス指定されるデータメモリ(RAM) , 特殊機能レジスタ(SFR) 、またはプログラムメモリ(ROM) が偶数番地の場合、Rsの下位8bitの内容を[Rd]に、Rsの上位8bitの内容を[Rd+1]に転送します。奇数番地の場合、Rsの上位8bitの内容を[Rd]に、Rsの下位8bitの内容を[Rd-1]に転送します。その後、Rdの内容を+2します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R7になっています。

#### 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

#### <注意事項>

Rsの内容をプログラムメモリ(ROM)に転送すると、3サイクル命令となります。但し、実際には“ROM”に転送することは出来ません。

## MOV[. W] (Rd++, ±n), Rs

命令コード	[0 1 1 0 0 0 1 1][d3d2d1d0 1 s2s1s0][0 0 0 0 n11 ~ n8][n7 ~ n0] 6308H
引数	Rd = 4bit(R select), n = 12bit(signed), Rs = 3bit(R select)
ワード数	2
サイクル数	3 or 4
機能	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
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

## 【説明】

Rdで指定する汎用レジスタの内容とnの演算結果\*1によってアドレス指定されるデータメモリ(RAM), 特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)が偶数番地の場合、Rsの下位8bitの内容を[(Rd±n)&FFFFh]に、Rsの上位8bitを[(Rd±n+1)&FFFFh]に転送します。奇数番地の場合、Rsの上位8bitの内容を[(Rd±n)&FFFFh]に、Rsの下位8bitを[(Rd±n-1)&FFFFh]に転送します。その後、Rdの内容を+2します。

なお、Rdで指定できる範囲はR0～R15、nで指定できる範囲は符号付12bitデータ(−2048～2047)、Rsで指定できる範囲はR0～R7になっています。

\*1:16ビット演算結果のキャリー・ボローは、無視されます。

## 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

## &lt;注意事項&gt;

Rsの内容をプログラムメモリ(ROM)に転送すると、4サイクル命令となります。但し、実際には“ROM”に転送することは出来ません。

## 命令

### MOV[.W]    m16, # imm16

命令コード	[0 1 1 1 0 X 1][m7m6m5m4m3m2m1m0][i15 ~ i8][i7 ~ i0] 7900H(RAM), 7B00H(SFR)
引数	m16 = 16bit (lower 8bit valid for operation code) imm16 = 16bit (immediate data)
ワード数	2
サイクル数	2
機能	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
影響を受けるフラグ	Z8,Z16,P,S

#### 【説明】

イミディエイトデータ(imm16, 16ビット)を、m16でアドレス指定する2バイト分の RAM(データメモリ)または SFR(内蔵する周辺機能を制御する専用レジスタ群)に転送します。

転送先の2バイト分のアドレスは、下記のルールに従って決まります。

- m16 の値が偶数の場合、奇数番地(m16+1)に imm16 の上位 8 ビットが転送され、偶数番地(m16)に下位 8 ビットが転送されます。
- m16 の値が奇数の場合、奇数番地(m16)に imm16 の上位 8 ビットが転送され、偶数番地(m16-1)に下位 8 ビットが転送されます。

m16(第 1 オペランドデータ)の値によって、コンパイラは転送先が RAM であるのか SFR であるのかを判断して命令コードを生成します。

- RAM を指定する場合には、m16の値は 00H～FFH(0000H～00FFH)の範囲で指定してください。この命令で 100H 番地以上の RAM を指定することはできません。
- SFR を指定する場合には、m16の値は 7F00H～7FFFH の範囲で指定してください。

なお、生成される命令コードの基本系は、それぞれ 7900H(RAM), 7B00H(SFR)になり、命令コードの動作に反映されるのは、m16の下位 8 ビットになります。

#### 【例】

		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

命令コード	[1 0 X 1 s2s1s0 1][m7m6m5m4m3m2m1m0]     9100H(RAM), B100H(SFR)
引数	m16 = 16bit (lower 8bit valid for operation code) Rs = 3bit (R select)
ワード数	1
サイクル数	1
機能	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
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

## 【説明】

Rs で指定する汎用レジスタの内容(16ビット)を、m16でアドレス指定する2バイト分の RAM(データメモリ)または SFR(内蔵する周辺機能を制御する専用レジスタ群)に転送します。なお、Rs で指定できる範囲は、R0～R7 になっています。

転送先の2バイト分のアドレスは、下記のルールに従って決まります。

- m16 の値が偶数の場合、奇数番地(m16+1)に Rs の内容の上位 8 ビットが転送され、偶数番地(m16)に下位 8 ビットが転送されます。
- m16 の値が奇数の場合、奇数番地(m16)に Rs の内容の上位 8 ビットが転送され、偶数番地(m16-1)に下位 8 ビットが転送されます。

m16(第 1 オペランドデータ)の値によって、コンパイラは転送先が RAM であるのか SFR であるのかを判断して命令コードを生成します。

- RAM を指定する場合には、m16の値は 00H～FFH(0000H～00FFH)の範囲で指定してください。この命令で 100H 番地以上の RAM を指定することはできません。
- SFR を指定する場合には、m16の値は 7F00H～7FFFH の範囲で指定してください。

なお、生成される命令コードの基本系は、それぞれ 9100H(RAM), B100H(SFR)になり、命令コードの動作に反映されるのは、m16の下位 8 ビットになります。

## 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

## 命令

### MOV[. W] Rd, #imm8

命令コード	[0 0 1 0 d2d1d0 1][i7i6i5i4i3i2i1i0]	2100H
引数	Rd = 3bit(R select), imm8 = 8bit(immediate data)	
ワード数	1	
サイクル数	1	
機能	(Rd) ← 16bit data(Hibyte=00H, Lobyte=#imm8), (PC)←(PC)+2	
影響を受けるフラグ	Z8,Z16,P,S,N0～N3	

#### 【説明】

imm8で指定するイミディエイトデータを、Rdで指定する汎用レジスタに転送します。

なお、Rdで指定できる範囲はR0～R7、imm8で指定できる範囲は0～FFになっています。

#### 【例】

	R0	R1	R2	R3	N3～ 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

#### <注意事項>

Rdの上位8bitには、“00H”が入ります。



## MOV[. W] Rd, #imm16

命令コード	[0 0 1 1 0 0 0 1][0 0 1 1 d3d2d1d0][i15 ~ i8][i7 ~ i0]	3130H
引数	Rd = 4bit(R select), imm16 = 16bit(immediate data)	
ワード数	2	
サイクル数	2	
機能	(Rd)←#imm16, (PC)←(PC)+4	
影響を受けるフラグ	Z8,Z16,P,S,N0～N3	

## 【説明】

imm16で指定するイミディエイトデータを、Rdで指定する汎用レジスタに転送します。

なお、Rdで指定できる範囲はR0～R15、imm16で指定できる範囲は0～FFFFになっています。

## 【例】

	R0	R1	R2	R3	N3～ 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

## 命令

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

命令コード	[0 1 1 1 0 0 0 1][s3s2s1s0 0 d2d1d0]7100H
引数	Rd = 3bit(R select),Rs = 4bit(R select)
ワード数	1
サイクル数	2 or 3
機能	if (Rs) =even data : Hbyte(Rd)←[Rs+1], Lbyte(Rd)← [Rs] if (Rs) =Odd data : Hbyte(Rd)←[Rs], Lbyte [Rs-1] (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

Rsで指定する汎用レジスタの内容によってアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)が偶数番地の場合、その内容をRdの下位8bitに、[Rs+1]の内容をRdの上位8bitに転送します。奇数番地の場合、[Rs]の内容をRdの上位8bitに、[Rs-1]の内容をRdの下位8bitに転送します。  
なお、Rdで指定できる範囲はR0～R7、Rsで指定できる範囲はR0～R15になっています。

#### 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

#### <注意事項>

プログラムメモリ(ROM)の内容をRdに転送すると、3サイクル命令となります。

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

命令コード	[0 1 1 0 1 0 0 1][s3s2s1s0 0 d2d1d0] 6900H
引数	Rd = 3bit(R select),Rs = 4bit(R select)
ワード数	1
サイクル数	2 or 3
機能	(Rs)←(Rs)-2 if (Rs) =even data : Hbyte(Rd)←[Rs+1], Lbyte(Rd)← [Rs] if (Rs) = odd data : Hbyte(Rd)←[Rs], Lbyte(Rd)← [Rs-1] (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

## 【説明】

最初に、Rsで指定する汎用レジスタの内容を減算(−2)します。

その後、Rsの内容によってアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)が偶数番地の場合、その内容をRdの下位8bitに、[Rs+1]の内容をRdの上位8bitに転送します。奇数番地の場合、[Rs]の内容をRdの上位8bitに、[Rs−1]の内容をRdの下位8bitに転送します。

なお、Rdで指定できる範囲はR0～R7、Rsで指定できる範囲はR0～R15になっています。

## 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

## &lt;注意事項&gt;

プログラムメモリ(ROM)の内容をRdに転送すると、3サイクル命令となります。

## 命令

### MOV[. W] Rd, (Rs, ±n)

命令コード	[0 1 1 1 0 0 0 1][s3s2s1s0 1 d2d1d0][0 0 0 0 n11 ~ n8][n7 ~ n0] 7108H
引数	Rd = 3bit(R select),Rs = 4bit(R select),n = 12bit(signed)
ワード数	2
サイクル数	3 or 4
機能	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
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

Rsで指定する汎用レジスタの内容とnの演算結果\*1によってアドレス指定されるデータメモリ(RAM), 特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)が偶数番地の場合、その内容をRdの下位8bitに、[(Rs±n+1)&FFFFh]の内容をRdの上位8bitに転送します。奇数番地の場合、[(Rs±n)&FFFFh]の内容をRdの上位8bitに、[(Rs±n-1)&FFFFh]の内容をRdの下位8bitに転送します。  
 なお、Rdで指定できる範囲はR0～R7、Rsで指定できる範囲はR0～R15、nで指定できる範囲は符号付12bitデータ(－2048～2047)になっています。

\*1:16ビット演算結果のキャリー・ボローは、無視されます。

#### 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

#### <注意事項>

プログラムメモリ(ROM)の内容をRdに転送すると、4サイクル命令となります。

## MOV[. W] Rd, ( --Rs, ±n)

命令コード	[0 1 1 0 1 0 0 1][s3s2s1s0 1 d2d1d0][0 0 0 0 n11 ~ n8][n7 ~ n0] 6908H
引数	Rd = 3bit(R select),Rs = 4bit(R select),n = 12bit(signed)
ワード数	2
サイクル数	3 or 4
機能	(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
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

