お客様各位

カタログ等資料中の旧社名の扱いについて

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

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

2010年4月1日
ルネサスエレクトロニクス株式会社

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

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

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したものですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。

   標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット

   高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）

   特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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

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

アプリケーション・ノート

RENESAS

μPD750008サブシリーズ

4ビット・シングルチップ・マイクロコンピュータ

μPD750004
μPD750006
μPD750008
μPD75P0016

© NEC Corporation 1995

資料番号 U10452JJ1V0AN00（第1版）
発行年月 October 1995 P

〔メ　モ〕

本製品が外国為替および外国貿易管理法の規定による戦略物資等（または役務）に該当するか否かは，ユーザ（仕様を決定した者）が判定してください。

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，後日変更する場合があります。
○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所有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。
○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。

○この製品は耐放射線設計をしておりません。

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μPD750008サブシリーズの機能を理解し，それ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ます。

μPD750008サブシリーズとは，μPD750004, 750006, 750008, 75P0016の総称です。

目　的　このアプリケーション・ノートは，μPD750008サブシリーズの持つ機能を，応用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。

読み方　このアプリケーション・ノートをご利用なさるお客様には，電気，論理回路，およびマイクロコンピュータに関する一般知識を必要とします。

また，特に機能面で違いがない限りは，μPD750008を代表品種として記述しています。

μPD750004, 750006, 75P0016のマニュアルとしてお使いの場合は，μPD750008をそれぞれの品種に読み替えてご利用ください。

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・・・××××または××××B
	10進数・・・××××
	16進数・・・××××H

関連資料

デバイス関連資料

資料 / 製品	パンフレット	データ・シート	ユーザーズ・マニュアル	インストラクション活用表	アプリケーション・ノート
μPD750004	IF-6367	U10738J	IEU-884	IEM-5593	U10452J（この資料）
μPD750006					
μPD750008					
μPD75P0016		U10328J			

〔メ　モ〕

目　　次

図の目次（1/2）

図の目次（2/2）

表の目次

〔メ　モ〕

第1章　概　　説

μPD750008サブシリーズは，豊富な製品展開を誇る75Xシリーズの後継品種，75XLシリーズの4ビット・シングルチップ・マイクロコンピュータです。

μPD750008サブシリーズの各製品間の違いを表1－1に示します。

表1－1　μPD750008サブシリーズの各製品間の違い

品　名	プログラム・メモリ（ROM）	ROMの構成	プログラム・カウンタ
μPD750004	4096バイト	マスクROM	12ビット
μPD750006	6144バイト		13ビット
μPD750008	8192バイト		
μPD75P0016	16384バイト	ワン・タイムPROM	14ビット

1.1　μPD75008とμPD750008との違い

μPD750008は，従来製品であるμPD75008（75Xシリーズ）の機能やインストラクションを継承しており，置き換えが容易です。

表1－2にμPD75008とμPD750008との違いを示します。

表1－2 μPD75008とμPD750008との違い（1/2）

項目		μPD75008	μPD750008
プログラム・メモリ		0000H-1F7FH 8064×8ビット	0000H-1FFFH 8192×8ビット
データ・メモリ		000H-1FFH（512×4ビット）	
CPU		75X Standard	75XL
発振安定ウエイト時間		31.3 msに固定	2^{15}/fx, 2^{17}/fx[注] （マスク・オプションで選択可能）
命令実行時間	メイン・システム・クロック選択時	0.95, 1.91, 15.3 μs （4.19 MHz動作のみ）	・0.95, 1.91, 3.81, 15.3 μs （4.19 MHz動作時） ・0.67, 1.33, 2.67, 10.7 μs （6.0 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）	
スタック	SBSレジスタ	なし	あり ・SBS.3＝1：Mk I モード選択 ・SBS.3＝0：Mk II モード選択
	スタック・エリア	000H-0FFH	n00H-nFFH（n＝0, 1）
	サブルーチン・コール命令のスタック動作	2バイト・スタック	Mk I モード時：2バイト・スタック Mk II モード時：3バイト・スタック
命令	BRA !addr1動作 CALLA !addr1動作	使用不可	Mk I モード時：使用不可 Mk II モード時：使用可能

	MOVT XA, @BCDE
	MOVT XA, @BCXA
	BR BCDE
	BR BCXA
			使用可能

	CALL !addr	3マシン・サイクル	Mk I モード時：3マシン・サイクル Mk II モード時：4マシン・サイクル
	CALLF !faddr	2マシン・サイクル	Mk I モード時：2マシン・サイクル Mk II モード時：3マシン・サイクル
タイマ		3チャネル ・ベーシック・インターバル・タイマ ・タイマ／イベント・カウンタ ・時計用タイマ	4チャネル ・ベーシック・インターバル・タイマ／ウォッチドッグ・タイマ ・タイマ／イベント・カウンタ ・タイマ・カウンタ ・時計用タイマ
クロック出力（PCL）		・Φ, 524, 262, 65.5 kHz （4.19 MHz動作時）	・Φ, 524, 262, 65.5 kHz （4.19 MHz動作時） ・Φ, 750, 375, 93.7 kHz （6.0 MHz動作時）

注 2^{15}/fxは，6.0 MHz時：5.46 ms，4.19 MHz時：7.81 ms。

2^{17}/fxは，6.0 MHz時：21.8 ms，4.19 MHz時：31.3 ms。

表1－2　μPD75008とμPD750008との違い（2/2）

項　　目		μPD75008	μPD750008
ブザー出力（BUZ）		2 kHz	・2.4, 32 kHz （4.19 MHz動作時） ・2.86, 5.72, 45.8 kHz （6.0 MHz動作時）
シリアル・インタフェース		3種類のモードに対応可能 ・3線式シリアルI/Oモード…MSB/LSB先頭切り替え可能 ・2線式シリアルI/Oモード ・SBIモード	
SOSレジスタ	フィードバック抵抗カット・フラグ（SOS.0）	マスク・オプションによりフィードバック抵抗を内蔵可能	内蔵
	サブ発振器電流カット・フラグ（SOS.1）	なし	
レジスタ・バンク選択レジスタ（RBS）		なし	あり
INT0によるスタンバイ解除		不可	可能
ベクタ割り込み		外部：3本，内部：3本	外部：3本，内部：4本
プロセッサ・クロック・コントロール・レジスタ（PCC）		PCC = 0, 2, 3を使用可能	PCC = 0-3を使用可能
電源電圧		V_{DD} = 2.7～6.0 V	V_{DD} = 2.2～5.5 V
パッケージ		・42ピン・プラスチック・シュリンクDIP（600 mil） ・44ピン・プラスチックQFP（□10 mm）	

