命令	ニーモニック	オペコート	アドレッシングモード（数値はステート数)									フラグ変化	説明	
	命令 オペランド	OP Rd Rx	Drct	Index	Imm	FP Rlt	Reg	Imm4	Indr	B Indr	Othr			
No Operation	NO	00h 0h 0h	--	--	--	--	--	--	--	--	3	×	何もしない	
Load	LD Rd,EA	08h Rd EA	7	7	5	7	4	4	6	6	--	×	Rd ← [EA]	
Load	LD Rd,FLAG	14h Rd 0h	--	--	--	--	--	--	--	--	4	×	Rd ← FLAG	
Store	ST Rd,EA	10h Rd EA	6	6	--	6	--	--	5	5	--	×	[Dsp] ← EA	
Add	ADD Rd,EA	18h Rd EA	7	7	5	7	5	4	6	6	--	○	Rd ← Rd + [EA]	
Subtract	SUB Rd,EA	20h Rd EA	7	7	5	7	5	4	6	6	--	○	Rd ← Rd - [EA]	
Compare	CMP Rd,EA	28h Rd EA	7	7	5	7	5	4	6	6	--	○	Rd - [EA]	
Logical And	AND Rd,EA	30h Rd EA	7	7	5	7	5	4	6	6	--	○	Rd ← Rd and [EA]	
Logical Or	OR Rd,EA	38h Rd EA	7	7	5	7	5	4	6	6	--	○	Rd ← Rd or [EA]	
Logical Xor	XOR Rd,EA	40h Rd EA	7	7	5	7	5	4	6	6	--	○	Rd ← Rd xor [EA]	
Add with Scale	ADDS Rd,EA	48h Rd EA	8	8	6	8	6	5	7	7	--	○	Rd ← Rd + [EA]*2	
Multiply	MUL Rd,EA	50h Rd EA	57	57	55	57	55	54	56	56	--	○	Rd ← Rd × [EA]	
Divide	DIV Rd,EA	58h Rd EA	73	73	71	73	71	70	72	72	--	○	Rd ← Rd / [EA]	
Modulo	MOD Rd,EA	60h Rd EA	73	73	71	73	71	70	72	72	--	○	Rd ← Rd % [EA]	
Multiply Long	MULL Rd,EA	680h Rd EA	57	57	55	57	55	54	56	56	--	○	(Rd+1,Rd) ← Rd × [EA]	注１
Divide Long	DIVL Rd,EA	70h Rd EA	73	73	71	73	71	70	72	72	--	○	Rd ← (Rd+1,Rd) / [EA], Rd+1 ← (Rd+1,Rd) % [EA]	注１
Shift Left Arithmetic	SHLA Rd,EA	80h Rd EA	8+n	8+n	6+n	8+n	6+n	5+n	7+n	7+n	--	○	Rd ← Rd << [EA]	
Shift Left Logical	SHLL Rd,EA	88h Rd EA	8+n	8+n	6+n	8+n	6+n	5+n	7+n	7+n	--	○	Rd ← Rd << [EA]	
Shift Right Arithmetic	SHRA Rd,EA	90h Rd EA	8+n	8+n	6+n	8+n	6+n	5+n	7+n	7+n	--	○	Rd ← Rd >> [EA]	
Shift Right Logical	SHRL Rd,EA	98h Rd EA	8+n	8+n	6+n	8+n	6+n	5+n	7+n	7+n	--	○	Rd ← Rd >>> [EA]	
Jump on Zero	JZ EA	A0h 0h EA	4/5	4/5	--	--	--	--	4/5	--	--	×	If (Z) PC ← EA	
Jump on Carry	JC EA	A0h 1h EA	4/5	4/5	--	--	--	--	4/5	--	--	×	If (C) PC ← EA	
Jump on Minus	JM EA	A0h 2h EA	4/5	4/5	--	--	--	--	4/5	--	--	×	If (S) PC ← EA	
Jump on Overflow	JO EA	A0h 3h EA	4/5	4/5	--	--	--	--	4/5	--	--	×	if (V) PC ← EA	
Jump on greater than	JGT EA	A0h 4h EA	4/5	4/5	--	--	--	--	4/5	--	--	×	If (not (Z or (S xor V))) PC ← EA	
Jump on greater or equal	JGE EA	A0h 5h EA	4/5	4/5	--	--	--	--	4/5	--	--	×	if (not (S xor V)) PC ← EA	
Jump on less or equal	JLE EA	A0h 6h EA	4/5	4/5	--	--	--	--	4/5	--	--	×	If (Z or (S xor V)) PC ← EA	
Jump on less than	JLT EA	A0h 7h EA	4/5	4/5	--	--	--	--	4/5	--	--	×	If (S xor V) PC ← EA	
Jump on Non Zero	JNZ EA	A0h 8h EA	4/5	4/5	--	--	--	--	4/5	--	--	×	If (not Z) PC ← EA	
Jump on Non Carry	JNC EA	A0h 9h EA	4/5	4/5	--	--	--	--	4/5	--	--	×	If (not C) PC ← EA	
Jump on Non Minus	JNM EA	A0h Ah EA	4/5	4/5	--	--	--	--	4/5	--	--	×	If (not S) PC ← EA	
Jump on Non Overflow	JNO EA	A0h Bh EA	4/5	4/5	--	--	--	--	4/5	--	--	×	If (not V) PC ← EA	
Jump on higher	JHI EA	A0h Ch EA	4/5	4/5	--	--	--	--	4/5	--	--	×	If (not (Z or C)) PC ← EA	
Jump on lower or same	JLS EA	A0h Eh EA	4/5	4/5	--	--	--	--	4/5	--	--	×	If (Z or C) PC ← EA	
Jump	JMP EA	A0h Fh EA	5	5	--	--	--	--	5	--	--	×	PC ← EA	
Call subroutine	CALL EA	A8h 0h EA	6	6	--	--	--	--	6	--	--	×	[--SP] ← PC, PC ← EA	
Input	IN Rd,EA	B0h Rd EA	7	--	--	--	--	--	6	6	--	×	Rd ← IO[EA]	
Output	OUT Rd,EA	B8h Rd EA	6	--	--	--	--	--	5	5	--	×	IO[EA] ← Rd	
Push Register	PUSH Rd	C0h Rd 0h	--	--	--	--	--	--	--	--	5	×	[--SP] ← Rd	
Pop Register	POP Rd	C4h Rd 0h	--	--	--	--	--	--	--	--	6	×	Rd ← [SP++]	
Return from Subroutine	RET	D0h 0h 0h	--	--	--	--	--	--	--	--	6	×	PC ← [SP++]	
Return from Interrupt	RETI	D4h 0h 0h	--	--	--	--	--	--	--	--	9	×	FLAG ← [SP++], PC ← [SP++]	
Enable Interrupt	EI	E0h 0h 0h	--	--	--	--	--	--	--	--	5	×	割込み許可	
Disable Interrupt	DI	E4h 0h 0h	--	--	--	--	--	--	--	--	5	×	割込み禁止	
Supervisor Call	SVC	F0h 0h 0h	--	--	--	--	--	--	--	--	12	×	システムコール	
Halt	HALT	FFh 0h 0h	--	--	--	--	--	--	--	--	5	×	CPU停止	

