

SANYO

三洋半導体ニュース

No. 4457

30893

新

# LC66562B CMOS LSI 12Kバイト/16Kバイト ROM内蔵

# LC66566B 制御用4ビット1チップマイクロコンピュータ

## 概要

LC66562B, 66566Bは、64ピンのパッケージにROM, RAM, I/Oポート, シリアルインタフェース, コンパレータ入力, 3値入力, タイマ, 割込み機能などの制御用コントローラとしての機能を全て1チップに集積したCMOS4ビットマイクロコンピュータである。

従来のLC66566Aシリーズに比べ、電源電圧範囲や一部の電気的特性等が異なる。

## 特長および機能

- (1)ROM容量12Kバイト/16Kバイト, RAM容量512×4ビット内蔵。
- (2)LC66000シリーズ共通の128種の命令セット
- (3)I/Oポート………54本
- (4)8ビットシリアルインタフェース………2本(16ビット・カスケード接続可能)
- (5)命令サイクル………0.92~10 $\mu$ s(3~5.5V)
- (6)強力なタイマ機能とプリスケラ
  - 12ビットタイマによる時限タイマ, イベントカウンタ, パルス幅測定, 方形波出力
  - 8ビットタイマによる時限タイマ, イベントカウンタ, PWM出力, 方形波出力
  - 12ビットプリスケラによるタイムベース機能
- (7)強力な11要因8ベクタ割込み
  - 外部割込 6要因 3ベクタ
  - 内部割込 5要因 5ベクタ
- (8)フレキシブルなI/O機能
  - コンパレータ入力, 三値入力, 20mAドライブ出力, 15V耐圧, プルアップ/オープンドレインのオプション切換え可
- (9)暴走検出機能(ウォッチドッグタイマ)オプション
- (10)8ビット入出力機能
- (11)HALT, HOLDモードによるパワーダウン機能

次ページへ続く

## [シリーズ構成]

機種名	ピン数	ROM容量	RAM容量	パッケージ	特長
LC66304A/306A/308A	42	4K/6K/8KB	512W	DIP42S QFP48E	ノーマル版
LC66404A/406A/408A	42	4K/6K/8KB	512W	DIP42S QFP48E	4.0~6.0V/0.92 $\mu$ s
LC66506B/508B/512B/516B	64	6K/8K/12K/16KB	512W	DIP64S QFP64A	
LC66354A/356A/358A	42	4K/6K/8KB	512W	DIP42S QFP48E	低電圧版
LC66354S/356S/358S *	42	4K/6K/8KB	512W	QFP44M	2.2~5.5V/3.92 $\mu$ s
LC66556A/558A/562A/566A	64	6K/8K/12K/16KB	512W	DIP64S QFP64E	
LC66354B/356B/358B *	42	4K/6K/8KB	512W	DIP42S QFP48E	低電圧高速版
LC66556B/558B *	64	6K/8K	512W	DIP64S QFP64E	3.0~5.5V/0.92 $\mu$ s
LC66562B/566B	64	12K/16KB	512W	DIP64S QFP64E	
LC66E308	42	EPROM 8KB	512W	DIC42S窓付QFC48窓付	評価用窓付き版& OTP
LC66P308	42	OTPROM 8KB	512W	DIP42S QFP48E	4.5~5.5V/0.92 $\mu$ s
LC66E408	42	EPROM 8KB	512W	DIC42S窓付QFC48窓付	
LC66P408	42	OTPROM 8KB	512W	DIP42S QFP48E	
LC66E516	64	EPROM 16KB	512W	DIC64S窓付QFC64窓付	
LC66P516	64	OTPROM 16KB	512W	DIP64S QFP64E	

\*これらの仕様は、改良などのため変更することがあります。

\*印：開発中

46951

# LC66562B, 66566B

前ページから続く

(12)パッケージ DIP64S, QFP64E

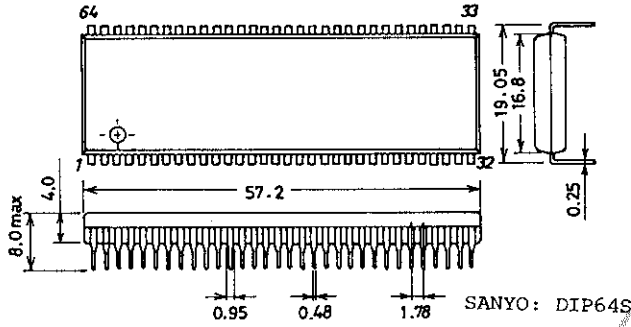
(13)評価用LSI : LC66599(エパチップ)+EVA850/800-TB665XX

LC66E516(EPROM内蔵マイコン)

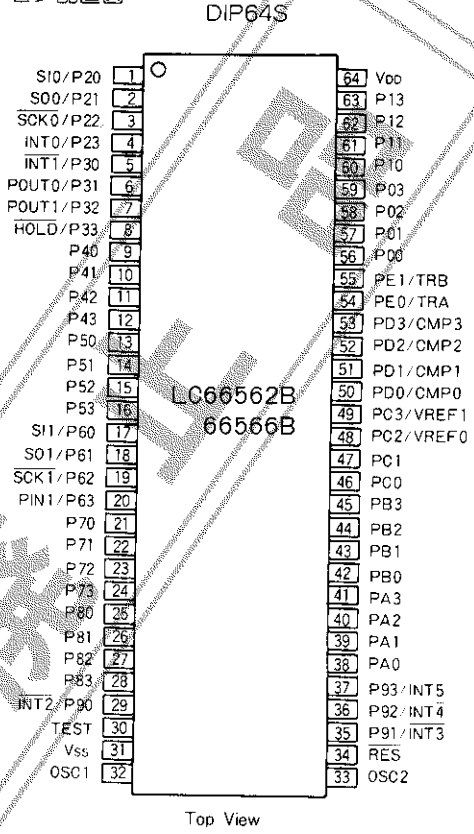
LC66P516(OTPROM内蔵マイコン)

外形図 3071

(unit : mm)



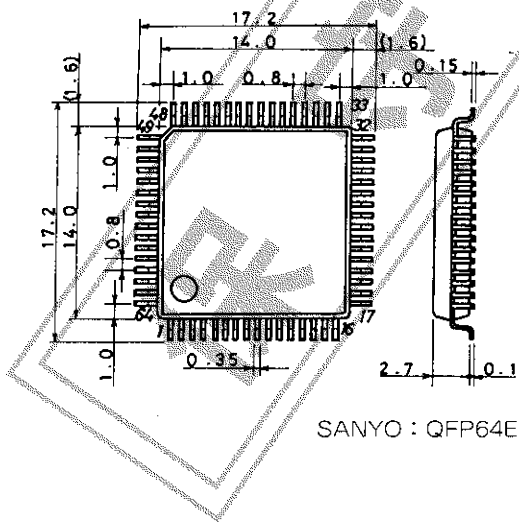
ピン配置図



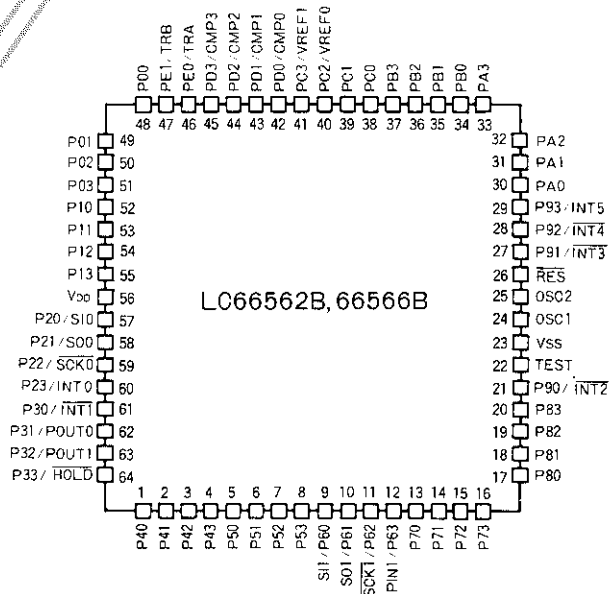
Top View

外形図 3159

(unit : mm)



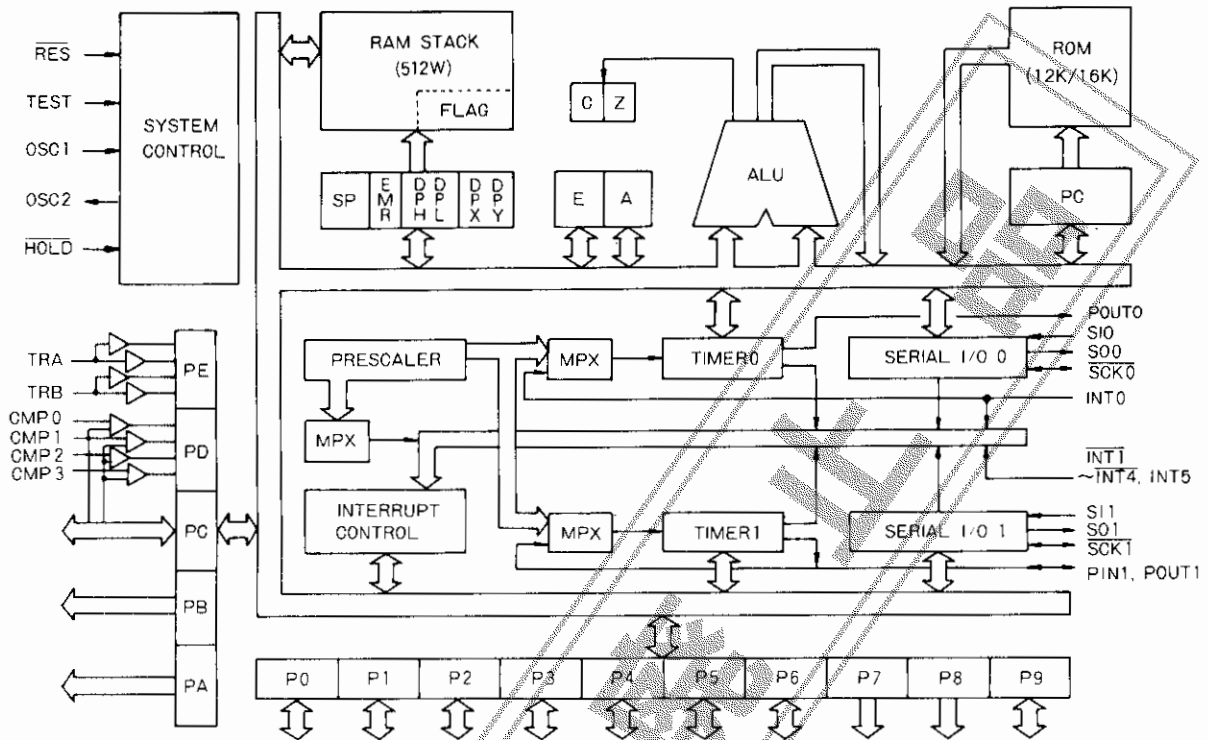
QFP64E



QFPのハンダ付け実装方法としては、リフロー法を推奨している。

ハンダディップ槽にパッケージ本体を直接浸す方法(ハンダジャブ付け)の場合は、条件等について弊社に相談すること。

システムブロック図



LC66562B, 666BとLC66516Bシリーズとの違い

項目	LC66516Bシリーズ (ツールEVA850/800-TB665XXを含む)	LC66562B, 66566B
システム上の相違点 ・HOLDモード解除時のハード的なウェイト時間(サイクル数)	66536サイクル 4MHz( $T_{cyc}=1\mu s$ )時 約64ms	16384サイクル 4MHz( $T_{cyc}=1\mu s$ )時 約16ms
・リセット時のタイマ0の値 (HOLDモード解除後の値を含む)	FF0がセットされる	FFCがセットされる
主な特性上の相違点 ・動作電源電圧/動作スピード	LC66512B, 516B 4.0~6.0V / 0.92~10 $\mu s$  LC66E516, P516 4.5~5.5V / 0.92~10 $\mu s$	3.0~5.5V / 0.92~10 $\mu s$

