

アプリケーション・ノート

78K/0 シリーズ

8 ビット・シングルチップ・マイクロコンピュータ

基礎編 ()

μPD78018F サブシリーズ

μPD78018FY サブシリーズ

μPD780024A サブシリーズ

μPD780024AY サブシリーズ

μPD780034A サブシリーズ

μPD780034AY サブシリーズ

μPD78014H サブシリーズ

〔メ モ〕

目 次 要 約

第1章	概 説	...	20
第2章	ソフトウェアの基礎	...	41
第3章	システム・クロック切り替えの応用	...	61
第4章	ウォッチドッグ・タイマの応用	...	70
第5章	16ビット・タイマ/イベント・カウンタの応用	...	77
第6章	8ビット・タイマ/イベント・カウンタの応用	...	123
第7章	時計用タイマの応用	...	140
第8章	シリアル・インタフェースの応用	...	151
第9章	A/Dコンバータの応用	...	247
第10章	キー入力 of 応用	...	280
付録A	SPDチャートの説明	...	285
付録B	改版履歴	...	293

CMOSデバイスの一般的注意事項

静電気対策（MOS全般）

注意 MOSデバイス取り扱いの際は静電気防止を心がけてください。

MOSデバイスは強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、NECが出荷梱包に使用している導電性のトレイやマガジン・ケース、または導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。

また、MOSデバイスを実装したボードについても同様の扱いをしてください。

未使用入力の処理（CMOS特有）

注意 CMOSデバイスの入力レベルは固定してください。

バイポーラやNMOSのデバイスと異なり、CMOSデバイスの入力に何も接続しない状態で動作させると、ノイズなどに起因する中間レベル入力が生じ、内部で貫通電流が流れて誤動作を引き起こす恐れがあります。プルアップかプルダウンによって入力レベルを固定してください。また、未使用端子が出力となる可能性（タイミングは規定しません）を考慮すると、個別に抵抗を介してV_{DD}またはGNDに接続することが有効です。

資料中に「未使用端子の処理」について記載のある製品については、その内容を守ってください。

初期化以前の状態（MOS全般）

注意 電源投入時、MOSデバイスの初期状態は不定です。

分子レベルのイオン注入量等で特性が決定するため、初期状態は製造工程の管理外です。電源投入時の端子の出力状態や入出力設定、レジスタ内容などは保証しておりません。ただし、リセット動作やモード設定で定義している項目については、これらの動作ののちに保証の対象となります。

リセット機能を持つデバイスの電源投入後は、まずリセット動作を実行してください。

FIPは日本電気株式会社の登録商標です。

EEPROM、IEBusは日本電気株式会社の商標です。

注意： μ PD78018FY, 780024AY, 780034AYサブシリーズはI²Cバス・インタフェース回路を内蔵しています。

I²Cバス・インタフェースを使用される場合には、カスタム・コードをご発注いただく時に、事前にその旨ご申告下さい。申告に基づき、以下の特典が受けられます。

日本電気株式会社のI²Cバス対応部品をご購入いただくことにより、これらの部品をI²Cシステムに使用する実施権がフィリップス社I²C特許に基づき許諾されることになります。ただし、これらのI²Cシステムはフィリップス社によって設定されたI²C標準規格に合致しているものとします。

Purchase of NEC I²C components conveys a license under the Philips I²C Patent Rights to use these components in an I²C system, provided that the system conforms to the I²C Standard Specification as defined by Philips.

本製品のうち、外国為替および外国貿易管理法の規定により規制貨物等（または役務）に該当するものについては、日本国外に輸出する際に、同法に基づき日本国政府の輸出許可が必要です。

本資料に掲載の応用回路および回路定数は、例示的に示したものであり、量産設計を対象とするものではありません。

- 本資料の内容は予告なく変更することがありますので、最新のものであることをご確認の上ご使用ください。
 - 文書による当社の承諾なしに本資料の転載複製を禁じます。
 - 本資料に記載された製品の使用もしくは本資料に記載の情報の使用に際して、当社は当社もしくは第三者の知的財産権その他の権利に対する保証または実施権の許諾を行うものではありません。上記使用に起因する第三者所有の権利にかかわる問題が発生した場合、当社はその責を負うものではありませんのでご了承ください。
 - 本資料に記載された回路、ソフトウェア、及びこれらに付随する情報は、半導体製品の動作例、応用例を説明するためのものです。従って、これら回路・ソフトウェア・情報をお客様の機器に使用される場合には、お客様の責任において機器設計をしてください。これらの使用に起因するお客様もしくは第三者の損害に対して、当社は一切その責を負いません。
 - 当社は品質、信頼性の向上に努めていますが、半導体製品はある確率で故障が発生します。当社半導体製品の故障により結果として、人身事故、火災事故、社会的な損害等を生じさせない冗長設計、延焼対策設計、誤動作防止設計等安全設計に十分ご注意願います。
 - 当社は、当社製品の品質水準を「標準水準」、「特別水準」およびお客様に品質保証プログラムを指定して頂く「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の上ご使用願います。
 - 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
 - 特別水準：輸送機器（自動車、列車、船舶等）、交通用信号機器、防災／防犯装置、各種安全装置、生命維持を直接の目的としない医療機器
 - 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、生命維持のための装置またはシステム等
- 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。当社製品を上記の「標準水準」の用途以外でご使用をお考えのお客様は、必ず事前に当社販売窓口までご相談頂きますようお願い致します。

M7 98.8

本版で改訂された主な箇所

箇 所	内 容
全 般	対象製品として次の製品を追加 μ PD780021A, 780022A, 780023A, 780024A, 780021A(A), 780022A(A), 780023A(A), 780024A(A), 780021AY, 780022AY, 780023AY, 780024AY, 780021AY(A), 780022AY(A), 780023AY(A), 780024AY(A), 780031A, 780032A, 780033A, 780034A, 780031A(A), 780032A(A), 780033A(A), 780034A(A), 780031AY, 780032AY, 780033AY, 780034AY, 780031AY(A), 780032AY(A), 780033AY(A), 780034AY(A) 対象製品として次の製品を削除 μ PD78001B, 78002B, 78001B(A), 78002B(A), 78001BY, 78002BY, 78011B, 78012B, 78013, 78014, 78P014, 78011B(A), 78012B(A), 78013(A), 78014(A), 78011BY, 78012BY, 78013Y, 78014Y, 78P014Y, 78001, 780021, 780022, 780023, 780024, 780021Y, 780022Y, 780023Y, 780024Y, 780031, 780032, 780033, 780034, 780031Y, 780032Y, 780033Y, 780034Y, 780921, 780922, 780923, 780924, 78F0924, 780961, 780962, 780963, 780964, 78F0964
p. 64	図3 - 2 プロセッサ・クロック・コントロール・レジスタのフォーマット (μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズ)のCPUクロックの選択を一部変更, 注意を追加
p. 74	図4 - 4 ウォッチドッグ・タイマ・モード・レジスタのフォーマット (μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズ) に注を追加
p. 80	図5 - 3 16 ビット・タイマ・モード・コントロール・レジスタのフォーマット (μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズ)のTOCの出力タイミングの選択を一部変更, 注意を一部変更
p. 82	図5 - 5 16 ビット・タイマ出力コントロール・レジスタのフォーマット (μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズ)のビット5, 6を変更, 注意の一部を削除
p. 85	図5 - 9 キャプチャ/コンペア・コントロール・レジスタ0のフォーマット (μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズ)の注意を追加
p. 86	図5 - 10 プリスケアラ・モード・レジスタ0のフォーマット (μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズ)の注意を追加
p. 126	図6 - 3 タイマ・クロック選択レジスタ51のフォーマット (μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズ)の備考を追加
p. 249	図9 - 2 A/Dコンバータ・モード・レジスタのフォーマット (μ PD780024AY, 780024AYサブシリーズ)の注意を追加

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。

巻末にアンケート・コーナを設けております。このドキュメントに対するご意見をお気軽にお寄せください。

はじめに

対象者 このアプリケーション・ノートは、78K/0シリーズの製品を理解し、それぞれのシリーズの製品を用いたアプリケーション・プログラムを設計するユーザのエンジニアを対象としています。

●各サブシリーズの対象製品

μPD78018Fサブシリーズ : μPD78011F, 78012F, 78013F, 78014F, 78015F, 78016F,
78018F, 78P018F, 78011F(A), 78012F(A), 78013F(A),
78014F(A), 78015F(A), 78016F(A), 78018F(A),
78P018F(A), 78012F(A2)

μPD78018FYサブシリーズ : μPD78011FY, 78012FY, 78013FY, 78014FY, 78015FY, 78016FY,
78018FY, 78P018FY

μPD78014Hサブシリーズ : μPD78011H, 78012H, 78013H, 78014H, 78011H(A), 78012H(A),
78013H(A), 78014H(A)

μPD780024Aサブシリーズ : μPD780021A, 780022A, 780023A, 780024A, 780021A(A),
780022A(A), 780023A(A), 780024A(A)

μPD780024AYサブシリーズ : μPD780021AY, 780022AY, 780023AY, 780024AY,
780021AY(A), 780022AY(A), 780023AY(A), 780024AY(A)

μPD780034Aサブシリーズ : μPD780031A, 780032A, 780033A, 780034A, 78F0034A,
780031A(A), 780032A(A), 780033A(A), 780034A(A)

μPD780034AYサブシリーズ : μPD780031AY, 780032AY, 780033AY, 780034AY,
78F0034AY, 780031AY(A), 780032AY(A), 780033AY(A),
780034AY(A)

備考 1 . μPD78011F(A), 78012F(A), 78013F(A), 78014F(A), 78015F(A), 78016F(A), 78018F(A),
78P018F(A)は、μPD78011F, 78012F, 78013F, 78014F, 78015F, 78016F, 78018F,
78P018Fに比べて高信頼性の製品です。

2 . μPD78012F(A2)は、μPD78012Fに比べて高信頼性の製品です。

3 . μPD78011H(A), 78012H(A), 78013H(A), 78014H(A)は、μPD78011H, 78012H, 78013H,
78014Hに比べて高信頼性の製品です。

4 . μPD780021A(A), 780022A(A), 780023A(A), 780024A(A)は、μPD780021A, 780022A,
780023A, 780024Aに比べて高信頼性の製品です。

5 . μPD780021AY(A), 780022AY(A), 780023AY(A), 780024AY(A)は、μPD780021AY,
780022AY, 780023AY, 780024AYに比べて高信頼性の製品です。

6 . μPD780031A(A), 780032A(A), 780033A(A), 780034A(A)は、μPD780031A, 780032A,
780033A, 780034Aに比べて高信頼性の製品です。

7 . μPD780031AY(A), 780032AY(A), 780033AY(A), 780034AY(A)は、μPD780031AY,
780032AY, 780033AY, 780034AYに比べて高信頼性の製品です。

目的 このアプリケーション・ノートは、78K/0シリーズの製品の基礎的な機能について、プログラム例を用いてユーザに理解していただくことを目的としています。

なお、掲載しているプログラムおよびハードウェアの構成は例示的に示したものであり、量産設計を対象とするものではありません。

構 成 このアプリケーション・ノートは、大きく分けて次の内容で構成しています。

- 概説
- ソフトウェア
- ハードウェア

★ なお、アプリケーション・ノートには次に示す種類があります。

資料名	資料番号		対象サブシリーズ	内 容
	和文	英文		
78K/0シリーズ アプリケーション・ノート 基礎編 ()	このマニュアル	U12704E	μ PD78018F, 78018FY μ PD780024A, 780024AY μ PD780034A, 780034AY μ PD78014H	78K/0シリーズの製品の基本的な機能をプログラム例を用いて説明しています。
78K/0シリーズ アプリケーション・ノート 基礎編 ()	U10121J	U10121E	μ PD78044F μ PD78044H μ PD780208 μ PD780228	
78K/0シリーズ アプリケーション・ノート 基礎編 ()	U10182J	U10182E	μ PD78054, 78054Y μ PD78064, 78064Y μ PD78078, 78078Y μ PD78083 μ PD780058, 780058Y μ PD780308, 780308Y μ PD78058F, 78058FY μ PD78064B μ PD78070A, 78070AY μ PD78075B μ PD78098B	
78K/0シリーズ アプリケーション・ノート 浮動小数点演算プログラム編	U13482J	IEA-1289	78K/0シリーズの全サブシリーズ	78K/0シリーズの製品の浮動小数点演算応用プログラムについて説明しています。

注 意 このアプリケーション・ノートでは、メイン・システム・クロックが10 MHzで動作しているときの応用例およびプログラム・リストを記載しています。メイン・システム・クロックは8.38 MHz動作時のものではありません。

- ★ **読み方** このアプリケーション・ノートは、78K/0シリーズの製品を対象としていますが、それぞれのサブシリーズの製品は機能の一部が異なります。各サブシリーズに対応する章は次のとおりです。それぞれのサブシリーズの応用例を知りたい場合は、○印の章のみお読みください。

サブシリーズ 章	μ PD78018F μ PD78018FY	μ PD780024A μ PD780024AY	μ PD780034A μ PD780034AY	μ PD78014H
第1章 概説	○	○	○	○
第2章 ソフトウェアの基礎	○	○	○	○
第3章 システム・クロック切り替えの応用	○	○	○	○
第4章 ウォッチドッグ・タイマの応用	○	○	○	○
第5章 16ビット・タイマ / イベント・カウンタの応用	○	○	○	○
第6章 8ビット・タイマ / イベント・カウンタの応用	○	○	○	○
第7章 時計用タイマの応用	○	○	○	○
第8章 シリアル・インタフェースの応用	○	-	-	○
第9章 A/Dコンバータの応用	○	○	-	○
第10章 キー入力 of 応用	○	-	-	○

(A)製品と標準品は品質水準のみが異なります。

μ PD78012F(A2)は、標準品、(A)製品と動作温度範囲、DC特性、AC特性が一部異なります。詳細は個別のデータ・シートを参照してください。

このマニュアルでは、(A)製品、(A2)製品については品名を次のように読み替えてください。

μ PD78011F → μ PD78011F(A)	μ PD78012F → μ PD78012F(A)
μ PD78013F → μ PD78013F(A)	μ PD78014F → μ PD78014F(A)
μ PD78015F → μ PD78015F(A)	μ PD78016F → μ PD78016F(A)
μ PD78018F → μ PD78018F(A)	μ PD78P018F → μ PD78P018F(A)
μ PD780021A → μ PD780021A(A)	μ PD780022A → μ PD780022A(A)
μ PD780023A → μ PD780023A(A)	μ PD780024A → μ PD780024A(A)
μ PD780031A → μ PD780031A(A)	μ PD780032A → μ PD780032A(A)
μ PD780033A → μ PD780033A(A)	μ PD780034A → μ PD780034A(A)
μ PD78011H → μ PD78011H(A)	μ PD78012H → μ PD78012H(A)
μ PD78013H → μ PD78013H(A)	μ PD78014H → μ PD78014H(A)
μ PD78012F → μ PD78012F(A2)	

凡 例	データ表記の重み	: 左が上位桁, 右が下位桁
	アクティブ・ロウの表記	: $\overline{\times \times \times}$ (端子, 信号名称に上線)
	注	: 本文中につけた注の説明
	注意	: 気をつけて読んでいただきたい内容
	備考	: 本文の補足説明
	数の表記	: 2進数... $\times \times \times \times$ または $\times \times \times \times B$ 10進数... $\times \times \times \times$ 16進数... $\times \times \times \times H$

★ 品質水準

● 標準 (一般電子機器用)

μ PD78011F, 78012F, 78013F, 78014F, 78015F, 78016F, 78018F, 78P018F

μ PD78011FY, 78012FY, 78013FY, 78014FY, 78015FY, 78016FY, 78018FY, 78P018FY

μ PD780021A, 780022A, 780023A, 780024A

μ PD780021AY, 780022AY, 780023AY, 780024AY

μ PD780031A, 780032A, 780033A, 780034A, 78F0034A

μ PD780031AY, 780032AY, 780033AY, 780034AY, 78F0034AY

μ PD78011H, 78012H, 78013H, 78014H

● 特別 (高信頼度電子機器用)

μ PD78011F(A), 78012F(A), 78013F(A), 78014F(A), 78015F(A), 78016F(A), 78018F(A), 78P018F(A), 78012F(A2)

μ PD780021A(A), 780022A(A), 780023A(A), 780024A(A)

μ PD780021AY(A), 780022AY(A), 780023AY(A), 780024AY(A)

μ PD780031A(A), 780032A(A), 780033A(A), 780034A(A)

μ PD780031AY(A), 780032AY(A), 780033AY(A), 780034AY(A)

μ PD78011H(A), 78012H(A), 78013H(A), 78014H(A)

品質水準とその応用分野の詳細については当社発行の資料「NEC 半導体デバイスの品質水準」(U11531J)をご覧ください。

応用分野 ● 民生機器分野

関連資料

関連資料は暫定版の場合がありますが、この資料では「暫定」の表示をしておりません。あらかじめご了承ください。

● 共通資料一覧

資 料 名	資料番号	
	和 文	英 文
78K/0シリーズ アプリケーション・ノート 基礎編 ()	このマニュアル	U12704E
78K/0シリーズ ユーザーズ・マニュアル 命令編	U12326J	U12326E

● μPD78018F, 78018FYサブシリーズの関連資料

資 料 名	資料番号	
	和 文	英 文
μPD78018F, 78018FYサブシリーズ ユーザーズ・マニュアル	U10659J	U10659E
μPD78011F, 78012F, 78013F, 78014F, 78015F, 78016F データ・シート	U10280J	U10280E
μPD78011F(A), 78012F(A), 78013F(A), 78014F(A), 78015F(A), 78016F(A), 78018F(A) データ・シート	U11921J	U11921E
μPD78011FY, 78012FY, 78013FY, 78014FY, 78015FY, 78016FY データ・シート	U10281J	U10281E
μPD78P018F データ・シート	U10955J	U10955E
μPD78P018F(A) データ・シート	U12132J	U12132E
μPD78P018FY データ・シート	U10989J	U10989E

★ ● μPD780024A, 780024AY, 780034A, 780034AYサブシリーズの関連資料

資 料 名	資料番号	
	和 文	英 文
μPD780024A, 780034A, 780024AY, 780034AYサブシリーズ ユーザーズ・マニュアル	U14046J	U14046E
μPD780021A, 780022A, 780023A, 780024A, 780021AY, 780022AY, 780023AY, 780024AY データ・シート	U14042J	U14042E
μPD780031A, 780032A, 780033A, 780034A, 780031AY, 780032AY, 780033AY, 780034AY データ・シート	U14044J	U14044E
μPD780021A(A), 780022A(A), 780023A(A), 780024A(A), 780021AY(A), 780022AY(A), 780023AY(A), 780024AY(A) データ・シート	U15131J	U15131E
μPD780031A(A), 780032A(A), 780033A(A), 780034A(A), 780031AY(A), 780032AY(A), 780033AY(A), 780034AY(A) データ・シート	U15132J	U15132E
μPD78F0034A, 78F0034AY データ・シート	U14040J	U14040E

★ ● μ PD78014Hサブシリーズの関連資料

資 料 名	資料番号	
	和 文	英 文
μ PD78014Hサブシリーズ ユーザーズ・マニュアル	U12220J	U12220E
μ PD78011H, 78012H, 78013H, 78014H データ・シート	U11898J	U11898E
μ PD78011H(A), 78012H(A), 78013H(A), 78014H(A) データ・シート	U12174J	U12174E

なお，上記関連資料は予告なしに内容を変更することがあります。設計などには必ず最新の資料をご使用ください。

目 次

第1章 概 説 ... 20

- 1.1 78K/0シリーズの展開 ... 20
- 1.2 78K/0シリーズの特徴 ... 23

第2章 ソフトウェアの基礎 ... 41

- 2.1 データ転送 ... 41
- 2.2 データ比較 ... 42
- 2.3 10進加算 ... 43
- 2.4 10進減算 ... 50
- 2.5 2進10進変換 ... 52
- 2.6 ビット演算操作命令 ... 53
- 2.7 2進乗算(16ビット×16ビット) ... 54
- 2.8 2進除算(32ビット÷16ビット) ... 57

第3章 システム・クロック切り替えの応用 ... 61

- 3.1 RESET後のPCCの切り替え ... 66
- 3.2 パワーオン/オフ時の切り替え ... 67

第4章 ウォッチドッグ・タイマの応用 ... 70

- 4.1 ウォッチドッグ・タイマ・モードの設定 ... 75
- 4.2 インターバル・タイマ・モードの設定 ... 76

第5章 16ビット・タイマ/イベント・カウンタの応用 ... 77

- 5.1 インターバル・タイマの設定 ... 88
- 5.2 PWM出力 ... 90
- 5.3 リモコン受信 ... 92
 - 5.3.1 カウンタ・クリアによるリモコン受信 ... 94
 - 5.3.2 PWM出力とフリーランニングによるリモコン受信(μPD78018F, 78018FY, 78014Hサブシリーズのみ) ... 108

第6章 8ビット・タイマ/イベント・カウンタの応用 ... 123

- 6.1 インターバル・タイマの設定 ... 131
 - 6.1.1 8ビット・タイマの設定 ... 132
 - 6.1.2 16ビット・タイマの設定 ... 133
- 6.2 音階発生 ... 135

第7章 時計用タイマの応用 ... 140

- 7.1 時計とLED表示プログラム ... 144

第8章 シリアル・インタフェースの応用	...	151
8.1 EEPROM™ (μPD6252) とのインタフェース	...	170
8.1.1 2線式シリアルI/Oモードの通信	...	172
8.1.2 I ² Cバス・モードの通信	...	182
8.2 OSD用LSI (μPD6451A) とのインタフェース	...	192
8.3 SBIモードのインタフェース	...	197
8.3.1 マスタCPUとしての応用	...	199
8.3.2 スレーブCPUとしての応用	...	207
8.4 3線式シリアルI/Oモードのインタフェース	...	210
8.4.1 マスタCPUとしての応用	...	211
8.4.2 スレーブCPUとしての応用	...	214
8.5 半二重調歩同期通信	...	217
8.5.1 3線式シリアルI/Oモードの半二重調歩同期通信	...	217
8.5.2 SBIモードの半二重調歩同期通信	...	231
第9章 A/Dコンバータの応用	...	247
9.1 レベル・メータ	...	252
9.2 温度計	...	261
9.3 アナログ・キー入力	...	270
9.4 4チャンネル入力A/D変換	...	276
第10章 キー入力の応用	...	280
付録A SPDチャートの説明	...	285
付録B 改版履歴	...	293

図 の 目 次 (1/4)

図番号	タイトル, ページ
1 - 1	μPD78018Fサブシリーズのブロック図 ... 24
1 - 2	μPD78018FYサブシリーズのブロック図 ... 27
1 - 3	μPD780024Aサブシリーズのブロック図 ... 30
1 - 4	μPD780024AYサブシリーズのブロック図 ... 32
1 - 5	μPD780034Aサブシリーズのブロック図 ... 34
1 - 6	μPD780034AYサブシリーズのブロック図 ... 36
1 - 7	μPD78014Hサブシリーズのブロック図 ... 38
2 - 1	データ交換 ... 41
2 - 2	データ比較 ... 42
2 - 3	10進加算 ... 43
2 - 4	10進減算 ... 50
2 - 5	2進10進変換 ... 52
2 - 6	ビット演算 ... 53
2 - 7	2進乗算 ... 54
2 - 8	2進除算 ... 57
3 - 1	プロセッサ・クロック・コントロール・レジスタのフォーマット (μPD78018F, 78018FY, 78014Hサブシリーズ) ... 62
3 - 2	プロセッサ・クロック・コントロール・レジスタのフォーマット (μPD780024A, 780024AY, 780034A, 780034AYサブシリーズ) ... 64
3 - 3	RESET後のCPUクロック切り替え (μPD78018Fサブシリーズの場合) ... 66
3 - 4	システム・クロック切り替え用回路例 ... 67
3 - 5	パワーオン/オフ時のシステム・クロックの切り替え (μPD78018Fサブシリーズの場合) ... 68
4 - 1	タイマ・クロック選択レジスタ2のフォーマット (μPD78018F, 78018FY, 78014Hサブシリーズ) ... 71
4 - 2	ウォッチドッグ・タイマ・クロック選択レジスタのフォーマット (μPD780024A, 780024AY, 780034A, 780034AYサブシリーズ) ... 72
4 - 3	ウォッチドッグ・タイマ・モード・レジスタのフォーマット (μPD78018F, 78018FY, 78014Hサブシリーズ) ... 73
4 - 4	ウォッチドッグ・タイマ・モード・レジスタのフォーマット (μPD780024A, 780024AY, 780034A, 780034AYサブシリーズ) ... 74
4 - 5	ウォッチドッグ・タイマのカウント・タイミング ... 76
5 - 1	タイマ・クロック選択レジスタ0のフォーマット (μPD78018F, 78018FY, 78014Hサブシリーズ) ... 78
5 - 2	16ビット・タイマ・モード・コントロール・レジスタのフォーマット (μPD78018F, 78018FY, 78014Hサブシリーズ) ... 79
5 - 3	16ビット・タイマ・モード・コントロール・レジスタのフォーマット (μPD780024A, 780024AY, 780034A, 780034AYサブシリーズ) ... 80

図 の 目 次 (2/4)

図番号	タイトル, ページ
5 - 4	16ビット・タイマ出力コントロール・レジスタのフォーマット (μPD78018F, 78018FY, 78014Hサブシリーズ) ... 81
5 - 5	16ビット・タイマ出力コントロール・レジスタのフォーマット (μPD780024A, 780024AY, 780034A, 780034AYサブシリーズ) ... 82
5 - 6	ポート・モード・レジスタ3のフォーマット (μPD78018F, 78018FY, 78014Hサブシリーズ) ... 83
5 - 7	外部割り込みモード・レジスタのフォーマット (μPD78018F, 78018FY, 78014Hサブシリーズ) ... 83
5 - 8	サンプリング・クロック選択レジスタのフォーマット (μPD78018F, 78018FY, 78014Hサブシリーズ) ... 84
5 - 9	キャプチャ/コンペア・コントロール・レジスタ0のフォーマット (μPD780024A, 780024AY, 780034A, 780034AYサブシリーズ) ... 85
5 - 10	プリスケラ・モード・レジスタ0のフォーマット (μPD780024A, 780024AY, 780034A, 780034AYサブシリーズ) ... 86
5 - 11	ポート・モード・レジスタ7のフォーマット (μPD780024A, 780024AY, 780034A, 780034AYサブシリーズ) ... 87
5 - 12	リモコン受信回路例 ... 92
5 - 13	リモコン送信用IC出力信号 ... 93
5 - 14	受信プリアンプの出力信号 ... 93
5 - 15	リモコン信号のサンプリング ... 94
6 - 1	タイマ・クロック選択レジスタ1のフォーマット (μPD78018F, 78018FY, 78014Hサブシリーズ) ... 124
6 - 2	タイマ・クロック選択レジスタ50のフォーマット (μPD780024A, 780024AY, 780034A, 780034AYサブシリーズ) ... 125
6 - 3	タイマ・クロック選択レジスタ51のフォーマット (μPD780024A, 780024AY, 780034A, 780034AYサブシリーズ) ... 126
6 - 4	8ビット・タイマ・モード・コントロール・レジスタのフォーマット (μPD78018F, 78018FY, 78014Hサブシリーズ) ... 127
6 - 5	8ビット・タイマ・モード・コントロール・レジスタ5nのフォーマット (μPD780024A, 780024AY, 780034A, 780034AYサブシリーズ) ... 128
6 - 6	8ビット・タイマ出力コントロール・レジスタのフォーマット (μPD78018F, 78018FY, 78014Hサブシリーズ) ... 129
6 - 7	ポート・モード・レジスタ3のフォーマット (μPD78018F, 78018FY, 78014Hサブシリーズ) ... 130
6 - 8	ポート・モード・レジスタ7のフォーマット (μPD780024A, 780024AY, 780034A, 780034AYサブシリーズ) ... 130
6 - 9	8ビット・タイマのカウント・タイミング ... 131
6 - 10	音階発生回路 ... 135
6 - 11	タイマ出力とインターバル ... 135

図 の 目 次 (3/4)

図番号	タイトル, ページ
7 - 1	タイマ・クロック選択レジスタ2のフォーマット (μPD78018F, 78018FY, 78014Hサブシリーズ) ... 141
7 - 2	時計用タイマ・モード・コントロール・レジスタのフォーマット (μPD78018F, 78018FY, 78014Hサブシリーズ) ... 142
7 - 3	時計用タイマ・モード・コントロール・レジスタのフォーマット (μPD780024A, 780024AY, 780034A, 780034AYサブシリーズ) ... 143
7 - 4	時計データの概念図 ... 144
7 - 5	LED表示タイミング ... 145
7 - 6	時計用タイマの回路例 ... 145
8 - 1	タイマ・クロック選択レジスタ3のフォーマット (μPD78018F, 78014Hサブシリーズ) ... 152
8 - 2	タイマ・クロック選択レジスタ3のフォーマット (μPD78018FYサブシリーズ) ... 153
8 - 3	シリアル動作モード・レジスタ0のフォーマット (μPD78018F, 78014Hサブシリーズ) ... 154
8 - 4	シリアル動作モード・レジスタ0のフォーマット (μPD78018FYサブシリーズ) ... 156
8 - 5	シリアル動作モード・レジスタ1のフォーマット (μPD78018F, 78018FY, 78014Hサブシリーズのみ) ... 158
8 - 6	割り込みタイミング指定レジスタのフォーマット (μPD78018F, 78014Hサブシリーズ) ... 159
8 - 7	割り込みタイミング指定レジスタのフォーマット (μPD78018FYサブシリーズ) ... 160
8 - 8	シリアル・バス・インタフェース・コントロール・レジスタのフォーマット (μPD78018F, 78014Hサブシリーズ) ... 162
8 - 9	シリアル・バス・インタフェース・コントロール・レジスタのフォーマット (μPD78018FYサブシリーズ) ... 164
8 - 10	自動データ送受信コントロール・レジスタのフォーマット (μPD78018F, 78018FY, 78014Hサブシリーズのみ) ... 166
8 - 11	自動データ送受信間隔指定レジスタのフォーマット (μPD78018F, 78018FY, 78014Hサブシリーズのみ) ... 167
8 - 12	μPD6252の端子接続図 ... 170
8 - 13	μPD6252との接続例 ... 172
8 - 14	μPD6252の通信フォーマット ... 174
8 - 15	μPD6252とI ² Cバス・モードの接続例 ... 182
8 - 16	μPD6252動作タイミング ... 183
8 - 17	μPD6451Aとの接続例 ... 192
8 - 18	μPD6451Aの通信フォーマット ... 192
8 - 19	SBIモードの接続例 ... 197
8 - 20	SBIモードの通信フォーマット ... 198
8 - 21	タイム・アウトのACK信号 ... 199
8 - 22	バス・ラインのテスト ... 199

図 の 目 次 (4/4)

図番号	タイトル, ページ
8 - 23	3 線式シリアルI/Oモードの接続例 ... 210
8 - 24	3 線式シリアルI/Oモードの通信フォーマット ... 210
8 - 25	ビジィ信号の出力 ... 214
8 - 26	システム構成図 (3 線式シリアルI/Oモード) ... 217
8 - 27	3 線式シリアルI/Oモードの送信フォーマット ... 218
8 - 28	3 線式シリアルI/Oモードの受信フォーマット ... 219
8 - 29	システム構成図 (SBIモード) ... 231
8 - 30	SBIモードの送信フォーマット ... 232
8 - 31	SBIモードの受信フォーマット ... 233
9 - 1	A/Dコンバータ・モード・レジスタのフォーマット (μ PD78018F, 78018FY, 78014Hサブシリーズ) ... 248
9 - 2	A/Dコンバータ・モード・レジスタのフォーマット (μ PD780024A, 780024AYサブシリーズ) ... 249
9 - 3	A/Dコンバータ入力選択レジスタのフォーマット (μ PD78018F, 78018FY, 78014Hサブシリーズ) ... 250
9 - 4	アナログ入力チャネル指定レジスタのフォーマット (μ PD780024A, 780024AYサブシリーズ) ... 251
9 - 5	レベル・メータ回路例 ... 252
9 - 6	A/D変換結果とLED表示 ... 252
9 - 7	ピーク・ホールド概念図 ... 253
9 - 8	温度計の回路例 ... 261
9 - 9	温度と出力特性 ... 262
9 - 10	アナログ・キー入力回路例 ... 271
9 - 11	4 チャネル・スキャン・モードのタイミング・チャート ... 276
10 - 1	キー・マトリクス回路 ... 280

表 目 次

表番号	タイトル, ページ
1 - 1	μ PD78018Fサブシリーズの機能概要 ... 25
1 - 2	μ PD78018FYサブシリーズの機能概要 ... 28
1 - 3	μ PD780024Aサブシリーズの機能概要 ... 31
1 - 4	μ PD780024AYサブシリーズの機能概要 ... 33
1 - 5	μ PD780034Aサブシリーズの機能概要 ... 35
1 - 6	μ PD780034AYサブシリーズの機能概要 ... 37
1 - 7	μ PD78014Hサブシリーズの機能概要 ... 39
3 - 1	CPUクロックの切り替えに要する最大時間 ... 61
3 - 2	CPUクロックと最小命令実行時間の関係 (μ PD78018F, 78018FY, 78014Hサブシリーズ) ... 63
3 - 3	CPUクロックと最小命令実行時間の関係 (μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズ) ... 65
5 - 1	入力信号の有効時間 ... 94
5 - 2	入力信号の有効時間 ... 108
6 - 1	音階と周波数 ... 136
8 - 1	各サブシリーズとシリアル・インタフェース・チャネルの有無 ... 151
8 - 2	シリアル・インタフェースのレジスタ一覧 ... 151
8 - 3	μ PD6252の端子の説明 ... 171
8 - 4	μ PD6252のコマンド一覧 ... 173
8 - 5	SBIモードの信号一覧 ... 198
9 - 1	A/D変換値と温度 ... 263
9 - 2	入力電圧とキー・コード ... 270
9 - 3	R1からR15の抵抗値 ... 271
A - 1	SPD記号とフロー・チャートの対比 ... 285

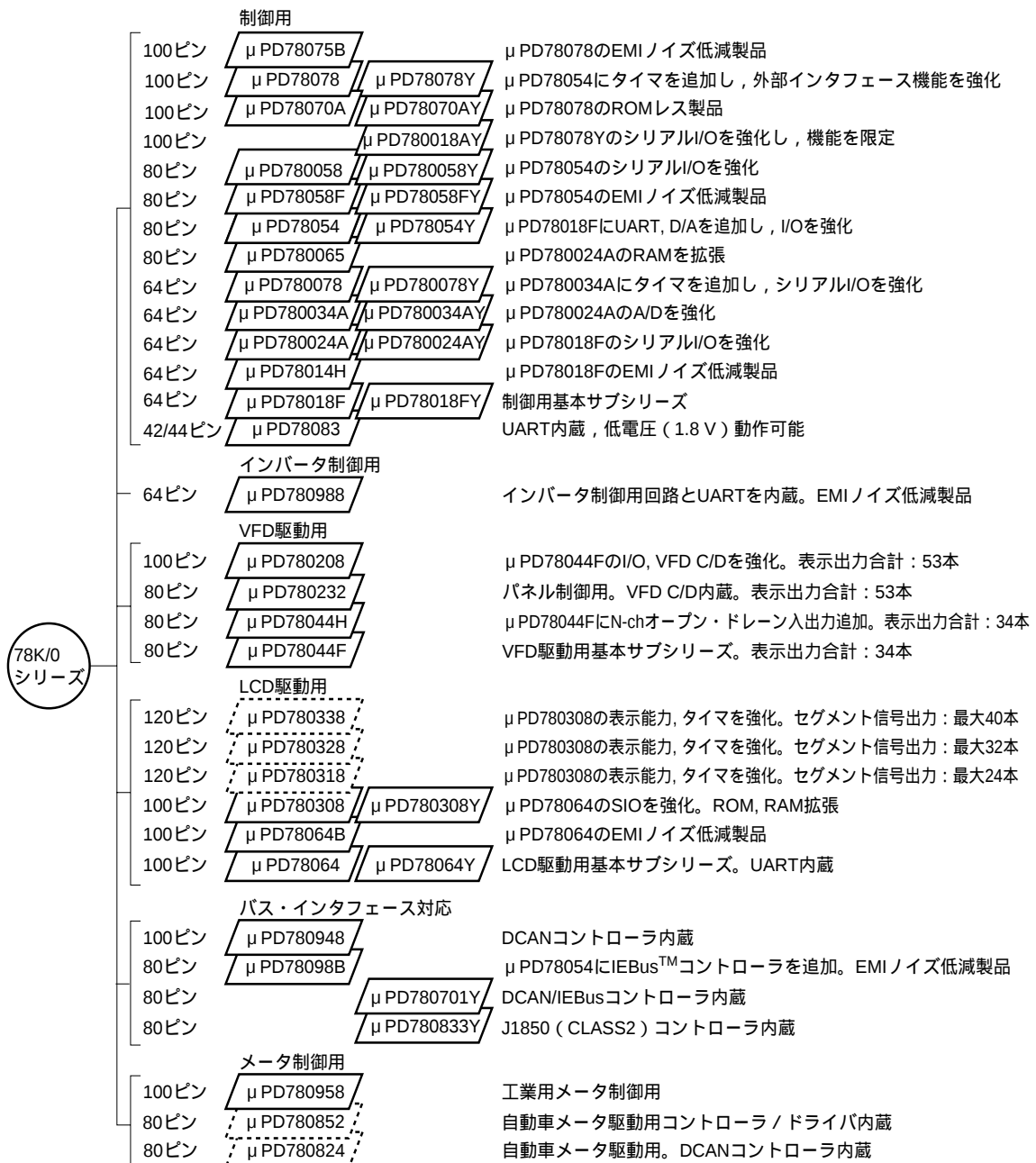
第1章 概 説

★ 1.1 78K/0シリーズの展開

78K/0シリーズの製品展開を次に示します。枠内はサブシリーズ名称です。



Yサブシリーズは、I²Cバス対応の製品です。



備考 蛍光表示管の一般的な英語名称はVFD (Vacuum Fluorescent Display) ですが、ドキュメントによっては FIP[®] (Fluorescent Indicator Panel) と記述しているものがあります。VFDとFIPは同等の機能です。

各サブシリーズ間の主な機能の違いを次に示します。

・Yなしサブシリーズ

機能 サブシリーズ名		ROM容量	タイマ				8-bit	10-bit	8-bit	シリアル・ インタフェース	I/O	V _{DD} MIN.値	外部 拡張			
			8-bit	16-bit	時計	WDT	A/D	A/D	D/A							
制御用	μ PD78075B	32 K-40 K	4ch	1ch	1ch	1ch	8ch	-	2ch	3ch(UART:1ch)	88本	1.8 V	○			
	μ PD78078	48 K-60 K														
	μ PD78070A	-									61本	2.7 V				
	μ PD780058	24 K-60 K	2ch							3ch(時分割UART:1ch)	68本	1.8 V				
	μ PD78058F	48 K-60 K								3ch(UART:1ch)	69本	2.7 V				
	μ PD78054	16 K-60 K										2.0 V				
	μ PD780065	40 K-48 K								-	4ch(UART:1ch)	60本		2.7 V		
	μ PD780078	48 K-60K									—	8ch		3ch(UART:2ch)	52本	1.8 V
	μ PD780034A	8 K-32 K												3ch(UART:1ch)	51本	
	μ PD780024A									8ch	-					
	μ PD78014H											2ch		53本		
	μ PD78018F	8 K-60 K														
μ PD78083	8 K-16 K									-	—		1ch(UART:1ch)	33本		-
インバータ 制御用	μ PD780988	16 K-60 K	3ch	注	—	1ch	—	8ch	-	3ch(UART:2ch)	47本	4.0 V	○			
VFD 駆動用	μ PD780208	32 K-60 K	2ch	1ch	1ch	1ch	8ch	-	-	2ch	74本	2.7 V	-			
	μ PD780232	16 K-24 K	3ch	—	—		4ch			40本	4.5 V					
	μ PD78044H	32 K-48 K	2ch	1ch	1ch		8ch			1ch	68本	2.7 V				
	μ PD78044F	16 K-40 K					2ch									
LCD 駆動用	μ PD780338	48 K-60 K	3ch	2ch	1ch	1ch	-	10ch	1ch	2ch(UART:1ch)	54本	1.8 V	-			
	μ PD780328											62本				
	μ PD780318											70本				
	μ PD780308	48 K-60 K	2ch	1ch			8ch	-	-	3ch(時分割UART:1ch)	57本	2.0 V				
	μ PD78064B	32 K								2ch(UART:1ch)						
	μ PD78064	16 K-32 K														
バス・イン タフェース 対応	μ PD780948	60 K	2ch	2ch	1ch	1ch	8ch	-	-	3ch(UART:1ch)	79本	4.0 V	○			
	μ PD78098B	40 K-60 K		1ch		2ch			69本	2.7 V	-					
メータ 制御用	μ PD780958	48 K-60 K	4ch	2ch	-	1ch	-	-	-	2ch(UART:1ch)	69本	2.2 V	-			
ダッシュ ボード制御用	μ PD780852	32 K-40 K	3ch	1ch	1ch	1ch	5ch	-	-	3ch(UART:1ch)	56本	4.0 V	-			
	μ PD780824	32 K-60 K								2ch(UART:1ch)	59本					

注 16ビット・タイマ：2チャンネル

10ビット・タイマ：1チャンネル

・ Yサブシリーズ

機能 サブシリーズ名		ROM容量	タイマ				8-bit	10-bit	8-bit	シリアル・ インタフェース	I/O	V _{DD} MIN.値	外部 拡張
			8-bit	16-bit	時計	WDT	A/D	A/D	D/A				
制御用	μ PD78078Y	48 K-60 K	4ch	1ch	1ch	1ch	8ch	-	2ch	3ch(UART:1ch, I ² C:1ch)	88本	1.8 V	○
	μ PD78070AY	-									61本	2.7 V	
	μ PD780018AY	48 K-60 K								-	3ch(I ² C:1ch)	88本	
	μ PD780058Y	24 K-60 K	2ch	1ch	1ch	1ch	8ch	-	2ch	3ch(時分割UART: 1ch, I ² C:1ch)	68本	1.8 V	
	μ PD78058FY	48 K-60 K								3ch(UART:1ch, I ² C:1ch)	69本	2.7 V	
	μ PD78054Y	16 K-60 K										2.0 V	
	μ PD780078Y	48 K-60 K								4ch(UART:2ch, I ² C:1ch)	52本	1.8 V	
	μ PD780034AY	8 K-32 K								3ch(UART:1ch, I ² C:1ch)	51本		
	μ PD780024AY												
	μ PD78018FY	8 K-60 K								2ch(I ² C:1ch)	53本		
LCD 駆動用	μ PD780308Y	48 K-60 K	2ch	1ch	1ch	1ch	8ch	-	-	3ch(時分割UART: 1ch, I ² C:1ch)	57本	2.0 V	-
	μ PD78064Y	16 K-32 K								2ch(UART:1ch, I ² C:1ch)			
バス・イン タフェース 対応	μ PD780701Y	60 K	3ch	2ch	1ch	1ch	16ch	-	-	4ch(UART:1ch, I ² C:1ch)	67本	3.5 V	-
	μ PD780833Y										65本	4.5 V	

備考 YなしサブシリーズとYサブシリーズは、シリアル・インタフェース以外の機能は共通です（Yなしサブシリーズがある場合）。

1.2 78K/0シリーズの特徴

78K/0シリーズは、民生分野に最適な8ビット・シングルチップ・マイクロコンピュータです。

μPD78018F, 78018FYサブシリーズは、高速、高性能CPUを搭載し、ROM、RAM、I/Oポート、タイマ、A/Dコンバータ、シリアル・インタフェース、割り込み制御などの周辺ハードウェアを内蔵した製品です。

★ μPD780024A, 780024AYサブシリーズは、μPD78018F, 78018FYサブシリーズのシリアルI/Oを強化した製品です。EMIノイズ低減製品です。

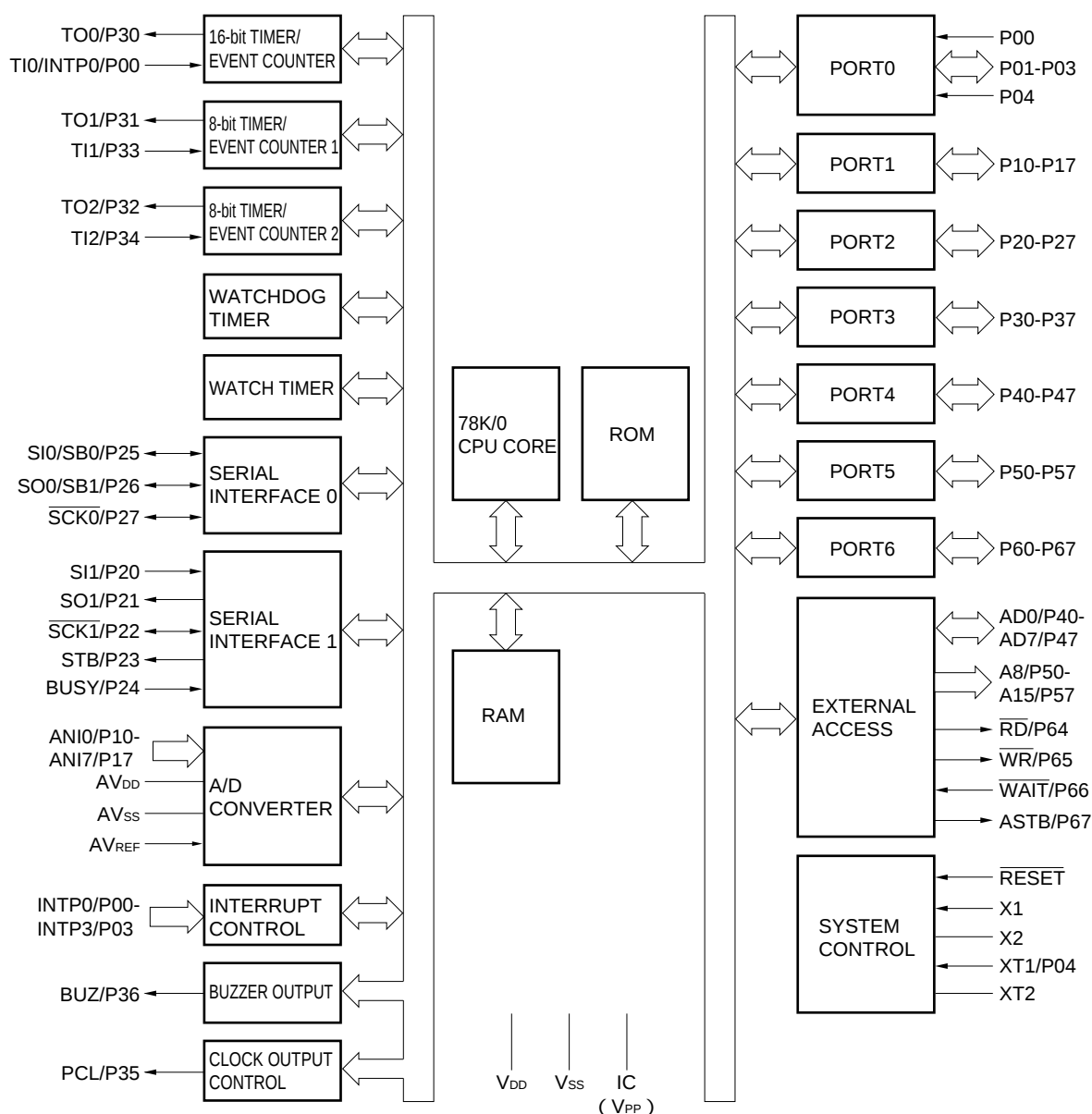
★ μPD780034A, 780034AYサブシリーズはμPD780024A, 780024AYサブシリーズのA/Dコンバータを強化した製品です。

μPD78014HサブシリーズはμPD78018FサブシリーズのEMIノイズ低減製品です。

また、μPD780024AY, 780034AYサブシリーズは、それぞれμPD780024A, 780034AサブシリーズにI²Cバス制御機能を追加しています。μPD78018FYサブシリーズは、μPD78018FサブシリーズのSBI機能に代わり、I²Cバス制御機能を追加しています。

また、マスクROM製品と同じ電源電圧で動作可能なワン・タイムPROMまたはフラッシュ・メモリ製品μPD78P018F, 78P018FY ($V_{DD} = 1.8 \sim 5.5V$)、μPD78F0034A, 78F0034AY ($V_{DD} = 1.8 \sim 5.5V$) も用意しており、応用システムの早期立ち上げ、少量生産時に最適です。

次ページ以降に各サブシリーズごとのブロック図と機能概要を示します。

図1 - 1 μ PD78018Fサブシリーズのブロック図

備考1．内部ROM，RAM容量は製品によって異なります。

2．()内は μ PD78P018Fのとき。

表1 - 1 μ PD78018Fサブシリーズの機能概要 (1/2)

品 名		μ PD78011F	μ PD78012F	μ PD78013F	μ PD78014F	μ PD78015F	μ PD78016F	μ PD78018F	μ PD78P018F
項 目	内部メモリ	ROM							PROM
		マスクROM							
		8 Kバイト	16 Kバイト	24 Kバイト	32 Kバイト	40 Kバイト	48 Kバイト	60 Kバイト	60 Kバイト ^{注1}
	高速RAM	512バイト							1024バイト ^{注1}
	拡張RAM	-							1024バイト ^{注2}
	バッファRAM	32バイト							
メモリ空間		64 Kバイト							
汎用レジスタ		8 ビット $\times$ 8 $\times$ 4 バンク							
最小命令実行時間	メイン・システム・クロック選択時	0.4 μ s / 0.8 μ s / 1.6 μ s / 3.2 μ s / 6.4 μ s (10.0 MHz動作時)							
	サブシステム・クロック選択時	122 μ s (32.768 kHz動作時)							
命令セット		・ 16ビット演算 ・ 乗除算 (8 ビット $\times$ 8 ビット , 16ビット $\div$ 8 ビット) ・ ビット操作 (セット , リセット , テスト , ブール演算) ・ BCD補正など							
I/Oポート		・ 合計 : 53本 ・ CMOS入力 : 2 本 ・ CMOS入出力 : 47本 (ソフトウェアで内蔵プルアップ抵抗のオン , オフ可能 : 47本) ・ N-chオープン・ドレイン入出力 : 4 本 (15 V耐圧 , マスクROM製品のみマスク・オプションによるプルアップ抵抗内蔵 : 4 本)							
A/Dコンバータ		・ 8 ビット分解能 $\times$ 8 チャネル ・ 低電圧動作可能 : $AV_{DD} = 1.8 \sim 5.5$ V							
シリアル・インタフェース		・ 3 線式シリアルI/O / SBI / 2 線式シリアルI/Oモード選択可能 : 1 チャネル ・ 3 線式シリアルI/Oモード (最大32バイト自動送受信機能内蔵) : 1 チャネル							
タイマ		・ 16ビット・タイマ/イベント・カウンタ : 1 チャネル ・ 8 ビット・タイマ/イベント・カウンタ : 2 チャネル ・ 時計用タイマ : 1 チャネル ・ ウォッチドッグ・タイマ : 1 チャネル							
タイマ出力		3 本 (14ビットPWM出力可能 : 1 本)							
クロック出力		39.1 kHz , 78.1 kHz , 156 kHz , 313 kHz , 625 kHz , 1.25 MHz (メイン・システム・クロック : 10.0 MHz動作時) 32.768 kHz (サブシステム・クロック : 32.768 kHz動作時)							
ブザー出力		2.4 kHz , 4.9 kHz , 9.8 kHz (メイン・システム・クロック : 10.0 MHz動作時)							

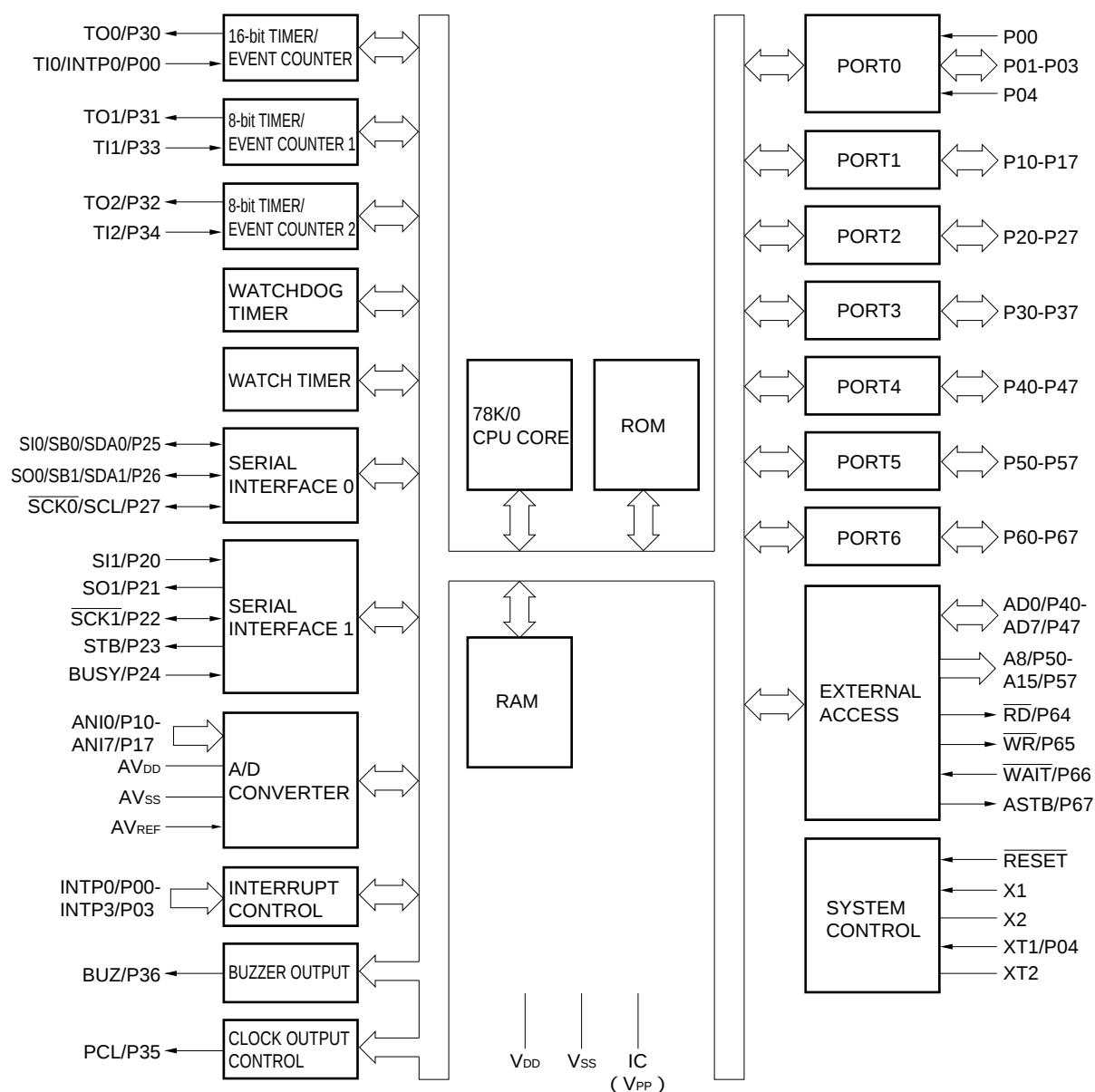
注1 . メモリ・サイズ切り替えレジスタ (IMS) により , 内部PROM , 内部高速RAM容量の変更可能。

2 . 内部拡張RAMサイズ切り替えレジスタ (IXS) により , 内部拡張RAM容量の変更可能。

表1 - 1 μ PD78018Fサブシリーズの機能概要 (2/2)

項 目		品 名	μ PD78011F	μ PD78012F	μ PD78013F	μ PD78014F	μ PD78015F	μ PD78016F	μ PD78018F	μ PD78P018F
ベクタ	マスカブル	内部：8，外部：4								
割り込み	ノンマスカブル	内部：1								
要因	ソフトウェア	1								
テスト入力		内部：1本，外部：1本								
電源電圧		$V_{DD} = 1.8 \sim 5.5 \text{ V}$								
動作周囲温度		$T_A = -40 \sim +85$								
★	パッケージ		・64ピン・プラスチックSDIP (19.05 mm (750)) ・64ピン・プラスチックQFP (14 x 14) ・64ピン・プラスチックLQFP (12 x 12)							

図1 - 2 μPD78018FYサブシリーズのブロック図



備考1．内部ROM，RAM容量は製品によって異なります。

2．()内はμPD78P018FYのとき。

表 1 - 2 μPD78018FYサブシリーズの機能概要 (1/2)

品 名		μPD78011FY	μPD78012FY	μPD78013FY	μPD78014FY	μPD78015FY	μPD78016FY	μPD78018FY	μPD78P018FY	
項 目										
内部メモリ	ROM	マスクROM								PROM
		8 Kバイト	16 Kバイト	24 Kバイト	32 Kバイト	40 Kバイト	48 Kバイト	60 Kバイト	60 Kバイト ^{注1}	
	高速RAM	512バイト		1024バイト					1024バイト ^{注1}	
	拡張RAM	-				512バイト		1024バイト	1024バイト ^{注2}	
	バッファRAM	32バイト								
メモリ空間		64 Kバイト								
汎用レジスタ		8 ビット× 8 × 4 バンク								
最小命令実行 時間	メイン・システム・ クロック選択時	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs (10.0 MHz動作時)								
	サブシステム・ クロック選択時	122 μs (32.768 kHz動作時)								
命令セット		・ 16ビット演算 ・ 乗除算 (8 ビット× 8 ビット , 16ビット÷ 8 ビット) ・ ビット操作 (セット , リセット , テスト , ブール演算) ・ BCD補正など								
I/Oポート		・ 合計 : 53本 ・ CMOS入力 : 2 本 ・ CMOS入出力 : 47本 (ソフトウエアで内蔵プルアップ抵抗のオン , オフ可能 : 47本) ・ N-chオープン・ドレイン入出力 : 4 本 (15 V耐圧 , マスクROM製品のみマスク・オプションによるプルアップ抵抗内蔵 : 4 本)								
A/Dコンバータ		・ 8 ビット分解能× 8 チャネル ・ 低電圧動作可能 : AV _{DD} = 1.8 ~ 5.5 V								
シリアル・インタフェース		・ 3 線式シリアルI/O / 2 線式シリアルI/O / I ² Cバス・モード選択可能 : 1 チャネル ・ 3 線式シリアルI/Oモード (最大32バイト自動送受信機能内蔵) : 1 チャネル								
タイマ		・ 16ビット・タイマ / イベント・カウンタ : 1 チャネル ・ 8 ビット・タイマ / イベント・カウンタ : 2 チャネル ・ 時計用タイマ : 1 チャネル ・ ウォッチドッグ・タイマ : 1 チャネル								
タイマ出力		3 本 (14ビットPWM出力可能 : 1 本)								
クロック出力		39.1 kHz , 78.1 kHz , 156 kHz , 313 kHz , 625 kHz , 1.25 MHz (メイン・システム・クロック : 10.0 MHz動作時) 32.768 kHz (サブシステム・クロック : 32.768 kHz動作時)								
ブザー出力		2.4 kHz , 4.9 kHz , 9.8 kHz (メイン・システム・クロック : 10.0 MHz動作時)								

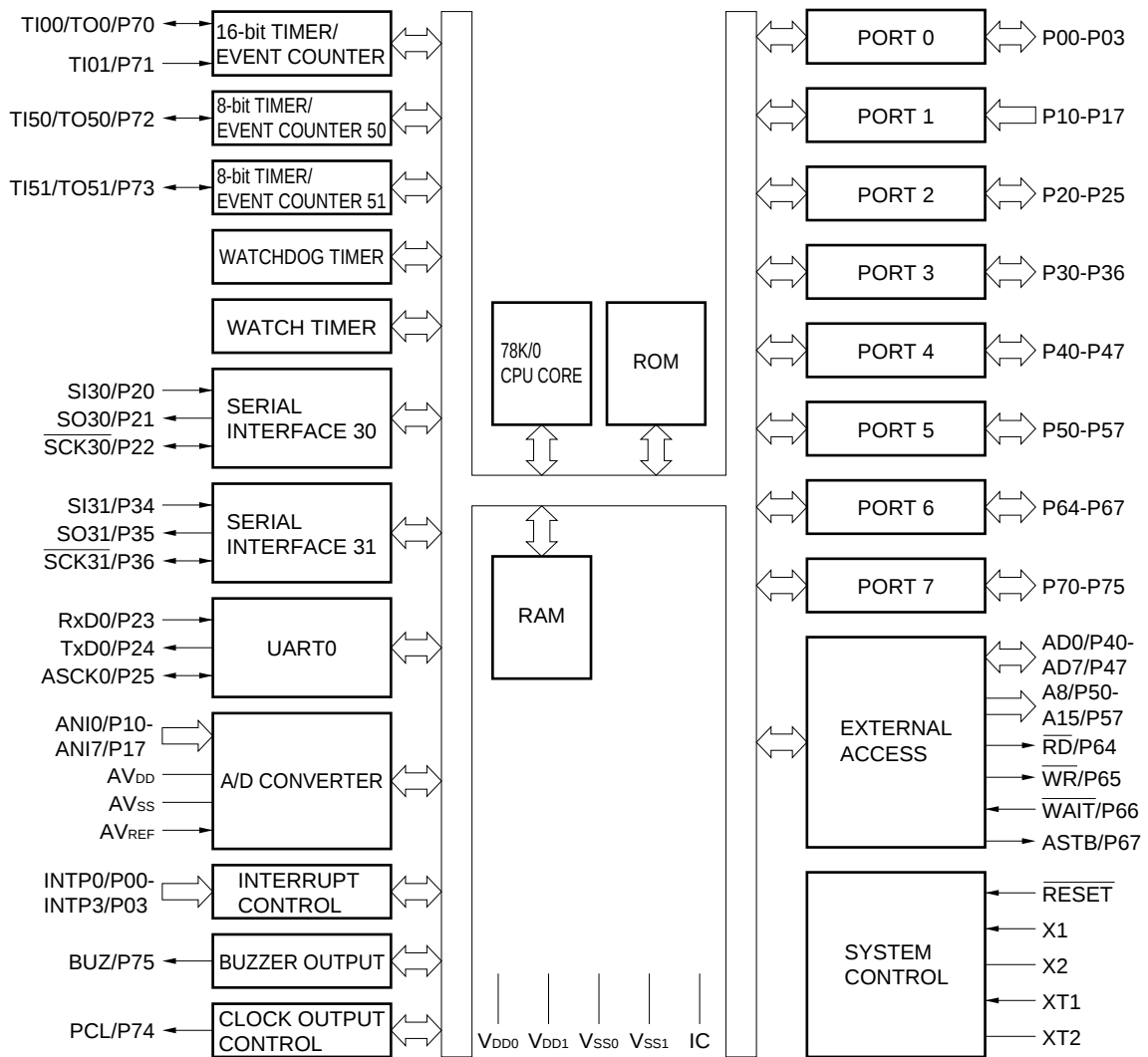
注 1 . メモリ・サイズ切り替えレジスタ (IMS) により , 内部PROM , 内部高速RAM容量の変更可能。

2 . 内部拡張RAMサイズ切り替えレジスタ (IXS) により , 内部拡張RAM容量の変更可能。

表 1 - 2 μ PD78018FYサブシリーズの機能概要 (2/2)

項 目		品 名	μ PD78011FY	μ PD78012FY	μ PD78013FY	μ PD78014FY	μ PD78015FY	μ PD78016FY	μ PD78018FY	μ PD78P018FY
ベクタ	マスカブル	内部：8，外部：4								
割り込み	ノンマスカブル	内部：1								
要因	ソフトウェア	1								
テスト入力		内部：1本，外部：1本								
電源電圧		$V_{DD} = 1.8 \sim 5.5 \text{ V}$								
動作周囲温度		$T_A = -40 \sim +85$								
★	パッケージ		・ 64ピン・プラスチックSDIP (19.05 mm (750)) ・ 64ピン・プラスチックQFP (14 x 14)							

図1 - 3 μ PD780024Aサブシリーズのブロック図

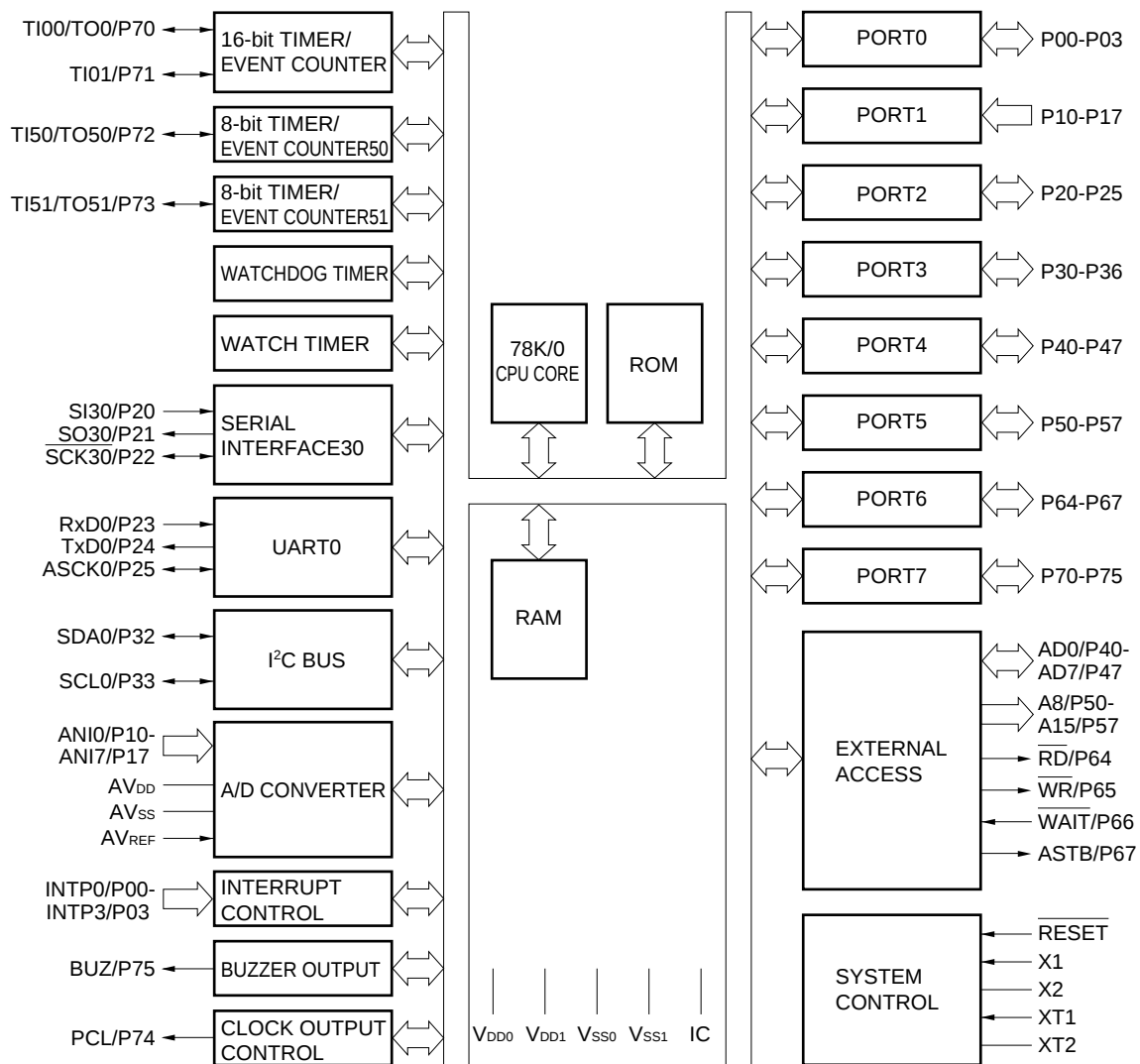


備考 内部ROM, RAM容量は製品によって異なります。

表 1 - 3 μ PD780024Aサブシリーズの機能概要

品 名		μ PD780021A	μ PD780022A	μ PD780023A	μ PD780024A
項 目					
内部メモリ	ROM	マスクROM			
		8 Kバイト	16 Kバイト	24 Kバイト	32 Kバイト
	高速RAM	512バイト		1024バイト	
メモリ空間		64 Kバイト			
汎用レジスタ		8 ビット × 8 × 4 バンク			
最小命令	メイン・システム・クロック選択時	0.24 μ s/0.48 μ s/0.95 μ s/1.91 μ s/3.81 μ s (8.38 MHz動作時)			
実行時間	サブシステム・クロック選択時	122 μ s (32.768 kHz動作時)			
命令セット		・ 16ビット演算 ・ 乗除算 (8 ビット × 8 ビット , 16ビット ÷ 8 ビット) ・ ビット操作 (セット , リセット , テスト , ブール演算) ・ BCD補正など			
I/Oポート		合計 : 51本 ・ CMOS入力 : 8 本 ・ CMOS入出力 : 39本 (ソフトウェアで内蔵プルアップ抵抗のオン , オフ可能 : 39本) ・ N-chオープン・ドレーン入出力 : 4 本 (5 V耐圧 , マスク・オプションによるプルアップ抵抗内蔵 : 4 本)			
A/Dコンバータ		・ 8 ビット分解能 × 8 チャンネル ・ 低電圧動作可能 : AV _{DD} = 1.8 ~ 5.5 V			
シリアル・インタフェース		・ 3 線式シリアルI/Oモード : 2 チャンネル ・ UARTモード : 1 チャンネル			
タイマ		・ 16ビット・タイマ / イベント・カウンタ : 1 チャンネル ・ 8 ビット・タイマ / イベント・カウンタ : 2 チャンネル ・ 時計用タイマ : 1 チャンネル ・ ウォッチドッグ・タイマ : 1 チャンネル			
タイマ出力		3 本 (8 ビットPWM出力可能 : 2 本)			
クロック出力		・ 65.5 kHz, 131 kHz, 262 kHz, 524 kHz, 1.05 MHz, 2.10 MHz, 4.19 MHz, 8.38 MHz (メイン・システム・クロック : 8.38 MHz動作時) ・ 32.768 kHz (サブシステム・クロック : 32.768 kHz動作時)			
ブザー出力		1.02 kHz, 2.05 kHz, 4.10 kHz, 8.19 kHz (メイン・システム・クロック : 8.38 MHz動作時)			
ベクタ	マスカブル	内部 : 13 , 外部 : 5			
割り込み	ノンマスカブル	内部 : 1			
要因	ソフトウェア	1			
電源電圧		V _{DD} = 1.8 ~ 5.5 V			
動作周囲温度		T _A = - 40 ~ + 85			
パッケージ		・ 64ピン・プラスチックSDIP (19.05 mm (750)) ・ 64ピン・プラスチックQFP (14 x 14) ・ 64ピン・プラスチックTQFP (12 x 12)			