## 【説明】

最初にRsで指定する汎用レジスタの内容を減算(−2)します。

その後、Rsの内容とnの演算結果\*1によってアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)が偶数番地の場合、その内容をRdの下位8bitに、[(Rs±n+1)&FFFFh]の内容をRdの上位8bitに転送します。奇数番地の場合、[(Rs±n)&FFFFh]の内容をRdの上位8bitに、[(Rs±n−1)&FFFFh]の内容をRdの下位8bitに転送します。

なお、Rdで指定できる範囲はR0～R7、Rsで指定できる範囲はR0～R15、nで指定できる範囲は符号付12bitデータ(−2048～2047)になっています。

\*1:16ビット演算結果のキャリー・ボローは、無視されます。

## 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

## &lt;注意事項&gt;

プログラムメモリ(ROM)の内容をRdに転送すると、4サイクル命令となります。

## 命令

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

命令コード	[0 1 1 0 0 0 1][s3s2s1s0 0 d2d1d0]6100H
引数	Rd = 3bit(R select),Rs = 4bit(R select)
ワード数	1
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

Rsで指定する汎用レジスタの内容によってアドレス指定されるデータメモリ(RAM), 特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)が偶数番地の場合、その内容をRdの下位8bitに、[Rs+1]の内容をRdの上位8bitに転送します。奇数番地の場合、[Rs]の内容をRdの上位8bitに、[Rs-1]の内容をRdの下位8bitに転送します。その後、Rsの内容を+2します。

なお、Rdで指定できる範囲はR0～R7、Rsで指定できる範囲はR0～R15になっています。

#### 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ N0	Z8	Z16	P	S
MOV.W	R3,#0x0050	-	-	-	-	-	-	-	-	-	-	-
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

#### <注意事項>

プログラムメモリ(ROM)の内容をRdに転送すると、3サイクル命令となります。

## MOV[. W] Rd, (Rs++, ±n)

命令コード	[0 1 1 0 0 0 0 1][s3s2s1s0 1 d2d1d0][0 0 0 0 n11 ~ n8][n7 ~ n0] 6108H
引数	Rd = 3bit(R select), Rs = 4bit(R select), n = 12bit(signed)
ワード数	2
サイクル数	3 or 4
機能	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
影響を受けるフラグ	Z8, Z16, P, S, N0 ~ N3

## 【説明】

Rsで指定する汎用レジスタの内容とnの演算結果\*<sup>1</sup>によってアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)が偶数番地の場合、その内容をRdの下位8bitに、[(Rs±n+1) & FFFFh]の内容をRdの上位8bitに転送します。奇数番地の場合、[(Rs±n) & FFFFh]の内容をRdの上位8bitに、[(Rs±n-1) & FFFFh]の内容をRdの下位8bitに転送します。その後、Rsの内容を+2します。

なお、Rdで指定できる範囲はR0~R7、Rsで指定できる範囲はR0~R15、nで指定できる範囲は符号付12bitデータ(-2048~2047)になっています。

\* 1: 16ビット演算結果のキャリー・ボローは、無視されます。

## 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3~ N0	Z8	Z16	P	S
MOV.W	R3, #0x0000	-	-	-	-	-	-	-	-	-	-	-
MOV.W	0x50, #0x5555	55h	55h	-	-	-	0000h	3	1	1	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

## &lt;注意事項&gt;

プログラムメモリ(ROM)の内容をRdに転送すると、4サイクル命令となります。

## 命令

### MOV[. W] Rd, m16

命令コード	[1 0 X 0 d2d1d0 1][m7m6m5m4m3m2m1m0] 8100H(RAM),A100H(SFR)
引数	Rd = 3bit(R select),m16 = 16bit(Lower 8bit valid for operation code)
ワード数	1
サイクル数	1
機能	if “m16” is even : Hbyte(Rd)←(m16+1), Lbyte(Rd)←(m16) if “m16” is odd : Hbyte(Rd)←(m16), Lbyte(Rd)←(m16-1) (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

m16で指定するRAM(データメモリ)またはSFR(内蔵する周辺機能を制御する専用のレジスタ群)の内容をRdで指定する汎用レジスタの下位8bitに転送します。

なお、Rdで指定できる範囲はR0～R7となっています。

転送先の2バイト分のアドレスは、下記のルールに従って決まります。

- m16の値が偶数の場合、奇数番地(m16+1)の内容をRdの上位8ビットが転送され、偶数番地(m16)に下位8ビットが転送されます。
- m16の値が奇数の場合、奇数番地(m16)の内容をRdの上位8ビットが転送され、偶数番地(m16-1)の内容を下位8ビットが転送されます。

m16(第2オペランドデータ)の値によって、コンパイラは転送先がRAMであるのかSFRであるのかを判断して命令コードを生成します。

- RAMを指定する場合には、m16の値は00H～FFH(0000H～00FFH)の範囲で指定してください。この命令で100H番地以上のRAMを指定することはできません。
- SFRを指定する場合には、m16の値は7F00H～7FFFHの範囲で指定してください。

なお、生成される命令コードの基本系は、それぞれ8100H(RAM), A100H(SFR)になり、命令コードの動作に反映されるのは、m16の下位8ビットになります。

#### 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

命令コード	[0 1 0 0 0 1 1 1][i7i6i5i4i3i2i1i0]	4700H
引数	imm8 = 8bit(immediate data)	
ワード数	1	
サイクル数	1	
機能	(Rx)←16bit data(Hibyte=00H, Lobyte=#imm8), (PC)←(PC)+2	
影響を受けるフラグ	Z8,Z16,P,S	

## 【説明】

imm8で指定するイミディエイトデータを、PSW の bit12～15 (N0～N3)の値によって間接指定される汎用レジスタ(Rx)に転送します。

なお、imm8で指定できる範囲は、0～FFになっています。

## 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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

## &lt;注意事項&gt;

Rxの上位8bitには、“00H”が入ります。

## 命令

### MOV.F.B (R<sub>b</sub>, R<sub>d</sub>, ±n), R<sub>s</sub>

命令コード	[0 1 1 1 0 1 1 0][d3d2d1d0 1 s2s1s0][0 b2b1b0 n11 ~ n8][n7 ~ n0] 7608H
引数	Rb = 3bit(Rb select), Rd = 4bit(R select), n = 12bit(signed), Rs = 3bit(R select)
ワード数	2
サイクル数	3 or 4
機能	[Rb<<16+Rd±n]←Lobyte (Rs) , (PC)←(PC)+4
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

Rsで指定する汎用レジスタの下位8bitの内容を、ベースレジスタ(Rb)の内容が上位16bit、Rdの内容とnの演算結果\*<sup>1</sup>が下位16bitの32bitでアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)に転送します。

なお、Rdで指定できる範囲はR0～R15、Rbで指定できる範囲はR8～R13、Rsで指定できる範囲はR0～R7、nで指定できる範囲は符号付12bitデータ(−2048～2047)となっています。

\*1: 下位16ビット演算結果のキャリー・ボローは、上位16ビットに反映されます。

#### 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	R8	N3 ～ 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
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.B	(R8,R3,0x50),R2	55h	66h	5634h	1881h	5555h	0000h	0000h	2	0	0	0	0

#### <注意事項>

Rsの内容をプログラムメモリ(ROM)に転送すると、4サイクル命令となります。但し、実際には“ROM”に転送することは出来ません。

MOV.F.B (R<sub>b</sub>, --R<sub>d</sub>, ±n), R<sub>s</sub>

命令コード	[0 1 1 0 1 1 0][d3d2d1d0 1 s2s1s0][0 b2b1b0 n11 ~ n8][n7 ~ n0] 6E08H
引数	Rb = 3bit(Rb select), Rd = 4bit(R select), n = 12bit(signed), Rs = 3bit(R select)
ワード数	2
サイクル数	3 or 4
機能	(Rd)←(Rd)-1, if Borrow : (Rb)←(Rb)-1 [Rb<<16+Rd±n]←Lobyte(Rs) (PC)←(PC)+4
影響を受けるフラグ	Z8,Z16,P,S,N0~N3

## 【説明】

最初に、Rdで指定する汎用レジスタの内容を減算(−1)します。Rdの減算結果にボローが発生する場合は、Rbがデクリメントされます。

その後、Rsの下位8bitの内容を、ベースレジスタ(Rb)の内容が上位16bit、Rdの内容とnの演算結果<sup>\*1</sup>が下位16bitの32bitでアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)に転送します。

なお、Rdで指定できる範囲はR0~R15、Rbで指定できる範囲はR8~R13、Rsで指定できる範囲はR0~R7、nで指定できる範囲は符号付12bitデータ(−2048~2047)となっています。

\*1: 下位16ビット演算結果のキャリー・ボローは、上位16ビットに反映されます。

## 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	R8	N3 ~ 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
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.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

## &lt;注意事項&gt;

Rsの内容をプログラムメモリ(ROM)に転送すると、4サイクル命令となります。但し、実際には“ROM”に転送することは出来ません。

## 命令

### MOV.F.B (Rb, Rd++, ±n), Rs

命令コード	[0 1 1 0 0 1 1 0][d3d2d1d0 1 s2s1s0][0 b2b1b0 n11 ~ n8][n7 ~ n0] 6608H
引数	Rb = 3bit(Rb select), Rd = 4bit(R select), n = 12bit(signed), Rs = 3bit(R select)
ワード数	2
サイクル数	3 or 4
機能	[Rb<<16+Rd±n] ← Lobyte (Rs) (Rd)←(Rd)+1, if Carry : (Rb)←(Rb)+1 (PC)←(PC)+4
影響を受けるフラグ	Z8,Z16,P,S,N0~N3

#### 【説明】

Rsで指定する汎用レジスタの下位8bitの内容を、ベースレジスタ(Rb)の内容が上位16bit, Rdの内容とnの演算結果\*<sup>1</sup>が下位16bitの32bitでアドレス指定されるデータメモリ(RAM), 特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)に転送します。その後、Rdの内容を+1します。Rdの加算結果にキャリーが発生する場合、Rbがインクリメントされます。

