

お客様各位

---

## カタログ等資料中の旧社名の扱いについて

---

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日  
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

## ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。  
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）  
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

保守/廃止

表現形式	記述
r	X(R0), A(R1), C(R2), B(R3), E(R4), D(R5), L(R6), H(R7)
r1	A, B
r2	B, C
r3	D, E, E+
r4	D, E
rp	AX(RP0), BC(RP1), DE(RP2), HL(RP3)
sfr	特殊機能レジスタ略号 (モード・レジスタ活用表参照)
sfrp	特殊機能レジスタ略号 (16ビット・レジスタ; モード・レジスタ活用表参照)
saddr	FE20H-FF1FH immediate data or label
saddrp	FE20H-FF1EH immediate data (bit0=0) or label(16ビット操作時)
addr16	0000H-FFFFH immediate data or label
addr11	800H-FFFFH immediate data or label
addr5	40H-7EH immediate data (bit0=0) or label
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
n	3-bit immediate data
RBn	RB0-RB3

### 備考 1. オペランドのレジスタ名記述方法について

- r, rpは、機能名称 (X, A, C, B, E, D, L, H, AX, BC, DE, HL)のほかに、絶対名称(R0-R7, RP0-RP3)でも記述可能です。
- イミディエイト・アドレッシングは、全空気をアドレス可能です。レラティブ・アドレッシングは、次に続く命令の先頭アドレスから-128~+127の範囲のみアドレス可能です。

### 2. クロック欄について

- 命令の1クロックは、内部システム・クロックの1クロック分に等しく、その周期は1/tCLKになります。
- クロック欄の数値は、内部ROMからプログラムをフェッチした場合の値です。ただし、クロック数は、アクセスするメモリ・エリアによって異なります。詳細はμPD78138 ユーザーズ・マニュアルを参照してください。
- シフト・ローテート命令のクロック欄のnの値は、シフトするビット数を示します。
- 条件付き分岐命令のクロック欄の( )内の値は、分岐しなかった場合のクロック数です。
- コール・リターン命令、スタック命令のクロック欄の( )内の値は、スタック・ポインタの上位8ビットが変化する場合のクロック数です。

### 3. オペレーション欄について

- オペレーション欄のPC<sub>H</sub>はPC<sub>15-8</sub>, PC<sub>L</sub>はPC<sub>7-0</sub>を表します。

### 4. フラグ動作欄の記号説明

記号	説明
(ブランク)	変化なし
0	0にクリアされる
1	1にセットされる
×	結果に従ってセット/クリアされる
R	以前に退避した値がリストアされる

本資料の内容は、後日変更する場合があります。

本文欄外の★印は、本版で改訂された主な箇所を示しています。

# 保守 / 廃止

命令群	ニモニック	オペランド	バイト	クロック	オペレーション	フラグ	
						Z	AC CY
8ビット命令	MOV	r, #byte	2	2	r←byte		
		saddr, #byte	3	3/5	(saddr)←byte		
		sfr, #byte <sup>注1</sup>	3	5	sfr←byte		
		r, r	2	2	r←r		
		A, r	1	2	A←r		
		A, saddr	2	2/4	A←(saddr)		
		saddr, A	2	3/5	(saddr)←A		
		A, sfr	2	4	A←sfr		
		sfr, A	2	5	sfr←A		
		A, [r3, <sup>注2</sup>	1	5	A←(FE00H+r3)r3-00H-FFH <sup>注3</sup>		
		, r3], A <sup>注2</sup>	1	5	(FE00H+r3)←A r3-00H-FFH <sup>注4</sup>		
		A, [HL]	1	5/7	A←(HL)		
		[HL], A	1	5/7	(HL)←A		
		A, [HL+]	1	8/10	A←(HL), HL←HL+1		
		[HL+], A	1	8/10	(HL)←A, HL←HL+1		
		A, [DE]	1	5/7	A←(DE)		
		[DE], A	1	5/7	(DE)←A		
		A, DE+	1	8/10	A←(DE), DE←DE+1		
		[DE+], A	1	8/10	(DE)←A, DE←DE+1		
		A, [addr16]	4	6/8	A←addr16		
[addr16], A	4	6/8	addr16←A				
A, [wordr1]	4	7/9	A←(word+r1)				
[wordr1], A	4	7/9	(word+r1)←A				
PSW, #byte	3	5	PSW←byte	×××			
PSW, A	2	5	PSW←A	×××			
A, PSW	2	4	A←PSW				
16ビット・データ転送命令	XCH	A, r	1	4	A←r		
		A, saddr	2	4/8	A←(saddr)		
		A, sfr	3	10	A←sfr		
		A, [r4]	1	9	A←(r4) r4=00H-FFH		
		A, [HL]	2	9/13	A←(HL)		
		A, [DE]	2	9/13	A←(DE)		
A, [wordr1]	4	9/13	A←(word+r1)				
MOVW	rp, #word	3	3	rp←word			
	saddrp, #word	4	4/8	(saddrp)←word			
	sfrp, #word	4	8	sfrp←word			
	rp, rp	2	4	rp←rp			
	AX, saddrp	2	6/10	AX←(saddrp)			
	saddrp, AX	2	5/9	saddrp←AX			
	AX, sfrp	2	10	AX←sfrp			
	sfrp, AX	2	9	sfrp←AX			