図1 - 4 μ PD780024AYサブシリーズのブロック図



備考 内部ROM, RAM容量は製品によって異なります。

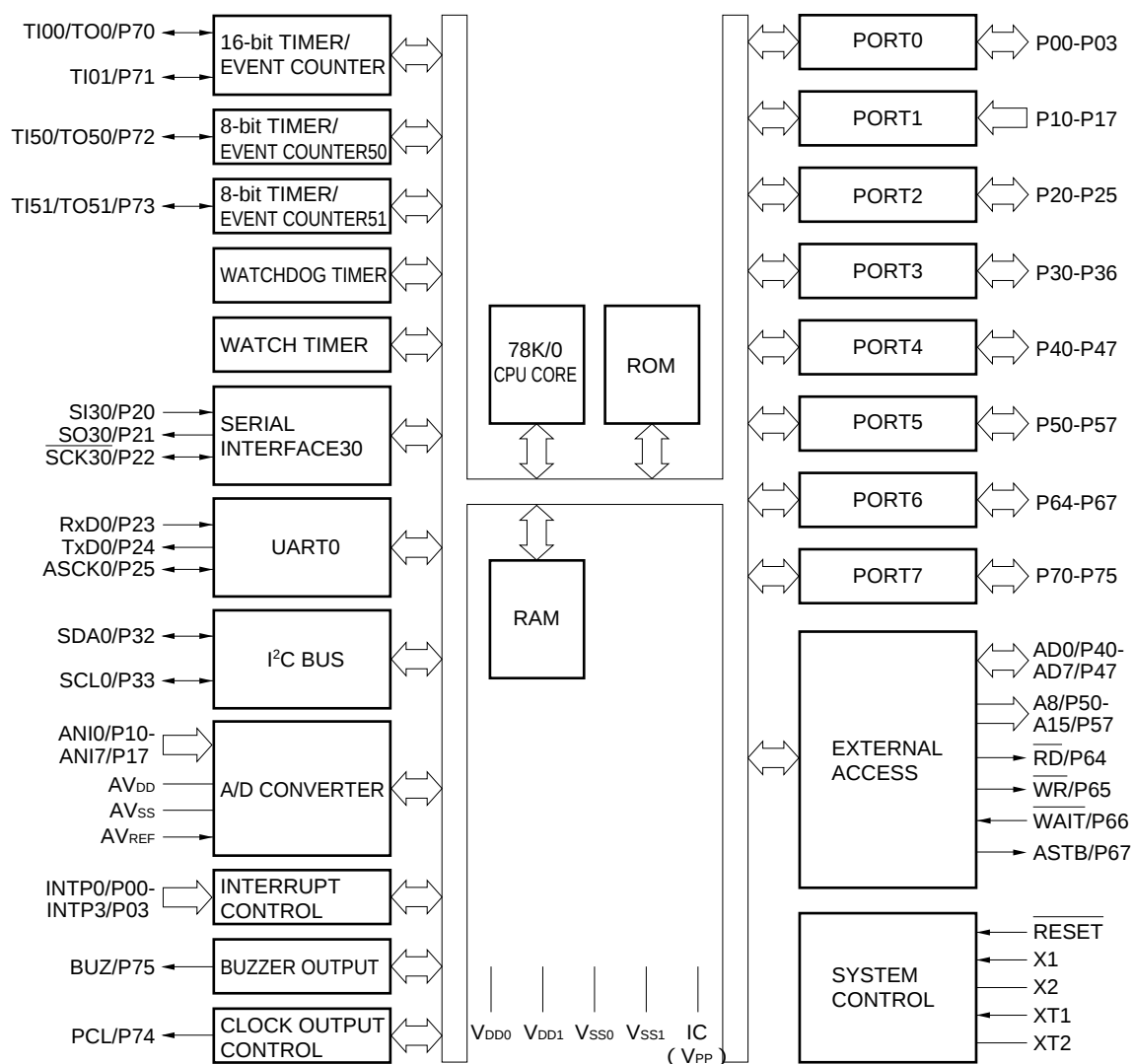
表1 - 4 μ PD780024AYサブシリーズの機能概要

品 名		μ PD780021AY	μ PD780022AY	μ PD780023AY	μ PD780024AY
項 目					
内部メモリ	ROM	マスクROM			
		8 Kバイト	16 Kバイト	24 Kバイト	32 Kバイト
	高速RAM	512バイト		1024バイト	
メモリ空間		64 Kバイト			
汎用レジスタ		8 ビット × 8 × 4 バンク			
最小命令	メイン・システム・クロック選択時	0.24 μs/0.48 μs/0.95 μs/1.91 μs/3.81 μs (8.38 MHz動作時)			
実行時間	サブシステム・クロック選択時	122 μs (32.768 kHz動作時)			
命令セット		・ 16ビット演算 ・ 乗除算 (8 ビット × 8 ビット , 16ビット ÷ 8 ビット) ・ ビット操作 (セット , リセット , テスト , ブール演算) ・ BCD補正など			
I/Oポート		合計 : 51本 ・ CMOS入力 : 8 本 ・ CMOS入出力 : 39本 (ソフトウェアで内蔵プルアップ抵抗のオン , オフ可能 : 39本) ・ N-chオープン・ドレイン入出力 : 4 本 (5 V耐圧 , マスク・オプションによるプルアップ抵抗内蔵 : 4 本)			
A/Dコンバータ		・ 8 ビット分解能 × 8 チャンネル ・ 低電圧動作可能 : AV _{DD} = 1.8 ~ 5.5 V			
シリアル・インタフェース		・ 3 線式シリアルI/Oモード : 1 チャンネル ・ UARTモード : 1 チャンネル ・ I ² Cバス・モード : 1 チャンネル			
タイマ		・ 16ビット・タイマ / イベント・カウンタ : 1 チャンネル ・ 8 ビット・タイマ / イベント・カウンタ : 2 チャンネル ・ 時計用タイマ : 1 チャンネル ・ ウォッチドッグ・タイマ : 1 チャンネル			
タイマ出力		3 本 (8 ビットPWM出力可能 : 2 本)			
クロック出力		・ 65.5 kHz, 131 kHz, 262 kHz, 524 kHz, 1.05 MHz, 2.10 MHz, 4.19 MHz, 8.38 MHz (メイン・システム・クロック : 8.38 MHz動作時) ・ 32.768 kHz (サブシステム・クロック : 32.768 kHz動作時)			
ブザー出力		1.02 kHz, 2.05 kHz, 4.10 kHz, 8.19 kHz (メイン・システム・クロック : 8.38 MHz動作時)			
ベクタ	マスカブル	内部 : 13 , 外部 : 5			
割り込み	ノンマスカブル	内部 : 1			
要因	ソフトウェア	1			
電源電圧		V _{DD} = 1.8 ~ 5.5 V			
動作周囲温度		T _A = - 40 ~ + 85			
パッケージ		・ 64ピン・プラスチックSDIP (19.05 mm (750)) ・ 64ピン・プラスチックQFP (14 x 14) ・ 64ピン・プラスチックTQFP (12 x 12)			

表1 - 5 μ PD780034Aサブシリーズの機能概要

品 名		μ PD780031A	μ PD780032A	μ PD780033A	μ PD780034A	μ PD78F0034A
項 目						
内部メモリ	ROM	マスクROM				フラッシュ・メモリ
		8 Kバイト	16 Kバイト	24 Kバイト	32 Kバイト	32 Kバイト注
	高速RAM	512バイト		1024バイト		1024バイト注
メモリ空間		64 Kバイト				
汎用レジスタ		8 ビット × 8 × 4 バンク				
最小命令	メイン・システム・クロック選択時	0.24 μs/0.48 μs/0.95 μs/1.91 μs/3.81 μs (8.38 MHz動作時)				
実行時間	サブシステム・クロック選択時	122 μs (32.768 kHz動作時)				
命令セット		・ 16ビット演算 ・ 乗除算 (8 ビット × 8 ビット , 16ビット ÷ 8 ビット) ・ ビット操作 (セット , リセット , テスト , ブール演算) ・ BCD補正など				
I/Oポート		合計 : 51本 ・ CMOS入力 : 8 本 ・ CMOS入出力 : 39本 (ソフトウェアで内蔵プルアップ抵抗のオン , オフ可能 : 39本) ・ N-chオープン・ドレイン入出力 : 4 本 (5 V耐圧 , マスク・オプションによるプルアップ抵抗内蔵 : 4 本)				
A/Dコンバータ		・ 10ビット分解能 × 8 チャンネル ・ 低電圧動作可能 : AVDD = 1.8 ~ 5.5 V				
シリアル・インタフェース		・ 3 線式シリアルI/Oモード : 2 チャンネル ・ UARTモード : 1 チャンネル				
タイマ		・ 16ビット・タイマ / イベント・カウンタ : 1 チャンネル ・ 8 ビット・タイマ / イベント・カウンタ : 2 チャンネル ・ 時計用タイマ : 1 チャンネル ・ ウォッチドッグ・タイマ : 1 チャンネル				
タイマ出力		3 本 (8 ビットPWM出力可能 : 2 本)				
クロック出力		・ 65.5 kHz, 131 kHz, 262 kHz, 524 kHz, 1.05 MHz, 2.10 MHz, 4.19 MHz, 8.38 MHz (メイン・システム・クロック : 8.38 MHz動作時) ・ 32.768 kHz (サブシステム・クロック : 32.768 kHz動作時)				
ブザー出力		1.02 kHz, 2.05 kHz, 4.10 kHz, 8.19 kHz (メイン・システム・クロック : 8.38 MHz動作時)				
ベクタ	マスカブル	内部 : 13 , 外部 : 5				
割り込み	ノンマスカブル	内部 : 1				
要因	ソフトウェア	1				
電源電圧		VDD = 1.8 ~ 5.5 V				
動作周囲温度		TA = - 40 ~ + 85				
パッケージ		・ 64ピン・プラスチックSDIP (19.05 mm (750)) ・ 64ピン・プラスチックQFP (14 × 14) ・ 64ピン・プラスチックTQFP (12 × 12)				

注 メモリ・サイズ切り替えレジスタ (IMS) により, 内部フラッシュ・メモリ, 内部高速RAM容量の変更可能。

図1 - 6 μ PD780034AYサブシリーズのブロック図

備考1 . 内部ROM, RAM容量は製品によって異なります。

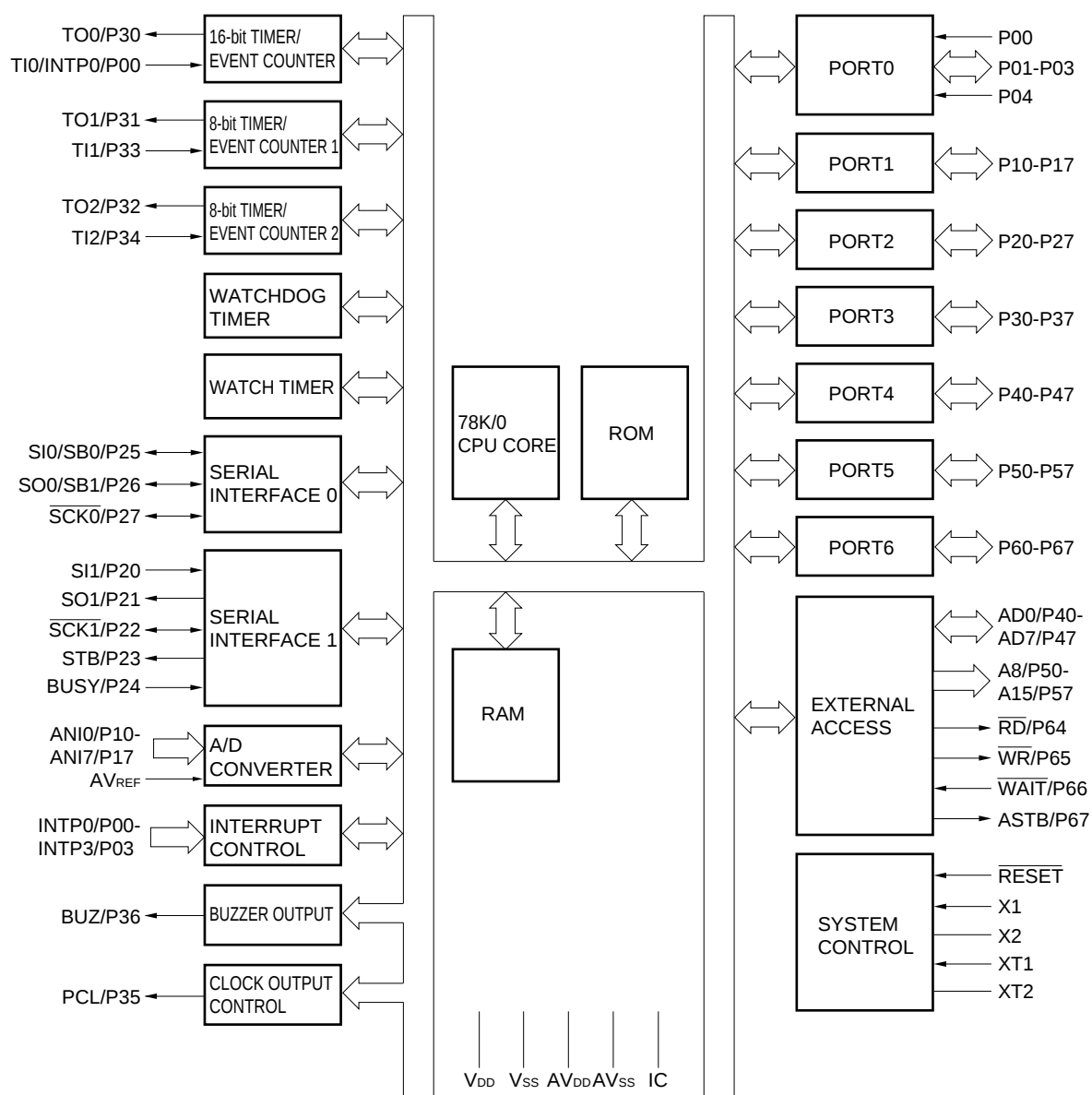
2 . () 内は μ PD78F0034AYのとき。

表1 - 6 μ PD780034AYサブシリーズの機能概要

品 名		μ PD780031AY	μ PD780032AY	μ PD780033AY	μ PD780034AY	μ PD78F0034AY
内部メモリ	ROM	マスクROM				フラッシュ・メモリ
		8 Kバイト	16 Kバイト	24 Kバイト	32 Kバイト	32 Kバイト注
	高速RAM	512バイト		1024バイト		1024バイト注
メモリ空間		64 Kバイト				
汎用レジスタ		8 ビット × 8 × 4 バンク				
最小命令	メイン・システム・クロック選択時	0.24 μs/0.48 μs/0.95 μs/1.91 μs/3.81 μs (8.38 MHz動作時)				
実行時間	サブシステム・クロック選択時	122 μs (32.768 kHz動作時)				
命令セット		・ 16ビット演算 ・ 乗除算 (8 ビット × 8 ビット , 16ビット ÷ 8 ビット) ・ ビット操作 (セット , リセット , テスト , ブール演算) ・ BCD補正など				
I/Oポート		合計 : 51本 ・ CMOS入力 : 8 本 ・ CMOS入出力 : 39本 (ソフトウエアで内蔵プルアップ抵抗のオン , オフ可能 : 39本) ・ N-chオープン・ドレイン入出力 : 4 本 (5 V耐圧 , マスク・オプションによるプルアップ抵抗内蔵 : 4 本)				
A/Dコンバータ		・ 10ビット分解能 × 8 チャンネル ・ 低電圧動作可能 : AV _{DD} = 1.8 ~ 5.5 V				
シリアル・インタフェース		・ 3 線式シリアルI/Oモード : 1 チャンネル ・ UARTモード : 1 チャンネル ・ I ² Cバス・モード : 1 チャンネル				
タイマ		・ 16ビット・タイマ / イベント・カウンタ : 1 チャンネル ・ 8 ビット・タイマ / イベント・カウンタ : 2 チャンネル ・ 時計用タイマ : 1 チャンネル ・ ウォッチドッグ・タイマ : 1 チャンネル				
タイマ出力		3 本 (8 ビットPWM出力可能 : 2 本)				
クロック出力		・ 65.5 kHz, 131 kHz, 262 kHz, 524 kHz, 1.05 MHz, 2.10 MHz, 4.19 MHz, 8.38 MHz (メイン・システム・クロック : 8.38 MHz動作時) ・ 32.768 kHz (サブシステム・クロック : 32.768 kHz動作時)				
ブザー出力		1.02 kHz, 2.05 kHz, 4.10 kHz, 8.19 kHz (メイン・システム・クロック : 8.38 MHz動作時)				
ベクタ	マスカブル	内部 : 13 , 外部 : 5				
	ノンマスカブル	内部 : 1				
要因	ソフトウェア	1				
電源電圧		V _{DD} = 1.8 ~ 5.5 V				
動作周囲温度		T _A = - 40 ~ + 85				
パッケージ		・ 64ピン・プラスチックSDIP (19.05 mm (750)) ・ 64ピン・プラスチックQFP (14 x 14) ・ 64ピン・プラスチックTQFP (12 x 12)				

注 メモリ・サイズ切り替えレジスタ (IMS) により, 内部フラッシュ・メモリ, 内部高速RAM容量の変更可能。

図1 - 7 μ PD78014Hサブシリーズのブロック図



備考 内部ROM , RAM容量は製品によって異なります。

表1 - 7 μ PD78014Hサブシリーズの機能概要 (1/2)

品 名		μ PD78011H	μ PD78012H	μ PD78013H	μ PD78014H
項 目					
内部メモリ	ROM	マスクROM			
		8 Kバイト	16 Kバイト	24 Kバイト	32 Kバイト
	高速RAM	512バイト		1024バイト	
	バッファRAM	32バイト			
メモリ空間		64 Kバイト			
汎用レジスタ		8 ビット× 8 × 4 バンク			
最小命令実行 時間	メイン・システム・ クロック選択時	0.4 μ s/0.8 μ s/1.6 μ s/3.2 μ s/6.4 μ s (10.0 MHz動作時)			
	サブシステム・ クロック選択時	122 μ s (32.768 kHz動作時)			
命令セット		・ 16ビット演算 ・ 乗除算 (8 ビット× 8 ビット , 16ビット÷ 8 ビット) ・ ビット操作 (セット , リセット , テスト , ブール演算) ・ BCD補正など			
I/Oポート		・ 合計 : 53本 ・ CMOS入力 : 2 本 ・ CMOS入出力 : 47本 (ソフトウェアで内蔵プルアップ抵抗のオン , オフ可能 : 47本) ・ N-chオープン・ドレイン入出力 : 4 本 (15 V耐圧 , マスク・オプションによるプルアップ抵抗内蔵 : 4 本)			
A/Dコンバータ		・ 8 ビット分解能× 8 チャンネル ・ 低電圧動作可能 : AV _{DD} = 1.8 ~ 5.5 V			
シリアル・インタフェース		・ 3 線式シリアルI/O / SBI / 2 線式シリアルI/Oモード選択可能 : 1 チャンネル ・ 3 線式シリアルI/Oモード (最大32バイト自動送受信機能内蔵) : 1 チャンネル			
タイマ		・ 16ビット・タイマ / イベント・カウンタ : 1 チャンネル ・ 8 ビット・タイマ / イベント・カウンタ : 2 チャンネル ・ 時計用タイマ : 1 チャンネル ・ ウォッチドッグ・タイマ : 1 チャンネル			
タイマ出力		3 本 (14ビットPWM出力可能 1 本)			
クロック出力		39.1 kHz , 78.1 kHz , 156 kHz , 313 kHz , 625 kHz , 1.25 MHz (メイン・システム・クロック : 10.0 MHz動作時) 32.768 kHz (サブシステム・クロック : 32.768 kHz動作時)			
ブザー出力		2.4 kHz , 4.9 kHz , 9.8 kHz (メイン・システム・クロック : 10.0 MHz動作時)			

表1 - 7 μ PD78014Hサブシリーズの機能概要 (2/2)

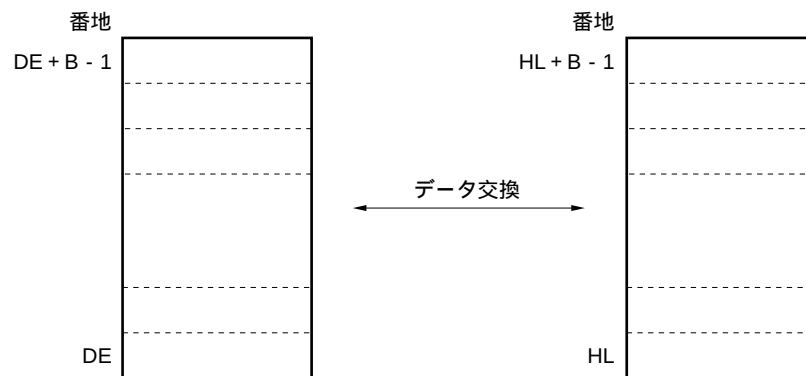
品 名		μ PD78011H	μ PD78012H	μ PD78013H	μ PD78014H
項 目	ベクタ	内部：8，外部：4			
	割り込み	内部：1			
	要因	1			
	ソフトウェア				
テスト入力		内部：1本，外部：1本			
電源電圧		$V_{DD} = 1.8 \sim 5.5 \text{ V}$			
動作周囲温度		$T_A = -40 \sim +85$			
★	パッケージ	・ 64ピン・プラスチックSDIP (19.05 mm (750))			
		・ 64ピン・プラスチックQFP (14 x 14)			
		・ 64ピン・プラスチックLQFP (12 x 12)			

第2章 ソフトウェアの基礎

2.1 データ転送

DEレジスタとHLレジスタで指定したアドレスを先頭アドレスとし、データを交換します。データを交換するバイト数は、Bレジスタで指定したバイト数になります。

図2 - 1 データ交換



(1) 使用するレジスタ

A, B, DE, HL

(2) プログラム・リスト

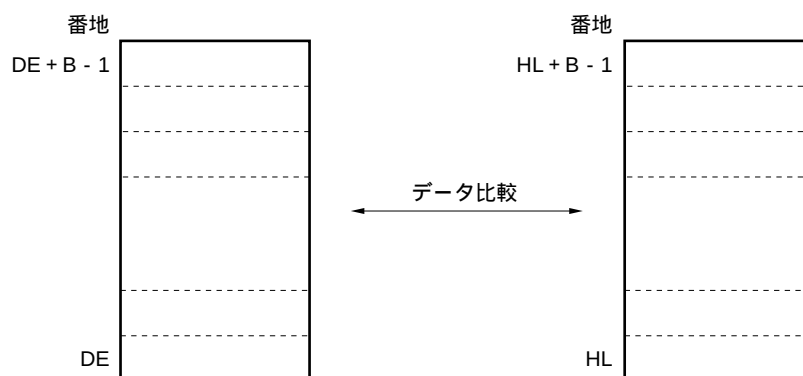
EXCH:

```
MOV    A,[DE]
XCH    A,[HL]
XCH    A,[DE]
INCW   DE
INCW   HL
DBNZ   B,$EXCH
RET
```

2.2 データ比較

DEレジスタとHLレジスタで指定したアドレスを先頭アドレスとし、データを比較します。データを比較するバイト数は、Bレジスタで指定したバイト数になります。比較結果が等しい場合はCY = 0、等しくない場合はCY = 1でリターンします。

図2 - 2 データ比較



(1) 使用するレジスタ

A, B, DE, HL

(2) プログラム・リスト

COMP:

```
MOV    A,[DE]
CMP    A,[HL]
BNZ    $ERROR
INCW   DE
INCW   HL
DBNZ   B,$COMP
CLR1   CY
BR     RTN
```

ERROR:

```
SET1   CY
```

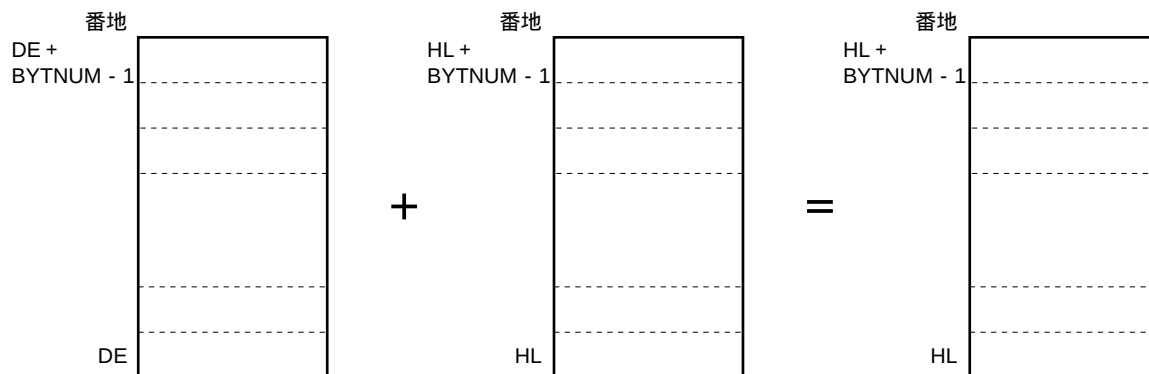
RTN:

```
RET
```

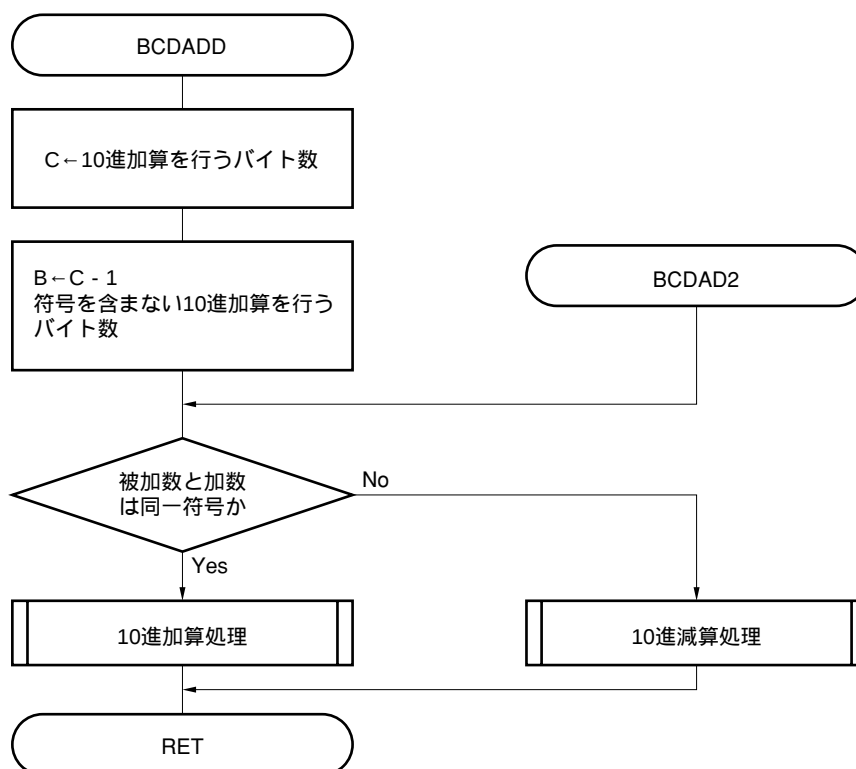
2.3 10進加算

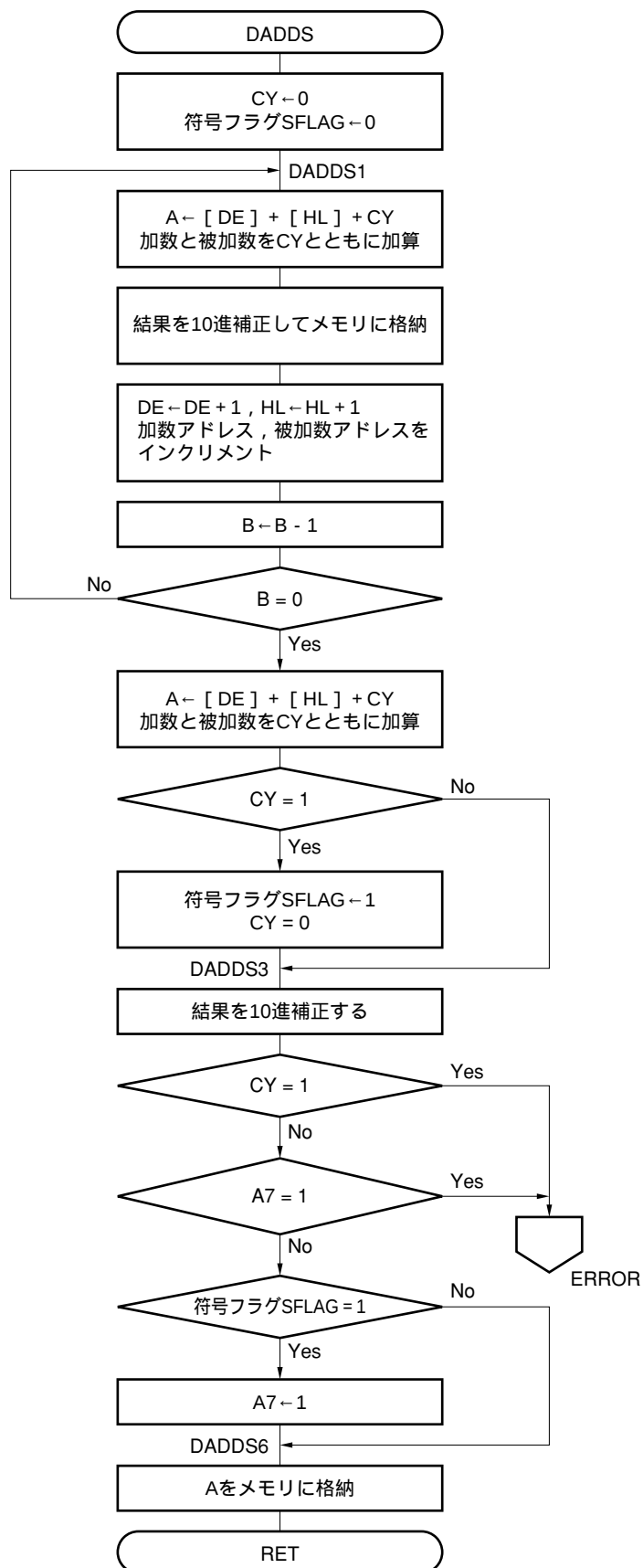
10進加算する最下位アドレスをDEレジスタとHLレジスタで指定し、BYTNUMで指定した桁数を加算します。加算結果は、HLレジスタで示される領域に格納します。また、加算した結果、オーバフロー、またはアンダフローが発生した場合は、エラー処理へ分岐します。分岐アドレスは、メインで'ERROR'と定義し、パブリック宣言してください。

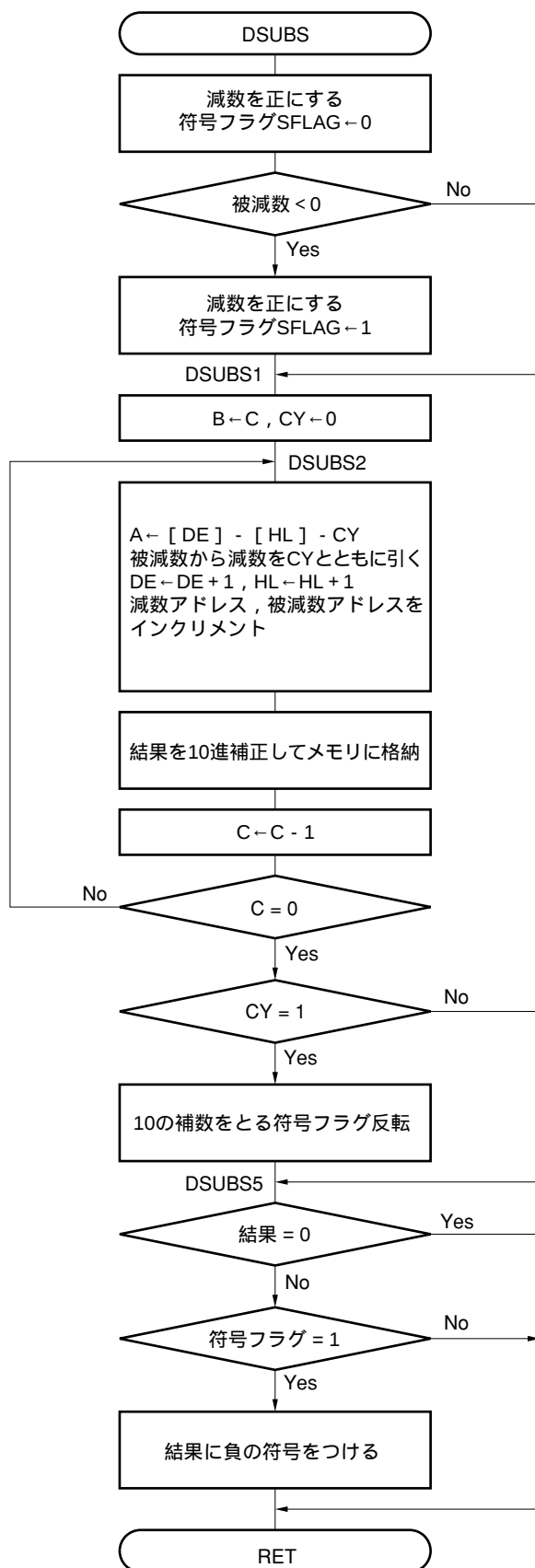
図2 - 3 10進加算



(1) フロー・チャート







(2) 使用するレジスタ

AX, BC, DE, HL

(3) プログラム・リスト

```

;*****
;
;      入力パラメータ
;      HL レジスタ:加数先頭アドレス
;      DE レジスタ:被加数先頭アドレス
;      出力パラメータ
;      HL レジスタ:演算結果先頭アドレス
;*****
PUBLIC BCDADD, BCDAD1, BCDAD2
PUBLIC DADDS
PUBLIC DSUBS
EXTRN ERROR ; エラー処理分岐アドレス
EXTBIT SFLAG ; 符号フラグ
;
BYTNUM EQU 4 ; 演算桁数の設定
;
CSEG
BCDADD:
MOV C, #BYTNUM ; 演算桁数をCレジスタに設定
BCDAD1:
MOV A, C
MOV B, A
DEC B
BCDAD2:
MOV A, [HL+BYTNUM-1] ; 被加数最上位ビット（符号データ）の取り込み
XCHW AX, DE
XCHW AX, HL
XCHW AX, DE
XOR A, [HL+BYTNUM-1] ; 被加数最上位ビット（符号データ）の取り込み
XCHW AX, HL
XCHW AX, DE
XCHW AX, HL
BT A, 7, $BCDAD3 ; 符号一致? ELSE 減算処理
CALL !DADDS ; THEN 加算処理
RET
BCDAD3:
CALL !DSUBS
RET

```

```

;=====
;          ***** 10進加算 *****
;=====

DADD1:
    CLR1    CY
    CLR1    SFLAG
DADD1:
    MOV     A, [DE]                ; 最下位桁から加算開始
    ADDC    A, [HL]
    ADJBA
    MOV     [HL], A
    INCW    HL
    INCW    DE
    DBNZ    B, $DADD1              ; (演算桁数-1) の加算終了

    MOV     A, [DE]
    ADDC    A, [HL]
DADD2:
    BNC     $DADD3                  ; 負の加算
    SET1    SFLAG                  ; THEN 負の状態に設定
    CLR1    CY
DADD3:
    ADJBA
    BNC     $DADD4
    BR      ERROR
DADD4:
    BF      A. 7, $DADD5
    BR      ERROR
DADD5:
    BF      SFLAG, $DADD6           ; 符号設定
    SET1    A. 7
DADD6:
    MOV     [HL], A
    RET

```

```

;=====
;          ***** 10進減算 *****
;=====

DSUBS:
    PUSH    HL
    CLR1    SFLAG
    MOV     A, [HL+BYTNUM-1]          ; 減数を正の値に設定
    CLR1    A. 7
    MOV     [HL+BYTNUM-1], A
    XCHW    AX, DE
    XCHW    AX, HL
    XCHW    AX, DE
    MOV     A, [HL+BYTNUM-1]
    BF      A. 7, $DSUBS1             ; 被減数が負
    CLR1    A. 7                     ; THEN 被減数を正の値に設定
    MOV     [HL+BYTNUM-1], A
    SET1    SFLAG                     ; 符号を負に設定

DSUBS1:
    XCHW    AX, HL
    XCHW    AX, DE
    XCHW    AX, HL
    MOV     A, C
    MOV     B, A
    CLR1    CY

DSUBS2:
    MOV     A, [DE]
    SUBC    A, [HL]
    ADJBS
    MOV     [HL], A
    INCW    HL
    INCW    DE
    DBNZ    C, $DSUBS2               ; 演算桁数の減算終了

    BNC     $DSUBS5                   ; THEN 減数 > 被減数
    POP     HL
    PUSH    HL
    MOV     A, B
    MOV     C, A

DSUBS3:
    MOV     A, #99H                  ; 減算結果に対して補数演算
    SUB     A, [HL]                  ;      (減算結果 - 99H)
    ADJBS
    MOV     [HL], A
    INCW    HL
    DBNZ    C, $DSUBS3

    POP     HL
    PUSH    HL
    SET1    CY

    MOV     A, B
    MOV     C, A

DSUBS4:
    MOV     A, #0                    ; 補数演算結果に1を加算
    ADDC    A, [HL]
    ADJBA

```

```

        MOV     [HL], A
        INCW    HL
        DBNZ    C, $DSUBS4
        MOV1    CY, SFLAG
        NOT1    CY
        MOV1    SFLAG, CY
;=====
;          ***** 演算結果 0 チェック *****
;=====

DSUBS5:
        MOV     A, B
        MOV     C, A
        POP     HL
        PUSH    HL
        MOV     A, #0

DSUBS6:
        CMP     A, [HL]                ; 最下位桁から 0 チェック
        INCW    HL
        BNZ     $DSUBS7
        DBNZ    C, $DSUBS6             ; 全桁 0 チェック終了
        POP     HL                    ; THEN 減算結果 = 0
        RET

DSUBS7:
        BF      SFLAG, $DSUBS8         ; 減算結果が負
        POP     HL                    ; THEN 符号設定
        PUSH    HL
        MOV     A, [HL+BYTNUM-1]
        SET1    A, 7
        MOV     [HL+BYTNUM-1], A

DSUBS8:
        POP     HL
        RET

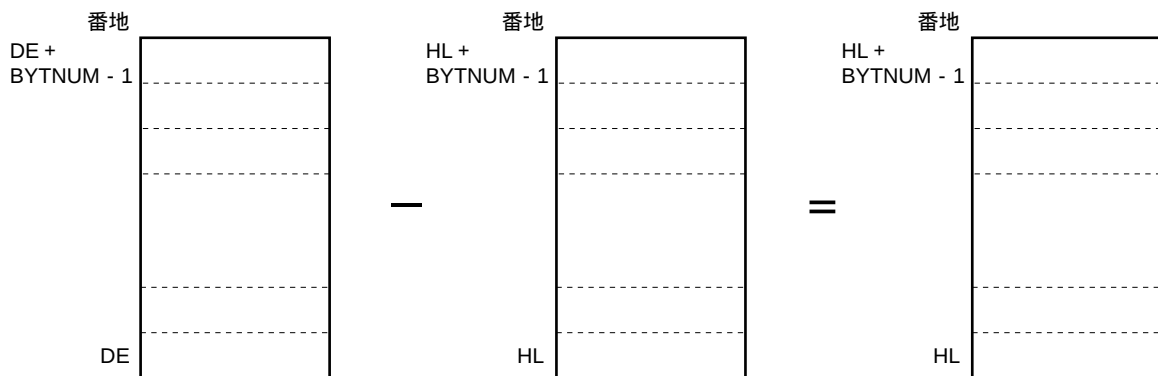
```

2.4 10進減算

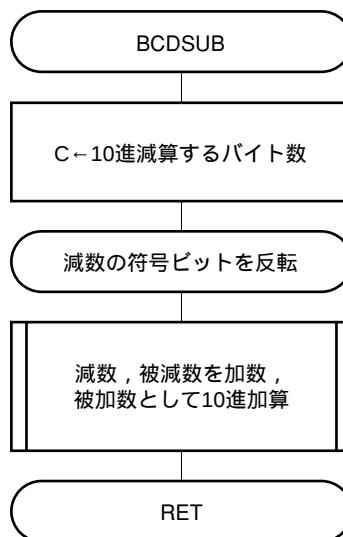
10進減算する最下位アドレスをDEレジスタとHLレジスタで指定し，BYTNUMで指定した桁数を減算します。減算結果は，HLレジスタで示される領域に格納します。また，減算した結果，オーバフロー，またはアンダフローが発生した場合は，エラー処理へ分岐します。分岐アドレスは，メインで'ERROR'と定義し，パブリック宣言してください。

このプログラムは，被減数，減数を被加数，加数に置き換え，10進加算のプログラムをコールしています。

図2 - 4 10進減算



(1) フロー・チャート



(2) 使用するレジスタ

AX , BC , DE , HL

(3) プログラム・リスト

```

;*****
;
;      入力パラメータ
;      H L レジスタ:減数先頭アドレス
;      D E レジスタ:被減数先頭アドレス
;      出力パラメータ
;      H L レジスタ:演算結果先頭アドレス
;*****
;
PUBLIC BYTNUM
PUBLIC BCDSUB
EXTRN BCDADD, BCDAD2
;
BYTNUM EQU 4 ; 演算桁数の設定
;
CSEG
BCDSUB:
MOV C, #BYTNUM ; 演算桁数をCレジスタに設定
BCDSU1:
MOV A, C
MOV B, A
DEC B

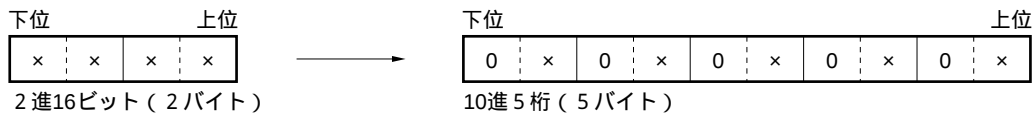
MOV A, [HL+BYTNUM-1] ; 減数の最上位ビット（符号データ）を加算用に設定
MOV1 CY, A. 7 ; 符号データを反転
NOT1 CY
MOV1 A. 7, CY
MOV [HL+BYTNUM-1], A
CALL !BCDAD2 ; 10進加算処理をコール
RET

```

2.5 2進10進変換

データ・メモリ内の2進16ビット・データを10進5桁のデータに変換し、データ・メモリ内に格納します。2進16ビットのデータを10進の10で桁数分（4回）除算し、そのときの演算結果と余りの値により変換しています。

図2 - 5 2進10進変換



例 FFHを10進変換



（1）使用レジスタ

AX, BC, HL

（2）プログラム・リスト

```

PUBLIC B_DCONV
DATDEC EQU 10

DSEG SADDRP
REGA: DS 2 ; 2進16ビット・データ格納
REGB: DS 5 ; 10進5桁データ格納

COLNUM EQU 4

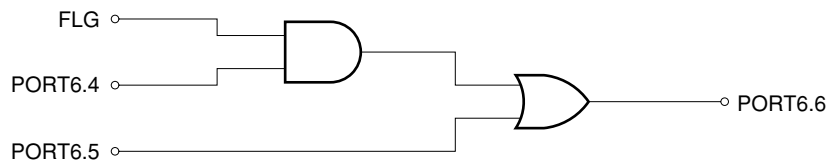
B_DCONV:
    MOVW AX, REGA
    MOV B, #COLNUM
    MOVW HL, #REGB
B_D1:
    MOV C, #DATDEC
    DIVUW C
    XCH A, C
    MOV [HL], A
    INCW HL
    XCH A, C
    DBNZ B, $B_D1
    MOV A, X
    MOV [HL], A
    RET

```

2.6 ビット演算操作命令

データ・メモリ内のフラグ1ビットとポート6のビット4の論理積をとり，その結果とポート6のビット5の論理和をポート6のビット6に出力します。

図2 - 6 ビット演算



(1) プログラム・リスト

```

PUBLIC BIT_OP,FLG

BSEG
FLG DBIT

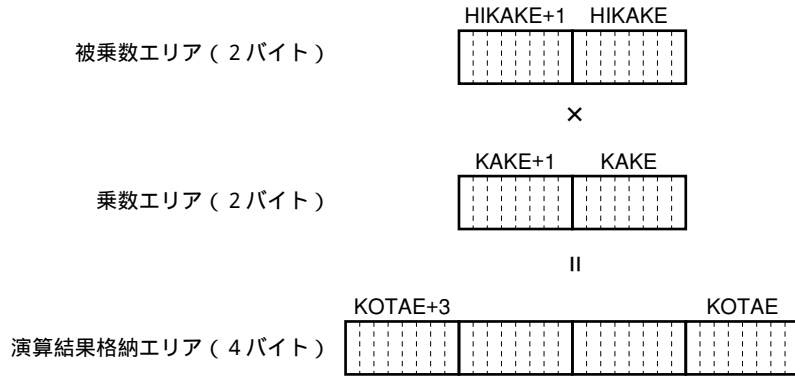
BIT_OP:
MOV1 CY,FLG
AND1 CY,P6.4
OR1 CY,P6.5
MOV1 P6.6,CY
RET

```

2.7 2進乗算 (16ビット×16ビット)

被乗数エリア (HIKAKE ; 16ビット) , 乗数エリア (KAKE ; 16ビット) のデータを乗算し , 結果を演算結果格納エリア (KOTAE) に格納します。

図2 - 7 2進乗算



<処理内容>

乗算は , 乗数のビット “ 1 ” の数だけ被乗数を加算する方法で , 実現しています。

<使用内容>

被乗数エリア (HIKAKE) , 乗数エリア (KAKE) にデータを設定し , サブルーチン S_KAKERU をコールしてください。

```

EXTRN  S_KAKERU
EXTRN  HIKAKE, KAKE, KOTAE
MAIN:                                     ; 乗数
      .
      .
      HIKAKE=WORKA (A)                   ;被乗数エリアに被乗数データ格納
      HIKAKE+1=WORKA+1 (A)               ;
      KAKE=WORKB (A)                     ;乗数エリアに乗数データ格納
      KAKE+1=WORKB+1 (A)                 ;
      CALL  !S_KAKERU                    ;乗数計算ルーチンコール
      HL=#KOTAE                          ;HL←演算結果格納エリアのRAM番地
      .                                  ;間接アドレス転送により結果を格納する
      .
      .

```

注意 データ・メモリは , 8 ビットで操作してください。

(1) 入出力条件

● 入力パラメータ

HIKAKE : 被乗数データを格納してください。

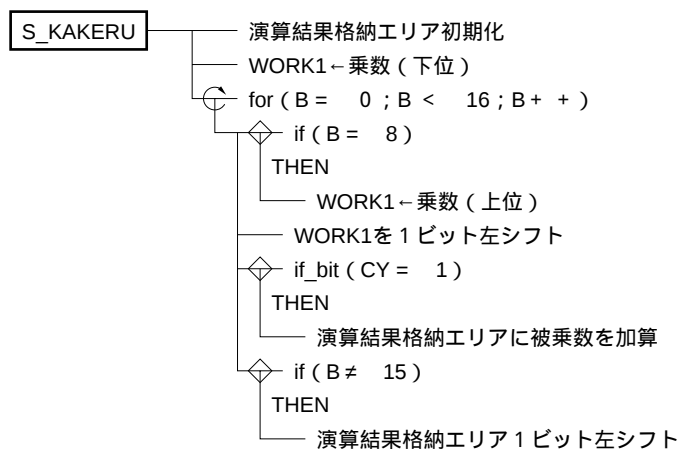
KAKE : 乗数データを格納してください。

● 出力パラメータ

KOTAE : 演算の結果を格納します。

(2) SPDチャート

[乗算サブルーチン]



(3) 使用するレジスタ

A , B

(4) プログラム・リスト

```

$PC(014)
;
PUBLIC HIKAKE, S_KAKERU, KAKE, KOTAE
;
;*****
;          R A M定義
;*****
;*****
;          DSEG      SADDR
HIKAKE:    DS        2          ;被乗数エリア
KAKE:      DS        2          ;乗数エリア
WORK1:     DS        1          ;ワークエリア
KOTAE:     DS        4          ;演算結果格納エリア
;
;*****
;          乗算
;*****
;          CSEG
S_KAKERU: ;
;          WORK1=KAKE+1 (A) ;ワークに乗数(下位)を格納
;          KOTAE=#0 ;演算結果格納エリア初期化
;          KOTAE+1=#0 ;
;          KOTAE+2=#0 ;
;          KOTAE+3=#0 ;
;          for(B=#0;B<#16;B++) (A) ;下位乗数が終了すれば
;          if(B == #8) (A) ;ワークに上位の乗数を格納
;          WORK1=KAKE (A) ;
;          endif ;
;          A=WORK1 ;乗数1ビット左シフト
;          CLR1 CY ;
;          ROLC A, 1 ;
;          WORK1=A ;
;          if_bit(CY) ;キャリーがでたら
;          KOTAE+=HIKAKE (A) ; 演算結果格納エリアに被乗数を加算する
;          (KOTAE+1)+=HIKAKE+1, CY (A) ;
;          (KOTAE+2)+=#0, CY (A) ;
;          (KOTAE+3)+=#0, CY (A) ;
;          endif ;
;          if(B != #15) (A) ;
;          KOTAE+=KOTAE (A) ;演算結果格納エリア1ビット左シフト
;          KOTAE+1+=KOTAE+1, CY (A) ;
;          KOTAE+2+=KOTAE+2, CY (A) ;
;          KOTAE+3+=KOTAE+3, CY (A) ;
;          endif ;
;          next ;
;          RET ;
;          END ;

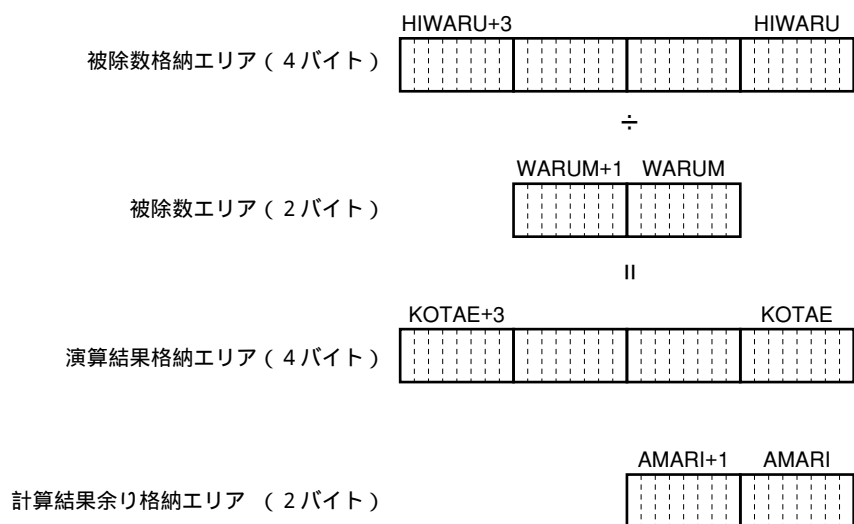
```

2.8 2進除算 (32ビット ÷ 16ビット)

被除数エリア (HIWARU ; 32ビット) , 除数エリア (WARUM ; 16ビット) で除算し , 結果を演算結果格納エリア (KOTAE) に格納します。余りが発生した場合は , 計算結果余り格納エリア (AMARI) に格納します。

除数を 0 として計算した場合は , エラーとします。

図 2 - 8 2 進除算



< 処理内容 >

被除数の上位桁からワーク・エリアに左シフトします。ワーク・エリアの内容が除数より大きければ , ワーク・エリアから除数を減算し , 被除数の最下位ビットを 1 にします。以上の方法で被除数のビット数だけ実行することにより除算が実現します。

除数を 0 とした場合は , エラー・フラグ (F_ERR) をセットします。

<使用方法>

被除数エリア（HIWARU）、除数エリア（WARUM）に、データを設定し、サブルーチンS_WARUをコールしてください。

```

        EXTRN    S_WARU
        EXTRN    HIWARU, WARUM, KOTAE
        EXBIT    F_ERR

MAIN:
        .        ;
        .        ;
        HIWARU=WORKA (A)      ;被除数エリアに被乗数データ格納
        HIWARU+1=WORKA+1 (A)  ;
        WARUM=WORKB (A)      ;除数エリアに乗数データ格納
        WARUM+1=WORKB+1 (A)  ;
        CALL     !S_WARU      ;除数計算ルーチンコール
        HL=#KOTAE            ;HL←演算結果格納エリアのRAM番地格納
        .        ;
        .        ;
        if_bit(F_ERR)        ;
        計算エラー処理      ;
        endif                ;
        .
        .
        .

```

注意 データ・メモリは、8ビットで操作してください。

(1) 入出力条件

● 入力パラメータ

HIWARU : 被除数データを格納してください。

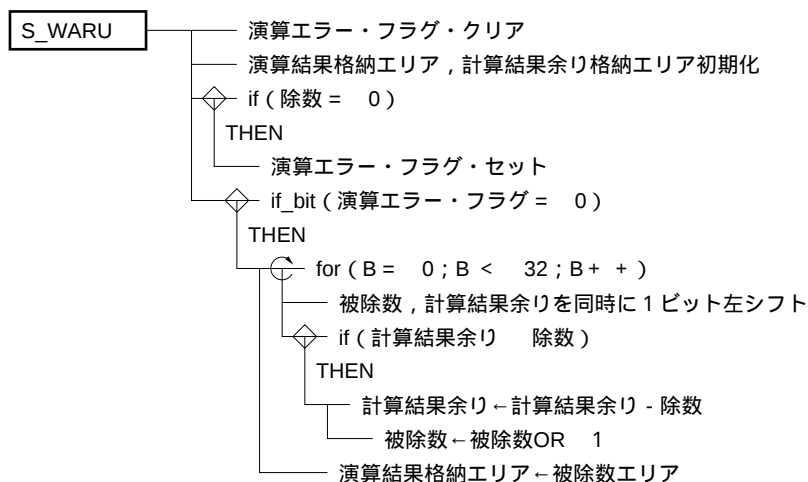
WARUM : 除数データを格納してください。

● 出力パラメータ

KOTAE : 計算の結果を格納します。

(2) SPDチャート

[除算サブルーチン]



(3) 使用するレジスタ

A, B

(4) プログラム・リスト

```

$PC(014)
;
PUBLIC S_WARU, HIWARU, WARUM, F_ERR
EXTRN KOTAE
;
;*****
;          R A M定義
;*****
                DSEG      SADDR
HIWARU:         DS         4                ;被除数エリア
WARUM:          DS         2                ;除数エリア
AMARI:          DS         2                ;計算結果余り格納エリア
                BSEG
F_ERR           DBIT                ;演算エラーフラグ
;*****
;          除算
;*****
                CSEG
S_WARU:
CLR1 F_ERR                ;演算エラーフラグクリア
AMARI=#0                ;計算結果余り格納エリアを0クリア
AMARI+1=#0
KOTAE=#0                ;演算結果格納エリアを0クリア
KOTAE+1=#0
KOTAE+2=#0
KOTAE+3=#0
if(WARUM == #0)          ;除数=0?
    if(WARUM+1 == #0)
        SET1 F_ERR        ; 除数が0なら演算エラーフラグセット
    endif
endif
if_bit(!F_ERR)           ;演算エラー?
    for(B=#0;B < #32;B++) (A) ;32ビット除算開始
        HIWARU+=HIWARU (A)    ;被除数および余りを1ビット左シフト
        HIWARU+1+=HIWARU+1, CY (A)
        HIWARU+2+=HIWARU+2, CY (A)
        HIWARU+3+=HIWARU+3, CY (A)
        AMARI+=AMARI, CY (A)
        AMARI+1+=AMARI+1, CY (A)
        ;
        if(AMARI+1 > WARUM+1) (A) ;余り ≥ 除数?
            AMARI-=WARUM (A) ; 余り=余り-除数
            AMARI+1-=WARUM+1, CY (A)
            HIWARU |= #1 ;被除数エリアの1ビット目に1を格納
        elseif_bit(Z)
            if(AMARI >= WARUM) (A)
                AMARI-=WARUM(A)
                AMARI+1-=WARUM+1, CY (A)
                HIWARU |= #1
            endif
        endif
    next
    KOTAE=HIWARU (A) ;演算結果を格納する
    KOTAE+1=HIWARU+1 (A)
    KOTAE+2=HIWARU+2 (A)
    KOTAE+3=HIWARU+3 (A)
endif
RET
END

```

第3章 システム・クロック切り替えの応用

78K/0シリーズは、プロセッサ・クロック・コントロール・レジスタ（PCC）を書き換えることにより、CPUクロックの選択および発振器の動作を制御できます。

CPUクロックを変更する場合は、PCCへの書き換え命令からCPUクロックが実際に切り替わるまで、表3 - 1に示す時間がかかります。したがって、PCCへの書き換え命令後、しばらくの間は、書き換え前後のどちらのクロックで動作しているか不定になりますので、メイン・システム・クロックの停止や、STOP命令を行う場合は、表3 - 1に示す命令分のウェイトが必要になります。

表3 - 1 CPUクロックの切り替えに要する最大時間

切り替え前の設定値				切 り 替 え 後 の 設 定 値																							
CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0
				0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	1	x	x	x
0	0	0	0					16命令				16命令				16命令				16命令				fx/4fxT命令 (77命令)			
	0	0	1					8命令				8命令				8命令				8命令				fx/8fxT命令 (39命令)			
	0	1	0					4命令				4命令				4命令				4命令				fx/16fxT命令 (20命令)			
	0	1	1					2命令				2命令				2命令				2命令				fx/32fxT命令 (10命令)			
	1	0	0					1命令				1命令				1命令				1命令				fx/64fxT命令 (5命令)			
1	x	x	x	1命令				1命令				1命令				1命令				1命令							

注意 CPUクロックの分周の選択（PCC0-PCC2）とメイン・システム・クロックからサブシステム・クロックへの切り替え（CSSを0→1）を同時に設定しないでください。

ただし、CPUクロックの分周の選択（PCC0-PCC2）とサブシステム・クロックからメイン・システム・クロックへの切り替え（CSSを1→0）は同時に設定可能です。

備考1 1命令は、切り替え前のCPUクロックの最小命令実行時間となります。

2. () 内は、fx = 10.0 MHz, fxT = 32.768 kHz動作時。

図3 - 1 プロセッサ・クロック・コントロール・レジスタのフォーマット
(μ PD78018F, 78018FY, 78014Hサブシリーズ)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
PCC	MCC	FRC	CLS	CSS	0	PCC2	PCC1	PCC0	FFFBH	04H	R/W ^{注1}

R/W	CSS	PCC2	PCC1	PCC0	CPUクロック (f _{CPU}) の選択
	0	0	0	0	f _x
		0	0	1	f _x /2
		0	1	0	f _x /2 ²
		0	1	1	f _x /2 ³
		1	0	0	f _x /2 ⁴
	1	0	0	0	f _{XT}
		0	0	1	
		0	1	0	
		0	1	1	
		1	0	0	
	上記以外				設定禁止

R	CLS	CPUクロックのステータス
	0	メイン・システム・クロック
	1	サブシステム・クロック

R/W	FRC	サブシステム・クロックのフィードバック抵抗の選択
	0	内蔵フィードバック抵抗を使用する
	1	内蔵フィードバック抵抗を使用しない

R/W	MCC	メイン・システム・クロックの発振の制御 ^{注2}
	0	発振可能
	1	発振停止

注1．ビット5は、Read Onlyです。

2．CPUがサブシステム・クロックで動作しているとき、メイン・システム・クロックの発振の停止は、MCCを使用してください。STOP命令は使用しないでください。

注意 ビット3には、必ず0を設定してください。

備考1．f_x：メイン・システム・クロック発振周波数

2．f_{XT}：サブシステム・クロック 発振周波数

μPD78018F, 78018FY, 78014Hサブシリーズの一番速い命令は、CPUクロック 4 クロックで実行されます。したがって、CPUクロック (f_{CPU}) と最小命令実行時間の関係は、表 3 - 2 のようになります。

表 3 - 2 CPUクロックと最小命令実行時間の関係
(μPD78018F, 78018FY, 78014Hサブシリーズ)

CPUクロック (f_{CPU})	最小命令実行時間: $4/f_{CPU}$
f_X	$0.4 \mu s$
$f_X/2$	$0.8 \mu s$
$f_X/2^2$	$1.6 \mu s$
$f_X/2^3$	$3.2 \mu s$
$f_X/2^4$	$6.4 \mu s$
f_{XT}	$122 \mu s$

備考 $f_X = 10.0 \text{ MHz}$, $f_{XT} = 32.768 \text{ kHz}$

f_X : メイン・システム・クロック発振周波数

f_{XT} : サブシステム・クロック発振周波数

図3 - 2 プロセッサ・クロック・コントロール・レジスタのフォーマット
(μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズ)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W ^{注1}
PCC	MCC	FRC	CLS	CSS	0	PCC2	PCC1	PCC0	FFFBH	04H	R/W

CSS	PCC2	PCC1	PCC0	CPUクロック(f_{CPU})の選択
0	0	0	0	f_x
	0	0	1	$f_x/2$
	0	1	0	$f_x/2^2$
	0	1	1	$f_x/2^3$
	1	0	0	$f_x/2^4$
1	0	0	0	$f_{xt}/2$
	0	0	1	
	0	1	0	
	0	1	1	
	1	0	0	
上記以外				設定禁止

CLS	CPUクロックのステータス
0	メイン・システム・クロック
1	サブシステム・クロック

FRC	サブシステム・クロックのフィードバック抵抗の選択
0	内蔵フィードバック抵抗を使用する
1	内蔵フィードバック抵抗を使用しない

MCC	メイン・システム・クロックの発振の制御 ^{注2}
0	発振可能
1	発振停止

注1．ビット5は、Read Onlyです。

2．CPUがサブシステム・クロックで動作しているとき、メイン・システム・クロックの発振の停止はMCCを使用してください。STOP命令は使用しないでください。

注意1．ビット3には、必ず0を設定してください。

2．外部クロックを入力しているとき、MCCをセットしないでください。これはX2端子が V_{DD1} にプルアップされるためです。

備考1． f_x ：メイン・システム・クロック発振周波数

2． f_{xt} ：サブシステム・クロック発振周波数

μPD780024A, 780024AY, 780034A, 780034AYサブシリーズの一番速い命令は, CPUクロック 2 クロックで実行されます。したがって, CPUクロック (f_{CPU}) と最小命令実行時間の関係は, 表 3 - 3 のようになります。

表 3 - 3 CPUクロックと最小命令実行時間の関係
(μPD780024A, 780024AY, 780034A, 780034AYサブシリーズ)

CPUクロック (f_{CPU})	最小命令実行時間: $2/f_{CPU}$
f_X	0.24 μs
$f_X/2$	0.48 μs
$f_X/2^2$	0.96 μs
$f_X/2^3$	1.9 μs
$f_X/2^4$	3.8 μs
$f_{XT}/2$	122 μs

備考 $f_X = 8.38 \text{ MHz}$, $f_{XT} = 32.768 \text{ kHz}$

f_X : メイン・システム・クロック発振周波数

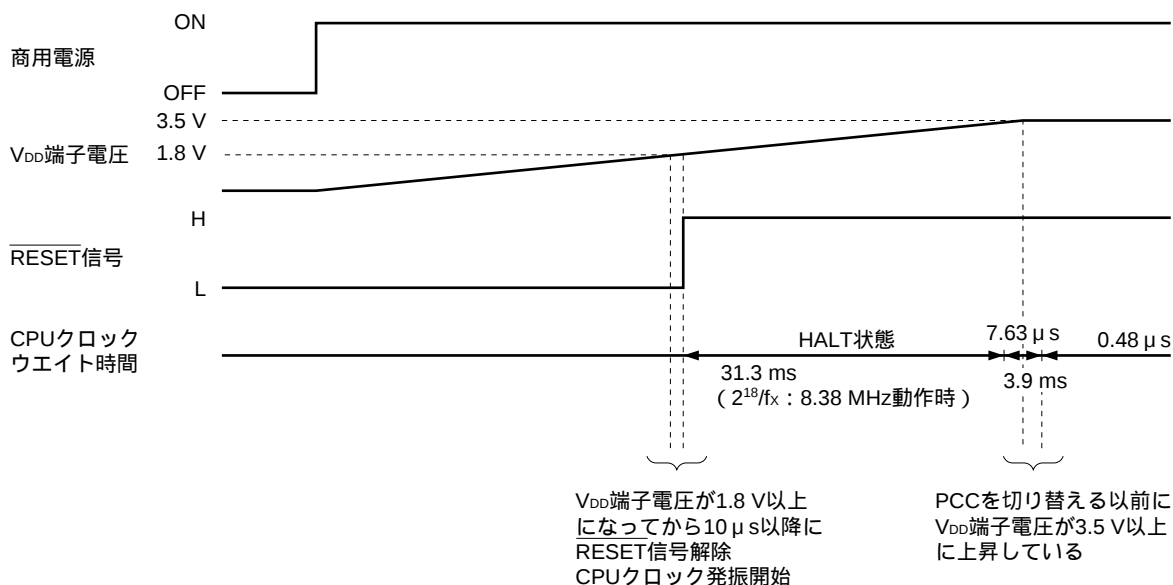
f_{XT} : サブシステム・クロック発振周波数

3.1 RESET後のPCCの切り替え

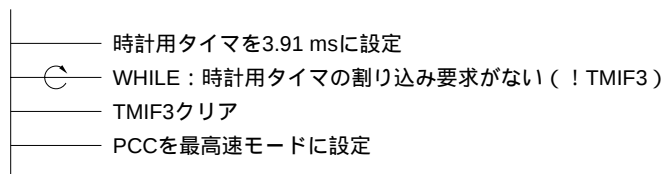
RESET信号発生により、CPUクロックはメイン・システム・クロックの最低速モードが選択されます。したがって、高速処理で動作させる場合は、プロセッサ・クロック・コントロール・レジスタ（PCC）を書き換えてCPUクロックを最高速に設定します。ただし、最高速モードで動作するためには、V_{DD}端子電圧が高速動作可能な範囲まで上昇し、安定している必要があります。

この例では、電圧が上昇するまでの時間を、時計用タイマ（インターバル時間は3.91 msを選択）でウエイトします。ウエイト後にCPUクロックを最高速に切り替えます。

図3-3 RESET後のCPUクロック切り替え（μPD78018Fサブシリーズの場合）



(1) SPDチャート



(2) プログラム・リスト

```

;*****
;*      ウェイト設定
;*****
TMC2=#00110110B      ; 時計用タイマ3.91msに設定
while_bit(!TMIF3)     ; 3.91ms?
endw
CLR1   WTIF
PCC=#00000000B        ; CPUクロックを最高速に設定

```

3.2 パワーオン / オフ時の切り替え

78K/0シリーズは、プロセッサ・クロック・コントロール・レジスタ（PCC）の設定によりサブシステム・クロックを選択することで、超低消費電流で動作できます。したがって、NiCd電池やスーパー・キャパシタなどのバックアップ電源をシステムに付加することにより、停電時でも動作を続けることができます。

この例では、電源のオン / オフをINTP1（検出エッジは、立ち上がり、立ち下りの両エッジ検出を選択）で検出し、そのときのポート・レベルにより、オン / オフを判別し、PCCを切り替えます。図3 - 4に回路例を、図3 - 5にシステム・クロックの切り替えタイミングを示します。

図3 - 4 システム・クロック切り替え用回路例

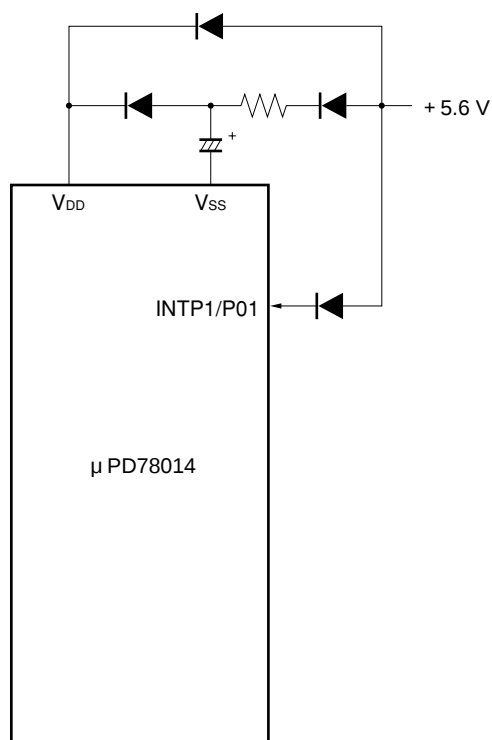
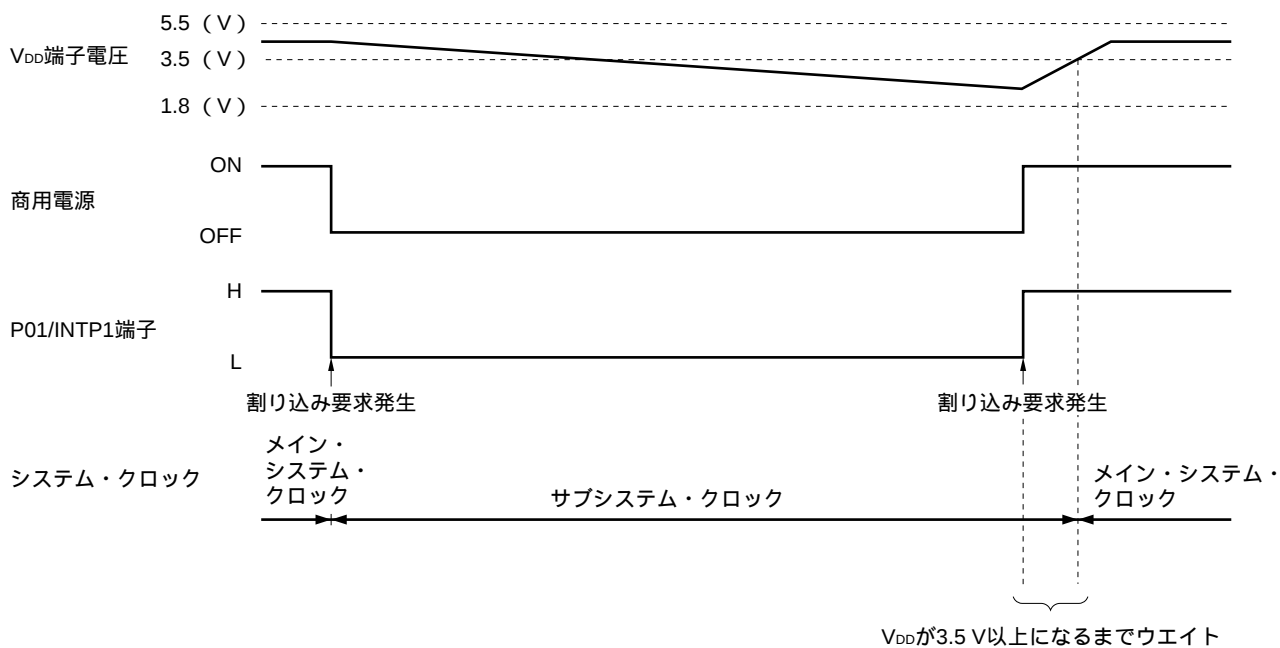
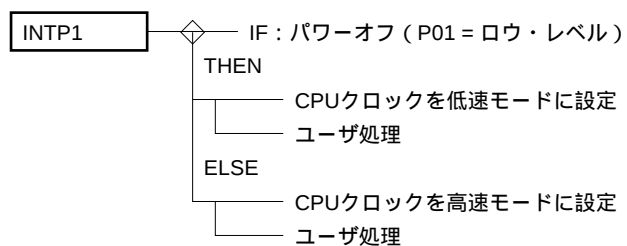


図3 - 5 パワーオン/オフ時のシステム・クロックの切り替え (μ PD78018Fサブシリーズの場合)



(1) SPDチャート



(2) プログラム・リスト

```

VEP0    CSEG    AT 08H
        DW      INT P1          ; I N T P 1 のベクタ・アドレスの設定

        MOV     INTMO, #00110000B    ; 両エッジ検出モード
        CLR1    PMK1
        EI

;*****
;*      低速／高速モード設定
;*****
INT P1:
        if_bit(!P0.1)
;          内蔵ハードウェアの設定 (低速)
;          ユーザ処理

        PCC=#10010000B          ; 低速モードに設定

        else
;          内蔵ハードウェアの設定 (高速)
;          ユーザ処理

        PCC=#00000000B          ; 高速モードに設定

        endif
        RETI