1.2 MkⅠモードとMkⅡモードの切り替え

1.2.1 MkⅠモードとMkⅡモードの使用法

μPD750008サブシリーズのCPUはMkⅠモードとMkⅡモードの2つのモードを持ち，どちらを使用するかの選択ができます。モードの切り替え操作は，スタック・バンク選択レジスタ（SBS）のビット3で行います。

表1－3にMkⅠモードとMkⅡモードの違いを，また図1－1にスタック・バンク選択レジスタのフォーマットを示します。

・MkⅠモード：μPD75008サブシリーズと上位互換性があります。
　ROM容量が16 Kバイトまでの75XL CPUで使用できます。

・MkⅡモード：μPD75008サブシリーズとの互換性がありません。
　ROM容量が16 Kバイト以上の製品も含め，75XL CPU全部で使用できます。

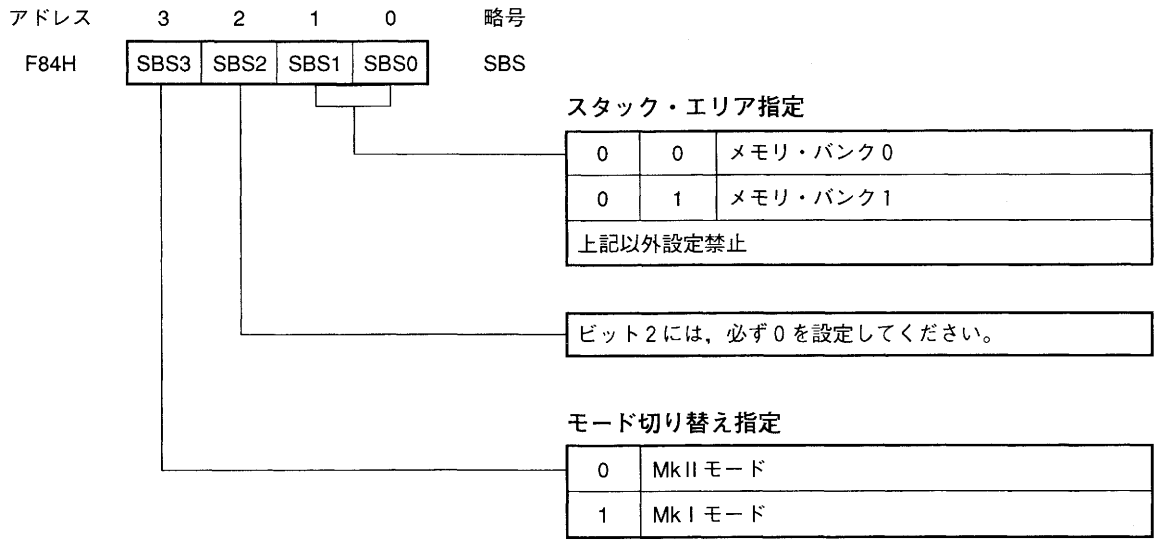
表1－3　MkⅠモードとMkⅡモードの違い

	MkⅠモード	MkⅡモード
サブルーチン命令のスタック・バイト数	2バイト	3バイト
BRA !addr1命令 CALLA !addr1命令	不定動作	正常動作
CALL !addr命令	3マシン・サイクル	4マシン・サイクル
CALLF !faddr命令	2マシン・サイクル	3マシン・サイクル

備考 MkⅡモードは，プログラム・メモリが24 Kバイト以上の75Xシリーズや75XLシリーズとのソフトウエア上の互換性を保つためのものです。
したがって，ROM効率やスピードを重視する場合はMkⅠモードを使用してください。

スタック・バンク選択レジスタは，4ビット・メモリ操作命令により設定します。MkⅠモードを使用する場合は，プログラムの初期で必ずスタック・バンク選択レジスタを10××B[注]にイニシャライズしてください。またMkⅡモードを使用する場合は，必ず00××B[注]にイニシャライズしてください。

図1－1　スタック・バンク選択レジスタのフォーマット



注　××には希望の値を設定してください。

注意　SBS.3は$\overline{\text{RESET}}$信号発生後“1”になるので，CPUはMkⅠモードで動作します。MkⅡモードの命令を使用する場合は，SBS.3を“0”にし，MkⅡモードに設定してから使用してください。

1.2.2　レジスタ・バンクの使用法

μPD750008サブシリーズは，X,A,B,C,D,E,H,Lの8つの汎用レジスタを1バンクとして，4つのレジスタ・バンクを内蔵しています。この汎用レジスタ・エリアはデータ・メモリのメモリ・バンク0の00H-1FH番地にマッピングされています（図1－3参照）。この汎用レジスタのバンクを指定するためにレジスタ・バンク許可フラグ（RBE）とレジスタ・バンク選択レジスタ（RBS）が内蔵されています。RBSはレジスタ・バンクを選択するためのレジスタで，RBEはRBSで選択されたレジスタ・バンクを有効とするか否かを決定するフラグです。命令実行の際に有効となるレジスタ・バンク（RB）は，次のようになります。

RB = RBE·RBS

表1－4　RBE, RBSと選択されるレジスタ・バンク

RBE	RBS				レジスタ・バンク
	3	2	1	0	
0	0	0	×	×	バンク0に固定
1	0	0	0	0	バンク0を選択
			0	1	バンク1 〃
			1	0	バンク2 〃
			1	1	バンク3 〃