注1. sfrにSTBCを記述した場合別の専用命令となり、バイト数、クロック数が変わります。(CPU制命令参照)  
 2. r3にE+を記述した場合: 6クロック  
 3. r3にE+を記述した場合: A←(E), E←E+1 E=00H-FFH  
 4. r3にE+を記述した場合: (E)←A, E←E+1 E=00H-FFH

命令群	ニモニック	オペランド	バイト	クロック	オペレーション	フラグ		
						Z	AC CY	
8ビット命令	ADD	A, #byte	2	2	A, CY←A+byte	×××		
		saddr, #byte	3	4/8	(saddr), CY←(saddr)+byte	×××		
		sfr, #byte	4	10	sfr, CY←sfr+byte	×××		
		r, r	2	3	r, CY←r+r	×××		
		A, saddr	2	3/5	A, CY←A+(saddr)	×××		
		A, sfr	3	7	A, CY←A+sfr	×××		
		A, [r4]	2	7	A, CY←A+(FE00H+r4) r4=00H-FFH	×××		
		A, [HL]	2	8/10	A, CY←A+(HL)	×××		
		A, [DE]	2	8/10	A, CY←A+(DE)	×××		
		A, [wordr1]	4	8/10	A, CY←A+(word+r1)	×××		
		ADDCC	A, #byte	2	2	A, CY←A+byte-CY	×××	
			saddr, #byte	3	4/8	(saddr), CY←(saddr)+byte+CY	×××	
			sfr, #byte	4	10	sfr, CY←sfr+byte-CY	×××	
			r, r	2	3	r, CY←r+r-CY	×××	
			A, saddr	2	3/5	A, CY←A+(saddr)+CY	×××	
A, sfr	3		7	A, CY←A+sfr+CY	×××			
A, [r4]	2		7	A, CY←A+(FE00H+r4)+CY r4=00H-FFH	×××			
A, [HL]	2		8/10	A, CY←A+(HL)+CY	×××			
A, [DE]	2		8/10	A, CY←A+(DE)+CY	×××			
A, [wordr1]	4		8/10	A, CY←A+(word+r1)+CY	×××			
SUB	A, #byte	2	2	A, CY←A-byte	×××			
	saddr, #byte	3	4/8	(saddr), CY←(saddr)-byte	×××			
	sfr, #byte	4	10	sfr, CY←sfr-byte	×××			
	r, r	2	3	r, CY←r-r	×××			
	A, saddr	2	3/5	A, CY←A-(saddr)	×××			
	A, sfr	3	7	A, CY←A-sfr	×××			
	A, [r4]	2	7	A, CY←A-(FE00H+r4) r4=00H-FFH	×××			
	A, [HL]	2	8/10	A, CY←A-(HL)	×××			
	A, [DE]	2	8/10	A, CY←A-(DE)	×××			
	A, [wordr1]	4	8/10	A, CY←A-(word+r1)	×××			
SUBC	A, #byte	2	2	A, CY←A-byte-CY	×××			
	saddr, #byte	3	4/8	(saddr), CY←(saddr)-byte-CY	×××			
	sfr, #byte	4	10	sfr, CY←sfr-byte-CY	×××			
	r, r	2	3	r, CY←r-r-CY	×××			
	A, saddr	2	3/5	A, CY←A-(saddr)-CY	×××			
	A, sfr	3	7	A, CY←A-sfr-CY	×××			
	A, [r4]	2	7	A, CY←A-(FE00H+r4)-CY r4=00H-FFH	×××			
	A, [HL]	2	8/10	A, CY←A-(HL)-CY	×××			
	A, [DE]	2	8/10	A, CY←A-(DE)-CY	×××			
	A, [wordr1]	4	8/10	A, CY←A-(word+r1)-CY	×××			
AND	A, #byte	2	2	A←A∧byte	×			
	saddr, #byte	3	4/8	(saddr)←(saddr)∧byte	×			
	sfr, #byte	4	10	sfr←sfr∧byte	×			
	r, r	2	3	r←r∧r	×			
	A, saddr	2	3/5	A←A∧(saddr)	×			
	A, sfr	3	7	A←A∧sfr	×			
	A, [r4]	2	7	A←A∧(FE00H+r4) r4=00H-FFH	×			
	A, [HL]	2	8-10	A←A∧(HL)	×			
	A, [DE]	2	8/10	A←A∧(DE)	×			
	A, [wordr1]	4	8/10	A←A∧(word+r1)	×			