なお、Rdで指定できる範囲はR0~R15、Rbで指定できる範囲はR8~R13、Rsで指定できる範囲はR0~R7、nで指定できる範囲は符号付12bitデータ(−2048~2047)となっています。

\*1: 下位16ビット演算結果のキャリー・ボローは、上位16ビットに反映されます。

#### 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	R8	N3 ~ 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.F.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
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.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

#### <注意事項>

Rsの内容をプログラムメモリ(ROM)に転送すると、4サイクル命令となります。但し、実際には“ROM”に転送することは出来ません。

MOV.F.B (R<sub>d</sub>), R<sub>s</sub>

命令コード	[0 1 1 1 0 1 1 0][d3d2d1d0 0 s2s1s0] 7600H
引数	Rd = 4bit(R select),Rs = 3bit(R select)
ワード数	1
サイクル数	2 or 3
機能	[R8<<16+Rd]←Lobyte(Rs) , (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

## 【説明】

Rsで指定する汎用レジスタの下位8bitの内容を、R8(Rb0)の内容が上位16bit、Rdの内容が下位16bitの32bitでアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)に転送します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R7となっています。

## 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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.F.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.F.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.F.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.F.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.F.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.F.B	(R3),R2	81h	66h	5634h	1881h	5555h	0050h	2	0	0	0	0

## &lt; 注意事項 &gt;

この場合、Rb0は、R8を示します。

Rsの内容をプログラムメモリ(ROM)に転送すると、3サイクル命令となります。但し、実際には“ROM”に転送することは出来ません。

## 命令

### MOV.F.B ( -- Rd ), Rs

命令コード	[0 1 1 0 1 1 1 0][d3d2d1d0 0 s2s1s0]6E00H
引数	Rd = 4bit(R select),Rs = 3bit(R select)
ワード数	1
サイクル数	2 or 3
機能	(Rd)←(Rd)-1, if Borrow : (R8)←(R8)-1 [R8<<16+Rd]←Lobyte(Rs) (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

最初に、Rdで指定する汎用レジスタの内容を減算(－1)します。Rdの減算結果にボローが発生する場合は、R8がデクリメントされます。

その後、Rsの下位8bitの内容を、R8(Rb0)の内容が上位16bit、Rdの内容が下位16bitの32bitでアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)に転送します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R7となっています。

#### 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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.F.B	(--R3),R0	55h	66h	5555h	-	-	0050h	0	0	0	0	0
MOV.W	R3	55h	66h	5555h	-	-	0051h	3	0	0	1	0
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.B	(--R3),R1	81h	66h	5634h	1881h	0000h	0050h	1	0	0	0	1

#### <注意事項>

この場合、Rb0は、R8を示します。

Rsの内容をプログラムメモリ(ROM)に転送すると、3サイクル命令となります。但し、実際には“ROM”に転送することは出来ません。

## MOV.F.B (Rd++), Rs

命令コード	[0 1 1 0 0 1 1 0][d3d2d1d0 0 s2s1s0] 6600H
引数	Rd = 4bit(R select),Rs = 3bit(R select)
ワード数	1
サイクル数	2 or 3
機能	[R8<<16+Rd]←Lobyte(Rs) (Rd)←(Rd)+1, if Carry : (R8)←(R8)+1 (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

## 【説明】

Rsで指定する汎用レジスタの下位8bitの内容を、R8(Rb0)の内容が上位16bit、Rdの内容が下位16bitの32bitでアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)に転送します。その後、Rdの内容を+1します。Rdの加算結果にキャリーが発生する場合、R8がインクリメントされます。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R7となっています。

## 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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.F.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.F.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.F.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.F.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.F.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.F.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

## &lt;注意事項&gt;

この場合、Rb0は、R8を示します。

Rsの内容をプログラムメモリ(ROM)に転送すると、3サイクル命令となります。但し、実際には“ROM”に転送することは出来ません。

## 命令

### MOVF. B Rd, (Rb, Rs, ±n)

命令コード	[0 1 1 1 0 1 0 0][s3s2s1s0 1 d2d1d0][0 b2b1b0 n11 ~ n8][n7 ~ n0] 7408H
引数	Rd = 3bit(R select), Rb = 3bit(Rb select), Rs = 4bit(R select), n = 12bit(signed)
ワード数	2
サイクル数	3 or 4
機能	Lobyte (Rd)← [Rb<<16+Rs±n], (PC)←(PC)+4
影響を受けるフラグ	Z8,Z16,P,S,N0~N3

#### 【説明】

ベースレジスタ(Rb)によって指定される汎用レジスタの内容が上位16bit, Rsの内容とnの演算結果\*1が下位16bitの32bitでアドレス指定されるデータメモリ(RAM), 特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)の内容をRdの下位8bitに転送します。

なお、Rdで指定できる範囲はR0~R7、Rbで指定できる範囲はR8~R13、Rsで指定できる範囲はR0~R15、nで指定できる範囲は符号付12bitデータ(−2048~2047)となっています。

\*1: 下位16ビット演算結果のキャリー・ボローは、上位16ビットに反映されます。

#### 【例】

	RAM (50h)	RAM (51h)	R0	R1	R2	R3	R8	N3 ~ 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

#### <注意事項>

Rdの上位8bitには、“00H”が入ります。

プログラムメモリ(ROM)の内容をRdに転送すると、4サイクル命令となります。



MOV.F.B  $R_d, (R_b, --R_s, \pm n)$ 

命令コード	[0 1 1 0 1 1 0 0][s3s2s1s0 1 d2d1d0][0 b2b1b0 n11 ~ n8][n7 ~ n0] 6C08H
引数	Rd = 3bit(R select), Rb = 3bit(Rb select), Rs = 4bit(R select), n = 12bit(signed)
ワード数	2
サイクル数	3 or 4
機能	(Rs)←(Rs)-1, if Borrow : (Rb)←(Rb)-1 Lobyte (Rd)← [Rb<<16+Rs±n] (PC)←(PC)+4
影響を受けるフラグ	Z8,Z16,P,S,N0~N3

## 【説明】

最初に、Rsによって指定される汎用レジスタの内容を減算(−1)します。Rsの減算結果にボローが発生する場合は、Rbがデクリメントされます。

その後、ベースレジスタ(Rb)の内容が上位16bit、Rsの内容とnの演算結果\*1が下位16bitの32bitでアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)の内容を、Rdの下位8bitに転送します。

なお、Rdで指定できる範囲はR0~R7、Rbで指定できる範囲はR8~R13、Rsで指定できる範囲はR0~R15、nで指定できる範囲は符号付12bitデータ(−2048~2047)となっています。

\*1: 下位16ビット演算結果のキャリー・ボローは、上位16ビットに反映されます。

## 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	R8 ~ 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.F.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
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.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

## &lt;注意事項&gt;

Rdの上位8bitには、“00H”が入ります。

プログラムメモリ(ROM)の内容をRdに転送すると、4サイクル命令となります。

## 命令

### MOV.F.B Rd, (Rb, Rs++, ±n)

命令コード	[0 1 1 0 0 1 0 0][s3s2s1s0 1 d2d1d0][0 b2b1b0 n11 ~ n8][n7 ~ n0] 6408H
引数	Rd = 3bit(R select), Rb = 3bit(Rb select), Rs = 4bit(R select), n = 12bit(signed)
ワード数	2
サイクル数	3 or 4
機能	Lobyte (Rd) ← [Rb << 16 + Rs ± n] (Rs) ← (Rs) + 1, if Carry : (Rb) ← (Rb) + 1 (PC) ← (PC) + 4
影響を受けるフラグ	Z8, Z16, P, S, N0 ~ N3

#### 【説明】

ベースレジスタ(Rb)によって指定される汎用レジスタの内容が上位16bit, Rsの内容とnの演算結果\*1が下位16bitの32bitでアドレス指定されるデータメモリ(RAM), 特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)の内容をRdの下位8bitに転送します。その後、Rsの内容を+1します。Rsの加算結果にキャリーが発生する場合、Rbがインクリメントされます。

なお、Rdで指定できる範囲はR0～R7、Rbで指定できる範囲はR8～R13、Rsで指定できる範囲はR0～R15、nで指定できる範囲は符号付12bitデータ(−2048～2047)となっています。

\*1: 下位16ビット演算結果のキャリー・ボローは、上位16ビットに反映されます。

#### 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	R8	N3 ~ 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.F.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
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.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

#### <注意事項>

Rdの上位8bitには、“00H”が入ります。

プログラムメモリ(ROM)の内容をRdに転送すると、4サイクル命令となります。

## MOV.F.B Rd, (Rs)

命令コード	[0 1 1 1 0 1 0 0][s3s2s1s0 0 d2d1d0] 7400H
引数	Rd = 3bit(R select),Rs = 4bit(R select)
ワード数	1
サイクル数	2 or 3
機能	Lobyte (Rd)← [R8<<16+Rs] , (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

## 【説明】

R8(Rb0)の内容が上位16bit, Rsによって指定される汎用レジスタの内容が、下位16bitの32bitでアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)の内容をRdの下位8bitに転送します。

なお、Rdで指定できる範囲はR0～R7、Rsで指定できる範囲はR0～R15となっています。

## 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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.F.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.F.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.F.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.F.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.F.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.F.B	R2,(R3)	55h	55h	0034h	0081h	0055h	0050h	2	0	0	0	0

## &lt;注意事項&gt;

この場合、Rb0は、R8を示します。

Rdの上位8bitには、“00H”が入ります。

プログラムメモリ(ROM)の内容をRdに転送すると、3サイクル命令となります。

## 命令

### MOV.F.B Rd, ( --Rs)

命令コード	[0 1 1 0 1 1 0 0][s3s2s1s0 0 d2d1d0]6C00H
引数	Rd = 3bit(R select),Rs = 4bit(R select)
ワード数	1
サイクル数	2 or 3
機能	(Rs)←(Rs)-1 , if Borrow : (R8)←(R8)-1 Lobyte (Rd)← [R8<<16+Rs] (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

Rsで指定する汎用レジスタの内容を減算(－1)します。Rsの減算結果にボローが発生する場合は、R8がデクリメントされます。

その後、R8(Rb0)の内容が上位16bit、Rsの内容が下位16bitの32bitでアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)の内容をRdの下位8bitに転送します。