```

第4章 ウォッチドッグ・タイマの応用

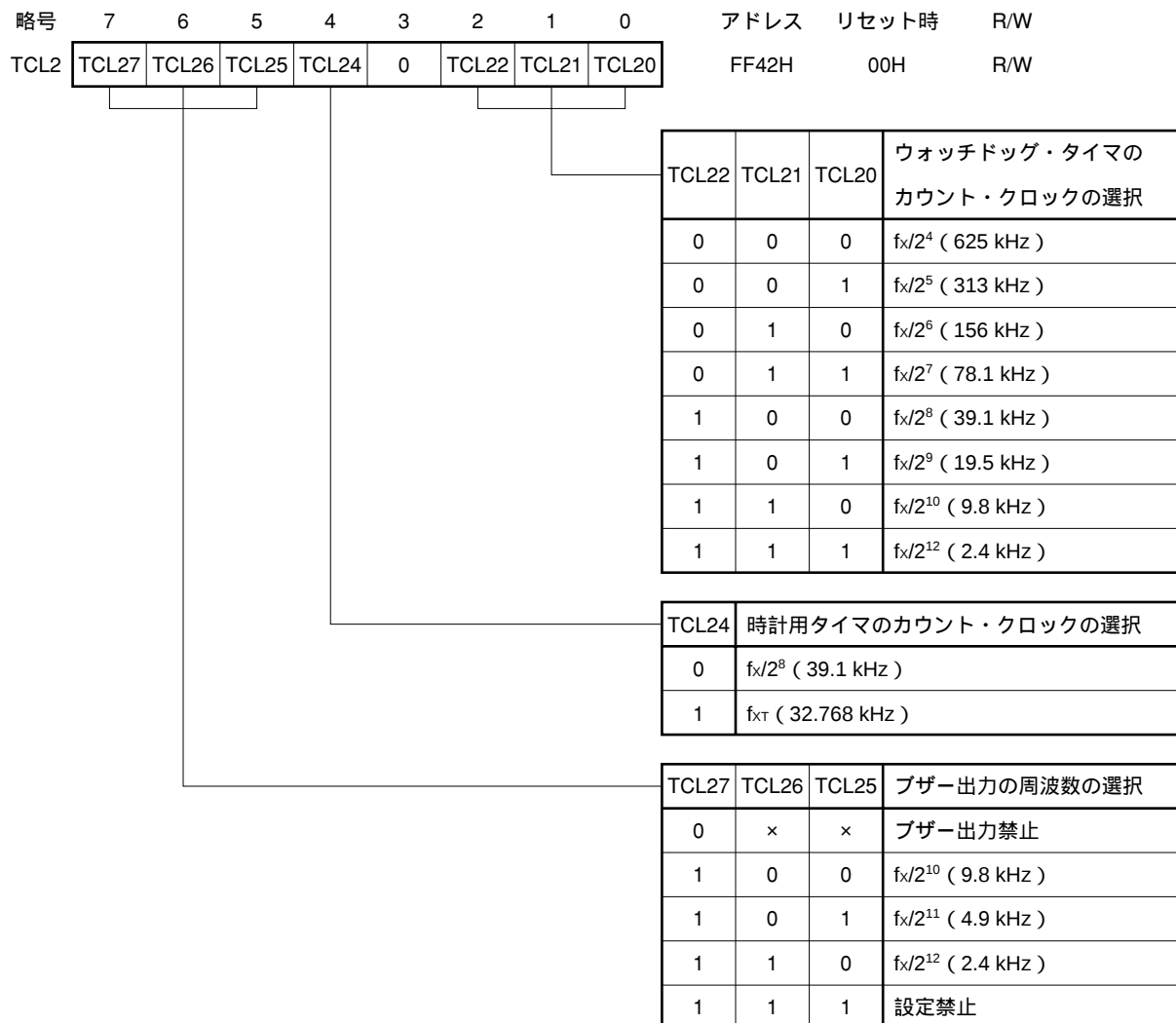
78K/0シリーズのウォッチドッグ・タイマには、マイコンの暴走を検出するウォッチドッグ・タイマ・モードと、インターバル・タイマ・モードの2種類の機能があります。

ウォッチドッグ・タイマは、次のレジスタにより設定します。

- ・タイマ・クロック選択レジスタ2 (TCL2) : μ PD78018F, 78018FY, 78014Hサブシリーズ
- ・ウォッチドッグ・タイマ・クロック選択レジスタ (WDCS) : μ PD780024A, 780024AY, 780034A, 780034AY
サブシリーズ
- ・ウォッチドッグ・タイマ・モード・レジスタ (WDTM)

注意 μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズに内蔵されているレジスタ・フォーマットは、この章のプログラム例で使用されているレジスタのフォーマットとは異なります。したがって、 μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズのいずれかを用いて、この章で記載されているプログラム例を使用する場合は、各レジスタの設定を各製品のレジスタの設定に置き換えて使用してください。

図4 - 1 タイマ・クロック選択レジスタ2のフォーマット
(μ PD78018F, 78018FY, 78014Hサブシリーズ)



★ 注意1. TCL2の書き換えは、変更対象の時計用タイマまたはブザーの動作を停止させて行ってください（同一データの上書きでは停止不要）。

停止方法は次のとおりです。

- ・ブザー出力 : TCL2のビット7 (TCL27) に0を入力
- ・時計用タイマ : 時計用タイマ・モード・コントロール・レジスタ (TMC2) のビット2 (TMC22) に0を入力

2. ウォッチドッグ・タイマ動作開始後、カウント・クロックの変更 (TCL20-TCL22の書き換え) は禁止です。

備考1. f_x : メイン・システム・クロック発振周波数

2. f_{XT} : サブシステム・クロック発振周波数

3. × : don't care

4. () 内は、 $f_x = 10.0$ MHz, $f_{XT} = 32.768$ kHz動作時。

図4 - 2 ウォッチドッグ・タイマ・クロック選択レジスタのフォーマット
(μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズ)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
WDCS	0	0	0	0	0	WDCS2	WDCS1	WDCS0	FF42H	00H	R/W

WDCS2	WDCS1	WDCS0	ウォッチドッグ・タイマ/インターバル・タイマのオーバフロー時間
0	0	0	$2^{12}/f_x$ (489 μ s)
0	0	1	$2^{13}/f_x$ (978 μ s)
0	1	0	$2^{14}/f_x$ (1.96 ms)
0	1	1	$2^{15}/f_x$ (3.91 ms)
1	0	0	$2^{16}/f_x$ (7.82 ms)
1	0	1	$2^{17}/f_x$ (15.6 ms)
1	1	0	$2^{18}/f_x$ (31.3 ms)
1	1	1	$2^{20}/f_x$ (125 ms)

備考1 . f_x : メイン・システム・クロック発振周波数

2 . () 内は , $f_x = 8.38$ MHz動作時。

図4 - 3 ウォッチドッグ・タイマ・モード・レジスタのフォーマット
(μ PD78018F, 78018FY, 78014Hサブシリーズ)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
WDTM	RUN	0	0	WDTM4	WDTM3	0	0	0	FFF9H	00H	R/W

WDTM4	WDTM3	ウォッチドッグ・タイマの動作モードの選択 ^{注1}
0	×	インターバル・タイマ・モード ^{注2} (オーバーフロー発生時, マスカブル割り込み要求発生)
1	0	ウォッチドッグ・タイマ・モード1 (オーバーフロー発生時, ノンマスカブル割り込み要求発生)
1	1	ウォッチドッグ・タイマ・モード2 (オーバーフロー発生時, リセット動作を起動)

RUN	ウォッチドッグ・タイマの動作の選択 ^{注3}
0	カウントの停止
1	カウンタをクリア後, カウントを開始

注1 . WDTM3, WDTM4は, 一度1にセットされると, ソフトウェアで0にクリアできません。

2 . RUNに1を設定した時点でインターバル・タイマとして動作を開始します。

3 . RUNは, 一度1にセットされると, ソフトウェアで0にクリアできません。したがって, カウントを開始すると, $\overline{\text{RESET}}$ 入力以外で停止させることはできません。

注意1 . RUNに1をセットし, ウォッチドッグ・タイマをクリアしたとき, 実際のオーバーフロー時間は, タイマ・クロック選択レジスタ2 (TCL2) で設定した時間より最大0.5 %短くなります。

2 . ウォッチドッグ・タイマ・モード1, 2を使用する場合は, 割り込み要求フラグ (TMIF4) が0になっていることを確認してからWDTM4を1にセットしてください。

TMIF4が1の状態ではWDTM4を1にセットすると, WDTM3の内容にかかわらず, ノンマスカブル割り込み要求が発生します。

備考 × : don't care

図4 - 4 ウォッチドッグ・タイマ・モード・レジスタのフォーマット
(μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズ)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
WDTM	RUN	0	0	WDTM4	WDTM3	0	0	0	FFF9H	00H	R/W

WDTM4	WDTM3	ウォッチドッグ・タイマの動作モードの選択 ^{注1}
0	×	インターバル・タイマ・モード ^{注2} (オーバーフロー発生時, マスカブル割り込み要求発生)
1	0	ウォッチドッグ・タイマ・モード1 (オーバーフロー発生時, ノンマスカブル割り込み要求発生)
1	1	ウォッチドッグ・タイマ・モード2 (オーバーフロー発生時, リセット動作を起動)

RUN	ウォッチドッグ・タイマの動作モードの選択 ^{注3}
0	カウントの停止
1	カウンタをクリアし, カウントを開始

注1 . WDTM3, WDTM4は, 一度1にセットされると, ソフトウェアで0にクリアできません。

- ★
- 2 . RUNに1を設定した時点で, インターバル・タイマとして動作を開始します。
 - 3 . RUNは, 一度1にセットされると, ソフトウェアで0にクリアできません。したがって, カウントを開始すると, $\overline{\text{RESET}}$ 入力以外で停止させることはできません。

注意 RUNに1をセットし, ウォッチドッグ・タイマをクリアしたとき, 実際のオーバーフロー時間は, ウォッチドッグ・タイマ・クロック選択レジスタ(WDCS)で設定した時間より最大0.5 %短くなります。

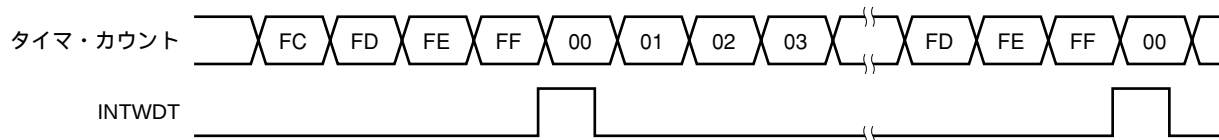
備考 × : don't care

4.2 インターバル・タイマ・モードの設定

インターバル・タイマ・モードとして使用する場合，インターバル時間はタイマ・クロック選択レジスタ2（TCL2）で設定します（インターバル時間 = 0.488 ms-125 ms）。このインターバル・タイマは，タイマのオーバーフローの発生により，割り込み要求フラグ（TMIF4）をセットします。

この例では，0.977 ms，7.82 ms，125 msの3種類の時間についての設定を示します。

図4 - 5 ウォッチドッグ・タイマのカウント・タイミング



（１）プログラム・リスト

0.977 msの設定

TCL2 = #00000001B ; 0.977 msに設定

WDTM = #10001000B ; インターバル・タイマ・モードを選択

7.82 msの設定

TCL2 = #00000100B ; 7.82 msに設定

WDTM = #10001000B ; インターバル・タイマ・モードを選択

125 msの設定

TCL2 = #00000111B ; 125 msに設定

WDTM = #10001000B ; インターバル・タイマ・モードを選択

第5章 16ビット・タイマ/イベント・カウンタの応用

78K/0シリーズの16ビット・タイマ/イベント・カウンタには、次の機能があります。

- ・ インターバル・タイマ
- ・ PWM出力 (μ PD78018F, 78018FY, 78014Hサブシリーズのみ)
- ・ パルス幅測定
- ・ 外部イベント・カウンタ
- ・ 方形波出力

16ビット・タイマ/イベント・カウンタは次のレジスタにより設定します。

< μ PD78018F, 78018FY, 78014Hサブシリーズ >

- ・ タイマ・クロック選択レジスタ0 (TCL0)
- ・ 16ビット・タイマ・モード・コントロール・レジスタ (TMC0)
- ・ 16ビット・タイマ出力コントロール・レジスタ (TOC0)
- ・ ポート・モード・レジスタ3 (PM3)
- ・ 外部割り込みモード・レジスタ (INTM0)
- ・ サンプリング・クロック選択レジスタ (SCS)

< μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズ >

- ・ 16ビット・タイマ・モード・コントロール・レジスタ (TMC0)
- ・ キャプチャ/コンペア・コントロール・レジスタ (CRC0)
- ・ 16ビット・タイマ出力コントロール・レジスタ (TOC0)
- ・ プリスケアラ・モード・レジスタ (PRM0)
- ・ ポート・モード・レジスタ7 (PM7)

注意 μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズに内蔵されているレジスタ・フォーマットは、この章のプログラム例で使用されているレジスタのフォーマットとは異なります。したがって、 μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズのいずれかを用いて、この章で記載されているプログラム例を使用する場合は、各レジスタの設定を各製品のレジスタの設定に置き換えて使用してください。

図5 - 1 タイマ・クロック選択レジスタ0のフォーマット
(μ PD78018F, 78018FY, 78014Hサブシリーズ)



- 注意 1 . TI0/INTP0端子の有効エッジは、外部割り込みモード・レジスタ (INTM0) で設定してください。また、サンプリング・クロックの周波数の選択は、サンプリング・クロック選択レジスタ (SCS) で行います。
- 2 . PCL出力を許可するときは、TCL00-TCL03を設定したのち、1ビット・メモリ操作命令でCLOEに1を設定してください。
- 3 . TM0のカウント・クロックがTI0で、カウント値を読み出すとき、キャプチャ・レジスタ (CR01) からではなく、TM0から読み出してください。
- 4 . TCL0を同一データ以外に書き換える場合は、いったんタイマ動作を停止させたのちに行ってください。

備考 1 . f_x : メイン・システム・クロック発振周波数

2 . f_{XT} : サブシステム・クロック発振周波数

3 . TI0 : 16ビット・タイマ/イベント・カウンタの入力端子

4 . TM0 : 16ビット・タイマ・レジスタ

5 . () 内は、 $f_x = 10.0$ MHz , $f_{XT} = 32.768$ kHz動作時。

図5 - 2 16ビット・タイマ・モード・コントロール・レジスタのフォーマット
(μ PD78018F, 78018FY, 78014Hサブシリーズ)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
TMC0	0	0	0	0	TMC03	TMC02	TMC01	OVF0	FF48H	00H	R/W

OVF0	16ビット・タイマ・レジスタのオーバーフロー検出
0	オーバーフローなし
1	オーバーフローあり

TMC03	TMC02	TMC01	動作モードおよび クリア・モードの選択	TO0の出力 タイミングの選択	割り込み要求の発生
0	0	0	動作停止 (TM0は0にクリア)	変化なし	発生しない
0	0	1	PWMモード (フリーランニング)	PWMパルス出力	TM0とCR00の一致で発生
0	1	0	フリーランニング・モード	TM0とCR00の一致	
0	1	1		TM0とCR00の一致または TI0の有効エッジ	
1	0	0	TI0の有効エッジでクリア& スタート	TM0とCR00の一致	
1	0	1		TM0とCR00の一致または TI0の有効エッジ	
1	1	0	TM0とCR00の一致でクリア &スタート	TM0とCR00の一致	
1	1	1		TM0とCR00の一致または TI0の有効エッジ	

注意1．クリア・モードおよびTO0の出力タイミングの切り替えは、タイマ動作を停止（TMC01-TMC03に0, 0, 0を設定）させたのちに行ってください。

2．TI0/INTP0端子の有効エッジは、外部割り込みモード・レジスタ（INTM0）で設定してください。また、サンプリング・クロックの周波数の選択は、サンプリング・クロック選択レジスタ（SCS）で行います。

3．PWMモードを使用するときは、PWMモード設定後、CR00にデータを設定してください。

4．TM0とCR00の一致でクリア&スタートするモードを選択した場合、CR00の設定値がFFFFHで、TM0の値がFFFFHから0000Hに変化するとき、OVF0フラグが1に設定されます。

備考1．TO0：16ビット・タイマ/イベント・カウンタの出力端子

2．TI0：16ビット・タイマ/イベント・カウンタの入力端子

3．TM0：16ビット・タイマ・レジスタ

4．CR00：コンペア・レジスタ00

★

図5 - 3 16ビット・タイマ・モード・コントロール・レジスタのフォーマット

(μPD780024A, 780024AY, 780034A, 780034AYサブシリーズ)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
TMC0	0	0	0	0	TMC03	TMC02	TMC01	OVF0	FF60H	00H	R/W

TMC03	TMC02	TMC01	動作モードおよび クリア・モードの選択	TO0の出力 タイミングの選択	割り込み要求の発生
0	0	0	動作停止 (TM0は 0 にクリア)	変化なし	発生しない
0	0	1			
0	1	0	フリーランニング・モード	TM0とCR00の一致または TM0とCR01の一致	TM0とCR00の一致 または TM0とCR01の一致 で発生
0	1	1		TM0とCR00の一致 , TM0とCR01の一致または TI00の有効エッジ	
1	0	0	TI00の有効エッジで	—	
1	0	1	クリア&スタート		
1	1	0	TM0とCR00の一致で クリア&スタート	TM0とCR00の一致または TM0とCR01の一致	
1	1	1		TM0とCR00の一致 , TM0とCR01の一致または TI00の有効エッジ	

OVF0	16ビット・タイマ・レジスタのオーバーフロー検出
0	オーバーフローなし
1	オーバーフローあり

注意1 . OVF0フラグ以外のビットには、タイマ動作を停止してから書き込んでください。

2 . TI00/TO0/P70端子の有効エッジは、プリスケアラ・モード・レジスタ0(PRM0)で設定します。

3 . TM0とCR00の一致でクリア&スタートするモードを選択した場合、CR00の設定値がFFFFHで、
TM0の値がFFFFHから0000Hに変化するとき、OVF0フラグが1に設定されます。

備考 TO0 : 16ビット・タイマ/イベント・カウンタの出力端子

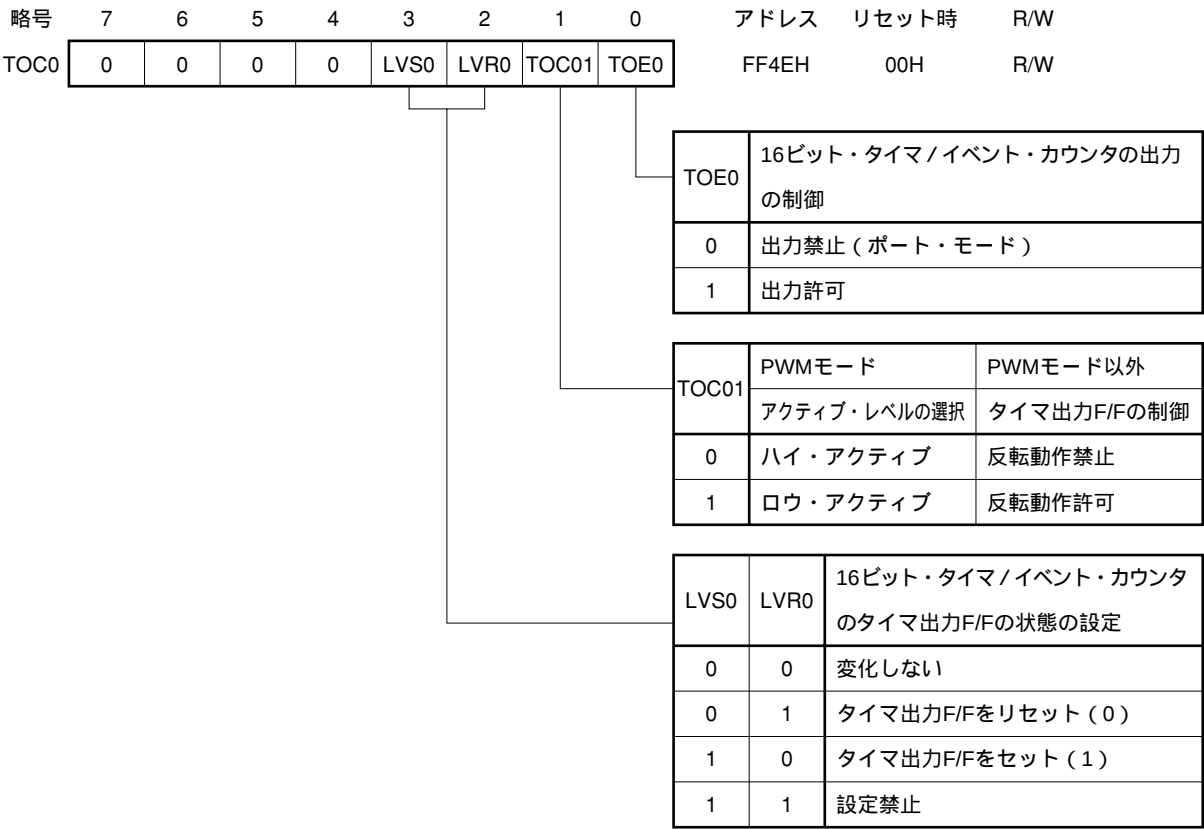
TI00 : 16ビット・タイマ/イベント・カウンタの入力端子

TM0 : 16ビット・タイマ・レジスタ

CR00 : コンペア・レジスタ00

CR01 : コンペア・レジスタ01

図5 - 4 16ビット・タイマ出力コントロール・レジスタのフォーマット
(μ PD78018F, 78018FY, 78014Hサブシリーズ)



注意 1 . TOC0は、必ずタイマ動作を停止させたのちに設定してください。

2 . LVS0 , LVR0は、データ設定後に読み出すと 0 が読み出されます。

★

図5 - 5 16ビット・タイマ出力コントロール・レジスタのフォーマット

(μPD780024A, 780024AY, 780034A, 780034AYサブシリーズ)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
TOC0	0	0	0	TOC04	LVS0	LVR0	TOC01	TOE0	FF63H	00H	R/W

TOE0	タイマ0の出力の制御
0	出力禁止 (出力は0レベルに固定)
1	出力許可

TOC01	CR00とTM0の一致によるタイマ出力F/Fの制御
0	反転動作禁止
1	反転動作許可

LVS0	LVR0	タイマ0のタイマ出力F/Fの状態の設定
0	0	変化しない
0	1	タイマ出力F/Fをリセット(0)
1	0	タイマ出力F/Fをセット(1)
1	1	設定禁止

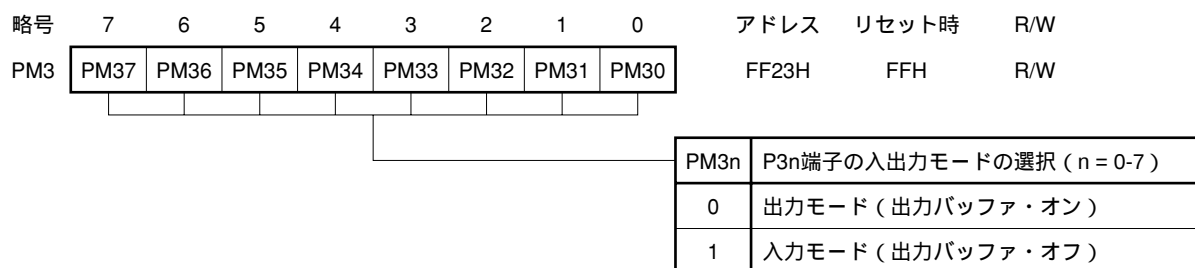
TOC04	CR01とTM0の一致によるタイマ出力F/Fの制御
0	反転動作禁止
1	反転動作許可

注意1．TOC0は、必ずタイマ動作を停止させてから設定してください(ただしOSPTは除く)。

2．データ設定後にLVS0, LVR0を読み出すと、0になっています。

図5 - 6 ポート・モード・レジスタ3のフォーマット

(μPD78018F, 78018FY, 78014Hサブシリーズ)



注意 P30/TO0端子をタイマ出力として使用するとき、PM30およびP30の出力ラッチに0を設定してください。

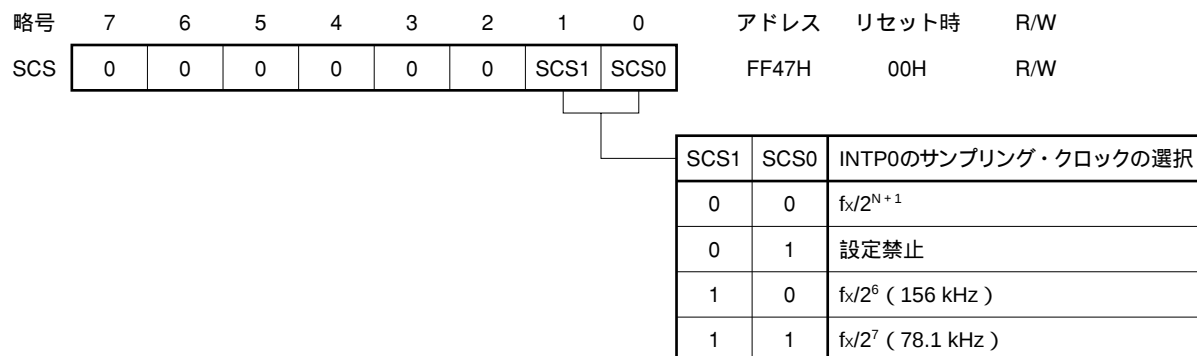
図5 - 7 外部割り込みモード・レジスタのフォーマット

(μPD78018F, 78018FY, 78014Hサブシリーズ)



注意 INTP0端子の有効エッジは、16ビット・タイマ・モード・コントロール・レジスタ (TMC0) のビット1-3 (TMC01-TMC03) に0, 0, 0を設定し、タイマ動作を停止させたのちに設定してください。

図5 - 8 サンプルング・クロック選択レジスタのフォーマット
(μ PD78018F, 78018FY, 78014Hサブシリーズ)



注意 $f_x/2^{N+1}$ はCPUへ供給されるクロック, $f_x/2^6$, $f_x/2^7$ は周辺ハードウェアへ供給されるクロックです。 $f_x/2^{N+1}$ はHALTモード中は停止します。

備考1 . N : プロセッサ・クロック・コントロール・レジスタ (PCC) のビット0-2 (PCC0-PCC2) に設定した値 (N = 0-4)。

2 . f_x : メイン・システム・クロック発振周波数

3 . () 内は, $f_x = 10.0$ MHz動作時。

図5 - 9 キャプチャ/コンペア・コントロール・レジスタ0のフォーマット

(μPD780024A, 780024AY, 780034A, 780034AYサブシリーズ)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
CRC0	0	0	0	0	0	CRC02	CRC01	CRC00	FF62H	04H	R/W

CRC00	CR00の動作モードの選択
0	コンペア・レジスタとして動作
1	キャプチャ・レジスタとして動作

CRC01	CR00のキャプチャ・トリガの選択
0	TI01の有効エッジでキャプチャする
1	TI00の有効エッジの逆相でキャプチャする

CRC02	CR01の動作モードの選択
0	コンペア・レジスタとして動作
1	キャプチャ・レジスタとして動作

注意1 . CRC0は、必ずタイマ動作を停止させてから設定してください。

2 . 16ビット・タイマ・モード・コントロール・レジスタ(TMC0)で、TM0とCR00の一致でクリア&スタート・モードを選択したとき、CR00をキャプチャ・レジスタに指定しないでください。

★ 3 . TI00の有効エッジに立ち上がり、立ち下りの両エッジを選択した場合には、キャプチャ動作しません。

★ 4 . キャプチャを確実にを行うために、キャプチャ・トリガはプリスケアラ・モード・レジスタα(PRM0)で選択したカウント・クロックの2回分より長いパルスが必要とします。

★

図5 - 10 プリスケアラ・モード・レジスタ0のフォーマット
(μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズ)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
PRM0	ES11	ES10	ES01	ES00	0	0	PRM01	PRM00	FF61H	00H	R/W

PRM01	PRM00	カウント・クロックの選択
0	0	f_x (8.38 MHz)
0	1	$f_x/2^2$ (2.09 MHz)
1	0	$f_x/2^6$ (131 kHz)
1	1	TI00有効エッジ ^注

ES01	ES00	TI00有効エッジの選択
0	0	立ち下がりエッジ
0	1	立ち上がりエッジ
1	0	設定禁止
1	1	立ち上がり, 立ち下がり両エッジ

ES11	ES10	TI01有効エッジの選択
0	0	立ち下がりエッジ
0	1	立ち上がりエッジ
1	0	設定禁止
1	1	立ち上がり, 立ち下がり両エッジ

注 外部クロックは内部クロック ($f_x/2^3$) の2回分より長いパルスが必要とします。

注意1. カウント・クロックにTI00の有効エッジを設定する場合, TI00有効エッジでクリア&スタート・モードおよびキャプチャ・トリガに設定しないでください。

また, P70/TI00/TO0端子をタイマ出力 (TO0) として使用できません。

2. PRM0は, 必ずタイマ動作を停止させてからデータを設定してください。

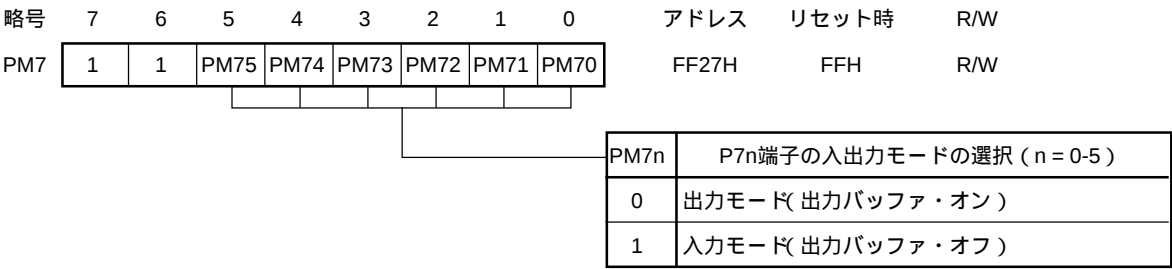
3. システム・リセット直後にTI00端子またはTI01端子がハイ・レベルの場合, TI00端子またはTI01端子の有効エッジを立ち上がりまたは両エッジに指定し, 16ビット・タイマ/カウンタ0 (TM0) の動作を許可すると, その直後の立ち上がりエッジを検出します。TI00端子またはTI01端子をプルアップしている場合などは注意してください。ただし, いったん動作を停止させたあとの再動作許可時には, 立ち上がりエッジは検出されません。

備考1. f_x : メイン・システム・クロック発振周波数

2. TI00, TI01: 16ビット・タイマ/イベント・カウンタの入力端子

3. () 内は, $f_x = 8.38$ MHz動作時。

図 5 - 11 ポート・モード・レジスタ7のフォーマット
(μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズ)



5.1 インターバル・タイマの設定

インターバル・タイマとして使用する場合、まずタイマ・クロック選択レジスタ0 (TCL0) と16ビット・タイマ・モード・レジスタ (TMC0) を設定します。16ビット・タイマのクリア・モードはTMC0で、インターバル時間はTCL0で設定します。

その後、設定時間とカウント・クロックからコンペア・レジスタ (CR00) の値を設定します。設定時間は、以下の式により決定します。

$$\text{設定時間} = (\text{コンペア・レジスタ値} + 1) \times \text{カウント・クロック周期}$$

この例では、インターバル・タイマの設定時間として、10 msと50 msの2種類について示します。

(a) 10msのインターバルの場合

TMC0の設定

TM0とCR00の一致でクリア&スタートを選択

TCL0の設定

10 ms以上の設定が可能で、分解能の最も高い $f_x/2$ モードを選択

CR00の設定

$$10 \text{ ms} = (N + 1) \times \frac{1}{8.38 \text{ MHz}/2}$$

$$N = 10 \text{ ms} \times 8.38 \text{ MHz}/2 - 1 = 41899$$

(1) プログラム・リスト

CR00=#41899

TCL0=#00100000B ; カウント・クロック $f_x/2$ を選択

TMC0=#00001100B ; 16ビット・タイマ/イベント・カウンタをTM0とCR00の一致でクリア&スタートに設定

(b) 50 msのインターバルの場合

TMC0の設定

TM0とCR00の一致でクリア & スタートを選択

TCL0の設定

50 ms以上の設定が可能で、分解能の最も高い $f_x/2^3$ モードを選択

CR00の設定

$$50 \text{ ms} = (N + 1) \times \frac{1}{8.38 \text{ MHz}/2^3}$$

$$N = 50 \text{ ms} \times 8.38 \text{ MHz}/2^3 - 1 = 52374$$

(1) プログラム・リスト

CR00=#52374

TCL0=#01000000B ; カウント・クロック $f_x/2^3$ を選択

TMC0=#00001100B ; 16ビット・タイマ/イベント・カウンタをTM0とCR00の一致でクリア & スタートに設定

5.2 PWM出力

PWM出力として使用する場合、16ビット・タイマ・モード・コントロール・レジスタ (TMC0) でPWMモードに設定し、16ビット・タイマ出力コントロール・レジスタ (TOC0) で16ビット・タイマ/イベント・カウンタを出力許可の状態に設定します。

PWMのパルス幅 (アクティブ・レベル) は、コンペア・レジスタ (CR00) に設定された値によって決定します。ただし、78K/0シリーズのPWMは14ビット分解能ですのでCR00の2-15ビットが有効となります (CR00の0, 1ビットは、'0, 0'を設定してください)。

この例では、PWMモードの基本周期を $61.0\mu\text{s}$ ($2/f_x \times 2^8$) に、アクティブ・レベルをロウ・アクティブに設定しています。また、パルス幅の設定はパラメータ (00H-0FH) の値により、上位4ビットを書き換えるプログラムになっています。したがって、この応用例は、16段階 (CR00=0FFCH-FFFCH) のPWM出力が可能です。

(1) パッケージの説明

<パブリック宣言シンボル>

PWM : PWM出力サブルーチン名称
PWMOUT : PWMアクティブ・レベルの入力パラメータ

<使用するレジスタ>

AX

<使用するRAM>

名 称	用 途	属 性	バイト
PWMOUT	PWMアクティブ・レベルの設定	SADDR	1

<ネスティング>

1 レベル 2 バイト

<使用するハードウェア>

- 16ビット・タイマ/イベント・カウンタ
- P30/TO0

<初期設定>

- 16ビット・タイマ/イベント・カウンタの設定
 - PWM出力モード TMC0=#00000010B
 - PWM基本周期 $61.0\mu\text{s}$ TCL0=#00100000B
 - ロウ・アクティブ出力 TOC0=#00000011B
- P30出力モード PM30=0
- P30出力ラッチ P30=0

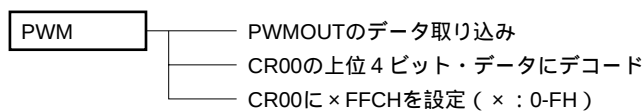
<起動方法>

RAMのPWMOUTにデータ設定後、サブルーチンPWMをコールしてください。

(2) 使用例

```
EXTRN    PWM,PWMOUT
        :
TOC0=#00000011B      ;PWM出力，ロウ・アクティブの設定
TCL0=#00100000B      ;カウント・クロックfx/2を選択
TMC0=#00000010B      ;PWMモードの設定
        :
PWMOUT=A              ;アクティブ・レベルの入力パラメータの設定
CALL     !PWM
```

(3) SPDチャート



(4) プログラム・リスト

```

PUBLIC    PWM, PWMOUT
PWM_DAT DSEG    SADDR
PWMOUT: DS    1                                ; P W M出力データ・エリア（0 - 1 5）
;*****
;*        P W M出力（1 6 段階）
;*****
PO_SEG  CSEG
PWM:
    A=PWMOUT                                ; P W M O U T の上位データ取り込み
    A<<=1
    A<<=1
    A<<=1
    A<<=1
    A|=#0FH                                ; 下位12ビットを0 F F C Hに設定
    X=#0FCH
    CR0=AX
    RET

```

5.3 リモコン受信

μPD78014Hサブシリーズの16ビット・タイマ/イベント・カウンタを使用して、リモコン受信を行うプログラム例を2種類紹介します。

- リモコンの有効エッジ検出ごとにカウンタをクリアし、次の有効エッジを検出したときのタイマ・カウント値（キャプチャ・レジスタCR01）からパルス幅を測定します。
- タイマをフリーランニングで動作させ、有効エッジ間のカウンタの差からパルス幅を測定します。また、PWM出力も同時に行います。

リモコン信号は、PIN受光ダイオードで受信され、リモコン用受信プリアンプμPC1490を介して、P00/INTP0端子に入力します。図5-12にリモコン受信回路例を、図5-13にリモコン信号のフォーマットを示します。

図5-12 リモコン受信回路例

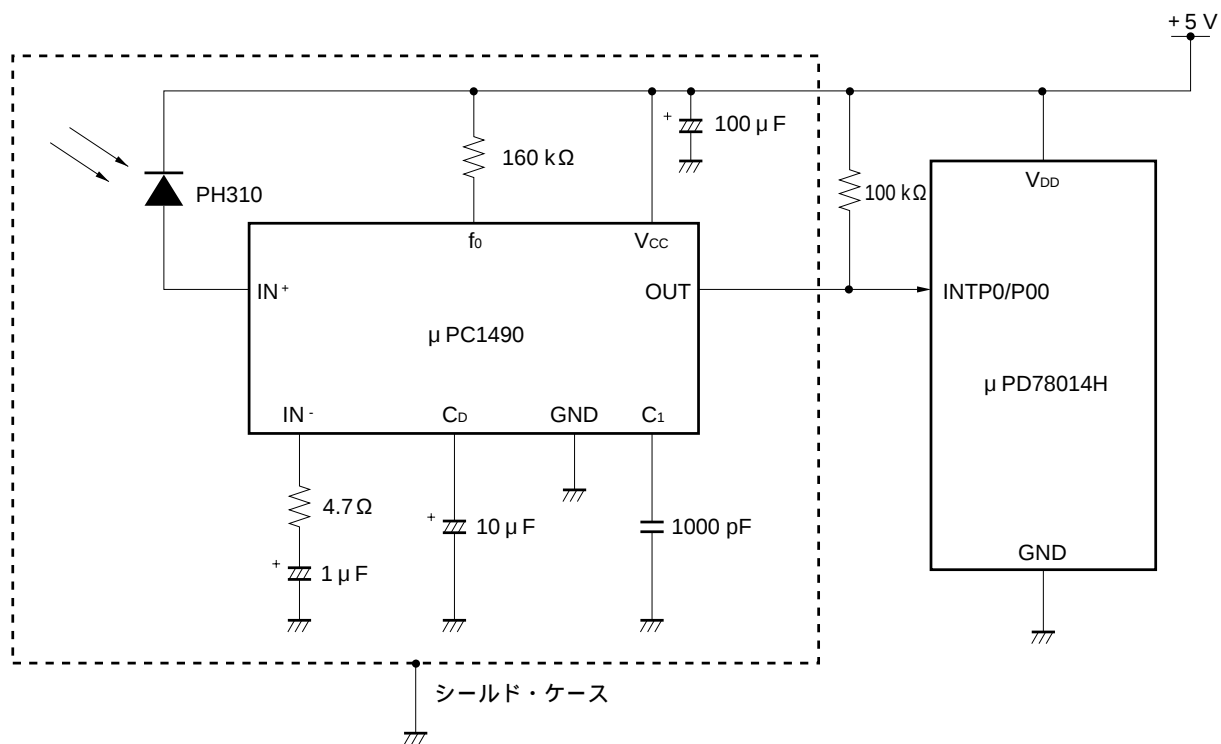
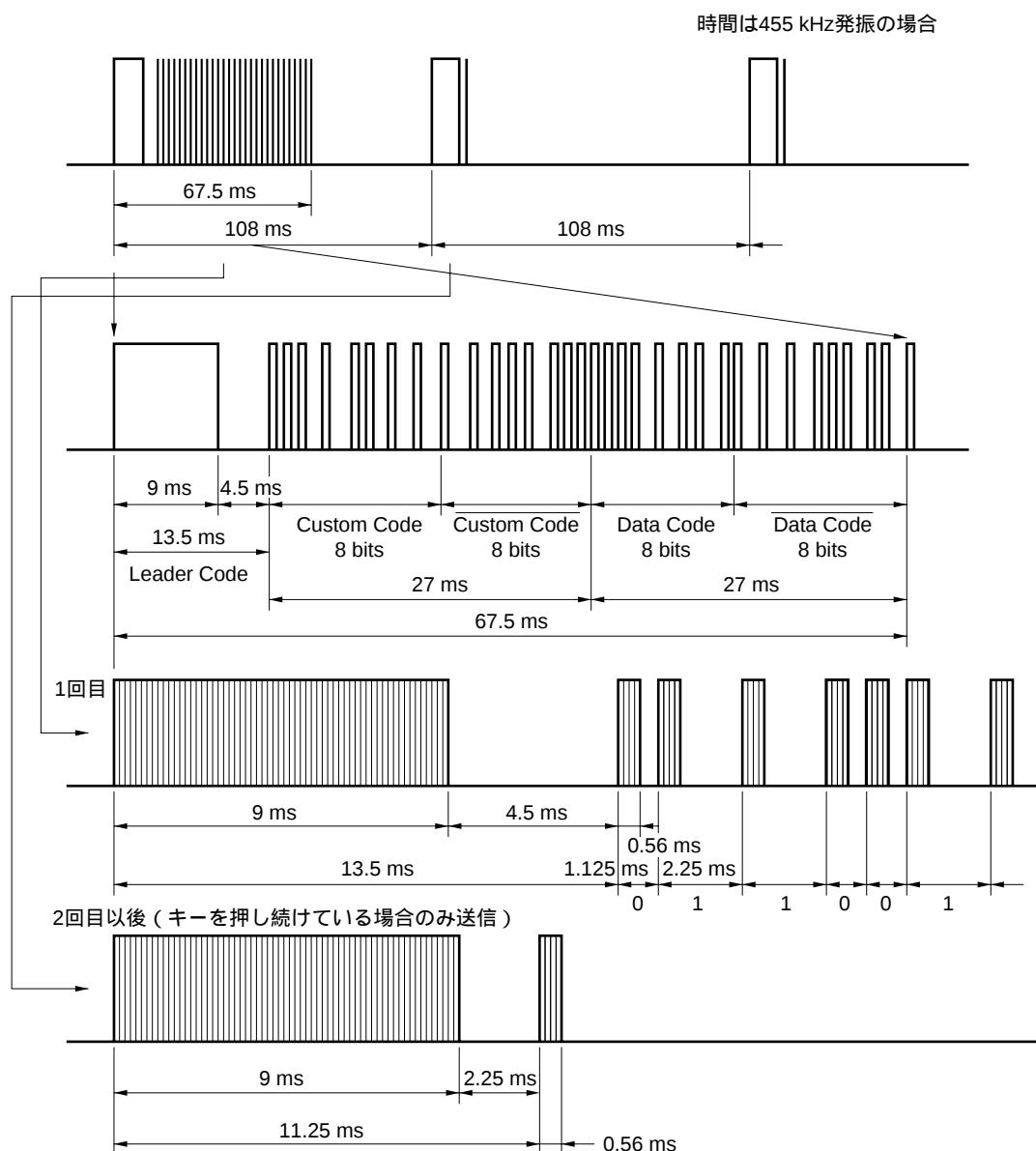
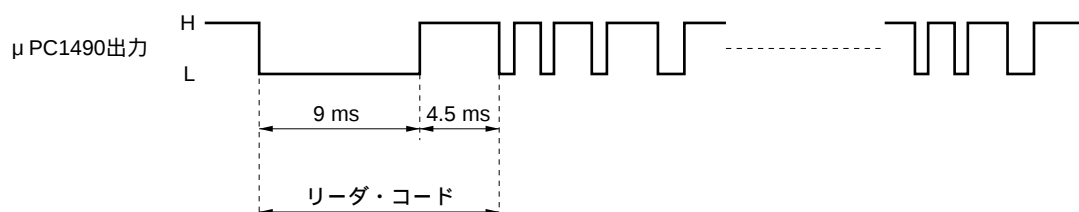


図5 - 13 リモコン送信用IC出力信号



回路例で使用しているリモコン用受信プリアンプ μ PC1490はロウ・アクティブとなるため、 μ PD78014Hサブシリーズに入力されるレベルはリモコン送信データの反転データになります。

図5 - 14 受信プリアンプの出力信号



5.3.1 カウンタ・クリアによるリモコン受信

このプログラム中で、リモコン信号を受信する場合のパルス有効幅を表5 - 1に、各信号の処理方法を から示します。リモコン信号のリピート信号は、有効入力があったから250 ms以内の場合のみ有効とします。また、正常データ取り込み後の3 msの時間内に信号を入力した場合は、データを無効とします。

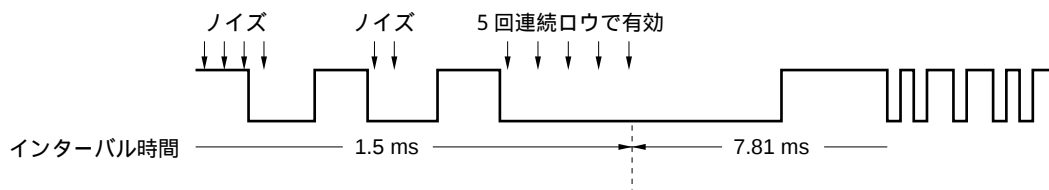
表5 - 1 入力信号の有効時間

信号名		出力時間	有効時間
リーダー・コード(ロウ)		9 ms	6.8 ms-11.8 ms
リーダー・コード (ハイ)	通常	4.5 ms	3 ms-5 ms
	リピート	2.25 ms	1.8 ms-3 ms
カスタム/ データ・コード	0	1.125 ms	0.5 ms-1.8 ms
	1	2.25 ms	1.8 ms-2.5 ms

リーダー・コード(ロウ)

16ビット・タイマ/イベント・カウンタのインターバルを1.5 msに設定し、割り込み処理でポート・レベルのサンプリングをし、5回連続でロウ・レベルを検出したときにリーダー・コードと判断しインターバルを7.81 msに変更します。その後、INTP0の立ち上がりエッジの割り込み要求によって、リーダー・コードのロウ・レベルのパルス幅を計測します。

図5 - 15 リモコン信号のサンプリング



リーダー・コード（ハイ）

INTP0の立ち下がりエッジの割り込み要求によって、リーダー・コードのハイ・レベル間のパルス幅をタイマのカウンタ値により計測します。

カスタム/データ・コード

INTP0の立ち下がりエッジの割り込み要求によって、1ビット（1周期）ごとのパルス幅を計測します。32ビット目のデータ取り込み後に、反転データとカスタム・コードの一致をテストします。さらに、33ビット目にデータの無いことを確認します。

リピート・コード検出

リーダー・コードのハイ・レベルが3 ms未満のときに、リーダー・コード出力後からINTP0立ち上がりまでのパルス幅を計測します。

リピート・コードの有効期間

有効なデータ入力後、16ビット・タイマ/イベント・カウンタの割り込み処理（1.5 msインターバル）でサンプリングし、リピート・コードの有効時間250 msを計測します。

パルス幅計測中のタイム・アウト

パルス幅計測中に16ビット・タイマ/イベント・カウンタの割り込み要求（7.81 ms）が発生したときは、タイム・アウトになりデータは無効になります。

（1）パッケージの説明

<パブリック宣言シンボル>

RMDATA : リモコン受信データを格納
RPT : リピート有効期間判断フラグ
IPDTFG : 有効データの有無判断フラグ
RMDTOK : 入力信号の有効の有無判断フラグ
RMDTSET : 入力信号の有無判断フラグ

<使用するレジスタ>

バンク 0 : AX, BC, HL

<使用するRAM>

名 称	用 途	属 性	バイト
RPTCT	リピート・コード有効時間カウンタ	SADDR	1
RMENDCT	データ入力後の無入力時間カウンタ		
SELMOD	モード選択		
LD_CT	リーダ信号検出カウンタ		
RMDATA	有効データ格納エリア		
WORKP	入力信号格納エリア	SADDRP	4

<使用するフラグ>

名 称	用 途
IPDTFG	有効データの有無
RMDTOK	入力信号の有効の有無
RMDTSET	入力信号の有無
RPT	リピート有効期間が経過したかどうかの判断

<ネスティング>

5 レベル12バイト

<使用するハードウェア>

- 16ビット・タイマ/イベント・カウンタ
- P00/INTP0

<初期設定>

- 16ビット・タイマ/イベント・カウンタの設定

TM0とCR00一致でタイマ・クリア・モード	TMC0=#00001100B
カウント/クロック $f_x/2$	TCL0=#00100000B
コンペア・レジスタ00	CR00=#6290
- INTP0サンプリング・クロック $f_x/2^7$

SCS=#00000011B
- INTP0高優先順位割り込み

PPR0=0
- 16ビット・タイマ/イベント・カウンタ割り込み許可

TMMK0=0
- カスタム・コードをCSTMに定義し、パブリック宣言
- RAMクリア

<起動方法>

INTP0, INTTM0の割り込み要求で起動

(2) 使用例

```

PUBLIC  CSTM
EXTRN  RMDATA, RPTCT
EXTBIT  RPT, RMDTSET, IPDTFG

CSTM  EQU      9DH                      ; リモコン・カスタム・コード

CR00=#6290
TCL0=#00100000B                      ; 1.5ms に設定
TMC0=#00001100B
SCS=#00000011B                      ; INTP0 サンプリング・クロックを  $f_x / 128$ 

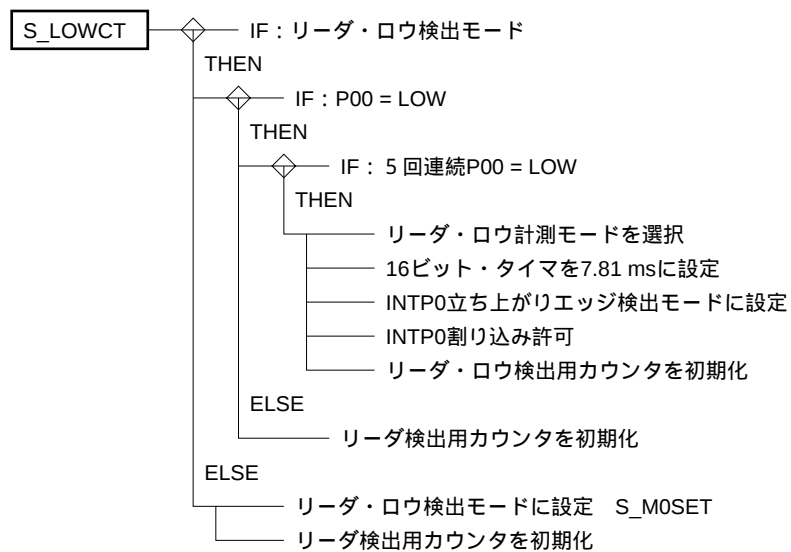
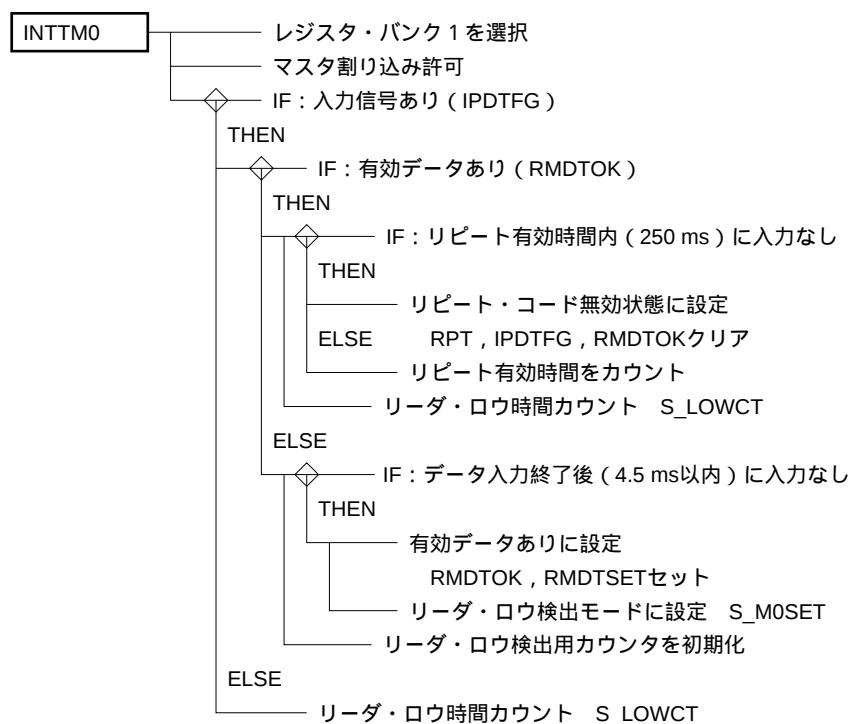
CLR1   PPR0                          ; INTP0 高優先
CLR1   RPT                          ; フラグ・クリア
CLR1   IPDTFG
CLR1   RMDTSET

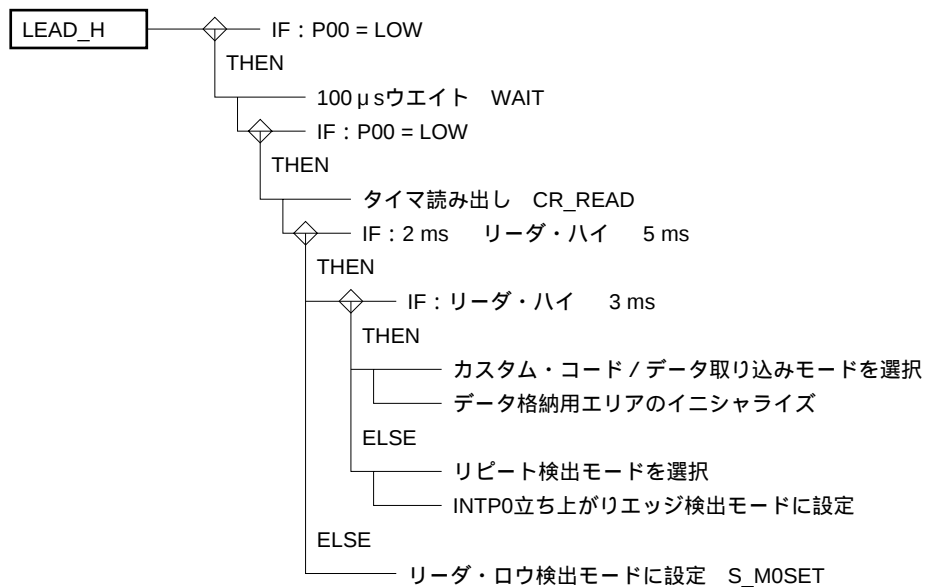
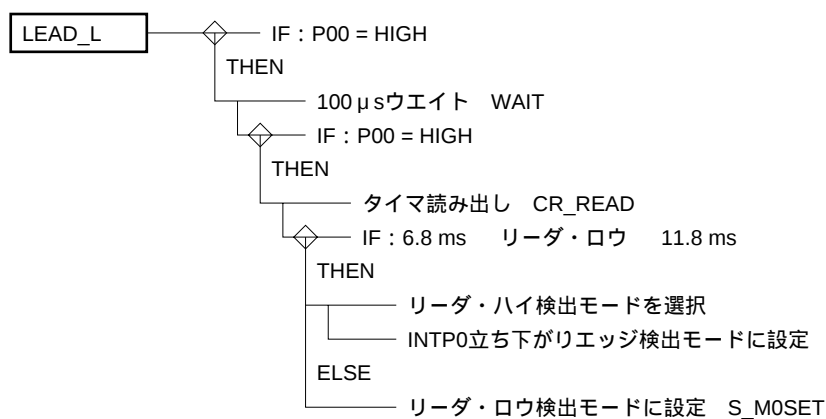
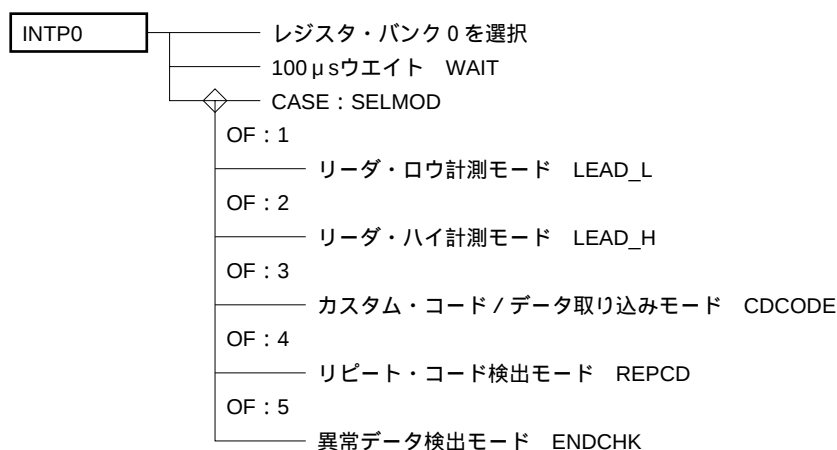
CLR1   TMMK0                          ; タイマ割り込み許可
EI

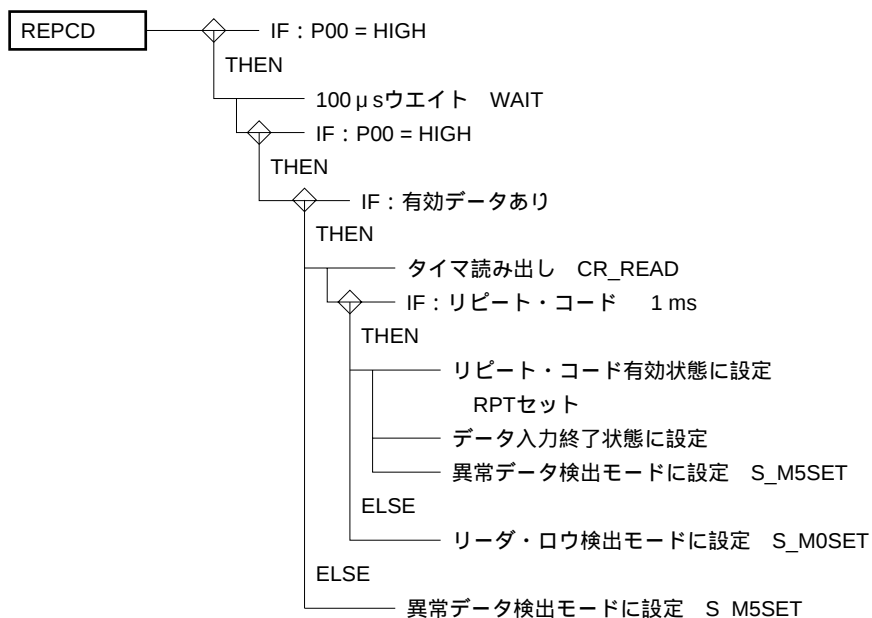
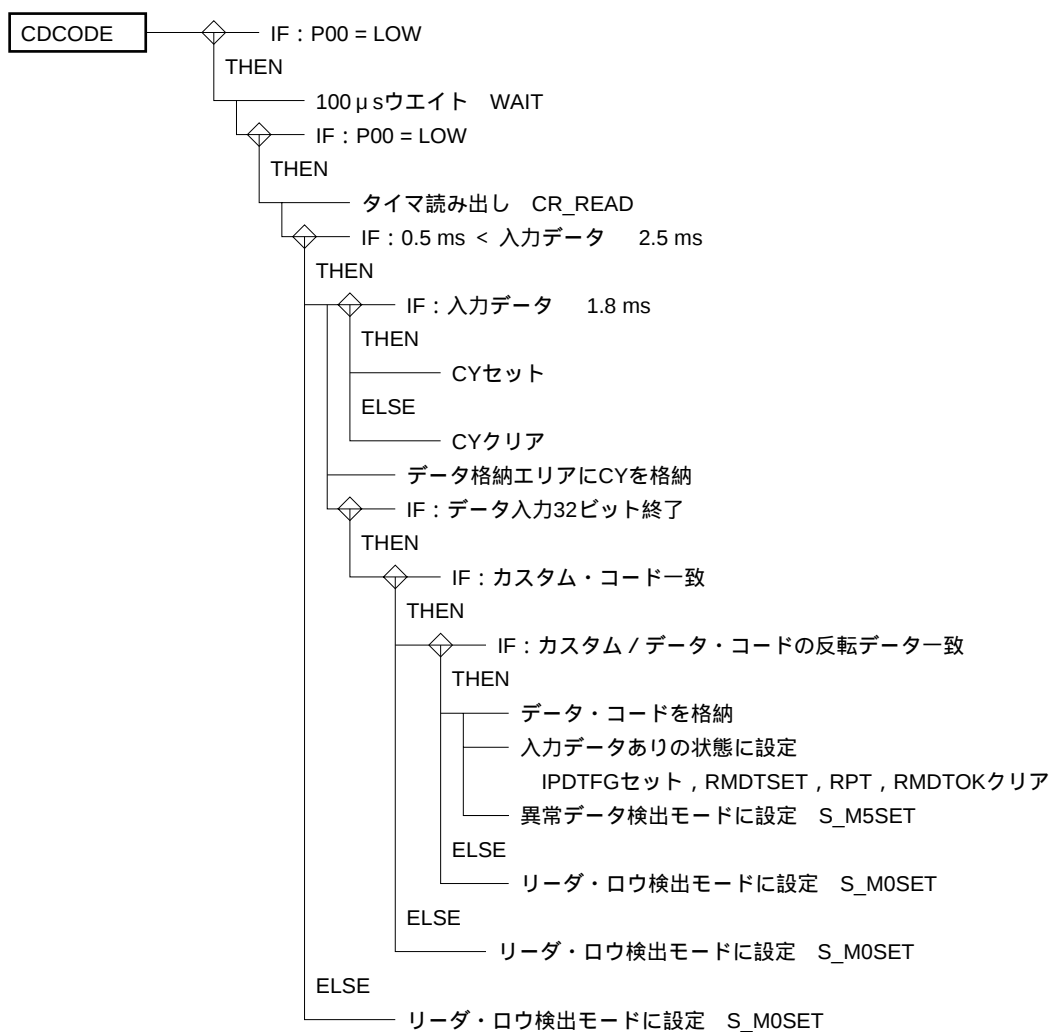
DT_TEST:
    if_bit(RMDTSET)
        CLR1   RMDTSET
        if_bit(RPT)
;
;           リピート処理
;
        else
;
;           入力ありの処理
;
        endif
    else
        if_bit(!RPT)
;
;           入力なしの処理
;
        endif
    endif
endif

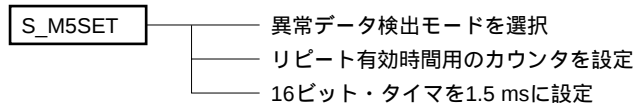
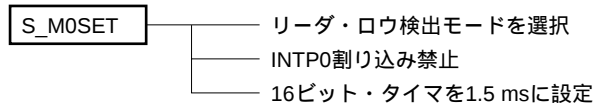
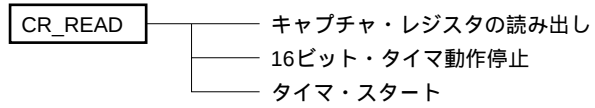
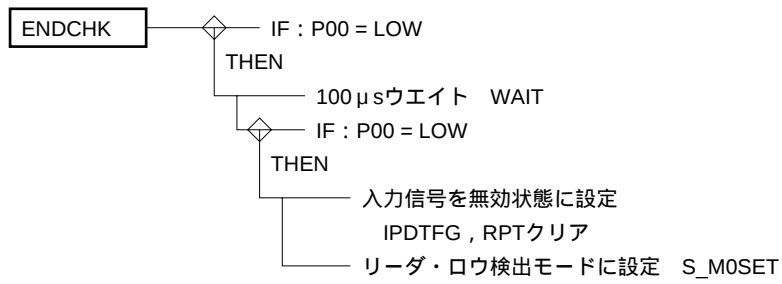
```

(3) SPDチャート









(4) プログラム・リスト

```

PUBLIC RPT, IPDTFG, RMDTOK, RMDTSET
PUBLIC RMENDCT, RPTCT, SELMOD, LD_CT, RMDATA
EXTRN  CSTM
RM_DAT DSEG  SADDR
RPTCT: DS    1                ; リピート・コード有効時間カウンタ
RMENDCT:DS   1                ; データ入力後の無入力時間カウンタ
SELMOD: DS   1                ; モード選択
LD_CT:  DS   1                ; リード信号検出カウンタ
RMDATA: DS   1                ; 有効データ格納エリア

RM_DATP DSEG  SADDRP
WORKP:  DS    4                ; 入力信号格納エリア

      BSEG
IPDTFG DBIT                    ; 有効データあり
RMDTOK DBIT                    ; 入力信号有効
RMDTSET DBIT                   ; 入力信号あり
RPT     DBIT                   ; リピート・コード有効期間

VEP0   CSEG   AT 06H
      DW     INTP0                ; I N T P 0 のベクタ・アドレスの設定

VETM0   CSEG   AT 14H
      DW     INTTMO                ; 1 6 ビット・タイマのベクタ・アドレスの設定

;*****
;      リモコン信号タイマ処理
;*****
TMO_SEG CSEG
INTTMO:

      SEL RB1
      EI                ; 割り込み許可 ( I N T P 0 )
      if_bit(IPDTFG)    ; 入力信号あり?
      if_bit(RMDTOK)    ; 有効データあり?
      RPTCT--
      if(RPTCT==#0)     ; リピート無効時間
      CLR1  RPT          ; リピート・コード無効状態
      CLR1  IPDTFG
      CLR1  RMDTOK
      endif
      CALL  !S_LOWCT
      else
      RMENDCT--
      if(RMENDCT==#0)
      SET1  RMDTOK        ; 有効データありに設定
      SET1  RMDTSET
      CALL  !S_M0SET      ; リード (ロウ) 検出モードに設定
      endif
      LD_CT=#5
      endif
      else
      CALL  !S_LOWCT
      endif
      RETI

```

```

S_LOWCT:
    if (SELMOD==#0)                                ; リード（ロウ）検出モード？
        if_bit(!P0.0)
            LD_CT--
            if (LD_CT==#0)
                SELMOD=#1                            ; リード（ロウ）計測モード
                TMC0=#00000000B
                CR00=#32767                            ; タイマ 7. 8 1 m s
                TMC0=#00001100B
                INTM0=#00000100B
                CLR1    PIF0
                CLR1    PMK0                            ; I N T P 0 割り込み許可
                LD_CT=#5
            endif
        else
            LD_CT=#5
        endif
    else
        CALL    !S_M0SET                            ; リード（ロウ）検出モードに設定
        LD_CT=#5
    endif
    RET

$EJECT
;*****
;*      リモコン信号エッジ検出処理
;*****
P0_SEG  CSEG
INTP0:

    SEL      RBO
    CALL     !WAIT                                    ; 1 0 0 μ s ウェイト
    switch(SELMOD)
    case 1:
        CALL  !LEAD_L                                ; リード・ロウ検出処理
        break
    case 2:
        CALL  !LEAD_H                                ; リード・ハイ検出処理
        break
    case 3:
        CALL  !CDCODE                                ; カスタム／データ・コード取り込み処理
        break
    case 4:
        CALL  !REPCD                                  ; リピート・コード検出処理
        break
    case 5:
        CALL  !ENDCHK                                  ; 異常データ検出処理
    ends
    RETI

```

```

;*****
;*      リード・ロウ検出
;*****
LEAD_L:

    if_bit(P0.0)                ; レベル・チェック P0.0 = 0 : ノイズ
    CALL    !WAIT              ; 100  $\mu$ s ウェイト
    if_bit(P0.0)
        CALL    !CR_READ        ; タイマ値読み出し
        if(AX)>#3354            ; 6.8ms - (1.5ms * 4)
            if(AX)<#18035        ; 11.8ms - (1.5ms * 5)
                SELMOD=#2        ; リード・ハイ検出モード
                INTMO=#00000000B ; I N T P 0 立ち下がりエッジ
            else
                CALL    !S_M0SET ; リード(ロウ)検出モードに設定
            endif
        else
            CALL    !S_M0SET ; リード(ロウ)検出モードに設定
        endif
    endif
    RET
$EJECT

;*****
;*      リード・ハイ検出
;*****
LEAD_H:

    if_bit(!P0.0)              ; レベル・チェック P0.0 = 1 : ノイズ
    CALL    !WAIT              ; 100  $\mu$ s ウェイト
    if_bit(!P0.0)
        CALL    !CR_READ        ; タイマ値読み出し
        if(AX)>#6710-160/2      ; 1.8ms-100  $\mu$ s*2-160クロック (エッジ検出→タイマ・スタート)
            if(AX)<#20132-160/2  ; 5ms-100  $\mu$ s*2-160クロック (エッジ検出→タイマ・スタート)
                if(AX)>#11743-160/2 ; カスタム/データ・コード (3ms - 100  $\mu$ s * 2) ?
                    SELMOD=#3      ; データ取り込みモード
                    WORKP=#0000H   ; ワーク・エリア・イニシャライズ
                    (WORKP)+2=#8000H ; 最上位ビットを1に設定(データ終了確認用)
                else
                    SELMOD=#4      ; リピート検出モード
                    INTMO=#00000100B ; I N T P 0 立ち上がり
                endif
            else
                CALL    !S_M0SET ; リード(ロウ)検出モードに設定
            endif
        else
            CALL    !S_M0SET ; リード(ロウ)検出モードに設定
        endif
    endif
    RET
$EJECT

```

```

;*****
;* カスタム／データ・コード取り込み
;*****
CDCODE:
    if_bit(!P0.0)                ; レベル・チェック P0.0 = 1 : ノイズ
    CALL !WAIT                  ; 100μs ウェイト
    if_bit(!P0.0)
        CALL !CR_READ           ; タイマ値読み出し
        if (AX>=#1257-190/2)     ; 0.5ms-100μs*2-190クロック (エッジ検出→タイマ・スタート)
            if (AX<#9646-190/2)  ; 2.5ms-100μs*2-190クロック (エッジ検出→タイマ・スタート)
                if (AX>=#6710-190/2) ; 1.8ms-100μs*2-190クロック (エッジ検出→タイマ・スタート)
                    SET1 CY
                else
                    CLR1 CY
                endif
            HL=#WORKP+3          ; ワーク・エリア・アドレスを設定
            C=#4                 ; ワーク・エリア桁数設定
        WKSHT:
            A=[HL]               ; 1ビット・データ格納
            RORC A,1             ; 1ビット・シフト
            [HL]=A
            HL--
            DBNZ C,$WKSHT        ; 全桁シフト終了
            if_bit(CY)           ; 32ビット入力終了?
                if (WORKP+0==#CSTM) (A)
                    ; カスタム・コード・チェック

                    A^=WORKP+1
                    if (A==#0FFH) ; カスタム・コード反転データ・チェック
                        A=WORKP+2
                        A^=WORKP+3 ; データ・コード反転データ・チェック
                        if (A==#0FFH)
                            ; 入力データを格納
                            RMDATA=WORKP+2 (A)
                            ; 入力データありの状態に設定
                            SET1 IPDTFG
                            CLR1 RMDTSET
                            CLR1 RPT
                            CLR1 RMDTOX
                            CALL !S_M5SET
                        else
                            ; リード（ロウ）検出モードに設定
                            CALL !S_M0SET
                        endif
                    else
                        ; リード（ロウ）検出モードに設定
                        CALL !S_M0SET
                    endif
                else
                    CALL !S_M0SET
            endif
        endif
    endif

```

```

        endif
    endif
    else
        CALL    !S_M0SET        ; リード（ロウ）検出モードに設定
    endif
    else
        CALL    !S_M0SET        ; リード（ロウ）検出モードに設定
    endif
    endif
endif
endif
RET
$EJECT
;*****
;*      リピート・コード検出
;*      *****
REPCD:
    if_bit(P0.0)                ; レベル・チェック P0.0 = 0 : ノイズ
        CALL    !WAIT          ; 100μs ウェイト
        if_bit(P0.0)
            if_bit(RMDTOK)      ; 有効データあり?
                CALL    !CR_READ ; タイマ値読み出し
                if (AX<=#3354-190/2) ; 1ms-100μs*2-190クロック（エッジ検出→タイマ・スタート）
                    SET1    RPT
                    CLR1    RMDTOK ; データ終了後の入力信号チェック
                    CLR1    RMDTSET
                    CALL    !S_M5SET
                else
                    CALL    !S_M0SET ; リード（ロウ）検出モードに設定
                endif
            else
                CALL    !S_M0SET ; リード（ロウ）検出モードに設定
            endif
        endif
    endif
endif
RET
$EJECT

```

```

;*****
;*      異常データ検出
;*****
ENDCHK:
    if_bit(!P0.0)                ; レベル・チェック P0.0 = 1 : ノイズ
        CALL    !WAIT           ; 100μs ウエイト
        if_bit(!P0.0)
            CLR1    IPDTEG        ; 異常データ入力
            CLR1    RPT           ; 入力信号無効
            CALL    !S_M0SET      ; リード(ロウ)検出モードに設定
        endif
    endif
    RET

;*****
;*      100μs ウエイト
;*****
WAIT:
    B=#(838-14-12-8)/12          ; CALL(14), RET(12), MOV(8)
WAITCT:
    DBNZ    B,$WAITCT           ; 100μs の設定
    RET                          ; 1命令12クロック

;*****
;*      リード(ロウ)検出モードの設定
;*****
S_M0SET:
    TMC0=#00000000B
    CR00=#6290
    TCL0=#00100000B            ; タイマ1.5ms に設定
    TMC0=#00001100B
    SELMOD=#0                  ; リード(ロウ)検出モード
    SET1    PMK0
    RET

;*****
;*      異常データ検出モードの設定
;*****
S_M5SET:
    RPTCT=#173                ; 250ms 測定用カウンタ
    SELMOD=#5                  ; データ入力終了モード
    RMENDCT=#3                ; 無入力確認用カウンタ
    TMC0=#00000000B          ; 動作停止
    CR00=#6290                ; 1.5ms に設定
    TMC0=#00001100B
    RET

;*****
;*      タイマ・カウント値の読み出し
;*****
CR_READ:
    AX=CR01
    TMC0=#00000000B          ; 動作停止
    TMC0=#00001100B          ; タイマ・スタート
    RET