# 保守 / 廃止

命令群	モニタ	オペランド	バイト	クロック	オペレーション	フラグ		
						Z	AC CY	
8ビット演算命令	OR	A, #byte	2	2	A ← A ∨ byte		×	
		saddr, #byte	3	4.8	(saddr) ← (saddr) ∨ byte		×	
		sfr, #byte	4	10	sfr ← sfr ∨ byte		×	
		r, r	2	3	r ← r ∨ r		×	
		A, saddr	2	3.5	A ← A ∨ (saddr)		×	
		A, sfr	3	7	A ← A ∨ sfr		×	
		A,  r4	2	7	A ← A ∨ (FE00H + r4) r4 = 00H - FFH		×	
		A,  HL	2	8/10	A ← A ∨ (HL)		×	
		A,  DE	2	8/10	A ← A ∨ (DE)		×	
	A,  word rl	4	8/10	A ← A ∨ (word + rl)		×		
	XOR	A, #byte	2	2	A ← A ⊕ byte		×	
		saddr, #byte	3	4.8	(saddr) ← (saddr) ⊕ byte		×	
		sfr, #byte	4	10	sfr ← sfr ⊕ byte		×	
		r, r	2	3	r ← r ⊕ r		×	
		A, saddr	2	3.5	A ← A ⊕ (saddr)		×	
		A, sfr	3	7	A ← A ⊕ sfr		×	
		A,  r4	2	7	A ← A ⊕ (FE00H + r4) r4 = 00H - FFH		×	
		A,  HL	2	8/10	A ← A ⊕ (HL)		×	
		A,  DE	2	8/10	A ← A ⊕ (DE)		×	
	A,  word rl	4	8/10	A ← A ⊕ (word + rl)		×		
	CMP	A, #byte	2	2	A - byte	×	×	×
		saddr, #byte	3	3/5	(saddr) - byte	×	×	×
		sfr, #byte	4	7	sfr - byte	×	×	×
		r, r	2	3	r - r	×	×	×
A, saddr		2	3/5	A - (saddr)	×	×	×	
A, sfr		3	7	A - sfr	×	×	×	
A,  r4		2	7	A - (FE00H + r4) r4 = 00H - FFH	×	×	×	
A,  HL		2	8/10	A - (HL)	×	×	×	
A,  DE		2	8/10	A - (DE)	×	×	×	
16ビット演算命令	ADDW	AX, #word	3	4	AX, CY ← AX + word	×	×	×
		AX, rp	2	6	AX, CY ← AX + rp	×	×	×
		AX, saddrp	2	7/11	AX, CY ← AX + saddrp	×	×	×
		AX, sfrp	3	13	AX, CY ← AX + sfrp	×	×	×
	SUBW	AX, #word	3	4	AX, CY ← AX - word	×	×	×
		AX, rp	2	6	AX, CY ← AX - rp	×	×	×
		AX, saddrp	2	7/11	AX, CY ← AX - saddrp	×	×	×
		AX, sfrp	3	13	AX, CY ← AX - sfrp	×	×	×
	CMPW	AX, #word	3	3	AX - word	×	×	×
		AX, rp	2	5	AX - rp	×	×	×
		AX, saddrp	2	6/10	AX - (saddrp)	×	×	×
		AX, sfrp	3	12	AX - sfrp	×	×	×
乗除算命令	MULSW	r	2	47	AX, r1 ← AX × r1			
	MULUW	r	2	47	AX, r ← AX × r			
	DIVUW	r	2	74	AX(商), r(余り) ← AX ÷ r			