なお、Rdで指定できる範囲はR0～R7、Rsで指定できる範囲はR0～R15となっています。

#### 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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.F.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
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.B	R1,(--R3)	81h	18h	0034h	0081h	0000h	0050h	1	0	0	0	1

#### <注意事項>

この場合、Rb0は、R8を示します。

Rdの上位8bitには、“00H”が入ります。

プログラムメモリ(ROM)の内容をRdに転送すると、3サイクル命令となります。

MOV.F.B Rd, (Rs++)

命令コード	[0 1 1 0 0 1 0 0][s3s2s1s0 0 d2d1d0] 6400H
引数	Rd = 3bit(R select),Rs = 4bit(R select)
ワード数	1
サイクル数	2 or 3
機能	Lobyte (Rd) ← [R8<<16+Rs] (Rs)←(Rs)+1 , if Carry : (R8)←(R8)+1 (PC) ← (PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

## 【説明】

R8(Rb0)の内容が上位16bit, Rsによって指定される汎用レジスタの内容が、下位16bitの32bitでアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)の内容をRdの下位8bitに転送します。その後、Rsの内容を+1します。Rsの加算結果にキャリーが発生する場合、R8がインクリメントされます。

なお、Rdで指定できる範囲はR0～R7、Rsで指定できる範囲はR0～R15となっています。

## 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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.F.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.F.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.F.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.F.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
MOV.F.B	R1,(R3++)	81h	18h	0034h	0081h	0000h	0051h	1	0	0	0	1

## &lt;注意事項&gt;

この場合、Rb0は、R8を示します。

Rdの上位8bitには、“00H”が入ります。

プログラムメモリ(ROM)の内容をRdに転送すると、3サイクル命令となります。

MOVF[. W] (Rb, Rd, ±n), Rs

5-164

MOV.F[. W] (R<sub>b</sub>, --R<sub>d</sub>, ±n), R<sub>s</sub>

命令コード	[0 1 1 0 1 1 1 1][d3d2d1d0 1 s2s1s0][0 b2b1b0 n11 ~ n8][n7 ~ n0] 6F08H
引数	Rb = 3bit(Rb select), Rd = 4bit(R select), n = 12bit(signed), Rs = 3bit(R select)
ワード数	2
サイクル数	3 or 4
機能	(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
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

## 【説明】

Rdによって指定される汎用レジスタの内容を減算(−2)します。Rdの減算結果にボローが発生する場合は、Rbがデクリメントされます。

その後、ベースレジスタ(Rb)の内容が上位16bit、Rdの内容とnの演算結果\*1が下位16bitの32bitでアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)が偶数番地の場合、Rsの下位8bitの内容を[Rb<<16+Rd±n]に、Rsの上位8bitを[Rb<<16+Rd±n+1]に転送します。奇数番地の場合、Rsの上位8bitの内容を[Rb<<16+Rd±n]に、Rsの下位8bitを[Rb<<16+Rd±n-1]に転送します。

なお、Rdで指定できる範囲はR0～R15、Rbで指定できる範囲はR8～R13、Rsで指定できる範囲はR0～R7、nで指定できる範囲は符号付12bitデータ(−2048～2047)となっています。

\*1: 下位16ビット演算結果のキャリー・ボローは、上位16ビットに反映されます。

## 【例】

	RAM (50h)	RAM (51h)	R0	R1	R2	R3	R8	N3 ～ 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.F.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
MOV.F.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
MOV.F.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
MOV.F.W (R8,--R3,0x50),R0	34h	56h	5634h	1200h	0000h	0000h	0000h	0	0	0	1	0

## &lt;注意事項&gt;

Rsの内容をプログラムメモリ(ROM)に転送すると、4サイクル命令となります。但し、実際には“ROM”に転送することは出来ません。

## 命令

MOVFB[. W] (Rb, Rd++, ±n), Rs

命令コード	[0 1 1 0 0 1 1 1][d3d2d1d0 1 s2s1s0][0 b2b1b0 n11 ~ n8][n7 ~ n0] 6708H
引数	Rb = 3bit(Rb select), Rd = 4bit(R select), n = 12bit(signed), Rs = 3bit(R select)
ワード数	2
サイクル数	3 or 4
機能	<div> <div>if (Rd±n) = even data : [Rb&lt;&lt;16+Rd±n+1]</div> <div>←Hibyte (Rs),</div> <div>[Rb&lt;&lt;16+Rd±n]</div> <div>←Lobyte(Rs)</div> <div>if (Rd±n) = odd data : [Rb&lt;&lt;16+Rd±n]</div> <div>←Hibyte(Rs),</div> <div>[Rb&lt;&lt;16+Rd±n-1]</div> <div>←Lobyte(Rs)</div> <div>(Rd)←(Rd)+2 , if Carry : (Rb)←(Rb)+1</div> <div>(PC)←(PC)+4</div> </div>
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

【説明】

ベースレジスタ(Rb)によって指定される汎用レジスタの内容が上位16bit、Rdの内容とnの演算結果\*1が下位16bitの32bitでアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)が偶数番地の場合、Rsの下位8bitの内容を[Rb<<16+Rd±n]に、Rsの上位8bitを[Rb<<16+Rd±n+1]に転送します。奇数番地の場合、Rsの上位8bitの内容を[Rb<<16+Rd±n]に、Rsの下位8bitを[Rb<<16+Rd±n-1]に転送します。その後、Rdの内容を+2します。Rdの加算結果にキャリーが発生する場合、Rbがインクリメントされます。

なお、Rdで指定できる範囲はR0～R15、Rbで指定できる範囲はR8～R13、Rsで指定できる範囲はR0～R7、nで指定できる範囲は符号付12bitデータ(－2048～2047)となっています。

\*1: 下位16ビット演算結果のキャリー・ボローは、上位16ビットに反映されます。

【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	R8	N3 ~ 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

＜注意事項＞

Rsの内容をプログラムメモリ(ROM)に転送すると、4サイクル命令となります。但し、実際には“ROM”に転送することは出来ません。



MOV.F[.W] (R<sub>d</sub>), R<sub>s</sub>

命令コード	[0 1 1 1 0 1 1 1][d3d2d1d0 0 s2s1s0] 7700H
引数	Rd = 4bit(R select),Rs = 3bit(R select)
ワード数	1
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

## 【説明】

R8(Rb0)の内容が上位16bit, Rdで指定する汎用レジスタの内容が下位16bitの32bitでアドレス指定されるデータメモリ(RAM), 特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)が偶数番地の場合、Rsの下位8bitの内容を[R8<<16+Rd]に、Rsの上位8bitの内容を[R8<<16+Rd+1]に転送します。奇数番地の場合、Rsの上位8bitの内容を[R8<<16+Rd]に、Rsの下位8bitの内容を[R8<<16+Rd-1]に転送します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R7となっています。

## 【例】

	RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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.F.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.F.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.F.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.F.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.F.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.F.W (R3),R2	34h	56h	5634h	1200h	0000h	0050h	2	0	0	0	0

## &lt;注意事項&gt;

この場合、Rb0は、R8を示します。

Rsの内容をプログラムメモリ(ROM)に転送すると、3サイクル命令となります。但し、実際には“ROM”に転送することは出来ません。

## 命 令

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

命令コード	[0 1 1 0 1 1 1 1][d3d2d1d0 0 s2s1s0]6F00H
引数	Rd = 4bit(R select),Rs = 3bit(R select)
ワード数	1
サイクル数	2 or 3
機能	(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
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容を減算(−2)します。Rdの減算結果にボローが発生する場合は、R8 がデクリメントされます。

その後、R8(Rb0)の内容が上位16bit、Rdの内容が下位16bitの32bitでアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)が偶数番地の場合、Rsの下位8bitの内容を[R8<<16+Rd]に、Rsの上位8bitの内容を[R8<<16+Rd+1]に転送します。奇数番地の場合、Rsの上位8bitの内容を[R8<<16+Rd]に、Rsの下位8bitの内容を[R8<<16+Rd−1]に転送します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R7となっています。

#### 【例】

```

MOV.W    0x50,#0x6666
MOV.W    R3,#0x0052
MOV.W    R0,#0x5555
MOV.F.W  (--R3),R0
INC      R3,#1
MOV.W    R1,#0x1200
MOV.F.W  (--R3),R1
INC      R3,#1
MOV.W    R2,#0x0000
MOV.F.W  (--R3),R2
INC      R3,#1
MOV.W    R0,#0x5634
MOV.F.W  (--R3),R0

```

RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ N0	Z8	Z16	P	S
-	-	-	-	-	-	-	-	-	-	-
66h	66h	-	-	-	-	-	0	0	0	0
66h	66h	-	-	-	0052h	3	0	0	1	0
66h	66h	5555h	-	-	0052h	0	0	0	0	0
55h	55h	5555h	-	-	0050h	0	0	0	0	0
55h	55h	5555h	-	-	0052h	3	0	0	1	0
55h	55h	5555h	1200h	-	0052h	1	1	0	0	0
00h	12h	5555h	1200h	-	0050h	1	1	0	0	0
00h	12h	5555h	1200h	-	0052h	3	0	0	1	0
00h	12h	5555h	1200h	0000h	0052h	2	1	1	0	0
00h	00h	5555h	1200h	0000h	0050h	2	1	1	0	0
00h	00h	5555h	1200h	0000h	0052h	3	0	0	1	0
00h	00h	5634h	1200h	0000h	0052h	0	0	0	1	0
34h	56h	5634h	1200h	0000h	0050h	0	0	0	1	0

#### <注意事項>

この場合、Rb0は、R8を示します。

Rsの内容をプログラムメモリ(ROM)に転送すると、3サイクル命令となります。但し、実際には“ROM”に転送することは出来ません。

## MOVWF[. W] (Rd++), Rs

命令コード	[0 1 1 0 0 1 1 1][d3d2d1d0 0 s2s1s0]	6700H
引数	Rd = 4bit(R select),Rs = 3bit(R select)	
ワード数	1	
サイクル数	2 or 3	
機能	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	
影響を受けるフラグ	Z8,Z16,P,S,N0～N3	