```

5.3.2 PWM出力とフリーランニングによるリモコン受信（ μ PD78018F, 78018FY, 78014Hサブシリーズのみ）

このプログラム中で、リモコン信号を受信する場合のパルス有効幅を表5 - 2に示します。また、各信号の処理方法を から に示します。

表5 - 2 入力信号の有効時間

信号名		出力時間	有効時間
リーダー・コード（ロウ）		9 ms	3 ms-10 ms
リーダー・コード （ハイ）	通常	4.5 ms	3 ms-5 ms
	リピート	2.25 ms	1.8 ms-3 ms
カスタム/ データ・コード	0	1.125 ms	0.5 ms-1.8 ms
	1	2.25 ms	1.8 ms-2.5 ms

リーダー・コード（ロウ）

INTP0の立ち下がりエッジ検出時の割り込み要求によって、キャプチャ・レジスタ（CR01）の値をメモリに格納します。

立ち上がりエッジが発生したときのコンペア・レジスタ（CR00）との差からパルス幅を計測します。

リーダー・コード（ハイ）

INTP0の立ち下がりエッジの割り込み要求によって、リーダー・コードのハイ・レベル間のパルス幅をタイマのカウント値により計測します。

カスタム/データ・コード

INTP0の立ち下がりエッジの割り込み要求によって、1ビット（1周期）ごとのパルス幅を計測します。32ビット目のデータ取り込み後に、反転データとカスタム・コードの一致をテストします。さらに、33ビット目にデータがないことを確認します。

リピート・コード検出

リーダー・コードのハイ・レベルが3 ms未満のときに、リーダー・コード出力後からINTP0立ち上がりまでのパルス幅を計測します。

リピート・コードの有効時間

有効なデータ入力後、16ビット・タイマ/イベント・カウンタのオーバーフロー・フラグ（OVF0）をメイン・プログラムでテストし、リピート・コード有効時間250 msを計測します。

パルス幅計測中のタイム・アウト

パルス幅計測中に16ビット・タイマ/イベント・カウンタのOVF0をメイン・プログラムでテストし、2回検出したときはタイム・アウトとしてデータは無効になります。

また、この例では16ビット・タイマ/イベント・カウンタをPWM出力モードで動作させていますので、5.2 PWM出力のプログラムをリンクすることにより、リモコン受信とPWM出力を同時に行うことができます。

(1) パッケージの説明

<パブリック宣言シンボル>

TIM_PRO : タイマ・オーバーフロー処理のサブルーチン名称

RMDATA : リモコン受信データを格納

RPT : リピート有効期間判断フラグ

IPDTFG : 有効データの有無判断フラグ

RMDTOK : 入力信号の有効の有無判断フラグ

RMDTSET : 入力信号の有無判断フラグ

OVSENS : INTP0処理タイマ・オーバーフロー検出フラグ

<使用するレジスタ>

バンク0 : AX, BC, HL

<使用するRAM>

名 称	用 途	属 性	バイト
RPTCT	リピート・コード有効時間カウンタ	SADDR	1
RMENDCT	データ入力後の無入力時間カウンタ		
SELMOD	モード選択		
LD_CT	リーダ信号検出カウンタ		
RMDATA	有効データ格納エリア		
TO_CNT	タイマ・オーバーフロー検出カウンタ		
CR01_NP	最新タイマ・カウント値格納エリア	SADDRP	2
CR01_OP	前回タイマ・カウント値格納エリア		
WORKP	入力信号格納エリア		4

<使用するフラグ>

名 称	用 途
IPDTFG	有効データの有無
RMDTOK	有効な入力信号の有無
RMDTSET	入力信号の有無
RPT	リピート有効期間が経過したかどうかの判断
TO_FLG	タイマ・オーバーフローあり
OVSENS	INTP0処理でタイマ・オーバーフロー検出

<ネスティング>

5 レベル11バイト

<使用するハードウェア>

- 16ビット・タイマ/イベント・カウンタ
- P00/INTP0
- P30/TO0

<初期設定>

- 16ビット・タイマ/イベント・カウンタの設定
 - PWM出力モード TMC0=#00000010B
 - PWM基本周期61.0 μ s TCL0=#00100000B
 - ロウ・アクティブ出力 TOC0=#00000011B
- P30出力モード PM30=0
- INTP0サンプリング・クロック $f_x/2^7$ SCS=#00000011B
- INTP0高優先順位割り込み PPR0=0
- INTP0割り込み許可 PMK0=0
- カスタム・コードをCSTMに定義し、パブリック宣言
- RAMクリア

<起動方法>

- 16ビット・タイマ/イベント・カウンタのOVF0をテストし、OVF0のセットのときは、サブルーチン TIM_PROをコールしてください。
- リモコン信号のエッジ検出による割り込み要求で起動

(2) 使用例

```

PUBLIC  CSTM
EXTRN  RMDATA, RPTCT, PWM, PWMOUT, TIM_PRO
EXTBIT  RPT, RMDTSET, IPDTFG, TO_FLG, OVSENS
CSTM    EQU      9DH                      ; カスタム・コード

TOC0=#00000011B                          ; PWM出力, ロウ・アクティブの設定
TCL0=#00100000B                          ; カウント・クロック  $f_x / 2$  を選択
TMC0=#00000010B                          ; PWMモード, オーバフローあり
INTM0=#00000000B                         ; INT P 0 立ち下がりエッジ
SCS=#00000011B                          ; INT P 0 サンプルング・クロック  $f_x / 128$ 

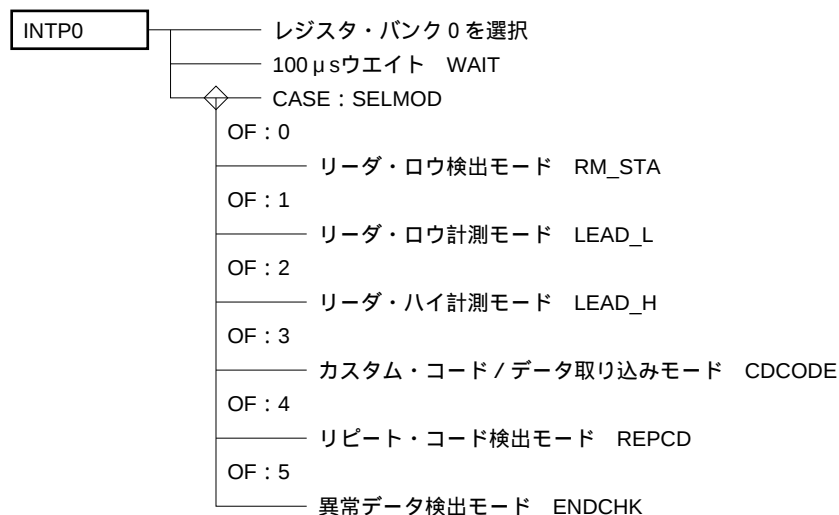
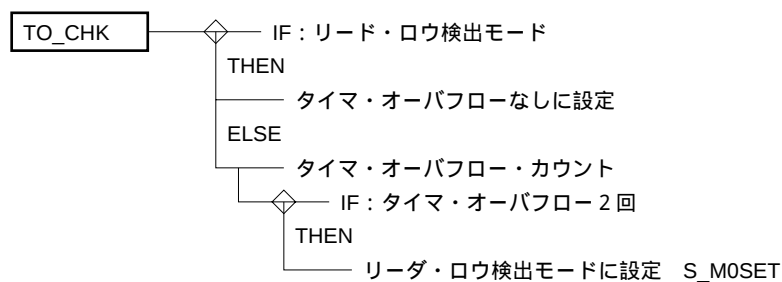
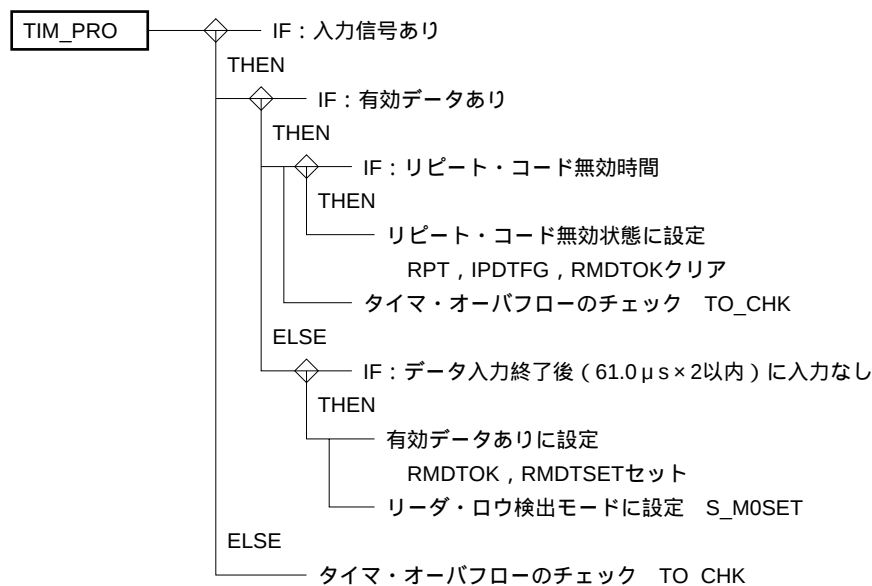
CLR1    PPR0                            ; INT P 0 高優先
CLR1    RPT                             ; フラグ・クリア
CLR1    IPDTFG
CLR1    RMDTSET

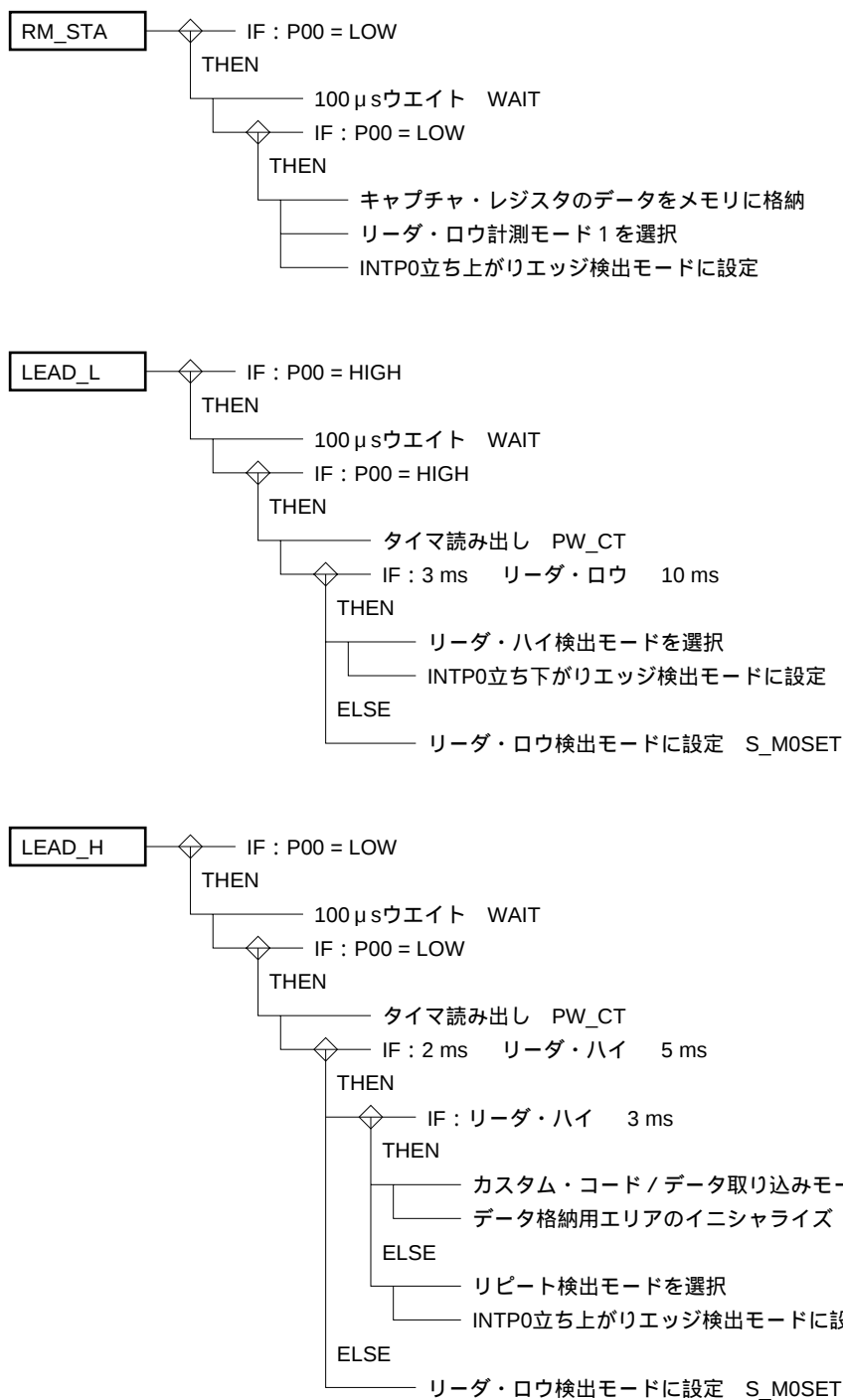
CLR1    PMK0                            ; INT P 0 割り込み許可
EI

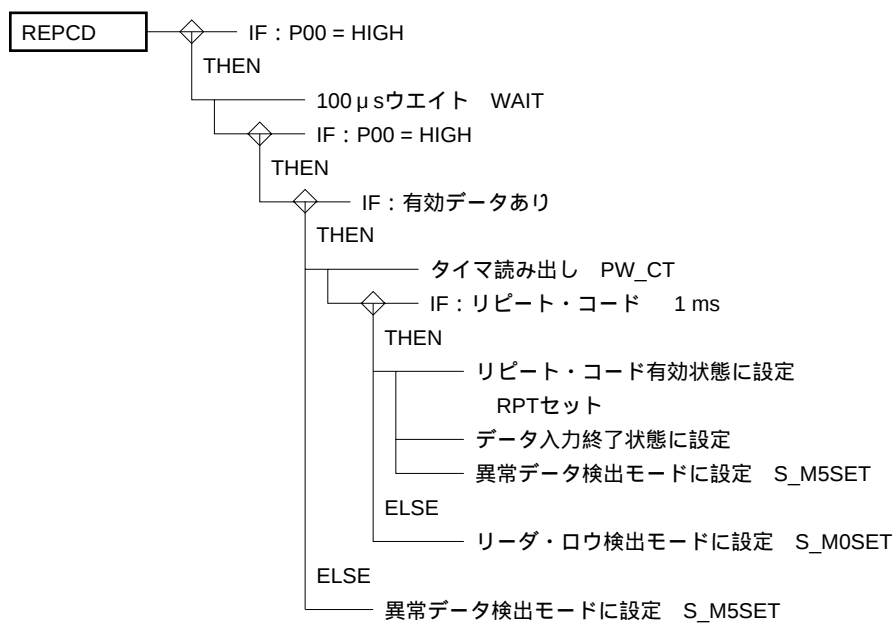
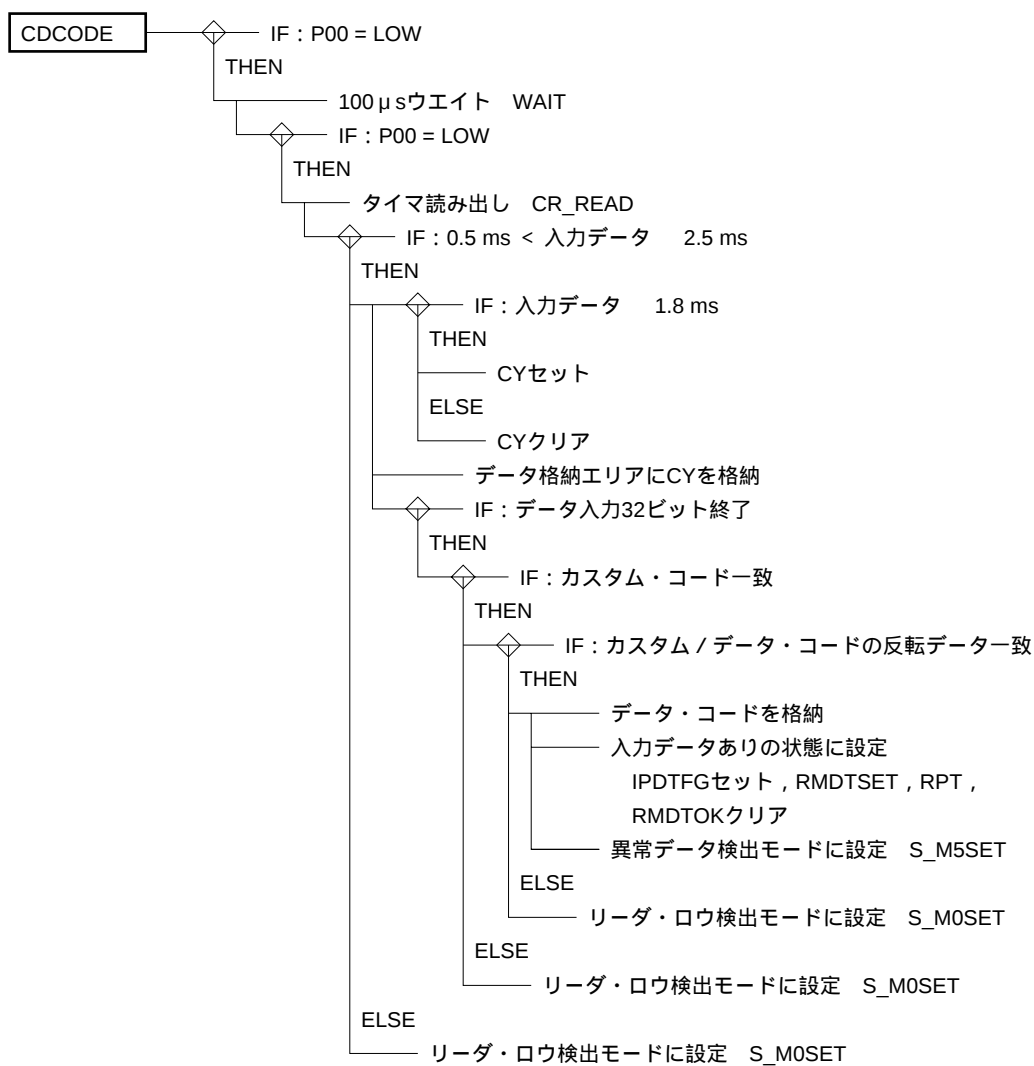
DT_TEST:
  if_bit(OVSENS)                        ; INT P 0 処理でタイマ・オーバフロー検出
    CLR1    OVSENS
    CALL    !TIM_PRO
  elseif_bit(OVF0)                      ; タイマ・オーバフローあり
    CLR1    OVFO
    SET1    TO_FLG
    CALL    !TIM_PRO
  endif
  if_bit(RMDTSET)
    CLR1    RMDTSET
    if_bit(RPT)
;
;      リピート処理
;
  else
;
;      入力ありの処理
;
  endif
else
  if_bit(!RPT)
;
;      入力なしの処理
;
  endif
endif
MOV      PWMOUT, A
CALL     !PWM

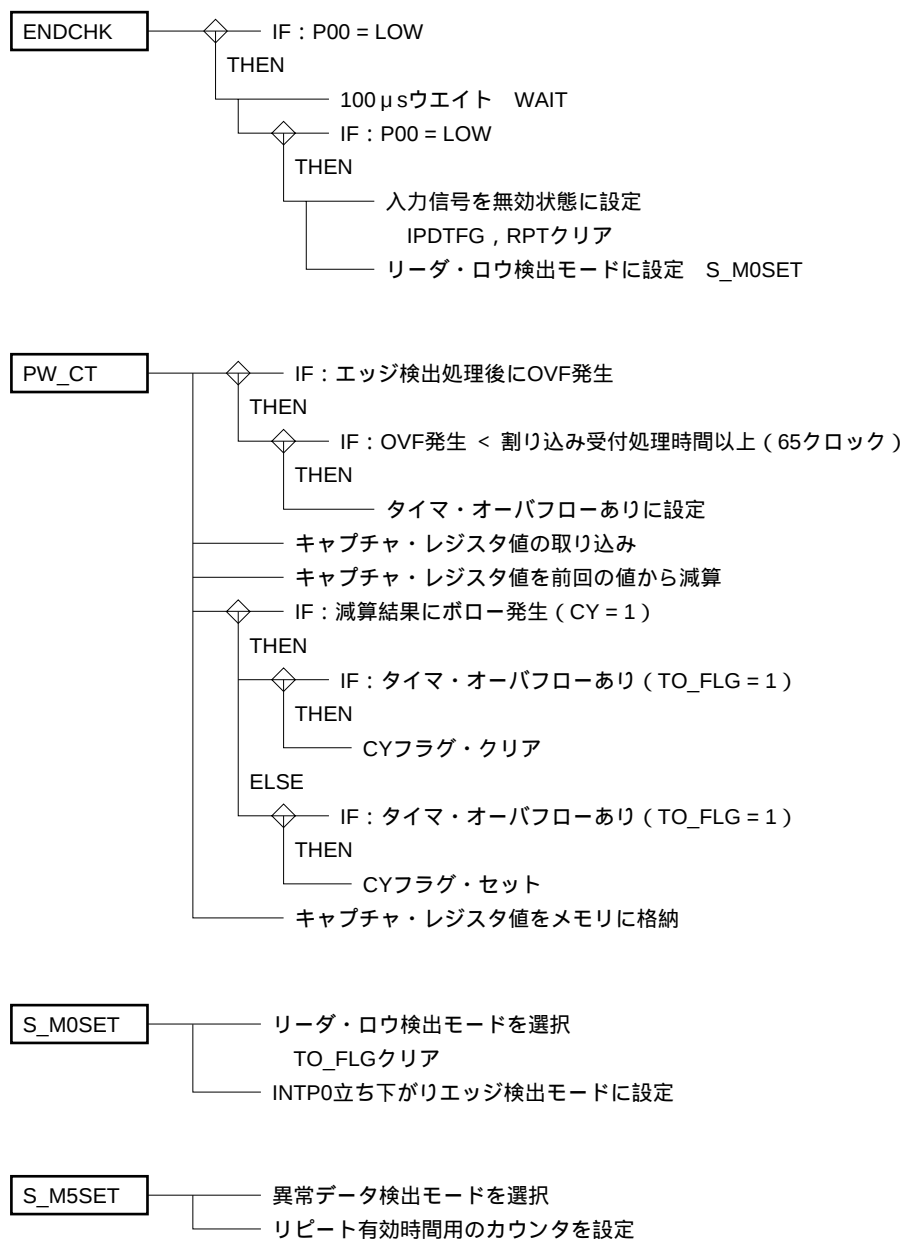
```

(3) SPDチャート









(4) プログラム・リスト

```

PUBLIC  TIM_PRO, RPT, IPDTFG, RMDTOK, RMDTSET
PUBLIC  RMENDCT, RPTCT, SELMOD, LD_CT, RMDATA
PUBLIC  TO_FLG, OVSENS
EXTRN  CSTM

RM_DAT  DSEG    SADDR
RPTCT:  DS      1                      ; リピート・コード有効時間カウンタ
RMENDCT:DS      1                      ; データ入力後の無入力時間カウンタ
SELMOD: DS      1                      ; モード選択
LD_CT:  DS      1                      ; リード信号検出カウンタ
RMDATA: DS      1                      ; 有効データ格納エリア
TO_CNT: DS      1                      ; タイマ・オーバーフロー・カウンタ

RM_DATP DSEG    SADDRP
CR01_NP:DS      2                      ; 最新タイマ・カウンタ値格納エリア
CR01_OP:DS      2                      ; 前回タイマ・カウンタ値格納エリア
WORKP:  DS      4                      ; 入力信号格納エリア

        BSEG
IPDTFG  DBIT                      ; 有効データあり
RMDTOK  DBIT                      ; 入力信号有効
RMDTSET DBIT                      ; 入力信号あり
RPT      DBIT                      ; リピート・コード有効期間
TO_FLG  DBIT                      ; タイマ・オーバーフローあり
OVSENS  DBIT                      ; I N T P 0 処理でタイマ・オーバーフロー検出

VEP0    CSEG    AT 06H
        DW      INTPO                      ; I N T P 0 のベクタ・アドレスの設定

$EJECT
;*****
;      リモコン信号タイマ処理
;*****
TMO_SEG CSEG
TIM_PRO:
    if_bit(IPDTFG)                  ; 入力信号あり?
        if_bit(RMDTOK)              ; 有効データあり?
            RPTCT--
            if(RPTCT==#0)            ; リピート無効時間
                CLR1    RPT          ; リピート・コード無効状態
                CLR1    IPDTFG
                CLR1    RMDTOK
            endif
        else
            RMENDCT--
            if(RMENDCT==#0)
                SET1    RMDTOK        ; 有効データありに設定
                SET1    RMDTSET
                CALL    !S_MOSET      ; リード(ロウ)検出モードに設定
            endif
        endif
    else
        CALL    !TO_CHK              ; タイマ・オーバーフローのチェック
    endif
RET

```

```

TO_CHK:
    if (SELMOD==#0)
        CLR1    TO_FLG
    else
        TO_CNT++
        if (TO_CNT==#2)
            CALL    !S_MOSSET           ; スタート・エッジ検出モードに設定
        endif
    endif
    RET
$EJECT
;*****
;*      リモコン信号エッジ検出処理
;*****
PO_SEG  CSEG
INTP0:
    SEL RBO

    CALL    !WAIT                     ; 100μs ウェイト

    switch(SELMOD)

    case 0:
        CALL    !RM_STA               ; スタート・エッジ検出処理
        break
    case 1:
        CALL    !LEAD_L               ; リード・ロウ検出処理
        break
    case 2:
        CALL    !LEAD_H               ; リード・ハイ検出処理
        break
    case 3:
        CALL    !CDCODE                ; カスタム／データ・コード取り込み処理
        break
    case 4:
        CALL    !REPCD                ; リピート・コード検出処理
        break
    case 5:
        CALL    !ENDCHK               ; 異常データ検出処理
    ends
    RETI

;*****
;*      スタート・エッジ検出
;*****
RM_STA:
    CLR1    TO_FLG                   ; タイマ・カウント開始
    if_bit(!P0.0)                   ; レベル・チェック P0.0 = 1 : ノイズ
        CALL    !WAIT               ; 100μs ウェイト
        if_bit(!P0.0)
            CR01_OP=CR01 (AX)       ; キャプチャ・レジスタを格納
            SELMOD=#1               ; リード・ロウ検出モード
            INTMO=#00000100B        ; INTP0 立ち上がりエッジ
            TO_CNT=#0
        endif
    endif
    RET

```

```

;*****
;*      リード・ロウ検出
;*****
LEAD_L:
    if_bit(P0.0)                ; レベル・チェック P0.0 = 1 : ノイズ
    CALL    !WAIT              ; 100  $\mu$ s ウェイト
    if_bit(P0.0)
    CALL    !PW_CT             ; タイマ値読み出し
    if_bit(!CY)
    TO_CNT=#0
    if (AX>=#12582)             ; 3 ms
        if (AX<#41942)         ; 10 ms
            SELMOD=#2          ; リード・ハイ検出モード
            INTMO=#00000000B   ; INT P0 立ち下がりエッジ
        else
            CALL    !S_M0SET    ; スタート・エッジ検出モードに設定
        endif
    else
        CALL    !S_M0SET    ; スタート・エッジ検出モードに設定
    endif
    else
        CALL    !S_M0SET    ; スタート・エッジ検出モードに設定
    endif
    endif
    RET
$EJECT
;*****
;*      リード・ハイ検出
;*****
LEAD_H:
    if_bit(!P0.0)              ; レベル・チェック P0.0 = 0 : ノイズ
    CALL    !WAIT              ; 100  $\mu$ s ウェイト
    if_bit(!P0.0)
    CALL    !PW_CT             ; タイマ値読み出し
    if_bit(!CY)
    TO_CNT=#0
    if (AX>=#7549)             ; 1.8 ms
        if (AX<#20971)        ; 5 ms
            if (AX>#12582)     ; カスタム/データ・コード (3 ms) ?
                SELMOD=#3     ; データ取り込みモード
                WORKP=#0000H   ; ワーク・エリア・イニシャライズ
                (WORKP)+2=#8000H ; 最上位ビットを1に設定 (データ終了確認用)
            else
                SELMOD=#4     ; リピート検出モード
                INTMO=#00000100B ; INT P0 立ち上がり
            endif
        else
            CALL    !S_M0SET    ; スタート・エッジ検出モードに設定
        endif
    else
        CALL    !S_M0SET    ; スタート・エッジ検出モードに設定
    endif
    else
        CALL    !S_M0SET    ; スタート・エッジ検出モードに設定
    endif
    endif
    RET
$EJECT

```

```

;*****
;* カスタム／データ・コード取り込み
;*****
CDCODE:
    if_bit(!P0.0)                ; レベル・チェック P0.0 = 1 : ノイズ
    CALL    !WAIT                ; 100μs ウェイト
    if_bit(!P0.0)
    CALL    !PW_CT                ; タイマ値読み出し
    if_bit(!CY)
    TO_CNT=#0
    if (AX>=#2096)                ; 0.5ms
    if (AX<#10485)                ; 2.5ms
    if (AX>=#7549)                ; 1.8ms
    SET1    CY
    else
    CLR1    CY
    endif

    HL=#WORKP+3                ; ワーク・エリア・アドレスを設定
    C=#4                        ; ワーク・エリア桁数設定
WKSHT:
    A=[HL]                      ; 1ビット・データ格納
    RORC    A,1                ; 1ビット・シフト
    [HL]=A
    HL--
    DBNZ    C,$WKSHT           ; 全桁シフト終了

    if_bit(CY)                  ; 32ビット入力終了?
                                ; カスタム・コード・チェック
    if (WORKP+0==#CSTM) (A)
    A^=WORKP+1
    if (A==#0FFH); カスタム・コード反転データ・チェック
    A=WORKP+2
                                ; データ・コード反転データ・チェック
    A^=WORKP+3
    if (A==#0FFH)
    ; 入力データを格納
    RMDATA=WORKP+2 (A)
    ; 入力データありの状態に設定
    SET1    IPDTFG
    CLR1    RMDTSET
    CLR1    RPT
    CLR1    RMDTOK
    CALL    !S_M5SET
    else
    ; スタート・エッジ検出モードに設定
    CALL    !S_M0SET
    endif
    else
    ; スタート・エッジ検出モードに設定
    CALL    !S_M0SET
    endif
    else
    CALL    !S_M0SET
    endif
    endif
    else
    CALL    !S_M0SET           ; スタート・エッジ検出モードに設定
    endif
    else

```

```

        CALL    !S_M0SET      ; スタート・エッジ検出モードに設定
    endif
else
    CALL    !S_M0SET      ; スタート・エッジ検出モードに設定
endif
endif
endif
RET
$EJECT

;*****
;*      リピート・コード検出
;*****
REPCD:
    if_bit(P0.0)           ; レベル・チェック P0.0 = 1 : ノイズ
    CALL    !WAIT          ; 100  $\mu$ s ウェイト
    if_bit(P0.0)
        if_bit(RMDTOK)      ; 有効データあり?
        CALL    !PW_CT      ; タイマ値読み出し
        if_bit(!CY)
            TO_CNT=#0
            if (AX<=#4193)    ; 1 m s
                SET1    RPT
                CLR1    RMDTOK ; データ終了後の入力信号チェック
                CLR1    RMDTSET
                CALL    !S_M5SET
            else
                CALL    !S_M0SET ; スタート・エッジ検出モードに設定
            endif
        else
            CALL    !S_M0SET ; スタート・エッジ検出モードに設定
        endif
    else
        CALL    !S_M0SET ; スタート・エッジ検出モードに設定
    endif
endif
endif
RET
$EJECT

```

```

;*****
;*      異常データ検出
;*****
ENDCHK:
    if_bit(!P0.0)                ; レベル・チェック P0.0 = 1 : ノイズ
    CALL    !WAIT                ; 100μs ウェイト
    if_bit(!P0.0)
        CLR1    IPDTFG           ; 異常データ入力
        CLR1    RPT              ; 入力信号無効
        CALL    !S_MOSET         ; スタート・エッジ検出モードに設定
    endif
endif
RET

;*****
;*      キャプチャ・レジスタ値の算出
;*****
PW_CT:
    if_bit(OVF0)                ; エッジ検出後のOVF0?
    if(CR01<#10000-33) (AX)     ; 割り込み受け付け処理時間=65クロック(MAX)
        CLR1    OVF0
        SET1    OVSENS
        SET1    TO_FLG
    endif
endif

CR01_NP=CR01 (AX)                ; キャプチャ・レジスタ値の取り込み

A=CR01_NP+0                      ; AX=CR01_NP-CR01_OP
A-=CR01_OP
X=A
A=CR01_NP+1
SUBC    A,CR01_OP+1

BC=AX                            ; 演算結果退避
if_bit(CY)                       ; CR01_NP>CR01_OP
    if_bit(TO_FLG)               ; タイマ・オーバフローあり(フラグ・テスト)
        CLR1    CY              ; 正常データ
    endif
else
    if_bit(TO_FLG)               ; タイマ・オーバフロー
        SET1    CY              ; エラー発生
    endif
endif

CR01_OP=CR01_NP (AX)
AX=BC                            ; 演算結果復帰
CLR1    TO_FLG
RET

```

```

;*****
;*      100μs ウェイト
;*****
WAIT:
    B=#(838-14-12-8)/12                ; CALL (14), RET (12), MOV (8)
WAITCT:                                ; 100μs の設定
    DBNZ    B,$WAITCT                  ; 1 命令 12 クロック
    RET

;*****
;*      スタート・エッジ検出モードの設定
;*****
S_M0SET:
    TO_CNT=#0
    SELMOD=#0                          ; スタート・エッジ検出モード
    INTM0=#00000000B                  ; INT P0 立ち下がりがりエッジ
    RET

;*****
;*      異常データ検出モードの設定
;*****
S_M5SET:
    RPTCT=#16                          ; 250ms 測定用カウンタ
    SELMOD=#5                          ; データ入力終了モード
    RMENDCT=#2                         ; 無入力確認用カウンタ
    RET

```

第6章 8ビット・タイマ/イベント・カウンタの応用

78K/0シリーズの8ビット・タイマ/イベント・カウンタには、次の機能があります。

- ・ インターバル・タイマ
- ・ 外部イベント・カウンタ
- ・ 方形波出力
- ・ PWM出力 (μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズのみ)

また、8ビット・タイマ/イベント・カウンタは2チャンネルまたは3チャンネル内蔵していますが、カスケード接続をすることにより16ビット・タイマ/イベント・カウンタとしても使用することができます。

8ビット・タイマ/イベント・カウンタは次のレジスタにより設定します。

< μ PD78018F, 78018FY, 78014Hサブシリーズ >

- ・ タイマ・クロック選択レジスタ1 (TCL1)
- ・ 8ビット・タイマ・モード・コントロール・レジスタ (TMC1)
- ・ 8ビット・タイマ出力コントロール・レジスタ (TOC1)
- ・ ポート・モード・レジスタ3 (PM3)
- ・ ポート3 (P3)

< μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズ >

- ・ タイマ・クロック選択レジスタ50, 51 (TCL50, TCL51)
- ・ 8ビット・タイマ・モード・コントロール・レジスタ50, 51 (TMC50, TMC51)
- ・ ポート・モード・レジスタ7 (PM7)

注意 μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズに内蔵されているレジスタ・フォーマットは、この章のプログラム例で使用されているレジスタのフォーマットとは異なります。したがって、 μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズのいずれかを用いて、この章で記載されているプログラム例を使用する場合は、各レジスタの設定を各製品のレジスタの設定に置き換えて使用してください。

図6 - 2 タイマ・クロック選択レジスタ50のフォーマット
(μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズ)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
TCL50	0	0	0	0	0	TCL502	TCL501	TCL500	FF71H	00H	R/W

TCL502	TCL501	TCL500	カウント・クロックの選択
0	0	0	TI50の立ち下がりエッジ
0	0	1	TI50の立ち上がりエッジ
0	1	0	f_x (8.38 MHz)
0	1	1	$f_x/2^2$ (2.09 MHz)
1	0	0	$f_x/2^4$ (523 kHz)
1	0	1	$f_x/2^6$ (131 kHz)
1	1	0	$f_x/2^8$ (32.7 kHz)
1	1	1	$f_x/2^{10}$ (8.18 kHz)

注意1 . TCL50を同一データ以外に書き換える場合は、いったんタイマ動作を停止させてから行ってください。

2 . ビット3 - 7には必ず“ 0 ”を設定してください。

★ 備考1 . カスケード接続時、TCL5n0-TCL5n2 ($n = 0, 1$) の設定は最下位タイマでのみ有効です。

2 . f_x : メイン・システム・クロック発振周波数

3 . () 内は、 $f_x = 8.38$ MHz動作時。

図6-3 タイマ・クロック選択レジスタ51のフォーマット
(μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズ)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
TCL51	0	0	0	0	0	TCL512	TCL511	TCL510	FF79H	00H	R/W

TCL512	TCL511	TCL510	カウント・クロックの選択
0	0	0	TI51の立ち下がりエッジ
0	0	1	TI51の立ち上がりエッジ
0	1	0	$f_x/2$ (4.19 MHz)
0	1	1	$f_x/2^3$ (1.04 MHz)
1	0	0	$f_x/2^5$ (261 kHz)
1	0	1	$f_x/2^7$ (65.4 kHz)
1	1	0	$f_x/2^9$ (16.3 kHz)
1	1	1	$f_x/2^{11}$ (4.09 kHz)

注意1．TCL51を同一データ以外に書き換える場合は、いったんタイマ動作を停止させてから行ってください。

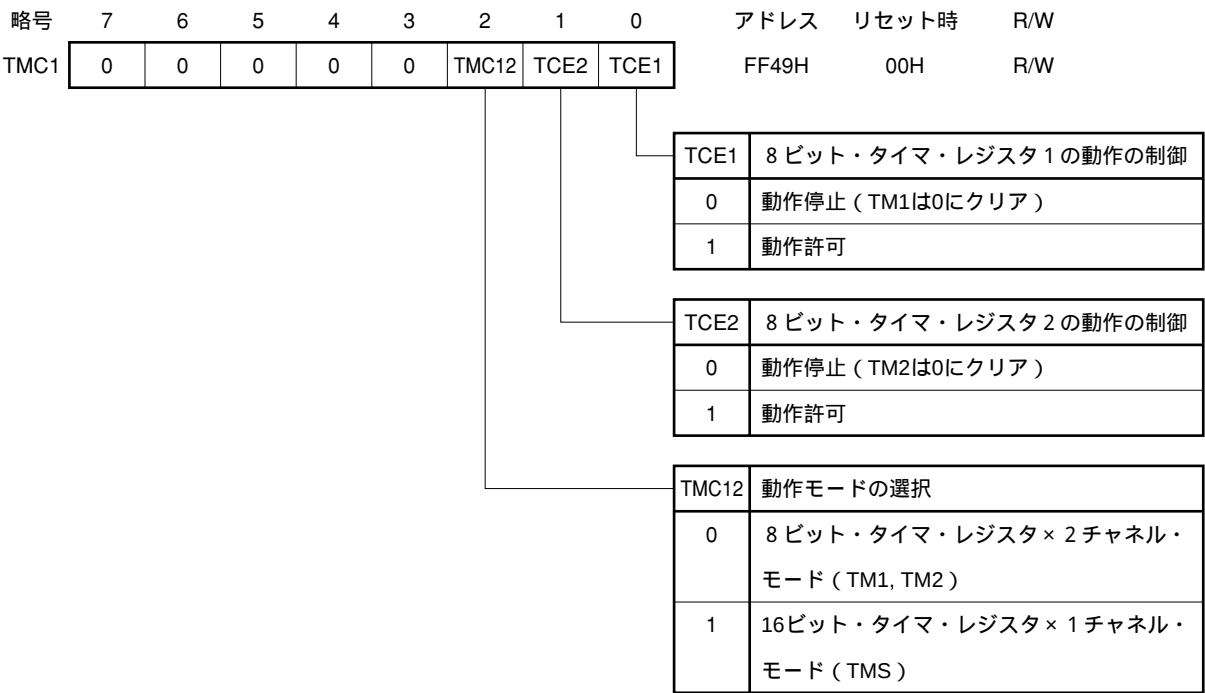
2．ビット3-7には必ず“0”を設定してください。

★ 備考1．カスケード接続時、TCL5n0-TCL5n2 (n = 0, 1) の設定は最下位タイマでのみ有効です。

2． f_x ：メイン・システム・クロック発振周波数

3．()内は、 $f_x = 8.38$ MHz動作時。

図6 - 4 8ビット・タイマ・モード・コントロール・レジスタのフォーマット
(μPD78018F, 78018FY, 78014Hサブシリーズ)



- 注意 1 . 動作モードは , タイマ動作を停止させたのちに切り替えてください。
- 2 . 16ビット・タイマ・レジスタとして使用する場合は動作許可/停止は , TCE1で設定してください。

図6 - 5 8ビット・タイマ・モード・コントロール・レジスタ 5nのフォーマット
(μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズ)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
TMC5n	TCE5n	TMC5n6	0	TMC5n4	LVS5n	LVR5n	TMC5n1	TOE5n	FF70H ^{注1} FF78H ^{注2}	00H	R/W

TOE5n	タイマ出力の制御
0	出力禁止 (ポート・モード)
1	出力許可

TMC5n1	PWMモード以外 (TMC5n6 = 0)	PWMモード (TMC5n6 = 1)
	タイマF/Fの制御	アクティブ・レベルの選択
0	反転動作禁止	ハイ・アクティブ
1	反転動作許可	ロウ・アクティブ

LVS5n	LVR5n	タイマ出力F/Fの状態設定
0	0	変化しない
0	1	タイマ出力F/Fをリセット (0)
1	0	タイマ出力F/Fをセット (1)
1	1	設定禁止

TMC5n4	単体モード / カスケード接続モードの選択
0	単体モード (最下位タイマで使用)
1	カスケード接続モード (下位タイマと接続)

TMC5n6	TM5nの動作モード選択
0	TM5nとCR5nの一致でクリア&スタート・モード
1	PWM (フリーランニング) モード

TCE5n	TM5nのカウント動作制御
0	カウンタを0にクリア後, カウント動作禁止 (プリスケアラ禁止)
1	カウント動作開始

注1 . TMC50のアドレス

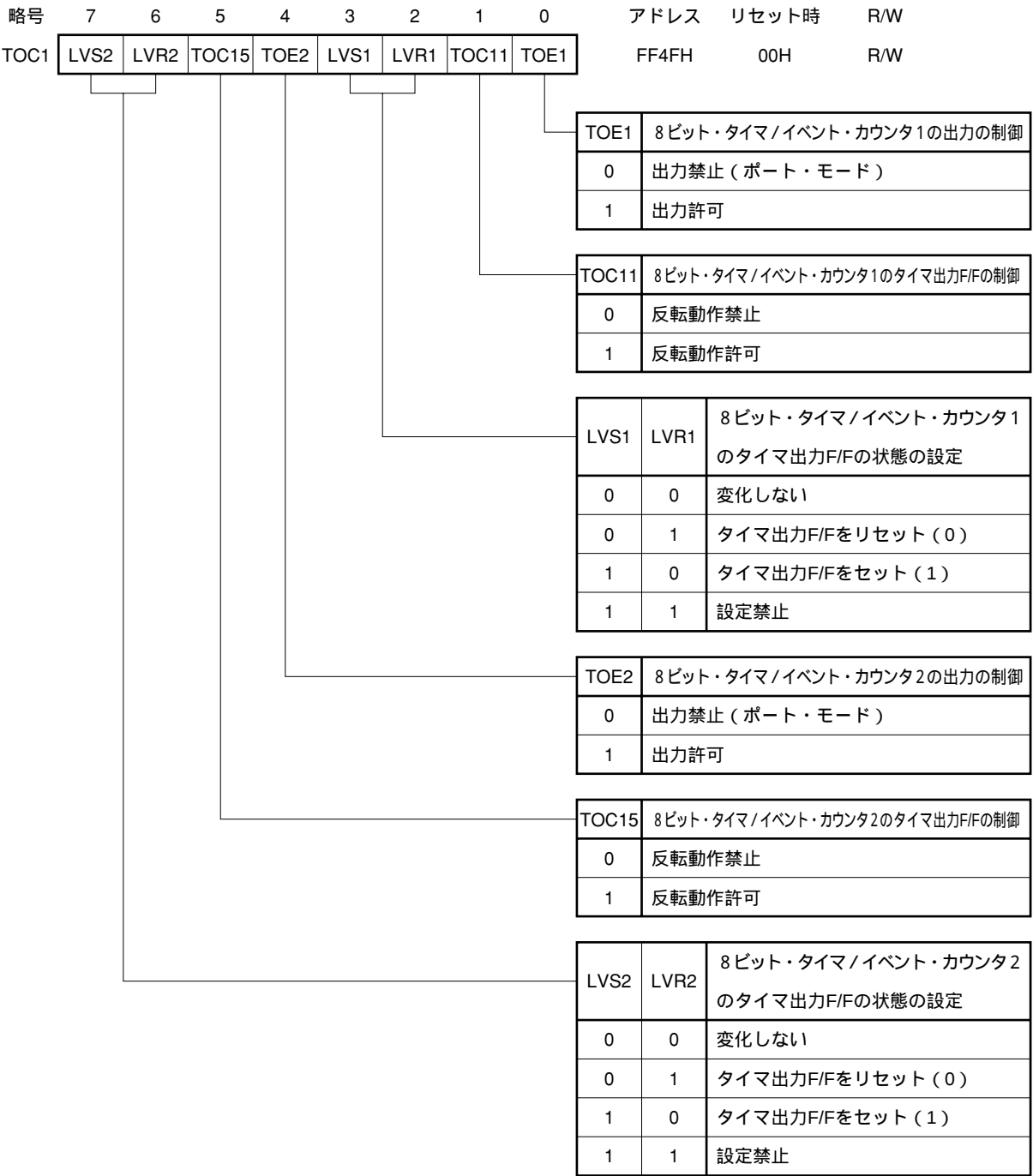
2 . TMC51のアドレス

備考1 . PWMモード時は, TCE5n = 0により, PWM出力はインアクティブ・レベルになります。

2 . データ設定後にLVS5n, LVR5n読み出すと, 0 が読み出せます。

3 . n = 0, 1

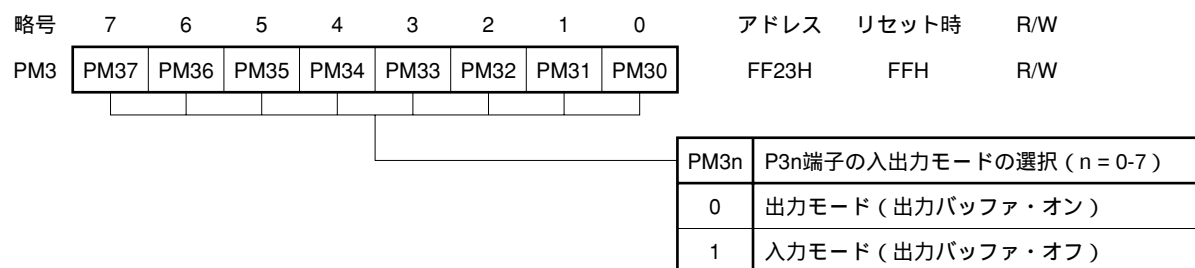
図6 - 6 8ビット・タイマ出力コントロール・レジスタのフォーマット
(μPD78018F, 78018FY, 78014Hサブシリーズ)



- 注意 1 . TOC1は、必ずタイマ動作を停止させたのちに設定してください。
- 2 . LVS1, LVS2, LVR1, LVR2は、データ設定後に読み出すと0が読み出せます。

図6-7 ポート・モード・レジスタ3のフォーマット

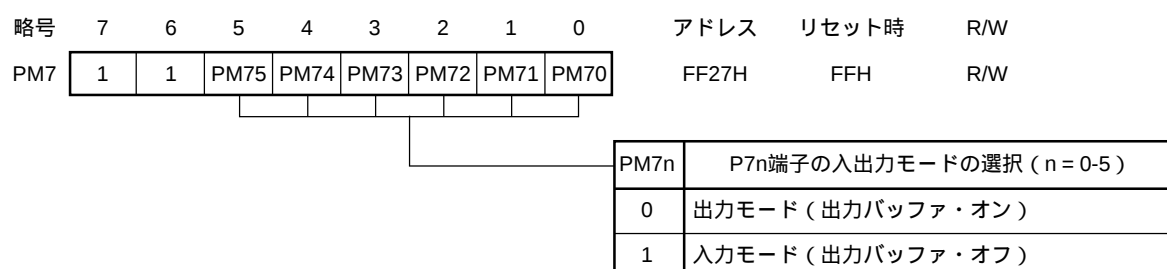
(μPD78018F, 78018FY, 78014Hサブシリーズ)



注意 P31/TO1, P32/TO2端子をタイマ出力として使用するとき, PM31, PM32に0を設定するだけでなく, P31, P32の出力ラッチにも0を設定してください。

図6-8 ポート・モード・レジスタ7のフォーマット

(μPD780024A, 780024AY, 780034A, 780034AYサブシリーズ)



6.1 インターバル・タイマの設定

インターバル・タイマとして使用する場合、8ビット・タイマの動作モードは8ビット・タイマ・モード・コントロール・レジスタ（TMC1）で、インターバル時間はタイマ・クロック選択レジスタ1（TCL1）で設定します。

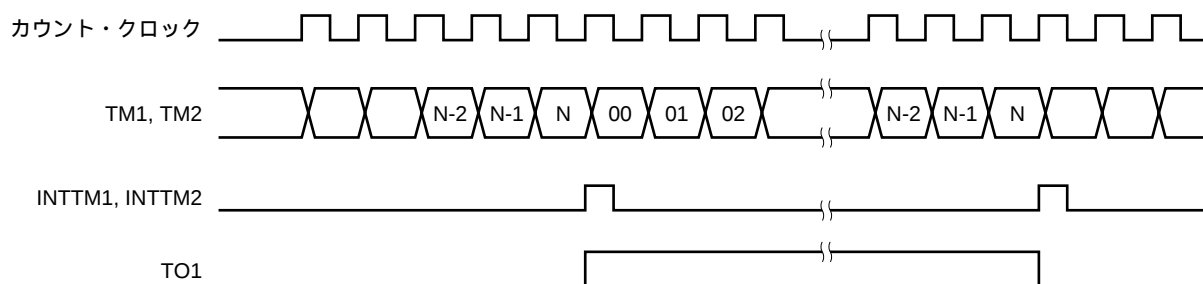
その後、設定時間とカウント・クロックからコンペア・レジスタ（CR10, CR20）の値を設定します。設定時間は、以下の式で決定します。

$$\text{設定時間} = (\text{コンペア・レジスタ値} + 1) \times \text{カウント・クロック周期}$$

設定時間は、8ビット・タイマとして使用する場合でも、16ビット・タイマとして使用する場合でも同様に求めることができます。ただし、16ビット・タイマとして使用するときのカウント・クロックは、TCL1のビット0-ビット3（TCL10-TCL13）で選択した値になります。

次に8ビット・タイマ/16ビット・タイマのそれぞれのモードについて例を示します。

図6 - 9 8ビット・タイマのカウント・タイミング



6.1.1 8ビット・タイマの設定

この例では、μPD78014Hサブシリーズの8ビット・タイマ2を使用し、インターバルの設定時間を500 μsと100 msとした2種類について示します。

(a) 500 μsのインターバルの場合

TMC1の設定

8ビット・タイマ・レジスタ×2チャンネル・モードを選択し8ビット・タイマ2を動作許可

TCL1の設定

500 μs以上の設定が可能で、分解能の最も高い $f_x/2^5$ を選択

CR20の設定

$$500 \mu s = (N + 1) \times \frac{1}{8.38 \text{ MHz}/2^5}$$

$$N = 500 \mu s \times 8.38 \text{ MHz}/2^5 - 1 = 130$$

(1) プログラム・リスト

TCL1 = #10011001B ; カウント・クロックに $f_x/2^5$ を選択

CR20 = #130

TMC1 = #00000010B

(b) 100 msのインターバルの場合

TMC1の設定

8ビット・タイマ・レジスタ×2チャンネル・モードを選択し8ビット・タイマ2を動作許可

TCL1の設定

100 ms以上の設定が可能で、分解能の最も高い $f_x/2^{12}$ を選択

CR20の設定

$$100 \text{ ms} = (N + 1) \times \frac{1}{8.38 \text{ MHz}/2^{12}}$$

$$N = 100 \text{ ms} \times 8.38 \text{ MHz}/2^{12} - 1 = 204$$

(1) プログラム・リスト

TCL1 = #11111111B ; カウント・クロックに $f_x/2^{12}$ を選択

CR20 = #204

TMC1 = #00000010B

6.1.2 16ビット・タイマの設定

この例では、μPD78014Hサブシリーズの8ビット・タイマ1と8ビット・タイマ2をカスケード接続し、インターバル・タイマの設定時間を500 msと10 sとした場合について示します。

(a) 500 msのインターバルの場合

TMC1の設定

16ビット・タイマ・レジスタ×1チャンネル・モードで8ビット・タイマ1, 2を動作許可

TCL1の設定

500 ms以上の設定が可能で、分解能の最も高い $f_x/2^6$ を選択

CR10, CR20の設定

$$500 \text{ ms} = \frac{N + 1}{8.38 \text{ MHz}/2^6}$$

$$N = 500 \text{ ms} \times 8.38 \text{ MHz}/2^6 - 1 \quad 65468 = \text{FF6CH}$$

$$\text{CR10} = 6\text{CH}, \text{CR20} = \text{FFH}$$

(1) プログラム・リスト

TCL1 = #00001010B

CR10 = #06CH ; CR10, CR20に65468を設定

CR20 = #0FFH ; CR10 = 6CH, CR20 = FFH

TMC1 = #00000111B

(b) 10 sのインターバルの場合

TMC1の設定

16ビット・タイマ・レジスタ×1チャンネル・モードで8ビット・タイマ1, 2を動作許可

TCL1の設定

10 s以上の設定が可能で, 分解能の最も高い $f_x/2^{12}$ を選択

CR10, CR20の設定

$$10\text{ s} = \frac{N + 1}{8.38\text{ MHz}/2^{12}}$$

$$N = 10\text{ s} \times 8.38\text{ MHz}/2^{12} - 1 \quad 20458 = 4\text{FEAH}$$

$$\text{CR10} = \text{EAH}, \text{CR20} = 4\text{FH}$$

(1) プログラム・リスト

TCL1 = #00001111B

CR10 = #0EAH ; CR10, CR20に20458を設定

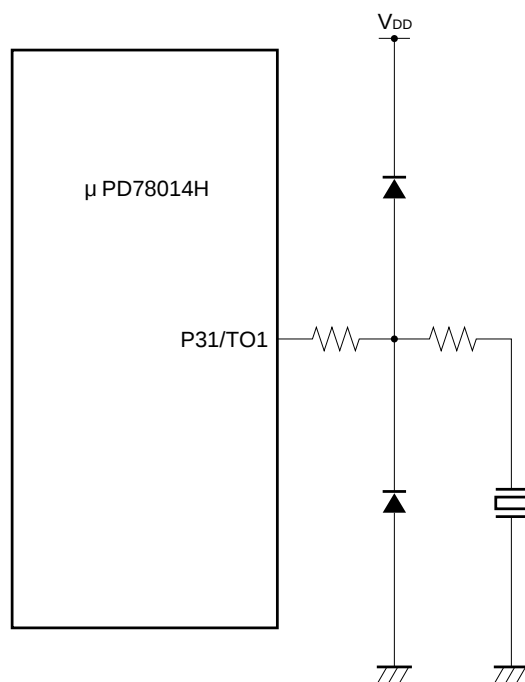
CR20 = #4FH ; CR10 = EAH, CR20 = 4FH

TMC1 = #00000111B

6.2 音階発生

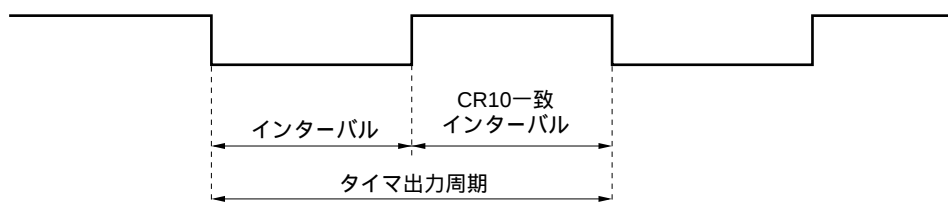
この例では、 μ PD78014Hサブシリーズの8ビット・タイマ/イベント・カウンタ1の方形波出力（P31/TO1）の機能を使用し、外付けのブザーなどにパルスを供給することで音階を発生するプログラムを示します。

図6 - 10 音階発生の回路



P31/TO1端子からの出力周波数は、カウント・クロックとコンペア・レジスタで設定されます。この例では、音階の周波数の中心を523 Hz-1046 Hzの範囲に設定していますので、カウント・クロックは $f_x/2^6$ を選択しています。表6 - 1 に音階とコンペア・レジスタの設定値および出力されるパルスの周波数を示します。ただし、タイマ出力はコンペア・レジスタを2回一致させることで1周期が作られますので、インターバルはタイマ出力周期の半分の時間で設定します。

図6 - 11 タイマ出力とインターバル



また、音の時間長については、8ビット・タイマ/イベント・カウンタ2でインターバル時間を設定し、その割り込み回数をカウントすることで出力時間を決定しています。この例では、8ビット・タイマ/イベント・カウンタ2を20 msに設定しています。

表6 - 1 音階と周波数

音階	音階周波数Hz	コンペア・レジスタ値	出力周波数Hz
ド	523.25	124	524.3
レ	587.33	111	585.1
ミ	659.25	98	662.0
ファ	698.46	93	697.2
ソ	783.98	83	780.2
ラ	880.00	73	885.6
シ	987.77	65	993.0
ド	1046.5	62	1040

このプログラムでのデータ・テーブルのフォーマットを以下に示します。

TABLE :

DB 音階データ1, 音長データ1

DB 音階データ2, 音長データ2

⋮ ⋮

DB 音階データn, 音長データn

DB 0, 0

休符のときは、音階データを0に、データ終了のときは、音長データを0に設定します。

例 音を1秒間出力する場合の8ビット・タイマ/イベント・カウンタ2のカウント数
 カウント数=1s/20 ms = 50 (カウント数のデータとして50を設定します)

このプログラムのデータは、ド、レ、ミ、...、ドを1秒ずつ順番に出力する例を示しています。

(1) パッケージの説明

<パブリック宣言シンボル>

MLDY：音階発生プログラムのサブルーチン名称

<使用するレジスタ>

バンク0：A, B, HL

<使用するRAM>

名 称	用 途	属 性	バイト
POINT	テーブル・データのポインタ値を格納	SADDR	1
LNG	音長データをカウント		

<ネスティング>

1 レベル3 バイト

<使用するハードウェア>

- 8ビット・タイマ/イベント・カウンタ1, 2
- P31/TO1

<初期設定>

- サブルーチンMLDYで設定しています。
- 割り込み許可

<起動方法>

- サブルーチンMLDYをコールしてください。

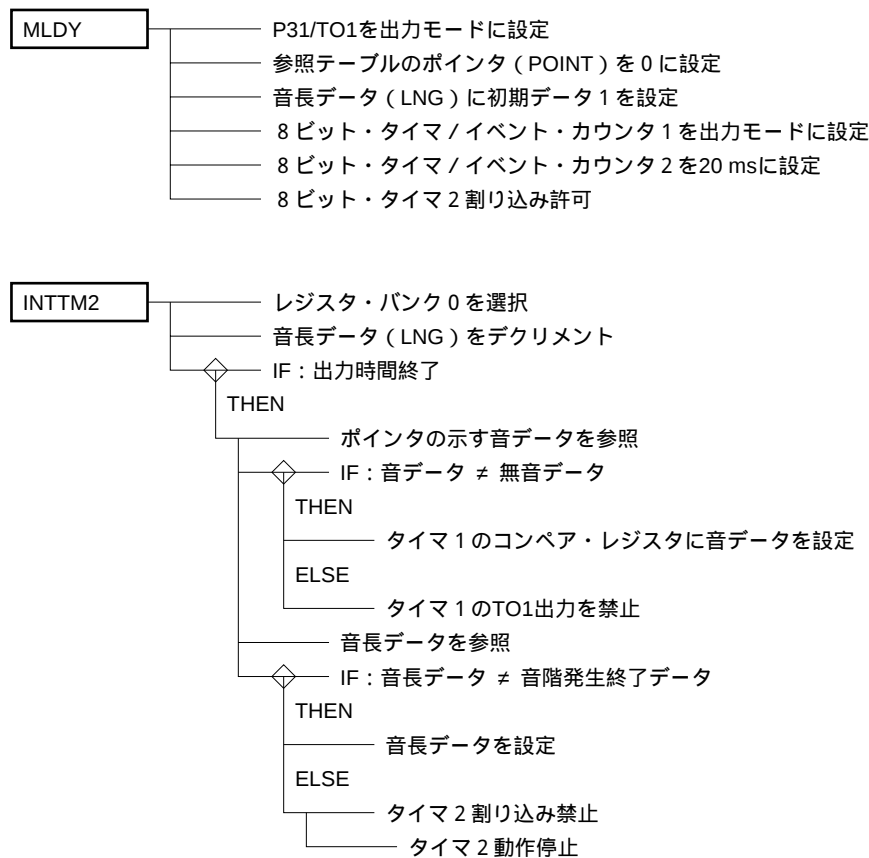
(2) 使用例

```

EXTRN  MLDY
      ⋮
CALL   !MLDY
EI

```

(3) SPDチャート



(4) プログラム・リスト

```

PUBLIC MLDY

VETM2 CSEG AT 18H
      DW INTTM2 ; 8ビット・タイマ/イベント・カウンタ 2 の
                  ; ベクタ・アドレスの設定

ML_DAT DSEG SADDR
POINT: DS 1 ; テーブル・データ用ポインタ
LNG: DS 1 ; 音長データ

;*****
;* 音階発生イニシャライズ
;*****
ML_SEG CSEG
MLDY:
  CLR1 PM3.1 ; P 3 . 1 を出力モード
  POINT=#0 ; ポインタの初期設定
  LNG=#1
  TOC1=#00000011B ; T O 1 出力モードに設定
  TCL1=#11101010B
  CR20=#163 ; タイマ 2 20 ms に設定
  TMC1=#00000010B ; タイマ 2 動作許可
  CLR1 TMMK2 ; タイマ 2 割り込み許可
  RET
$EJECT

```

```

;*****
;*      音階発生データ設定
;*****
TM2_SEG CSEG
INTTM2:
    SEL RBO
    LNG--
    if (LNG==#0)
        B=POINT (A)
        HL=#TABLE                                ; テーブル先頭アドレス設定
        A=[HL+B]
        if (A!=#0)
            CLR1    TCE1                        ; 音データ設定
            CR10=A
            SET1    TOE1
            SET1    TCE1
        else
            CLR1    TOE1
        endif
        B++
        A=[HL+B]                                ; ポインタ・インクリメント
        if (A!=#0)                                ; 音長データ取り込み
            LNG=A                                ; 音出力中?
            B++                                    ; 音長データ設定
            POINT=B (A)
        else
            SET1    TMMK2                        ; タイマ2 割り込み禁止
            CLR1    TCE2                        ; タイマ2 動作停止
        endif
    endif
    RETI
;*****
;*      音階データ・テーブル
;*****
TABLE:
    DB    124, 50                                ; ド
    DB    111, 50                                ; レ
    DB    98, 50                                  ; ミ
    DB    93, 50                                  ; ファ
    DB    83, 50                                  ; ソ
    DB    73, 50                                  ; ラ
    DB    65, 50                                  ; シ
    DB    62, 50                                  ; ド
    DB    00, 00                                ; 終了

```

第7章 時計用タイマの応用

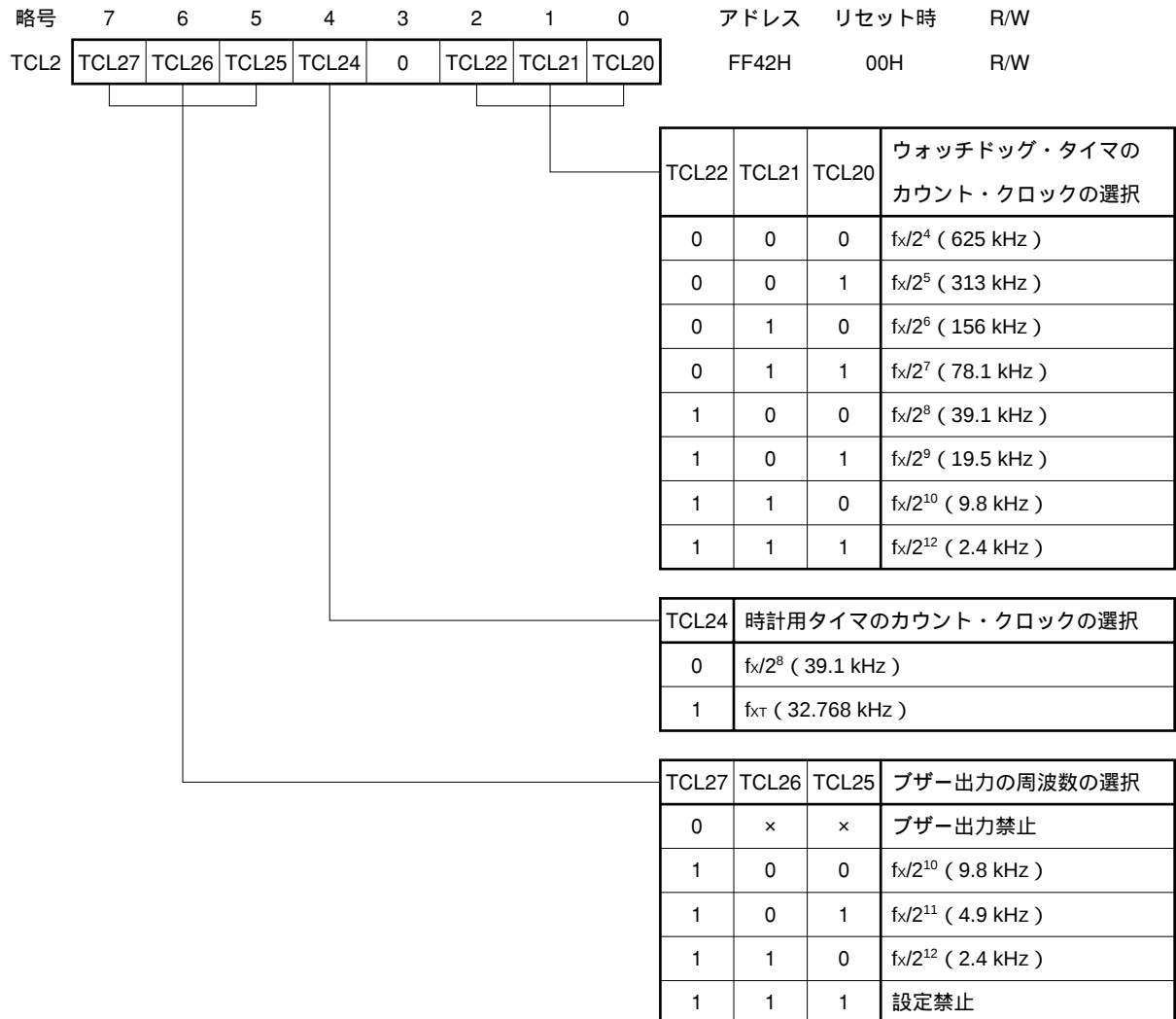
78K/0シリーズの時計用タイマには、メイン・システム・クロックまたはサブシステム・クロックをソースとし0.5秒ごとにオーバーフローする時計用タイマ機能と、6種類の基準時間が設定できるインターバル・タイマ機能があります。この2種類の機能は、同時に使用できます。

時計用タイマは、次のレジスタにより設定します。

- ・タイマ・クロック選択レジスタ2 (TCL2) : μ PD78018F, 78018FY, 78014Hサブシリーズ
- ・時計用タイマ・モード・コントロール・レジスタ (TMC2) : μ PD78018F, 78018FY, 78014Hサブシリーズ
- ・時計用タイマ・モード・コントロール・レジスタ (WTM) : μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズ

注意 μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズに内蔵されているレジスタ・フォーマットは、この章のプログラム例で使用されているレジスタのフォーマットとは異なります。したがって、 μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズのいずれかを用いて、この章で記載されているプログラム例を使用する場合は、各レジスタの設定を各製品のレジスタの設定に置き換えて使用してください。

図7 - 1 タイマ・クロック選択レジスタ2のフォーマット
(μ PD78018F, 78018FY, 78014Hサブシリーズ)



★ 注意 1. TCL2の書き換えは、変更対象の時計用タイマまたはブザーの動作を停止させて行ってください（同一データの上書きでは停止不要）。

停止方法は次のとおりです。

- ・ブザー出力 : TCL2のビット7 (TCL27) に0を入力
- ・時計用タイマ : 時計用タイマ・モード・コントロール・レジスタ (TMC2) のビット2 (TMC22) に0を入力

2. ウォッチドッグ・タイマ動作開始後、カウント・クロックの変更 (TCL20-TCL22の書き換え) は禁止です。

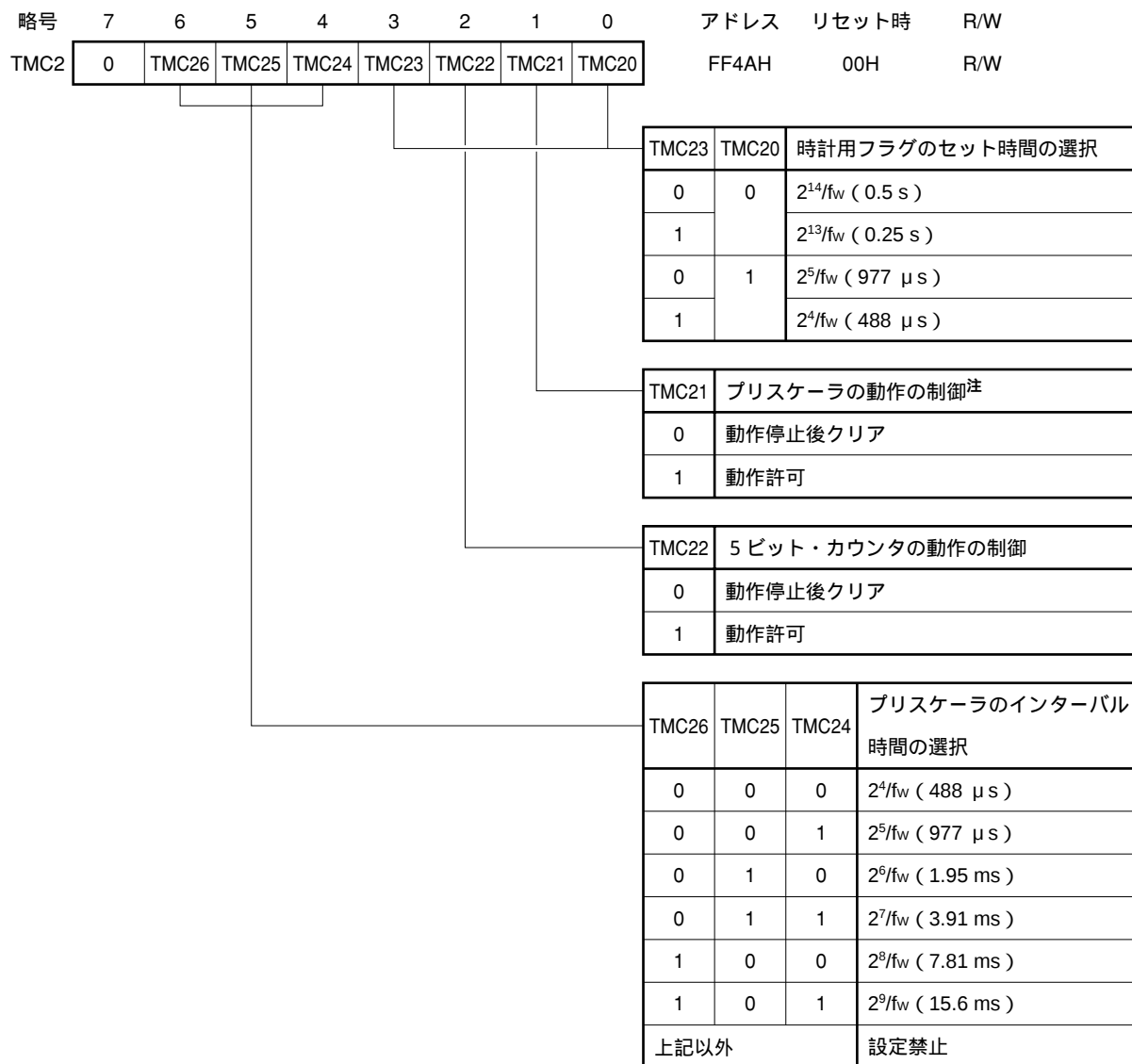
備考 1. f_x : メイン・システム・クロック発振周波数

2. f_{XT} : サブシステム・クロック発振周波数

3. × : don't care

4. () 内は、 $f_x = 10.0$ MHz, $f_{XT} = 32.768$ kHz動作時。

図7-2 時計用タイマ・モード・コントロール・レジスタのフォーマット
(μ PD78018F, 78018FY, 78014Hサブシリーズ)