命令群	モニタ	オペランド	バイト	クロック	オペレーション	フラグ		
						Z	AC CY	
増減命令	INC	r	1	2	r ← r + 1	×	×	
		saddr	2	3/7	(saddr) ← (saddr) + 1	×	×	
	DEC	r	1	2	r ← r - 1	×	×	
		saddr	2	3/7	(saddr) ← (saddr) - 1	×	×	
	INCW	rp	1	3	rp ← rp + 1			
	DECW	rp	1	3	rp ← rp - 1			
シフト・ローテート命令	ROR	r, n	2	3 + 2n	((CY, r) ← (r0, r <sub>m-1</sub> ← r <sub>m</sub> ) × n)		×	
	ROL	r, n	2	3 + 2n	(CY, r) ← (r <sub>0</sub> , r <sub>m-1</sub> ← r <sub>m</sub> ) × n		×	
	RORC	r, n	2	3 + 2n	(CY ← r <sub>0</sub> , r <sub>m</sub> ← CY, r <sub>m-1</sub> ← r <sub>m</sub> ) × n		×	
	ROLC	r, n	2	3 + 2n	(CY ← r <sub>0</sub> , r <sub>m</sub> ← CY, r <sub>m-1</sub> ← r <sub>m</sub> ) × n		×	
	SHR	r, n	2	3 + 2n	(CY ← r <sub>0</sub> , r <sub>m</sub> ← 0, r <sub>m-1</sub> ← r <sub>m</sub> ) × n	×	0 ×	
	SHL	r, n	2	3 + 2n	(CY ← r <sub>0</sub> , r <sub>m</sub> ← 0, r <sub>m+1</sub> ← r <sub>m</sub> ) × n	×	0 ×	
	SHRW	rp, n	2	3 + 3n	(CY ← rp <sub>0</sub> , rp <sub>15</sub> ← 0, rp <sub>m-1</sub> ← rp <sub>m</sub> ) × n	×	0 ×	
	SHLW	rp, n	2	3 + 3n	(CY ← rp <sub>0</sub> , rp <sub>15</sub> ← 0, rp <sub>m+1</sub> ← rp <sub>m</sub> ) × n	×	0 ×	
	ROR4	r4	2	22	A <sub>3,0</sub> ← (r4) <sub>3,0</sub> ; (r4) <sub>7,4</sub> ← A <sub>3,0</sub> ; (r4) <sub>11,8</sub> ← (r4) <sub>7,4</sub> ; r4 ← 40H - FFH			
	ROL4	r4	2	23	A <sub>3,0</sub> ← (r4) <sub>7,4</sub> ; (r4) <sub>11,8</sub> ← A <sub>3,0</sub> ; (r4) <sub>15,12</sub> ← (r4) <sub>11,8</sub> ; r4 ← 40H - FFH			
B補正命令	ADJBA		1	3	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		1	3	Decimal Adjust Accumulator after Subtract	×	×	×
ビット操作命令	MOV1	CY, saddr.bit	3	5/7	CY ← (saddr.bit)		×	
		CY, sfr.bit	3	7	CY ← sfr.bit		×	
		CY, A.bit	2	5	CY ← A.bit		×	
		CY, X.bit	2	5	CY ← X.bit		×	
		CY, PSW.bit	2	5	CY ← PSW.bit		×	
		saddr.bit, CY	3	8/12	(saddr.bit) ← CY			
		sfr.bit, CY	3	12	sfr.bit ← CY			
		A.bit, CY	2	8	A.bit ← CY			
		X.bit, CY	2	8	X.bit ← CY			
	PSW.bit, CY	2	7	PSW.bit ← CY		×	×	
	AND1	CY, saddr.bit	3	5/7	CY ← CY ∧ (saddr.bit)		×	
		CY, /saddr.bit	3	5/7	CY ← CY ∧ (/saddr.bit)		×	
		CY, sfr.bit	3	7	CY ← CY ∧ sfr.bit		×	
		CY, /sfr.bit	3	7	CY ← CY ∧ (/sfr.bit)		×	
CY, A.bit		2	5	CY ← CY ∧ A.bit		×		
CY, /A.bit	2	5	CY ← CY ∧ (/A.bit)		×			
CY, X.bit	2	5	CY ← CY ∧ X.bit		×			
CY, /X.bit	2	5	CY ← CY ∧ (/X.bit)		×			
CY, PSW.bit	2	5	CY ← CY ∧ PSW.bit		×			
CY, /PSW.bit	2	5	CY ← CY ∧ (/PSW.bit)		×			