- ・LC66562B, 66566BではRC発振が使用できない。
- ・その他出力電流やコンパレータの入力電圧等が異なる。  
詳細は、LC66516B, LC66E516, LC66P516の単品カタログを参照すること。  
評価用としてLC66E516, P516等を使用する場合には上記の相違点に注意すること。

## 端子機能概要

端子名	入出力	機能	出力ドライバタイプ	オプション	リセット時
P00 P01 P02 P03	I/O	入出力ポートP00~P03 ・4ビットおよび1ビット単位での入力, 出力 ・P00~P03はHALTモードのコントロール機能あり	・Pch: Pu MOSタイプ ・Nch: シンク小電流タイプ	・Pu MOS付またはNch OD出力 ・リセット時出力レベル	HまたはL (オプション)
P10 P11 P12 P13	I/O	入出力ポートP10~P13 ・4ビットおよび1ビット単位での入力, 出力	・Pch: Pu MOSタイプ ・Nch: シンク小電流タイプ	・Pu MOS付またはNch OD出力 ・リセット時出力レベル	HまたはL (オプション)
P20/SI0 P21/SO0 P22/SCK0 P23/INT0	I/O	入出力ポートP20~P23 ・4ビットおよび1ビット単位での入力, 出力 ・P20はシリアル入力SI0と兼用 P21はシリアル出力SO0と兼用 P22はシリアルクロックSCK0と兼用 ・P23はINT0割込み要求およびタイマ0によるイベントカウント, バルス幅測定入力と兼用	・Pch: CMOSタイプ ・Nch: シンク小電流タイプ ・Nch OD時は+15V耐圧	・CMOSまたはNch OD出力	H
P30/INT1 P31/POUT0 P32/POUT1	I/O	入出力ポートP30~P32 ・3ビットおよび1ビット単位での入力, 出力 ・P30はINT1割込み要求と兼用 ・P31はタイマ0からの方形波出力と兼用 P32はタイマ1からの方形波出力, PWM出力と兼用	・Pch: CMOSタイプ ・Nch: シンク小電流タイプ ・Nch OD時は+15V耐圧	・CMOSまたはNch OD出力	H
P33/HOLD	I	HOLDモードコントロール入力 ・HOLD=LのときHOLD命令によりHOLDモードをセットする ・HOLDモードのとき, HOLDをHレベルにすることでCPUを再起動する。 ・P30~P32とともに入力ポートP33として使用することができる ・P33/HOLDがLレベルのときはRESをLにしてもCPUはリセットされなし。従って電源投入時にP33/HOLD=Lとなる様な応用には使用できない			
P40 P41 P42 P43	I/O	入出力ポートP40~P43 ・4ビットおよび1ビット単位での入力, 出力 ・P50~P53とともに8ビット単位での入出力 ・P50~P53とともにROMデータの8ビット出力	・Pch: Pu MOSタイプ ・Nch: シンク小電流タイプ	・Pu MOS付またはNch OD出力	H
P50 P51 P52 P53	I/O	入出力ポートP50~P53 ・4ビットおよび1ビット単位での入力, 出力 ・P40~P43とともに8ビット単位での入出力 ・P40~P43とともにROMデータの8ビット出力	・Pch: Pu MOSタイプ ・Nch: シンク小電流タイプ	・Pu MOS付またはNch OD出力	H

次ページへ続く

前ページから続く

端子名	入出力	機能	出力ドライバタイプ	オプション	リセット時
P60/SI1 P61/SO1 P62/SCK1 P63/PIN1	I/O	入出力ポートP60~P63 • 4ビットおよび1ビット単位での入力, 出力 • P60はシリアル入力SI1と兼用 • P61はシリアル出力SO1と兼用 • P62はシリアルクロックSCK1と兼用 • P63はタイマ1へのイベントカウンタ入力と兼用	• Pch: CMOSタイプ • Nch: シンク小電流タイプ • Nch OD時は+15V耐圧	• CMOSまたはNch OD出力	H
P70 P71 P72 P73	O	出力専用ポートP70~P73 • 4ビットおよび1ビット単位での出力 • 入力関係の命令では出力ラッチの内容が入力される	• Pch: Pu MOSタイプ • Nch: シンク中電流 • Nch OD時は+15V耐圧	• Pu MOS付またはNch OD出力	H
P80 P81 P82 P83	O	出力専用ポートP80~P83 • 4ビットおよび1ビット単位での出力 • 入力関係の命令では出力ラッチの内容が入力される • Pch OD出力オプションあり	• Pch: CMOSタイプ • Nch: シンク小電流	• CMOSまたはPch OD出力 • リセット時出力レベル	HまたはL (オプション)
P90/INT2 P91/INT3 P92/INT4 P93/INT5	I/O	入出力ポートP90~P93 • 4ビットおよび1ビット単位での入力, 出力 • P90はINT2割込み要求と兼用 • P91はINT3割込み要求と兼用 • P92はINT4割込み要求と兼用 • P93はINT5割込み要求と兼用	• Pch: CMOSタイプ • Nch: シンク小電流	• CMOSまたはNch OD出力	H
PA0 PA1 PA2 PA3	O	出力専用ポートPA0~PA3 • 4ビットおよび1ビット単位での出力 • 入力関係の命令では出力ラッチの内容が入力される	• Pch: Pu MOSタイプ • Nch: シンク中電流 • Nch OD時+15V耐圧	• Pu MOS付またはNch OD出力	H
PB0 PB1 PB2 PB3	O	出力専用ポートPB0~PB3 • 4ビットおよび1ビット単位での出力 • 入力関係の命令では出力ラッチの内容が入力される	• Pch: Pu MOSタイプ • Nch: シンク中電流	• Pu MOS付またはNch OD出力	H
PC0 PC1 PC2/VREF0 PC3/VREF1	I/O	入出力ポートPC0~PC3 • 4ビットおよび1ビット単位での入力, 出力 • PC2はVREF0コンパレータ比較電圧端子と兼用 • PC3はVREF1コンパレータ比較電圧端子と兼用	• Pch: CMOSタイプ • Nch: シンク小電流タイプ	• CMOSまたはNch OD出力	H
PD0/CMP0 PD1/CMP1 PD2/CMP2 PD3/CMP3	I	入力専用ポートPD0~PD3 • ソフト切換えによりコンパレータ入力となる PD0の比較電圧はVREF0 PD1~PD3の比較電圧はVREF1 PD0, PD1, (PD2, PD3)単位でコンパレータ指定			ノーマル入力
PE0/TRA PE1/TRB	I	入力専用ポート • ソフト切換えにより三値入力ポートとなる			ノーマル入力
OSC1 OSC2	I O	システムクロック発振子外付け端子 外部クロック使用時は、OSC2をオープンにしOSC1から入力		• セラミック発振, 外部クロックの選択	

次ページへ続く

前ページから続く

端子名	入出力	機能	出カドライバタイプ	オプション	リセット時
RES	I	システムリセット入力端子 ・P33/HOLDがHレベルのとき、 RESをLレベルにするとCPUがイニシャライズされる			
TEST	I	CPUのテスト用端子 動作時は必ずVssへ接続する			
VDD Vss		電源用端子			

備考：Pu MOS付出 …… ブルアップMOSトランジスタ付出力  
CMOS出力 …… コンプリメンタリ出力  
OD出力 …… オープンドレイン出力

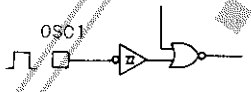
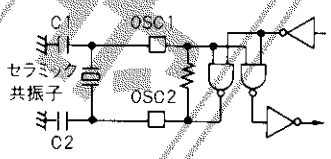
ユーザオプションの種類

(1)リセット時のポート0, 1, 8出力レベルオプション

入出力ポート0, 1, 8は、それぞれ4ビット一括で、リセット時の出力レベルを次の2つのオプションから選択できる。

オプション名	条件・その他
1.リセット時出力“H”レベル	ポート0, 1, 8の4ビット一括
2.リセット時出力“L”レベル	ポート0, 1, 8の4ビット一括

(2)発振回路オプション

オプション名	回路	条件・その他
1.外部クロック		・入力はシュミット仕様
2.セラミック発振		

注. RC発振オプションはない。

(3)ウォッチドッグタイマオプション

暴走検出機能(ウォッチドッグタイマ)の有/無をオプション選択できる。

(4)ポート出力形式オプション

i) P0, P1, P2, P3 (P33/ $\overline{\text{HOLD}}$ を除く), P4, P5, P6, P7, P9, PA, PB, PCの各ポートは、個別に次の2つのオプションを選択できる(ビット単位)。

オプション名	回路	条件・その他
1. オープンドレイン出力		P7, PA, PBは出力のみ P2, P3, P6, P9はシュミット入力
2. ブルアップ抵抗付出力		P7, PA, PBは出力のみ P2, P3, P6, P9はシュミット入力  Pch Trのドライブ能力により CMOS出力(P2, P3, P6, P9, PC) Pu MOS出力(P0, P1, P4, P5, P7, PA, PB) を区別している。

ii) P8は次の2つのオプションを選択できる(ビット単位)

オプション名	回路	条件・その他
1. オープンドレイン出力		
2. ブルダウン抵抗付出力		

iii) PDのコンパレータ入力, PEの三値入力はソフト選択である。

(1)絶対最大定格 / Ta=25℃, Vss=0V

項目	記号	適用端子, 備考	条件	規格値	unit	注記
最大電源電圧	VDD max	VDD		-0.3~+7.0	V	
入力電圧	VIN(1)	P2, P3(P33/HOLDを除く), P6		-0.3~+15.0	V	1
	VIN(2)	その他の入力		-0.3~VDD+0.3	V	2
出力電圧	VOUT(1)	P2, P3(P33/HOLDを除く), P6, P7, PA		-0.3~+15.0	V	1
	VOUT(2)	その他の出力		-0.3~VDD+0.3	V	2
一端子当り出力電流	ION(1)	P0, P1, P2, P3(P33/HOLDを除く), P4, P5, P6, P8, P9, PC		4	mA	3
	ION(2)	P7, PA, PB		20	mA	3
	-IOP(1)	P0, P1, P4, P5, P7, PA, PB		2	mA	4
	-IOP(2)	P2, P3(P33/HOLDを除く), P6, P8, P9, PC		4	mA	4
端子合計電流	ΣION(1)	P2, P3(P33/HOLDを除く), P4, P5, P6, P7, P8		75	mA	3
	ΣION(2)	P0, P1, P9, PA, PB, PC		75	mA	3
	-ΣIOP(1)	P2, P3(P33/HOLDを除く), P4, P5, P6, P7, P8		25	mA	4
	-ΣIOP(2)	P0, P1, P9, PA, PB, PC		25	mA	4
許容消費電力	Pd max	Ta=-30~+70℃	DIP64S(QFP64E)	600, 430	mW	5
動作周囲温度	Topg			-30~+70	℃	
保存周囲温度	Tstg			-55~+125	℃	