注意 時計用タイマを使用するときは、ひんばんにプリスケアラをクリアしないでください。

備考1 f_w : 時計用タイマ・クロック周波数 ($f_x/2^8$ または f_{XT})

2. () 内は、 $f_w = 32.768$ kHz動作時。

図7-3 時計用タイマ・モード・コントロール・レジスタのフォーマット
(μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズ)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
WTM	WTM7	WTM6	WTM5	WTM4	0	0	WTM1	WTM0	FF41H	00H	R/W

WTM0	時計用タイマの動作許可
0	動作停止 (プリスケアラ, タイマともにクリア)
1	動作許可

WTM1	5ビット・カウンタの動作制御
0	動作停止後クリア
1	スタート

WTM6	WTM5	WTM4	プリスケアラのインターバル時間の選択
0	0	0	$2^4/f_w$ (488 μ s)
0	0	1	$2^5/f_w$ (977 μ s)
0	1	0	$2^6/f_w$ (1.95 ms)
0	1	1	$2^7/f_w$ (3.91 ms)
1	0	0	$2^8/f_w$ (7.81 ms)
1	0	1	$2^9/f_w$ (15.6 ms)
上記以外			設定禁止

WTM7	時計用タイマのカウント・クロック選択
0	$f_x/2^7$ (65.4 kHz)
1	f_{XT} (32.768 kHz)

備考1. f_w : 時計用タイマ・クロック周波数 ($f_x/2^7$ または f_{XT})

2. f_x : メイン・システム・クロック発振周波数

3. f_{XT} : サブシステム・クロック発振周波数

4. () 内は, $f_x = 8.38$ MHz, $f_w = 32.768$ kHz動作時。

7.1 時計とLED表示プログラム

μPD78014Hサブシリーズの時計用タイマを使った例として、0.5秒のオーバーフローを使用した時刻カウントと、1.95 msのインターバルを使用したLEDダイナミック表示を示します。

時刻カウントはサブルーチンがコールされるごとにオーバーフロー・フラグをテストし、セットされているときは秒のカウント・アップ処理を行います。オーバーフローは0.5秒で発生しますので、120回カウントされたときに1分のカウント・アップを行います。オーバーフローのテストは、取りこぼしのないように1.95 msのインターバルでテストしています。このプログラムの時計表示は、24時間表示で行います。分データ、時データは、上位下位桁を別々にメモリに持つようにします。

図7 - 4 時計データの概念図

秒データ	分データ		時データ	
0-120	下位 0-9	上位 0-5	下位 0-9	上位 0-2

LEDダイナミック表示は、1.95 msごとのインターバルに表示桁を切り替える4桁表示を行います。この例では、ディジット信号にP3の上位4ビットを、セグメント信号にLEDが直接ドライブできるP5を選択しています。

LED表示は、LED表示エリア（LEDDP）の表示桁エリア（DIGCT）で示された桁を表示します。また、ディジット信号は、隣の桁表示がうつらないようにセグメント信号をオフにしてから切り替えます。

図7 - 5 LED表示タイミング

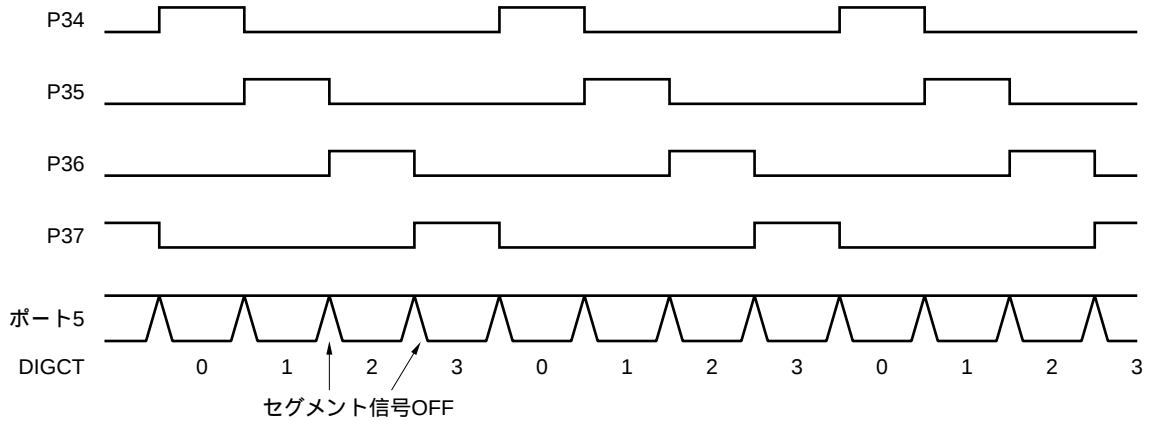
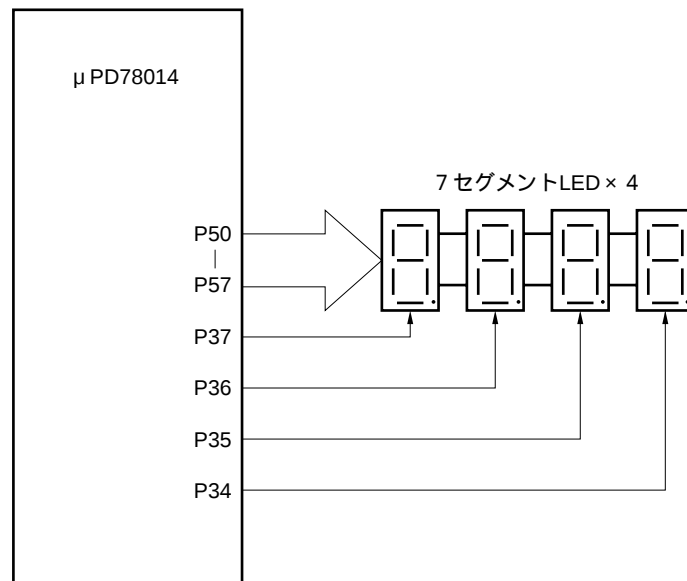


図7 - 6 時計用タイマの回路例



(1) パッケージの説明

<パブリック宣言シンボル>

SECD : 秒データ格納エリア
 MINDP : 分データ格納エリア
 HOURDP : 時データ格納エリア
 LEDDP : LED表示エリア

<使用するレジスタ>

バンク 0 : AX, B, HL

<使用するRAM>

名 称	用 途	属 性	バイト
MINDP	分データ格納	SADDRP	2
HOURDP	時データ格納		
SECD	秒データ格納		1
DIGCT	LED表示桁データ格納		
LEDDP	LED表示データ		4

<使用するハードウェア>

- 時計用タイマ
- P34-37
- P5

<初期設定>

- 時計動作0.5 s , インターバル1.95 ms TMC2 = #00100110B
- 時計用タイマ割り込み許可 TMMK3 = 0

<起動方法>

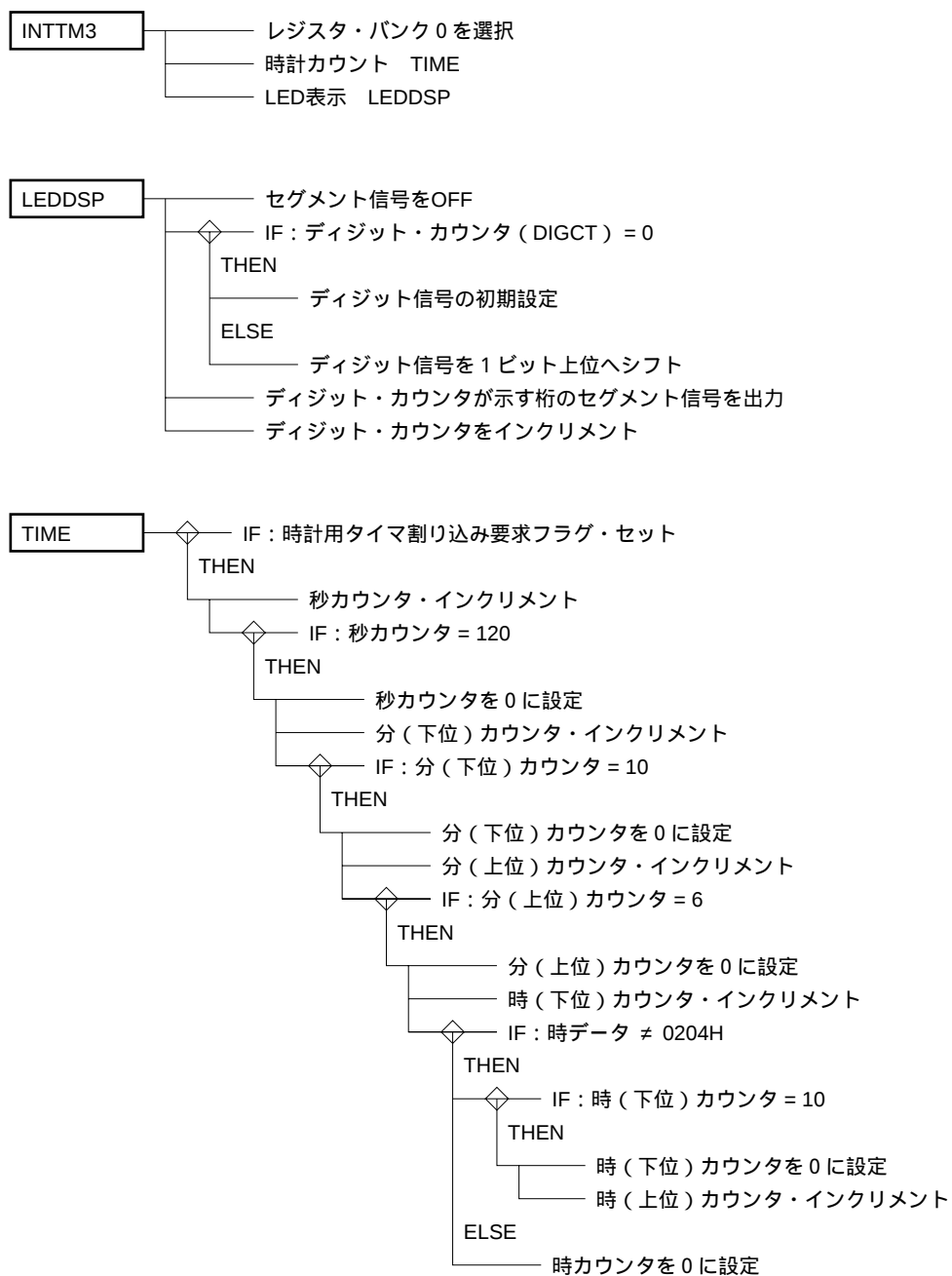
時計用タイマのインターバル・タイマ割り込み要求で起動します。

(2) 使用例

EXTRN MINDP, HOURDP, SECD, LEDDP

```
TMC2 = #00100110B          ; 時計動作0.5 s , インターバル1.95 ms
CLR1  TMMK3                 ; 時計用タイマ割り込み許可
EI
```

(3) SPDチャート



(4) プログラム・リスト

```

PUBLIC  HOURDP, MINDP, SECD, LEDDP

WT_DATP DSEG    SADDRP
MINDP: DS      2                ; 分データ格納エリア
HOURDP: DS     2                ; 時データ格納エリア
SECD: DS       1                ; 秒データ格納エリア
DIGCT: DS       1                ; LED表示桁エリア
LEDDP: DS       4                ; LED表示エリア

VETM3  CSEG     AT 12H
        DW      INTTM3          ; 時計用タイマのベクタ・アドレスの設定

;*****
;*   インターバル割り込み処理
;*****
TM3_SEG CSEG
INTTM3:
        SEL RBO
        CALL !TIME
        CALL !LEDDPSP
        RETI

```

```

;*****
;*          L E D 表示
;*****
LEDDPSP:
    P5=#0FFH                ; セグメント出力OFF
    DIGCT&=#00000011B       ; デジット・カウンタの補正 (0-3)
    if (DIGCT==#0)
        A=P3
        A&=#00001111B       ; デジット信号初期設定 (上位4ビット)
        A|=#00010000B
        P3=A
    else
        A=P3
        A&=#11110000B       ; 上位4ビット・シフト
        X=A
        A=P3
        A+=X
        P3=A
    endif

    B=DIGCT (A)              ; 表示データのアドレス設定
    HL=#LEDDP                ; 表示エリア先頭アドレス
    B=[HL+B] (A)             ; 表示データの設定
    HL=#SEGDT                ; セグメント・データに変換
    P5=[HL+B] (A)            ; セグメント信号出力

    DIGCT++
    RET

SEGDT:
    DB      11000000B        ; 0
    DB      11111001B        ; 1
    DB      10100100B        ; 2
    DB      10110000B        ; 3
    DB      10011001B        ; 4
    DB      10010010B        ; 5
    DB      10000010B        ; 6
    DB      11111000B        ; 7
    DB      10000000B        ; 8
    DB      10010000B        ; 9
    DB      10001000B        ; A
    DB      10000011B        ; B
    DB      11000110B        ; C
    DB      10100001B        ; D
    DB      10000110B        ; E
    DB      10001110B        ; F

$EJECT

```

```

;*****
;*      時計カウント・アップ
;*****
TIME:
    if_bit(WTIF)                                ; 0.5 s テスト
        CLR1    WTIF
        SECD++                                    ; 120 = 60秒 / 0.5
        if(SECD==#120)
            SECD=#0
            (MINDP+0)++                            ; 分下位インクリメント
            if((MINDP+0)==#10)                        ; 桁上げあり
                (MINDP+0)=#0
                (MINDP+1)++                            ; 分上位インクリメント
                if(MINDP+1==#6)                        ; 桁上げあり
                    (MINDP+1)=#0
                    (HOURDP+0)++
                    if(HOURDP!=#0204H) (AX) ; 時データ 24?
                        if((HOURDP+0)==#10) ; 桁上げあり
                            (HOURDP+0)=#0
                            (HOURDP+1)++
                        endif
                    else
                        HOURDP=#0000H
                    endif
                endif
            endif
        endif
    endif
endif
RET

```

第8章 シリアル・インタフェースの応用

78K/0シリーズのシリアル・インタフェースは、表8 - 1 のようになります。

★

表8 - 1 各サブシリーズとシリアル・インタフェース・チャンネルの有無

シリアル・インタフェースの構成 サブシリーズ	チャンネル0				チャンネル1		SIO3	UART0
	3線式	2線式	SBI	I ² Cバス	3線式	自動送受信機能付き3線式	3線式	アシンクロナス・シリアル・インタフェース
μPD78018F	○	○	○	×	○	○	×	×
μPD78018FY	○	○	×	○	○	○	×	×
μPD780024A	×	×	×	×	×	×	○	○
μPD780024AY	×	×	×	×	×	×	○	○
μPD780034A	×	×	×	×	×	×	○	○
μPD780034AY	×	×	×	×	×	×	○	○
μPD78014H	○	○	○	×	○	○	×	×

備考 ○：機能を持っている ×：機能を持っていない

シリアル・インタフェースは、次のレジスタで設定します。

表8 - 2 シリアル・インタフェースのレジスタ一覧

シリアル・インタフェース	使用するレジスタ
チャンネル0	・タイマ・クロック選択レジスタ3 (TCL3) ・シリアル動作モード・レジスタ0 (CSIM0) ・シリアル・バス・インタフェース・コントロール・レジスタ (SBIC) ・割り込みタイミング指定レジスタ (SINT)
チャンネル1	・タイマ・クロック選択レジスタ3 (TCL3) ・シリアル動作モード・レジスタ1 (CSIM1) ・自動データ送受信コントロール・レジスタ (ADTC) ・自動データ送受信間隔指定レジスタ (ADTI)

備考 この章ではシリアル・インタフェース・チャンネル0, 1のレジスタ・フォーマット, 応用例のみ記載してあります。シリアル・インタフェースSIO3, UART0のレジスタのフォーマットを知りたい場合は、各サブシリーズのユーザズ・マニュアルを参照してください。

図8-1 タイマ・クロック選択レジスタ3のフォーマット (μPD78018F, 78014Hサブシリーズ)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
TCL3	TCL37	TCL36	TCL35	TCL34	TCL33	TCL32	TCL31	TCL30	FF43H	88H	R/W

TCL33	TCL32	TCL31	TCL30	シリアル・インタフェース・チャンネル0のシリアル・クロックの選択
0	1	1	0	$f_x/2^2$ 注
0	1	1	1	$f_x/2^3$ (1.25 MHz)
1	0	0	0	$f_x/2^4$ (625 kHz)
1	0	0	1	$f_x/2^5$ (313 kHz)
1	0	1	0	$f_x/2^6$ (156 kHz)
1	0	1	1	$f_x/2^7$ (78.1 kHz)
1	1	0	0	$f_x/2^8$ (39.1 kHz)
1	1	0	1	$f_x/2^9$ (19.5 kHz)
上記以外				設定禁止

TCL37	TCL36	TCL35	TCL34	シリアル・インタフェース・チャンネル1のシリアル・クロックの選択
0	1	1	0	$f_x/2^2$ 注
0	1	1	1	$f_x/2^3$ (1.25 MHz)
1	0	0	0	$f_x/2^4$ (625 kHz)
1	0	0	1	$f_x/2^5$ (313 kHz)
1	0	1	0	$f_x/2^6$ (156 kHz)
1	0	1	1	$f_x/2^7$ (78.1 kHz)
1	1	0	0	$f_x/2^8$ (39.1 kHz)
1	1	0	1	$f_x/2^9$ (19.5 kHz)
上記以外				設定禁止

注 メイン・システム・クロックが4.19 MHz以下で発振しているときのみ設定できます。

注意 TCL3を同一データ以外に書き換える場合は、いったんシリアル転送を停止させたのちに行ってください。

備考1 . f_x : メイン・システム・クロック発振周波数

2 . () 内は、 $f_x = 10.0$ MHz動作時。

図8 - 2 タイマ・クロック選択レジスタ3のフォーマット (μPD78018FYサブシリーズ)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
TCL3	TCL37	TCL36	TCL35	TCL34	TCL33	TCL32	TCL31	TCL30	FF43H	88H	R/W

TCL33	TCL32	TCL31	TCL30	シリアル・インタフェース・チャンネル0のシリアル・クロックの選択	
				I ² Cバス・モード時のシリアル・クロック	3線式シリアルI/O / SBI / 2線式シリアルI/Oモード時のシリアル・クロック ^{注1}
0	1	1	0	$f_x/2^6$ (156 kHz)	$f_x/2^2$ ^{注2}
0	1	1	1	$f_x/2^7$ (78.1 kHz)	$f_x/2^3$ (1.25 MHz)
1	0	0	0	$f_x/2^8$ (39.1 kHz)	$f_x/2^4$ (625 kHz)
1	0	0	1	$f_x/2^9$ (19.5 kHz)	$f_x/2^5$ (313 kHz)
1	0	1	0	$f_x/2^{10}$ (9.8 kHz)	$f_x/2^6$ (156 kHz)
1	0	1	1	$f_x/2^{11}$ (4.9 kHz)	$f_x/2^7$ (78.1 kHz)
1	1	0	0	$f_x/2^{12}$ (2.4 kHz)	$f_x/2^8$ (39.1 kHz)
1	1	0	1	$f_x/2^{13}$ (1.2 kHz)	$f_x/2^9$ (19.5 kHz)
上記以外				設定禁止	

TCL37	TCL36	TCL35	TCL34	シリアル・インタフェース・チャンネル1のシリアル・クロックの選択	
0	1	1	0	$f_x/2^2$ ^{注2}	
0	1	1	1	$f_x/2^3$ (1.25 MHz)	
1	0	0	0	$f_x/2^4$ (625 kHz)	
1	0	0	1	$f_x/2^5$ (313 kHz)	
1	0	1	0	$f_x/2^6$ (156 kHz)	
1	0	1	1	$f_x/2^7$ (78.1 kHz)	
1	1	0	0	$f_x/2^8$ (39.1 kHz)	
1	1	0	1	$f_x/2^9$ (19.5 kHz)	
上記以外				設定禁止	

注1 μPD78018FYサブシリーズにはSBIモードはありません。

2 メイン・システム・クロックが4.19 MHz以下で発振しているときのみ設定できます。

注意 TCL3を同一データ以外に書き換える場合は、いったんシリアル転送を停止させたのちに行ってください。

備考1 f_x : メイン・システム・クロック発振周波数

2 () 内は、 $f_x = 10.0$ MHz動作時。

図8-3 シリアル動作モード・レジスタ0のフォーマット(μPD78018F, 78014HサブシリーズⅠ1/2)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
CSIM0	CSIE 0	COI	WUP	CSIM 04	CSIM 03	CSIM 02	CSIM 01	CSIM 00	FF60H	00H	R/W ^{注1}

R/W	CSIM 01	CSIM 00	シリアル・インタフェース・チャンネル0のクロックの選択								
	0	×	SCK0端子への外部からの入力クロック								
	1	0	8ビット・タイマ・レジスタ(TM2)の出力								
	1	1	タイマ・クロック選択レジスタ3(TCL3)のビット0-ビット3で指定されたクロック								

R/W	CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	動作モード	先頭ビット	SI0/SB0/P25 端子の機能	SO0/SB1/P26 端子の機能	SCK0/P27 端子の機能
	0	×	0	1	×	0	0	0	1	3線式シリアル I/Oモード	MSB	SI0 ^{注2} (入力)	SO0 (CMOS出力)	SCK0 (CMOS入出力)
			1								LSB			
	1	0	0	注3 ×	注3 ×	0	0	0	1	SBIモード	MSB	P25 (CMOS入出力)	SB1 (N-chオープン・ ドレイン入出力)	SCK0 (CMOS入出力)
			1	0	0	注3 ×	注3 ×	0	1			SB0 (N-chオープン・ ドレイン入出力)	P26 (CMOS入出力)	
	1	1	0	注3 ×	注3 ×	0	0	0	1	2線式シリアル I/Oモード	MSB	P25 (CMOS入出力)	SB1 (N-chオープン・ ドレイン入出力)	SCK0 (N-chオープン・ ドレイン入出力)
			1	0	0	注3 ×	注3 ×	0	1			SB0 (N-chオープン・ ドレイン入出力)	P26 (CMOS入出力)	

R/W	WUP	ウエイク・アップ機能の制御 ^{注4}									
	0	すべてのモードで、シリアル転送ごとに割り込み要求信号を発生									
	1	SBIモード使用時、バス・リリース後(CMDD = RELD = 1のとき)に受信したアドレスがスレーブ・アドレス・レジスタのデータと一致したとき、割り込み要求信号を発生									

注1．ビット6(COI)は、Read Onlyです。

2．送信のみ使用するときには、P25(CMOS入力)として使用できます。

3．ポート機能として自由に使用できます。

4．ウエイク・アップ機能を使用(WUP = 1)するときには、割り込みタイミング指定レジスタ(SINT)のビット5(SIC)に0を設定してください。

注意 シリアル・インタフェース・チャンネル0の動作許可中に動作モード(3線式シリアルI/O / 2線式シリアルI/O / SBI)を切り替えないでください。動作モードは、いったんシリアル動作を停止させたのちに切り替えてください。

備考 × : don't care

PM × × : ポート・モード・レジスタ

P × × : ポートの出力ラッチ

図8 - 3 シリアル動作モード・レジスタ0のフォーマット(μPD78018F, 78014HサブシリーズⅠ2/2)

R	COI	スレープ・アドレス比較結果フラグ ^注
	0	スレープ・アドレス・レジスタとシリアルI/Oシフト・レジスタ0のデータが一致しない
	1	スレープ・アドレス・レジスタとシリアルI/Oシフト・レジスタ0のデータが一致する

R/W	CSIE0	シリアル・インタフェース・チャンネル0の動作の制御
	0	動作停止
	1	動作許可

注 CSIE0 = 0のとき、COIは0になります。

注意 シリアル・インタフェース・チャンネル0の動作許可中に動作モード(3線式シリアルI/O / 2線式シリアルI/O / SBI)を切り替えしないでください。動作モードは、いったんタイマ動作を停止させたのちに切り替えてください。

図8-4 シリアル動作モード・レジスタ0のフォーマット(μPD78018FYサブシリーズ)(1/2)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
CSIM0	CSIE 0	COI	WUP	CSIM 04	CSIM 03	CSIM 02	CSIM 01	CSIM 00	FF60H	00H	R/W ^{注1}

R/W	CSIM 01	CSIM 00	シリアル・インタフェース・チャンネル0のクロックの選択								
	0	×	SCK0/SCL端子への外部からの入力クロック								
	1	0	8ビット・タイマ・レジスタ(TM2)の出力 ^{注2}								
	1	1	タイマ・クロック選択レジスタ3(TCL3)のビット0-ビット3で指定されたクロック								

R/W	CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	動作モード	先頭ビット	SI0/SB0/SDA0 /P25端子の機能	SO0/SB1/SDA1 /P26端子の機能	SCK0/SCL/P27 端子の機能
	0	×	0	1	×	0	0	0	1	3線式シリアル I/Oモード	MSB	SI0 ^{注3} (入力)	SO0 (CMOS出力)	SCK0 (CMOS入出力)
			1								LSB			
	1	1	0	注4 ×	注4 ×	0	0	0	1	2線式シリアル I/Oモードまたは I ² Cバス・モード	MSB	P25 (CMOS入出力)	SB1/SDA1 (N-chオープン・ ドレイン入出力)	SCK0/SCL (N-chオープン・ ドレイン入出力)
			1	0	0	注4 ×	注4 ×	0	1			SB0/SDA0 (N-chオープン・ ドレイン入出力)	P26 (CMOS入出力)	

注1．ビット6(COI)は、Read Onlyです。

2．I²Cバス・モード時、クロックはTO2が出力するクロック周波数の1/16となります。

3．送信のみ使用するときは、P25(CMOS入力)として使用できます。

4．ポート機能として自由に使用できます。

注意 シリアル・インタフェース・チャンネル0の動作許可中に動作モード(3線式シリアルI/O / 2線式シリアルI/O / I²Cバス)を切り替えしないでください。動作モードは、いったんシリアル動作を停止させたのちに切り替えてください。

備考 × : don't care

PM×× : ポート・モード・レジスタ

P×× : ポートの出力ラッチ

図8 - 4 シリアル動作モード・レジスタ0のフォーマット (μPD78018FYサブシリーズ) (2/2)

R/W	WUP	ウェイク・アップ機能の制御 ^{注1}
	0	すべてのモードで、シリアル転送ごとに割り込み要求信号を発生
	1	I ² Cバス・モード使用時、スタート・コンディション検出後 (CMDD = 1 のとき) に受信したアドレスがスレーブ・アドレス・レジスタのデータと一致したとき、割り込み要求信号を発生
R	COI	スレープ・アドレス比較結果フラグ ^{注2}
	0	スレープ・アドレス・レジスタとシリアルI/Oシフト・レジスタ0のデータが一致しない
	1	スレープ・アドレス・レジスタとシリアルI/Oシフト・レジスタ0のデータが一致する
R/W	CSIE0	シリアル・インタフェース・チャンネル0の動作の制御
	0	動作停止
	1	動作許可

注1 ウェイク・アップ機能を使用 (WUP = 1) するときは、割り込みタイミング指定レジスタ (SINT) のビット5 (SIC) に1を設定してください。また、WUP = 1の期間にシリアルI/Oシフト・レジスタ0 (SIO0) への書き込み命令を実行しないでください。

2 CSIE0 = 0のときCOIは0になります。

注意 シリアル・インタフェース・チャンネル0の動作許可中に動作モード (3線式シリアルI/O / 2線式シリアルI/O / I²Cバス) を切り替えしないでください。動作モードは、いったんシリアル動作を停止させたのちに切り替えてください。

図8-5 シリアル動作モード・レジスタ1のフォーマット
(μ PD78018F, 78018FY, 78014Hサブシリーズのみ)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
CSIM1	CSIE 1	DIR	ATE	0	0	0	CSIM 11	CSIM 10	FF68H	00H	R/W

CSIM 11	CSIM 10	シリアル・インタフェース・チャンネル1のクロックの選択
0	×	SCK1端子への外部クロック入力 ^{注1}
1	0	8ビット・タイマ・レジスタ2 (TM2) の出力
1	1	タイマ・クロック選択レジスタ3 (TCL3) のビット4-ビット7で指定されたクロック

ATE	シリアル・インタフェース・チャンネル1の動作モードの選択
0	3線式シリアルI/Oモード
1	自動送受信機能付き3線式シリアルI/Oモード

DIR	先頭ビット	SI1端子の機能	SO1端子の機能
0	MSB	SI1/P20 (入力)	SO1
1	LSB		(CMOS出力)

CSIE 1	CSIM 11	PM20	P20	PM21	P21	PM22	P22	シフト・レジスタ1 の動作	シリアル・クロックの カウンタの動作の制御	SI1/P20 端子の機能	SO1/P21 端子の機能	$\overline{\text{SCK1}}$ /P22 端子の機能
0	×	注2 ×	注2 ×	注2 ×	注2 ×	注2 ×	注2 ×	動作停止	クリア	P20 (CMOS入出力)	P21 (CMOS入出力)	P22 (CMOS入出力)
1	0	注3 1	注3 ×	0	0	1	×	動作許可	カウント動作	SI1 ^{注3} (入力)	SO1 (CMOS出力)	$\overline{\text{SCK1}}$ (入力)
	1					0	1					$\overline{\text{SCK1}}$ (CMOS出力)

注1．CSIM11を0にして外部クロック入力を選択したとき、自動データ送受信コントロール・レジスタ (ADTC) のビット2, 1 (STRB, BUSY1) を0, 0に設定してください。

2．ポート機能として自由に使用できます。

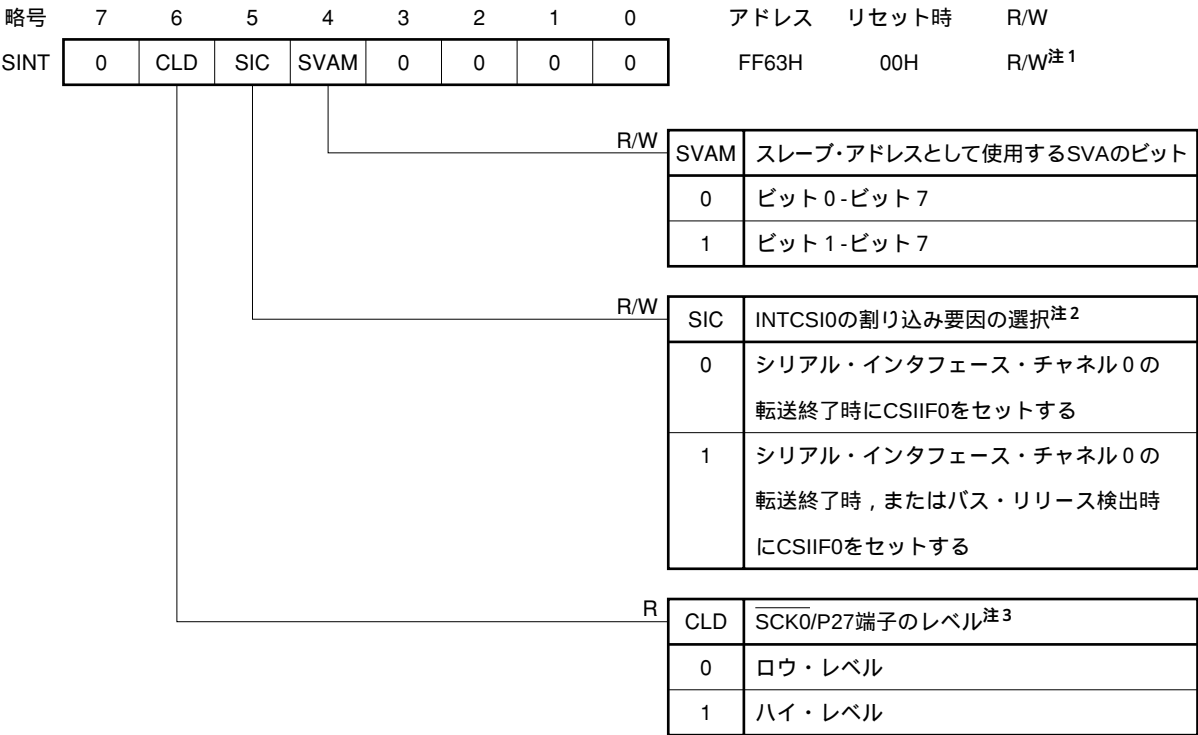
3．送信のみ使用するときは、P20 (CMOS入出力) として使用できます (ADTCのビット7 (RE) に0を設定してください)。

備考 × : don't care

PM×× : ポート・モード・レジスタ

P×× : ポートの出力ラッチ

図8 - 6 割り込みタイミング指定レジスタのフォーマット (μPD78018F, 78014Hサブシリーズ)



- 注1 . ビット6 (CLD) は、Read Onlyです。
- 2 . SBIモードでウェイク・アップ機能を使用するときは、SICに0を設定してください。
- 3 . CSIE0 = 0のとき、CLDは0になります。

注意 ビット0-3には、必ず0を設定してください。

備考 SVA : スレーブ・アドレス・レジスタ
CSIF0 : INTCSI0に対応する割り込み要求フラグ
CSIE0 : シリアル動作モード・レジスタ0 (CSIM0) のビット7

図8-7 割り込みタイミング指定レジスタのフォーマット (μPD78018FYサブシリーズ) (1/2)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
SINT	0	CLD	SIC	SVAM	CLC	WREL	WAT1	WAT0	FF63H	00H	R/W ^{注1}

R/W	WAT1	WAT0	ウェイトおよび割り込み処理要求の制御
	0	0	SCK0の8クロック目の立ち上がりで割り込み処理要求を発生する（クロック出力はハイ・インピーダンス）。
	0	1	設定禁止
	1	0	I ² Cバス・モード使用時に使用する（8クロック・ウェイト）。 SCLの8クロック目の立ち上がりで割り込み処理要求を発生する（マスタの場合、8クロック出力後、SCL出力をロウ・レベルにしてウェイトする。スレーブの場合、8クロック入力後、SCL端子をロウ・レベルにしてウェイト要求する）。
	1	1	I ² Cバス・モード使用時に使用する（9クロック・ウェイト）。 SCLの9クロック目の立ち上がりで割り込み処理要求を発生する（マスタの場合、9クロック出力後、SCL出力をロウ・レベルにしてウェイトする。スレーブの場合、9クロック入力後、SCL端子をロウ・レベルにしてウェイト要求する）。

R/W	WREL	ウェイト解除の制御
	0	ウェイト解除状態
	1	ウェイト状態を解除する。 解除後自動的にクリア（0）される（WAT0, WAT1によるウェイト状態の解除に使用する）。

R/W	CLC	クロック・レベルの制御 ^{注2}
	0	I ² Cバス・モード使用時に使用する。 シリアル転送時以外の場合、SCL端子の出力レベルをロウ・レベルにする。
	1	I ² Cバス・モード使用時に使用する。 シリアル転送時以外の場合、SCL端子の出力レベルをハイ・インピーダンスにする（クロック・ラインはハイ・レベル）。 マスタがスタート/ストップ・コンディションを生成するために使用する。

注1．ビット6（CLD）は、Read Onlyです。

2．I²Cバス・モードを使用しない場合は、CLCに0を設定してください。

図8 - 7 割り込みタイミング指定レジスタのフォーマット (μPD78018FYサブシリーズ) (2/2)

R/W	SVAM	スレープ・アドレスとして使用するSVAのビット
	0	ビット0-ビット7
	1	ビット1-ビット7
R/W	SIC	INTCSI0の割り込み要因の選択 ^{注1}
	0	シリアル・インタフェース・チャンネル0の転送終了時にCSIIF0をセット(1)する。
	1	シリアル・インタフェース・チャンネル0の転送終了時、またはI ² Cバス・モードでストップ・コンディショ ン検出時にCSIIF0をセット(1)する。
R	CLD	SCK0/SCL/P27端子のレベル ^{注2}
	0	ロウ・レベル
	1	ハイ・レベル

注1 . I²Cバス・モードでウエイク・アップ機能を使用するときは、SICに1を設定してください。

2 . CSIE0 = 0のとき、CLDは0になります。

備考 SVA : スレープ・アドレス・レジスタ

CSIIF0 : INTCSI0に対応する割り込み要求フラグ

CSIE0 : シリアル動作モード・レジスタ0 (CSIM0) のビット7

図8 - 8 シリアル・バス・インタフェース・コントロール・レジスタのフォーマット
(μ PD78018F, 78014Hサブシリーズ) (1/2)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W ^注
R/W	RELT	バス・リリース信号出力のために使用する。 RELT=1により、SOラッチがセット（1）される。SOラッチをセット後、自動的にクリア（0）される。 また、CSIE0=0のときもクリア（0）される。									
R/W	CMDT	コマンド信号出力のために使用する。 CMDT=1により、SOラッチがクリア（0）される。SOラッチをクリア後、自動的にクリア（0）される。 また、CSIE0=0のときもクリア（0）される。									
R	RELD	バス・リリース検出									
		クリアされる条件（RELD=0）					セットされる条件（RELD=1）				
		● 転送スタート命令実行時 ● アドレス受信時にSIO0とSVAの値が一致しないとき ● CSIE0=0のとき ● RESET入力時					● バス・リリース信号（REL）検出時				
R	CMDD	コマンド検出									
		クリアされる条件（CMDD=0）					セットされる条件（CMDD=1）				
		● 転送スタート命令実行時 ● バス・リリース信号（REL）検出時 ● CSIE0=0のとき ● RESET入力時					● コマンド信号（CMD）検出時				
R/W	ACKT	セット（1）する命令実行直後のSCK0のクロックの立ち下がりエッジに同期してアクノリッジ信号を出力し、出力後、自動的にクリア（0）される。ACKE=0として使用する。 また、シリアル・インタフェースの転送開始、CSIE0=0のときもクリア（0）される。									

注 ビット2, 3, 6 (RELD, CMDD, ACKD) はRead Onlyです。

備考1. ビット0, 1, 4 (RELT, CMDT, ACKT) はデータ設定後に読み出すと0になっています。

2. CSIE0: シリアル動作モード・レジスタ0 (CSIM0) のビット7

図8 - 8 シリアル・バス・インタフェース・コントロール・レジスタのフォーマット
(μ PD78018F, 78014Hサブシリーズ) (2/2)

R/W	ACKE	アクノリッジ信号出力の制御	
	0	アクノリッジ信号の自動出力禁止 (ACKTによる出力は可能)	
	1	転送完了前	SCK0の9クロック目の立ち下がりエッジに同期してアクノリッジ信号を出力 ($\overline{\text{ACKE}} = 1$ により, 自動出力される)
		転送完了後	セット (1) する命令実行直後のSCK0のクロックの立ち下がりエッジに同期してアクノリッジ信号を出力する ($\overline{\text{ACKE}} = 1$ により, 自動出力される)。ただし, アクノリッジ信号を出力後, 自動的にクリア (0) されない。

R	ACKD	アクノリッジ検出	
	クリアされる条件 ($\text{ACKD} = 0$)		セットされる条件 ($\text{ACKD} = 1$)
	- 転送スタート命令実行後, ビジィ・モードを解除した直後のSCK0のクロックの立ち下がり時 $\overline{\text{CSIE0}} = 0$のとき $\overline{\text{RESET}}$入力時		転送完了後のSCK0のクロックの立ち上がりエッジでアクノリッジ信号 ($\overline{\text{ACK}}$) 検出時

R/W	BSYE ^注	同期ビジィ信号出力の制御	
	0	クリア (0) する命令実行直後のSCK0のクロックの立ち下がりエッジに同期した, ビジィ信号の出力を禁止する。	
	1	アクノリッジ信号に続くSCK0のクロックの立ち下がりエッジからビジィ信号を出力する。	

注 シリアル・インタフェースの転送開始によってビジィ・モードを解除できます。ただし, BSYEフラグは0にクリアされません。

備考 CSIE0: シリアル動作モード・レジスタ0 (CSIM0) のビット7

図8 - 9 シリアル・バス・インタフェース・コントロール・レジスタのフォーマット
(μ PD78018FYサブシリーズ) (1/2)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W ^注

R/W	RELT	I ² Cバス・モード使用時，ストップ・コンディション出力のために使用する。 RELT=1により，SOラッチがセット（1）される。SOラッチをセット後，自動的にクリア（0）される。 また，CSIE0=0のときもクリア（0）される。
-----	------	--

R/W	CMDT	I ² Cバス・モード使用時，スタート・コンディション出力のために使用する。 CMDT=1により，SOラッチがクリア（0）される。SOラッチをクリア後，自動的にクリア（0）される。 また，CSIE0=0のときもクリア（0）される。
-----	------	--

R	RELD	ストップ・コンディション検出
クリアされる条件（RELD=0）		セットされる条件（RELD=1）
- 転送スタート命令実行時 - アドレス受信時にSIO0とSVAの値が一致しないとき - CSIE0=0のとき - RESET入力時		I²Cバス・モードでストップ・コンディション検出時

R	CMDD	スタート・コンディション検出
クリアされる条件（CMDD=0）		セットされる条件（CMDD=1）
- 転送スタート命令実行時 - I²Cバス・モードでストップ・コンディション検出時 - CSIE0=0のとき - RESET入力時		I²Cバス・モードでスタート・コンディション検出時

R/W	ACKT	セット命令（ACKT=1）実行直後から次のSCLの立ち下がりまでSDA0（SDA1）をロウ・レベルにする。 8クロック・ウェイト選択時に，ソフトウェアでACK信号を生成するために使用する。 シリアル・インタフェースの転送開始，CSIE0=0のときクリア（0）される。
-----	------	---

注 ビット2，3，6（RELD，CMDD，ACKD）はRead Onlyです。

備考1．ビット0，1，4（RELT，CMDT，ACKT）はデータ設定後に読み出すと0になっています。

2．CSIE0：シリアル動作モード・レジスタ0（CSIM0）のビット7

図8 - 9 シリアル・バス・インタフェース・コントロール・レジスタのフォーマット
(μ PD78018FYサブシリーズ) (2/2)

R/W	ACKE	アクノリッジ信号の自動出力の制御 ^{注1}	
	0	アクノリッジ信号の自動出力禁止 (ACKTによる出力は可能)。 送信時または8クロック・ウェイト選択時で受信の場合に使用する。 ^{注2}	
	1	アクノリッジ信号の自動出力許可。 SCLの9クロック目の立ち下がりエッジに同期して、アクノリッジ信号を出力する (ACKE = 1により、自動出力される)。出力後、自動的にクリア (0) されない。 9クロック・ウェイト選択時で受信の場合に使用する。	
R	ACKD	アクノリッジ検出	
		クリアされる条件 (ACKD = 0)	セットされる条件 (ACKD = 1)
		- 転送スタート命令実行時 - CSIE0 = 0のとき - RESET入力時	転送完了後のSCLのクロックの立ち上がりエッジでアクノリッジ信号検出時
R/W	BSYE ^{注3}	I ² Cバス・モード使用時の送信用N-chオープン・ドレイン出力の制御 ^{注4}	
	0	出力許可 (送信)	
	1	出力禁止 (受信)	

注1．転送開始前に設定してください。

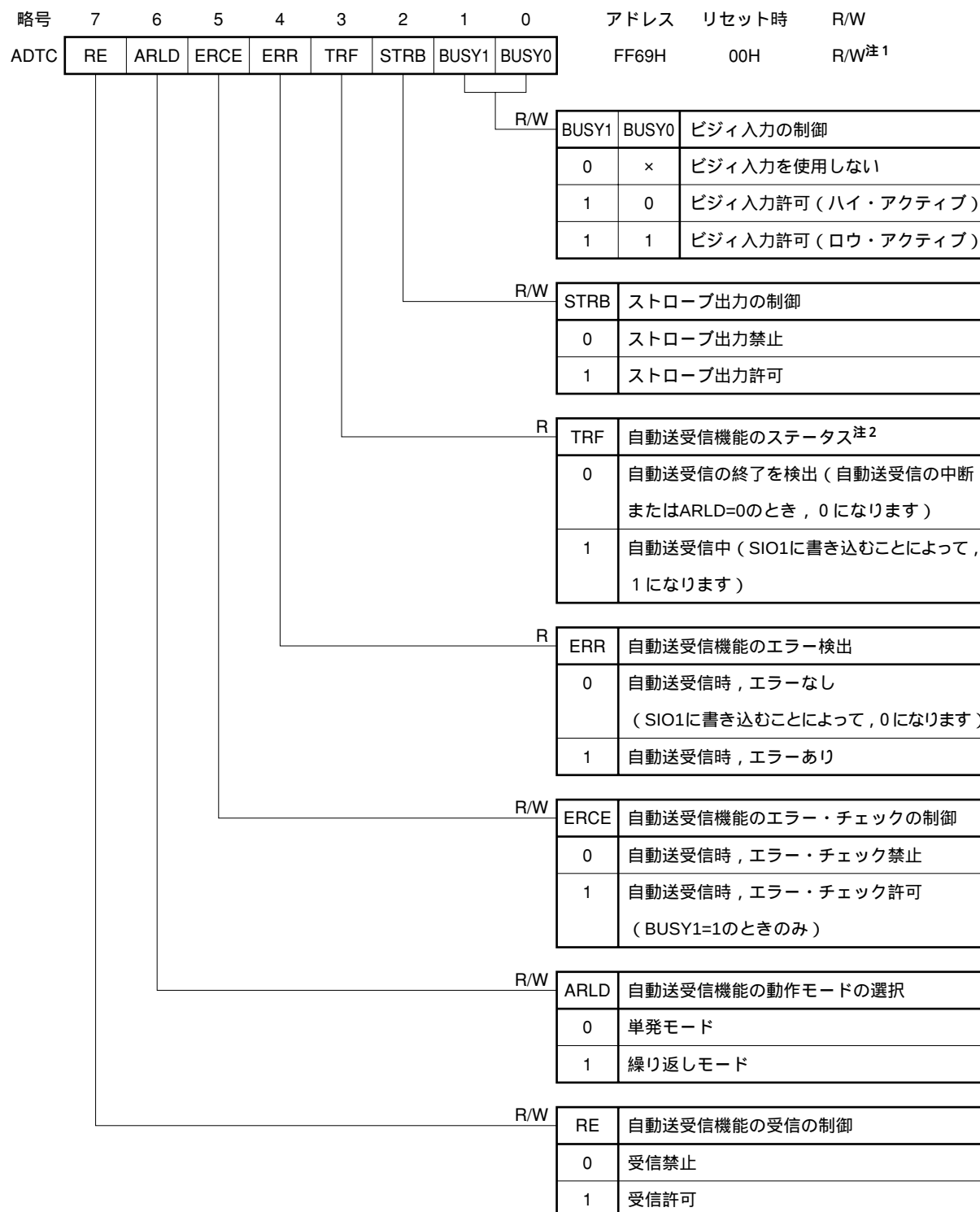
2．8クロック・ウェイト選択時では、受信時のアクノリッジ信号はACKTを用いて出力してください。

3．シリアル・インタフェースの転送開始によってウェイト状態を解除できます。ただし、BSYEはクリア (0) されません。

4．ウェイク・アップ機能を使用するときには、必ずBSYEに1を設定してください。

備考 CSIE0：シリアル動作モード・レジスタ (CSIM0) のビット7

図8 - 10 自動データ送受信コントロール・レジスタのフォーマット
(μ PD78018F, 78018FY, 78014Hサブシリーズのみ)



注1．ビット3, 4 (TRF, ERR) は, Read Onlyです。

2．自動送受信の終了判定はCSIIF1 (割り込み要求フラグ) ではなくTRFで行ってください。

(次ページへ続く)

注意 シリアル動作モード・レジスタ1 (CSIM1) のビット1 (CSIM11) を0にして外部クロック入力を選択したとき, ADTCのビット1, 2 (BUSY1, STRB) を0, 0に設定してください。

備考 × : don't care

図8 - 11 自動データ送受信間隔指定レジスタのフォーマット
(μ PD78018F, 78018FY, 78014Hサブシリーズのみ) (1/2)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI7	データ転送のインターバル時間の制御
0	ADTIによるインターバル時間の制御なし ^{注1}
1	ADTI (ADTI0-ADTI4) によるインターバル時間の制御あり

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	データ転送のインターバル時間の指定 ($f_x = 10.0 \text{ MHz}$ 動作時)	
					最小値 ^{注2}	最大値 ^{注2}
0	0	0	0	0	$18.4 \mu s + 0.5/f_{SCK}$	$20.0 \mu s + 1.5/f_{SCK}$
0	0	0	0	1	$31.2 \mu s + 0.5/f_{SCK}$	$32.8 \mu s + 1.5/f_{SCK}$
0	0	0	1	0	$44.0 \mu s + 0.5/f_{SCK}$	$45.6 \mu s + 1.5/f_{SCK}$
0	0	0	1	1	$56.8 \mu s + 0.5/f_{SCK}$	$58.4 \mu s + 1.5/f_{SCK}$
0	0	1	0	0	$69.6 \mu s + 0.5/f_{SCK}$	$71.2 \mu s + 1.5/f_{SCK}$
0	0	1	0	1	$82.4 \mu s + 0.5/f_{SCK}$	$84.0 \mu s + 1.5/f_{SCK}$
0	0	1	1	0	$95.2 \mu s + 0.5/f_{SCK}$	$96.8 \mu s + 1.5/f_{SCK}$
0	0	1	1	1	$108.0 \mu s + 0.5/f_{SCK}$	$109.6 \mu s + 1.5/f_{SCK}$
0	1	0	0	0	$120.8 \mu s + 0.5/f_{SCK}$	$122.4 \mu s + 1.5/f_{SCK}$
0	1	0	0	1	$133.6 \mu s + 0.5/f_{SCK}$	$135.2 \mu s + 1.5/f_{SCK}$
0	1	0	1	0	$146.4 \mu s + 0.5/f_{SCK}$	$148.0 \mu s + 1.5/f_{SCK}$
0	1	0	1	1	$159.2 \mu s + 0.5/f_{SCK}$	$160.8 \mu s + 1.5/f_{SCK}$
0	1	1	0	0	$172.0 \mu s + 0.5/f_{SCK}$	$173.6 \mu s + 1.5/f_{SCK}$
0	1	1	0	1	$184.8 \mu s + 0.5/f_{SCK}$	$186.4 \mu s + 1.5/f_{SCK}$
0	1	1	1	0	$197.6 \mu s + 0.5/f_{SCK}$	$199.2 \mu s + 1.5/f_{SCK}$
0	1	1	1	1	$210.4 \mu s + 0.5/f_{SCK}$	$212.0 \mu s + 1.5/f_{SCK}$

(次ページへ続く)

注1．インターバル時間は，CPU処理にのみ依存します。

- 2．データ転送のインターバル時間には，誤差が含まれています。各データ転送のインターバル時間の最小値と最大値は次の式により求められます（ n ：ADTI0-ADTI4に設定した値）。ただし，次の式から計算された最小値が $2/f_{SCK}$ よりも小さい場合，インターバル時間の最小値は $2/f_{SCK}$ となります。

$$\text{最小値} = (n + 1) \times \frac{2^7}{f_x} + \frac{56}{f_x} + \frac{0.5}{f_{SCK}}$$

$$\text{最大値} = (n + 1) \times \frac{2^7}{f_x} + \frac{72}{f_x} + \frac{1.5}{f_{SCK}}$$

注意1．自動送受信機能動作中は，ADTIへの書き込みを行わないでください。

- 2．ビット5,6には，必ず0を設定してください。

- 3．ADTIを使用して自動送受信によるデータ転送のインターバル時間を制御する場合，ビジィ制御は無効になります。

備考1． f_x ：メイン・システム・クロック発振周波数

- 2． f_{SCK} ：シリアル・クロック周波数

図8 - 11 自動データ送受信間隔指定レジスタのフォーマット
(μPD78018F, 78018FY, 78014Hサブシリーズのみ) (2/2)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	データ転送のインターバル時間の指定 (fx = 10.0 MHz動作時)	
					最小値 ^注	最大値 ^注
1	0	0	0	0	$223.2 \mu s + 0.5/f_{SCK}$	$224.8 \mu s + 1.5/f_{SCK}$
1	0	0	0	1	$236.0 \mu s + 0.5/f_{SCK}$	$237.6 \mu s + 1.5/f_{SCK}$
1	0	0	1	0	$248.8 \mu s + 0.5/f_{SCK}$	$250.4 \mu s + 1.5/f_{SCK}$
1	0	0	1	1	$261.6 \mu s + 0.5/f_{SCK}$	$263.2 \mu s + 1.5/f_{SCK}$
1	0	1	0	0	$274.4 \mu s + 0.5/f_{SCK}$	$276.0 \mu s + 1.5/f_{SCK}$
1	0	1	0	1	$287.2 \mu s + 0.5/f_{SCK}$	$288.8 \mu s + 1.5/f_{SCK}$
1	0	1	1	0	$300.0 \mu s + 0.5/f_{SCK}$	$301.6 \mu s + 1.5/f_{SCK}$
1	0	1	1	1	$312.8 \mu s + 0.5/f_{SCK}$	$314.4 \mu s + 1.5/f_{SCK}$
1	1	0	0	0	$325.6 \mu s + 0.5/f_{SCK}$	$327.2 \mu s + 1.5/f_{SCK}$
1	1	0	0	1	$338.4 \mu s + 0.5/f_{SCK}$	$340.0 \mu s + 1.5/f_{SCK}$
1	1	0	1	0	$351.2 \mu s + 0.5/f_{SCK}$	$352.8 \mu s + 1.5/f_{SCK}$
1	1	0	1	1	$364.0 \mu s + 0.5/f_{SCK}$	$365.6 \mu s + 1.5/f_{SCK}$
1	1	1	0	0	$376.8 \mu s + 0.5/f_{SCK}$	$378.4 \mu s + 1.5/f_{SCK}$
1	1	1	0	1	$389.6 \mu s + 0.5/f_{SCK}$	$391.2 \mu s + 1.5/f_{SCK}$
1	1	1	1	0	$402.4 \mu s + 0.5/f_{SCK}$	$404.0 \mu s + 1.5/f_{SCK}$
1	1	1	1	1	$415.2 \mu s + 0.5/f_{SCK}$	$416.8 \mu s + 1.5/f_{SCK}$

注 データ転送のインターバル時間には、誤差が含まれています。各データ転送のインターバル時間の最小値と最大値は次の式により求められます (n : ADTI0-ADTI4に設定した値)。ただし、次の式から計算された最小値が $2/f_{SCK}$ よりも小さい場合、インターバル時間の最小値は $2/f_{SCK}$ となります。

$$\text{最小値} = (n + 1) \times \frac{2^7}{f_x} + \frac{56}{f_x} + \frac{0.5}{f_{SCK}}$$

$$\text{最大値} = (n + 1) \times \frac{2^7}{f_x} + \frac{72}{f_x} + \frac{1.5}{f_{SCK}}$$

注意1．自動送受信機能動作中は、ADTIへの書き込みを行わないでください。

2．ビット5, 6には、必ず0を設定してください。

3．ADTIを使用して自動送受信によるデータ転送のインターバル時間を制御する場合、ビジィ制御は無効になります。

備考1．fx : メイン・システム・クロック発振周波数

2．fSCK : シリアル・クロック周波数

8.1 EEPROM™ (μ PD6252) とのインタフェース

μ PD6252^注は、2048ビットの電氣的に書き込み消去可能なEEPROMです。 μ PD6252への書き込み / 読み出し方法は、3線式シリアル・インタフェースで行います。

注 μ PD6252は保守製品です。

図8 - 12 μ PD6252の端子接続図

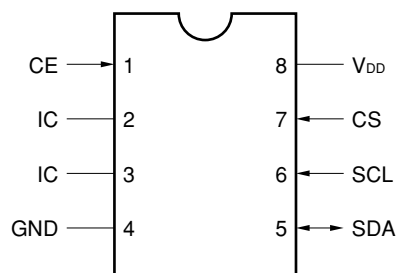
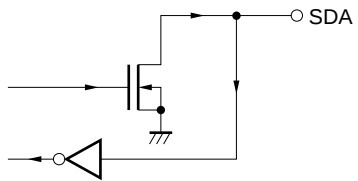


表 8 - 3 μ PD6252の端子の説明

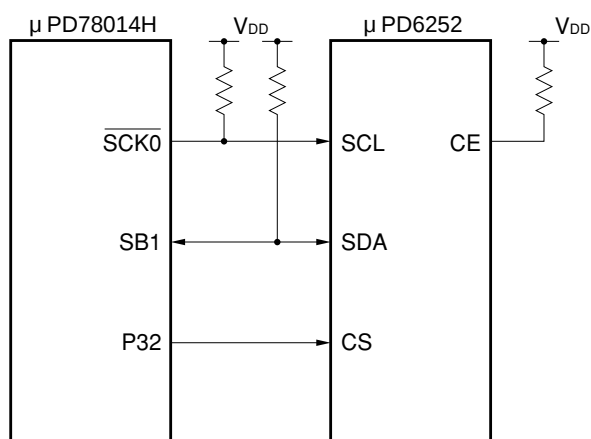
端子番号	端子名称	入出力	機能
1	CE	CMOS入力	データ転送中はハイ・レベルにしてください。 注意 データ転送中にこの端子をハイ・レベルからロウ・レベルに切り替えないでください。 この端子をハイ・レベルからロウ・レベルに切り替える場合は、CS端子（7ピン）をロウ・レベルに設定した状態で行ってください。なお、CE端子およびCS端子をどちらもロウ・レベルにすることで、スタンバイ状態となります。スタンバイ状態では低消費電力となります。
2	IC	-	IC端子は外部で個別に抵抗を介して、ハイ・レベルまたはロウ・レベルに固定してください。
3			
4	GND	-	グラウンド
5	SDA	CMOS入力 / Nchオープン・ドレイン出力	データ入出力用端子です。 Nchオープン・ドレインの入出力のため外部にプルアップ抵抗を付加して使用してください。 
6	SCL	CMOS入力	データ転送用のクロック入力端子です。
7	CS	CMOS入力	チップ・セレクト端子です。ハイ・レベル入力により μPD6252は動作可能となります。 ロウ・レベルにすることによりメモリ・セルの読み出しや書き込み動作ができなくなります。 SCL端子がハイの状態、この端子をロウからハイへ変化させることがシリアル・バス・インタフェースの動作開始の信号となります。また、この端子をハイからロウへ変化させることがシリアル・バス・インタフェースの動作終了の信号となります。
8	V _{DD}	-	正電源 + 5 V $\pm$ 10 %

8.1.1 2線式シリアルI/Oモードの通信

μPD6252^準の3線式とは、シリアル・クロック（SCL）、データ（SDA）、チップ・セレクト（CS）の3本を示しています。したがって、ハンドシェイクを除くと、インタフェースに必要な線はクロックとデータの2線になりますので、78K/0シリーズを使用してインタフェースを行うときは2線式シリアルI/Oモードを選択します。ここではμPD78014Hサブシリーズを例に説明します。

注 μPD6252は保守製品です。

図8 - 13 μPD6252との接続例



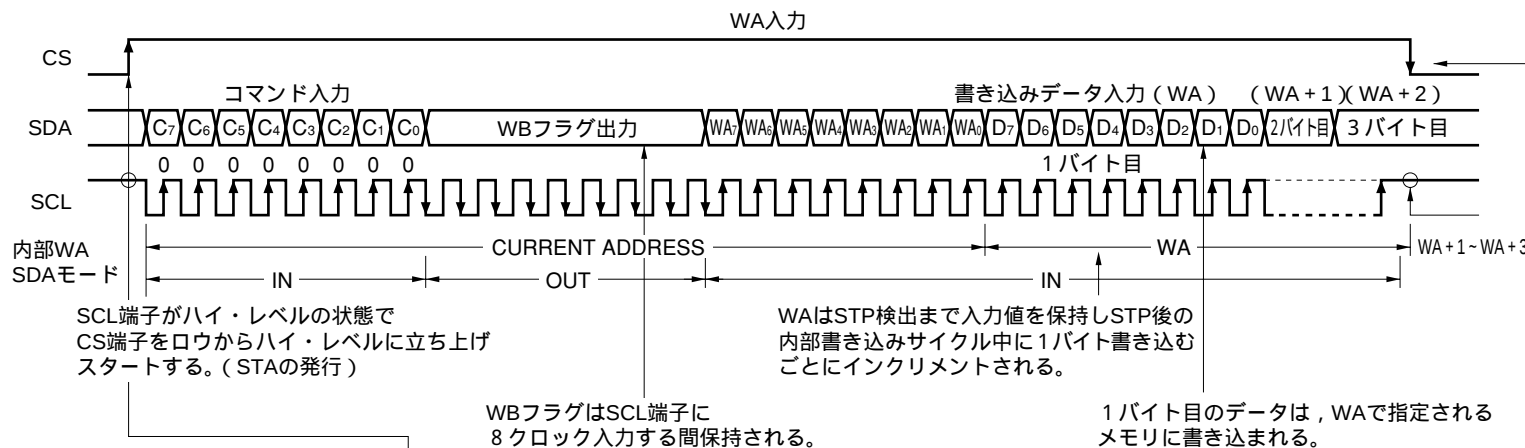
μPD6252への書き込み、読み出しを行うときのコマンド、および通信フォーマットを表8 - 4、図8 - 14に示します。

表 8 - 4 μ PD6252のコマンド一覧

コマンド名	コマンド	動 作 内 容								
RANDOM WRITE	00000000B〔00H〕 MSB C7-C0	ワード・アドレス（WA）（8ビット）を設定後書き込みデータを転送します。書き込みデータは連続して、3バイトまで設定できます。 <table><tr><th colspan="2">ワード・アドレスとの対応</th></tr><tr><td>WA</td><td>1バイト目のデータ</td></tr><tr><td>WA + 1</td><td>2バイト目のデータ</td></tr><tr><td>WA + 2</td><td>3バイト目のデータ</td></tr></table> 書き込み動作はCS端子をハイからロウへ立ち下げたタイミング後の内部書き込みサイクルで実行されます。	ワード・アドレスとの対応		WA	1バイト目のデータ	WA + 1	2バイト目のデータ	WA + 2	3バイト目のデータ
ワード・アドレスとの対応										
WA	1バイト目のデータ									
WA + 1	2バイト目のデータ									
WA + 2	3バイト目のデータ									
CURRENT READ	10000000B〔80H〕 MSB C7-C0	コマンドを設定したときのワード・アドレス（WA）（カレント・アドレス）で指定されるメモリの内容が読み出し用データ・バッファへ転送されます。SDA端子よりデータを読み出すと、8ビット読み出すごとにワード・アドレス（WA）はインクリメントされ、対応するメモリの内容が読み出し用データ・バッファへ転送されます。								
RANDOM READ	11000000B〔C0H〕 MSB C7-C0	ワード・アドレス（WA）を設定後、設定したワード・アドレス（WA）を先頭番地としてデータ読み取りを実行します。 CURRENT READと異なる点は、コマンドを実行後、ワード・アドレス（WA）を設定することです。 ワード・アドレス（WA）の設定後は、CURRENT READと同じ動作をします。								

図 8 - 14 μ PD6252の通信フォーマット (1/2)

(1) RANDOM WRITE



(2) CURRENT READ

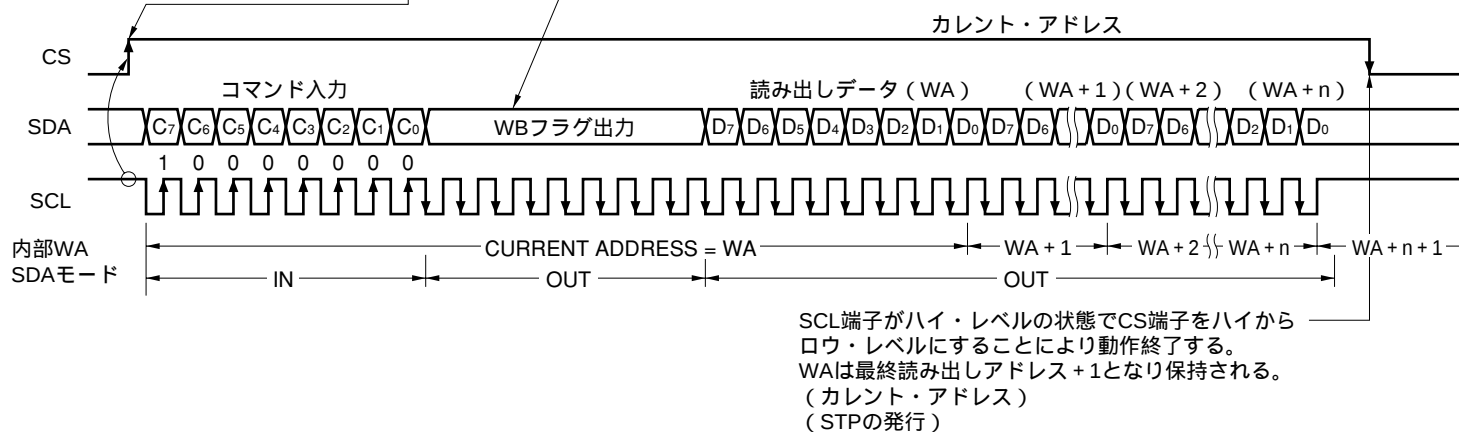
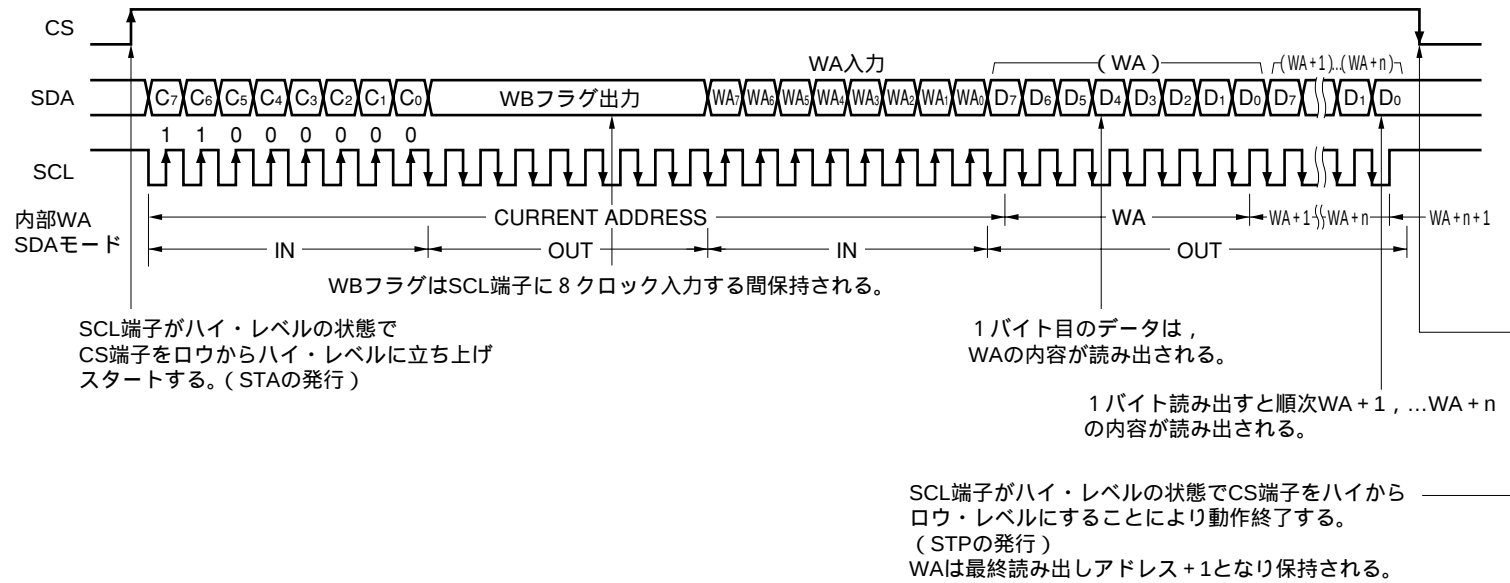


図 8 - 14 μ PD6252の通信フォーマット (2/2)

(3) RANDOM READ



から にμPD6252の手順を示します。この例では、1回のインタフェースでの書き込み、読み出しのデータ数は1バイト固定です。また、インタフェース時にμPD6252がライト・ビジィ(WB)であったときは、ビジィ・フラグをセットします。

インタフェースを開始するためにCS端子(P32)をハイ・レベルにします。

書き込み、読み出しのコマンドを送信します。

WRITE BUSYのデータを受信します。μPD6252のインタフェース可能な状態であれば、00Hを受信します。00H以外を受信したときは、WRITE BUSY状態と判断します。そして、通信の中止処理を行います。

コマンドに対応したデータを転送します。

通信を終了するためにCS端子(P32)をロウ・レベルに設定します。

(1) パッケージの説明

<パブリック宣言シンボル>

T3_6252 : μPD6252転送サブルーチン名称
 RWRITE : RANDOM WRITEコマンド値
 RREAD : RANDOM READコマンド値
 CREAD : CURRENT READコマンド値
 WADAT : ワード・アドレス格納エリア
 TRNDAT : 送信データ格納エリア
 RCVDAT : 受信データ格納エリア
 CMDDAT : コマンド・データ格納エリア
 BUSYFG : ビジィ状態テスト用フラグ
 CS6252 : μPD6252のCS端子(P32)

<使用するレジスタ>

A

<使用するRAM>

名 称	用 途	属 性	バイト
WAADR	ワード・アドレスを格納(転送開始前)	SADDR	1
TRNDAT	送信データを格納(転送開始前)		
RCVDAT	受信データを格納(転送終了後)		
CMDDAT	コマンド・データを格納(転送開始前)		

<使用するフラグ>

名 称	用 途
BUSYFG	WRITE BUSY状態の設定

<ネスティング>

1 レベル 3 バイト

<使用するハードウェア>

- シリアル・インタフェース・チャンネル 0
- P32

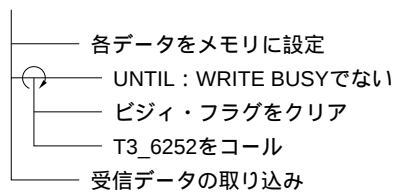
<初期設定>

- シリアル・インタフェース・チャンネル 0 の設定
2 線式シリアルI/Oモード, SB1端子選択 CSIM0 = #10011011B
- シリアル・クロック $f_x/2^5$ TCL3 = # × × × × 1001B
- SB1ラッチをハイ・レベル RELT= 1

<起動方法>

コマンドに対応した必要なデータを設定し, T3_6252をコールします。サブルーチンから復帰後, ビジィ・フラグ (BUSYFG) をテストします。ビジィ・フラグがセットされているときは転送されていませんので, 再度転送する必要があります。受信モードのときは, サブルーチンから復帰後, 受信データがRCVDATに格納されています。

(2) 使用例



```

EXTRN  RWRITE, RREAD, CREAD
EXTRN  WADAT, TRNDAT, RCVDAT, CMDDAT, T3_6252
EXTBIT  BUSYFG, CS6252

```

```

CSIM0=#10011011B      ; 2線式シリアル I/Oモード, S B 1 端子の設定
TCL3=#10011001B      ; S C K 0 = 2 6 2 k H z に設定
CLR1    SB0
CLR1    CS6252          ; μ P D 6 2 5 2 の C S をロウ・レベルに設定
CLR1    PM3. 2

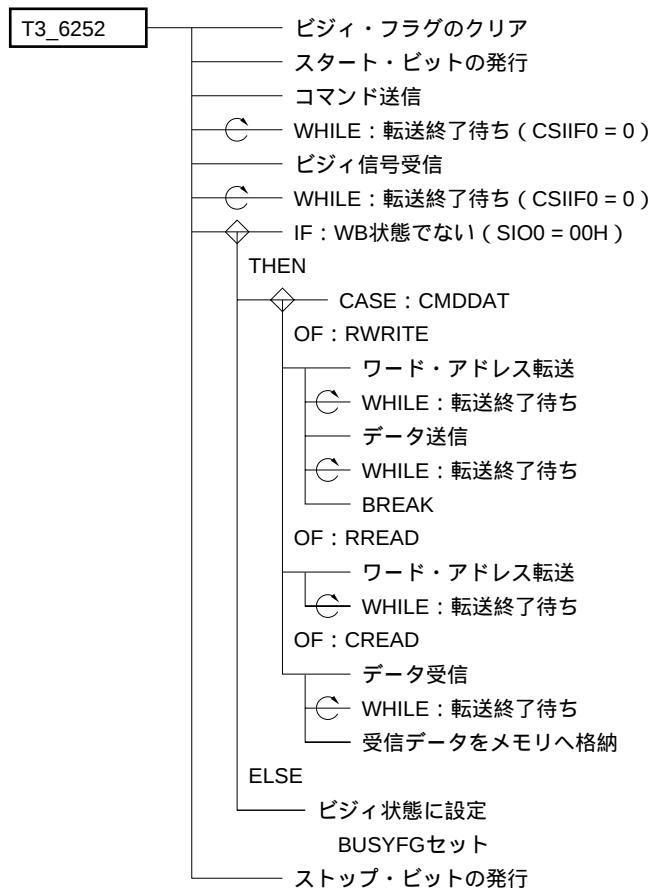
```

```

CMDDAT=A
:
:
WADAT=A
:
:
TRNDAT=A
:
:
repeat
    CLR1    BUSYFG
    CALL    !T3_6252
until_bit(!BUSYFG)
:
:
A=RCVDAT