【説明】

R8(Rb0)の内容が上位16bit, Rdで指定する汎用レジスタの内容が下位16bitの32bitでアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)が偶数番地の場合、Rsの下位8bitの内容を[R8<<16+Rd]に、Rsの上位8bitの内容を[R8<<16+Rd+1]に転送します。奇数番地の場合、Rsの上位8bitの内容を[R8<<16+Rd]に、Rsの下位8bitの内容を[R8<<16+Rd-1]に転送します。その後、Rdの内容を+2します。Rdの加算結果にキャリーが発生する場合、R8がインクリメントされます。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R7となっています。

【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3~ 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

＜注意事項＞

この場合、Rb0は、R8を示します。

Rsの内容をプログラムメモリ(ROM)に転送すると、3サイクル命令となります。但し、実際には“ROM”に転送することは出来ません。

## 命令

### MOV.F[.W] Rd, (Rb, Rs, ±n)

命令コード	[0 1 1 1 0 1 0 1][s3s2s1s0 1 d2d1d0][0 b2b1b0 n11 ~ n8][n7 ~ n0] 7508H
引数	Rd = 3bit(R select), Rb = 3bit(Rb select), Rs = 4bit(R select), n = 12bit(signed)
ワード数	2
サイクル数	3 or 4
機能	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
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

ベースレジスタ(Rb)で指定する汎用レジスタの内容が上位16bit, Rsの内容とnの演算結果\*<sup>1</sup>が下位16bitの32bitでアドレス指定されるデータメモリ(RAM), 特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)が偶数番地の場合、その内容をRdの下位8bitに、[Rb<<16+Rs±n+1]の内容をRdの上位8bitに転送します。奇数番地の場合、[Rb<<16+Rs±n]の内容をRdの上位8bitに、[Rb<<16+Rs±n-1]の内容をRdの下位8bitに転送します。

なお、Rdで指定できる範囲はR0～R7、Rbで指定できる範囲はR8～R13、Rsで指定できる範囲はR0～R15、nで指定できる範囲は符号付12bitデータ(−2048～2047)となっています。

\*1: 下位16ビット演算結果のキャリー・ボローは、上位16ビットに反映されます。

#### 【例】

	RAM (50h)	RAM (51h)	R0	R1	R2	R3	R8	N3 ～ 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.F.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
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.W R2,(R8,R3,0x50)	55h	55h	5634h	8118h	5555h	0000h	0000h	2	0	0	0	0

#### <注意事項>

プログラムメモリ(ROM)の内容をRdに転送すると、4サイクル命令となります。

MOV.F[.W] R<sub>d</sub>, (R<sub>b</sub>, --R<sub>s</sub>, ±n)

命令コード	[0 1 1 0 1 1 0 1][s3s2s1s0 1 d2d1d0][0 b2b1b0 n11 ~ n8][n7 ~ n0] 6D08H
引数	Rd = 3bit(R select), Rb = 3bit(Rb select), Rs = 4bit(R select), n = 12bit(signed)
ワード数	2
サイクル数	3 or 4
機能	(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
影響を受けるフラグ	Z8,Z16,P,S,N0~N3

## 【説明】

Rsで指定する汎用レジスタの内容を減算(−2)します。Rsの減算結果にボローが発生する場合は、Rbがデクリメントされます。

その後、ベースレジスタ(Rb)の内容が上位16bit、Rsの内容とnの演算結果\*1が下位16bitの32bitでアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)が偶数番地の場合、その内容をRdの下位8bitに、[Rb<<16+Rs±n+1]の内容をRdの上位8bitに転送します。奇数番地の場合、[Rb<<16+Rs±n]の内容をRdの上位8bitに、[Rb<<16+Rs±n-1]の内容をRdの下位8bitに転送します。

なお、Rdで指定できる範囲はR0~R7、Rbで指定できる範囲はR8~R13、Rsで指定できる範囲はR0~R15、nで指定できる範囲は符号付12bitデータ(−2048~2047)となっています。

\*1: 下位16ビット演算結果のキャリー・ボローは、上位16ビットに反映されます。

## 【例】

	RAM (50h)	RAM (51h)	R0	R1	R2	R3	R8	N3 ~ 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.F.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
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.W R1,(R8,--R3,0x50)	18h	81h	5634h	8118h	0000h	0000h	0000h	1	0	0	0	1

## &lt;注意事項&gt;

プログラムメモリ(ROM)の内容をRdに転送すると、4サイクル命令となります。

## 命令

### MOV.F[.W] Rd, (Rb, Rs++, ±n)

命令コード	[0 1 1 0 0 1 0 1][s3s2s1s0 1 d2d1d0][0 b2b1b0 n11 ~ n8][n7 ~ n0] 6508H
引数	Rd = 3bit(R select), Rb = 3bit(Rb select), Rs = 4bit(R select), n = 12bit(signed)
ワード数	2
サイクル数	3 or 4
機能	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
影響を受けるフラグ	Z8, Z16, P, S, N0~N3

#### 【説明】

ベースレジスタ(Rb)で指定する汎用レジスタの内容が上位16bit, Rsの内容とnの演算結果\*<sup>1</sup>が下位16bitの32bitでアドレス指定されるデータメモリ(RAM), 特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)が偶数番地の場合、その内容をRdの下位8bitに、[Rb<<16+Rs±n+1]の内容をRdの上位8bitに転送します。奇数番地の場合、[Rb<<16+Rs±n]の内容をRdの上位8bitに、[Rb<<16+Rs±n-1]の内容をRdの下位8bitに転送します。その後、Rsの内容を+2します。Rsの加算結果にキャリーが発生する場合、Rbがインクリメントされます。

なお、Rdで指定できる範囲はR0~R7、Rbで指定できる範囲はR8~R13、Rsで指定できる範囲はR0~R15、nで指定できる範囲は符号付12bitデータ(-2048~2047)となっています。

\* 1: 下位16ビット演算結果のキャリー・ボローは、上位16ビットに反映されます。

#### 【例】

	RAM (50h)	RAM (51h)	R0	R1	R2	R3	R8	N3 ~ 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.F.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
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.W R1,(R8,R3++,0x50)	18h	81h	5634h	8118h	0000h	0002h	0000h	1	0	0	0	1

#### <注意事項>

プログラムメモリ(ROM)の内容をRdに転送すると、4サイクル命令となります。

## MOV.F[. W] Rd, (Rs)

命令コード	[0 1 1 1 0 1 0 1][s3s2s1s0 0 d2d1d0]7500H
引数	Rd = 3bit(R select),Rs = 4bit(R select)
ワード数	1
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

## 【説明】

R8(Rb0)の内容が上位16bit, Rsで指定する汎用レジスタの内容が下位16bitの32bitでアドレス指定されるデータメモリ(RAM), 特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)が偶数番地の場合、その内容をRdの下位8bitに、[R8<<16+Rs+1]の内容をRdの上位8bitに転送します。奇数番地の場合、[R8<<16+Rs]の内容をRdの上位8bitに、[R8<<16+Rs-1]の内容をRdの下位8bitに転送します。

なお、Rdで指定できる範囲はR0～R7、Rsで指定できる範囲はR0～R15となっています。

## 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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.F.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
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.W	R2,(R3)	55h	55h	5634h	8118h	5555h	0050h	2	0	0	0	0

## &lt;注意事項&gt;

この場合、Rb0は、R8を示します。

プログラムメモリ(ROM)の内容をRdに転送すると、3サイクル命令となります。

## 命 令

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

命令コード	[0 1 1 0 1 1 0 1][s3s2s1s0 0 d2d1d0] 6D00H
引数	Rd = 3bit(R select),Rs = 4bit(R select)
ワード数	1
サイクル数	2 or 3
機能	(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
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

Rsで指定する汎用レジスタの内容を減算(−2)します。Rsの減算結果にボローが発生する場合は、R8がデクリメントされます。

その後、R8(Rb0)の内容が上位16bit、Rsの内容が下位16bitの32bitでアドレス指定されるデータメモリ(RAM)、特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)が偶数番地の場合、その内容をRdの下位8bitに、[R8<<16+Rs+1]の内容をRdの上位8bitに転送します。奇数番地の場合、[R8<<16+Rs]の内容をRdの上位8bitに、[R8<<16+Rs−1]の内容をRdの下位8bitに転送します。

なお、Rdで指定できる範囲はR0～R7、Rsで指定できる範囲はR0～R15となっています。

#### 【例】

	RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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.F.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
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.W R1,(--R3)	81h	18h	5634h	8118h	0000h	0050h	1	0	0	0	1

#### <注意事項>

この場合、Rb0は、R8を示します。

プログラムメモリ(ROM)の内容をRdに転送すると、3サイクル命令となります。



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

命令コード	[0 1 1 0 0 1 0 1][s3s2s1s0 0 d2d1d0] 6500H
引数	Rd = 3bit(R select), Rs = 4bit(R select)
ワード数	1
サイクル数	2 or 3
機能	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
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

## 【説明】

R8(Rb0)の内容が上位16bit, Rsで指定する汎用レジスタの内容が下位16bitの32bitでアドレス指定されるデータメモリ(RAM), 特殊機能レジスタ(SFR)、またはプログラムメモリ(ROM)が偶数番地の場合、その内容をRdの下位8bitに、[R8<<16+Rs+1]の内容をRdの上位8bitに転送します。奇数番地の場合、[R8<<16+Rs]の内容をRdの上位8bitに、[R8<<16+Rs-1]の内容をRdの下位8bitに転送します。その後、Rsの内容を+2します。Rsの加算結果にキャリーが発生する場合、R8がインクリメントされます。なお、Rdで指定できる範囲はR0～R7、Rsで指定できる範囲はR0～R15となっています。

## 【例】

		RAM (50h)	RAM (51h)	R0	R1	R2	R3	N3～ 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.F.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
MOV.F.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
MOV.F.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
MOV.F.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
MOV.F.W	R1,(R3++)	81h	18h	5634h	8118h	0000h	0052h	1	0	0	0	1

## &lt;注意事項&gt;

この場合、Rb0は、R8を示します。

プログラムメモリ(ROM)の内容をRdに転送すると、3サイクル命令となります。

## 命令

### MUL

命令コード	[0 0 0 0 0 0 0 0][1 1 0 1 0 0 0 0]	00D0H
引数		
ワード数	1	
サイクル数	4 or 18 cycles	
機能	(R0)×(R2)=Result(32bit) , R1= Result >>16 , R0= Result &FFFFh, (PC)←(PC) +2	
影響を受けるフラグ	Z8,Z16,P,S CY,HC,OV,N3～N0 は全てクリアされる。	