0に固定

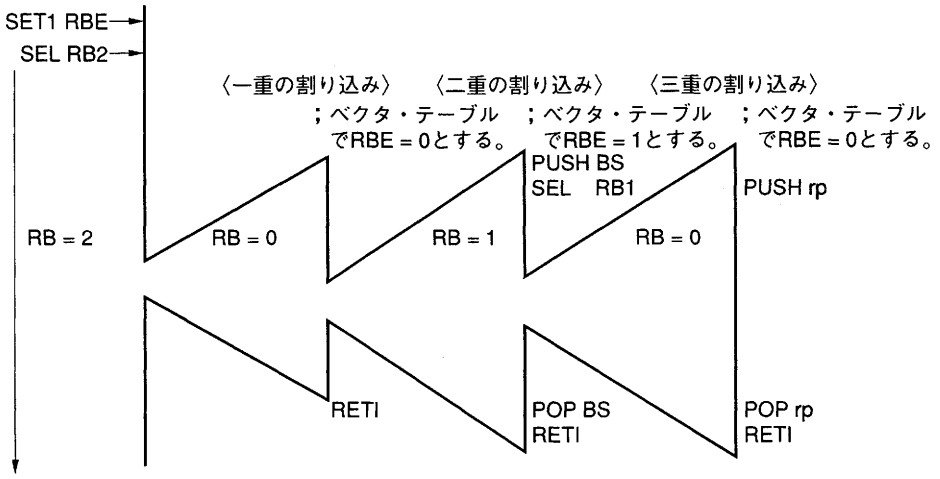
備考　×：don't care

RBEは，サブルーチン処理時に自動的に退避／復帰されますので，サブルーチン処理中に自由に設定することができます。また，割り込み処理時は，自動的に退避／復帰されるとともに，割り込みベクタ・テーブルの設定によって，割り込み処理開始と同時に，割り込み処理中のRBEを設定することができます。したがって，表1－5に示すように，通常処理と割り込み処理で，レジスタ・バンクを使い分ければ，一重割り込みでは，汎用レジスタの退避／復帰は不要，二重割り込みでは，RBSの退避／復帰のみとなり，割り込み処理の高速化が図れます。

表1－5　通常ルーチンと割り込みルーチンでのレジスタ・バンクの使い分けの例

通常の処理	RBE = 1とし，レジスタ・バンク2，3を使用する。
1重割り込み処理	RBE = 0とし，レジスタ・バンク0を使用する。
2重割り込み処理	RBE = 1とし，レジスタ・バンク1を使用する。 （このとき，RBSの退避／復帰が必要）
3重以上の割り込み処理	PUSH，POPでレジスタ退避をする。

図1－2　レジスタ・バンクの使い分け例



RBSを，サブルーチン処理あるいは割り込み処理で変更する場合は，PUSH/POP命令によって退避／復帰します。

RBEの設定は，SET1/CLR1命令によって行います。RBSの設定は，SEL命令により行います。

```
例　SET1　RBE　；RBE←1
    CLR1　RBE　；RBE←0
    SEL　 RB0　；RBS←0
    SEL　 RB3　；RBS←3
```

μPD750008に内蔵されている汎用レジスタ・エリアは，4ビット・レジスタとしての使用のほかに，レジスタ・ペアによる8ビット・レジスタとして使用ができ，8ビット・マイコンに匹敵する転送，演算，比較，増減命令によって，汎用レジスタ中心のプログラミングが可能となります。

（1）4ビット・レジスタとして使用する場合

汎用レジスタ・エリアを4ビット・レジスタとして使用する場合には，図1－3に示すように，RB＝RBE･RBSで指定されたレジスタ・バンクのX, A, B, C, D, E, H, L, 計8個の汎用レジスタを使うことができます。このうちAレジスタは4ビット・アキュームレータとして，4ビット・データの転送，演算，比較などに中心的な働きをします。ほかの汎用レジスタは，アキュームレータとの転送，比較，増減ができます。

（2）8ビット・レジスタとして使用する場合

汎用レジスタ・エリアを8ビット・レジスタとして使用する場合には，図1－4に示すように，RB＝RBE･RBSで指定されたレジスタ・バンクのレジスタ・ペアをXA, BC, DE, HLとし，レジスタ・バンク（RB）のビット0を反転したレジスタ・バンクのレジスタ・ペアをXA', BC', DE', HL'として計8個の8ビット・レジスタを使うことができます。このうちXAレジスタ・ペアは，8ビット・アキュームレータとして，8ビット・データの転送，演算，比較などに中心的な働きをします。ほかのレジスタ・ペアは，アキュームレータとの転送，演算，比較，増減ができます。また，HLレジスタ・ペアは，主にデータ・ポインタとして機能します。DE, DLレジスタ・ペアも補助的なデータ・ポインタとして機能します。

```
例1．  INCS    HL          ;HL←HL＋1，HL＝00Hでスキップ
       ADDS    XA，BC      ;XA←XA＋BC，キャリーでスキップ
       SUBC    DE'，XA     ;DE'←DE'－XA－CY
       MOV     XA，XA'     ;XA←XA'
       MOVT    XA，@PCDE   ;XA←（PC12-8＋DE）ROM，テーブル参照
       SKE     XA，BC      ;XA＝BCならスキップ
```

2．タイマ／イベント・カウンタ0のカウント・レジスタ（T0）の値がBC'レジスタ・ペアの値より大きいかどうかをテストし大きくなるまで待つ。

```
       CLR1    MBE         ;
NO：   MOV     XA，T0      ;カウント・レジスタ読み取り
       SUBS    XA，BC'     ;XA≧BC?
       BR      YES         ;YES
       BR      NO          ;NO
```