```

(3) SPDチャート



(4) プログラム・リスト

```

PUBLIC RWRITE, RREAD, CREAD
PUBLIC WADAT, TRNDAT, RCVDAT, CMDDAT, T3_6252
PUBLIC BUSYFG, CS6252
CSI_DAT DSEG SADDR
WADAT: DS 1 ; ワード・アドレス格納エリア
TRNDAT: DS 1 ; 送信データ格納エリア
RCVDAT: DS 1 ; 受信データ格納エリア
CMDDAT: DS 1 ; コマンド・データ格納エリア

CSI_FLG BSEG
BUSYFG DBIT ; ビジィ状態の設定

RWRITE EQU 00H ; RANDOM WRITE モード
RREAD EQU 0C0H ; RANDOM READ モード
CREAD EQU 080H ; CURRENT READ モード
CS6252 EQU 0FF03H.2 ; 0FF03H = PORT 3

CSI_SEG CSEG
;*****
;* μ P D 6 2 5 2 (3 線式) 通信
;*****
T3_6252:
CLR1 BUSYFG
SET1 CS6252 ; スタート・ビット発行
SIO0=CMDDAT (A) ; コマンド転送
while_bit(!CSIIF0) ; 転送終了待ち
endw
CLR1 CSIIF0
SIO0=#0FFH ; ビジィ信号受信開始
while_bit(!CSIIF0) ; 転送終了待ち
endw
CLR1 CSIIF0
if(SIO0==#00H) ; ビジィ・チェック
switch (CMDDAT)
case RWRITE:
SIO0=WADAT (A) ; ワード・アドレス転送
while_bit(!CSIIF0) ; 転送終了待ち
endw
CLR1 CSIIF0
SIO0=TRNDAT (A) ; データ送信開始
while_bit(!CSIIF0) ; 転送終了待ち
endw
CLR1 CSIIF0
break
case RREAD:
SIO0=WADAT (A) ; ワード・アドレス転送
while_bit(!CSIIF0) ; 転送終了待ち
endw
CLR1 CSIIF0
case CREAD:
SIO0=#0FFH ; データ受信開始
while_bit(!CSIIF0) ; 転送終了待ち
endw
CLR1 CSIIF0
RCVDAT=SIO0 (A) ; 受信データ格納
ends

```

```
else
    SET1    BUSYFG          ; ビジィ状態に設定
endif
CLR1      CS6252
RET
```

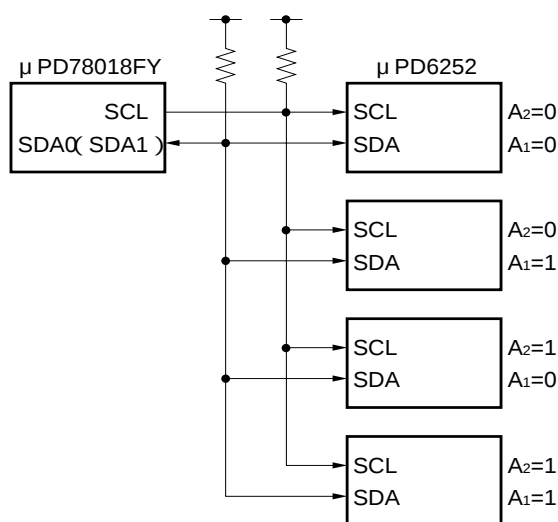
8.1.2 I²Cバス・モードの通信

μPD6252^準の2線式モードは、シリアル・クロック（SCL）とデータ（SDA）の2本で通信をしています。また、I²Cの通信フォーマットにも準拠しています。したがって、μPD78018FYサブシリーズを使用して通信をする場合は、I²Cバス・モードを設定します。

ここでは、μPD78018FYサブシリーズを例に説明します。

注 μPD6252は保守製品です。

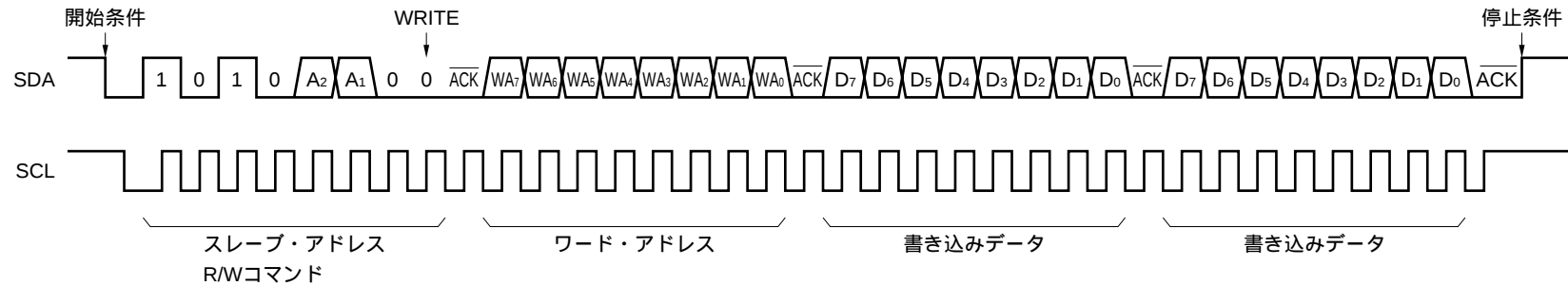
図8 - 15 μPD6252とI²Cバス・モードの接続例



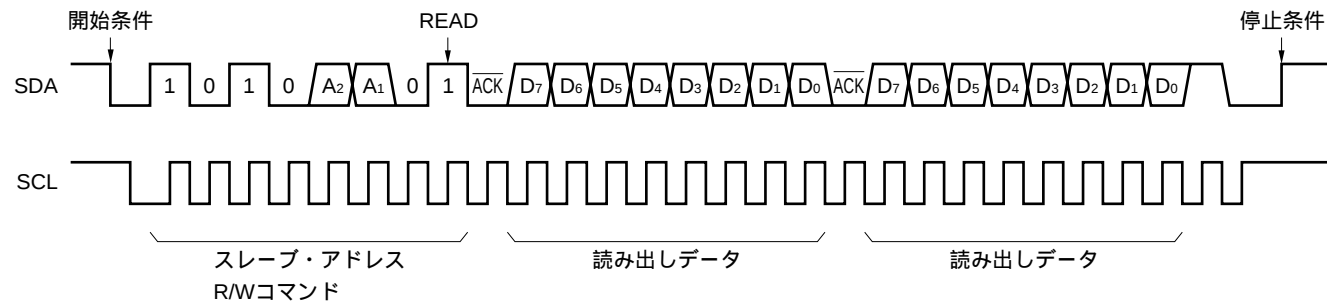
μPD6252へ書き込み、読み出しを行うときの通信フォーマットを図8 - 16に示します。

図8 - 16 μ PD6252動作タイミング

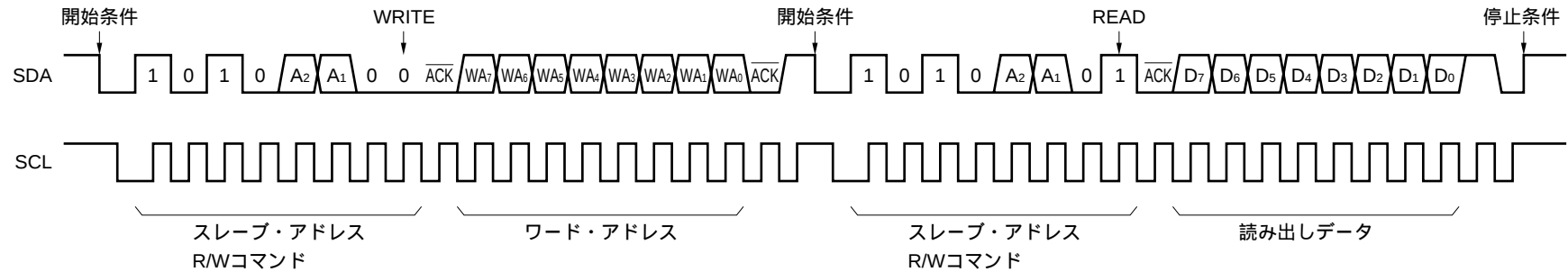
(a) μ PD6252への送信



(b) μ PD6252からの受信 (ワード・アドレス指定なし)



(c) μ PD6252からの受信 (ワード・アドレス指定あり)



から に μ PD6252の通信手順を示します。ここでは、書き込み、読み出しのデータ数は、1バイト固定です。I²Cフォーマットで、マスタが受信の場合、最終データを受信したときは $\overline{\text{ACK}}$ 信号を出力しません。この例では、マスタ側で $\overline{\text{ACK}}$ 信号を出力することはないため、シリアル・バス・インタフェース・コントロール・レジスタ (SBIC) のビット5 (ACKE) は常に0で使します。

通信を開始するために開始条件を設定します。

シリアル・クロック=ハイ・レベルの状態データを立ち下げます。

μ PD6252のスレーブ・アドレス値 (ビット1-ビット7) と書き込み (ビット0=0) / 読み出し (ビット0=1) 選択ビットを送信します。



データを転送します。

●送信モードの場合

- (i) μ PD6252のワード・アドレスを送信します。
- (ii) 書き込みデータを送信します。

●受信モードの場合

読み出しデータを受信します。

通信を終了するために終了条件を設定します。

シリアル・クロック=ハイ・レベルの状態データを立ち上げます。

ワード・アドレスの指定は、書き込みモードのみになっているので、アドレス指定により読み出すときは、一度書き込みモードでアドレスを指定する必要があります。

各データ転送中で、 μ PD6252から $\overline{\text{ACK}}$ 信号が返ってこないときは、通信を中止します。

開始、終了条件は、シリアル・クロックの操作をするときはCLC (割り込みタイミング指定レジスタ (SINT) のビット3) で設定し、データの操作をするときはRELT, CMDT (SBICのビット0, 1) で設定します。

(1) パッケージの説明

<パブリック宣言シンボル>

T2_6252 : μ PD6252転送サブルーチン名称
 WAADR : ワード・アドレス格納エリア
 TRNDAT : 送信データ格納エリア
 RCVDAT : 受信データ格納エリア
 SLVADR : スレーブ・アドレス格納エリア
 BUSYFG : ビジィ状態テスト用フラグ
 WRCHG : 書き込み→読み出しモード変更フラグ
 ERRFG : エラー状態テスト用フラグ

<使用するレジスタ>

A

<使用するRAM>

名 称	用 途	属 性	バイト
WAADR	ワード・アドレスを格納（転送開始前）	SADDR	1
TRNDAT	送信データを格納（転送開始前）		
RCVDAT	受信データを格納（転送終了後）		
SLVADR	スレーブ・アドレスを格納		

<使用するフラグ>

名 称	用 途
BUSYFG	WRITE BUSY状態の設定
WRCHG	書き込み→読み出しモード変更
ERRFG	エラー状態の設定

<ネスティング>

1 レベル 2 バイト

<使用するハードウェア>

シリアル・インタフェース・チャンネル 0

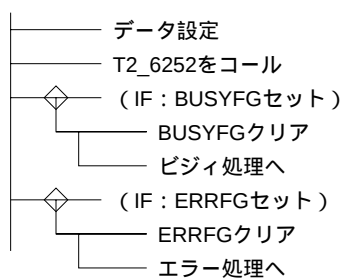
<初期設定>

- シリアル・インタフェース・チャンネル0の設定
2線式シリアルI/Oモード, SB0端子選択 CSIM0=#10011011B
- シリアル・クロック $f_x/2^4$, 16選択 TCL3=#××××1000B
- シリアル・クロック9発立ち上がりで割り込み,
クロック・ラインをハイ・レベルに設定 SINT=#00001011B

<起動方法>

- コマンドに対応した必要なデータを設定し, T2_6252をコールします。受信モードのときは, サブルーチン復帰後に受信データがRCVDATに格納されています。
- 通信開始のときにシリアル・クロックがロウ・レベルだったり(ピジィ状態), データ転送でACKが受信できなかったときは, BUSYFG, ERRFGがセットされるので, メイン処理でテスト, クリアを行ってください。

(2) 使用例



```

EXTRN  WAADR, TRNDAT, RCVDAT, SLVADR, T2_6252
EXTBIT  BUSYFG, WRCHG, ERRFG

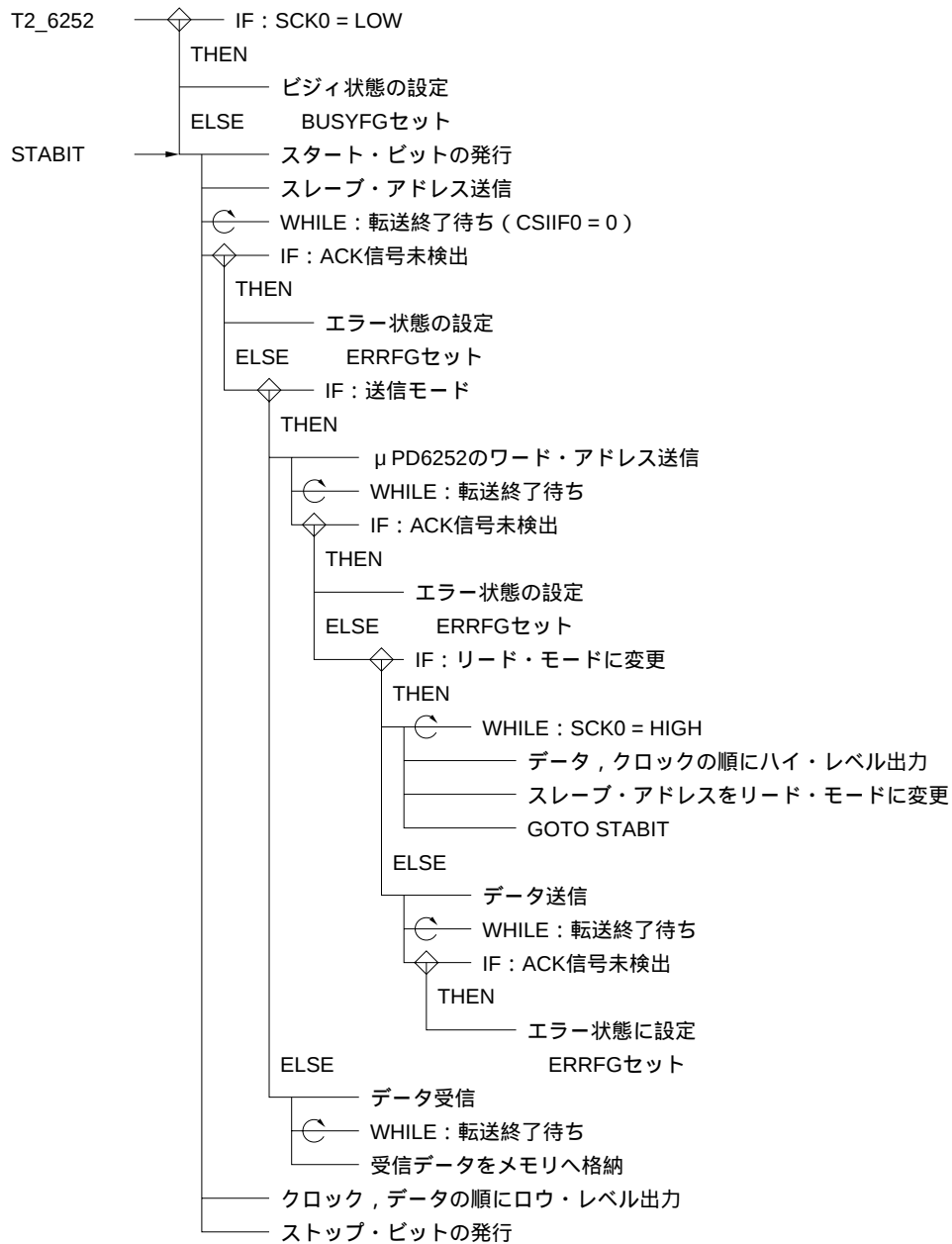
```

```

SET1    SB0
CSIM0=#10011011B          ; シリアル・インタフェース2線式, SB0
SINT=#00001011B          ; I2Cモードに設定
TCL3=#10001000B          ; SCK=32.7kHz
SET1    RELT
SET1    SCK0
CLR1    SB0
...
WAADR=A
...
TRNDAT=A
...
SLVADR=A
CALL    !T2_6252
if_bit(BUSYFG)
    CLR1    BUSYFG
...
endif
...
if_bit(ERRFG)
    CLR1    ERRFG
...
ENDIF

```

(3) SPDチャート



(4) プログラム・リスト

```

PUBLIC WAADR, TRNDAT, RCVDAT, SLVADR, T2_6252
PUBLIC BUSYFG, WRCHG, ERRFG

CSI_DAT DSEG SADDR
WAADR: DS 1 ; ワード・アドレス格納エリア
TRNDAT: DS 1 ; 送信データ格納エリア
RCVDAT: DS 1 ; 受信データ格納エリア
SLVADR: DS 1 ; スレーブ・アドレス格納エリア

CSI_FLG BSEG
BUSYFG DBIT ; ビジィ状態の設定
WRCHG DBIT ; モード変更
ERRFG DBIT ; エラー状態の設定

SCK0 EQU P2.7

CSI_SEG CSEG
;*****
;* μ P D 6 2 5 2 (2線式) 通信
;*****
T2_6252:
    if_bit(!CLD)
        SET1 BUSYFG ; ビジィ状態
    else
STABIT:
        SET1 CMDT ; スタート・ビットの発行
        NOP ; スタート・ビット有効幅ウエイト
        NOP
        NOP
        NOP
        NOP
        CLR1 CLC ; クロックをロウ・レベルに設定
        S100=SLVADR (A) ; スレーブ・アドレス送信開始
        while_bit(!CSIIF0) ; 転送終了待ち
        endw
        CLR1 CSIIF0
        if_bit(!ACKD) ; ACK信号未検出
            SET1 ERRFG
        elseif_bit(!SLVADR.0) ; 送信モード
            S100=WAADR (A) ; ワード・アドレス送信開始
            while_bit(!CSIIF0) ; 転送終了待ち
            endw
            CLR1 CSIIF0
            if_bit(!ACKD) ; ACK信号未検出
                SET1 ERRFG
            elseif_bit(WRCHG)
                while_bit(CLD)
                endw
                SET1 RELT
                SET1 CLC
                while_bit(!CLD) ; クロックのハイ・レベル確認
                endw
                NOP ; クロックのハイ・レベル有効幅ウエイト
                NOP
                NOP
                NOP
                NOP
                NOP
                SET1 SLVADR.0 ; リード・モード・アドレスに変更
                goto STABIT
        else

```

```
EXTRN  WAADR, TRNDAT, RCVDAT, SLVADR, T2_6252
EXTBIT  BUSYFG, WRCHG, ERRFG

SET1    SB0
CSIMO=#10011011B          ; シリアル・インタフェース2線式, SB0
SINT=#00001011B          ; I2Cモードに設定
TCL3=#10001000B          ; SCK=32.7kHz
SET1    RELT
SET1    SCK0
CLR1    SB0
...
WAADR=A
...
TRNDAT=A
...
SLVADR=A
CALL    !T2_6252
if_bit(BUSYFG)
    CLR1    BUSYFG
    ...
endif
...
if|bit(ERRFG)
    CLR1    ERRFG
    ...
ENDIF
```

(6) I²Cバス・モード使用時の制限事項 (μPD78018FYサブシリーズ)

μPD78018FYサブシリーズには、次の制限事項があります。

・I²Cバス・モードのスレーブ・デバイスとして使用する場合の制限事項

対象製品：μPD78011FY, 78012FY, 78013FY, 78014FY, 78015FY, 78016FY, 78018FY, 78P018FY

内 容：シリアル転送状態^注でウェイク・アップ機能を実行 (WUPフラグ (シリアル動作モード・レジスタ0 (CSIM0) のビット5) をセット (1) することにより実行) すると、他のスレーブ・デバイスとマスタ・デバイス間のデータに対してアドレス判定を行ってしまいます。したがって、そのときのデータがμPD78018FYサブシリーズのスレーブ・アドレスと一致すると、μPD78018FYサブシリーズが通信に参加し、通信データを破壊してしまいます。

注 シリアル転送状態とは、シリアルI/Oシフト・レジスタ0 (SIO0) の書き込み後、シリアル転送終了により割り込み要求フラグ (CSIF0) がセット (1) されるまでの状態を示します。

回避策：この制限事項はプログラムを変更することによって回避できます。

ウェイク・アップ機能を実行する前に、次に示すシリアル転送状態を解除するプログラムを実行してください。また、ウェイク・アップ機能を実行するときは、SIO0への書き込み命令を実行しないでください。SIO0への書き込み命令を実行しなくても、ウェイク・アップ機能実行時にデータ受信は可能です。

このプログラムは、シリアル転送状態を解除するためのプログラムです。シリアル転送状態を解除するためには、一度シリアル・インタフェース・チャンネル0を動作停止状態 (CSIE0フラグ (シリアル動作モード・レジスタ (CSIM0) のビット7) をクリア (0)) にする必要があります。ただし、I²Cバス・モードでシリアル・インタフェース・チャンネル0を動作停止状態にすると、SCL端子からはハイ・レベルを、SDA0 (SDA1) 端子からはロウ・レベルを出力してしまい、I²Cバスの通信に影響を与える可能性があります。そのためこのプログラムは、I²Cバスへの影響を避けるためにSCL端子およびSDA0 (SDA1) 端子をハイ・インピーダンス状態にしています。

また、この例ではシリアル・データ入力/出力端子をSDA0 (/P25) として説明しています。シリアル・データ入力/出力端子をSDA1 (/P26) にしている場合は、プログラム中のP2.5, PM2.5をそれぞれP2.6, PM2.6に読み替えてください。

・シリアル転送状態を解除するプログラム例

```

SET1 P2.5      ;
SET1 PM2.5     ;
SET1 PM2.7     ;
CLR1 CSIE0     ;
SET1 CSIE0     ;
SET1 RELT      ;
CLR1 PM2.7     ;
CLR1 P2.5      ;
CLR1 PM2.5     ;

```

の命令でI²Cバス・モードに復帰したときに、SDA0端子からロウ・レベルを出力しないようにします。SDA0端子の出力はハイ・インピーダンス状態になります。

の命令でポート・モードに移行したときに、SDA0ラインに影響を与えないようにするために、P25（/SDA0）端子を入力モードにします。入力モードに変化するタイミングは の命令実行時です。

の命令でポート・モードに移行したときに、SCLラインに影響を与えないようにするために、P27（/SCL）端子を入力モードにします。入力モードに変化するタイミングは の命令実行時です。

I²Cバス・モードからポート・モードに移行します。

ポート・モードからI²Cバス・モードに復帰します。

の命令でSDA0端子からロウ・レベルを出力しないようにします。

I²Cバス・モードでは、P27端子を出力モードにする必要がありますので、P27端子を出力モードにします。

I²Cバス・モードでは、P25端子を出力ラッチに0を設定する必要がありますので、P25端子の出力ラッチに0を設定します。

I²Cバス・モードでは、P25端子を出力モードにする必要がありますので、P25端子を出力モードにします。

備考 RELT：シリアル・バス・インタフェース・コントロール・レジスタ（SBIC）のビット0

8.2 OSD用LSI (μ PD6451A) とのインタフェース

OSD (On Screen Display) 用LSIである μ PD6451Aは、マイコンと組み合わせて使用することにより、ディスプレイ上にVTRのプログラム情報や、TVのチャンネルなどの表示をします。 μ PD6451Aとのインタフェースは、DATA, CLK, STB, BUSYの4本の線により行います。ここでは μ PD78014Hサブシリーズを例に説明します。

図8 - 17 μ PD6451Aとの接続例

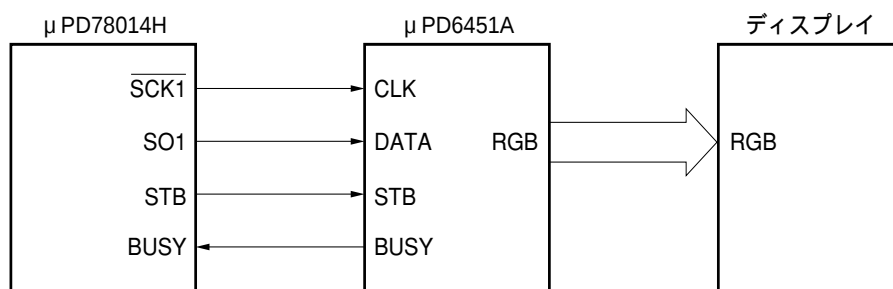
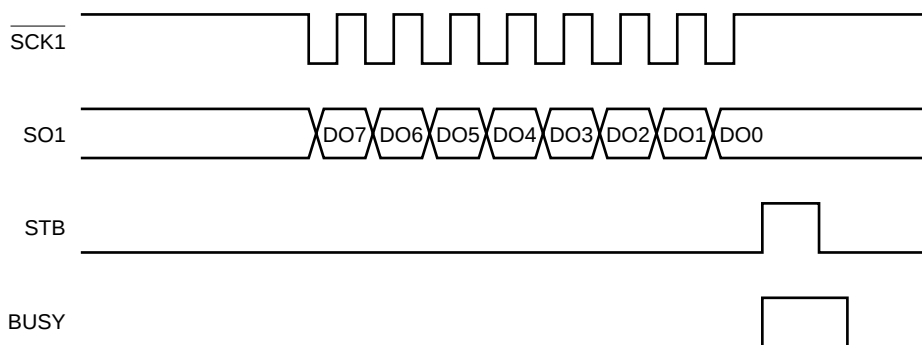


図8 - 18 μ PD6451Aの通信フォーマット



μ PD6451Aとのインタフェースで使用しているハンドシェイク用のストローブ信号（STB）の出力とビジィ信号（BUSY）のテストは、78K/0シリーズのシリアル・インタフェース・チャンネル1で自動的に行います。 μ PD6451Aの通信フォーマットに合わせるため、ストローブ信号出力許可、ビジィ信号入力許可（ハイ・アクティブ）モードを選択します。バッファRAM領域（FAC0H-FADFH）に送信データ（最大32バイト）を、自動データ送受信アドレス・ポインタ（ADTP）に送信データ数を設定することで、複数のデータを連続して自動送信します。

(1) パッケージの説明

<パブリック宣言シンボル>

TR6451 : μ PD6451A転送サブルーチン名称

DTVAL : 送信データ数設定エリア

<使用するレジスタ>

A

<使用するRAM>

名 称	用 途	属 性	バイト
DTVAL	送信データ数を格納	SADDR	1

<ネスティング>

1 レベル 2 バイト

<使用するハードウェア>

- シリアル・インタフェース・チャンネル1

<初期設定>

- シリアル・インタフェース・チャンネル1の設定

自動送受信動作許可, MSB先頭

CSIM1=#10100011B

ビジィ入力許可(ハイ・アクティブ), ストロープ出力許可,

単発モード

ADTC=#00000110B

- シリアル・クロック $f_x/2^5$

TCL3=#1001 × × × × B

- P22出力ラッチをハイ・レベル

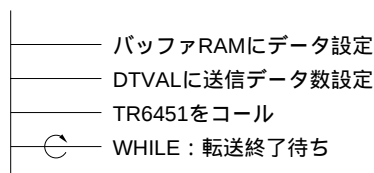
- P21, P22, P23出力モード, P24入力モード

PM2=# × × × 1000 × B

<起動方法>

バッファRAMに送信するデータ（上位アドレスから送信）を，DTVALに送信データ数を設定し，TR6451をコールします。データ転送の終了は，自動データ送受信コントロール・レジスタ（ADTC）のビット3（TRF）をテストすることで確認できます。

(2) 使用例



```

        EXTRN    TR6451, DTVAL

SCK1    EQU      P2.2
        :
        :
        P2=#00000100B
        PM2=#11110001B
        CSIM1=#10100011B      ; 自動送受信機能に設定
        TCL3=#10011001B      ; SCK1 = 262 kHz
        ADTC=#00000110B      ; ストローブ・ビジィ信号あり
        :
        DE=#TABLE1           ; 送信データのテーブル参照アドレス設定
        HL=#0FAC0H           ; バッファRAMの先頭アドレス設定
        B=32                 ; 送信データ数の設定

        while(B>#0)          ; 送信データをバッファRAMに転送
            B--
            [HL+B]=[DE] (A)
            DE++
        endw

        DATVAL=#32           ; 送信データ数の設定
        CALL    !TR6451
        while_bit (TRF)      ; 転送終了待ち
        endw

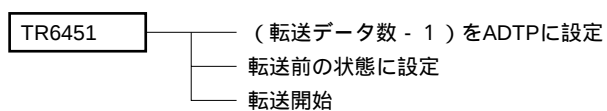
```

TABLE1:

DB	11111111B	; パワーオン・リセット, コマンド1
DB	01000000B	; 垂直アドレス0
DB	11000000B	; 水平アドレス0
DB	10000000B	; 文字サイズ
DB	11111100B	; コマンド0
DB	11101001B	; LC発信ON, 点滅OFF, 表示ON
DB	10001100B	; ブリンクON, 文字: 赤
DB	11011011B	; 色指定, 背景ベタ: シアン
DB	10010101B	; 表示行5
DB	10100000B	; 表示桁0
DB	07H	; 7
DB	08H	; 8
DB	1BH	; K
DB	6DH	; /
DB	00H	; 0
DB	10H	
DB	11H	; A
DB	20H	; P
DB	20H	; P
DB	1CH	; L
DB	19H	; I
DB	13H	; C
DB	11H	; A
DB	24H	; T
DB	19H	; I
DB	00H	; 0
DB	1EH	; N
DB	10H	
DB	1EH	; N
DB	00H	; 0
DB	24H	; T
DB	15H	; E

備考 出力テーブル・データのコマンド, およびデータについては, μ PD6451A データ・シート(資料番号 IC-7812)を参照してください。

(3) SPDチャート



(4) プログラム・リスト

```

PUBLIC TR6451, DTVAL

CSI_DAT DSEG SADDR
DTVAL: DS 1 ; データ数設定エリア

CSI_SEG CSEG
;*****
;* μ P D 6 4 5 1 A通信
;*****
TR6451:
    A=DTVAL ; データ数の設定
    A--
    ADTP=A
    SIO1=#0FFH ; 転送開始
    RET

```

8.3 SBIモードのインタフェース

78K/0シリーズは、NECシリアル・バス・フォーマット準拠のSBIモードを持っています。SBIモードは、クロックとデータの2線により、1つのマスタCPUが複数のスレーブCPUと通信できます。ここではμPD78014Hサブシリーズを例に説明します。

SBIモードを使用するときの接続例を図8 - 19に、通信フォーマットを図8 - 20に示します。

図8 - 19 SBIモードの接続例

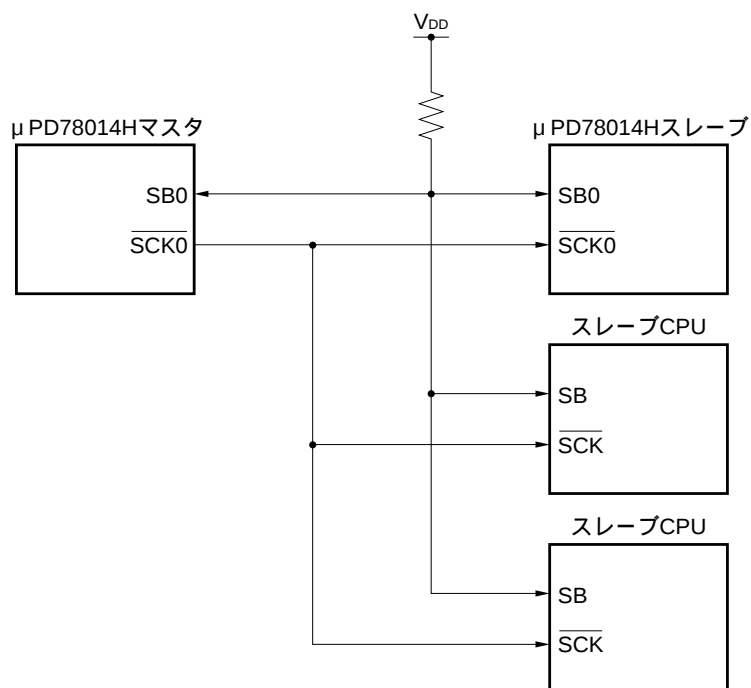
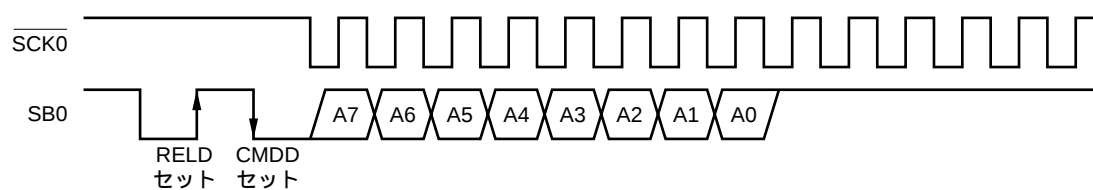
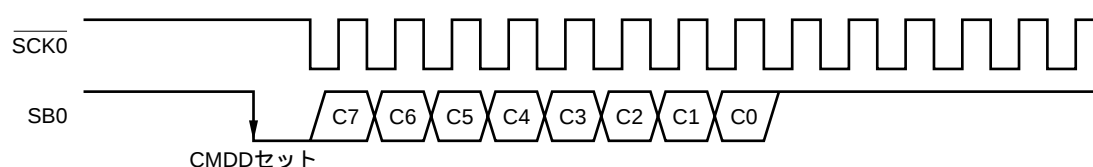


図8 - 20 SBIモードの通信フォーマット

(a) アドレス送信



(b) コマンド送信



(c) データ送受信

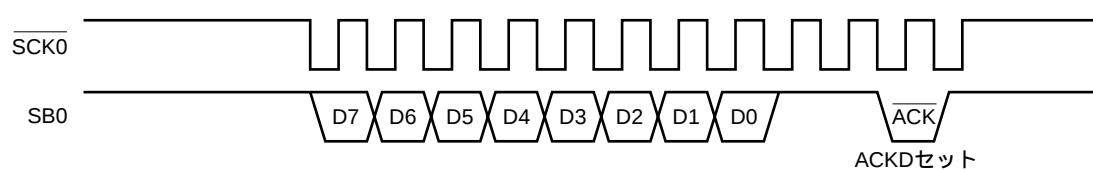


表8 - 5 SBIモードの信号一覧

信号名	出力側	意 味
アドレス	マスタ	スレーブ・デバイスの選択
コマンド	マスタ	スレーブ・デバイスへの指示
データ	マスタ/スレーブ	スレーブ、またはマスタが処理するデータ
クロック	マスタ	シリアル・データの送受信同期信号
ACK	受信側 ^注	受信応答信号
BUSY	スレーブ	通信不可能状態

注 正常動作時は、受信側が出力しますがタイム・アウト処理のようなエラー発生時は、マスタCPUが出力します。

8.3.1 マスタCPUとしての応用

スレーブCPUに対して (a) から (d) の処理を行います。

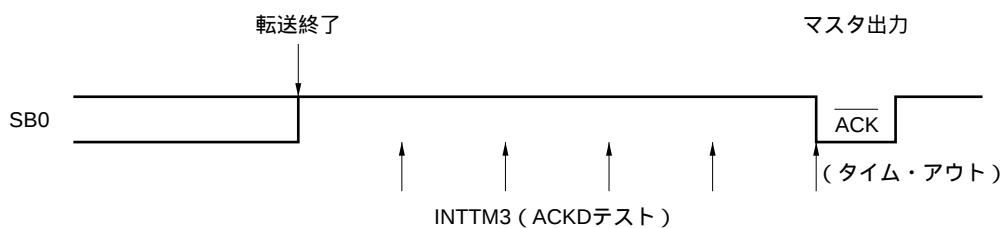
- (a) アドレス送信
- (b) コマンド送信
- (c) データ送信
- (d) データ受信

(a) から (d) の通信において , , のエラーをチェックします。

タイム・アウト処理

マスタCPUが送信したとき , 一定時間内 (ここでは , 時計用タイマの割り込み要求 5 回発生以内) に $\overline{\text{ACK}}$ 信号が返ってこない場合はエラーと判断し , マスタCPUが $\overline{\text{ACK}}$ 信号を出力して処理を終了します。

図 8 - 21 タイム・アウトの $\overline{\text{ACK}}$ 信号



バス・ラインのテスト

マスタCPUは , 送信データをシリアルI/Oシフト・レジスタ 0 (SIO0) とスレーブ・アドレス・レジスタ (SVA) に設定することにより , バス・ラインに正しくデータが出力されたかをテストします。バス・ラインのデータは , SIO0で受信されますので , 転送終了後のシリアル動作モード・レジスタ 0 (CSIM0) のビット 6 (COI) (SIO0とSVAの一致でセット) のテストにより , 正常にデータが出力されたことを確認します。

図 8 - 22 バス・ラインのテスト

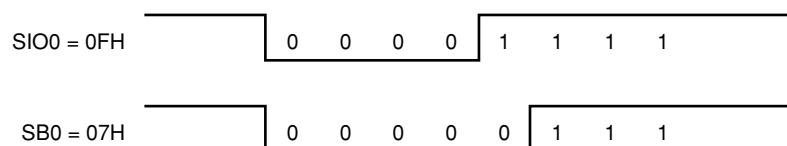


図 8 - 22では , 転送終了後の値が一致しない (SIO0=07H, SVA=0FH) ので , COI=0となりバス・ラインでエラーが発生します。

(1) パッケージの説明

<パブリック宣言シンボル>

M_TRANS : マスタSBI転送サブルーチン名称
 TR_MODE : 転送モードの選択値の格納エリア
 TRNDAT : 送信データ格納エリア
 RCVDAT : 受信データ格納エリア
 TRADR : アドレス送信モードの選択値
 TRCMD : コマンド送信モードの選択値
 TRDAT : データ送信モードの選択値
 RCDAT : データ受信モードの選択値
 ERRORF : エラー状態テスト用フラグ

<使用するレジスタ>

サブルーチン A

<使用するRAM>

名 称	用 途	属 性	バイト
TR_MODE	転送モードの選択値を格納	SADDR	1
ACKCT	タイム・アウト用カウンタ		
TRNDAT	送信データを格納		
RCVDAT	受信データを格納		

<使用するフラグ>

名 称	用 途
RCVFLG	受信モードの設定
BUSYFG	ビジィ状態の設定
ERRORF	エラー状態の設定
ACKWFG	ACK信号待ち状態の設定

<ネスティング>

2レベル5バイト

<使用するハードウェア>

- シリアル・インタフェース・チャンネル0
- 時計用タイマ

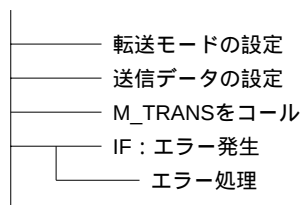
<初期設定>

- シリアル・インタフェース・チャンネル0の設定
SBIモード, SB1端子選択 CSIM0=#10010011B
- シリアル・クロック $f_x/2^5$ TCL3=# × × × × 1001B
- SO0ラッチをハイ・レベル RELT=1
- P27出力ラッチをハイ・レベル P27=1
- 時計用タイマのインターバル1.95 ms TMC2=#00100110B
- 時計用タイマ割り込み許可

<起動方法>

転送モードに必要なデータを設定し, M_TRANSをコールします。サブルーチンから復帰後, エラー・フラグ (ERRORF) のテストにより, 転送エラーの有無を判断できます。また, 受信モードのときは, サブルーチンから復帰後, 受信データがRCVDATに格納されています。

(2) 使用例

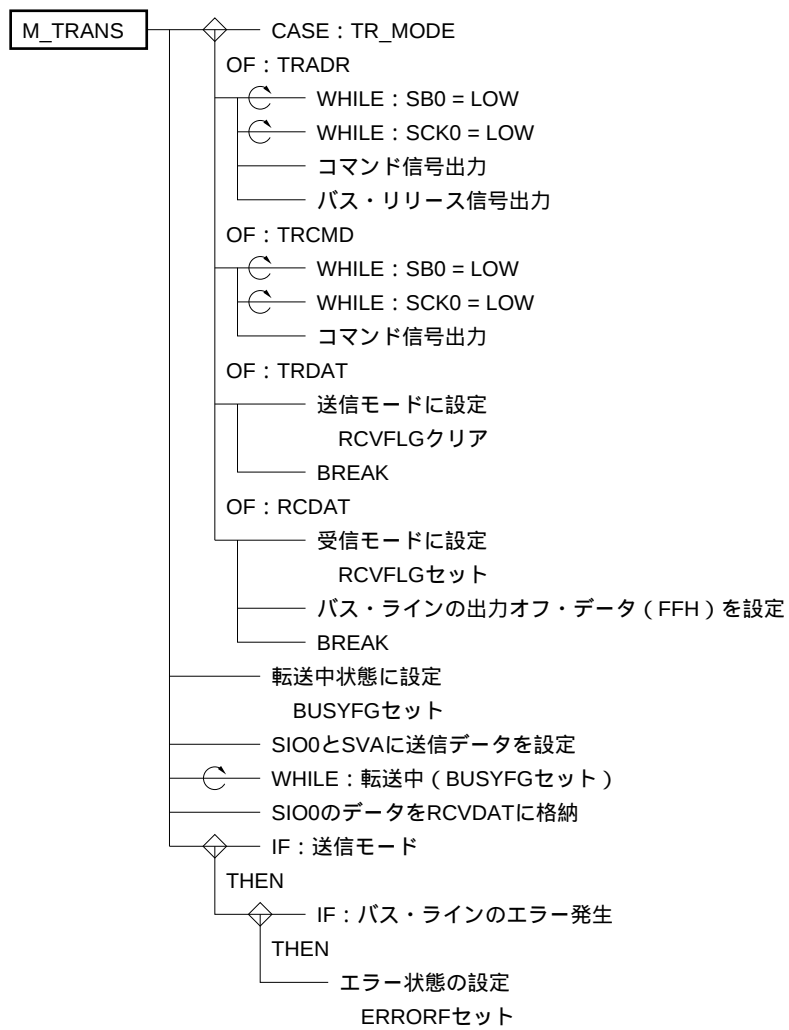


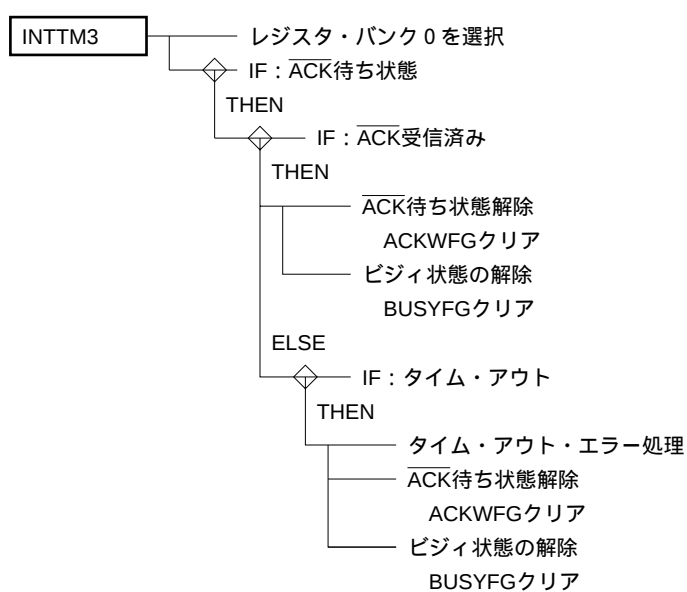
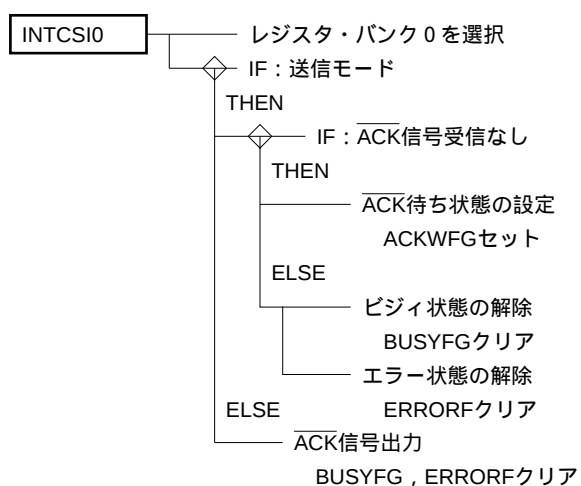
```

EXTRN  M_TRANS, TR_MODE, TRADR, TRCMD, TRDAT, RCDAT
EXTRN  TRNDAT, RCVDAT
EXTBIT  ERRORF

SCK0   EQU      P2.7
SB1    EQU      P2.5
      :
      :
SET1    SB1
CSIM0=#10010111B      ; SBIモードで動作
TCL3=#10011001B       ; SCK0 = 262 kHz
TMC2=#00100110B       ; 時計用タイマのインターバル 1.95 ms に設定
CLR1    BSYE          ; ビジィ信号出力禁止
SET1    RELT          ; 出力ラッチをセット
SET1    SCK0
CLR1    SB1
CLR1    CSIMK0         ; シリアル・インタフェース・チャンネル0 割り込み許可
CLR1    TMMK3          ; 時計用タイマ割り込み許可
EI              ; マスタ割り込み許可
      :
      :
TR_MODE=#TRADR
TRNDAT=#5AH
CALL    !M_TRANS
if_bit(ERRORF)
      エラー処理
endif
  
```

(3) SPDチャート





(4) プログラム・リスト

```

PUBLIC  M_TRANS, TR_MODE, TRADR, TRCMD, TRDAT, RCDAT
PUBLIC  TRNDAT, RCVDAT, ERRORF

VECS10  CSEG    AT 0EH
        DW      INTCS10      ; シリアル・インタフェース・チャンネル0のベクタ・アドレスの設定
VETM3   CSEG    AT 12H
        DW      INTTM3       ; 時計用タイマのベクタ・アドレスの設定

SBI_DAT DSEG    SADDR
TRNDAT: DS      1            ; 送信データ
RCVDAT: DS      1            ; 受信データ
TR_MODE: DS     1            ; 転送モード設定
ACKCT:  DS      1            ; ACKタイム・アウト・カウント

SBI_FLG BSEG
RCVFLG  DBIT      ; 受信モード設定
BUSYFG  DBIT      ; 転送中状態
ERRORF  DBIT      ; エラー表示
ACKWFG  DBIT      ; ACK待ち状態

SB0      EQU      P2.5
SCK0     EQU      P2.7

TRADR    EQU      1          ; アドレス送信モード選択
TRCMD    EQU      2          ; コマンド送信モード選択
TRDAT    EQU      3          ; データ送信モード選択
RCDAT    EQU      4          ; データ受信モード選択

```

```

;*****
;*      S B I データ転送処理
;*****
SBI_SEG CSEG
M_TRANS:
    switch(TR_MODE)
    case TRADR:
        SET1    PM2.5
        while_bit(!SB0)                ; S B 0 = ハイ?
        CLR1    PM2.5
        endw
        while_bit(!SCK0)                ; S C K = ハイ?
        endw
        SET1    CMDT                    ; コマンド信号出力
        NOP                                           ; ウェイト
        SET1    RELT                    ; バス・リリース信号出力
        A=#TRCMD
    case TRCMD:
        SET1    PM2.5
        while_bit(!SB0)                ; S B 0 = ハイ?
        CLR1    PM2.5
        endw
        while_bit(!SCK0)                ; S C K = ハイ?
        endw
        SET1    CMDT                    ; コマンド信号出力
        A=#TRDAT
    case TRDAT:
        CLR1    RCVFLG                ; 送信モードに設定
        A=TRNDAT                      ; 送信データの設定
        break
    case RCDAT:
        SET1    RCVFLG                ; 受信モードに設定
        MOV     A, #OFFH              ; 受信バッファ・オフ
        break
    ends

    SET1    BUSYFG                    ; 転送中状態に設定
    SVA=A                             ; バス・ライン・テスト用
    SIO0=A                             ; 転送開始

    while_bit(BUSYFG)                ; 転送中
    endw
    RCVDAT=SIO0 (A)                  ; 受信データ格納
    if_bit(!RCVFLG)                  ; 送信モード
        if_bit(!COL)                  ; バス・ライン出力NG
            SET1    ERRORF            ; エラー状態に設定
        endif
    endif
    RET

```

```

;*****
;*      I N T C S I O 割り込み処理
;*****
CSI_SEG CSEG
INTCS10:
    SEL RBO
    if_bit(!RCVFLG)                ; 送信モード
        if_bit(!ACKD)              ; アクノリッジ信号受信なし
            ACKCT=#5               ; アクノリッジ信号待ち状態の設定
            SET1    ACKWFG
        else
            CLR1    BUSYFG          ; ビジィ状態の解除
            CLR1    ERRORF          ; エラー状態の解除
        endif
    else
        SET1    ACKT                ; アクノリッジ信号出力
        CLR1    BUSYFG              ; ビジィ状態の解除
        CLR1    ERRORF              ; エラー状態の解除
    endif
    RETI

;*****
;*      タイム・アウト処理
;*****
TM3_SEG CSEG
INTTM3:
    SEL RBO
    if_bit(ACKWFG)                  ; アクノリッジ信号待ち状態?
        if_bit(ACKD)                ; アクノリッジ信号受信済み?
            CLR1    ACKWFG           ; アクノリッジ信号待ち状態解除
            CLR1    BUSYFG           ; ビジィ状態の解除
        else
            ACKCT--
            if(ACKCT==#0)             ; タイム・アウト?
                SET1    ACKT          ; タイム・アウト・エラー処理
                SET1    ERRORF
                CLR1    ACKWFG        ; アクノリッジ信号待ち状態解除
                CLR1    BUSYFG        ; ビジィ状態解除
            endif
        endif
    endif
endif

```

8.3.2 スレーブCPUとしての応用

マスタCPUからのアドレス／コマンド／データの受信とマスタCPUへのデータを送信します。

この例では、ウエイク・アップ機能を使用してアドレスを受信します。ウエイク・アップ機能とは、SBIモード使用時、マスタが送信したアドレス値とスレーブ・アドレス・レジスタ（SVA）に設定された値が一致したときのみ、割り込み要求信号が発生する機能です。したがって、マスタCPUから選択されたスレーブCPUのみINTCSIOが発生し、選択されていないスレーブCPUは不用意な割り込み要求を発生させずに動作させることができます。

スレーブCPUは、選択されたときにウエイク・アップ機能を解除し（転送終了で割り込み要求信号発生）、マスタCPUとインタフェースします。また、通信中のアドレス／コマンド／データの判別は、RELD, CMDDフラグ（シリアル・バス・インタフェース・コントロール・レジスタ（SBIC）のビット2，3）をテストして行います。

なお、スレーブCPUを選択していない状態に自動的に戻らないので、マスタ-スレーブ間で、コマンド処理などにより、非選択状態に戻すプログラムを作る必要があります。

（１）パッケージの説明

<パブリック宣言シンボル>

RCVDAT：受信データ格納エリア

<使用するレジスタ>

バンク0：A

<使用するRAM>

名 称	用 途	属 性	バイト
RCVDAT	受信データを格納	SADDR	1

<使用するフラグ>

名 称	用 途
RCVFLG	受信モードの設定

<ネスティング>

1 レベル3 バイト

<使用するハードウェア>

- シリアル・インタフェース・チャンネル0

<初期設定>

- シリアル・インタフェース・チャンネル0の設定
SBIモード，SB1端子，ウエイク・アップ・モード，
シリアル・クロックを外部クロック入力 CSIM0=#10010011B
- 同期ビジ信号出力 BYSE=1
- SO0ラッチをハイ・レベル RELT=1
- スレーブ・アドレス SVA=#SLVADR
- シリアル・インタフェース・チャンネル0 割り込み許可

< 起動方法 >

INTCSI0の発生により、割り込み処理を起動します。割り込み処理では、以下の処理を行っています。

- アドレス / コマンド / データの判別
- ACK信号の出力
- 受信データをRCVDATに格納

(2) 使用例

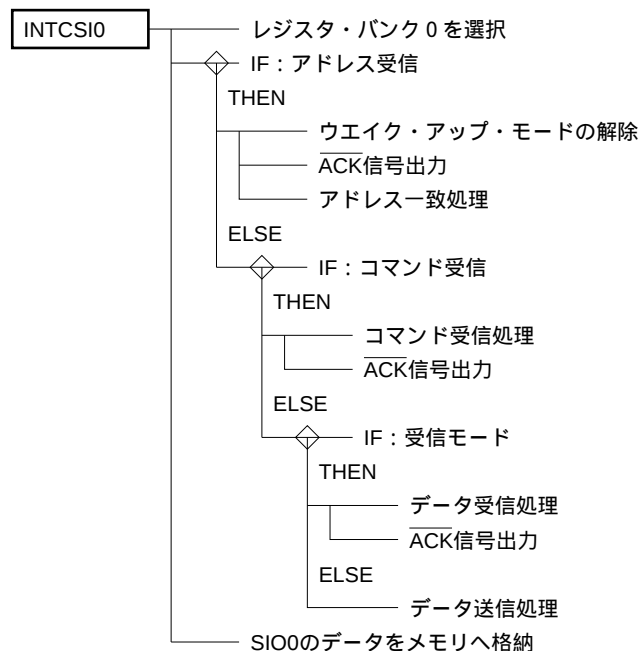
```

EXTRN  RCVDAT
EXTBIT  RCVFLG

SLVADR EQU    5AH
SB1     EQU    P2.5
:
:
SET1    SB1
CSIM0=#10110100B      ; 外部クロック入力, S B 1 端子, ウェイク・アップ・モード選択
SET1    RELT          ; 出力ラッチをハイ・レベルに設定
SET1    BSYE          ; ビジィ自動出力モードに設定
SVA=#SLVADR           ; スレーブ・アドレスの設定
SIO0=#0FFH            ; シリアル転送スタート指示
CLR1    SB1
CLR1    CSIMK0        ; シリアル・インタフェース・チャンネル0 割り込み許可
EI                      ; マスタ割り込み許可

```

(3) SPDチャート



(4) プログラム・リスト

```

VECS10  CSEG      AT 0EH
        DW        INTCS10                ; シリアル・インタフェース・チャンネル0の
                                           ; ベクタ・アドレス設定

CSI_DAT  DSEG      SADDR
RCVDAT:  DS        1                    ; 受信データ格納エリア

CSI_FLG  BSEG
RCVFLG   DBIT                      ; 受信モード設定

CSI_SEG  CSEG
;*****
;*      I N T C S I O 割り込み処理
;******
INTCS10:
        SEL RBO
        if_bit(RELD)                    ; アドレス受信へ
            CLR1      WUP                ; ウェイク・アップ・モード解除
            SET1      ACKT                ; アクノリッジ信号出力
;      ユーザ処理 (アドレス受信)

;*****

        elseif_bit(CMDD)                 ; コマンド受信へ
;      ユーザ処理 (コマンド受信)

        SET1      ACKT                    ; アクノリッジ信号出力
    else

        if_bit(RCVFLG)
;      ユーザ処理 (データ受信処理)
            SET1      ACKT                ; アクノリッジ信号出力
        else
;      ユーザ処理 (データ送信処理)
        endif
;*****
    endif
    RCVDAT=S100 (A)

    RETI

```

8.4 3線式シリアルI/Oモードのインタフェース

78K/0シリーズのシリアル・チャンネル0の3線式シリアルI/Oモード（シリアル・クロック，データ入力，データ出力）の機能を使用して，マスター・スレーブ間で通信を行います。この例では，ハンドシェーク用の信号として，ピジィ信号を1本追加したマスター・スレーブの同時送受信を示します。ピジィ信号はロウ・アクティブとし，スレーブ側が出力します。また，データは8ビット，MSB先頭で通信します。ここではμPD78014Hサブシリーズを例に説明しています。

図8 - 23 3線式シリアルI/Oモードの接続例

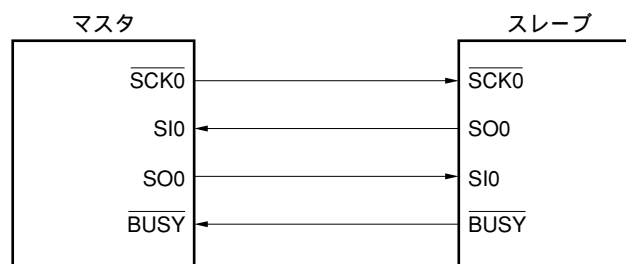
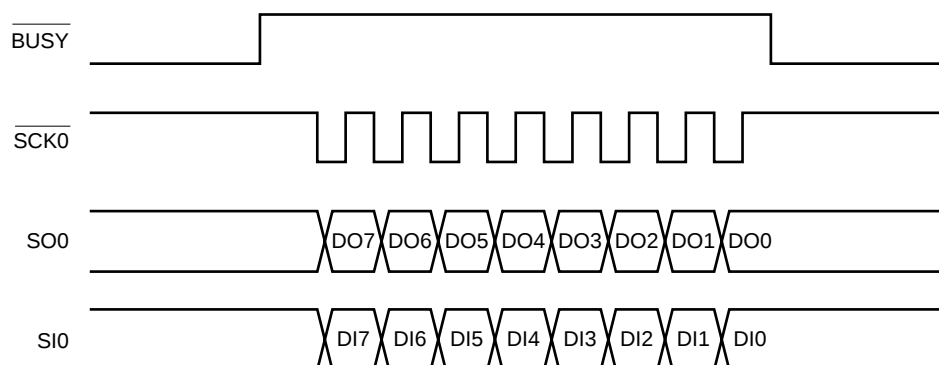


図8 - 24 3線式シリアルI/Oモードの通信フォーマット



8.4.1 マスタCPUとしての応用

シリアル・クロックを $fx/2^5$ に設定し、このシリアル・クロックに同期してスレーブCPUとの通信を行います。

マスタCPUは、送信データ設定後、転送を開始します。ただし、スレーブCPUがビジィ状態（ビジィ信号がロウ・レベル）のときは転送せず、ビジィ・フラグ（BUSYFG）をセットします。

（１）パッケージの説明

<パブリック宣言シンボル>

TRANS : マスタ3線式転送サブルーチン名称
 TDATA : 送信データ格納エリア
 RDATA : 受信データ格納エリア
 BUSY : ビジィ信号入力用ポート
 TREND : 転送終了テスト用フラグ
 BUSYFG : ビジィ状態テスト用フラグ

<使用するレジスタ>

割り込み バンク0 A
 サブルーチン A

<使用するRAM>

名 称	用 途	属 性	バイト
TDATA	送信データを格納	SADDR	1
RDATA	受信データを格納		

<使用するフラグ>

名 称	用 途
TREND	転送終了状態の設定
BUSYFG	ビジィ状態の設定

<ネスティング>

2レベル5バイト

<使用するハードウェア>

- シリアル・インタフェース・チャンネル0
- P33

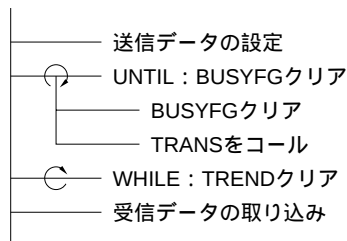
<初期設定>

- シリアル・インタフェース・チャンネル0の設定
3線式シリアルI/Oモード，MSB先頭 CSIM0=#10000011B
- シリアル・クロック $fx/2^5$ TCL3=# × × × × 1001B
- P27出力ラッチをハイ・レベル P27=1
- P33入力モード
- シリアル・インタフェース・チャンネル0割り込み許可

<起動方法>

送信データをTDATAに設定し、TRANSをコールします。サブルーチンから復帰後ビジィ・フラグ（BUSYFG）をテストします。ビジィ・フラグがセットされているときは、転送されていませんので再度転送する必要があります。また、ビジィ・フラグがクリアされているときは、転送が終了していますので受信データがRDATAに格納されています。

(2) 使用例

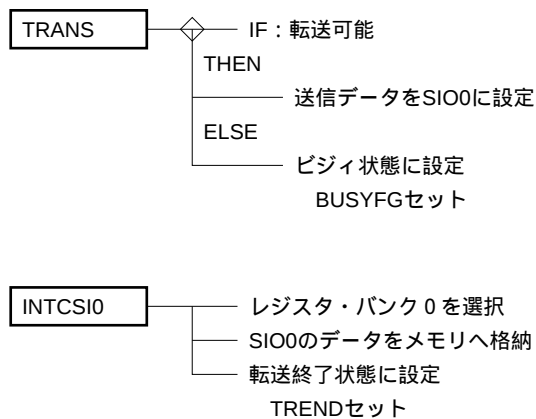


```

EXTRN  TDATA, RDATA, TRANS
EXTBIT  TREND, BUSYFG, BUSY

SCK0    EQU    P2.7
:
:
CSIM0=#10000011B      ; 3線式シリアルI/Oモード, MSB先頭に設定
TCL3=#10011001B      ; SCK0=262kHzに設定
SET1     SCK0
SET1     PM3.3        ; P3.3入力モード
CLR1     CSIMK0        ; シリアル・インタフェース・チャンネル0割り込み許可
EI
:
:
TDATA=A                ; 送信データの設定
repeat
    CLR1     BUSYFG    ; ビジィ・テスト
    CALL     !TRANS
until_bit(!BUSYFG)
while_bit(!TREND)      ; 転送終了
endw
A=RDATA                ; 受信データの取り込み
  
```

(3) SPDチャート



(4) プログラム・リスト

```

PUBLIC TRANS, RDATA, TDATA, BUSY, TREND, BUSYFG
VECSIO CSEG AT 0EH
DW INTCSIO ; シリアル・インタフェース・チャンネル0の
; ベクタ・アドレス設定
BUSY EQU OFF03H.3 ; 0FF03H=PORT3

CSI_DAT DSEG SADDR
RDATA: DS 1 ; 受信データ格納エリア
TDATA: DS 1 ; 送信データ格納エリア

CSI_FLG BSEG
TREND DBIT ; 転送終了状態設定
BUSYFG DBIT ; ビジィ状態設定

CSI_SEG CSEG

;*****
;:* INTCSIO 割り込み処理
;*****
INTCSIO:
    SEL RBO
    RDATA=SIO0 (A) ; 受信データを格納
    SET1 TREND ; 転送終了状態に設定
    RETI

;*****
;:* 3線式(マスタ)
;*****
TRANS:
    if_bit(BUSY) ; 転送可能状態
        SIO0=TDATA (A) ; 送信データの設定
    else
        SET1 BUSYFG ; ビジィ状態に設定
    endif
    RET

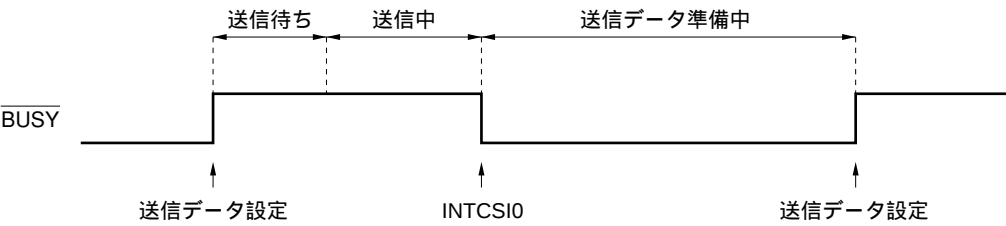
```

8.4.2 スレーブCPUとしての応用

マスタCPUからのシリアル・クロックに同期し、8ビット・データを同時に送受信します。スレーブCPUが出力するビジー信号は、送信データの準備をしている間、ロウ・レベル（ビジー状態）を出力します。このビジー信号の出力タイミングは、送信データの設定（CALL !TRANS）により、ビジー信号を解除（ハイ・レベル）し、転送終了時のINTCSI0の割り込み処理により、ビジー信号（ロウ・レベル）を出力します。

したがって、転送終了後からデータ設定までの間がビジー状態になります。

図8 - 25 ビジー信号の出力



(1) パッケージの説明

<パブリック宣言シンボル>

- TRANS：スレーブ3線式転送サブルーチン名称
- TDATA：送信データ格納エリア
- RDATA：受信データ格納エリア
- BUSY：ビジー信号出力用ポート
- TREND：転送終了テスト用フラグ

<使用するレジスタ>

- 割り込み バンク 0 A
- サブルーチン A

<使用するRAM>

名 称	用 途	属 性	バイト
TDATA	送信データを格納	SADDR	1
RDATA	受信データを格納		

<使用するフラグ>

名 称	用 途
TREND	転送終了状態の設定

<ネスティング>

- 2 レベル 5 バイト

<使用するハードウェア>

- シリアル・インタフェース・チャンネル0
- P33

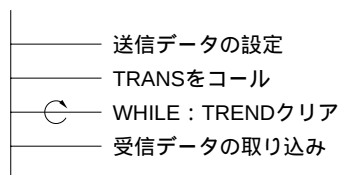
<初期設定>

- シリアル・インタフェース・チャンネル0の設定
3線式シリアルI/Oモード，MSB先頭，外部クロック入力 CSIM0=#10000000B
- P33出力モード P33=0
- ビジィ状態の設定
- シリアル・インタフェース・チャンネル0 割り込み許可

<起動方法>

送信データをTDATAに設定し，TRANSをコールします。TRANSの処理で，ビジィ信号を解除するので，マスタCPUとの通信待ち状態になります。通信終了後は，INTCSI0の発生により割り込み処理が起動されます。転送終了は，TRENDをテストすることで確認できます。TRENDセット後は，受信データがRDATAに格納されています。

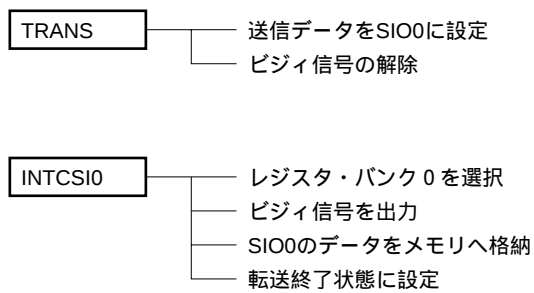
(2) 使用例



```

EXTRN  TDATA, RDATA, TRANS
EXTBIT  TREND, BUSY
:
:
CSIM0=#10000000B      ; 3線式シリアルI/Oモード，MSB先頭に設定
CLR1    BUSY          ; ビジィ状態
CLR1    PM3.3         ; P3.3出力モード
CLR1    CSIMK0        ; シリアル・インタフェース・チャンネル0 割り込み許可
EI
:
:
TDATA=A               ; 送信データの設定
CALL    !TRANS
while_bit(!TREND)     ; 転送終了
endw
A=RDATA               ; 受信データの取り込み
  
```

(3) SPDチャート



(4) プログラム・リスト

```

PUBLIC RDATA, TDATA, BUSY, TREND, BUSYFG
PUBLIC TRANS
VECSIO CSEG AT 0EH
DW INTCSIO ; シリアル・インタフェース・チャンネル0の
           ; ベクタ・アドレス設定

CSI_DAT DSEG SADDR
RDATA: DS 1 ; 受信データ格納
TDATA: DS 1 ; 送信データ格納

CSI_FLG BSEG
TREND DBIT ; 転送終了状態設定
BUSYFG DBIT ; ビジィ状態設定

BUSY EQU 0FF03H.3 ; 0FF03H = PORT 3

CSI_SEG CSEG
;*****
;* INTCSIO 割り込み処理
;*****
INTCSIO:
    SEL RBO
    CLR1 BUSY ; ビジィ状態に設定
    RDATA=SIO0 (A) ; 受信データ格納
    SET1 TREND ; 転送終了状態に設定
    RETI

;*****
;* 3線式 (スレーブ)
;*****
TRANS:
    SIO0=TDATA (A) ; 送信データの設定
    SET1 BUSY ; ビジィ状態を解除
    RET

```

8.5 半二重調歩同期通信

クロック同期式シリアル・インタフェース・チャンネル0を用いて、半二重調歩同期通信を行います。3線式シリアルI/OモードとSBIモードを使用した、2とおりの応用例があります。ここではμPD78014Hサブシリーズを例に説明しています。通信プロトコルは、次のとおりです。

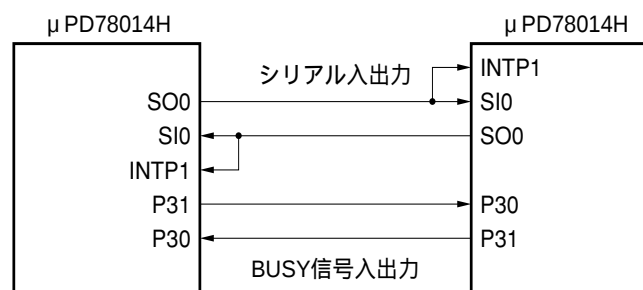
転送速度 : 9600 bps
 スタート・ビット : 1ビット
 キャラクタ長 : 8ビット (LSBファースト)
 パリティ・ビット : 1ビット (偶数 / 奇数パリティ選択可)
 ストップ・ビット : 2ビット

シリアル・クロックは、転送速度を9600 bpsに設定していますので、8ビット・タイマ/イベント・カウンタ2を使用して生成しています。

8.5.1 3線式シリアルI/Oモードの半二重調歩同期通信

図8-26にシステム構成を示します。シリアル入出力は、SI0端子およびSO0端子により行います。ポート3の0ビット、1ビットは、BUSY信号入出力として使用しています。BUSY信号が ' L ' のとき、シリアル通信が可能です。

図8-26 システム構成図 (3線式シリアルI/Oモード)



(1) 3線式シリアルI/Oモードによる送信

データ送信の処理を以下に示します。

スタート・ビット→シリアル・インタフェースの出力ラッチの操作と8ビット・タイマ/イベント・カウンタ2による送信時間待ち

注意 スタート・ビットの取りこぼしによりデータ受信のタイミングが遅れるのを防ぐため、INTP1割り込み要求を高優先にしてください。

データ →シリアル・バッファによる送信

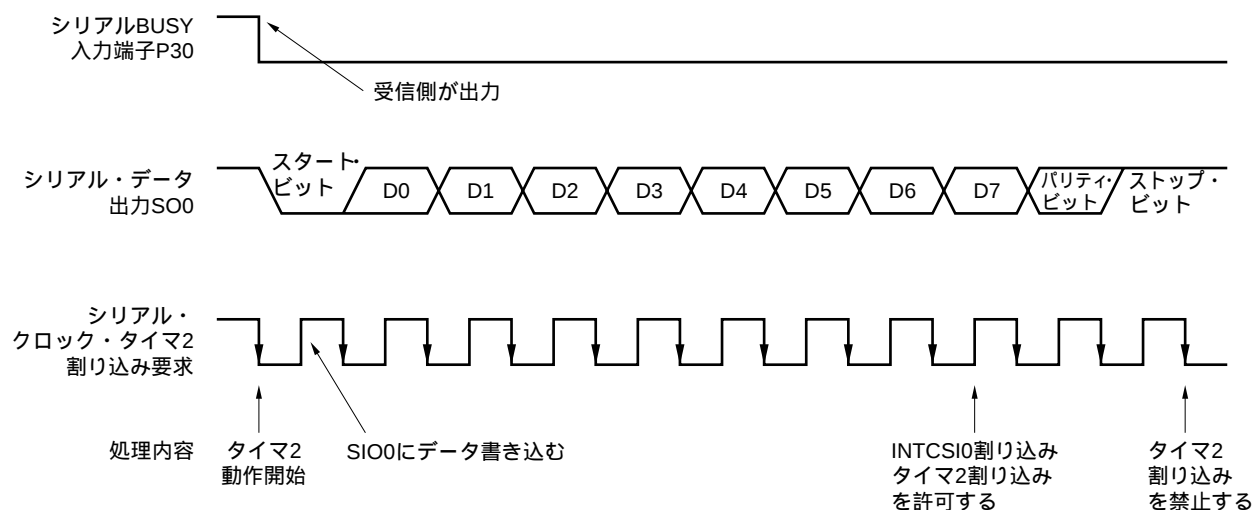
パリティ・ビット→8ビット・タイマ/イベント・カウンタ2の割り込み処理でシリアル・インタフェースの出力ラッチを操作し、パリティ・ビットを出力

注意 送信タイミングが遅れるのを防ぐために、8ビット・タイマ/イベント・カウンタ2割り込み要求を高優先にしてください。

ストップ・ビット→8ビット・タイマ/イベント・カウンタ2の割り込み処理でシリアル・インタフェースの出力ラッチをセットし、ストップ・ビットを出力

注意 送信タイミングが遅れるのを防ぐために、8ビット・タイマ/イベント・カウンタ2割り込み要求を高優先にしてください。

図8-27 3線式シリアルI/Oモードの送信フォーマット



(2) 3線式シリアルI/Oモードによる受信

データ受信の処理を以下に示します。

スタート・ビット→INTP1端子の立ち下がり検出とポート・テストにより受信開始。

注意 スタート・ビットの取りこぼしによりデータ受信のタイミングが遅れるのを防ぐため、INTP1割り込み要求を高優先にしてください。

データ →シリアル・バッファによる受信

パリティ・ビット→8ビット・タイマ/イベント・カウンタ2の割り込み処理でポート・テストをし、パリティ・ビットを出力

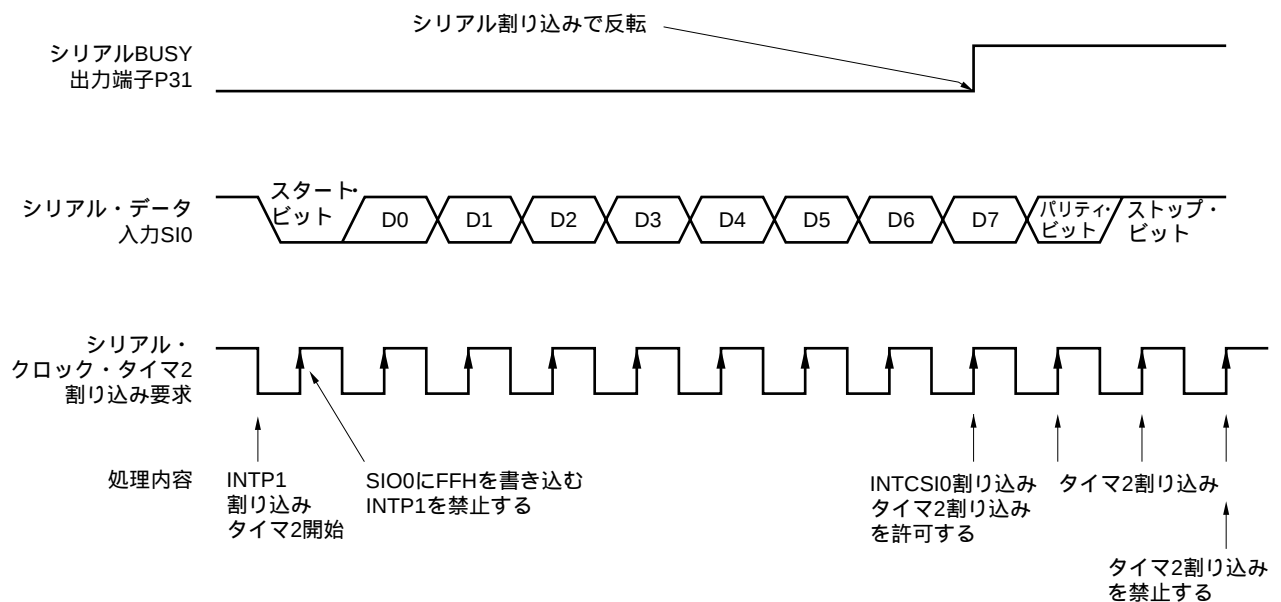
注意 受信タイミングが遅れるのを防ぐために、8ビット・タイマ/イベント・カウンタ2による割り込み要求を高優先にしてください。

ストップ・ビット→8ビット・タイマ/イベント・カウンタ2の割り込み処理でポート・テストをし、ストップ・ビットを出力

注意 受信タイミングが遅れるのを防ぐために、8ビット・タイマ/イベント・カウンタ2による割り込み要求を高優先にしてください。

パリティ・エラーまたはオーバラン・エラーが発生したときは、フラグをセットします。

図8 - 28 3線式シリアルI/Oモードの受信フォーマット



(3) パッケージの説明

<パブリック宣言シンボル>

- サブルーチン名称
 - S_SOSHIN : 送信用サブルーチン名称
 - S_JUSHIN : 受信用サブルーチン名称
- 入力パラメータ
 - SODATA : 送信データを格納します。
 - F_PARITY : 偶数 / 奇数パリティの選択状態を示します。
 - F_TUSHIN : 受信 / 送信中の状態を示します。
- 出力パラメータ
 - JUDATA : 受信データを格納します。
 - F_DATA : 受信が終了したらセットされます。
 - F_ERRP : パリティのエラーを示します。
 - F_ERRE : エンド・ビットのエラーを示します。
- 入出力パラメータ
 - F_PADATA : 送受信パリティ・ビットを格納します。

<使用するレジスタ>

バンク 0 A
 バンク 1 A
 バンク 2 A

<使用するRAM>

名 称	用 途	属 性	バイト
SODATA	送信用データ格納エリア	SADDR	1
JUDATA	受信用データ格納エリア	SADDR	1
C_WORK	状態格納カウンタ	SADDR	1
i	ループ処理用ワーク・カウンタ	SADDR	1
j	ループ処理用ワーク・カウンタ	SADDR	1

<使用するフラグ>

名 称	用 途
F_PARITY	パリティ選択フラグ
F_PADATA	パリティ・ビット格納フラグ
F_TUSHIN	通信フラグ
F_ERRP	パリティ・エラー・フラグ
F_ERRE	エンド・ビット・エラー・フラグ
F_DATA	受信終了フラグ
F_WORK	ワーク・フラグ

<ネスティング>

1 レベル 3 バイト

<使用するハードウェア>

- シリアル・インタフェース・チャンネル 0 (3 線式シリアルI/Oモード)
- 8 ビット・タイマ/イベント・カウンタ 2
- 外部割り込みエッジ検出 (INTP1端子)

<初期設定>

- サブルーチンS_SOSHIN, S_JUSHINで設定しています。
- ポート 2 : 5 ビット入力ポート, 6 ビット出力ポート設定 PM2=# × 01 × × × × × B
- ポート 3 : 0 ビット入力ポート, 1 ビット出力ポート設定 PM3=# × × × × × × 01B
- シリアル・インタフェース・チャンネル 0 の設定
3 線式シリアルI/Oモード, シリアル・クロック= 8 ビット・タイマ/イベント・カウンタ 2 選択
CSIM0=#10000110B
- 8 ビット・タイマ/イベント・カウンタ 2 の設定
ボー・レート9600 bps設定 CR20=#54
8 ビット・タイマ・レジスタ× 2 チャンネル・モード TCL1=#01110000B
8 ビット・タイマ/イベント・カウンタ 2 動作禁止 TOC1=#00000000B
TMC1=#00000000B
- INTP1の設定 INTP1立ち下がりエッジ INTM0=#00000000B
- 8 ビット・タイマ/イベント・カウンタ 2 割り込み高優先 CLR1 TMPR2
- INTP1割り込み高優先 CLR1 PPR1
- シリアル・インタフェース割り込み許可 CLR1 CSIMK0

<起動方法>

- データ送信, データ受信を開始するには, 次の順番で設定してください。

○ データ送信開始

SODATAエリアに送信データ格納

送信フラグをセット

サブルーチンS_SOSHINをコール

○ データ受信開始

通信フラグ (F_TUSHIN) をクリア (0 に設定)

BUSY信号を反転

サブルーチンS_JUSHINをコール

- 78K/0シリーズのパッケージ以外の割り込み要求を使用する場合は, 高優先割り込みを許可するために, 割り込み処理のはじめにISPフラグを 0 に設定し, 割り込み要求を許可してください。

(4) 使用例

パリティ・ビットの偶数／奇数の選択，および送信／受信の選択をキー入力により行う場合の例を示します。

```

EXTRN  SODATA
EXTRN  JUDATA, S_SOSHIN, S_JUSHIN
EXTBIT  F_PARITY, F_DATA, F_PADATA, F_TUSHIN
EXTBIT  F_ERRE, F_ERRP
;
BUSY_0 EQU P3.1
BUSY_1 EQU P3.0
PARIKEY EQU 22                                ;パリティキーのデコード値
JYUSHIN EQU 21                                ;受信キーのデコード値
TUSHIN EQU 20                                 ;送信キーのデコード値
;*****
;                      イニシャライズ
;*****
VERES  CSEG  AT 00H
      DW  RES_STA
M3      CSEG
RES_STA:
      MOV P2, #0BFH                            ;P2.5=H, P2.6=L
      MOV P3, #0FFH
      MOV PM2, #00100000B                       ;P2.5=入力ポート, P2.6=出力ポート
      MOV PM3, #00000001B                       ;P3.0=入力ポート, P3.1=出力ポート
;***8ビットタイマレジスタ設定***
      CR20=#54
      TCL1=#01110000B                           ;カウンタクロック 1.05MHz
      TOC1=#00000000B
      TMC1=#00000000B                           ;8ビットタイマレジスタ選択, タイマ2動作禁止
;***シリアルインタフェース0設定***
      CSIM0=#10000110B                           ;3線式モード, シリアルクロック選択8ビットタイマ2
      SET1  RELT
;***INTP1の設定***
      INTMO=#00000000B                           ;INTP1立ち下がりエッジ
      CLR1  TMPR2                                ;タイマ2割り込み高優先
      CLR1  PPR1                                ;INTP1割り込み高優先
      CLR1  PIF1                                ;INTP1要求フラグクリア
      CLR1  TMIF2                                ;タイマ2要求フラグクリア
      CLR1  CSIIF0                              ;シリアルインタフェース要求フラグクリア
      CLR1  CSIMK0                              ;シリアルインタフェース割り込み許可
      while(forever)
          .
          .
          .