#### 【説明】

汎用レジスタR0とR2の乗算結果の上位16bitをR1に転送し、下位16bitをR0に転送します。

#### 【例】

```
MOV.W R0,#0x48D0
MOV.W R1,#0x5678
MOV.W R2,#0x4000
MOV.W R3,#0xDEF0
MUL
```

R0	R1	R2	R3	PSW
-	-	-	-	-
48D0h	-	-	-	0020h
48D0h	5678h	-	-	1000h
48D0h	5678h	4000h	-	2021h
48D0h	5678h	4000h	DEF0h	3040h
0000h	1234h	4000h	DEF0h	0003h

#### <注意事項>

フラグ(Z8, Z16, P, S)は、R0(下位16bitの結果)に影響されます。  
サイクル数は機種により異なりますので、データシートを参照してください。

## NOP

命令コード	[0 0 0 0 0 0 0 0][0 0 0 0 0 0 0 0]	0000H
引数		
ワード数	1	
サイクル数	1	
機能	(PC)←(PC)+2	
影響を受けるフラグ		

## 【説明】

システムクロックを、1 サイクル消費し何もしません。

## 命令

### NOT Rd

命令コード	[0 0 1 1 0 0 0 0][1 0 1 1 d3d2d1d0]30B0H
引数	Rd = 4bit(R select)
ワード数	1
サイクル数	1
機能	(Rd)← ~(Rd) , (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容を反転します。

なお、Rdで指定できる範囲はR0～R15となっています。

#### 【例】

	R0	R1	R2	R3	N3～ 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

命令コード	[0 1 0 0 0 1 0][s3s2s1s 0 d3d2d1d0]	4200H
引数	Rd = 4bit(R select),Rs = 4bit(R select)	
ワード数	1	
サイクル数	1	
機能	(Rd)←(Rd)   (Rs) , (PC)←(PC)+2	
影響を受けるフラグ	Z8,Z16,P,S,N0～N3	

## 【説明】

Rdで指定する汎用レジスタの内容と、Rsで指定する汎用レジスタの内容の論理和をとり、その結果をRdに転送します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15となっています。

## 【例】

	R0	R1	R2	R3	N3～ 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

## 命令

### OR Rd, #imm16

命令コード	[0 0 1 1 0 0 0 1][0 0 0 1 d3d2d1d0][i15 ~ i8][i7 ~ i0]	3110H
引数	Rd = 4bit(R select), imm16 = 16bit(immediate data)	
ワード数	2	
サイクル数	2	
機能	(Rd) ← (Rd)   #imm16 , (PC) ← (PC) + 4	
影響を受けるフラグ	Z8, Z16, P, S, N0 ~ N3	

#### 【説明】

Rdで指定する汎用レジスタの内容と、imm16で指定するイミディエイトデータの論理和をとり、その結果をRdに転送します。

なお、Rdで指定できる範囲はR0～R15、imm8で指定できる範囲は0～FFとなっています。

#### 【例】

	R0	R1	R2	R3	N3～ 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

命令コード	[0 1 0 0 0 1 1][i7i6i5i4i3i2i1i0]	4300H
引数	imm8 = 8bit(immediate data)	
ワード数	1	
サイクル数	1	
機能	(Rx)← (Rx)   16bit data(Hibyte=00H, Lobyte=#imm8) , (PC)←(PC)+2	
影響を受けるフラグ	Z8,Z16,P,S	

## 【説明】

PSW の bit12～15 (N0～N3)の値によって間接指定される汎用レジスタ(Rx)の内容と、上位8bitを“00h”  
下位8bitをimm8で指定するイミディエイトデータとした16bitデータの論理和をとり、その結果をRxに転送  
します。

なお、imm8で指定できる範囲は0～FFとなっています。

## 【例】

	R0	R1	R2	R3	N3～ 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

## 命 令

### POP PSW

命令コード	[0 0 0 0 0 0 0 0][1 0 0 1 1 1 0]	009EH
引数		
ワード数	1	
サイクル数	1	
機能	(SP)←(SP)-2 Hibyte(PSW)←[SP+1], Lobyte(PSW)←[SP], (PC)←(PC)+2	
影響を受けるフラグ		

#### 【説明】

スタックポインタ(SP)をデクリメントし、さらにSPで指定されるデータメモリ(RAM)の内容をプログラムステータスワード(PSW)に転送します。

#### 【例】

		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

命令コード	[0 0 0 0 0 0 0 0][1 0 0 1 s3s2s1s0]0090H
引数	Rs = 4bit(R select)
ワード数	1
サイクル数	1
機能	(SP)←(SP)-2, Hibyte(Rs)←[SP+1], Lobyte(Rs)← [SP], (PC)←(PC)+2
影響を受けるフラグ	

## 【説明】

スタックポインタ(SP)をデクリメントし、さらにSPで指定されるデータメモリ(RAM)の内容を、Rsで指定する汎用レジスタに転送します。

なお、Rsで指定できる範囲はR0～R15となっています。

## 【例】

	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

## 命令

### PUSH PSW

命令コード	[0 0 0 0 0 0 0 0][1 0 0 0 1 1 1 0]008EH
引数	
ワード数	1
サイクル数	1
機能	[SP+1]←Hibyte(PSW) , [SP]←Lobyte(PSW), (SP)←(SP)+2,(PC)←(PC)+2
影響を受けるフラグ	

#### 【説明】

プログラムステータスワード(PSW)の内容をスタックポインタ(SP)が示すデータメモリ(RAM)に転送し、さらにSPをインクリメントします。

#### 【例】

		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

命令コード	[0 0 0 0 0 0 0 0][1 0 0 0 s3s2s1s0]0080H
引数	Rs = 4bit(R select)
ワード数	1
サイクル数	1
機能	[SP+1]←Hibyte(Rs), [SP]←Lobyte(Rs), (SP)←(SP)+2,(PC)←(PC)+2
影響を受けるフラグ	

## 【説明】

Rsで指定する汎用レジスタの内容をスタックポインタ(SP)が示すデータメモリ(RAM)に転送し、さらにSPをインクリメントします。

なお、Rsで指定できる範囲はR0～R15となっています。

## 【例】

	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

## 命 令

### RESET

命令コード	[0 0 0 0 0 0 0 0][0 0 0 0 1 1 1 1]	000FH
引数		
ワード数	1	
サイクル数	1	
機能	Initialize	
影響を受けるフラグ		

#### 【説明】

RESET命令実行後、初期化されます。

# RET

命令コード	[0 0 0 0 0 0 0 0][0 0 0 0 0 0 1 1]0003H
引数	
ワード数	1
サイクル数	3
機能	(PC)←(SP-1<<24+SP-2<<16+SP-3<<8+SP-4), (SP)←(SP)-4
影響を受けるフラグ	

## 【説明】

スタックポインタ (SP) をデクリメントし、さらに SP で指定されるデータメモリ (RAM) の内容をプログラムカウンタ (PC) に転送します。

【例】ラベル LA の値は、910AH です。

	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

## 命 令

### REV Rd

命令コード	[0 0 1 1 0 0 0 0][1 1 1 1 d3d2d1d0] 30F0H
引数	Rd = 4bit(R select)
ワード数	1
サイクル数	1
機能	(Rd)←mirror(Rd) , (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容を逆転(=MSB側とLSB側を入れ替える)します。

なお、Rdで指定できる範囲はR0～R15となっています。

#### 【例】

	R0	R1	R2	R3	N3～ 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

命令コード	[0 0 1 1 1 0 1 1][i3i2i1i0d3d2d1d0] 3B00H
引数	Rd = 4bit(R select),imm4 = 4bit(immediate data)
ワード数	1
サイクル数	1
機能	(Rd)←(Rd) rotate left #imm4 bit through carry (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,CY,P,S,N0～N3

## 【説明】

imm4で指定するイミディエイトデータが示す値(ローテート量)で、Rdの内容をキャリーフラグ(CY)を含め(17bitの空間)左にローテートします。

なお、Rdで指定できる範囲はR0～R15、imm4で指定できる範囲は0～Fとなっています。

## 【例】

		R0	R1	R2	R3	N3～ 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

## 命令

### RLC Rd, Rs

命令コード	[0 0 1 1 1 0 1 0][s3s2s1s0d3d2d1d0]3A00H
引数	Rd = 4bit(R select),Rs = 4bit(R select)
ワード数	1
サイクル数	1
機能	(Rd)←(Rd) rotate left (Rs)&000Fh bit through carry (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,CY,P,S,N0～N3

#### 【説明】

Rsで指定する汎用レジスタの内容の下位4bitが示す値(ローテート量)で、Rdの内容とキャリーフラグ(CY)を含め(17bitの空間)左にローテートします。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15となっています。

#### 【例】

	R0	R1	R2	R3	N3～ 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

命令コード	[0 0 1 1 1 0 0 1][i3i2i1i0d3d2d1d0]	3900H
引数	Rd = 4bit(R select),imm4 = 4bit(immediate data)	
ワード数	1	
サイクル数	1	
機能	(Rd)←(Rd) rotate right #imm4 bit through carry (PC)←(PC)+2	
影響を受けるフラグ	Z8,Z16,CY,P,S,N0～N3	

## 【説明】

imm4で指定するイミディエイトデータが示す値(ローテート量)で、Rdの内容とキャリーフラグ(CY)を含め(17bitの空間)右にローテートします。

なお、Rdで指定できる範囲はR0～R15、imm4で指定できる範囲は0～Fとなっています。

## 【例】

	R0	R1	R2	R3	N3～ 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

## 命令

### RRC Rd, Rs

命令コード	[0 0 1 1 1 0 0 0][s3s2s1s0d3d2d1d0]3800H
引数	Rd = 4bit(R select),Rs = 4bit(R select)
ワード数	1
サイクル数	1
機能	(Rd)←(Rd) rotate right (Rs)&000Fh bit through carry (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,CY,P,S,N0～N3

#### 【説明】

Rsで指定する汎用レジスタの内容の下位4bitが示す値(ローテート量)で、Rdの内容とキャリーフラグ(CY)を含め(17bitの空間)右にローテートします。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15となっています。