- 注 1. オープンドレイン出力仕様の時の端子に適用する。オープンドレイン出力以外の仕様の時は、その他の端子欄の規格を適用する。
2. 発振入力、出力に関しては自走発振のレベルまで許容する。
3. 吸込電流 (P8に関しては、CMOS出力仕様に対し適用する。)
4. 吐き出し電流 (P8を除く端子に関しては、プルアップ出力仕様・CMOS出力仕様に対し適用する。)
5. QFPのハンダ付け実装方法としては、リフロー法を推奨している。  
ハンダディップ槽にパッケージ本体を直接浸す方法 (ハンダジャブつけ) の場合は、条件等について弊社に相談すること。

(2)許容動作範囲 / 特に指定のない場合は、Ta=-30~+70℃, Vss=0V, VDD=3.0~5.5V

項目	記号	適用端子	条件	規格			unit	注記	
				VDD(V)	min	typ			max
動作電源電圧	VDD	VDD			3.0		5.5	V	
メモリ保持電源電圧	VDD(H)	VDD	HOLDモード時		1.8		5.5	V	
入力"H"レベル電圧	VIH(1)	P2, P3(P33/HOLDを除く), P6	出力Nch Trオフ		0.8VDD		13.5	V	1
	VIH(2)	P33/HOLD, P9, RES, OSC1	出力Nch Trオフ		0.8VDD		VDD	V	2
	VIH(3)	P0, P1, P4, P5, PC, PD, PE	出力Nch Trオフ		0.75VDD		VDD	V	3
	VIH(4)	PE	三値入力使用時		0.8VDD		VDD	V	

次ページへ続く



項目	記号	適用端子	条件	規格			unit	注記	
				V <sub>DD</sub> (V)	min	typ			max
中間レベル 入力電圧	V <sub>IM</sub>	PE	三値入力使用時		0.4V <sub>DD</sub>		0.6V <sub>DD</sub>	V	
同相入力電圧範 囲	V <sub>CMM</sub> (1)	PD0, PC2	コンパレータ 入力使用時		1.5		V <sub>DD</sub>	V	
	V <sub>CMM</sub> (2)	PD1, PD2, PD3, PC3			V <sub>SS</sub>		V <sub>DD</sub> -1.5		
入力“L”レベル 電圧	V <sub>IL</sub> (1)	P2, P3(P33/HOLD を除く), P6, P9, RES, OSC1	出力Nch Trオフ		V <sub>SS</sub>		0.2V <sub>DD</sub>	V	2
	V <sub>IL</sub> (2)	P33/HOLD		1.8~5.5	V <sub>SS</sub>		0.2V <sub>DD</sub>	V	
	V <sub>IL</sub> (3)	P0, P1, P4, P5, PC, PD, PE, TEST	出力Nch Trオフ		V <sub>SS</sub>		0.25V <sub>DD</sub>	V	3
	V <sub>IL</sub> (4)	PE	三値入力使用時		V <sub>SS</sub>		0.2V <sub>DD</sub>	V	
動作周波数 (命令サイクル タイム)	f <sub>op</sub> (T <sub>CYC</sub> )				0.4 (10)		4.35 (0.92)	MHz ( $\mu$ s)	
外部クロック 入力条件	周波数	fext	OSC1	図1による。OSC1に 入力しOSC2はオー プン(発振回路オプ ションは外部クロッ ク入力)		0.4	4.35	MHz	
	パルス幅	t <sub>extH</sub> t <sub>extL</sub>		図1による。OSC1に 入力しOSC2はオー プン(発振回路オプ ションは外部クロッ ク入力)		100		ns	
	立上り, 立下り時間	t <sub>extR</sub> t <sub>extF</sub>		図1による。OSC1に 入力しOSC2はオー プン(発振回路オプ ションは外部クロッ ク入力)			30	ns	

注 1. オープンドレイン仕様の端子に適用する。ただしP33/HOLDに対しては、V<sub>IL</sub>(2)を適用する。

CMOS出力仕様のP2, P3, P6は、入力端子として使用不可。

2. オープンドレイン仕様の端子に適用する。CMOS出力仕様のP9は入力端子として使用不可。

3. PEを三値入力として使用する時は、V<sub>IL</sub>(4)、V<sub>IM</sub>、V<sub>IL</sub>(4)を適用する。CMOS出力仕様のPCは入力端子として使用不可。

LC66562B,66566B

(3)電気的特性/特に指定のない場合は、 $T_a = -30 \sim +70^{\circ}\text{C}$ 、 $V_{SS} = 0\text{V}$ 、 $V_{DD} = 3.0 \sim 5.5\text{V}$

項目	記号	適用端子	条件	規格			unit	注記		
				$V_{DD}(\text{V})$	min	typ			max	
入力“H”レベル 電流	I <sub>IH</sub> (1)	P2,P3(P33/HOLD を除く),P6	$V_{IN}=13.5\text{V}$ 出力Nch Trオフ				5.0	$\mu\text{A}$	1	
	I <sub>IH</sub> (2)	P0,P1,P4,P5,P9, PC,OSC1,RES, P33/HOLD (PD,PE,PC2, PC3を除く)	$V_{IN}=V_{DD}$ 出力Nch Trオフ				1.0	$\mu\text{A}$	1	
	I <sub>IH</sub> (3)	PD,PE,PC2,PC3	$V_{IN}=V_{DD}$ 出力Nch Trオフ				1.0	$\mu\text{A}$	1	
入力“L”レベル 電流	I <sub>IL</sub> (1)	PD,PE,PC2,PC3 以外の入力	$V_{IN}=V_{SS}$ 出力Nch Trオフ				-1.0	$\mu\text{A}$	2	
	I <sub>IL</sub> (2)	PC2,PC3,PD,PE	$V_{IN}=V_{SS}$ 出力Nch Trオフ				-1.0	$\mu\text{A}$	2	
出力“H”レベル 電圧	V <sub>OH</sub> (1)	P2,P3(P33/HOLD を除く),P6,P8, P9,PC	$I_{OH} = -1\text{mA}$				$V_{DD}-1.0$	V	3	
			$I_{OH} = -0.1\text{mA}$				$V_{DD}-0.5$			
出力“L”レベル 電圧	V <sub>OL</sub> (2)	P0,P1,P4,P5, P7,PA,PB	$I_{OH} = -50\mu\text{A}$				$V_{DD}-1.0$	V	4	
			$I_{OH} = -30\mu\text{A}$				$V_{DD}-0.5$			
出力 プルアップ電流	I <sub>PO</sub>	P0,P1,P4,P5, P7,PA,PB	$V_{IN}=V_{SS}$	5.5			-1.6	mA	4	
出力“L”レベル 電圧	V <sub>OL</sub> (1)	P0,P1,P2,P3,P4, P5,P6,P8,P9,PC (P33/HOLDを除く)	$I_{OL} = 1.6\text{mA}$				0.4	V	5	
	V <sub>OL</sub> (2)	P7,PA,PB	$I_{OL} = 8\text{mA}$				1.5	V		
出力 オフリーク電流	I <sub>OFF</sub> (1)	P2,P3,P6,P7,PA	$V_{IN}=13.5\text{V}$				5.0	$\mu\text{A}$	6	
	I <sub>OFF</sub> (2)	(P2,P3,P6,P7,P8, PAを除く)	$V_{IN}=V_{DD}$				1.0	$\mu\text{A}$	6	
	I <sub>OFF</sub> (3)	P8	$V_{IN}=V_{SS}$				-1.0	$\mu\text{A}$	7	
コンパレータ オフセット電圧	V <sub>OFF</sub> (1)	PD1,PD2,PD3	$V_{IN}=V_{SS}$ $\sim V_{DD}-1.5\text{V}$				$\pm 50$	$\pm 300$	mV	
	V <sub>OFF</sub> (2)	PD0	$V_{IN}=1.5\text{V} \sim V_{DD}$				$\pm 50$	$\pm 300$	mV	
シミュ ット特 性	ヒステリシ ス電圧	V <sub>HIS</sub>	P2,P3,RES,P6,P9 OSC1(PC,EXT)				0.1V <sub>DD</sub>		V	
	Hレベル スレッショ ルド電圧	V <sub>t H</sub>					0.5V <sub>DD</sub>	0.8V <sub>DD</sub>	V	
	Lレベル スレッショ ルド電圧	V <sub>t L</sub>					0.2V <sub>DD</sub>	0.5V <sub>DD</sub>	V	
セラミック 発振	発振周波数	f <sub>CF</sub>	OSC1,OSC2	図2による	4MHz		4.0		MHz	
	発振安定時 間	t <sub>CF5</sub>		図3による	4MHz			10	ms	

次ページへ続く

項目	記号	適用端子	条件	規格			unit	注記	
				V <sub>DD</sub> (V)	min	typ			max
シリアルクロック	サイクルタイム	入力 出力	t <sub>CKCY</sub>	SCK0, SCK1	図4のタイミングおよび図5のテスト負荷	0.9			μs
						2.0			T <sub>CYC</sub>
	低レベル 高レベル パルス幅	入力 出力	t <sub>CKL</sub>			0.4			μs
			t <sub>CKH</sub>			1.0			T <sub>CYC</sub>
	立上り 立下り 時間	出力	t <sub>CKR</sub> t <sub>CKF</sub>				0.1		μs
シリアル入力	データセット アップ時間	t <sub>ICK</sub>	S10, S11 S10, S11	図4のタイミング SCK0, SCK1の立上り(↑)に対して規定	0.3			μs	
	データホールド 時間	t <sub>ICKI</sub>			0.3			μs	
シリアル出力	出力遅延 時間	t <sub>CKO</sub>	SO0, SO1	図5のタイミングおよび図5のテスト負荷 SCK0, SCK1の立下り(↓)に対して規定			0.3	μs	
パルス入力条件	INT0 高・低レベル パルス幅	t <sub>I0H</sub> t <sub>I0L</sub>	INT0	図6 ・INT0割込みが受付けられる条件 ・タイマ0によるイベントカウンタ/パルス幅メジャー入力が受付けられる条件	2			T <sub>CYC</sub>	
	INT0以外の 割込み入力 高・低レベル パルス幅	t <sub>I1H</sub> t <sub>I1L</sub>	INT1, INT2, INT3, INT4, INT5	・各割込みが受けられる条件	2			T <sub>CYC</sub>	
	PIN1 高・低レベル パルス幅	t <sub>P1NH</sub> t <sub>P1NL</sub>	PIN1	・タイマ1によるイベントカウント入力が受けられる	2			T <sub>CYC</sub>	
	RES 高・低レベル パルス幅	t <sub>RS<sub>H</sub></sub> t <sub>RS<sub>L</sub></sub>	RES	・リセットがかかる条件	3			T <sub>CYC</sub>	
コンパレータ 応答速度	T <sub>RS</sub>	PD	図7				20	ms	
動作モード 消費電流	I <sub>DD-OP</sub>	V <sub>DD</sub>		4 MHz セラミック発振		3.0	5.0	mA	8
				4 MHz外部クロック		3.0	5.0	mA	

次ページへ続く

項目	記号	適用端子	条件	規格			unit	注記	
				V <sub>DD</sub> (V)	min	typ			max
HALTモード 消費電流	IDDHALT	V <sub>DD</sub>	4 MHz セラミック発振			1.0	2.0	mA	
			4 MHz 外部クロック			1.0	2.0	mA	
HOLDモード 消費電流	IDDHOLD	V <sub>DD</sub>		1.8~5.5		0.01	10	μA	