図1－3　汎用レジスタの構成（4ビット処理の場合）

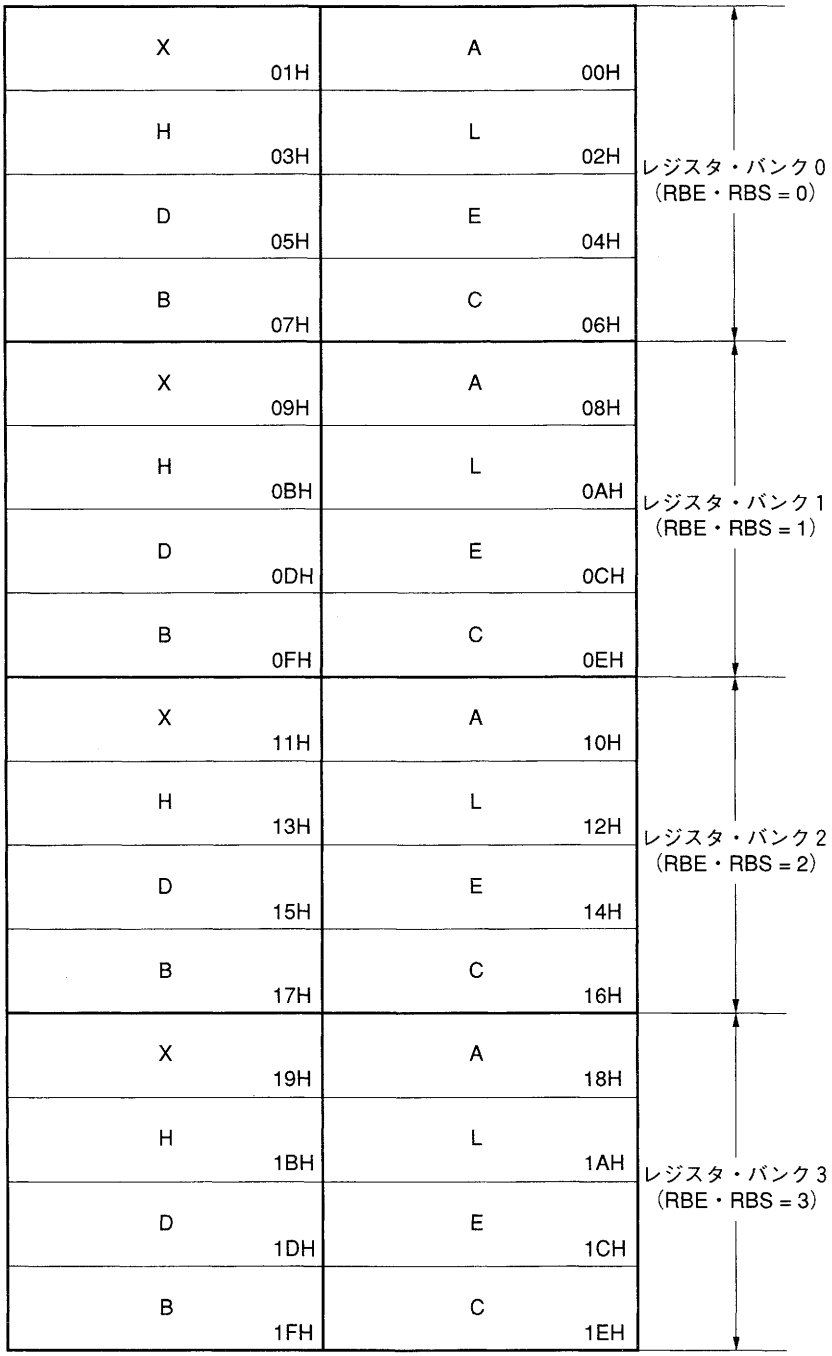
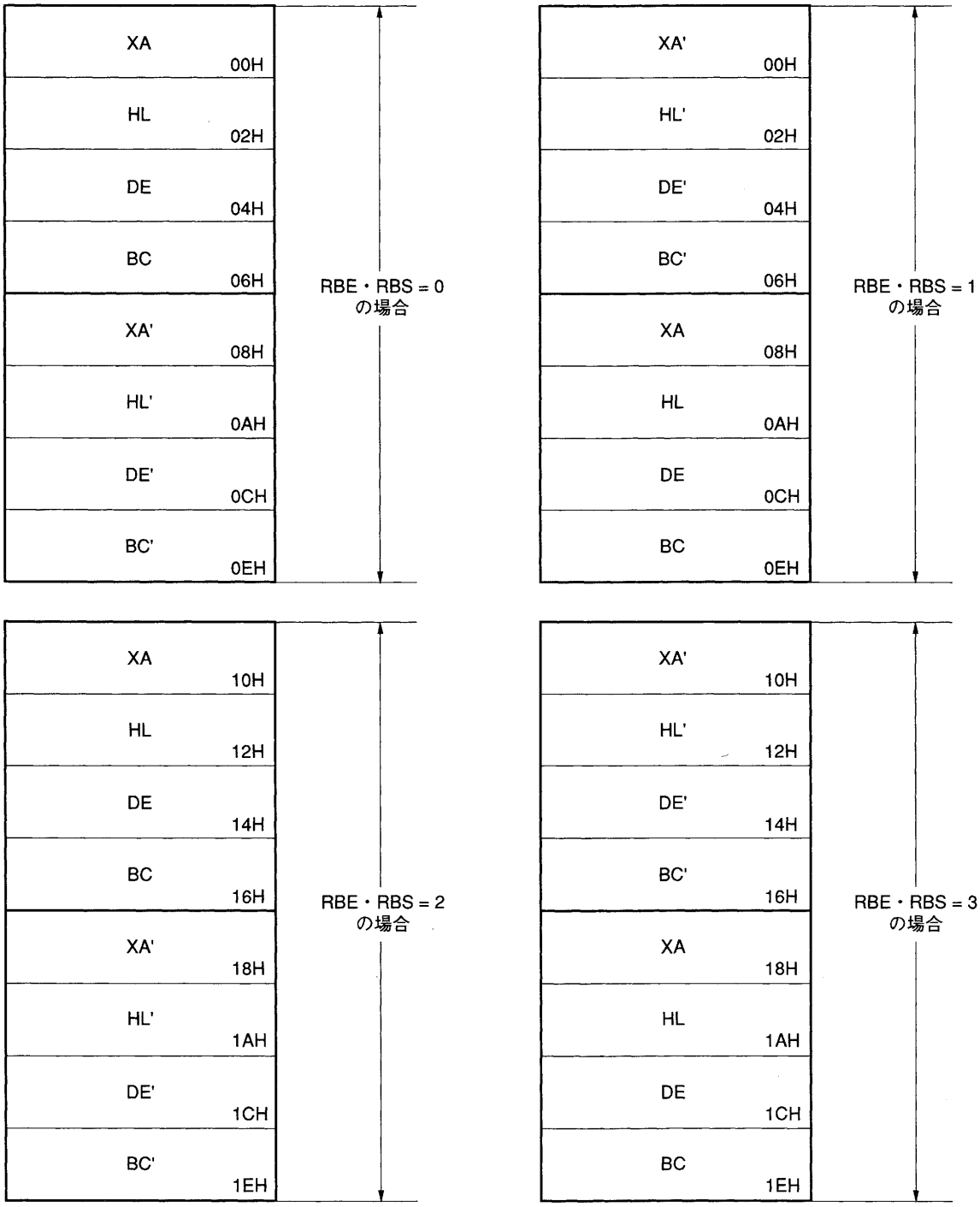