```

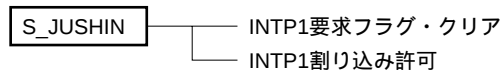
```

if_bit(F_KEYON)                                ;キ-ONフラグ 1?
switch(M_KEYON)                                ;
case PARIKEY:                                   ;押されたキ-がパリティキ-
    SET1    CY                                  ; 偶/奇パリティ判断反転
    CY ^= F_PARITY                             ;
    F_PARITY=CY                               ;
    break                                       ;
case TUSHIN:                                   ;押されたキ-が通信キ-
    SET1    F_TUSHIN                           ; 通信フラグセット(送信中)
    CLR1    F_SOEND                             ;
    break                                       ;
case JYUSHIN:                                   ;押されたキ-が受信キ-
    CLR1    F_TUSHIN                           ;通信フラグクリア(受信中)
    CY=BUSY_0                                   ; BUSY信号反転データを
    NOT1    CY                                  ; 出力する
    BUSY_0=CY                                  ;
    if_bit(CY)                                  ;
        SET1    PMK1                           ;INTP1割り込み禁止
    else                                         ;
        CLR1    F_ERRP                         ;
        CLR1    F_ERRE                         ;
        CALL    !S_JUSHIN                      ;
    endif                                       ;
    break                                       ;
ends                                           ;
endif                                           ;
.
.
.
if_bit(!F_SOEND)                               ;
    if_bit(F_TUSHIN)                           ;通信フラグセット?
        CY=BUSY_1                              ; BUSY信号ノイズチェック?
        if_bit(!CY)                             ;
            SODATA=#0                           ;
            SET1    F_SOEND                     ;
            SODATA=WORK                         ;送信用データ格納エリア←送信データ
            CALL    !S_SOSHIN                   ;
        endif                                   ;
    endif                                       ;
endif                                           ;
endif                                           ;

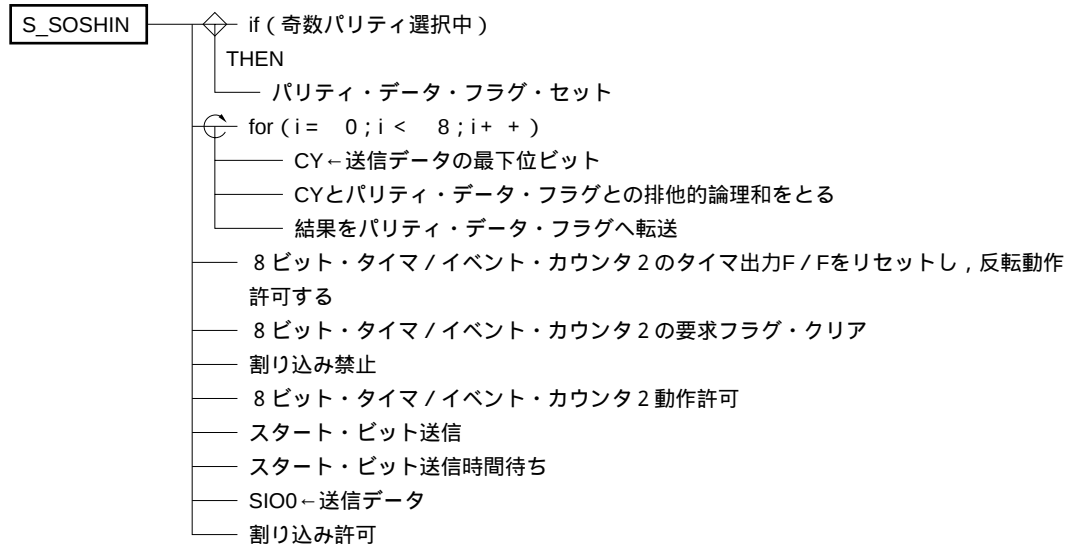
```

(5) SPDチャート

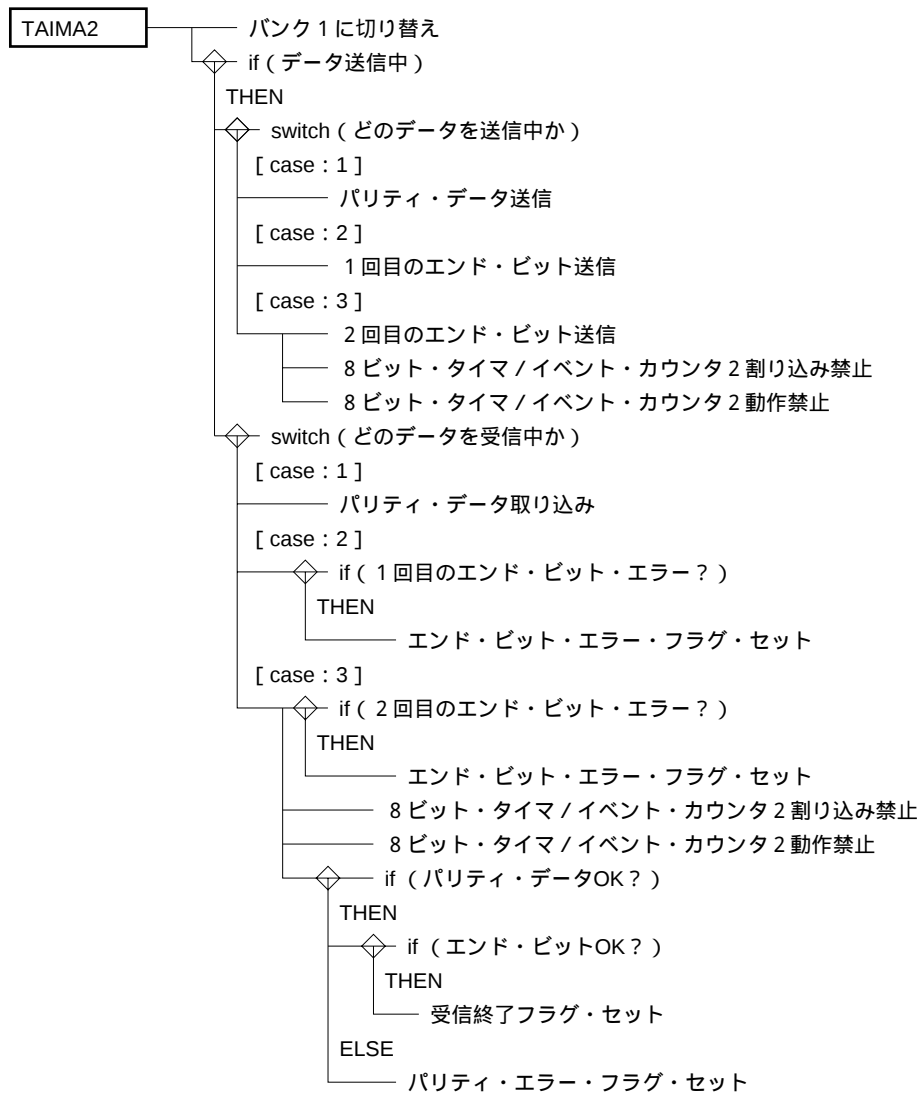
[受信サブルーチン]



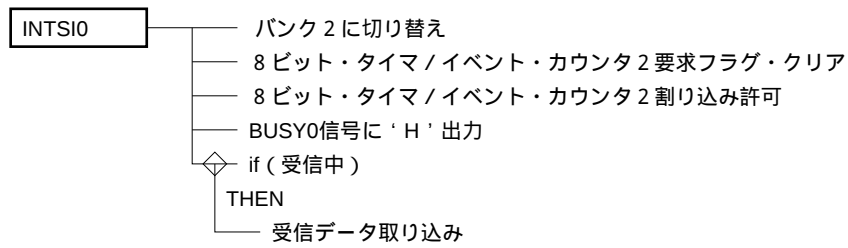
[送信サブルーチン]



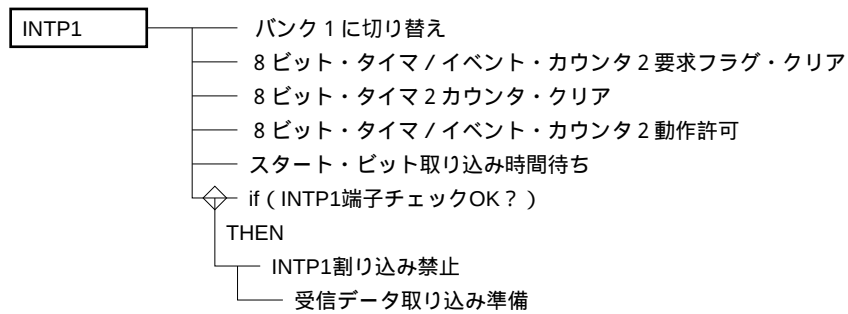
[パリティ・エンド・ビット送受信処理 (8 ビット・タイマ/イベント・カウンタ2 割り込み)]



[データ送受信完了処理]



[データ受信開始処理 (INTP1 割り込み処理)]



(6) プログラム・リスト

```

PUBLIC  F_PADATA, F_PARITY
PUBLIC  F_DATA, F_TUSHIN
PUBLIC  JUDATA, SODATA, S_JUSHIN, S_SOSHIN
PUBLIC  F_ERRP, F_ERRE
;
VEINTP1    CSEG    AT 08H
            DW      INTP1                ;INTP1のベクタ・アドレス設定
VEINTSIO    CSEG    AT 0EH
            DW      INTSIO              ;シリアル・インタフェース・チャネル0の
            ;ベクタ・アドレス設定
VETIM2      CSEG    AT 18H
            DW      TAIMA2              ;8ビット・タイマ2のベクタ・アドレス設定
;
SIO         EQU     P2.5
BUSY_0      EQU     P3.1
BUSY_1      EQU     P3.0
;
MORAM       DSEG    SADDR
SODATA:     DS      1                  ;送信データ格納エリア
C_WORK:     DS      1                  ;ワークカウンタ
JUDATA:     DS      1                  ;受信データ格納エリア
i:          DS      1                  ;ワークカウンタ
k:          DS      1                  ;ワークカウンタ
;
MOFLG       BSEG
F_PARITY     DBIT                    ;パリティ選択フラグ
F_ERRP       DBIT                    ;パリティエラーフラグ
F_ERRE       DBIT                    ;エンドビットエラーフラグ
F_DATA       DBIT                    ;受信終了フラグ
F_PADATA     DBIT                    ;パリティデータフラグ
F_WORK       DBIT                    ;ワークフラグ
F_TUSHIN     DBIT                    ;通信フラグ
;*****
;                      受信ルーチン
;*****
JUSHIN CSEG
;
S_JUSHIN:
    CLR1    PIF1                ;INTP1要求フラグクリア
    CLR1    PMK1                ;INTP1割り込み許可
    RET
;

```

```

;*****
;                               送信ルーチン
;*****
SOSHIN CSEG
S_SOSHIN:
    CLR1    F_PADATA           ;パリティクリア
    if_bit(F_PARITY)           ;奇数パリティ選択中?
        SET1    F_PADATA       ;パリティセット
    endif
    A=SODATA
    for(i=#0;i<#8;i++)         ;パリティ決定
        RORC    A,1
        CY ^= F_PADATA
        F_PADATA = CY
    next
    TOC1=#01100000B (A)
    CLR1    TMIF2              ;タイマ2要求フラグクリア
    DI
    SET1    TCE2               ;8ビットタイマ動作許可
    SET1    CMDT               ;スタートビット送信
    while_bit(!TMIF2)          ;スタートビット送信時間待ち
    endw
    CLR1    TMIF2
    SIO0=SODATA (A)            ;データ送信開始
    EI
    RET
;*****
;                               タイマ2割り込み処理
;*****
TIM2 CSEG
TAIMA2:
    SEL RB1                    ;バンク1に設定
    if_bit(F_TUSHIN)           ;通信フラグセット?
        if(C_WORK <= #4)      ;ワークカウンタの内容
            switch(C_WORK)    ;0:パリティデータ送信
            case 0:
                if_bit(F_PADATA)
                    SET1    RELT
                else
                    SET1    CMDT
                endif
                break
            case 2:             ;2:エンドビット送信
                SET1    RELT    ;"H"送信
                break
            case 4:             ;4:エンドビット送信
                SET1    RELT    ;"H"送信
                SET1    TMMK2   ;タイマ2割り込み禁止
                CLR1    TCE2    ;8ビットタイマ動作禁止
                C_WORK=#0
                break
            ends
            C_WORK++
        else
            C_WORK=#0
        endif
    endif

```

```

else
    if(C_WORK <= #6)
        switch(C_WORK)
        case 1:
            CY=SIO
            F_PADATA=CY
            break
        case 3:
            if_bit(!SIO)
                SET1    F_ERRE
            endif
            break
        case 5:
            if_bit(!SIO)
                SET1    F_ERRE
            endif
            C_WORK=#0
            SET1    TMMK2
            CLR1    TCE2
            CLR1    F_WORK
            if_bit(F_PARITY)
                SET1    F_WORK
            endif
            A=JUDATA
            for(i=#0;i<#8;i++)
                RORC    A,1
                CY ^= F_WORK
                F_WORK = CY
            next
            CLR1    F_ERRP
            CLR1    F_DATA
            F_WORK ^= F_PADATA (CY)
            if_bit(!F_WORK)
                if_bit(!F_ERRE)
                    SET1    F_DATA
                endif
            else
                SET1    F_ERRP
            endif
            break
        ends
        C_WORK++
    else
        C_WORK=#0
    endif
endif
RETI

```

```

;
;受信中?
; 7-カウツの内容
; 1:ハ`リテイ`-タ取り込み
;
;
; 3:エンド`ビ`ットチェック
; エラ`ならエンド`ビ`ットエラー`フラグ`セット
;
;
; 5:エンド`ビ`ットチェック
; エラ`ならエンド`ビ`ットエラー`フラグ`セット
;
;
; タイマ2割り込み禁止
; 8ビ`ットタイマ動作禁止
;
;
;受信デ`-タをたしこむ。
;
;
;
;
;ハ`リテイデ`-タチェック
;エンド`ビ`ットデ`-タチェック
;
;
;ハ`リテイデ`-タOKならF_DATAセット
;ハ`リテイデ`-タNGならハ`リテイエラー`フラグ`セット
;
;
;
;
;
;
;
;
;

```

```

;*****
;          INTS10割り込み処理 (受信)
;*****
S_S10    CSEG
INTS10:
    SEL RB2
    CLR1    TMIF2
    CLR1    TMMK2
    SET1    BUSY_0
    if_bit(!F_TUSHIN)
        JUDATA=S100 (A)
    endif
    C_WORK=#0
    RETI
;*****
;          INTP1割り込み処理 (受信)
;*****
S_P1     CSEG
INTP1:
    SEL RB1
    CLR1    TMIF2
    CLR1    TCE2
    SET1    TCE2
    while_bit(!TMIF2)
    endw
    CLR1    TMIF2
    if_bit(!S10)
        TOC1=#10100000B
        SET1    PMK1
        S100=#0FFH
    endif
    RETI
END

```

8.5.2 SBIモードの半二重調歩同期通信

図8 - 29にシステム構成を示します。シリアル入出力は、SB0端子で行います。ポート3の0ビット、1ビットは、BUSY信号入出力として使用しています。BUSY信号が ' L ' のとき、シリアル通信が可能です。

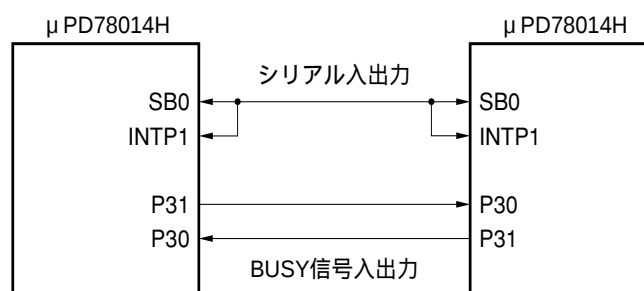
SBIモード使用についての注意点を次に示します。

ポート2の5ビット目（SB0）は、リセット・スタート時から出力モードに設定してください。ただし、SB0をポート・テストするときは、SB0を入力モードに設定してください。ポート・テスト終了後、再度出力モードに設定してください。

シリアル送受信で最後のストップ・ビットの送信・検出が終了した後、シリアル動作を禁止にしてから再度許可にしてください。

本来、SBI通信の終了は、アクノリッジ信号検出後のレディ信号を確認することで判断します。しかしここでは、アクノリッジ信号をパリティ・ビットの送受信に使用しているため、パリティ・ビット“ 1 ”を送受信するとSBI通信終了条件が成り立ちません。シリアル通信終了とみなされないと次の通信が正常動作しなくなる場合があります。

図8 - 29 システム構成図（SBIモード）



(1) SBIモードによる送信

データ送信の処理を以下に示します。

スタート・ビット → シリアル・インタフェースの出力ラッチの操作と8ビット・タイマ/イベント・カウンタ2による送信時間待ち

注意 スタート・ビットの取りこぼしによりデータ受信のタイミングが遅れるのを防ぐため、INTP1割り込み要求を高優先にしてください。

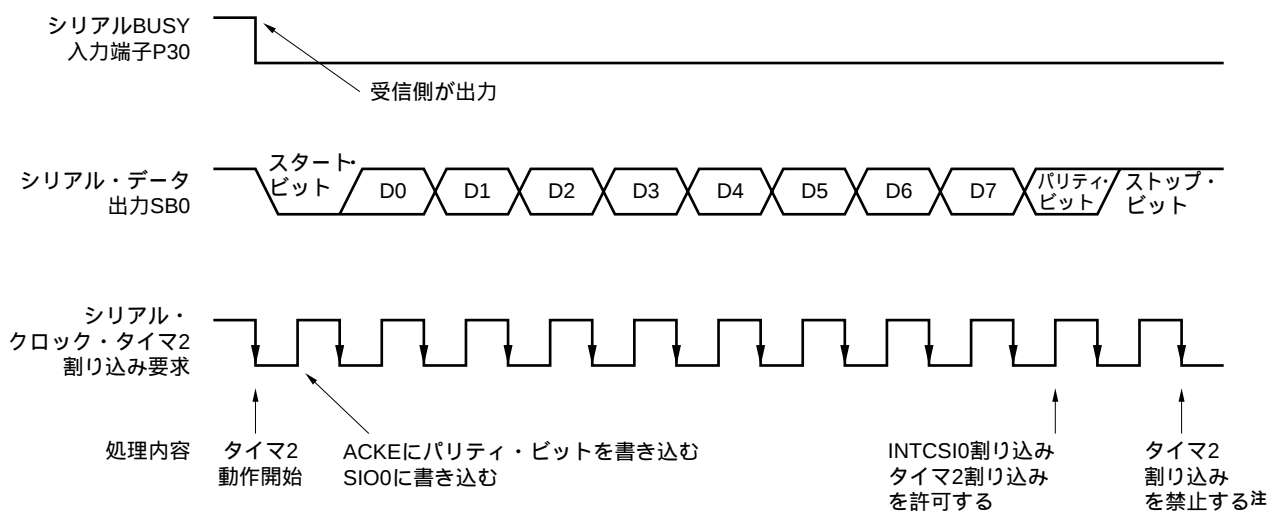
データとパリティ・ビット → シリアル・バッファとアクノリッジ信号による9ビット送信

ストップ・ビット → 8ビット・タイマ/イベント・カウンタ2の割り込み処理でシリアル・インタフェースの出力ラッチをセットし、ストップ・ビットを出力

注意1 . 送信タイミングが遅れるのを防ぐために、8ビット・タイマ/イベント・カウンタ2割り込み要求を高優先にしてください。

2 . 2ビット目のストップ・ビットの送信が終了したら、送信終了を確認するためシリアル動作を一度禁止にしてから、再度許可に設定してください。

図8 - 30 SBIモードの送信フォーマット



注 シリアル動作を一度禁止したあと、許可に設定してください。

(2) SBIモードによる受信

データ受信の処理を以下に示します。

スタート・ビット → INTP1端子の立ち下がり検出とポート・テストにより受信開始。

注意1. ポート・テストをする際に、次の順番で設定してください。

ポート2の5ビット(SI0)を入力ポートに設定する。

ポート・テスト, SI00に書き込みをする。

再度、ポート2の5ビットを出力モードに設定し直す。

2. スタート・ビットの取りこぼしによりデータ受信のタイミングが遅れるのを防ぐため、INTP1割り込み要求を高優先にしてください。

データとパリティ・ビット→シリアル・バッファとアクノリッジ検出による受信

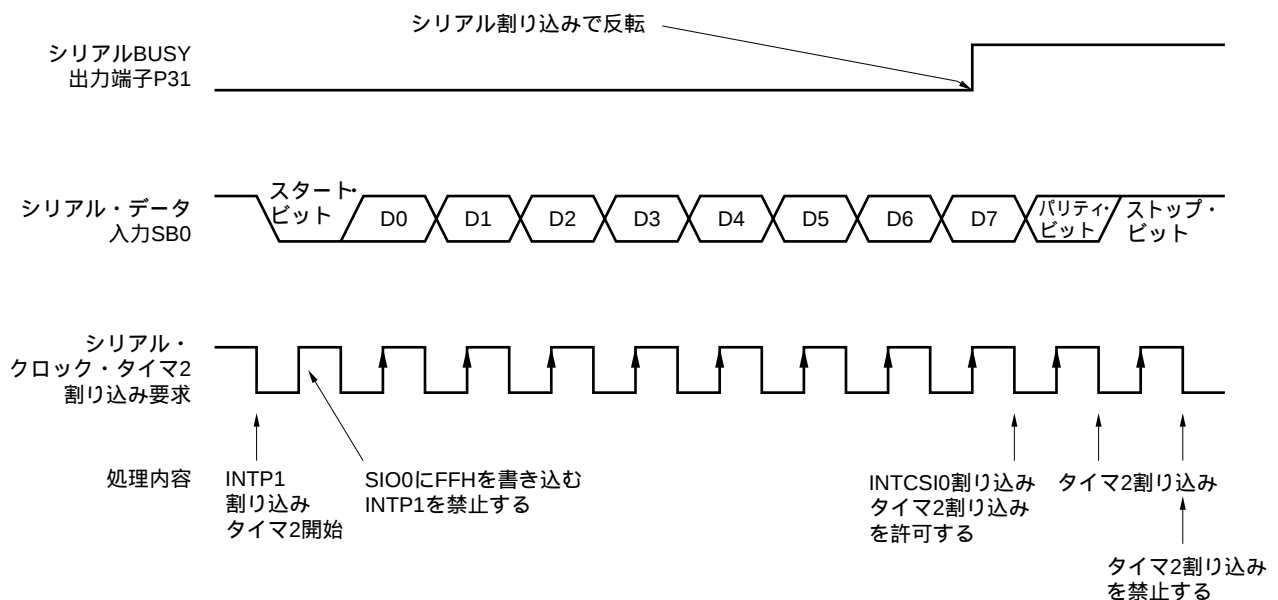
ストップ・ビット → 8ビット・タイマ/イベント・カウンタ2の割り込み処理でポート・テストをし、パリティ・ビットを出力

注意1. 送信タイミングが遅れるのを防ぐために、8ビット・タイマ/イベント・カウンタ2割り込み要求を高優先にしてください。

2. 2ビット目のストップ・ビットの送信が終了したら、送信終了を確認するためシリアル動作を一度禁止にしてから、再度許可に設定してください。

パリティ・エラーまたはオーバラン・エラーが発生したときは、フラグをセットします。

図8-31 SBIモードの受信フォーマット



(3) パッケージの説明

<パブリック宣言シンボル>

- サブルーチン名称
 - S_SOSHIN : 送信用サブルーチン名称
 - S_JUSHIN : 受信用サブルーチン名称
- 入力パラメータ
 - SODATA : 送信データを格納します。
 - F_PARITY : 偶数 / 奇数パリティの選択状態を示します。
 - F_TUSHIN : 受信 / 送信中の状態を示します。
- 出力パラメータ
 - JUDATA : 受信データを格納します。
 - F_DATA : 受信が終了したらセットされます。
 - F_ERRP : パリティのエラーを示します。
 - F_ERRE : エンド・ビットのエラーを示します。
- 入出力パラメータ
 - F_PADATA : 送受信のパリティ・ビットを格納します。

<使用するレジスタ>

- バンク0 A
 バンク1 A
 バンク2 A

<使用するRAM>

名 称	用 途	属 性	バイト
SODATA	送信用データ格納エリア	SADDR	1
JUDATA	受信用データ格納エリア	SADDR	1
C_WORK	状態格納カウンタ	SADDR	1
i	ループ処理用ワーク・カウンタ	SADDR	1
j	ループ処理用ワーク・カウンタ	SADDR	1

<使用するフラグ>

名 称	用 途
F_PARITY	パリティ選択フラグ
F_PADATA	パリティ・ビット格納フラグ
F_TUSHIN	通信フラグ
F_ERRP	パリティ・エラー・フラグ
F_ERRE	エンド・ビット・エラー・フラグ
F_DATA	受信終了フラグ
F_WORK	ワーク・フラグ

<ネスティング>

1 レベル 3 バイト

<使用するハードウェア>

- シリアル・インタフェース・チャンネル 0 (SBIモード)
- 8 ビット・タイマ/イベント・カウンタ 2
- 外部割り込みエッジ検出 (INTP1端子)

<初期設定>

- データの入出力として使用する端子 (P25) には、リセット・スタート後、1 バイト目のシリアル転送の前に次の設定をしてください。

P25の出力ラッチに 1 を設定します。

シリアル・バス・コントロール・レジスタ (SBIC) のビット 0 (RELT) に 1 を設定します。

1 を設定したP25の出力ラッチに今度は、0 を設定します。

- サブルーチンS_SOSHIN, S_JUSHINで設定しています。
- ポート 2 : 5 ビット入力ポート, 6 ビット出力ポート設定 PM2=#×01×××××B
- ポート 3 : 0 ビット入力ポート, 1 ビット出力ポート設定 PM3=#××××××01B
- シリアル・インタフェース・チャンネル 0 の設定
SBIモード, シリアル・クロック=8 ビット・タイマ 2 選択 CSIM0=#10010110B
- 8 ビット・タイマ/イベント・カウンタ 2 の設定
ボー・レート9600 bps設定 CR20=#54
8 ビット・タイマ・レジスタ×2 チャンネル・モード TCL1=#01110000B
8 ビット・タイマ/イベント・カウンタ 2 動作禁止 TOC1=#00000000B
TMC1=#00000000B
- INTP1の設定 INTP1立ち下がりエッジ INTM0=#00000000B
- 8 ビット・タイマ/イベント・カウンタ 2 割り込み高優先 CLR1 Tmpr2
- INTP1割り込み高優先 CLR1 PPR1
- シリアル・インタフェース割り込み許可 CLR1 CSIMK0

<起動方法>

- データ送信, データ受信を開始するには、次の順番で設定してください。

○ データ送信開始

SODATAエリアに送信データ格納

送信フラグをセット

サブルーチンS_SOSHINをコール

○ データ受信

通信フラグ (F_TUSHIN) をクリア (0 に設定)

BUSY信号を反転

サブルーチンS_JUSHINをコール

- 78K/0シリーズのパッケージ以外の割り込み要求を使用する場合は、高優先割り込み要求を許可するために、割り込み処理のはじめにISPフラグを 0 に設定し、割り込み要求を許可してください。

(4) 使用例

パリティ・ビットの偶数／奇数および、送信／受信をキー入力により選択する場合の例を示します。

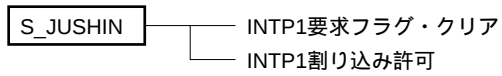
```

EXTRN  SODATA
EXTRN  JUDATA, S_SOSHIN, S_JUSHIN
EXTBIT  F_PADATA, F_PARITY, F_DATA, F_TUSHIN
EXTBIT  F_ERRP, F_ERRE
;
TUSHIN  EQU 20
JYUSHIN EQU 21
PARIKEY EQU 22
BUSY_0  EQU P3.1
BUSY_I  EQU P3.0
SB0     EQU P2.5
;*****
;               イニシャライズ
;*****
M3S      CSEG
RES_STA:
    MOV P2, #9FH          ;P2. 5=L, P2. 6=L
    MOV P3, #0FFH
    MOV PM2, #00000000B   ;P2. 5=出力モード
    MOV PM3, #00000001B   ;P3. 0=入力モード, P3. 1=出力モード
;***8ビットタイマレジスタ設定***
    CR20=#54
    TCL1=#01110000B      ;カウントクロック 1. 05MHz
    TOC1=#00000000B
    TMC1=#00000000B      ;8ビットタイマレジスタ選択, タイマ2動作禁止
;***シリアルインタフェース0設定***
    SET1  SB0
    CSIM0=#10000110B      ;SBIモード, シリアルクロック選択8ビットタイマ2
    SET1  RELT
    CLR1  SB0
;***INTP1の設定***
    CLR1  TMPR2            ;タイマ2割り込み高優先
    CLR1  PPR1            ;INTP1割り込み高優先
    INTM0=#00000000B      ;INTP1立ち下がりエッジ
    CLR1  PIF1            ;INTP1要求フラグクリア
    CLR1  TMIF2           ;タイマ2要求フラグクリア
    CLR1  CSIIF0          ;シリアルインタフェース要求フラグクリア
    CLR1  KSIF            ;割り込み要求フラグクリア
    CLR1  CSIMK0          ;シリアルインタフェース割り込み許可
    CLR1  KSMK            ;INTKS割り込み許可
;

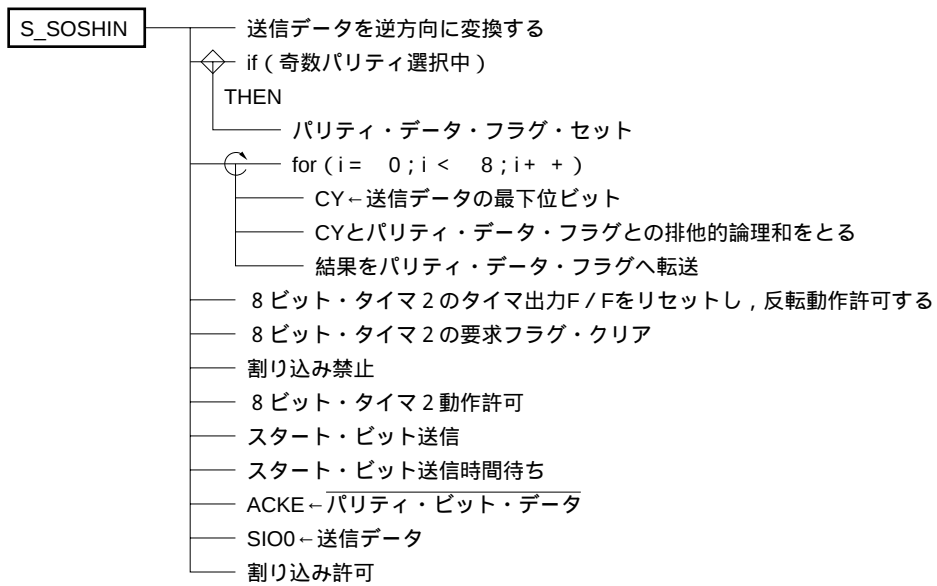
```


(5) SPDチャート

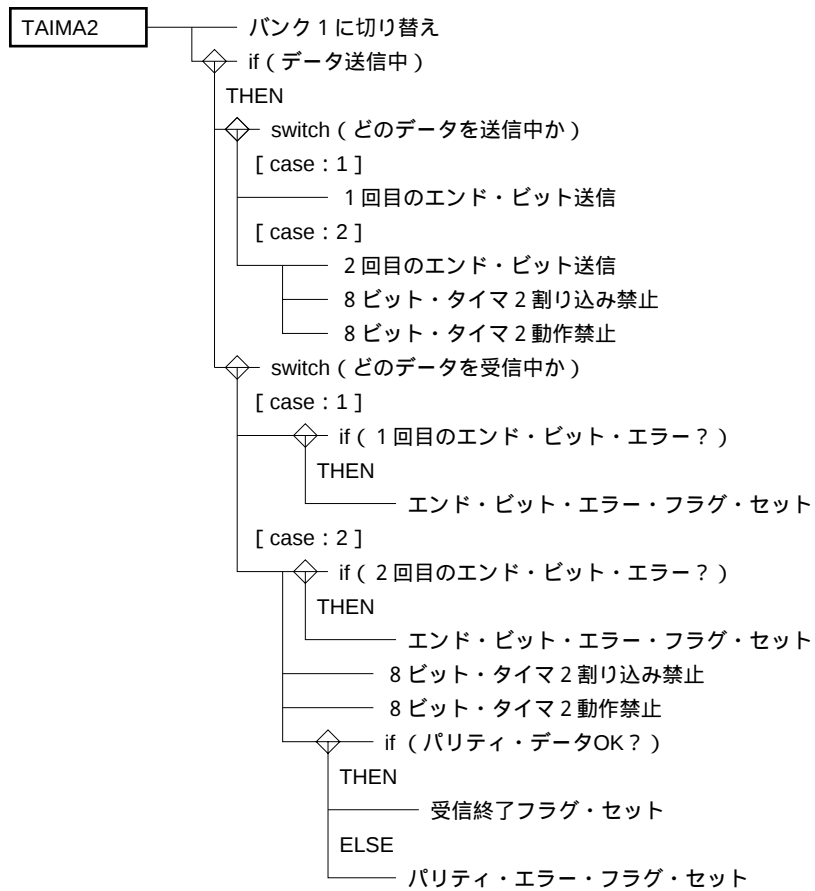
[受信サブルーチン]



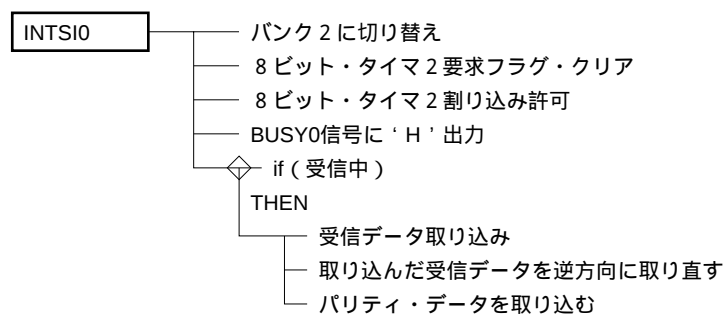
[送信サブルーチン]



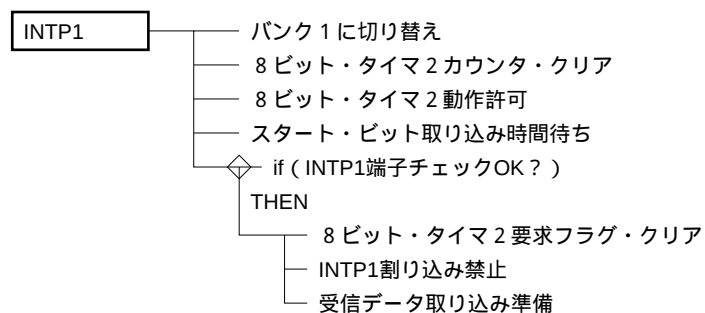
[ストップ・ビット送受信処理 (8 ビット・タイマ/イベント・カウンタ2 割り込み処理)]



[データ送受信完了処理 (INTSI0割り込み処理)]



[データ受信開始処理 (INTP1割り込み処理)]



(6) プログラム・リスト

```

PUBLIC  JUDATA
PUBLIC  SODATA, F_PARITY, S_SOSHIN
PUBLIC  F_DATA, S_JUSHIN, F_PADATA, F_TUSHIN
PUBLIC  F_ERRE, F_ERRP
;
VEINTP1    CSEG    AT 08H
           DW      INTP1
VEINTSIO    CSEG    AT 0EH
           DW      INTSIO
VETIM2      CSEG    AT 18H
           DW      TAIMA2
;
SB0         EQU     P2.5
BUSY_0      EQU     P3.1
BUSY_I      EQU     P3.0
PORT25      EQU     PM2.5
;
MOSRAM      DSEG    SADDR
SODATA:     DS      1                ;送信データ格納エリア
C_WORK:     DS      1                ;ワークカウンタ
JUDATA:     DS      1                ;受信データ格納エリア
i:          DS      1                ;ワークカウンタ
k:          DS      1                ;ワークカウンタ
;
MOSFLG      BSEG
F_ERRP      DBIT                    ;パリティエラーフラグ
F_ERRE      DBIT                    ;エンドビットエラーフラグ
F_DATA      DBIT                    ;受信終了フラグ
F_PADATA    DBIT                    ;パリティデータフラグ
F_PARITY     DBIT                    ;パリティ選択フラグ
F_WORK      DBIT                    ;フラグワークエリア
F_TUSHIN     DBIT                    ;通信フラグ
;
;*****
;               受信ルーチン
;*****
JUSHIN CSEG                          ;
S_JUSHIN:                          ;
    CLR1    PIF1                      ;要求フラグクリア
    CLR1    PMK1                      ;INTP1割り込み許可
    RET                                ;

```

```

;
;*****
;               送信ルーチン
;*****
SOSHON CSEG
S_SOSHIN:
    A=SODATA                ;送信データを逆方向にデータを変換する
    SODATA=#0
    if_bit(A.7)
        SET1    SODATA.0
    endif
    if_bit(A.6)
        SET1    SODATA.1
    endif
    if_bit(A.5)
        SET1    SODATA.2
    endif
    if_bit(A.4)
        SET1    SODATA.3
    endif
    if_bit(A.3)
        SET1    SODATA.4
    endif
    if_bit(A.2)
        SET1    SODATA.5
    endif
    if_bit(A.1)
        SET1    SODATA.6
    endif
    if_bit(A.0)
        SET1    SODATA.7
    endif
    CLR1    F_PADATA        ;パリティフラグクリア
    if_bit(F_PARITY)        ;奇数パリティ選択中?
        SET1    F_PADATA    ;パリティセット
    endif
    A=SODATA
    for(k=#0;k<#8;k++)      ;パリティ決定
        RORC    A,1
        CY ^= F_PADATA
        F_PADATA = CY
    next
    TOC1=#01100000B (A)
    CLR1    TMIF2           ;タイマ2要求フラグクリア
    DI
    SET1    TCE2            ;8ビットタイマ動作許可
    SET1    CMDT            ;スタートビット送信
    while_bit(!TMIF2)       ;スタートビット送信時間待ち
    endw
    CLR1    TMIF2
    SET1    ACKE            ;アクリッジクリア
    if_bit(F_PADATA)        ;パリティデータが1の時アクリッジをクリアする
        CLR1    ACKE
    endif
    SI00=SODATA (A)        ;データ送信開始
    EI
    RET

```

```

;
;*****
;          タイマ2割り込み処理
;*****
TIM2    CSEG
TAIMA2:
    SEL RB1
    if_bit(F_TUSHIN)
        if(C_WORK < #3)
            switch(C_WORK)
                case 0:
                    SET1    RELT
                    break
                case 2:
                    SET1    RELT
                    SET1    TMMK2
                    CLR1    TCE2
                    SET1    SB0
                    CLR1    CS1E0
                    SET1    CS1E0
                    SET1    RELT
                    CLR1    SB0
                    C_WORK=#0
                    break
            ends
            C_WORK++
        else
            C_WORK=#0
        endif

```

;バンク1に設定する
;通信中?
;ワークモードの内容
;0:エンドビット送信
;2:エンドビット送信
;8ビット・タイマ2割り込み禁止
;8ビット・タイマ2動作禁止
;ポート2.5を入力モードに設定
;シリアル動作禁止
;シリアル動作許可
;ポート2.5を出力モードに設定

```

else
  if(C_WORK < #4)
    SET1 PORT25
    switch(C_WORK)
    case 1:
      if_bit(!SB0)
        SET1 F_ERRE
      endif
      break
    case 3:
      if_bit(!SB0)
        SET1 F_ERRE
      endif
      SET1 SB0
      CLR1 CSIE0
      SET1 CSIE0
      SET1 RELT
      CLR1 SB0
      C_WORK=#0
      SET1 TMMK2
      CLR1 TCE2
      CLR1 F_WORK
      if_bit(F_PARITY)
        SET1 F_WORK
      endif
      A=JUDATA
      for(i=#0;i<#8;i++)
        RORC A,1
        CY ^= F_WORK
        F_WORK = CY
      next
      CLR1 F_ERRP
      CLR1 F_DATA
      F_WORK ^= F_PADATA (CY)
      if_bit(!F_WORK)
        if_bit(!F_ERRE)
          SET1 F_DATA
        endif
      else
        SET1 F_ERRP
      endif
      CLR1 F_WORK
      break
    ends
    CLR1 PORT25
    C_WORK++
  else
    C_WORK=#0
  endif
endif
RETI

```

```

;
;受信中?
;ポート2.5入力ポートに設定
;ワークモードの内容
;1:エンドビット"H"ならエンドビットエラーフラグセット
;
;
;
;3:エンドビット"H"ならエンドビットエラーフラグセット
;
;
;ポート2.5=H
;シリアル動作禁止
;シリアル許可
;
;ポート2.5=L
;
;8ビットタイマ2割り込み禁止
;8ビットタイマ動作禁止
;
;
;受信データをたしこむ。
;
;
;
;
;パリティデータチェック
;
;正常受信ならF_DATAフラグセット
;
;
;パリティエラーならF_ERRPフラグセット
;
;
;
;
;ポート2.5を出力ポートに設定
;
;
;
;
;

```

```

;
;*****
;          INTSIO割り込み処理（受信）
;*****
S_SIO      CSEG
INTSIO:
    SEL RB2
    CLR1    TMIF2          ;タイマ2要求フラグクリア
    CLR1    TMMK2          ;タイマ2割り込み許可
    SET1    BUSY_0
    if_bit(!F_TUSHIN)
        A=SIO0
        JUDATA=#0
        if_bit(A.7)        ;受信データを逆に取り込み直す。
            SET1    JUDATA.0
        endif
        if_bit(A.6)
            SET1    JUDATA.1
        endif
        if_bit(A.5)
            SET1    JUDATA.2
        endif
        if_bit(A.4)
            SET1    JUDATA.3
        endif
        if_bit(A.3)
            SET1    JUDATA.4
        endif
        if_bit(A.2)
            SET1    JUDATA.5
        endif
        if_bit(A.1)
            SET1    JUDATA.6
        endif
        if_bit(A.0)
            SET1    JUDATA.7
        endif
        CLR1    F_PADATA    ;パリティデータ取り込み
        CY=ACKD
        NOT1    CY
        F_PADATA=CY
    endif
    C_WORK=#0
    RETI

```

```

;*****
;                               INTP1割り込み処理（受信）
;*****
S_P1      CSEG
INTP1:
    SEL RB1
    CLR1   TMIF2      ;タイマ2要求フラグクリア
    CLR1   TCE2      ;タイマ2カウンタクリア
    SET1    TCE2      ;タイマ動作許可
    while_bit(!TMIF2)
    endw
    CLR1   TMIF2
    SET1    PORT25    ;ポート入力ポートに設定
    if_bit(!SB0)      ;INTP1のチャタリング処理
        CLR1   ACKE
        TOC1=#10100000B
        SET1    PMK1      ;INTP1割り込み禁止
        SIO0=#0FFH
    endif
    CLR1    PORT25      ;ポート2.5出力ポートに設定
    RETI
END

```

第9章 A/Dコンバータの応用

78K/0シリーズのA/Dコンバータは、8ビット分解能または10ビット分解能で8チャンネルの逐次比較型になっています。動作モードは、セレクト・モードのみですが外部トリガによる変換開始指定も可能です。また、外部トリガをしないときは、選択されたチャンネルを繰り返しA/D変換します。

A/Dコンバータは、次のレジスタにより設定します。

- ・ A/Dコンバータ・モード・レジスタ (ADM, ADM0)
- ・ A/Dコンバータ入力選択レジスタ (ADIS) : μ PD78018F, 78018FY, 78014Hサブシリーズ
- ・ アナログ入力チャンネル指定レジスタ (ADS0) : μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズ

注意 μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズに内蔵されているレジスタ・フォーマットは、この章のプログラム例で使用されているレジスタのフォーマットとは異なります。したがって、 μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズのいずれかを用いて、この章で記載されているプログラム例を使用する場合は、各レジスタの設定を各製品のレジスタの設定に置き換えて使用してください。

図9 - 1 A/Dコンバータ・モード・レジスタのフォーマット
(μ PD78018F, 78018FY, 78014Hサブシリーズ)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
ADM	CS	TRG	FR1	FR0	ADM3	ADM2	ADM1	HSC	F F 8 0 H	0 1 H	R/W

ADM3	ADM2	ADM1	アナログ入力チャネルの選択			
0	0	0	ANI0			
0	0	1	ANI1			
0	1	0	ANI2			
0	1	1	ANI3			
1	0	0	ANI4			
1	0	1	ANI5			
1	1	0	ANI6			
1	1	1	ANI7			

FR1	FR0	HSC	A/D変換時間の選択 ^{注1}				
				$f_x = 10.0 \text{ MHz}$ 時	$f_x = 8.38 \text{ MHz}$ 時	$f_x = 5.0 \text{ MHz}$ 時	$f_x = 4.19 \text{ MHz}$ 時
0	0	1	$160/f_x$	設定禁止 ^{注2}	$19.1 \mu s$	$32.0 \mu s$	$38.1 \mu s$
0	1	1	$80/f_x$	設定禁止 ^{注2}	設定禁止 ^{注2}	設定禁止 ^{注2}	$19.1 \mu s$
1	0	0	$100/f_x$	設定禁止 ^{注2}	設定禁止 ^{注2}	$20.0 \mu s$	$23.9 \mu s$
1	0	1	$200/f_x$	$20.0 \mu s$	$23.9 \mu s$	$40.0 \mu s$	$47.7 \mu s$
上記以外			設定禁止				

TRG	外部トリガの選択
0	外部トリガなし (ソフトウェア・スタート・モード)
1	外部トリガにより変換開始 (ハードウェア・スタート・モード)

CS	A/D変換動作の制御
0	動作停止
1	動作開始

注1 . A/D変換時間が $19.1 \mu s$ 以上になるように設定してください。

2 . A/D変換時間が $19.1 \mu s$ 未満となりますので、設定禁止です。

注意1 . スタンバイ機能使用時にA/Dコンバータ部の消費電力を低減させるためには、ビット7 (CS) を0にクリアし、A/D変換動作を停止させてから、HALT命令またはSTOP命令を実行してください。

2 . 停止しているA/D変換動作を再開するときは、割り込み要求フラグ (ADIF) を0にクリアしたのちにA/D変換動作を開始してください。

備考 f_x : メイン・システム・クロック発振周波数

図9 - 2 A/Dコンバータ・モード・レジスタのフォーマット
(μ PD780024A, 780024AYサブシリーズ)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
ADM0	ADCS0	TRG0	FR02	FR01	FR00	EGA01	EGA00	0	FF80H	00H	R/W

EGA01	EGA00	外部トリガ信号, エッジ指定
0	0	エッジ検出なし
0	1	立ち下がりエッジ検出
1	0	立ち上がりエッジ検出
1	1	立ち下がり, 立ち上がりの両エッジ検出

FR02	FR01	FR00	変換時間の選択 ^{注1}
0	0	0	144/f _x (17.1 μ s)
0	0	1	120/f _x (14.3 μ s)
0	1	0	96/f _x (設定禁止 ^{注2})
1	0	0	72/f _x (設定禁止 ^{注2})
1	0	1	60/f _x (設定禁止 ^{注2})
1	1	0	48/f _x (設定禁止 ^{注2})
上記以外			設定禁止

TRG0	ソフトウェア・スタート/ハードウェア・スタートの選択
0	ソフトウェア・スタート
1	ハードウェア・スタート

ADCS0	A/D変換動作の制御
0	変換動作停止
1	変換動作許可

注1 . A/D変換時間が, 14 μ s以上になるように設定してください。

2 . A/D変換時間が14 μ s未満となりますので, 設定禁止です。

★ **注意** FR00-FR02を同一データ以外に書き換える場合は, いったんA/D変換動作を停止させたのちに行ってください。

備考1 . f_x: メイン・システム・クロック発振周波数

2 . () 内は, f_x = 8.38 MHz動作時。

図9 - 3 A/Dコンバータ入力選択レジスタのフォーマット
(μ PD78018F, 78018FY, 78014Hサブシリーズ)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
ADIS	0	0	0	0	ADIS3	ADIS2	ADIS1	ADIS0	FF84H	00H	R/W

ADIS3	ADIS2	ADIS1	ADIS0	アナログ入力 チャンネル数の選択
0	0	0	0	アナログ入力 チャンネルなし(P10-P17)
0	0	0	1	1 チャンネル (ANI0, P11-P17)
0	0	1	0	2 チャンネル (ANI0, ANI1, P12-P17)
0	0	1	1	3 チャンネル (ANI0-ANI2, P13-P17)
0	1	0	0	4 チャンネル (ANI0-ANI3, P14-P17)
0	1	0	1	5 チャンネル (ANI0-ANI4, P15-P17)
0	1	1	0	6 チャンネル (ANI0-ANI5, P16, P17)
0	1	1	1	7 チャンネル (ANI0-ANI6, P17)
1	0	0	0	8 チャンネル (ANI0-ANI7)
上記以外				設定禁止

注意1 . アナログ入力のチャンネルは , 次の順序で設定してください。

ADISでアナログ入力のチャンネル数を設定します。

ADISでアナログ入力として設定したチャンネルのうち , A/D変換するチャンネルをA/Dコンバータ・モード・レジスタ (ADM) で1チャンネル選択します。

2 . ADISでアナログ入力として設定したチャンネルは , ブルアップ抵抗オプション・レジスタ (PUO) のビット1 (PUO1) の値にかかわらず , 内蔵ブルアップ抵抗は使用されません。

図9 - 4 アナログ入力チャネル指定レジスタのフォーマット
(μ PD780024A, 780024AYサブシリーズ)

略号	7	6	5	4	3	2	1	0	アドレス	リセット時	R/W
ADS0	0	0	0	0	0	ADS02	ADS01	ADS00	FF81H	00H	R/W

ADS02	ADS01	ADS00	アナログ入力チャネルの指定
0	0	0	ANI0
0	0	1	ANI1
0	1	0	ANI2
0	1	1	ANI3
1	0	0	ANI4
1	0	1	ANI5
1	1	0	ANI6
1	1	1	ANI7

9.1 レベル・メータ

A/Dコンバータに入力されたアナログ電圧を，16個のLEDで表示します。LED表示は，4 × 4 マトリクスで行います。ここでは，μPD78014Hサブシリーズを例に説明しています。

この例ではレベル・メータが対象となっていますので，LEDはデシベル表示で行います。図9 - 5 にレベル・メータの回路を，図9 - 6 にA/D変換結果と表示桁数の関係を示します。

図9 - 5 レベル・メータ回路例

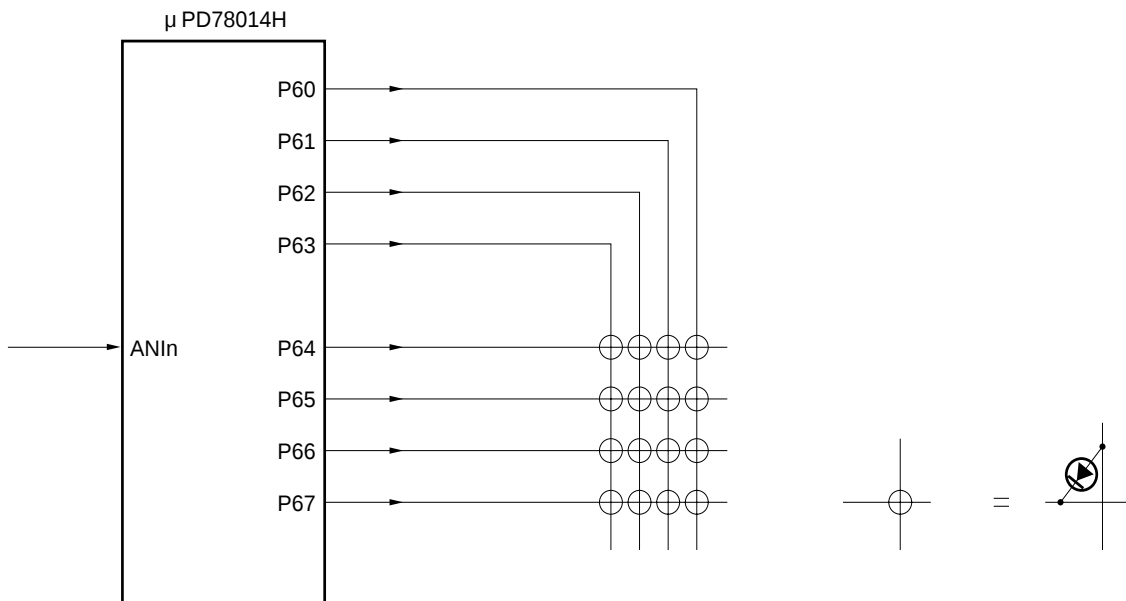
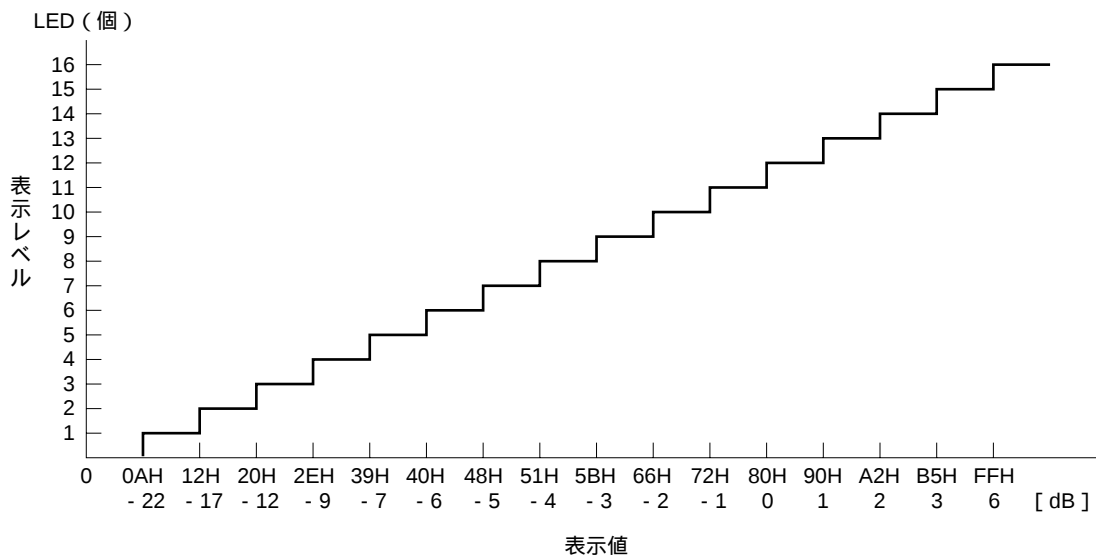


図9 - 6 A/D変換結果とLED表示



この例でのレベル・メータは， から に示す仕様で動作しています。

計測方法

20 msごとにA/D変換を行い，過去4回のデータを平均してLED表示データに使用します。

表示方法

LED表示は20 msごとに更新します。LED表示は， $4 \times 4 = 16$ のダイナミック表示で行います。ダイナミック表示用に8ビット・タイマ/イベント・カウンタ1（インターバル時間：2 ms）を使用しています。

ピーク・ホールド

一定期間（1秒）内の最大表示レベル・ホールドすることをピーク・ホールドといいます。一定期間内に表示レベルが下がった場合でも，最大表示レベルのLEDのみホールドします。したがって，ホールド・レベルのホールド期間は，20 msから1 sになります。

図9 - 7 ピーク・ホールド概念図

	← 一定期間（1秒） →																	
ホールド・レベル	6	6	6	6	7	8	9	9	9	9	9	9	4	4	4	5	6	6
表示レベル	6	5	4	5	7	8	9	8	7	6	5	5	4	3	3	5	6	2

（1）パッケージの説明

<パブリック宣言シンボル>

LEVEL : LED表示サブルーチン名称
 DSPLEV : 表示レベル格納エリア
 HLDLEV : ホールド・レベル格納エリア
 CT20MS : 20 ms計測用カウンタ
 CT1S : 1 s計測用カウンタ

<使用するレジスタ>

AX, HL, BC（サブルーチン処理）
 バンク0 : A, HL, B（割り込み処理）

<使用するRAM>

名 称	用 途	属 性	バイト
ADDAT	A/D変換値格納	SADDR	4
DSPLEV	表示レベル格納		1
HLDLEV	ホールド・レベル格納		
CT20MS	20 ms計測用カウンタ		
CT1S	1 s計測用カウンタ		
DIGCNT	表示桁カウンタ		
DSPDAT	表示データ格納		4
WORKCT	ループ処理用ワーク・カウンタ		1

<使用するフラグ>

名 称	用 途
T20MSF	20 msごとにセット
T1SF	1 sごとにセット

<ネスティング>

2 レベル 5 バイト

<使用するハードウェア>

- A/Dコンバータ
- 8 ビット・タイマ/イベント・カウンタ 1
- P6

<初期設定>

- A/Dコンバータのチャンネル選択，動作開始 ADM=#1000 × × × 1B
- 8 ビット・タイマ/イベント・カウンタ 1 のインターバル 2 ms TCL1=#10111011B
TMC1=#00000001B
CR10=130
- P6出力モード
- P6出力ラッチをロウ・レベル
- INTTM1割り込み許可

<起動方法>

このプログラムは、A/D変換処理（サブルーチン）と、LED表示処理（割り込み）の2つに別れています。

- A/D変換処理

メイン処理から20 msに1回以上LEVELをコールしてください。LEVEL処理では、20 ms経過したときのみA/D変換処理を行います。

- LED表示

4 × 4マトリクスLED表示は、8ビット・タイマ/イベント・カウンタ1の割り込み処理（インターバル：2 ms）で、ダイナミック表示を行っています。また、8ビット・タイマ/イベント・カウンタ1の割り込み処理では、A/D変換処理で使用するT20MSF（A/D変換値の取り込み）とT1SF（ホールド期間の終了）のフラグを、インターバル（2 ms）を利用してセットしています。

(2) 使用例

```
EXTRN    LEVEL, CT20MS, CT1S
```

```
MOV      CT20MS, #10
```

```
MOV      CT1S, #50
```

```
MOV      TMC2, #00100110B
```

```
CLR1     TMMK3
```

```
P6=#00H ; LED表示OFF
```

```
PM6=#00000000B
```

```
ADM=#10000001B ; ANI0端子,動作開始
```

```
TCL1=#10111011B ; 8ビット・タイマ/イベント・カウンタ1を  
2msに設定
```

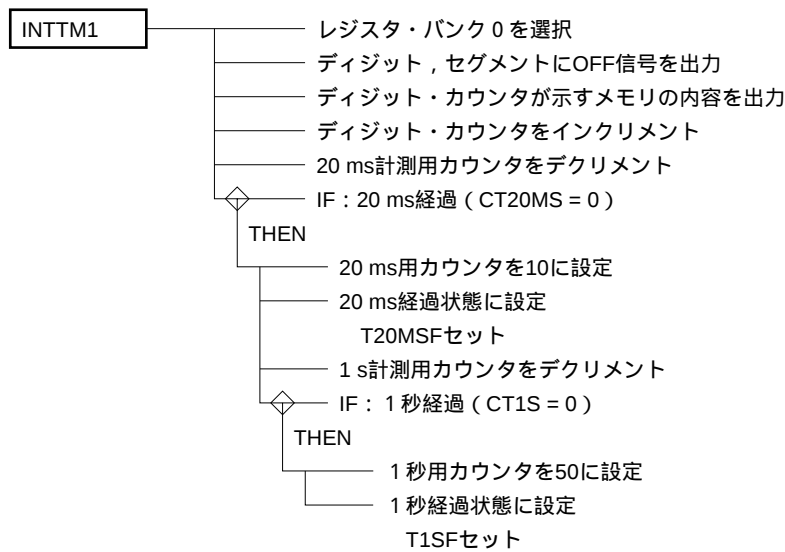
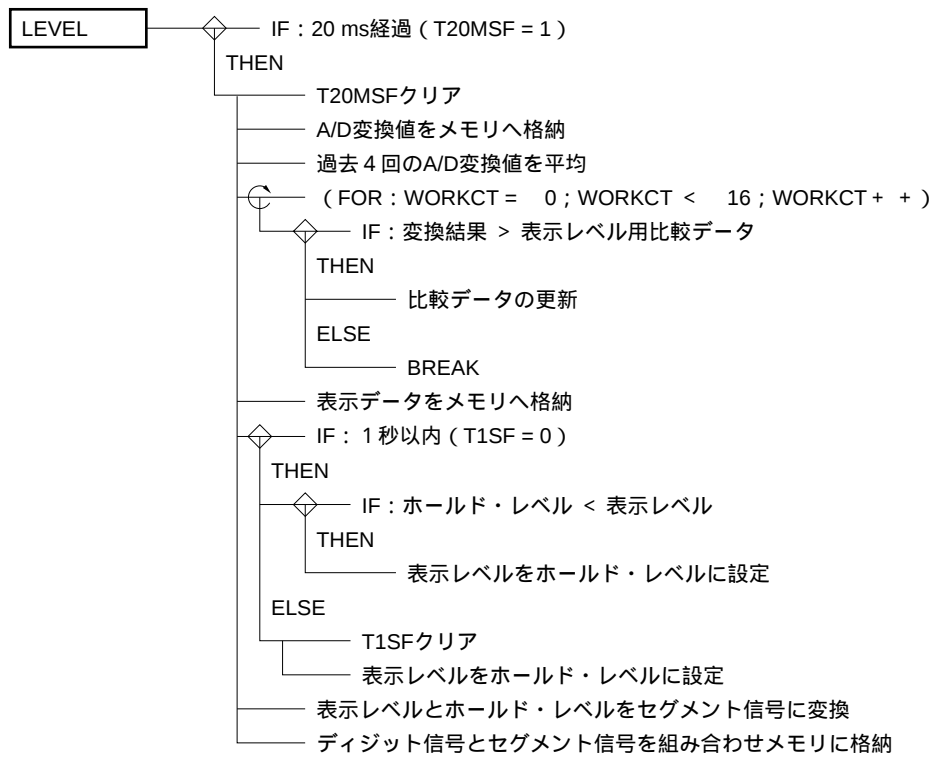
```
CR10=#130
```

```
TMC1=#00000001B
```

```
CLR1     TMMK1 ; 8ビット・タイマ/イベント・カウンタ1割り込み許可
```

```
EI
```