- 注 1. 入出力共通ポートは、オープンドレイン出力仕様で出力Nch Trオフの時。CMOS出力仕様の際は入力端子として使用不可。
2. 入出力共通ポートは、オープンドレイン出力仕様で出力Nch Trオフの時。プルアップ出力仕様の規格は出力プルアップ電流IPOで規定。CMOS出力仕様の際は入力端子として使用不可。
3. CMOS出力仕様の時で、出力Nch Trオフ時(P8はPch オープンドレイン時も適用)。
4. プルアップ出力仕様の時で、出力Nch Trオフ時。
5. P8はCMOS出力仕様の時。
6. オープンドレイン出力仕様で出力Nch Trオフの時。
7. オープンドレイン出力仕様で出力Pch Trオフの時。
8. リセット状態。

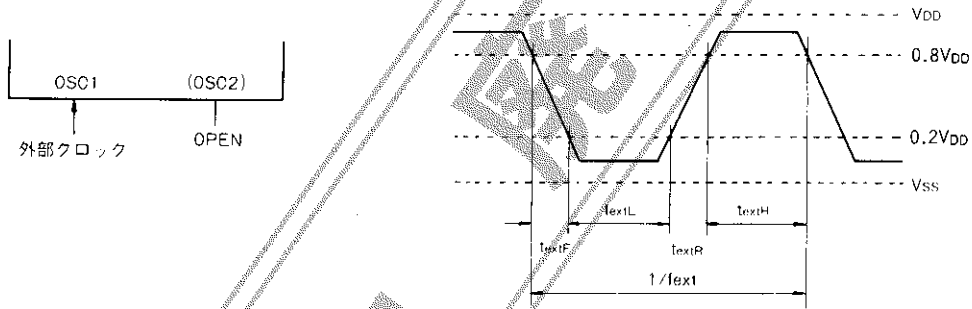


図1 外部クロック入力波形

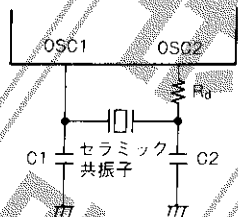


図2 セラミック発振回路

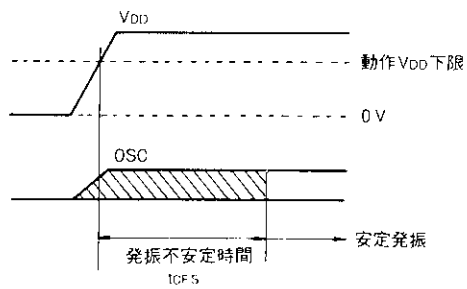


図3 発振安定時間

容量外付け	4 MHz (ムラタ) CSA4.00MG	C1 = 33pF ± 10%	4 MHz (京セラ) KBR4.0MS	C1 = 33pF ± 10%
		C1 = 33pF ± 10%		
		Rd = 0Ω		Rd = 0Ω
容量内蔵	4 MHz (ムラタ) CST4.00MG		4 MHz (京セラ) KBR4.0MES	

表1 セラミック発振保証定数

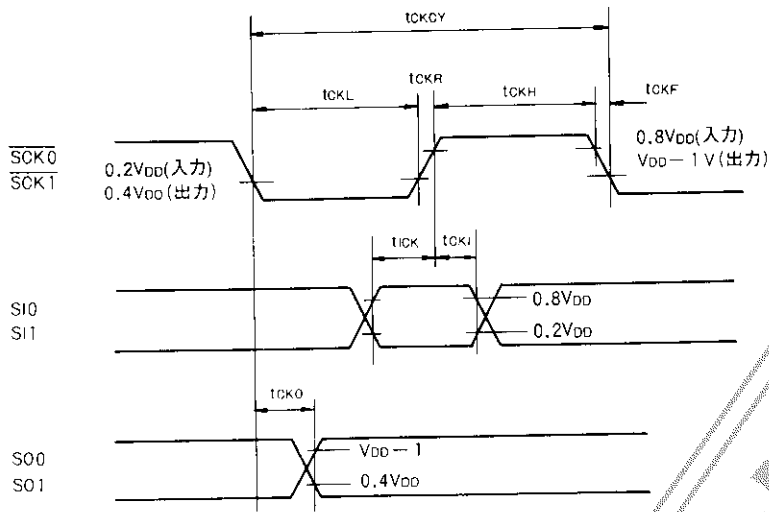


図4 シリアル入出力タイミング

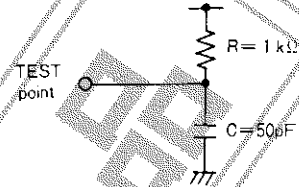


図5 タイミング負荷

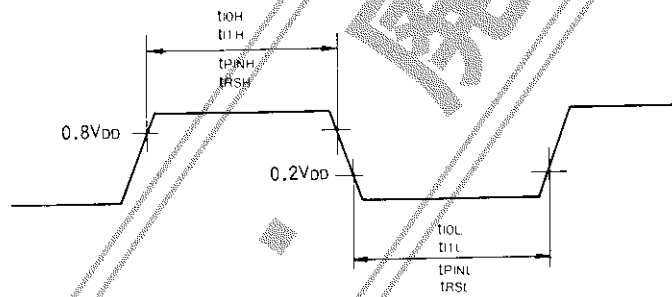


図6  $INT0$ ,  $INT1$ ,  $INT2$ ,  $INT3$ ,  $INT4$ ,  $INT5$ ,  $PIN1$ ,  $RES$ 入出力タイミング

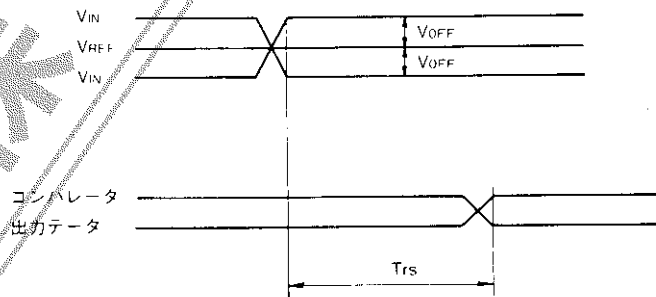
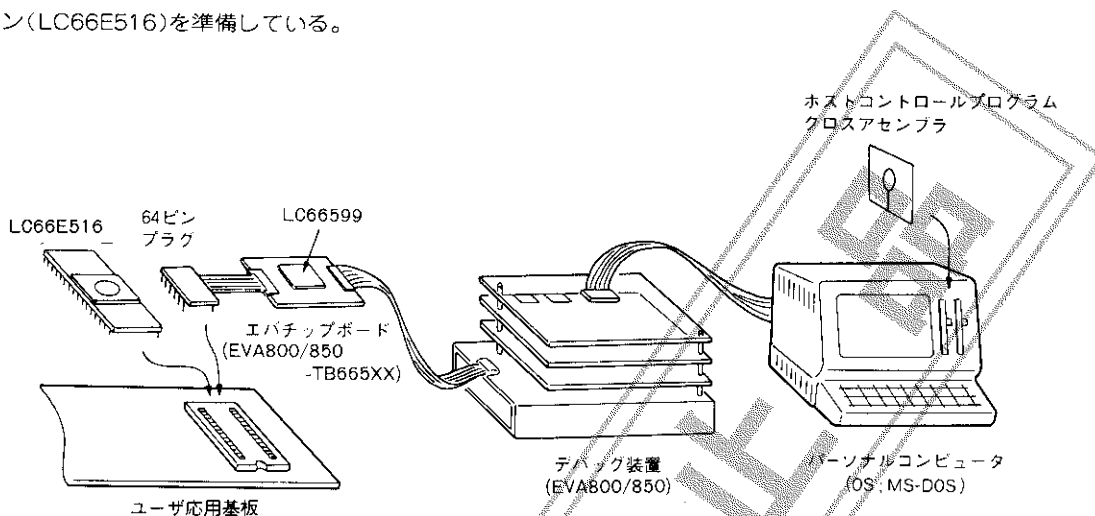


図7 コンパレータ応答速度 $T_{rs}$ のタイミング

応用開発ツール

LC66562B, 66566Bのプログラム作成は、MS-DOS (IBM-PC系)をオペレーティングシステムにもつパーソナルコンピュータで行い、クロスアセンブラが各々準備されている。また応用開発を行うのに便利のように、プログラムデバッグ装置(EVA850/800)、エバチップボード(EVA850/800-TB665XX)、エバリュエーションチップ(LC66599)、EPROM内蔵マイコン(LC66E516)を準備している。



応用開発ツールの構成

(1)プログラムデバッグ装置(EVA850/800)

EPROM WRITE機能、シリアルによる外部機器(ホストコンピュータ等)とのデータコミュニケーション機能を持ったエミュレータで、マシン語による応用開発、プログラムの修正およびデバッグが可能である。主なデバッグ機能として、ブレーク、ステップ、トレース等がある(EVA850/800のモニタROMにはMPM665XXを使用すること)。

(2)エバチップボード(EVA800/850-TB665XX)

エバチップボードはエバチップの制御信号、ポートが64ピンのコネクタに出力されており、出力ケーブルを接続すると、量産チップと同一ピン配置に変換される。エバチップボード上にはオプション設定用、状態設定用のジャンパ類があり、これらを設定することにより量産品チップと同等の入出力形式、機能を実現することができる。

ただし、HOLDモード解除タイミングや、電氣的仕様が異なる。

ジャンパ

分類	OSG		リセット方法		ユーザ応用基板への電源供給方法	
ジャンパ名	ジャンパ1 (J1)		ジャンパ2 (J2: RES)		ジャンパ3 (J3: VDD)	
ジャンパ設定およびモード	EXT	外部外付け発振(外部クロック)	INT (a)	ホストコンピュータからのRUN命令でリセット	ON (a)	エバチップボード出力をとおしてユーザ応用基板にVDDを供給する。
	RC	RC発振	EXT	ユーザ応用基板上のリセット回路でリセット	OFF (b)	ユーザ応用基板とエバチップ電源をべつにする。
	CF	CF発振	(b)			

スイッチ (SW1)

分類	ポート0、1、8のリセット時の出力レベル						ウォッチドッグタイマ有無の設定	
	P0S		P1S		P8S		WDC	
スイッチ設定およびモード	ON	ポート0 "H"	ON	ポート1 "H"	ON	ポート8 "H"	ON	ウォッチドッグタイマ有り
	OFF	ポート0 "L"	OFF	ポート1 "L"	OFF	ポート8 "L"	OFF	ウォッチドッグタイマ無し

RC0, RC1は必ず共にオンにすること。

スイッチSW2~SW14: プルアップ抵抗オプションの設定

① プルアップ抵抗を内蔵する場合はオン側に設定し、オープンドレイン出力を指定する場合はオープン側に設定する。

(SW10-ポート8は、プルダウン抵抗の選択になる。)

② 1端子毎に設定可能である。

(3)クロスアセンブラ

クロスアセンブラ名 (ファイル名)	対象機種	プログラム作成上の制限
LC66S. EXE	LC66562B/566B (LC66E516/P516) (LC66599)	SB命令の制限 ・LC66562B : SB0, SB1, SB2使用可 ・LC66566B : SB0, SB1, SB2, SB3使用可 ・LC66E516/P516 : SB0, SB1, SB2, SB3使用可 ・LC66599 : SB0, SB1, SB2, SB3使用可

(4)シミュレーションチップ(詳細についてはLC66E516半導体開発ニュースを参照すること)