図1－4　汎用レジスタの構成（8ビット処理の場合）



1.3　アプリケーション・プログラムの説明

このアプリケーション・ノートの各章では，応用プログラムを機能別にパッケージ化して記述しています。したがって，リンカを用いてユーザ・プログラム（メイン・プログラム）と組み合わせれば，システム・プログラムの一部として機能させることもできます。

また，**第11章　このアプリケーション・プログラムの応用例**では，第10章までに説明した応用プログラムのいくつかを使用し，1つのシステム・プログラムを作成しています。

各パッケージを使用するにあたっては，そのパッケージの**プログラムの説明**に従ってください。**プログラムの説明**の各項目の内容を次に示します。

〈パブリック宣言シンボル〉　：パッケージで使用するサブルーチンを示します。
このシンボルを外部参照宣言すると，そのパッケージ内で参照することができます。

〈外部参照宣言シンボル〉　：パッケージで使用するデータ・メモリの領域を示します。
このシンボルを定義し，パブリック宣言すると，そのパッケージ内で使用することができます。

〈使用するレジスタ〉　：パッケージで使用するレジスタを示します。

〈使用するRAM〉　：パッケージで使用するデータ・メモリの領域を示します。

〈ネスティング〉　：パッケージで許可されたネスティング・レベルを示します。
（　）内の値は使用するスタックの最大値を示します。

〈使用するハードウエア〉　：パッケージで使用するハードウエアを示します。

〈割り込み〉　：パッケージで使用する割り込みを示します。

〈初期設定〉　：パッケージを動作させるために必要な初期設定を示します。
ただし，各パッケージでは，特に記述がない場合はSCC = 0, PCC = 3となっています。

〈起動方法〉　：パッケージを動作させるために必要な処置を示します。

各パッケージは，MkIIモードで動作します。レジスタ・バンクは次の各割り込みで切り替えられます。

RBE = 0, RBS = 0　：メイン・ルーチン
RBE = 1, RBS = 1　：INT0割り込み（リモコンの受信）
RBE = 1, RBS = 2　：ベーシック・インターバル・タイマ
RBE = 1, RBS = 3　：SBI

次に，プログラムの記述例を示します。

```
        VENT0   MBE=0,RBE=0,INIT        ;RESET
        VENT1   MBE=0,RBE=1,INTBT       ;INTBT/INT4
        VENT2   MBE=0,RBE=1,INT0        ;INT0
        VENT4   MBE=0,RBE=1,SBI         ;SBI
;*********************************
;       初期設定
;*********************************
INIT:
        DI                              ;すべての割り込みを禁止
        CLR1    MBE                     ;MBE←0
        MOV     XA,#00H                 ;
        MOV     SP,XA                   ;スタック・ポインタ←00H
        MOV     SBS,A                   ;メモリ・バンク←0,MkIIモードを選択
;+++++++++++++++++++++++++++++++++
;       リモコン処理
;+++++++++++++++++++++++++++++++++
INT0:
        DI                              ;
        PUSH    BS                      ;
        SEL     RB1                     ;レジスタ・バンク←1
;+++++++++++++++++++++++++++++++++
;       タイマ
;+++++++++++++++++++++++++++++++++
INTBT:
        SEL     RB2                     ;レジスタ・バンク←2
;+++++++++++++++++++++++++++++++++
;       SBI処理
;+++++++++++++++++++++++++++++++++
SBI;
        SEL     RB3                     ;レジスタ・バンク←3
```

第2章　システム・クロック切り替え機能の応用

μPD750008は，プロセッサ・クロック・コントロール・レジスタ（PCC）とシステム・クロック・コントロール・レジスタ（SCC）を書き換えることにより，CPUクロックおよびシステム・クロックを切り替えることが可能です。図2－1にPCC，図2－2にSCCのフォーマットを示します。

図2－1　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

アドレス	3	2	1	0	略号
FB3H	PCC3	PCC2	PCC1	PCC0	PCC

CPUクロック選択ビット

（fx = 6.0 MHz時の場合）

		SCC3, SCC0 = 00 （ ）内はfx = 6.0 MHz時		SCC3, SCC0 = 01 or 11 （ ）内はfXT = 32.768 kHz時	
		CPUクロック周波数	1マシン・サイクル	CPUクロック周波数	1マシン・サイクル
0	0	Φ = fx/64 (93.7 kHz)	10.7 μs	Φ = fXT/4 (8.192 kHz)	122 μs
0	1	Φ = fx/16 (375 kHz)	2.67 μs		
1	0	Φ = fx/8 (750 kHz)	1.33 μs		
1	1	Φ = fx/4 (1.5 MHz)	0.67 μs		

（fx = 4.19 MHz時の場合）

		SCC3, SCC0 = 00 （ ）内はfx = 4.19 MHz時		SCC3, SCC0 = 01 or 11 （ ）内はfXT = 32.768 kHz時	
		CPUクロック周波数	1マシン・サイクル	CPUクロック周波数	1マシン・サイクル
0	0	Φ = fx/64 (65.5 kHz)	15.3 μs	Φ = fXT/4 (8.192 kHz)	122 μs
0	1	Φ = fx/16 (261.8 kHz)	3.82 μs		
1	0	Φ = fx/8 (524 kHz)	1.91 μs		
1	1	Φ = fx/4 (1.05 MHz)	0.95 μs		