#### 【例】

	R0	R1	R2	R3	N3～ 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

命令コード	[0 1 0 1 0 1 1 1][i3i2i1i0d3d2d1d0] 5700H
引数	Rd = 4bit(R select), imm4 = 4bit(immediate data)
ワード数	1
サイクル数	1
機能	(Rd) ← (Rd) - #imm4 - CY, (PC) ← (PC) + 2
影響を受けるフラグ	Z8, Z16, CY, HC, OV, P, S, N0 ~ N3

## 【説明】

Rdで指定する汎用レジスタの内容からimm4で指定するイミディエイトデータとキャリーフラグ(CY)を減算し、その結果をRdに転送します。

なお、Rdで指定できる範囲はR0～R15、imm4で指定できる範囲は0～Fとなっています。

## 【例】

		R0	R1	R2	R3	N3～ 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

## 命令

### SBC Rd, #imm16

命令コード	[0 0 1 1 0 0 0 1][0 1 1 1 d3d2d1d0][i15 ~ i8][i7 ~ i0]	3170H
引数	Rd = 4bit(R select), imm16 = 16bit(immediate data)	
ワード数	2	
サイクル数	2	
機能	(Rd) ← (Rd) - #imm16 - CY, (PC) ← (PC) + 4	
影響を受けるフラグ	Z8, Z16, CY, HC, OV, P, S, N0 ~ N3	

#### 【説明】

Rdで指定する汎用レジスタの内容からimm16で指定するイミディエイトデータとキャリーフラグ(CY)を減算し、その結果をRdに転送します。

なお、Rdで指定できる範囲はR0～R15、imm16で指定できる範囲は0～FFFFとなっています。

#### 【例】

		R0	R1	R2	R3	N3～ 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

命令コード	[0 1 0 1 1 1 1][i7i6i5i4i3i2i1i0]	5F00H
引数	imm8 = 8bit(immediate data)	
ワード数	1	
サイクル数	1	
機能	(Rx)←(Rx) - #imm8 - CY,(PC)←(PC)+2	
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S	

## 【説明】

PSW の bit12～15 (N0～N3)の値によって間接指定される汎用レジスタ(Rx)の内容からimm8で指定するイミディエイトデータとキャリーフラグ(CY)を減算し、その結果を Rxに転送します。

なお、imm8で指定できる範囲は0～FFとなっています。

## 【例】

	R0	R1	R2	R3	N3～ 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

## 命 令

### SBC Rd, Rs

命令コード	[0 1 0 0 1 1 1 1][s3s2s1s0d3d2d1d0]4F00H
引数	Rd = 4bit(R select),Rs = 4bit(R select)
ワード数	1
サイクル数	1
機能	(Rd)←(Rd) - (Rs) - CY , (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容からRsで指定する汎用レジスタの内容とキャリーフラグ(CY)を減算し、その結果をRdに転送します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15となっています。

#### 【例】

		R0	R1	R2	R3	N3～ 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

命令コード	[0 0 0 0 0 0 0 0][1 1 0 0 1 0 0 0]	00C8H
引数		
ワード数	1	
サイクル数	18～19 cycles	
機能	(R0 : quotient)...(R1 : remainder) $\leftarrow$ (R0) $\div$ (R2)(signed division), (PC) $\leftarrow$ (PC)+2	
影響を受けるフラグ	Z8,Z16,P,S,CY(equal to S) HC,OV,N3～N0 は全てクリアされる。	

## 【説明】

汎用レジスタR0(符号付16bitデータ)とR2(符号付16bitデータ)の除算結果をR0に、剰余をR1に転送します。

R2が“0”の場合、有効な結果は保証されません。

## 【例】

```

MOV.W  R0,#0x89AB
MOV.W  R1,#0x5678
MOV.W  R2,#0x1234
MOV.W  R3,#0xDEF0
SDIV
MOV.W  R0,#0x8000
MOV.W  R2,#0x0002
SDIV
MOV.W  R0,#0xFFFF
SDIV

```

R0	R1	R2	R3	PSW
-	-	-	-	-
89ABh	-	-	-	0040h
89ABh	5678h	-	-	1000h
89ABh	5678h	1234h	-	2020h
89ABh	5678h	1234h	DEF0h	3040h
FFFAh	E6E3h	1234h	DEF0h	0044h
8000h	E6E3h	1234h	DEF0h	0065h
8000h	E6E3h	0002h	DEF0h	2024h
C000h	0000h	0002h	DEF0h	0045h
FFFFh	0000h	0002h	DEF0h	0044h
0000h	FFFFh	0002h	DEF0h	0003h

## &lt;注意事項&gt;

この命令サイクル数は、可変型です。

剰余の符号は、被除数の符号と同じになります。

フラグ(Z8, Z16, P, S)は、R0(商)に影響されます。

## 命令

### SDIVLH

命令コード	[0 0 0 0 0 0 0 0][1 1 1 0 1 0 0 0]	00E8H
引数		
ワード数	1	
サイクル数	18～19 cycles	
機能	(R0 : quotient)...(R1 : remainder) $\leftarrow$ (R1 $\ll$ 16+R0) $\div$ (R2)(signed division), (PC) $\leftarrow$ (PC)+2	
影響を受けるフラグ	Z8,Z16,P,S,CY(equal to S)      HC,OV,N3～N0 は全てクリアされる。	

#### 【説明】

符号付32bitデータ(R1 $\ll$ 16+R0)とR2(符号付16bitデータ)の除算結果をR0に転送し、剰余をR1に転送します。

R2が“0”または、商(R0)が8000h(−32768)～7FFFh(32767)の範囲を越える場合、有効な結果は保証されません。

#### 【例】

```

MOV.W  R0,#0x0A9F
MOV.W  R1,#0x3AB0
MOV.W  R2,#0x8001
MOV.W  R3,#0xDEFO
SDIVLH
MOV.W  R0,#0x0AA0
MOV.W  R1,#0x3AB0
MOV.W  R2,#0x8001
SDIVLH
MOV.W  R0,#0xF560
MOV.W  R1,#0xC54F
MOV.W  R2,#0x7FFF
SDIVLH

```

R0	R1	R2	R3	PSW
-	-	-	-	-
0A9Fh	-	-	-	0000h
0A9Fh	3AB0h	-	-	1020h
0A9Fh	3AB0h	8001h	-	2040h
0A9Fh	3AB0h	8001h	DEF0h	3040h
8A9Fh	0000h	8001h	DEF0h	0064h
0AA0h	0777h	8001h	DEF0h	0004h
0AA0h	3AB0h	8001h	DEF0h	1024h
0AA0h	3AB0h	8001h	DEF0h	2044h
8A9Fh	0001h	8001h	DEF0h	0064h
F560h	0001h	8001h	DEF0h	0044h
F560h	C54Fh	8001h	DEF0h	1064h
F560h	C54Fh	7FFFh	DEF0h	2024h
8A9Fh	FFFFh	7FFFh	DEF0h	0064h

#### <注意事項>

この命令サイクル数は、可変型です。

剰余の符号は、被除数の符号と同じになります。

フラグ(Z8, Z16, P, S)は、R0(商)に影響されます。



SET1 m16, #imm3

命令コード	[1 1 1 X i2i1i0 1][m7m6m5m4m3m2m1m0] E100H(RAM),F100H(SFR)
引数	m16 = 16bit(Lower 8bit valid for operation code),imm3 = 3bit(bit select)
ワード数	1
サイクル数	2
機能	(m16)←(m16) of bit #imm3 ←1,(PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

## 【説明】

m16でアドレス指定する2バイト分の RAM(データメモリ)または SFR(内蔵する周辺機能を制御する専用レジスタ群)のimm3で指定できるイミディエイトデータが示すbitを1にセットします。

なお、imm3で指定できる範囲は0～8となっています。

m16(第1オペランドデータ)の値によって、コンパイラは転送先が RAM であるのか SFR であるのかを判断して命令コードを生成します。

- RAM を指定する場合には、m16の値は 00H～FFH(0000H～00FFH)の範囲で指定してください。この命令で 100H 番地以上の RAM を指定することはできません。
- SFR を指定する場合には、m16の値は 7F00H～7FFFH の範囲で指定してください。

なお、生成される命令コードの基本系は、それぞれ E100H(RAM), F100H(SFR)になり、命令コードの動作に反映されるのは、m16の下位 8 ビットになります。

## 【例】

```
MOV.B 0x50,#0xFF
MOV.B 0x51,#0x32
MOV.B 0x52,#0x00
MOV.B 0x53,#0x54
SET1 0x50,#0x02
SET1 0x51,#0x00
SET1 0x52,#0x04
SET1 0x53,#0x07
```

RAM (50h)	RAM (51h)	RAM (52h)	RAM (53h)	Z8	Z16	P	S
-	-	-	-	-	-	-	-
FFh	-	-	-	0	0	0	1
FFh	32h	-	-	0	0	1	0
FFh	32h	00h	-	1	1	0	0
FFh	32h	00h	54h	0	0	1	0
FFh	32h	00h	54h	0	0	0	1
FFh	33h	00h	54h	0	0	0	0
FFh	33h	10h	54h	0	0	1	0
FFh	33h	10h	D4h	0	0	0	1

## 命 令

### SET1 Rd, #imm4

命令コード	[0 0 0 0 1 0 0 1][i3i2i1i0d3d2d1d0]0900H
引数	Rd = 4bit(R select),imm4 = 4bit(bit select)
ワード数	1
サイクル数	1
機能	(Rd)←(Rd) of bit #imm4←1,(PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容のimm4で指定するイミディエイトデータが示すbitを1にセットします。  
 なお、Rdで指定できる範囲はR0～R15、imm4で指定できる範囲は0～Fとなっています。

#### 【例】

	R0	R1	R2	R3	N3～ 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

命令コード	[0 0 0 0 1 0 1 1][s3s2s1s0d3d2d1d0]0B00H
引数	Rd = 4bit(R select),Rs = 4bit(bit select)
ワード数	1
サイクル数	1
機能	(Rd)←(Rd) of bit (Rs)&000Fh ← 1,(PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

## 【説明】

Rdで指定する汎用レジスタの内容で、Rsで指定する汎用レジスタの内容の下位4bitが示すbitを1にセットします。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15となっています。

## 【例】

	R0	R1	R2	R3	N3～ 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

## 命 令

### SHL Rd, #imm4

命令コード	[0 0 1 1 1 1 1 1][i3i2i1i0d3d2d1d0]3F00H
引数	Rd = 4bit(R select),imm4 = 4bit(immediate data)
ワード数	1
サイクル数	1
機能	(Rd)←(Rd) logical shift left #imm4 bit (CY)←last shift bit , (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,CY,P,S,N0～N3