シミュレーションチップ(LC66E516)は、EPROM内蔵型マイコンである。専用の変換基板(DIP用W66E516DH、QFP用W66E516QH)を使用して、市販のPROMライタでプログラムを書き込むことにより、応用製品に組込んで実装動作確認ができる。

①形状

LC66E516は、LC66562B, 66566Bと同等のピン配置、機能を持つ。

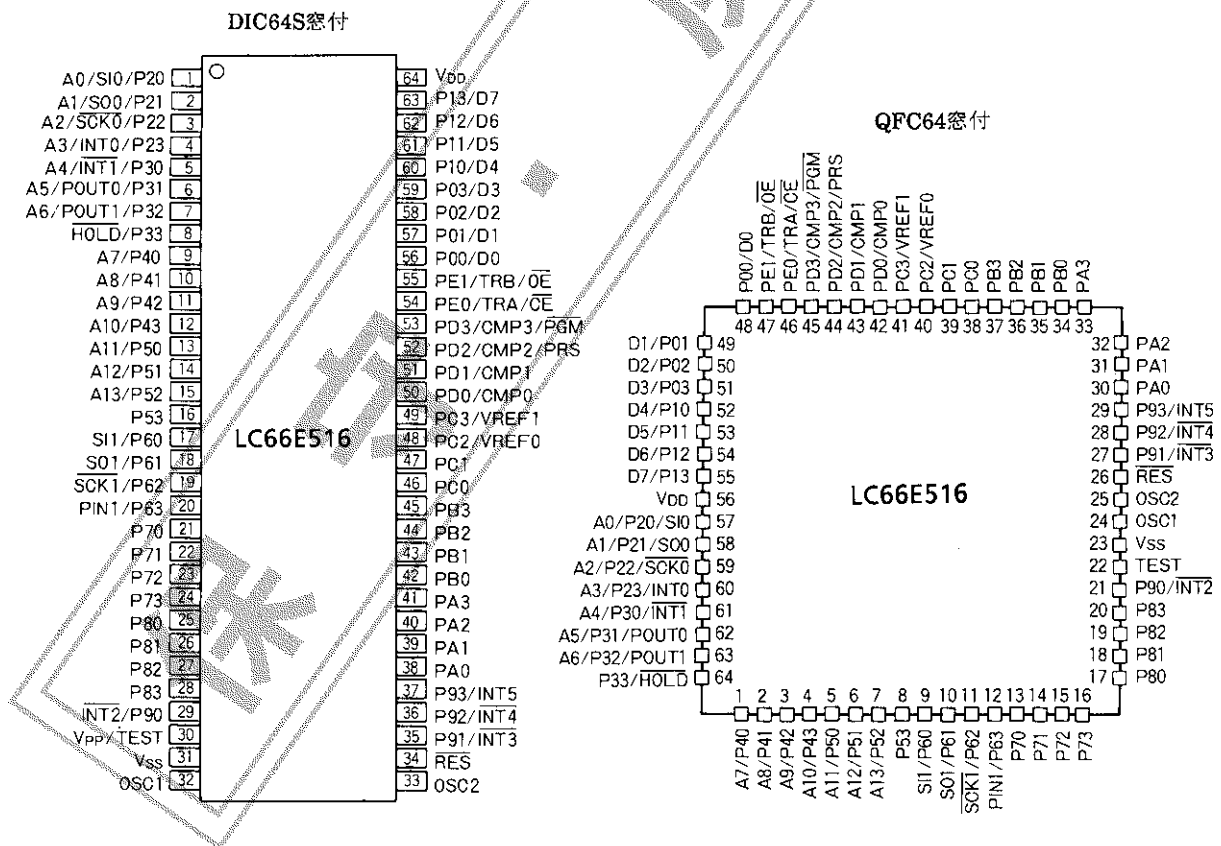
ただし、HOLDモード解除タイミングや、電気的仕様が異なる。

下図にピン配置を示す。

②オプション

評価するマイコンのオプション(ポート0, 1, 8のリセット時レベル, ウォッチドックタイマ, ポート出力形式)は、内蔵EPROMのデータによって指定可能である。これにより、量産セットの基板と同じ周辺回路を使用しての評価が可能である。

ピン配置図



## LC665XXシリーズ命令一覧(機能別)

## 略語の説明

- AC : アキュムレータ  
 E : Eレジスタ  
 CF : キャリーフラグ  
 ZF : ゼロフラグ  
 HL : データポインタDPH, DPL  
 XY : データポインタDPX, DPY  
 M : データメモリ  
 M (HL) : データポインタDPH, DPLで示されるデータメモリ  
 M (XY) : 補助データポインタDPX, DPYで示されるデータメモリ  
 M2 (HL) : データポインタDPH, DPLで示される(偶数アドレスから始まる)2ワード分のデータメモリ  
 SP : スタックポインタ  
 M2 (SP) : スタックポインタで示される2ワード分のデータメモリ  
 M4 (SP) : スタックポインタで示される4ワード分のデータメモリ  
 in : nビットのイミディエトデータ  
 t2 : ビット指定

t2	11	10	01	00
ビット	$2^3$	$2^2$	$2^1$	$2^0$

- PCh : PCの8~11ビット  
 PCm : PCの4~7ビット  
 PCI : PCの0~3ビット  
 Fn : ユーザズフラグ n=0~15  
 TIMER0 : タイマ0  
 TIMER1 : タイマ1  
 SIO : シリアルレジスタ  
 P : ポート  
 P (i4) : 4ビットのイミディエトデータで示されるポート  
 INT : 割り込み許可フラグ  
 ( ), { } : 内容を示す  
 ← : 転送方向, 結果  
 ∨ : 排他的理和  
 ∧ : 論理積  
 ∨ : 論理和  
 + : 加算  
 - : 減算  
 ~ : 1の補数をとる



命令群	ニーモニック	命令コード		バイト数	サイクル数	動作	動作説明	変化するステータス	備考	
		D <sub>7</sub> :D <sub>6</sub> :D <sub>5</sub> :D <sub>4</sub>	D <sub>3</sub> :D <sub>2</sub> :D <sub>1</sub> :D <sub>0</sub>							
アキユムレータ操作命令	CLA	Clear AC	1 0 0 0	0 0 0 0	1	1	AC ← 0 (LAI 0と等価)	ACをクリアする	ZF	ただ横スキップ機能あり
	DAA	Decimal adjust AC in addition	1 1 0 0 0 0 1 0	1 1 1 1 0 1 1 0	2	2	AC ← (AC) + 6 (ADI 6と等価)	ACに6を加える	ZF	
	DAS	Decimal adjust AC in subtraction	1 1 0 0 0 0 1 0	1 1 1 1 1 0 1 0	2	2	AC ← (AC) - 10 (ADI 0AHと等価)	ACに10を加える	ZF	
	CLC	Clear CF	0 0 0 1	1 1 1 0	1	1	CF ← 0	CFをクリアする	CF	
	STC	Set CF	0 0 0 1	1 1 1 1	1	1	CF ← 1	CFをセットする	CF	
	CMA	Complement AC	0 0 0 1	1 0 0 0	1	1	AC ← (AC)	ACの1の補数をとる	ZF	
	IA	Increment AC	0 0 0 1	0 1 0 0	1	1	AC ← (AC) + 1	ACを+1する	ZF, CF	
	DA	Decrement AC	0 0 1 0	0 1 0 0	1	1	AC ← (AC) - 1	ACを-1する	ZF, CF	
	RAR	Rotate AC right through CF	0 0 0 1	0 0 0 0	1	1	AC ← (CF), ACn ← (A Cn - 1), CF ← (AC)	CFを含めてACを右シフトする	CF	
	RAL	Rotate AC left through CF	0 0 0 0	0 0 0 1	1	1	AC ← (CF), ACn ← 1 - (ACn), CF ← (AC)	CFを含めてACを左シフトする	CF, ZF	
	TAE	Transfer AC to E	0 1 0 0	0 1 0 1	1	1	E ← (AC)	ACの内容をEに転送する		
	TEA	Transfer E to AC	0 1 0 0	0 1 1 0	1	1	AC ← (E)	Eの内容をACに転送する	ZF	
	XAE	Exchange AC with E	0 1 0 0	0 1 0 0	1	1	(AC) ← (E)	ACとEの内容を交換する		
	IM	Increment M	0 0 0 1	0 0 1 0	1	1	M(HL) ← (M(HL)) + 1	M(HL)を+1する	ZF, CF	
DM	Decrement M	0 0 1 0	0 0 1 0	1	1	M(HL) ← (M(HL)) - 1	M(HL)を-1する	ZF, CF		
メモリ操作命令	IMDR #8	Increment M direct	1 1 0 0 1 <sub>7</sub> :1 <sub>6</sub> :1 <sub>5</sub> :1 <sub>4</sub>	0 1 1 1 1 <sub>7</sub> :1 <sub>6</sub> :1 <sub>5</sub> :1 <sub>4</sub>	2	2	M(#8) ← (M(#8)) + 1	M(#8)を+1する	ZF, CF	
	DMDR #8	Decrement M direct	1 1 0 0 1 <sub>7</sub> :1 <sub>6</sub> :1 <sub>5</sub> :1 <sub>4</sub>	0 0 1 1 1 <sub>7</sub> :1 <sub>6</sub> :1 <sub>5</sub> :1 <sub>4</sub>	2	2	M(#8) ← (M(#8)) - 1	M(#8)を-1する	ZF, CF	
	SMB #2	Set M data bit	0 0 0 0	1 1 1 1	1	1	M(HL) ← (M(HL))   (12) ← 1	M(HL)の(1,10)で指定されるビットをセットする		
	RMB #2	Reset M data bit	0 0 1 0	1 1 1 1	1	1	M(HL) ← (M(HL)) & (12) ← 0	M(HL)の(1,10)で指定されるビットをリセットする	ZF	
算術命令	AD	Add M to AC	0 0 0 0	0 1 1 0	1	1	AC ← (AC) + (M(HL))	ACとM(HL)の内容を2進加算し、結果をACへ入れる	ZF, CF	
	ADDR #8	Add M direct to AC	1 1 0 0 1 <sub>7</sub> :1 <sub>6</sub> :1 <sub>5</sub> :1 <sub>4</sub>	1 0 0 1 1 <sub>7</sub> :1 <sub>6</sub> :1 <sub>5</sub> :1 <sub>4</sub>	2	2	AC ← (AC) + (M(#8))	ACとM(#8)の内容を2進加算し、結果をACへ入れる	ZF, CF	
	ADC	Add M to AC with CF	0 0 0 0	0 1 0 0	1	1	AC ← (AC) + (M(HL)) + (CF)	ACとM(HL)とCFの内容を2進加算し、結果をACへ入れる	ZF, CF	
	ADI #4	Add immediate data to AC	1 1 0 0 0 0 1 0	1 1 1 1 1 <sub>7</sub> :1 <sub>6</sub> :1 <sub>5</sub> :1 <sub>4</sub>	2	2	AC ← (AC) + (1 <sub>7</sub> :1 <sub>6</sub> :1 <sub>5</sub> :1 <sub>4</sub> )	ACの内容とイミディエートデータを2進加算し、結果をACへ入れる	ZF	
	SUBC	Subtract AC from M with CF	0 0 0 1	0 1 1 1	1	1	AC ← (M(HL)) - (AC) - (CF)	M(HL)からACとCFの内容を2進減算し、結果をACへ入れる	ZF, CF	ホローかあ れはCF 0 ホローから ければCF 1
	ANDA	And M with AC then store AC	0 0 0 0	0 1 1 1	1	1	AC ← (AC) & (M(HL))	ACとM(HL)の内容の論理積をとり結果をACへ入れる	ZF	
	ORA	Or M with AC then store AC	0 0 0 0	0 1 0 1	1	1	AC ← (AC)   (M(HL))	ACとM(HL)の内容の論理和をとり結果をACへ入れる	ZF	
	EXL	Exclusive or M with AC then store AC	0 0 0 1	0 1 0 1	1	1	AC ← (AC) ⊕ (M(HL))	ACとM(HL)の内容の排他的論理和をとり結果をACへ入れる	ZF	
	ANDM	And M with AC then store M	0 0 0 0	0 0 1 1	1	1	M(HL) ← (AC) & (M(HL))	ACとM(HL)の内容の論理積をとり結果をM(HL)へ入れる	ZF	
	ORM	Or M with AC then store M	0 0 0 0	0 1 0 0	1	1	M(HL) ← (AC)   (M(HL))	ACとM(HL)の内容の論理和をとり結果をM(HL)へ入れる	ZF	