備考1． fx：メイン・システム・クロック発振回路出力周波数

2． fXT：サブシステム・クロック発振回路出力周波数

CPU動作モード制御ビット

0	0	通常動作モード
0	1	HALTモード
1	0	STOPモード
1	1	設定禁止

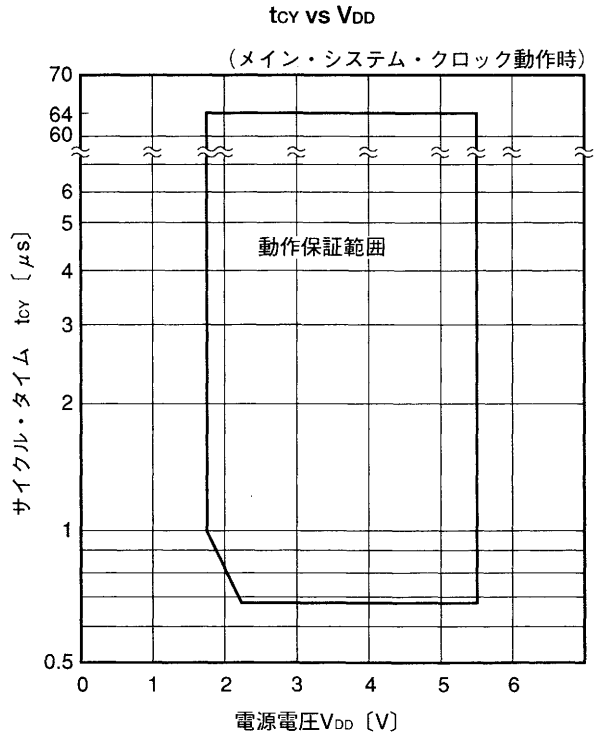
図2－2　システム・クロック・コントロール・レジスタのフォーマット

アドレス	3	2	1	0	略号
FB7H	SCC3	－	－	SCC0	SCC

SCC3	SCC0	CPUクロックの選択	メイン・システム・クロックの発振
0	0	メイン・システム・クロック	発振可能
0	1	サブシステム・クロック	
1	0	設定禁止	
1	1	サブシステム・クロック	発振停止

図2－3に電源電圧（V_{DD}）に対する最小命令実行時間（t_{CY}）を示します。

図2－3　電源電圧（V_{DD}）に対する最小命令実行時間（t_{CY}）



2.1　$\overline{\text{RESET}}$後のPCCの切り替え

$\overline{\text{RESET}}$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，高速処理を行う場合には，PCCを書き換えてCPUクロックを最高速モードとします。ただしPCCを書き換える以前に，V_{DD}端子電圧が高速で動作できる電圧まで，上昇していなければなりません（図2－3参照）。またサブシステム・クロックを使用するシステムではあらかじめ，サブシステム・クロックの発振が開始していることを確認しておく必要があります。

ここでは，時計用タイマを用いて，V_{DD}端子電圧が上昇するまでのウエイトおよびサブシステム・クロックの発振の確認を行い，CPUクロックを最高速に切り替えるプログラム例（f_X = 4.19 MHz，f_{XT}= 32.768 kHz時）を示します。また図2－4にそのタイミング図を示します。

・プログラム例

（1）V_{DD}電圧上昇までのウエイト

```
        EXTRN   BIT(SSCOKF)          ;サブシステム・クロック発振確認フラグ(初期値 = 0)
        VENTO   MBE=0,RBE=0,START
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
START   CSEG    INBLOCK              ;プログラム・メモリの4Kバイトのブロック内に配置することを指定
        MOV     XA,#00000110B        ;時計用タイマ早送りモード
        MOV     WM,XA                ;(メイン・システム・クロックで動作)
INILOP1:
        SKTCLR  IRQW                 ;3.9 msウエイト注
        BR      INILOP1
        MOV     A,#0010B             ;ドライブ電流少
        MOV     SOS,A
        MOV     XA,#00000110B        ;時計用タイマ・サブシステム・クロックで動作
        MOV     WM,XA
        MOV     A,#0011B             ;CPUクロック最高速
        MOV     PCC,A                ;以下イニシャライズ処理
```

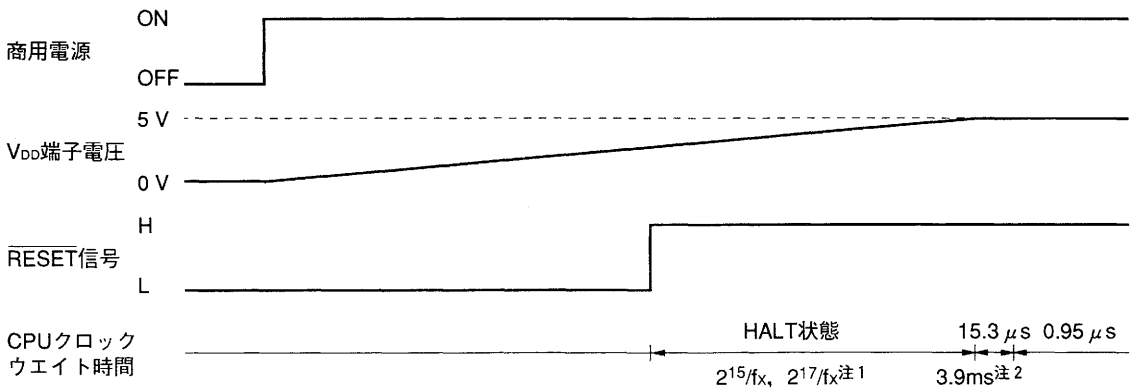
注　この時間はハードウエアに依存します。各システムに合った時間を設定してください。

（2）サブシステム・クロック発振確認サブルーチン（ユーザ処理にてコールします）

```
SSCCHK  CSEG    INBLOCK
        SKF     SSCOKF          ;すでに1回サブシステム・クロックの発振を確認した場合は何もしない
        RET
        SKTCLR  IRQW            ;サブシステム・クロック発振
        RET
        MOV     XA,#00000100B   ;時計用タイマ通常動作モード
        MOV     WM,XA
        SET1    SSCOKF          ;サブシステム・クロック発振確認フラグをセット
        RET
        END
```