# 保守 / 廃止

命令群	ニモニック	オペランド	バイト	クロック	オペレーション	フラグ	
						Z	AC CY
ヒット	ORI	CY, saddr. bit	3	5/7	CY ← CY ∨ (saddr. bit)		×
		CY, saddr. bit	3	5/7	CY ← CY ∨ (saddr. bit)		×
		CY, sfr. bit	3	7	CY ← CY ∨ sfr. bit		×
		CY, sfr. bit	3	7	CY ← CY ∨ sfr. bit		×
		CY, A. bit	2	5	CY ← CY ∨ A. bit		×
		CY, A. bit	2	5	CY ← CY ∨ A. bit		×
		CY, X. bit	2	5	CY ← CY ∨ X. bit		×
		CY, X. bit	2	5	CY ← CY ∨ X. bit		×
		CY, PSW. bit	2	5	CY ← CY ∨ PSW. bit		×
		CY, /PSW. bit	2	5	CY ← CY ∨ PSW. bit		×
ビット	XORI	CY, saddr. bit	3	5/7	CY ← CY ⊕ (saddr. bit)		×
		CY, sfr. bit	3	7	CY ← CY ⊕ sfr. bit		×
		CY, A. bit	2	5	CY ← CY ⊕ A. bit		×
		CY, X. bit	2	5	CY ← CY ⊕ X. bit		×
		CY, PSW. bit	2	5	CY ← CY ⊕ PSW. bit		×
操作	SETI	saddr. bit	2	3/7	(saddr. bit) ← 1		
		sfr. bit	3	10	sfr. bit ← 1		
		A. bit	2	6	A. bit ← 1		
		X. bit	2	6	X. bit ← 1		
命令	CLR1	saddr. bit	2	3/7	(saddr. bit) ← 0		
		sfr. bit	3	10	sfr. bit ← 0		
		A. bit	2	6	A. bit ← 0		
		X. bit	2	6	X. bit ← 0		
		PSW. bit	2	5	PSW. bit ← 0	×	×
命令	NOTI	saddr. bit	3	6/10	(saddr. bit) ← (saddr. bit)		
		sfr. bit	3	10	sfr. bit ← sfr. bit		
		A. bit	2	6	A. bit ← A. bit		
		X. bit	2	6	X. bit ← X. bit		
		PSW. bit	2	5	PSW. bit ← PSW. bit	×	×
SETI	CY	1	2	CY ← 1		1	
CLR1	CY	1	2	CY ← 0		0	
NOTI	CY	1	2	CY ← CY		×	
コール	CALL	*addr16	3	11/15	(SP-1) ← (PC+3) <sub>H</sub> , (SP-2) ← (PC+3) <sub>L</sub> , PC ← addr16, SP ← SP-2		
		rp	2	12/16	(SP-1) ← (PC+2) <sub>H</sub> , (SP-2) ← (PC+2) <sub>L</sub> , PC ← rp, SP ← SP-2		
リターン	CALLF	*addr11	2	11/15	(SP-1) ← (PC+2) <sub>H</sub> , (SP-2) ← (PC+2) <sub>L</sub> , PC <sub>12-11</sub> ← 01, PC <sub>10-0</sub> ← addr11, SP ← SP-2		
命令	CAI.IT	[addr5]	1	14/18	(SP-1) ← (PC+1) <sub>H</sub> , (SP-2) ← (PC+1) <sub>L</sub> , PC <sub>H</sub> ← (00000000, addr5+1), PC <sub>L</sub> ← (00000000, addr5), SP ← SP-2		
	RET		1	10/14	PC <sub>L</sub> ← (SP), PC <sub>H</sub> ← (SP+1), SP ← SP+2		
	RETI		1	15/21	PC <sub>L</sub> ← (SP), PC <sub>H</sub> ← (SP+1), PSW ← (SP+2), SP ← SP+3	R	R R R