命令群	一 モ ニ ッ ク		命令コード		バイト数	サイクル数	動作	動作説明	変化するステータス	備考												
	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub>	D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub>	D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>																		
演算・比較命令	CM	Compare AC with M	0 0 0 1	0 1 1 0	1	1	$\{M(HL)\} + (AC) + 1$	ACとM(HL)の内容を比較しCFとZFをセット、リセットする <table border="1"> <tr><th>大小関係</th><th>CF</th><th>ZF</th></tr> <tr><td><math>\{M(HL)\} &gt; (AC)</math></td><td>0</td><td>0</td></tr> <tr><td><math>\{M(HL)\} = (AC)</math></td><td>1</td><td>1</td></tr> <tr><td><math>\{M(HL)\} &lt; (AC)</math></td><td>1</td><td>0</td></tr> </table>	大小関係	CF	ZF	$\{M(HL)\} > (AC)$	0	0	$\{M(HL)\} = (AC)$	1	1	$\{M(HL)\} < (AC)$	1	0	ZF, CF	
	大小関係	CF	ZF																			
	$\{M(HL)\} > (AC)$	0	0																			
	$\{M(HL)\} = (AC)$	1	1																			
$\{M(HL)\} < (AC)$	1	0																				
CI i4	Compare AC with immediate data	1 1 0 0 1 0 1 0	1 1 1 1 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	$\{i_3 i_2 i_1 i_0\} - (AC) - 1$	ACとイミディエイトデータi <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub> を比較しZFとCFをセット、リセットする <table border="1"> <tr><th>大小関係</th><th>CF</th><th>ZF</th></tr> <tr><td>i<sub>3</sub> i<sub>2</sub> i<sub>1</sub> i<sub>0</sub> &gt; AC</td><td>0</td><td>0</td></tr> <tr><td>i<sub>3</sub> i<sub>2</sub> i<sub>1</sub> i<sub>0</sub> = AC</td><td>1</td><td>1</td></tr> <tr><td>i<sub>3</sub> i<sub>2</sub> i<sub>1</sub> i<sub>0</sub> &lt; AC</td><td>1</td><td>0</td></tr> </table>	大小関係	CF	ZF	i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub> > AC	0	0	i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub> = AC	1	1	i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub> < AC	1	0	ZF, CF		
大小関係	CF	ZF																				
i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub> > AC	0	0																				
i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub> = AC	1	1																				
i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub> < AC	1	0																				
CLI i4	Compare DP <sub>1</sub> with immediate data	1 1 0 0 1 0 1 1	1 1 1 1 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	ZF←1 if(DP <sub>1</sub> ) = i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub> ZF←0 if(DP <sub>1</sub> ) = i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	DP <sub>1</sub> の内容とイミディエイトデータを比較し、等しい時ZFをセットし、等しくない時リセットする	ZF														
CMB i2	Compare AC bit with M data bit	1 1 0 0 1 1 0 1	1 1 1 1 0 0 i <sub>1</sub> i <sub>0</sub>	2	2	ZF←1 if(AC.i2) = (M(HL).i2) ZF←0 if(AC.i2) = (M(HL).i2)	ACとM(HL)の内容のうち命令の2ビット(i <sub>1</sub> , i <sub>0</sub> )で指定されたビットの比較をし、等しい時ZFをセットし等しくない時リセットする	ZF														
LAE	Load AC and E from M2 (HL)	0 1 0 1	1 1 0 0	1	1	AC←M(HL) E←M(HL+1)	M2(HL)の内容をACとEへロードする															
LAI i4	Load AC with immediate data	1 0 0 0	i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	1	1	AC←i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	ACにイミディエイトデータをロードする	ZF	たて横スキップ機能あり													
LADR i8	Load AC from M direct	1 1 0 0 i <sub>7</sub> i <sub>6</sub> i <sub>5</sub> i <sub>4</sub>	0 0 0 1 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	AC←{M(i8)}	ACにM(i8)の内容をロードする	ZF														
S	Store AC to M	0 1 0 0	0 1 1 1	1	1	M(HL)←(AC)	ACの内容をM(HL)へストアする															
SAE	Store AC and E to M2 (HL)	0 1 0 1	1 1 1 0	1	1	M(HL)←(AC) M(HL+1)←(E)	ACとEの内容をM2(HL)へストアする															
LA reg	Load AC from M(reg)	0 1 0 0	1 0 i <sub>0</sub> 0	1	1	AC←{M(reg)}	M(reg)の内容をACへロードする regはHLまたはXY <table border="1"> <tr><th>reg</th><th>i<sub>0</sub></th></tr> <tr><td>HL</td><td>0</td></tr> <tr><td>XY</td><td>1</td></tr> </table>	reg	i <sub>0</sub>	HL	0	XY	1	ZF								
reg	i <sub>0</sub>																					
HL	0																					
XY	1																					
LA reg.i	Load AC from M(reg) then increment reg	0 1 0 0	1 0 i <sub>0</sub> 1	1	2	AC←{M(reg)} DP <sub>1</sub> ←(DP <sub>1</sub> ) + 1 または DP <sub>1</sub> ←(DP <sub>1</sub> ) + 1	M(reg)の内容をACへロードする。 regはHLまたはXY。その後DP <sub>1</sub> またはDP <sub>1</sub> の内容を+1する。regとi <sub>0</sub> の関係はLA regと同じ	ZF	ZFはDP <sub>1</sub> またはDP <sub>1</sub> をインクリメントした結果による													
LA reg.D	Load AC from M(reg) then decrement reg	0 1 0 1	1 0 i <sub>0</sub> 1	1	2	AC←{M(reg)} DP <sub>1</sub> ←(DP <sub>1</sub> ) - 1 または DP <sub>1</sub> ←(DP <sub>1</sub> ) - 1	M(reg)の内容をACへロードする。 regはHLまたはXY。その後DP <sub>1</sub> またはDP <sub>1</sub> の内容を-1する。regとi <sub>0</sub> の関係はLA regと同じ	ZF	ZFはDP <sub>1</sub> またはDP <sub>1</sub> をデクリメントした結果による													
XA reg	Exchange AC with M (reg)	0 1 0 0	1 1 i <sub>0</sub> 0	1	1	(AC)↔{M(reg)}	ACとM(reg)の内容を交換する regはHLまたはXY <table border="1"> <tr><th>reg</th><th>i<sub>0</sub></th></tr> <tr><td>HL</td><td>0</td></tr> <tr><td>XY</td><td>1</td></tr> </table>	reg	i <sub>0</sub>	HL	0	XY	1									
reg	i <sub>0</sub>																					
HL	0																					
XY	1																					
XA reg.i	Exchange AC with M (reg) then increment reg	0 1 0 0	1 1 i <sub>0</sub> 1	1	2	(AC)↔{M(reg)} DP <sub>1</sub> ←(DP <sub>1</sub> ) + 1 または DP <sub>1</sub> ←(DP <sub>1</sub> ) + 1	ACとM(reg)の内容を交換する。regはHLまたはXY。その後DP <sub>1</sub> またはDP <sub>1</sub> の内容を+1する。regとi <sub>0</sub> の関係はXA regと同じ	ZF	ZFはDP <sub>1</sub> またはDP <sub>1</sub> をインクリメントした結果による													
XA reg.D	Exchange AC with M (reg) then decrement reg	0 1 0 1	1 1 i <sub>0</sub> 1	1	2	(AC)↔{M(reg)} DP <sub>1</sub> ←(DP <sub>1</sub> ) - 1 または DP <sub>1</sub> ←(DP <sub>1</sub> ) - 1	ACとM(reg)の内容を交換する。regはHLまたはXY。その後DP <sub>1</sub> またはDP <sub>1</sub> の内容を-1する。regとi <sub>0</sub> の関係はXA regと同じ	ZF	ZFはDP <sub>1</sub> またはDP <sub>1</sub> をデクリメントした結果による													
XADR i8	Exchange AC with M direct	1 1 0 0	1 0 0 0 i <sub>7</sub> i <sub>6</sub> i <sub>5</sub> i <sub>4</sub>	2	2	(AC)↔{M(i8)}	ACとM(i8)の内容を交換する															