アドレッシングモード（上の表中EAの詳細）に付いて

アドレッシングモード	略記	ニーモニック (EA部分の標記方法)	命令フォーマット		EA(実効アドレス)の決め方		
			第１ワード	第２ワード	略記	解説	
Direct	Drct	OP Rd,Dsp	OP+0 Rd0h	Dsp	[Dsp]	Dsp番地	
Indexed	Index	OP Rd,Dsp,Rx	OP+1 RdRx	Dsp	[Dsp+Rx]	（Dsp＋Rxレジスタの内容）番地	
Immediate	Imm	OP Rd,#Imm	OP+2 Rd0h	Imm	Imm	Immそのもの	
FP Rerative	FP Rlt	OP Rd,Dsp4,FP	OP+3 RdD4	--	[Dsp4+FP]	（D4を符号拡張した値×2 ＋FPレジスタの内容)番地(D4=Dsp4/2)	注２
Register	Reg	OP Rd,Rs	OP+4 RdRs	--	Rs	Rsレジスタの内容	
4bit Signed Immediate	Imm4	OP Rd,#Imm4	OP+5 RdI4	--	Imm4	I4を符号拡張した値そのもの	注３
Register Indirect	Indr	OP Rd,0,Rx	OP+6 RdRx	--	[Rx]	Rxレジスタの内容番地	
Byte Register Indirect	B Indr	OP Rd,@Rx	OP+7 RdRx	--	[Rx]	Rxレジスタの内容番地（但し番地の内容は８bitデータ)	
Other	Othr	OP Rd	OP Rd0h	--		なし	
		OP	OP 0h0h	--		なし	

注４

※アセンブリ言語でDspとDsp4、ImmとImm4の標記は同じ（値によりアセンブラが自動判定）。
※FP相対で、Dsp4は-16～+14の偶数

注１：MULL、DIVL命令ではRdは偶数番号のレジスタ
注２：D4はDsp4(4bitディスプレースメント)の1/2の値
注３：I4はImm４（４bit即値)のこと
注４：アドレッシングモードによりOPの値が変化する