(3) SPDチャート



(4) プログラム・リスト

```

PUBLIC  LEVEL, HLDLEV, DSPLEV, CT20MS, CT1S

AD_DAT  DSEG      SADDR
ADDAT:  DS        4                ; A/D変換結果格納エリア
DSPLEV: DS        1                ; 表示レベル値
HLDLEV: DS        1                ; ホールド・レベル値
CT20MS: DS        1                ; 20ms計測カウンタ
CT1S:   DS        1                ; 1s計測カウンタ
DIGCNT: DS        1                ; 表示桁カウンタ
DSPDAT: DS        4                ; 表示データ
WORKCT: DS        1

AD_FLG  BSEG
T20MSF  DBIT                ; 20ms測定
T1SF    DBIT                ; 1s測定

VETM1   CSEG      AT 16H
        DW        INTTM1        ; 8ビット・タイマ/イベント・カウンタ1の
                                ; ベクタ・アドレス設定

AD_SEG  CSEG
;*****
;*      レベル・メータ・データ設定
;*****
LEVEL:
        IF_BIT(T20MSF)        ; 20msチェック
            CLR1  T20MSF
            A=ADCR              ; A/D変換値入力
            A<->ADDAT          ; A/D変換値を格納
            A<->ADDAT+1
            A<->ADDAT+2
            A<->ADDAT+3
                                ; A/D変換値4回の平均化
            AX=#0H
            HL=#ADDAT          ; データ格納アドレス
            for(WORKCT=#0;WORKCT<#4;WORKCT++)
                A+=[HL]
                HL++
                if_bit(CY)      ; 桁上げ
                X++            ; 上位桁
            endif
        next

        A<->X
        C=#4                  ; 4回の平均
        AX/=C                  ;  $AX \div C = AX(\text{商}) \cdot \cdot \cdot C(\text{余り})$ 
        if(C>=#2) (A)          ; 余り処理(2以上桁上げ)
            X++                ; 桁上げ処理
        endif

        HL=#LEVTBL
        B=#0                  ; 変換結果格納用レジスタ
        for(WORKCT=#0;WORKCT<#16;WORKCT++)
            if(X>=[HL+B]) (A)    ; データ比較
                B++
            else
                break
            endif
        next

```

```

DSPLEV=B (A)                                ; 表示データ決定

if_bit(!T1SF)                                ; 1 s (ホールド更新)
    X=HLDLEV (A)                              ; ホールド, 表示レベルの比較
    if(X<DSPLEV) (A)
        HLDLEV=DSPLEV (A)
    endif
else
    CLR1    T1SF
    HLDLEV=DSPLEV (A)
endif

HL=#DSPTBL
A=DSPLEV                                ; 表示レベル作成
A+=A
B=A
A=HLDLEV
A+=A
C=A
X=[HL+B] (A)
B++
A=[HL+B]
HL=#HLDTBL                                ; ホールド・レベル作成
A<->X
A|=[HL+C]
A<->X
C++
A|=[HL+C]
BC=AX

HL=#DSPDAT                                ; 1桁目セグメント信号の設定
A=C
A&=#0FH
A|=#00010000B                            ; デジタル信号の設定
[HL]=A
HL++
A=C                                ; 2桁目セグメント信号の設定
A>>=1
A>>=1
A>>=1
A>>=1
A&=#0FH
A|=#00100000B                            ; デジタル信号の設定
[HL]=A
HL++
A=B                                ; 3桁目セグメント信号の設定
A&=#0FH
A|=#01000000B                            ; デジタル信号の設定
[HL]=A
HL++
A=B                                ; 4桁目セグメント信号の設定
A>>=1
A>>=1
A>>=1
A>>=1
A&=#0FH
A|=#10000000B                            ; デジタル信号の設定
[HL]=A
endif

```

RET

LEVTL:

DB	0AH
DB	12H
DB	20H
DB	2EH
DB	39H
DB	40H
DB	48H
DB	51H
DB	5BH
DB	66H
DB	72H
DB	80H
DB	90H
DB	0A2H
DB	0B5H
DB	0FFH

DSPTBL:

DW	0000000000000000B
DW	0000000000000001B
DW	0000000000000011B
DW	0000000000000111B
DW	0000000000001111B
DW	0000000000011111B
DW	0000000000111111B
DW	0000000001111111B
DW	0000000011111111B
DW	0000000111111111B
DW	0000001111111111B
DW	0000011111111111B
DW	0000111111111111B
DW	0001111111111111B
DW	0011111111111111B
DW	0111111111111111B
DW	1111111111111111B

HLDTL:

DW	0000000000000000B
DW	0000000000000001B
DW	0000000000000010B
DW	0000000000000100B
DW	0000000000001000B
DW	0000000000010000B
DW	0000000000100000B
DW	0000000001000000B
DW	0000000010000000B
DW	0000000100000000B
DW	0000001000000000B
DW	0000010000000000B
DW	0000100000000000B
DW	0001000000000000B
DW	0010000000000000B
DW	0100000000000000B
DW	1000000000000000B

\$EJECT

```

;*****
;*      レベル・メータ出力
;******
TM1_SEG CSEG
INTTM1:
    SEL RBO
    P6=#00000000B                ; デジット,セグメント信号OFF
    HL=#DSPDAT
    B=DIGCNT (A)
    P6=[HL+B] (A)
    DIGCNT++
    DIGCNT&=#00000011B
    CT20MS--                    ; 20ms?
    if(CT20MS==#0)
        CT20MS=#10              ; 初期カウンタ値設定
        SET1    T20MSF
        CT1S--                  ; 1s?
        if(CT1S==#0)
            CT1S=#50            ; 初期カウンタ値設定
            SET1    T1SF
        endif
    endif
    RETI

```

9.2 温度計

この例では、温度センサにサーミスタ（ $6\text{ k}\Omega / 0^\circ\text{C}$ ）を使用して、 -20°C から $+50^\circ\text{C}$ の温度を計測します。サーミスタの温度に対する抵抗の変化は、次の式で表すことができます。

$$R = R_0 \exp \{ B (1/T - 1/T_0) \}$$

R : 任意の温度 $T [^\circ\text{K}]$ における抵抗値

T : 任意の温度 $[^\circ\text{K}]$

R_0 : 基準となる温度 $T_0 [^\circ\text{K}]$ における抵抗値

T_0 : 基準となる温度 $[^\circ\text{K}]$

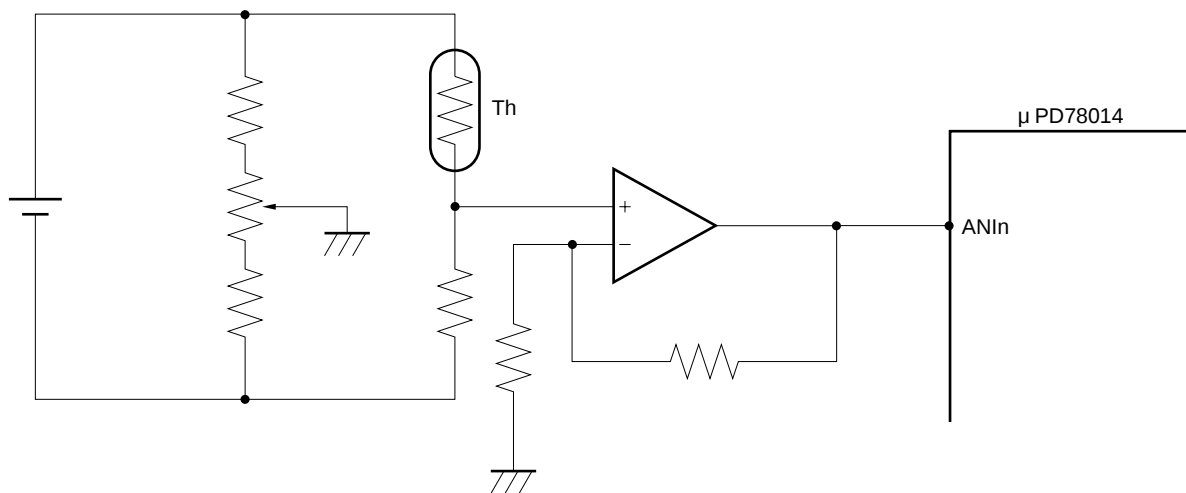
B : 基準となる温度 $T_0 [^\circ\text{K}]$ と $T_0 [^\circ\text{K}]$ において求められる定数

ただし、 B 定数は一定ではなく温度により変化します。 B 定数は、上記の式を変形して次の式で求めることができます。

$$B = \frac{1}{(1/T - 1/T_0)} \ln \frac{R}{R_0}$$

回路例を図9 - 8 に示します。この回路は、 -20°C で 0 V 、 $+50^\circ\text{C}$ で 5 V が入力されるように設定されているものとします。

図9 - 8 温度計の回路例



この回路例ではサーミスタの特性が直線的でないため、入力されたアナログ電圧を計算方式で変換せず、テーブル・データとの比較により - 20 から + 50 の温度に変換します。この変換結果は、BCD 2桁でRAM (DSPDAT) に格納します。図9 - 9にサーミスタの特性を、表9 - 1に温度とA/D変換値の関係を示します。

また、計測方法は4回の変換結果の平均値を温度に変換します。変換した結果を表示エリアに格納します。したがって、4回に1回データが更新されることになります。たとえば、250 msごとに計測処理を実行した場合、表示更新周期は1秒となります。

図9 - 9 温度と出力特性

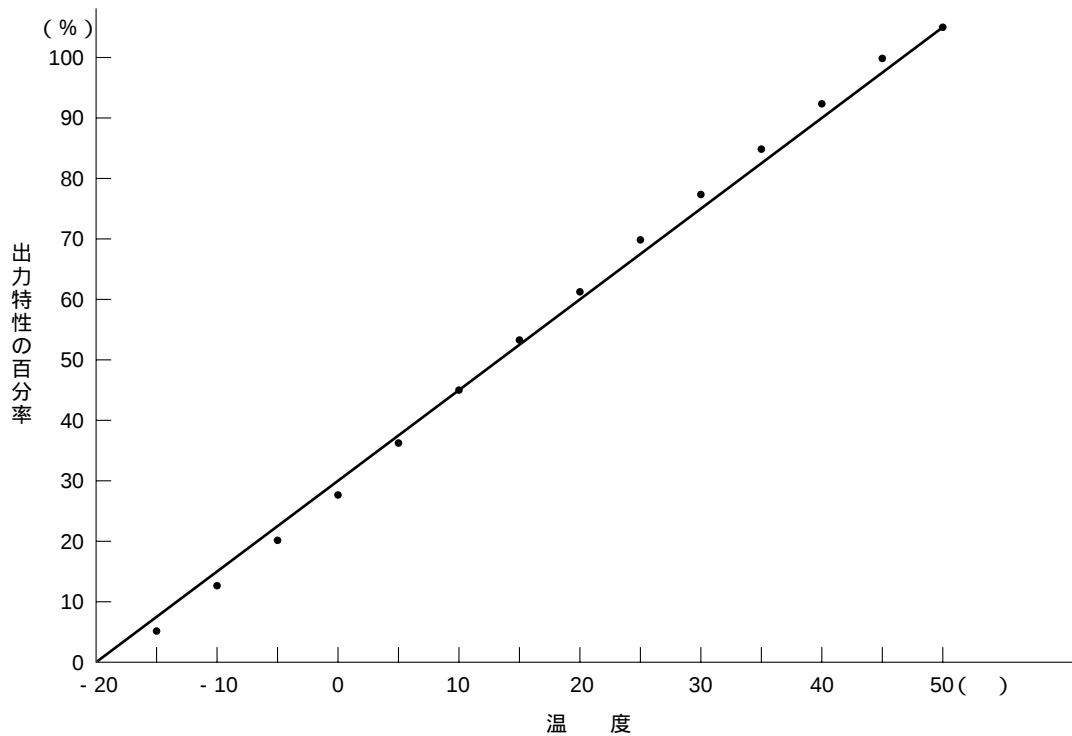


表9 - 1 A/D変換値と温度

変換値	温度 [°C]	変換値	温度 [°C]	変換値	温度 [°C]	変換値	温度 [°C]
00	- 20.0	38	- 2.5	82	15.5	CB	33.5
01	- 19.5	3C	- 1.5	86	16.5	CE	34.5
04	- 18.5	40	- 0.5	8B	17.5	D2	35.5
07	- 17.5	44	0.5	8F	18.5	D6	36.5
0A	- 16.5	48	1.5	93	19.5	D9	37.5
0C	- 15.5	4C	2.5	97	20.5	DC	38.5
0F	- 14.5	50	3.5	9B	21.5	E0	39.5
12	- 13.5	54	4.5	9F	22.5	E3	40.5
16	- 12.5	58	5.5	A3	23.5	E7	41.5
19	- 11.5	5C	6.5	A8	24.5	EA	42.5
1C	- 10.5	60	7.5	AC	25.5	ED	43.5
1F	- 9.5	64	8.5	B0	26.5	F0	44.5
23	- 8.5	69	9.5	B4	27.5	F3	45.5
26	- 7.5	6D	10.5	B7	28.5	F6	46.5
2A	- 6.5	71	11.5	BB	29.5	F9	47.5
2D	- 5.5	75	12.5	BF	30.5	FC	48.5
31	- 4.5	7A	13.5	C3	31.5	FE	49.5
35	- 3.5	7E	14.5	C7	32.5	FF	50.0

(1) パッケージの説明

<パブリック宣言シンボル>

THMETER : 温度計サブルーチン・コール名称

DSPDAT : 表示データ格納エリア

CNTPRO : 入力回数のテスト用カウンタ

MINUSF : マイナス温度表示用フラグ

T250MSF : 250 ms設定用フラグ

<使用するレジスタ>

AX, BC, HL

<使用するRAM>

名 称	用 途	属 性	バイト
ADDAT	A/D変換値格納	SADDR	4
DSPDAT	表示データ格納		2
CNTPRO	入力回数のテスト用カウンタ		1
WORKCT	ループ処理用ワーク・カウンタ		

<使用するフラグ>

名 称	用 途
T250MSF	セットされることで計測処理を実行
MINUSF	温度がマイナスのときセット

<ネスティング>

1 レベル 2 バイト

<使用するハードウェア>

A/Dコンバータ

<初期設定>

A/Dコンバータのチャンネル選択，動作開始 ADM=#1000 × × × 1B

<起動方法>

タイマ処理などで，計測周期ごとにT250MSFフラグをセットしてください。その後，計測周期以内に THMETERを 1 回以上コールしてください。

(2) 使用例

```

EXTRN  THMETER, DSPDAT, CNTPRO
EXTBIT  MINUSF, T250MSF

AD_DAT  DSEG      SADDR
CT250MS:DS      1          ; 250ms計測カウンタ
LEDD:   DS       4          ; LED表示エリア
DIGCT:  DS       1          ; LED表示桁カウンタ

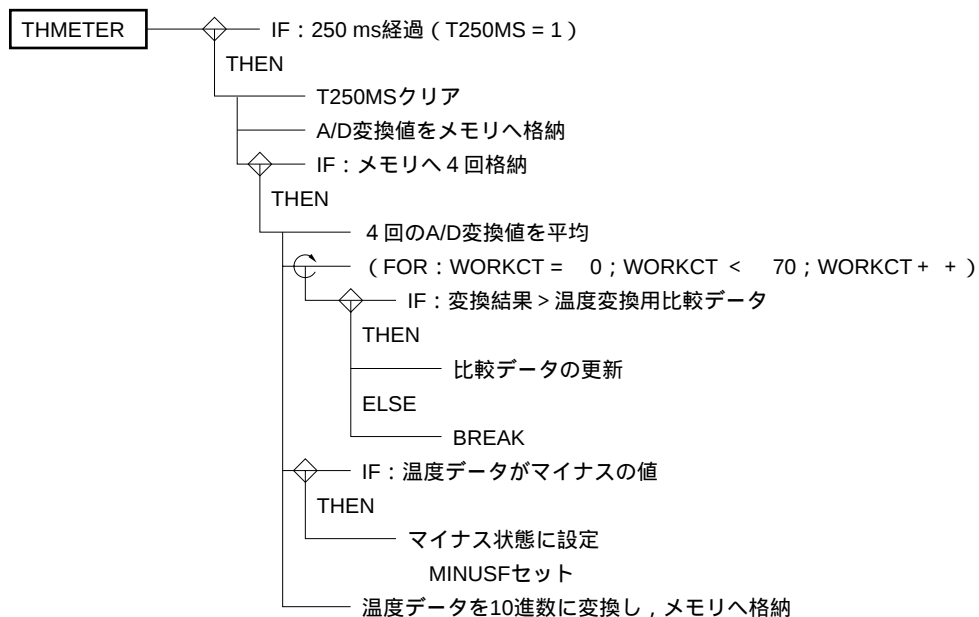
VETM3   CSEG      AT 12H
        DW        INTTM3          ; 時計用タイマのベクタ・アドレス設定

        MOV       TMC2, #00100110B ; 時計用タイマ1.95msの設定
        CLR1      TMMK3
        :
        :
        CT250MS=#128
        CNTPRO=#4
        ADM=#10000011B          ; ANI1端子,動作開始
        :
        :

;*****
;      時計用タイマ割り込み処理
;      インターバル1.95ms
;*****
INTTM3:          ; 1.95ms割り込み処理
        :
        :
        DBNZ      CT250MS, $RTNTM3
        MOV       CT250MS, #128      ; 250ms経過
        SET1      T250MSF
RTNTM3:
        :
        :
        RETI

```

(3) SPDチャート



(4) プログラム・リスト

```

PUBLIC  THMETER, DSPDAT, CNTPRO, T250MSF, MINUSF

AD_DAT  DSEG    SADDR
ADDAT:  DS      4                      ; A / D変換結果格納エリア
DSPDAT: DS      2                      ; 表示データ
CNTPRO: DS      1                      ; 入力回数のテスト
WORKCT: DS      1

AD_FLG  BSEG
T250MSF DBIT                      ; 250ms 設定
MINUSF  DBIT                      ; マイナス・データの設定

TH_SEG  CSEG
;*****
;*      温度データ設定
;*****
THMETER:
    if_bit(T250MSF)                ; 250ms
        CLR1    T250MSF
        A=ADCR
        A<->ADDAT
        A<->ADDAT+1
        A<->ADDAT+2
        A<->ADDAT+3

    CNTPRO--
    if(CNTPRO==#0)
        CNTPRO=#4
        AX=#0H
        HL=#ADDAT                ; データ格納アドレス
        for(WORKCT=#0;WORKCT<#4;WORKCT++)
            A+=[HL]
            HL++
            if_bit(CY)            ; 桁上げあり
                X++                ; 桁上げ
            endif
        next

        A<->X
        C=#4
        AX/=C                    ; AX / C = AX (商) ... C (余り)
        if(C>=#2) (A)            ; 余り処理 (2 以上桁上げ)
            X++                    ; 桁上げ処理
        endif

        A=X                      ; 温度データに変換
        B=#0
        HL=#THRTBL
        if(A=#OFFH)
            B=#70
        else
            for(WORKCT=#0;WORKCT<#70;WORKCT++)
                if(X>=[HL+B]) (A)
                    B++
                else
                    break
                endif
            next

```

```
endif
CLR1    MINUSF
A=#20           ; 温度データ 2 0
B-=A
if_bit(CY)      ; 1 0 進変換へ
    SET1    MINUSF
    A=#0
    A-=B      ; データの絶対値化
    A<->B
endif
X=#0           ; 1 0 進変換
A=B
A<->X
C=#10
AX/=C          ; 温度データ / 1 0
DSPDAT=C (A)   ; 表示データの更新
(DSPDAT+1)=X (A)
endif
endif
RET
```

THRTBL:

;

DB	1	; -19.5
DB	4	; -18.5
DB	7	; -17.5
DB	0AH	; -16.5
DB	0CH	; -15.5
DB	0FH	; -14.5
DB	12H	; -13.5
DB	16H	; -12.5
DB	19H	; -11.5
DB	1CH	; -10.5
DB	1FH	; -9.5
DB	23H	; -8.5
DB	26H	; -7.5
DB	2AH	; -6.5
DB	2DH	; -5.5
DB	31H	; -4.5
DB	35H	; -3.5
DB	38H	; -2.5
DB	3CH	; -1.5
DB	40H	; -0.5
DB	44H	; +0.5
DB	48H	; 1.5
DB	4CH	; 2.5
DB	50H	; 3.5
DB	54H	; 4.5
DB	58H	; 5.5
DB	5CH	; 6.5
DB	60H	; 7.5
DB	64H	; 8.5
DB	69H	; 9.5
DB	6DH	; 10.5
DB	71H	; 11.5
DB	75H	; 12.5
DB	7AH	; 13.5
DB	7EH	; 14.5
DB	82H	; 15.5
DB	86H	; 16.5
DB	8BH	; 17.5
DB	8FH	; 18.5
DB	93H	; 19.5
DB	97H	; 20.5
DB	9BH	; 21.5
DB	9FH	; 22.5
DB	0A3H	; 23.5
DB	0A8H	; 24.5
DB	0ACH	; 25.5
DB	0B0H	; 26.5
DB	0B4H	; 27.5
DB	0B7H	; 28.5
DB	0BBH	; 29.5
DB	0BFH	; 30.5
DB	0C3H	; 31.5
DB	0C7H	; 32.5
DB	0CBH	; 33.5
DB	0CEH	; 34.5
DB	0D2H	; 35.5
DB	0D6H	; 36.5

DB	0D9H	; 37.5
DB	0DCH	; 38.5
DB	0E0H	; 39.5
DB	0E3H	; 40.5
DB	0E7H	; 41.5
DB	0EAH	; 42.5
DB	0EDH	; 43.5
DB	0F0H	; 44.5
DB	0F3H	; 45.5
DB	0F6H	; 46.5
DB	0F9H	; 47.5
DB	0FCH	; 48.5
DB	0FEH	; 49.5

9.3 アナログ・キー入力

A/Dコンバータを使用して、16個のキーを取り込みます。キー入力を行うためには、キーが押されたときにそのキー固有の電圧がA/Dコンバータに入力されるように回路を構成します。

ここでは、 μ PD78014Hサブシリーズを例に説明します。

この例では16個のキーを取り込みますので、 V_{DD} を16分割し、そのときの電圧からキー・コードに変換します。入力電圧とキー・コード（00H-0FH）の関係を表9 - 2 に示します。また、キー入力がないときはキー・コードを10Hとします。

表9 - 2 入力電圧とキー・コード

入力電圧 V	A/D変換値	キー・コード
GND	00-07H	00H
1/16 V_{DD}	08-17H	01H
2/16 V_{DD}	18-27H	02H
3/16 V_{DD}	28-37H	03H
4/16 V_{DD}	38-47H	04H
5/16 V_{DD}	48-57H	05H
6/16 V_{DD}	58-67H	06H
7/16 V_{DD}	68-77H	07H
8/16 V_{DD}	78-87H	08H
9/16 V_{DD}	88-97H	09H
10/16 V_{DD}	98-A7H	0AH
11/16 V_{DD}	A8-B7H	0BH
12/16 V_{DD}	B8-C7H	0CH
13/16 V_{DD}	C8-D7H	0DH
14/16 V_{DD}	D8-E7H	0EH
15/16 V_{DD}	E8-F7H	0FH
V_{DD}	F8-FFH	10H

入力電圧とキー・コードの関係を実現するための回路例を図9 - 10に示します。ただし、この回路では、2個以上のキーが押された場合、番号の小さいキーを優先して取り込みます。

図9 - 10 アナログ・キー入力回路例

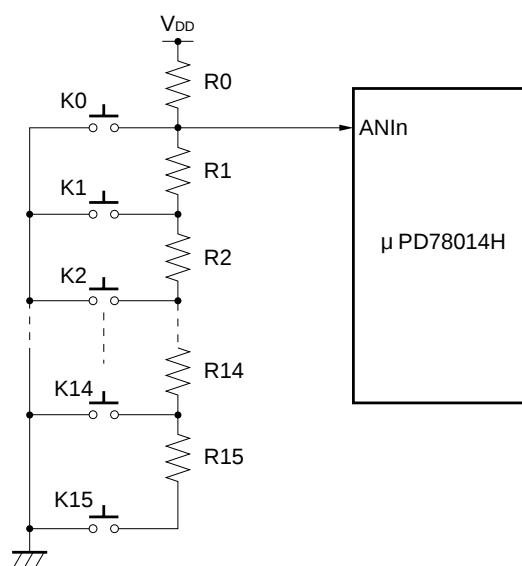


図9 - 10の回路で使用している抵抗R0からR15は、次の式で求めることができます。

$$\sum_{K=1}^n R_K = \frac{n \times R_0}{16 - n}$$

この式でR0を1 kΩにしたときのR1からR15の抵抗値を表9 - 3に示します（抵抗値は市販されている抵抗のカラー・コード表示の値などに置き換えていますので、算出結果とは誤差があります）。

表9 - 3 R1からR15の抵抗値

抵抗番号	抵抗値Ω	抵抗番号	抵抗値Ω	抵抗番号	抵抗値Ω
R1	68	R6	150	R11	560
R2	75	R7	180	R12	750
R3	82	R8	220	R13	1.3 k
R4	100	R9	270	R14	2.7 k
R5	120	R10	390	R15	8.2 k

このプログラムでは、入力したアナログ電圧を表9 - 2のキー・コードに変換し、チャタリング吸収後にRAMに格納します。チャタリングの吸収は、キー・コードが5回連続一致したときに、キーを有効にする方法を用いています。たとえば5 msごとにサンプリングしたときは、20 msから25 msのチャタリングを吸収することになります。また、キー入力に変化があったときは、キー変化フラグ（KEYCHG）をセットします。

(1) パッケージの説明

<パブリック宣言シンボル>

AKEYIN : アナログ・キー入力サブルーチン名称
 KEYDAT : キー・コード格納エリア
 PASTDT : チャタリング吸収用キー・コード格納エリア
 CHATCT : チャタリング吸収カウンタ
 KEYCHG : キー変化テスト用フラグ
 CHTENDF : チャタリング吸収終了のテスト用フラグ
 KEYOFF : キー入力がないときのキー・コード

<使用するレジスタ>

A

<使用するRAM>

名 称	用 途	属 性	バイト
PASTDAT	チャタリング吸収用キー・コード格納	SADDR	1
KEYDAT	キー・コード格納		
CHATCNT	チャタリング・カウンタ		

<使用するフラグ>

名 称	用 途
KEYCHG	キーの変化でセット
CHTENDF	チャタリング吸収終了でセット

<ネスティング>

1 レベル 2 バイト

<使用するハードウェア>

A/Dコンバータ

<初期設定>

A/Dコンバータのチャンネル選択，動作開始 ADM=#1000 × × × 1B

<起動方法>

- 一定のインターバルごとにAKEYINをコールしてください。
- キー変化フラグをテストしてから，キー・コードを取り込んでください。また，キー変化フラグはサブルーチンでクリアしていませんので，フラグのテストのあとにクリアしてください。

(2) 使用例

```

EXTRN  AKEYIN, KEYDAT, PASTDT, CHATCT
EXTRN  KEYOFF

EXTBIT  KEYCHG, CHTENDF

VETM3   CSEG    AT 12H
        DW      INTTM3                ; 時計用タイマのベクタ・アドレス設定

MAINDAT DSEG    SADDR
CT5MS:  DS      1

        TMC2=#00100110B
        CLR1    TMMK3
        CT5MS=#3

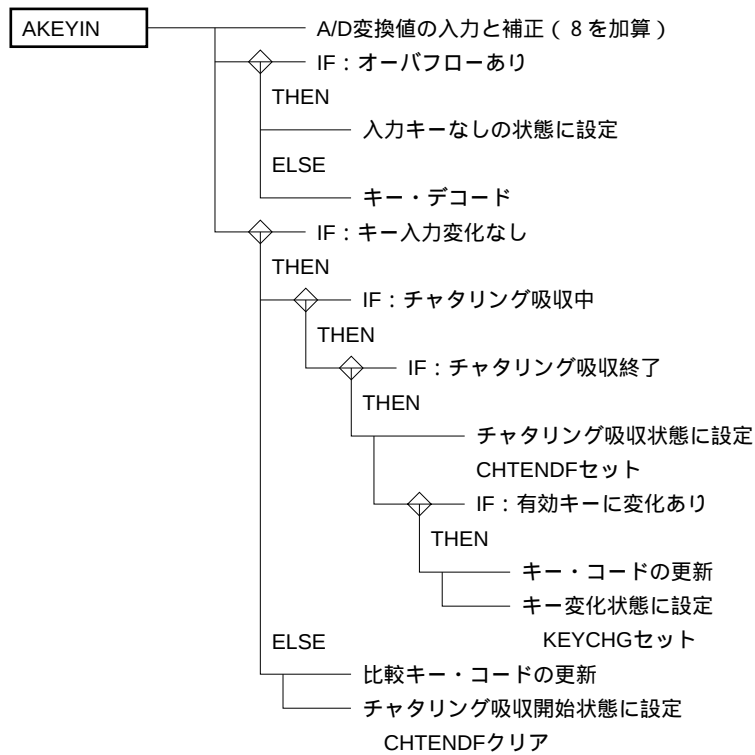
        KEYDAT=#KEYOFF                ; キー・データにOFFデータを設定
        PASTDT=#KEYOFF
        CHATCT=#CHVAL                  ; チャタリング回数を5回に設定
        CLR1    CHTENDF
        CLR1    KEYCHG
        ADM=#10000101B                ; ANI2端子, 動作開始
        EI
        :
        :
        if_bit(KEYCHG)                ; キー変化あり?
            CLR1    KEYCHG
            ; キー入力処理
        endif
        :
        :

;*****
;      時計用タイマ割り込み処理
;      インターバル1.95ms
;*****
INTTM3:                ; 1.95ms 割り込み処理
        :
        :
        DBNZ    CT5MS, $RTNTM3
        MOV     CT5MS, #3              ; 1.95ms × 3 経過
        CALL    !AKEYIN

RTNTM3:
        :
        :
        RETI

```

(3) SPDチャート



(4) プログラム・リスト

```

PUBLIC  AKEYIN, KEYDAT, PASTDT
PUBLIC  CHATCT, KEYOFF
PUBLIC  KEYCHG, CHTENDF
AK_DAT  DSEG    SADDR
KEYDAT: DS      1                ; キー・データ格納エリア
PASTDT: DS      1                ; チャタリング・キー・データ
CHATCT: DS      1                ; チャタリング・カウンタ

AK_FLG  BSEG
KEYCHG  DBIT                ; キー変化
CHTENDF DBIT                ; チャタリング吸収終了状態

KEYOFF  EQU     10H          ; OFFキー・データ
CHVAL   EQU     5            ; チャタリング吸収の回数

AK_SEG  CSEG
;*****
;*      アナログ・キー入力
;*****
AKEYIN:
    A=ADCR                ; A/D変換値入力
    A+=#8                 ; データ補正
    if_bit(CY)
        A=#KEYOFF        ; 入力キーなしの状態に設定
    else
        A>>=1            ; キー・デコード
        A>>=1
        A>>=1
        A>>=1
        A&=#0FH
    endif
    if(A==PASTDT)         ; キー変化なし
        if_bit(!CHATCT)  ; チャタリング吸収中
            CHATCT--      ; チャタリング吸収終了
            if(CHATCT==#0)
                SET1  CHTENDF ; チャタリング吸収終了状態に設定
                A=PASTDT
                if(A!=KEYDAT) ; 有効キー変化あり
                    KEYDAT=A ; キー・データ更新
                    SET1  KEYCHG ; キー変化状態に設定
                endif
            endif
        endif
    else
        PASTDT=A          ; 前キー・データ更新
        CHATCT=#CHVAL-1    ; チャタリング吸収開始
        CLR1  CHTENDF
    endif
RET

```

9.4 4チャンネル入力A/D変換

ここでは、4チャンネルをスキャンし、A/D変換する方法を説明します。A/D変換動作は、ソフトウェア・スタートにより開始します。

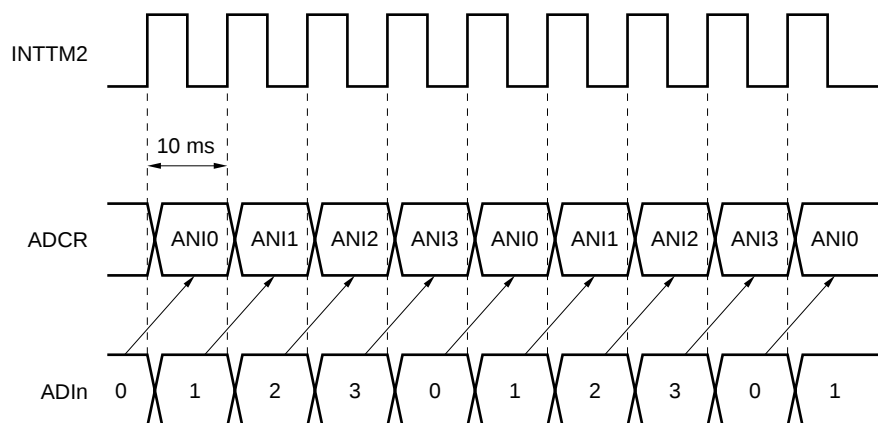
選択した4チャンネルに入力されたアナログ電圧をA/D変換し、各チャンネルの変換結果をRAMに格納します。

8ビット・タイマ/イベント・カウンタ1により、割り込み要求を発生させ、その割り込み要求内の処理で変換結果を取り込み、チャンネル変換します。ここでは、8ビット・タイマ/イベント・カウンタ1の時間を10 msで設定しているので、A/D変換の待ち時間を計測する必要はありません。

注意 割り込み処理時間を変更するときは、次の設定をしてください。

- ・タイマを $\left\{ \begin{array}{l} \text{A/D変換終了時間} + \text{割り込み移行} \\ \text{復帰時間} + \text{割り込み処理時間} \end{array} \right\}$ より長く設定する。
- ・変換終了のフラグ・テスト処理をする。

図9 - 11 4チャンネル・スキャン・モードのタイミング・チャート



(1) パッケージの説明

<パブリック宣言シンボル>

●出力パラメータ

M_CH0 : チャンネル0 変換結果を格納します。

M_CH1 : チャンネル1 変換結果を格納します。

M_CH2 : チャンネル2 変換結果を格納します。

M_CH3 : チャンネル3 変換結果を格納します。

<使用するレジスタ>

A

<使用するRAM>

名 称	用 途	属 性	バイト
M_CH0	チャンネル0 変換結果格納エリア	SADDR	1
M_CH1	チャンネル1 変換結果格納エリア	SADDR	1
M_CH2	チャンネル2 変換結果格納エリア	SADDR	1
M_CH3	チャンネル3 変換結果格納エリア	SADDR	1
M_MODE	モード格納エリア	SADDR	1

<ネスティング>

1 レベル 3 バイト

<使用するハードウェア>

- A/Dコンバータ
- 8 ビット・タイマ/イベント・カウンタ1
- ポート1 (P10-P13)

<初期設定>

- A/Dコンバータのチャンネル選択, 動作開始 ADM=#1000 × × × B
- A/Dコンバータのチャンネル数選択 ADIS=#00000100B
- 8 ビット・タイマ/イベント・カウンタ1 のインターバル10 ms TCL1=#00001110B
TMC1=#00000001B
CR10=#81
- TMMK1割り込み許可

(2) 使用例

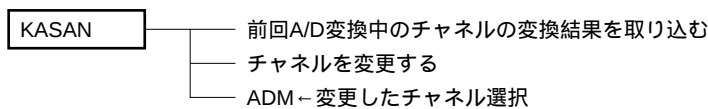
```

EXTRN    M_CH0, M_CH1, M_CH2, M_CH3, M_MODE
;*****
;          イニシャライズ
;*****
M4        CSEG                                ;
RES_STA:
    SEL RBO                                ;
    DI                                     ;
    .
    .
    ADM=#10000001B                        ;A/D動作開始, 外部トリガなし, チャンネル0選択
    ADIS=#00000100B                      ;アナログ入力4チャンネル選択
    CR10=#81                             ;コンバージョンスタ81設定
    TCL1=#00001110B                     ;カウント/クロック 8.2kHz
    TMC1=#00000001B                     ;8ビット/タイマ/レジスタ1動作許可
    CLR1    TMIF1                        ;タイマ1割り込み要求フラグクリア
    CLR1    TMMK1                       ;タイマ1割り込み許可
    EI                                     ;
    M_MODE=#0                            ;モードエリアに初期値(0チャンネル)設定
    .
    .
    while(forever)                        ;
    .
    .
    A=M_CH0                             ;A←チャンネル0のデータ
    .
    .
    A=M_CH1                             ;A←チャンネル1のデータ
    .
    .
    A=M_CH2                             ;A←チャンネル2のデータ
    .
    .
    A=M_CH3                             ;A←チャンネル3のデータ
    .
    .

```

(3) SPDチャート

[A/D変換処理]



(4) プログラム・リスト

```

;
;*****
;                               A/D変換
;*****
;
$PC(014)                               ;
;
PUBLIC  M_CH0,M_CH1,M_CH2,M_CH3,M_MODE ;
;
VEINTM1 CSEG    AT 16H
        DW  KASAN
;*****
;                               R A M定義
;*****
                DSEG    SADDR
M_CH0:        DS      1                ;チャネル0加算用RAMエリア
M_CH1:        DS      1                ;チャネル1加算用RAMエリア
M_CH2:        DS      1                ;チャネル2加算用RAMエリア
M_CH3:        DS      1                ;チャネル3加算用RAMエリア
M_MODE:       DS      1                ;モード格納エリア
;
                CSEG
KASAN:
        SEL  RB2                        ;バンク2に切り換える
        switch(M_MODE)                 ;現在選択中のチャネル?
        case 0:                         ;チャネル0:
                M_CH0=ADCR (A)          ;変換結果をRAMに転送する
                M_MODE++                ;
                ADM=#10000011B          ;チャネル選択を1に変更する
                break                  ;
        case 1:                         ;チャネル1:
                M_CH1=ADCR (A)          ;変換結果をRAMに転送する
                M_MODE++                ;
                ADM=#10000101B          ;チャネル選択を2に変更する
                break                  ;
        case 2:                         ;チャネル2:
                M_CH2=ADCR (A)          ;変換結果をRAMに転送する
                M_MODE++                ;
                ADM=#10000111B          ;チャネル選択を3に変更する
                break                  ;
        case 3:                         ;チャネル3:
                M_CH3=ADCR (A)          ;変換結果をRAMに転送する
                M_MODE=#0               ;
                ADM=#10000001B          ;チャネル選択を0に変更する
                break                  ;
        ends
        RETI
        END

```

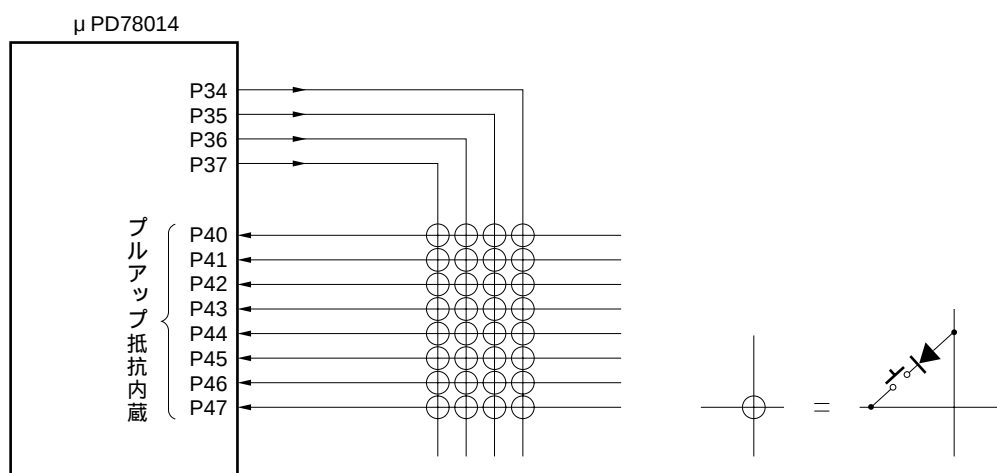
第10章 キー入力の実用

4 × 8 キー・マトリクスキーの入力を行うプログラム例を示します。キー入力の使用は、連続押し、多重押しを可能とします。この回路の構成は、キー・スキャン信号にポート3の上位4ビット（P34-P37）を使用し、キー・リターン信号にポート4を使用しています。ポート4のキー・リターン用プルアップ抵抗は、ソフトウェアの設定による内蔵プルアップ抵抗を使用しています（図10 - 1参照）。

また、78K/0シリーズのポート4は8本並列に立ち下がりエッジ検出ができる機能を持っています。したがって、ポート4をキー・リターン信号用に使用すれば、立ち下がりエッジ検出、つまりキー入力により、スタンバイ状態を解除できます。

ここでは、μPD78014Hサブシリーズを例に説明します。

図10 - 1 キー・マトリクス回路



入力キーを1キー・1ビット対応でRAMに格納します。RAMデータは、押されているキーに対応するビットをセットし、離されているキーをクリアしています。したがって、RAMデータを先頭から1ビットずつテストしていくことで、キーの状態を知ることができます。チャタリングの吸収は、キー・コードが4回連続一致したときに、キーを有効にする方法を用いています。たとえば5 msごとにサンプリングしたときは、15 msから20 msのチャタリングを吸収することになります。また、キー入力に変化があったときは、キー変化フラグ（KEYCHG）をセットします。

(1) パッケージの説明

<パブリック宣言シンボル>

KEYIN : キー入力サブルーチン名称

KEYDATA : キー・データ格納エリア

CHATCT : チャタリング・カウンタ

KEYCHG : キー変化テスト用フラグ

<使用するレジスタ>

AX, DE, HL

<使用するRAM>

名 称	用 途	属 性	バイト
KEYDATA	有効なキー・データ格納	SADDR	4
WORK	チャタリング中のキー・データ格納		
CHATCT	チャタリング・カウンタ		1
WORKCT	ループ処理用ワーク・カウンタ		

<使用するフラグ>

名 称	用 途
CHGFG	キー入力変化でセット
KEYCHG	有効キーの変化でセット
CHTEND	チャタリング吸収終了確認

<ネスティング>

1 レベル 2 バイト

<使用するハードウェア>

- P4
- P3 (P34-P37)

<初期設定>

- P4プルアップ抵抗内蔵 PU04=1
- P3上位 4 ビット出力モード PM3=#0000 × × × × B

<起動方法>

- 一定のインターバルごとにKEYINをコールしてください。
- キー変化フラグをテストしてから、キー・データを取り込んでください。また、キー変化フラグはサブルーチンでクリアしていませんので、フラグのテストのあとにクリアしてください。

(2) 使用例

```

EXTRN    AKEYIN, KEYDAT, PASTDT, CHATCT
EXTRN    KEYOFF

EXTBIT    KEYCHG, CHTENDF

VETM3    CSEG    AT 12H
          DW      INTTM3                ; 時計用タイマのベクタ・アドレス設定

MAINDAT   DSEG    SADDR
CT5MS:    DS      1

          TMC2=#00100110B
          CLR1     TMMK3
          CT5MS=#3

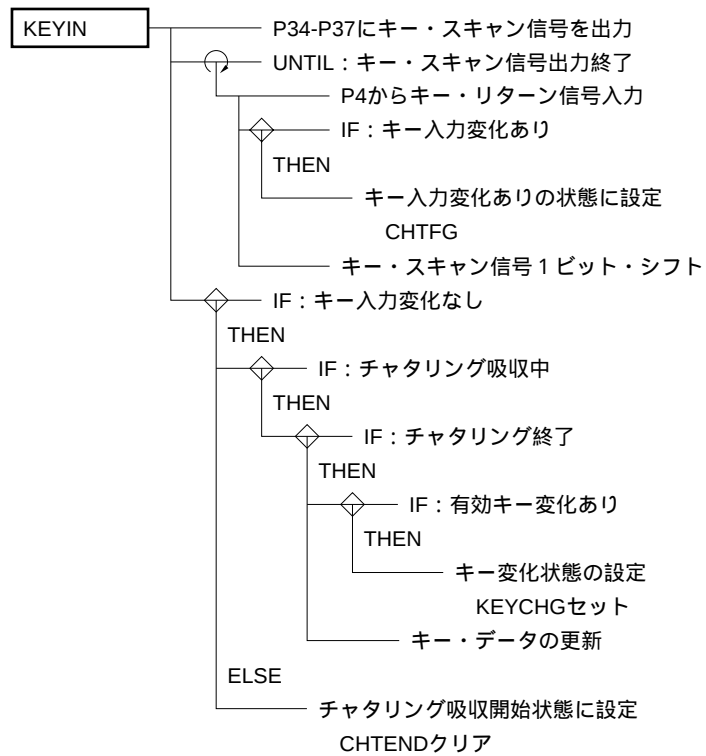
          KEYDAT=#KEYOFF                ; キー・データにOFFデータを設定
          PASTDT=#KEYOFF
          CHATCT=#CHVAL                  ; チャタリング回数を5回に設定
          CLR1     CHTENDF
          CLR1     KEYCHG
          ADM=#10000101B                ; ANI2端子, 動作開始
          EI
          :
          :
          if_bit(KEYCHG)                ; キー変化あり?
              CLR1     KEYCHG
              ; キー入力処理
          endif
          :
          :

;*****
;      時計用タイマ割り込み処理
;      インターバル1.95ms
;*****
INTTM3:                ; 1.95ms 割り込み処理
          :
          :
          DBNZ     CT5MS, $RTNTM3
          MOV      CT5MS, #3            ; 1.95ms × 3 経過
          CALL     !ANKEYIN

RTNTM3:
          :
          :
          RETI

```

(3) SPDチャート



(4) プログラム・リスト

```

PUBLIC KEYDATA, KEYCHG, KEYIN, CHATCT

KEY_DAT DSEG      SADDR
KEYDATA: DS       4                ; キー・データ格納エリア
WORK:   DS       4                ; チャタリング・キー・データ
CHATCT: DS       1                ; チャタリング・カウンタ
WORKCT: DS       1

KEY_FLG BSEG
CHGFG   DBIT                ; キー変更状態
KEYCHG  DBIT                ; キー変化あり
CHTEND  DBIT                ; チャタリング吸収終了状態

KEY_SEG CSEG
;*****
;*      マトリクス・キー入力
;*****
KEYIN:
    CLR1    CHGFG
    P3&=#00001111B
    P3|=#00010000B
    HL=#WORK                ; キー・ワーク・エリアのアドレス設定
    repeat
        A=P4
        A^=#11111111B        ; データ反転
        if(A!=[HL])          ; キー変化あり?
            SET1    CHGFG
            [HL]=A
        endif
        HL++
        A=P3                ; キー・スキャン1ビット・シフト
        A&=#11110000B
        X=A
        A=P3
        A+=X
        P3=A
    until_bit(CY)

    if_bit(!CHGFG)            ; キー変化
        if_bit(!CHTEND)      ; チャタリング吸収中
            CHATCT--          ; チャタリング終了
            if(CHATCT==#0)
                SET1    CHTEND
                DE=#WORK
                HL=#KEYDATA
                for(WORKCT=#0;WORKCT<#4;WORKCT++)
                    if([DE]!=[HL]) (A) ; キー変化あり
                        SET1    KEYCHG
                    endif
                    A<->[HL]        ; WORKをKEYDATAに転送
                    HL++
                    DE++
                next
            endif
        endif
    else
        CHATCT=#3
        CLR1    CHTEND
    endif
RET

```

付録A SPDチャートの説明

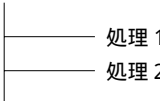
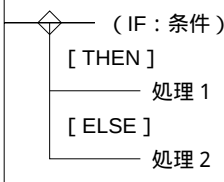
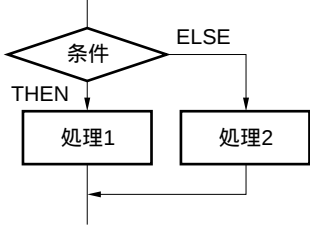
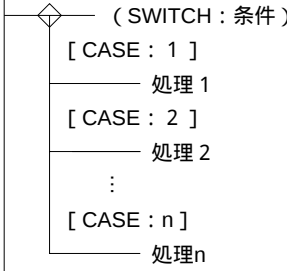
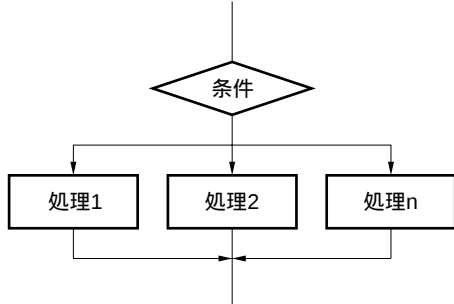
SPDとは、Structured Programming Diagramsの頭文字をとったもので、文字どおり訳せば「構造化プログラム図」と言えます。

構造化というのは、プログラムの論理処理の構造化のことで、論理の基本構造を用いて論理の設計、組み立てを行うことです。

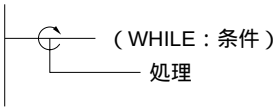
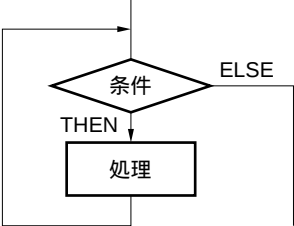
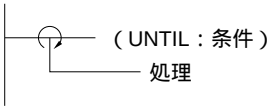
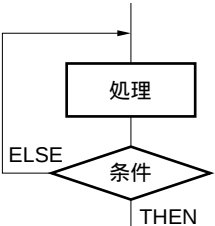
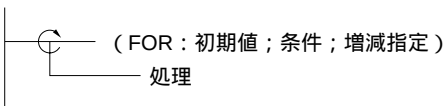
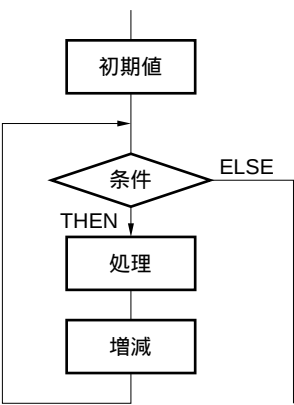
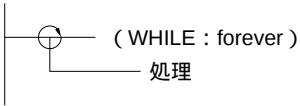
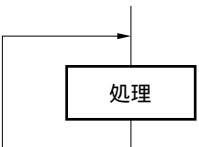
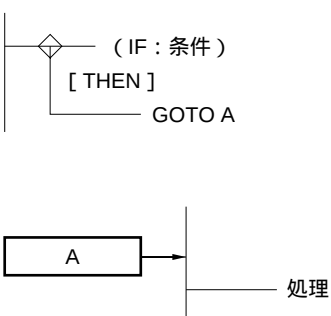
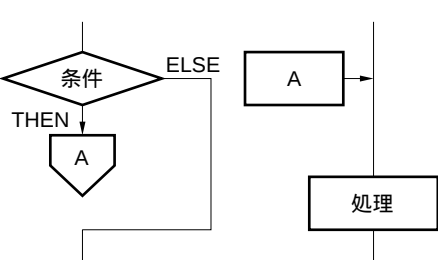
すべてのプログラムは、論理の基本構造（順次、選択、繰り返し）の組み合わせのみで作成でき（これを構造化定理といいます）、構造化することでプログラムの流れが明確になり、信頼性が向上します。プログラムの構造化を表現するには、いろいろな方法がありますが、当社では、SPDという図式化技法を用いています。

以下に、SPD技法で用いるSPD記号の説明、およびフロー・チャート記号との対比を示します。

表A - 1 SPD記号とフロー・チャートの対比 (1/2)

処理名称	SPD記号	フロー・チャート記号
順次処理		
条件分岐 (IF)		
条件分岐 (SWITCH)		

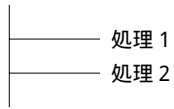
表A - 1 SPD記号とフロー・チャートの対比 (2/2)

処理名称	SPD記号	フロー・チャート記号
条件ループ (WHILE)	 (WHILE : 条件) 処理	
条件ループ (UNTIL)	 (UNTIL : 条件) 処理	
条件ループ (FOR)	 (FOR : 初期値 ; 条件 ; 増減指定) 処理	
無限ループ	 (WHILE : forever) 処理	
結合子	 (IF : 条件) [THEN] GOTO A A 処理	

1. 順次処理

順次処理は、処理を上から下へ出現順に実行します。

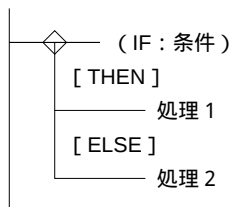
●SPDチャート



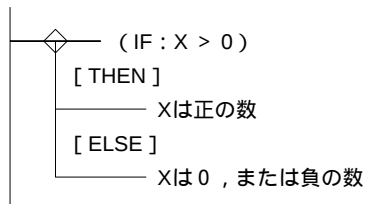
2. 条件分岐：2分岐（IF）

IFに示した条件の真偽（THEN/ELSE）により処理内容を選択します。

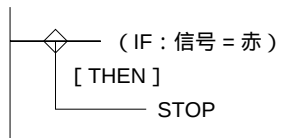
●SPDチャート



例 1 . Xの正負判別



2 . 信号が赤ならSTOPする

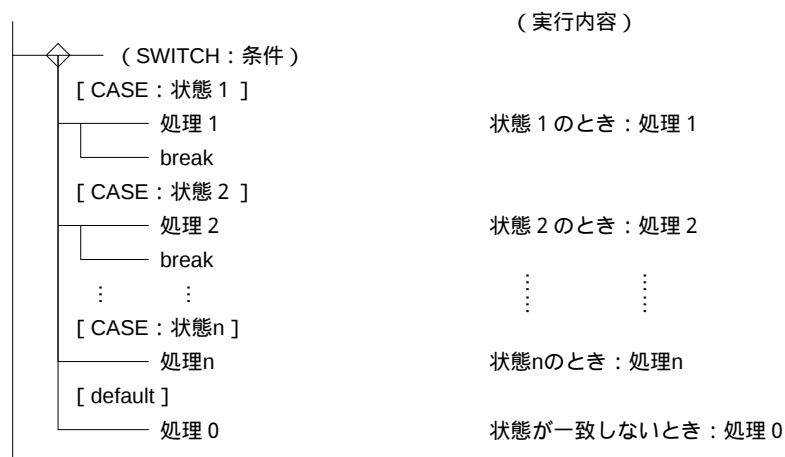


3. 条件分岐：多分岐（SWITCH）

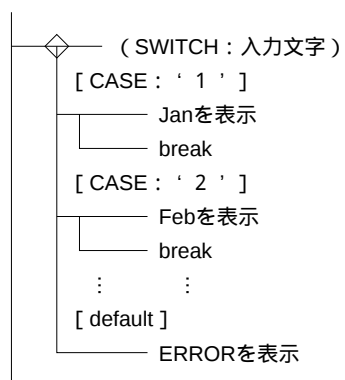
SWITCHに示した条件を、CASEで示された状態と比較し、処理を選択します。SWITCH文の処理は、一致した状態のみの処理を実行する場合と、一致した状態から下へ処理を続ける場合の二通りがあります（処理が下へ続かない場合は、‘break’を記述）。また、一致した状態がない場合は、‘default’の処理を実行します（‘default’の記述は任意）。

（1）一致した状態のみの場合

●SPDチャート

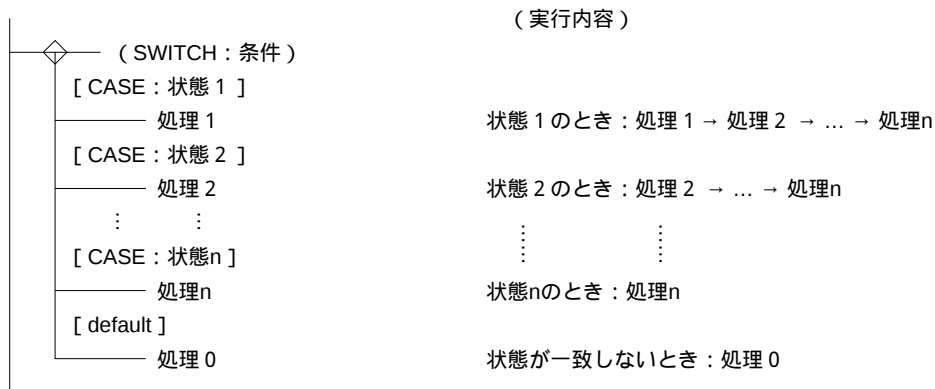


例．入力文字により，月名を表示する

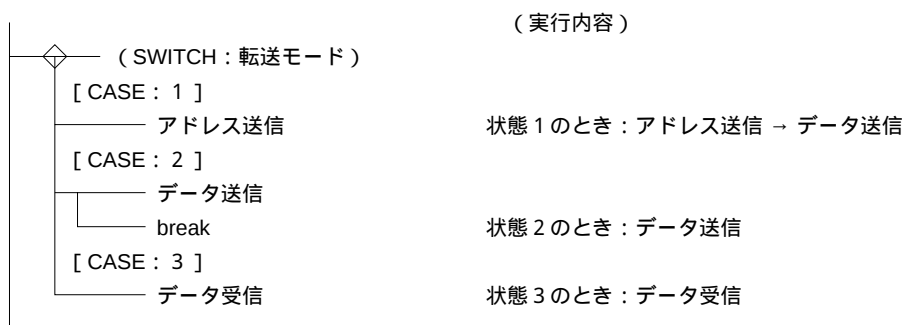


(2) 一致した状態から処理が続く場合

●SPDチャート



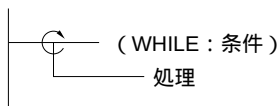
例．シリアル・インタフェースの送受信



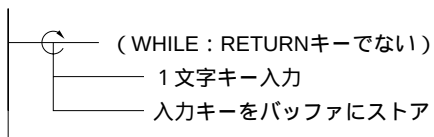
4．条件ループ (WHILE)

WHILEに示した条件を判定し、条件が成立している間、処理を繰り返し実行します（初めから条件が不成立の場合は、処理を実行しません）。

●SPDチャート



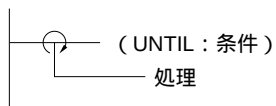
例．RETURNキー入力があるまで、キーをバッファリングする



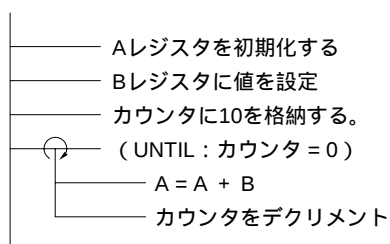
5．条件ループ（UNTIL）

処理を実行したあとにUNTILに示した条件を判定し、条件が成立するまで処理を繰り返し実行します（初めから条件が不成立の場合でも、処理を一度実行します）。

●SPDチャート



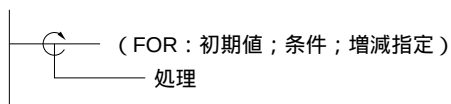
例．Bレジスタの値を10倍してAレジスタに格納する



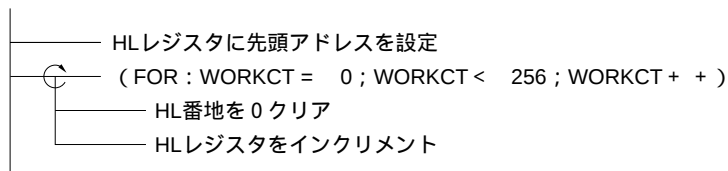
6．条件ループ（FOR）

FORに示されたパラメータの条件が成立している間、処理を繰り返し実行します。

●SPDチャート



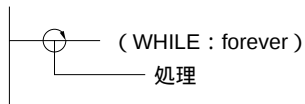
例．HL番地から256バイトを0クリアする



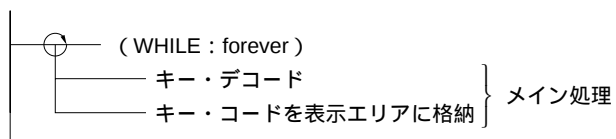
7．無限ループ

WHILEの条件として ' forever ' を設定すると、処理を無限に繰り返し実行します。

●SPDチャート



例．メイン処理を繰り返し実行する

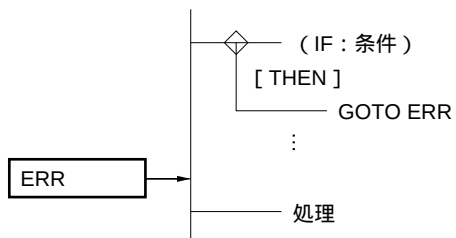


8．結合子 (GOTO)

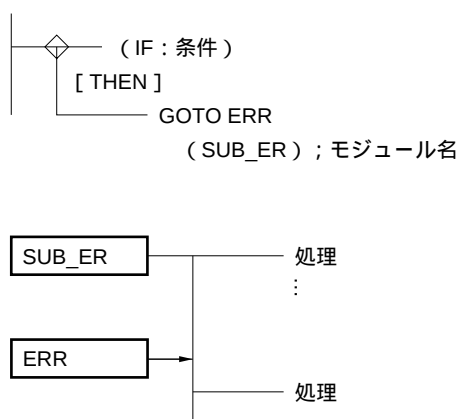
無条件に指定されたアドレスに分岐します。

●SPDチャート

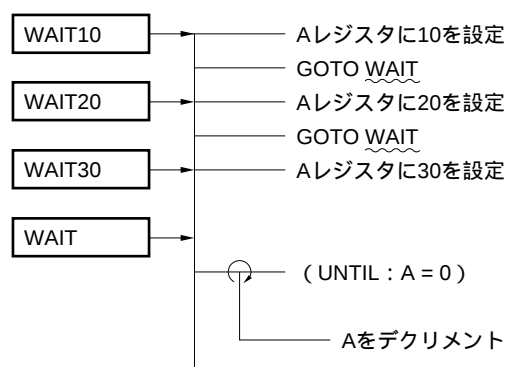
(1) 同じモジュールに分岐



(2) 異なるモジュールに分岐



例．サブルーチンの開始アドレスで，パラメータを選択しウエイトを設定する



9. 結合子 (継続)

1 モジュールのSPDが，複数のページにおよぶ場合に使用し，処理の流れを示します。

● SPDチャート



付録B 改版履歴

これまでの改版履歴を次に示します。なお，適用箇所は各版での章を示します。

(1/4)

版 数	前版からの主な改版内容	適用箇所
第 5 版	78K/0シリーズ アプリケーション・ノート（基礎編）から次の節を追加 2.7 2進乗算（16ビット×16ビット） 2.8 2進除算（32ビット÷16ビット） 8.5 半二重調歩同期通信 9.4 4チャンネル入力A/D変換	全般
	対象製品からμPD78044, 78054, 78064シリーズを削除（削除分は基礎編， に記載予定）	
	タイマ・クロック選択レジスタ2のフォーマットのウォッチドッグ・タイマの カウント・クロックの周波数を修正	第4章 ウォッチドッグ・タイマ の応用
		第7章 時計用タイマの応用
	ポート・モード・レジスタ3のフォーマットに，P30/TO0端子をタイマ出力と して使用する場合の注意事項を追加	第5章 16ビット・タイマ/イベ ント・カウンタの応用
	タイマ・クロック選択レジスタ3のフォーマットにμPD78002, 78002Yシ リーズの場合を追加	第8章 シリアル・インタフェー スの応用
	割り込みタイミング指定レジスタのフォーマットに，ウエイク・アップ機能を 使用する場合について注を追加	
	シリアル・バス・インタフェース・コントロール・レジスタのフォーマット に，ビジィ・モードの解除について注を追加	
	8.1.1 2線式シリアルI/Oモードの通信の（2）使用例を修正	
	8.2 OSD用LSI（μPD6451A）とのインタフェースを次のように修正 （1）パッケージの説明 使用するフラグ を削除 ネスティング を1レベル2バイトに修正 初期設定 からシリアル・インタフェース・チャンネル1割り込み許可を削除 起動方法 からTRENDを削除 （2）使用例 TRENDを削除 シリアル・インタフェース・チャンネル1割り込み許可を削除 （3）SPDチャート，（4）プログラム・リスト 転送終了状態の設定を削除	
	8.3.1 マスタCPUとしての応用の（4）プログラム・リストのうち，SBI データの転送処理のプログラムを修正	

版 数	前版からの主な改版内容	適用箇所
第 6 版	対象製品として次の製品を追加 μ PD78001B (A) , 78002B (A) μ PD78011B (A) , 78012B (A) , 78013 (A) , 78014 (A) μ PD78011F, 78012F, 78013F, 78014F, 78015F, 78016F, 78P018F μ PD78011FY, 78012FY, 78013FY, 78014FY, 78015FY, 78016FY, 78P018FY	全般
	タイマ・クロック選択レジスタ (TCL0, TCL1, TCL2, TCL3) のフォーマットに、同一データ以外に書き換える場合について注意事項を追加	
	16ビット・タイマ・モード・コントロール・レジスタのフォーマットに、TM0とCR00の一致でクリア&スタートするモードを選択した場合について注意事項を追加	第 5 章 16ビット・タイマ / イベント・カウンタの応用
	次のレジスタのフォーマットをサブシリーズごとに分けて記載 タイマ・クロック選択レジスタ 3 のフォーマット シリアル動作モード・レジスタ 0 のフォーマット 割り込みタイミング指定レジスタのフォーマット シリアル・バス・インタフェース・コントロール・レジスタのフォーマット	第 8 章 シリアル・インタフェースの応用
	付録 B 改版履歴 追加	付録 B 改版履歴
第 7 版	対象製品として次の製品を追加 μ PD780024, 780024Y, 780034, 780034Y, 78014H, 780924, 780964サブシリーズ, μ PD78018F, 78018FY, 780001, 78011F(A), 78012F(A), 78013F(A), 78014F(A), 78015F(A), 78016F(A), 78018F(A), 78P018F(A)	全般
	図 4 - 3 ウォッチドッグ・タイマ・クロック選択レジスタのフォーマットを追加	第 4 章 ウォッチドッグ・タイマの応用
	図 4 - 4 ウォッチドッグ・タイマ・モード・レジスタのフォーマットに注 2 , 注意 2 を追加	
	図 5 - 7 外部割り込みモード・レジスタのフォーマットに注意を追加	第 5 章 16ビット・タイマ / イベント・カウンタの応用
	次のレジスタ・フォーマットを追加 図 5 - 10 プリスケラ・モード・レジスタ 0 のフォーマット 図 5 - 11 ポート・モード・レジスタ 7 のフォーマット	
	次のレジスタ・フォーマットを追加 図 6 - 2 , 6 - 3 タイマ・クロック選択レジスタ 50 のフォーマット 図 6 - 4 , 6 - 5 タイマ・クロック選択レジスタ 51 のフォーマット 図 6 - 6 タイマ・クロック選択レジスタ 52 のフォーマット 図 6 - 8 8 ビット・タイマ・モード・コントロール・レジスタ 5n のフォーマット 図 6 - 9 8 ビット・タイマ・モード・コントロール・レジスタ 50 のフォーマット 図 6 - 10 8 ビット・タイマ・モード・コントロール・レジスタ 51 のフォーマット 図 6 - 11 8 ビット・タイマ・モード・コントロール・レジスタ 52 のフォーマット 図 6 - 14 ポート・モード・レジスタ 7 のフォーマット	第 6 章 8 ビット・タイマ / イベント・カウンタの応用

版 数	前版からの主な改版内容	適用箇所
第7版	表8-2 シリアル・インタフェースのレジスタ一覧を追加	第8章 シリアル・インタフェースの応用
	図8-6～8-8 シリアル動作モード・レジスタ0のフォーマットに注意，ウエイク・アップ機能の制御に注を追加	
	図8-19 自動データ送受信間隔指定レジスタのフォーマットに注意を追加	
	8.1 EEPROM™ (μPD6252) とのインタフェース μPD6252を保守製品化	
	8.1.2 I²Cバス・モードの通信に(5)，(6) I²Cバス・モード使用時の制限事項を追加	
	図9-2 A/Dコンバータ・モード・レジスタのフォーマットにHSCビットを追加	第9章 A/Dコンバータの応用
	図9-7 アナログ入力チャネル指定レジスタのフォーマットを追加	
第8版	対象製品として次の製品を追加 μPD780021A, 780022A, 780023A, 780024A, 780021A(A), 780022A(A), 780023A(A), 780024(A), 780021AY, 780022AY, 780023AY, 780024AY, 780021AY(A), 780022AY(A), 780023AY(A), 780024AY(A), 780031A, 780032A, 780033A, 780034A, 780031A(A), 780032A(A), 780033A(A), 780034A(A), 780031AY, 780032AY, 780033AY, 780034AY, 780031AY(A), 780032AY(A), 780033AY(A), 780034AY(A) 対象製品として次の製品を削除 μPD78001B, 78002B, 78001B(A), 78002B(A), 78001BY, 78002BY, 78011B, 78012B, 78013, 78014, 78P014, 78011B(A), 78012B(A), 78013(A), 78014(A), 78011BY, 78012BY, 78013Y, 78014Y, 78P014Y, 78001, 780021, 780022, 780023, 780024, 780021Y, 780022Y, 780023Y, 780024Y, 780031, 780032, 780033, 780034, 780031Y, 780032Y, 780033Y, 780034Y, 780921, 780922, 780923, 780924, 78F0924, 780961, 780962, 780963, 780964, 78F0964	全般
	図3-2 プロセッサ・クロック・コントロール・レジスタのフォーマット (μPD780024A, 780024AY, 780034A, 780034AYサブシリーズ) のCPUクロックの選択を一部変更，注意を追加	第3章 システム・クロック切り替えの応用
	図4-4 ウォッチドッグ・タイマ・モード・レジスタのフォーマット (μPD780024A, 780024AY, 780034A, 780034AYサブシリーズ) に注を追加	第4章 ウォッチドッグ・タイマの応用
	図5-3 16ビット・タイマ・モード・コントロール・レジスタのフォーマット (μPD780024A, 780024AY, 780034A, 780034AYサブシリーズ) のTOCの出力タイミングの選択を一部変更，注意を一部変更	第5章 16ビット・タイマ/イベント・カウンタの応用
	図5-5 16ビット・タイマ出力コントロール・レジスタのフォーマット (μPD780024A, 780024AY, 780034A, 780034AYサブシリーズ) のビット5, 6を変更，注意の一部を削除	
	図5-9 キャプチャ/コンペア・コントロール・レジスタ0のフォーマット (μPD780024A, 780024AY, 780034A, 780034AYサブシリーズ) の注意を追加	

(4/4)

版 数	前版からの主な改版内容	適用箇所
第 8 版	図 5 - 10 プリスケラ・モード・レジスタ0のフォーマット (μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズ)の注意を追加	第 5 章 16 ビット・タイマ / イベント・カウンタの応用
	図 6 - 3 タイマ・クロック選択レジスタ51のフォーマット(μ PD780024A, 780024AY, 780034A, 780034AYサブシリーズ)の備考を追加	第 6 章 8 ビット・タイマ / イベント・カウンタの応用
	図 9 - 2 A/Dコンバータ・モード・レジスタのフォーマット (μ PD780024A, 780024AYサブシリーズ)の注意を追加	第 9 章 A/Dコンバータの応用

〔メ モ〕

— お問い合わせ先 —

【技術的なお問い合わせ先】

NEC半導体テクニカルホットライン
(電話 : 午前 9:00 ~ 12:00 , 午後 1:00 ~ 5:00)

電 話 : 044-435-9494
FAX : 044-435-9608
E-mail : s-info@saed.tmg.nec.co.jp

【営業関係お問い合わせ先】

第一販売事業部

東 京 (03)3798-6106, 6107,
6108
大 阪 (06)6945-3178, 3200,
3208, 3212
仙 台 (022)267-8740
郡 山 (024)923-5591
千 葉 (043)238-8116

第二販売事業部

東 京 (03)3798-6110, 6111,
6112
立 川 (042)526-5981, 6167
松 本 (0263)35-1662
静 岡 (054)254-4794
金 沢 (076)232-7303
松 山 (089)945-4149

第三販売事業部

東 京 (03)3798-6151, 6155, 6586,
1622, 1623, 6156
水 戸 (029)226-1702
広 島 (082)242-5504
前 橋 (027)243-6060
鳥 取 (0857)27-5313
太 田 (0276)46-4014
名古屋 (052)222-2170, 2190
福 岡 (092)261-2806

【資料の請求先】

上記営業関係お問い合わせ先またはNEC特約店へお申しつけください。

【NECエレクトロニクス デバイス ホームページ】

NECエレクトロニクスデバイスの情報がインターネットでご覧になれます。

URL(アドレス)

<http://www.ic.nec.co.jp/>

キリトリ

[ドキュメント名] 78K0シリーズ アプリケーション・ノート 基礎編 ()
(U12704JJ8V0AN00 (第 8 版))

御社名（学校名，その他）（
ご住所（
お電話番号（
お仕事の内容（
お名前（

項 目	大変良い	良 い	普 通	悪 い	大変悪い
全体の構成					
説明内容					
用語解説					
調べやすさ					
デザイン，字の大きさなど					
その他（ （					

3. わかりにくい所(第 章,第 章,第 章,第 章,その他)

理由 []

--

2000.6