命令群	ニーモニック		命令コード		バイト数	サイクル数	動作	動作説明	変化するステータス	備考	
			D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub>	D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>							
ロード・ストア命令	LEA	i8	Load E & AC with immediate data	1 1 0 0 i <sub>7</sub> i <sub>6</sub> i <sub>5</sub> i <sub>4</sub>	0 1 1 0 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	E ← i <sub>7</sub> i <sub>6</sub> i <sub>5</sub> i <sub>4</sub> AC ← i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	EとACへイミディエトデータi8をロードする		
	RTBL		Read table data from program ROM	0 1 0 1	1 0 1 0	1	2	E, AC ← (ROM(PCh, E, AC))	PCのうち下位8ビットをE, ACで置きかえた番地のROMデータをE, ACにロードする		
	RTBLP		Read table data from program ROM then output to P4,5	0 1 0 1	1 0 0 0	1	2	Port 4, 5 ← (ROM(PCh, E, AC))	PCのうち下位8ビットをE, ACで置きかえた番地のROMデータをポート4, 5に出力する		
データ・操作命令	LDZ	i4	Load DP <sub>i</sub> with zero and DP <sub>i</sub> , with immediate data respectively	0 1 1 0	i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	1	1	DP <sub>i</sub> ← 0 DP <sub>i</sub> ← i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	DP <sub>i</sub> にゼロを、DP <sub>i</sub> にイミディエトデータi4をそれぞれロードする		
	LHI	i4	Load DP <sub>i</sub> with immediate data	1 1 0 0 0 0 0 0	1 1 1 1 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	DP <sub>i</sub> ← i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	DP <sub>i</sub> にイミディエトデータi4をロードする		
	LLI	i4	Load DP <sub>i</sub> with immediate data	1 1 0 0 0 0 0 1	1 1 1 1 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	DP <sub>i</sub> ← i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	DP <sub>i</sub> にイミディエトデータi4をロードする		
	LHLI	i8	Load DP <sub>i</sub> , DP <sub>i</sub> with immediate data	1 1 0 0 i <sub>7</sub> i <sub>6</sub> i <sub>5</sub> i <sub>4</sub>	0 0 0 0 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	DP <sub>i</sub> ← i <sub>7</sub> i <sub>6</sub> i <sub>5</sub> i <sub>4</sub> DP <sub>i</sub> ← i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	DP <sub>i</sub> , DP <sub>i</sub> にイミディエトデータをロードする		
	LXYI	i8	Load DP <sub>x</sub> , DP <sub>y</sub> with immediate data	1 1 0 0 i <sub>7</sub> i <sub>6</sub> i <sub>5</sub> i <sub>4</sub>	0 0 1 0 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	DP <sub>x</sub> ← i <sub>7</sub> i <sub>6</sub> i <sub>5</sub> i <sub>4</sub> DP <sub>y</sub> ← i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	DP <sub>x</sub> , DP <sub>y</sub> にイミディエトデータをロードする		
	IL		Increment DP <sub>i</sub>	0 0 0 1	0 0 0 1	1	1	DP <sub>i</sub> ← (DP <sub>i</sub> ) + 1	DP <sub>i</sub> の内容を+1する	ZF	
	DL		Decrement DP <sub>i</sub>	0 0 1 0	0 0 0 1	1	1	DP <sub>i</sub> ← (DP <sub>i</sub> ) - 1	DP <sub>i</sub> の内容を-1する	ZF	
	IY		Increment DP <sub>y</sub>	0 0 0 1	0 0 1 1	1	1	DP <sub>y</sub> ← (DP <sub>y</sub> ) + 1	DP <sub>y</sub> の内容を+1する	ZF	
	DY		Decrement DP <sub>y</sub>	0 0 1 0	0 0 1 1	1	1	DP <sub>y</sub> ← (DP <sub>y</sub> ) - 1	DP <sub>y</sub> の内容を-1する	ZF	
	TAH		Transfer AC to DP <sub>i</sub>	1 1 0 0 1 1 1 1	1 1 1 1 0 0 0 0	2	2	DP <sub>i</sub> ← (AC)	ACの内容をDP <sub>i</sub> に転送する		
	THA		Transfer DP <sub>i</sub> to AC	1 1 0 0 1 1 1 0	1 1 1 1 0 0 0 0	2	2	AC ← (DP <sub>i</sub> )	DP <sub>i</sub> の内容をACに転送する	ZF	
	XAH		Exchange AC with DP <sub>i</sub>	0 1 0 0	0 0 0 0	1	1	(AC) ← (DP <sub>i</sub> )	ACとDP <sub>i</sub> の内容を交換する		
	TAL		Transfer AC to DP <sub>i</sub>	1 1 0 0 1 1 1 1	1 1 1 1 0 0 0 0	2	2	DP <sub>i</sub> ← (AC)	ACの内容をDP <sub>i</sub> に転送する		
	TLA		Transfer DP <sub>i</sub> to AC	1 1 0 0 1 1 1 0	1 1 1 1 0 0 0 1	2	2	AC ← (DP <sub>i</sub> )	DP <sub>i</sub> の内容をACに転送する	ZF	
	XAL		Exchange AC with DP <sub>i</sub>	0 1 0 0	0 0 0 1	1	1	(AC) ← (DP <sub>i</sub> )	ACとDP <sub>i</sub> の内容を交換する		
	TAX		Transfer AC to DP <sub>x</sub>	1 1 0 0 1 1 1 1	1 1 1 1 0 0 1 0	2	2	DP <sub>x</sub> ← (AC)	ACの内容をDP <sub>x</sub> に転送する		
TXA		Transfer DP <sub>x</sub> to AC	1 1 0 0 1 1 1 0	1 1 1 1 0 0 1 0	2	2	AC ← (DP <sub>x</sub> )	DP <sub>x</sub> の内容をACに転送する	ZF		
XAX		Exchange AC with DP <sub>x</sub>	0 1 0 0	0 0 1 0	1	1	(AC) ← (DP <sub>x</sub> )	ACとDP <sub>x</sub> の内容を交換する			
TAY		Transfer AC to DP <sub>y</sub>	1 1 0 0 1 1 1 1	1 1 1 1 0 0 1 1	2	2	DP <sub>y</sub> ← (AC)	ACの内容をDP <sub>y</sub> に転送する			
TYA		Transfer DP <sub>y</sub> to AC	1 1 0 0 1 1 1 0	1 1 1 1 0 0 1 1	2	2	AC ← (DP <sub>y</sub> )	DP <sub>y</sub> の内容をACに転送する	ZF		
XAY		Exchange AC with DP <sub>y</sub>	0 1 0 0	0 0 1 1	1	1	(AC) ← (DP <sub>y</sub> )	ACとDP <sub>y</sub> の内容を交換する			
フラグ操作命令	SFB	n4	Set flag bit	0 1 1 1	n <sub>3</sub> n <sub>2</sub> n <sub>1</sub> n <sub>0</sub>	1	1	Fn ← 1	n4で指定されるフラグをセットする		
	RFB	n4	Reset flag bit	0 0 1 1	n <sub>3</sub> n <sub>2</sub> n <sub>1</sub> n <sub>0</sub>	1	1	Fn ← 0	n4で指定されるフラグをリセットする	ZF	
ジャンプ命令	JMP	addr	Jump in the current bank	1 1 1 0 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC13-12 ← PC13, 12 PC11-0 ← P <sub>7</sub> P <sub>6</sub>	同一バンク内のイミディエトデータP12で指定される番地へジャンプする		Bank命令の直後のときはPC12(PC12)となる
	JPEA		Jump to the address stored at E and AC in the current page	0 0 1 0	0 1 1 1	1	1	PC13-PC8 ← PC13-PC8 PC7-4 ← (E) PC3-0 ← (AC)	PCのうち下位8ビットをE, ACで置きかえた番地へジャンプする		

命令群	ニーモニック		命令コード		バイト数	サイズ数	動作	動作説明	変化するステータス	備考															
	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub>	D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub>	D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>																					
ジャンプ・サブルーチン命令	CAL addr	Call subroutine	0 1 0 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	0 P <sub>10</sub> P <sub>9</sub> P <sub>8</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC13~11←0 PC10←0→P <sub>10</sub> ~P <sub>0</sub> M4(SP)←(CF, ZF, PC13~0) SP←(SP)-4	サブルーチンをコールする																	
	CZP addr	Call subroutine in the zero page	1 0 1 0	P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	1	2	PC13~6, PC10←0 PC5~2←P <sub>3</sub> ~P <sub>0</sub> M4(SP)←(CF, ZF, PC12~0) SP←(SP)-4	Bank 0の0ページ内のサブルーチンをコールする																	
	BANK	Change bank	0 0 0 1	1 0 1 1	1	1		メモリバンク、レジスタバンクを変更する																	
	PUSH reg	Push reg on M2(SP)	1 1 0 0 1 1 1 1	1 1 1 1 t <sub>1</sub> t <sub>0</sub> 0	2	2	M2(SP)←(reg) SP←(SP)-2	M2(SP)にregの内容をストアする。その後(SP) 2する。 <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>reg</td> <td>is</td> <td>in</td> </tr> <tr> <td>HL</td> <td>0</td> <td>0</td> </tr> <tr> <td>XY</td> <td>0</td> <td>1</td> </tr> <tr> <td>AE</td> <td>1</td> <td>0</td> </tr> <tr> <td>禁止</td> <td>1</td> <td>1</td> </tr> </table>	reg	is	in	HL	0	0	XY	0	1	AE	1	0	禁止	1	1		
	reg	is	in																						
	HL	0	0																						
	XY	0	1																						
	AE	1	0																						
	禁止	1	1																						
	POP reg	Pop reg off M2(SP)	1 1 0 0 1 1 1 0	1 1 1 1 t <sub>1</sub> t <sub>0</sub> 0	2	2	SP←(SP)+2 reg←(M2(SP))	SPを+2し、regにM2(SP)の内容をロードする t <sub>1</sub> t <sub>0</sub> とregの関係はPUSH regと同じ																	
RT	Return from subroutine	0 0 0 1	1 1 0 0	1	2	SP←(SP)+4 PC←(M4(SP))	サブルーチン、インタラプト処理ルーチンからリターンする。ZF, CFは復帰しない																		
RTI	Return from interrupt routine	0 0 0 1	1 1 0 1	1	2	SP←(SP)+4 PC←(M4(SP)) CF, ZF←(M4(SP))	サブルーチン、インタラプト処理ルーチンからリターンする。ZF, CFは復帰する	ZF, CF																	
BAt2 addr	Branch on AC bit	1 1 0 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	0 0 t <sub>1</sub> t <sub>0</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC7←0→P <sub>7</sub> ~P <sub>4</sub> P <sub>3</sub> ~P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> if((AC, t2)=1)	ACのイミディエータt <sub>1</sub> t <sub>0</sub> で指定されるビットが1のとき同一ページ内のP <sub>7</sub> ~P <sub>4</sub> で指定される番地へ分岐する																		
BNA12 addr	Branch on no AC bit	1 0 0 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	0 0 t <sub>1</sub> t <sub>0</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC7←0→P <sub>7</sub> ~P <sub>4</sub> P <sub>3</sub> ~P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> if((AC, t2)=0)	ACのイミディエータt <sub>1</sub> t <sub>0</sub> で指定されるACのビットが0のとき同一ページ内のP <sub>7</sub> ~P <sub>4</sub> で指定される番地へ分岐する																		
BM12 addr	Branch on M bit	1 1 0 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	0 1 t <sub>1</sub> t <sub>0</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC7←0→P <sub>7</sub> ~P <sub>4</sub> P <sub>3</sub> ~P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> if((M(HL), t2)=1)	M(HL)のイミディエータt <sub>1</sub> t <sub>0</sub> で指定されるビットが1のとき同一ページ内のP <sub>7</sub> ~P <sub>4</sub> で指定される番地へ分岐する																		
BNM12 addr	Branch on no M bit	1 0 0 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	0 1 t <sub>1</sub> t <sub>0</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC7←0→P <sub>7</sub> ~P <sub>4</sub> P <sub>3</sub> ~P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> if((M(HL), t2)=0)	M(HL)のイミディエータt <sub>1</sub> t <sub>0</sub> で指定されるビットが0のとき同一ページ内のP <sub>7</sub> ~P <sub>4</sub> で指定される番地へ分岐する																		
分岐命令	BPt2 addr	Branch on Port bit	1 1 0 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	1 0 t <sub>1</sub> t <sub>0</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC7←0→P <sub>7</sub> ~P <sub>4</sub> P <sub>3</sub> ~P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> if((P(DP <sub>1</sub> ), t2)=1)	ポート(DP <sub>1</sub> )のイミディエータt <sub>1</sub> t <sub>0</sub> で指定されるビットが0のとき同一ページ内のP <sub>7</sub> ~P <sub>4</sub> で指定される番地へ分岐する		Bank命令と続けることにより内部コントロールレジスタにも有効 ただし読み出し可能なレジスタに限る															
	BNPt2 addr	Branch on no Port bit	1 0 0 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	1 0 t <sub>1</sub> t <sub>0</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC7←0→P <sub>7</sub> ~P <sub>4</sub> P <sub>3</sub> ~P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> if((P(DP <sub>1</sub> ), t2)=0)	ポート(DP <sub>1</sub> )のイミディエータt <sub>1</sub> t <sub>0</sub> で指定されるビットが0のとき同一ページ内のP <sub>7</sub> ~P <sub>4</sub> で指定される番地へ分岐する		同上															
	BC addr	Branch on CF	1 1 0 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	1 1 0 0 P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC7←0→P <sub>7</sub> ~P <sub>4</sub> P <sub>3</sub> ~P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> if((CF)=1)	CFが1のとき同一ページ内のP <sub>7</sub> ~P <sub>4</sub> で指定される番地へ分岐する																	
	BNC addr	Branch on no CF	1 0 0 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	1 1 0 0 P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC7←0→P <sub>7</sub> ~P <sub>4</sub> P <sub>3</sub> ~P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> if((CF)=0)	CFが0のとき同一ページ内のP <sub>7</sub> ~P <sub>4</sub> で指定される番地へ分岐する																	
	BZ addr	Branch on ZF	1 1 0 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	1 1 0 1 P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC7←0→P <sub>7</sub> ~P <sub>4</sub> P <sub>3</sub> ~P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> if((ZF)=1)	ZFが1のとき同一ページ内のP <sub>7</sub> ~P <sub>4</sub> で指定される番地へ分岐する																	
	BNZ addr	Branch on no ZF	1 0 0 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	1 1 0 1 P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC7←0→P <sub>7</sub> ~P <sub>4</sub> P <sub>3</sub> ~P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> if((ZF)=0)	ZFが0のとき同一ページ内のP <sub>7</sub> ~P <sub>4</sub> で指定される番地へ分岐する																	