命令群	ニモニック	オペランド	バイト	クロック	オペレーション	フラグ	
						Z	AC CY
スタック操作命令	PUSH	rp	1	8/12	(SP-1) ← rp, (SP-2) ← rp, SP ← SP-2		
		PSW	1	5/7	(SP-1) ← PSW, SP ← SP-1		
	POP	rp	1	11/15	PSW ← (SP), SP ← SP+2		
		PSW	1	6/8	PSW ← (SP), SP ← SP+1	R	R R
	MOVW	SP, #word	4	8	SP ← word		
		SP, AX	2	9	SP ← AX		
AX, SP		2	10	AX ← SP			
無条件分岐命令	BR	*addr16	3	5	PC ← addr16		
		rp	2	6	PC <sub>H</sub> ← rp <sub>H</sub> , PC <sub>L</sub> ← rp <sub>L</sub>		
		Saddr16	2	4	PC ← PC+2+jdisp8		
条件命令	BC	Saddr16	2	4/2	PC ← PC+2+jdisp8 if CY=1		
		HL					
	BNC	Saddr16	2	4/2	PC ← PC+2+jdisp8 if CY=0		
	BNL	Saddr16	2	4/2	PC ← PC+2+jdisp8 if Z=1		
	BZ	Saddr16	2	4/2	PC ← PC+2+jdisp8 if Z=1		
	BNZ	Saddr16	2	4/2	PC ← PC+2+jdisp8 if Z=0		
	BT	saddr. bit, Saddr16	3	7(5)	PC ← PC+3+jdisp8 if (saddr. bit)=1		
		sfr. bit, Saddr16	4	9(7)	PC ← PC+4+jdisp8 if sfr. bit=1		
		A. bit, Saddr16	3	7(5)	PC ← PC+3+jdisp8 if A. bit=1		
		X. bit, Saddr16	3	7(5)	PC ← PC+3+jdisp8 if X. bit=1		
BF	PSW. bit, Saddr16	3	7(5)	PC ← PC+3+jdisp8 if PSW. bit=1			
	saddr. bit, Saddr16	4	7(5)	PC ← PC+4+jdisp8 if (saddr. bit)=0			
BTCLR	sfr. bit, Saddr16	4	9(7)	PC ← PC+4+jdisp8 if sfr. bit=0			
	A. bit, Saddr16	3	7(5)	PC ← PC+3+jdisp8 if A. bit=0			
	X. bit, Saddr16	3	7(5)	PC ← PC+3+jdisp8 if X. bit=0			
	PSW. bit, Saddr16	3	7(5)	PC ← PC+3+jdisp8 if PSW. bit=0			
DBNZ	saddr. bit, Saddr16	4	9(5)	PC ← PC-4+jdisp8 if (saddr. bit)=1 then reset (saddr. bit)			
	sfr. bit, Saddr16	4	13(7)	PC ← PC-4+jdisp8 if sfr. bit=1 then reset sfr. bit			
	A. bit, Saddr16	3	9(5)	PC ← PC+3+jdisp8 if A. bit=1 then reset A. bit			
	X. bit, Saddr16	3	9(5)	PC ← PC+3+jdisp8 if X. bit=1 then reset X. bit			
CPU制御命令	MOV	STBC, #byte	4	9	STBC ← byte		
	SEL	Rn	2	2	RBS1 ← 0-n		
	NOP		1	2	No Operation		
	EI		1	2	IE ← 1 (Enable Interrupt)		
	DI		1	2	IE ← 0 (Disable Interrupt)		