#### 【説明】

imm4で指定するイミディエイトデータが示す値(シフト量)で、Rdで指定する汎用レジスタの内容を左にシフトします。そして、最後にMSBからの溢れビットをキャリーフラグ(CY)に転送します。

なお、Rdで指定できる範囲はR0～R15、imm4で指定できる範囲は0～Fとなっています。

#### 【例】

	R0	R1	R2	R3	N3～ 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

#### <注意事項>

Rdの内容は、左シフトされてLSB側から“0”が入ります。

SHL Rd, Rs

命令コード	[0 0 1 1 1 1 0][s3s2s1s0d3d2d1d0] 3E00H
引数	Rd = 4bit(R select),Rs = 4bit(R select)
ワード数	1
サイクル数	1
機能	(Rd)←(Rd) logical shift left (Rs)&000Fh bit (CY)←last shift bit,(PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,CY,P,S,N0～N3

## 【説明】

Rsで指定する汎用レジスタの内容の下位4bitが示す値(シフト量)で、Rdで指定する汎用レジスタの内容を左にシフトします。そして、最後にMSBからの溢れビットをキャリーフラグ(CY)に転送します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15となっています。

## 【例】

		R0	R1	R2	R3	N3～ 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

## &lt;注意事項&gt;

Rdの内容は、左シフトされてLSB側から“0”が入ります。

## 命 令

### SHR Rd, #imm4

命令コード	[0 0 1 1 1 0 1][i3i2i1i0d3d2d1d0]3D00H
引数	Rd = 4bit(R select),imm4 = 4bit(immediate data)
ワード数	1
サイクル数	1
機能	(Rd)←(Rd) logical shift right #imm4 bit (CY)←last shift bit,(PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,CY,P,S,N0～N3

#### 【説明】

imm4で指定するイミディエイトデータが示す値(シフト量)で、Rdで指定する汎用レジスタの内容を右にシフトします。そして、最後にLSBからの溢れビットをキャリーフラグ(CY)に転送します。

なお、Rdで指定できる範囲はR0～R15、imm4で指定できる範囲は0～Fとなっています。

#### 【例】

	R0	R1	R2	R3	N3～ 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

#### <注意事項>

Rdの内容は、右シフトされてMSB側から“0”が入ります。

SHR Rd, Rs

命令コード	[0 0 1 1 1 0 0][s3s2s1s0d3d2d1d0] 3C00H
引数	Rd = 4bit(R select),Rs = 4bit(R select)
ワード数	1
サイクル数	1
機能	(Rd)←(Rd) logical shift right (Rs)&000Fh bit (CY)←last shift bit,(PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,CY,P,S,N0～N3

## 【説明】

Rsで指定する汎用レジスタの内容の下位4bitが示す値(シフト量)で、Rdで指定する汎用レジスタの内容を右にシフトします。そして、最後にLSBからの溢れビットをキャリーフラグ(CY)に転送します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15となっています。

## 【例】

	R0	R1	R2	R3	N3～ 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

## &lt;注意事項&gt;

Rdの内容は、右シフトされてMSB側から“0”が入ります。

## 命令

### SUB Rd, #imm4

命令コード	[0 1 0 1 0 1 0 1][i3i2i1i0d3d2d1d0]5500H
引数	Rd = 4bit(R select),imm4 = 4bit(immediate data)
ワード数	1
サイクル数	1
機能	(Rd)←(Rd) - #imm4,(PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容から、imm4で指定するイミディエイトデータを減算し、その結果をRdに転送します。

なお、Rdで指定できる範囲はR0～R15、imm4で指定できる範囲は0～Fとなっています。

#### 【例】

	R0	R1	R2	R3	N3～ 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

命令コード	[0 0 1 1 0 0 0 1][0 1 1 0 d3d2d1d0][i15 ~ i8][i7 ~ i0]	3160H
引数	Rd = 4bit(R select), imm16 = 16bit(immediate data)	
ワード数	2	
サイクル数	2	
機能	(Rd) ← (Rd) - #imm16, (PC) ← (PC) + 4	
影響を受けるフラグ	Z8, Z16, CY, HC, OV, P, S, N0 ~ N3	

## 【説明】

Rdで指定する汎用レジスタの内容からimm16で指定するイミディエイトデータを減算し、その結果をRdに転送します。

なお、Rdで指定できる範囲はR0～R15、imm16で指定できる範囲は0～FFFFとなっています。

## 【例】

		R0	R1	R2	R3	N3～ 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

## 命 令

### SUB Rx, #imm8

命令コード	[0 1 0 1 1 0 1][i7i6i5i4i3i2i1i0]5D00H
引数	imm8 = 8bit(immediate data)
ワード数	1
サイクル数	1
機能	(Rx)←(Rx) - #imm8,(PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S

#### 【説明】

PSW の bit12～15 (N0～N3)の値によって間接指定される汎用レジスタ(Rx)の内容からimm8で指定するイミディエイトデータを減算し、その結果をRxに転送します。

なお、imm8で指定できる範囲は0～FFとなっています。

#### 【例】

	R0	R1	R2	R3	N3～ 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

命令コード	[0 1 0 0 1 1 0 1][s3s2s1s0d3d2d1d0]4D00H
引数	Rd = 4bit(R select),Rs = 4bit(R select)
ワード数	1
サイクル数	1
機能	(Rd)← (Rd) - (Rs), (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,CY,HC,OV,P,S,N0～N3

## 【説明】

Rdで指定する汎用レジスタの内容からRsで指定する汎用レジスタの内容を減算し、その結果をRdに転送します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15となっています。

## 【例】

		R0	R1	R2	R3	N3～ 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

## 命令

### SWPB Rd

命令コード	[0 0 1 1 0 0 0 0][1 0 0 0 d3d2d1d0]3080H
引数	Rd = 4bit(R select)
ワード数	1
サイクル数	1
機能	Hibyte(Rd) ⇔ Lobyte(Rd) (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容の上位8bitと下位8bitを交換します。

なお、Rdで指定できる範囲はR0～R15となっています。

#### 【例】

	R0	R1	R2	R3	N3～ 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

命令コード	[0 0 1 1 0 0 0 0][1 0 0 1 d3d2d1d0] 3090H
引数	Rd = 4bit(R select)
ワード数	1
サイクル数	1
機能	Hibyte(Rd)←Hibyte(Rd) , Lobyte(Rd)←(Rd)&000Fh<<4+(Rd)&00F0h>>4 (PC)←(PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

## 【説明】

Rdで指定する汎用レジスタの内容の下位8bitの上位側4bitと下位側4bitを交換します。

なお、Rdで指定できる範囲はR0～R15となっています。

## 【例】

	R0	R1	R2	R3	N3～ 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

## &lt;注意事項&gt;

Rdの内容の上位8bitは変わりません。

## 命令

### SWPW Rd, Rs

命令コード	[0 0 1 1 0 0 1 0][s3s2s1s0d3d2d1d0]3200H
引数	Rd = 4bit(R select),Rs = 4bit(R select)
ワード数	1
サイクル数	2
機能	(Rd) $\leftrightarrow$ (Rs) exchange,(PC) $\leftarrow$ (PC)+2
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容と、Rsで指定する汎用レジスタの内容を交換します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15となっています。

#### 【例】

	R0	R1	R2	R3	N3～ 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

命令コード	[0 1 0 0 0 1 0 0][s3s2s1s0d3d2d1d0]	4400H
引数	Rd = 4bit(R select),Rs = 4bit(R select)	
ワード数	1	
サイクル数	1	
機能	(Rd)←(Rd) ^ (Rs),(PC)←(PC)+2	
影響を受けるフラグ	Z8,Z16,P,S,N0～N3	

## 【説明】

Rdで指定する汎用レジスタの内容と、Rsで指定する汎用レジスタの内容の排他的論理和をとり、その結果をRdに転送します。

なお、Rdで指定できる範囲はR0～R15、Rsで指定できる範囲はR0～R15となっています。

## 【例】

	R0	R1	R2	R3	N3～ 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

## 命 令

### XOR Rd, #imm16

命令コード	[0 0 1 1 0 0 0 1][0 0 1 0 d3d2d1d0][i15 ~ i8][i7 ~ i0]3120H
引数	Rd = 4bit(R select),imm16 = 16bit(immediate data)
ワード数	2
サイクル数	2
機能	(Rd)←(Rd) ^ #imm16,(PC)←(PC)+4
影響を受けるフラグ	Z8,Z16,P,S,N0～N3

#### 【説明】

Rdで指定する汎用レジスタの内容と、imm16で指定するイミディエイトデータの排他的論理和をとり、その結果をRdに転送します。

なお、Rdで指定できる範囲はR0～R15、imm16で指定できる範囲は0～FFFFとなっています。

#### 【例】

	R0	R1	R2	R3	N3～ 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

命令コード	[0 1 0 0 0 1 0 1][i7i6i5i4i3i2i1i0]	4500H
引数	imm8 = 8bit(immediate data)	
ワード数	1	
サイクル数	1	
機能	(Rx) ← (Rx) ^ 16bit data(Hibyte=00H, Lobyte=#imm8), (PC) ← (PC)+2	
影響を受けるフラグ	Z8,Z16,P,S	

## 【説明】

PSW の bit12～15 (N0～N3) の値によって間接指定される汎用レジスタ(Rx)の内容と、上位8bitを“00h”  
下位8bitをイミディエイトデータ(imm8)とした16bitデータの論理和をとり、その結果をRxに転送します。  
なお、imm8で指定できる範囲は0～FFになっています。

## 【例】

	R0	R1	R2	R3	N3～ 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

## ご注意

本資料に掲載されている記事は、読者が正しく、且つ容易にデバイスの使用法を理解できるように作成したものです。記載されている応用例などをそのまま用いて製品を製造するために書かれているものではありません。したがって、この資料にもとづいて試作・製造が行われ、その結果、安全性・特許権・その他の権利侵害などの問題がありましても当社は一切責任を負いません。

---

LC88 シリーズ 第 5 章 命令  
ユーザーズマニュアル

---

Rev. 0                      2015.12.18 版

---

オン・セミコンダクター  
マイクロコントローラビジネスユニット

---