サブシステム・クロックは，発振が安定するまでに約1 sかかるので，イニシャライズ処理内で発振が安定するまでウエイトせずに，メイン・ルーチン内でサブシステム・クロック発振確認の処理を行い，フラグをセットすることによって，サブシステム・クロックが発振を開始していることを他の処理に知らせます。

図2－4　$\overline{\text{RESET}}$後のCPUクロック切り替え



注1．マスク・オプションにより，選択することができます（4.19 MHz動作時は，$2^{15}/fx$ = 7.81 ms，$2^{17}/fx$ = 31.3 ms）。

2．この時間はハードウエアに依存します。各システムに合った時間を設定してください。

2.2　商用電源の停電検出時のシステム・クロックの切り替え

μPD750008は，SCCの設定によりサブシステム・クロック（32.768 kHz）を選択し，超低消費電流で動作させることができます。したがってNiCd電池やスーパ・キャパシタなどのバックアップ用電源をシステム上に付加することにより，停電時に，時計カウントなどの処理を超低消費電力で継続することができます。

ここでは，商用電源のオン／オフを外部割り込みINT4により検出し，システム・クロックを切り替え，超低消費電力で時計機能を継続するプログラム例を示します。

図2－5および図2－6に従ってシステム・クロックの切り替え手順を説明します。

図2－5　商用電源オン／オフ時のシステム・クロックの切り替え

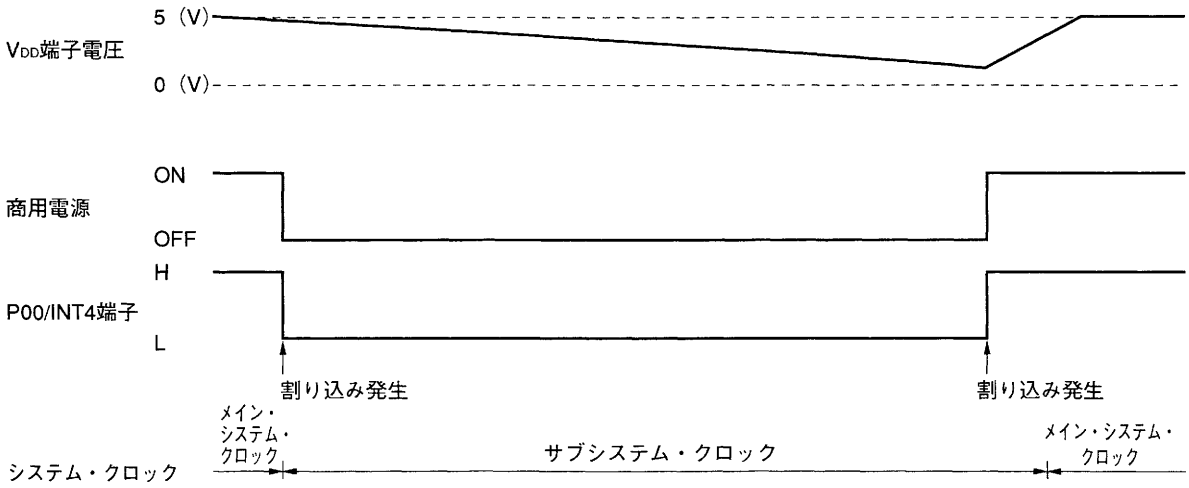
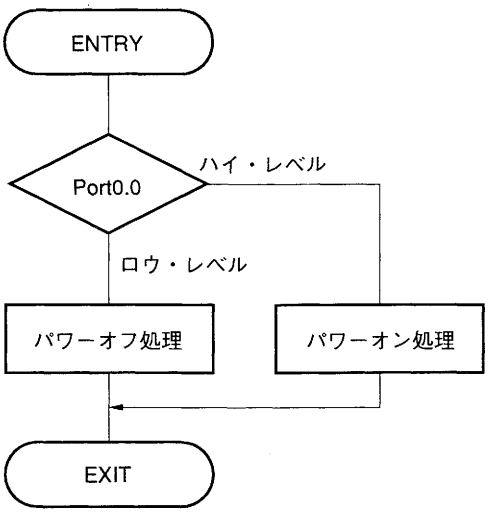


図2－6　INT4割り込み処理のアルゴリズム



2.2.1　パワーオフ処理

INT4割り込み処理の中でPort0.0をチェックし，ロウ・レベルのときは商用電源オフと判断し，パワーオフ処理を行います。

次にパワーオフ処理の処理手順を示します。

① PCCに0011Bを設定します。

② システム・クロックをサブシステム・クロックにするためSCCのビット0をセットします。

システム・クロックが変化する前に時計用タイマとベーシック・インターバル・タイマ（停止することができない）を除く周辺ハードウエアを停止させます。

③ 入出力端子を消費電流が最小となるように処理します。

④ SCCのビット0をセットしてから32マシン・サイクル以上経過後，SCCのビット3をセットし，メイン・システム・クロックの発振を停止させます。

注意　パワーオフ時の端子の処理について

商用電源がオフになった場合，通常周辺回路の電源はオフになります。そのため，接続されている周辺回路により，次のような処置が必要です。

1．端子に接続されている周辺回路がハイ・インピーダンスになる場合

・入出力が切り替えられる端子は出力モードとし，ロウ・レベルを出力します。

・入力端子は，あらかじめプルダウンまたはプルアップしておきます。

2．端子に接続されている周辺回路がハイ・インピーダンスにならない場合

・出力端子は電流が流れないレベルを出力します。

・入力端子は入力がハイ・レベルまたはロウ・レベルで安定していれば，特に処置をする必要はありません。

その他，周辺回路の状況により必要な処置を行ってください。

2.2.2　パワーオン処理

INT4割り込み処理の中で，Port0.0をチェックし，ハイ・レベルのときは商用電源オンと判断し，パワーオン処理を行います。

次にパワーオン処理の処理手順を示します。

① V_{DD}端子電圧が最高速で動作できる電圧に上昇するまでウエイトします。

② SCCのビット3をクリアしてメイン・システム・クロックの発振を開始します。

③ メイン・システム・クロックの発振が安定するのに必要な時間だけウエイトし，SCCのビット0をクリアして，システム・クロックを切り替えます。

2.2.3　パワーオン／オフ処理の応用

パワーオン／オフ処理を使用したプログラムの例を次に示します。

このプログラムは，**第11章　このアプリケーション・プログラムの応用例**には使用されていません。

（1）プログラムの説明

〈外部参照宣言シンボル〉

SSCOKF ：サブシステム・クロック発振確認フラグ
INTD0-5 ：割り込み関係イニシャライズ・データ
INDBTM ：BTMイニシャライズ・データ
INTM0 ：TM0イニシャライズ・データ
INCSIM ：CSIMイニシャライズ・データ
INWM ：WMイニシャライズ・データ
WATCH ：時計カウント処理エントリ・ラベル