命令群	一 モ ニ ッ ク		命令コード		バイト数	サイズ数	動作	動作説明	変化するステータス	備考
			D <sub>7</sub> :D <sub>6</sub> :D <sub>5</sub> :D <sub>4</sub>	D <sub>3</sub> :D <sub>2</sub> :D <sub>1</sub> :D <sub>0</sub>						
分岐命令	BFn4 addr	Branch on flag bit	1 1 1 1 P <sub>7</sub> :P <sub>6</sub> :P <sub>5</sub> :P <sub>4</sub>	n <sub>3</sub> n <sub>2</sub> n <sub>1</sub> n <sub>0</sub> P <sub>3</sub> :P <sub>2</sub> :P <sub>1</sub> :P <sub>0</sub>	2	2	PC7~0←P <sub>7</sub> :P <sub>6</sub> :P <sub>5</sub> :P <sub>4</sub> , P <sub>3</sub> :P <sub>2</sub> :P <sub>1</sub> :P <sub>0</sub> if(Fn)=1	16個のフラグのうちn <sub>3</sub> n <sub>2</sub> n <sub>1</sub> n <sub>0</sub> で指定されるフラグが1のとき同一ページ内のP <sub>7</sub> ~P <sub>0</sub> で指定される番地にジャンプする		
	BNFn4 addr	Branch on no flag bit	1 0 1 1 P <sub>7</sub> :P <sub>6</sub> :P <sub>5</sub> :P <sub>4</sub>	n <sub>3</sub> n <sub>2</sub> n <sub>1</sub> n <sub>0</sub> P <sub>3</sub> :P <sub>2</sub> :P <sub>1</sub> :P <sub>0</sub>	2	2	PC7~0←P <sub>7</sub> :P <sub>6</sub> :P <sub>5</sub> :P <sub>4</sub> , P <sub>3</sub> :P <sub>2</sub> :P <sub>1</sub> :P <sub>0</sub> if(Fn)=0	16個のフラグのうちn <sub>3</sub> n <sub>2</sub> n <sub>1</sub> n <sub>0</sub> で指定されるフラグが0のとき同一ページ内のP <sub>7</sub> ~P <sub>0</sub> で指定される番地にジャンプする		
入出力命令	IP0	Input port 0 to AC	0 0 1 0	0 0 0 0	1	1	AC←(P0)	ポート0の内容をACに入力する	ZF	
	IP	Input port to AC	0 0 1 0	0 1 1 0	1	1	AC←(P(DP <sub>1</sub> ))	ポートP(DP <sub>1</sub> )の内容をACに入力する	ZF	
	IPM	Input port to M	0 0 0 1	1 0 0 1	1	1	M(HL)←(P(DP <sub>1</sub> ))	ポートP(DP <sub>1</sub> )の内容をM(HL)に入力する		
	IPDR i4	Input port to AC direct	1 1 0 0 0 1 1 0	1 1 1 1 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	AC←(P(i4))	P(i4)の内容をACに入力する	ZF	
	IP45	Input port 4,5 to E, AC respectively	1 1 0 0 1 1 0 1	1 1 1 1 0 1 0 0	2	2	E←(P(4)) AC←(P(5))	ポートP(4),(5)の内容をE,ACにそれぞれ入力する		
	OP	Output AC to port	0 0 1 0	0 1 0 1	1	1	P(DP <sub>1</sub> )←(AC)	ACの内容をポートP(DP <sub>1</sub> )に出力する		
	OPM	Output M to port	0 0 0 1	1 0 1 0	1	1	P(DP <sub>1</sub> )←(M(HL))	M(HL)の内容をポートP(DP <sub>1</sub> )に出力する		
	OPDR i4	Output AC to port direct	1 1 0 0 0 1 1 1	1 1 1 1 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	P(i4)←(AC)	ACの内容をP(i4)に出力する		
	OP45	Output E, AC to port 4,5 respectively	1 1 0 0 1 1 0 1	1 1 1 1 0 1 0 1	2	2	P(4)←(E) P(5)←(AC)	E,ACの内容をポートP(4),P(5)にそれぞれ出力する		
	タイマ制御命令	SPB t2	Set port bit	0 0 0 0	1 0 t <sub>1</sub> t <sub>0</sub>	1	1	(P(DP <sub>1</sub> ))t <sub>2</sub> ←1	ポートP(DP <sub>1</sub> )のイミティエトテータt <sub>2</sub> で指定されるビットをセットする	
RPB t2		Reset port bit	0 0 1 0	1 0 t <sub>1</sub> t <sub>0</sub>	1	1	(P(DP <sub>1</sub> ))t <sub>2</sub> ←0	ポートP(DP <sub>1</sub> )のイミティエトテータt <sub>2</sub> で指定されるビットをリセットする	ZF	
ANDPDR i4, p4		And port with immediate data then output	1 1 0 0 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	0 1 0 1 P <sub>3</sub> :P <sub>2</sub> :P <sub>1</sub> :P <sub>0</sub>	2	2	P(P <sub>3</sub> :P <sub>0</sub> )←(P(P <sub>3</sub> :P <sub>0</sub> ))	P(P <sub>3</sub> :P <sub>0</sub> )とイミティエトテータi <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub> の論理積をとり、結果をP(P <sub>3</sub> :P <sub>0</sub> )に出力する	ZF	
ORPDR i4, p4		Or port with immediate data then output	1 1 0 0 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	0 1 0 0 P <sub>3</sub> :P <sub>2</sub> :P <sub>1</sub> :P <sub>0</sub>	2	2	P(P <sub>3</sub> :P <sub>0</sub> )←(P(P <sub>3</sub> :P <sub>0</sub> ))   i <sub>3</sub> ...i <sub>0</sub>	P(P <sub>3</sub> :P <sub>0</sub> )とイミティエトテータi <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub> の論理和をとり、結果をP(P <sub>3</sub> :P <sub>0</sub> )に出力する	ZF	
WTTM0		Write timer 0	1 1 0 0	1 0 1 0	1	2	TIMER0←(M2(HL)), (AC)	M2(HL),ACの内容をタイマ0リポートレジスタに書き込む		
WTTM1		Write timer 1	1 1 0 0 1 1 1 1	1 1 1 1 0 1 0 0	2	2	TIMER1←(E), (AC)	E,ACの内容をタイマ1リポートレジスタに書き込む		
RTIM0		Read timer 0	1 1 0 0	1 0 1 1	1	2	M2(HL), AC←(TIMER0)	タイマ0カウンタの内容をM2(HL),ACに読み込む		
RTIM1		Read timer 1	1 1 0 0 1 1 1 1	1 1 1 1 0 1 0 1	2	2	E, AC←(TIMER1)	タイマ1カウンタの内容をE,ACに読み込む		
START0		Start timer 0	1 1 0 0 1 1 1 0	1 1 1 1 0 1 1 0	2	2	Start timer 0 counter	タイマ0カウンタをスタートする		
START1		Start timer 1	1 1 0 0 1 1 1 0	1 1 1 1 0 1 1 1	2	2	Start timer 1 counter	タイマ1カウンタをスタートする		
STOP0	Stop timer 0	1 1 0 0 1 1 1 1	1 1 1 1 0 1 1 0	2	2	Stop timer 0 counter	タイマ0カウンタをストップする			
STOP1	Stop timer 1	1 1 0 0 1 1 1 1	1 1 1 1 0 1 1 1	2	2	Stop timer 1 counter	タイマ1カウンタをストップする			

命令群	ニーモニック		命令コード		バイト数	サイクル数	動作	動作説明	変化するステータス	備考
			D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub>	D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>						
割込み制御命令	MSET	Set Interrupt Master Enable Flag	1 1 0 0 0 1 0 1	1 1 0 1 0 0 0 0	2	2	MSE←1	インタラプトマスタイネーブルフラグをセットする		
	MRESET	Reset Interrupt Master Enable Flag	1 1 0 0 1 0 0 1	1 1 0 1 0 0 0 0	2	2	MSE←0	インタラプトマスタイネーブルフラグをリセットする		
	EIH i4	Enable interrupt high	1 1 0 0 0 1 0 1	1 1 0 1 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	EDIH←(EDIH)∨i4	インタラフトイネーブルフラグをセットする		
	EIL i4	Enable interrupt low	1 1 0 0 0 1 0 0	1 1 0 1 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	EDIL←(EDIL)∨i4	インタラフトイネーブルフラグをセットする		
	DIH i4	Disable interrupt high	1 1 0 0 1 0 0 1	1 1 0 1 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	EDIH←(EDIH)∧i4	インタラフトイネーブルフラグをリセットする	ZF	
	DIL i4	Disable interrupt low	1 1 0 0 1 0 0 0	1 1 0 1 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	EDIL←(EDIL)∧i4	インタラフトイネーブルフラグをリセットする	ZF	
	WTSP	Write SP	1 1 0 0 1 1 0 1	1 1 1 1 i 0 1 0	2	2	SP←(E), (AC)	E, ACの内容をSPに転送する		
	RSP	Read SP	1 1 0 0 1 1 0 1	1 1 1 1 1 0 1 1	2	2	E, AC←(SP)	SPの内容をE, ACに転送する		
スタンバイ制御命令	HALT	HALT	1 1 0 0 1 1 0 1	1 1 1 1 1 1 1 0	2	2	HALT	HALTモードをセットする		
	HOLD	HOLD	1 1 0 0 1 1 0 1	1 1 1 1 1 1 1 1	2	2	HOLD	HOLDモードをセットする		
シリアルI/O制御命令	STARTS	Start serial I/O	1 1 0 0 1 1 1 0	1 1 1 1 1 1 1 0	2	2	START SIO	SIO動作をスタートさせる		
	WTSIO	Write serial I/O	1 1 0 0 1 1 1 0	1 1 1 1 1 1 1 1	2	2	SIO←(E), (AC)	E, ACの内容をSIOに書き込む		
	RSIO	Read serial I/O	1 1 0 0 1 1 1 1	1 1 1 1 1 1 1 1	2	2	E, AC←(SIO)	SIOの内容をE, ACに読み込む		
その他	NOP	No operation	0 0 0 0	0 0 0 0	1	1	No operation	何も実行せず1フロンツァイクル消費する		
	SB i2	Select bank	1 1 0 0 1 1 0 0	1 1 1 1 0 0 i <sub>1</sub> i <sub>0</sub>	2	2	PC13, PC12←i <sub>1</sub> i <sub>0</sub>	メモリアバンクを指定する		

保 特