〈使用するレジスタ〉

XA, HL

〈ネスティング〉

4レベル（20ワード）

〈初期設定〉

INT4割り込み許可

〈起動方法〉

（a）パワーオン／オフ処理

INT4割り込みにより起動したとき，Port0.0の状態により次の2つの処理に分かれます。

（i）Port0.0がハイ・レベルの場合（パワーオン処理）

システム・クロックをメイン・システム・クロックに切り替え，各周辺ハードウエアを起動します。

（ii）Port0.0がロウ・レベルの場合（パワーオフ処理）

・サブシステム・クロックが発振しているとき（SSCOKF = 1）

時計用タイマ，ベーシック・インターバル・タイマを除く周辺ハードウエアを停止します。その後，システム・クロックをサブシステム・クロックに切り替え，メイン・システム・クロックを停止します。

・サブシステム・クロックが停止しているとき（SSCOKF = 0）

時計用タイマ，ベーシック・インターバル・タイマを除く周辺ハードウエアを停止します。

（b）時計カウント・チェック・サブルーチン

TIMCNTをコールすると，サブシステム・クロックが発振しているときには時計カウントの処理ルーチン（ユーザ・プログラム）をコールします。サブシステム・クロックが停止しているときには，時計カウントの処理ルーチンをコールせずにリターンします。

また，パワーオフ（Port0.0がロウ・レベル）の間は次の処理を繰り返し，パワーオン（Port0.0がハイ・レベル）になったときにリターンします。

（i）HALTモードの設定

（ii）IRQWにより復帰

（iii）時計カウントの処理ルーチンのコール

（２）プログラム例

〈INT4割り込み処理〉

```
        VENT1   MBE=0,RBE=0,INT4
        EXTRN   NUMBER(INTD0,INTD1,INTD2,INTD3,INTD4,INTD5)
        EXTRN   NUMBER(INDBTM,INTM0,INCSIM,INWM)
        EXTRN   BIT(SSC0KF)
        EXTRN   CODE(WATCH)
AN12DT1 DSEG    0       AT 0
R0A:    DS      1H              ;Aレジスタ
AN12DT  DSEG    0       AT 0D0H
PTSVA:  DS      4
;
;
INT4    CSEG    INBLOCK
        PUSH    BS
        PUSH    XA
        PUSH    HL
        SEL     MB15
        SKT     PORT0.0         ;Port0.0チェック
        BR      POWOFF
        CLR1    SCC.3           ;パワーオン処理
        MOV     XA,#56
ONWAIT:
        DECS    A               ;システム・クロックの発振が安定するまでウエイト
        BR      ONWAIT
        DECS    X
        BR      ONWAIT
        CLR1    SCC.0           ;システム・クロックの切り替え
        CALLF   !HRDSET         ;ハードウエア再起動サブルーチン
        BR      REINT4
POWOFF:
        SKT     SSC0KF          ;サブシステム・クロックの発振確認
        BR      $HSTPR0
        MOV     A,#0011B
        MOV     PCC,A
        SET1    SCC.0           ;システム・クロックの切り替え
HSTPR0:
        MOV     XA,#0           ;ハードウエア停止処理
        MOV     TM0,XA
        MOV     CSIM,XA
        MOV     A,#0101B
        SKF     SSC0KF
        MOV     WM,XA
        MOV     A,#8            ;INT4以外の割り込みを禁止
        MOV     0B8H,A
        SET1    MBE
        MOV     HL,#0BCH
        MOV     A,#0
DILOP:
        MOV     @HL,A
        INCS    L
        BR      DILOP
        CLR1    MBE
        EI      IEW
```

```
        ;〈サブルーチン　入出力ポート処理〉
          SKF       SSCOKF
          SET1      SCC.3           ;メイン・システム・クロックを停止
REINT4:
          POP       HL
          POP       XA
          POP       BS
          RETI

        ;〈サブルーチン　ハードウエア再起動〉
HRDSET    CSEG      SENT
          MOV       A,#INDBTM       ;ハードウエアを再起動
          MOV       BTM,A
          MOV       XA,#INTM0
          MOV       TM0,XA
          MOV       XA,#INCSIM
          MOV       CSIM,XA
          MOV       XA,#INWM
          MOV       WM,XA
;
;
;
;         PORT RECOVER PROCESS
;
;                                   ;割り込みを許可
;
          MOV       A,#INTD0
          MOV       0B8H,A
          MOV       A,#INTD1
          SKT       R0A.1
          DI        IEW
          MOV       A,#INTD2
          MOV       0BCH,A
          MOV       A,#INTD3
          MOV       0BDH,A
          MOV       A,#INTD4
          MOV       0BEH,A
          MOV       A,#INTD5
          MOV       0BFH,A
          RET

        ;〈サブルーチン　時計カウント・チェック〉
TIMCNT    CSEG      INBLOCK
          SKF       SSCOKF          ;サブシステム・クロック，チェック中は時計カウントを行わない
          CALL      !WATCH
          SKF       PORT0.0         ;パワーダウンのチェック
          RET
          HALT                      ;スタンバイ・モードの設定
          NOP
          BR        TIMCNT
```

第3章　ベーシック・インターバル・タイマの応用

3.1　基準時間発生

μPD750008は8ビットのベーシック・インターバル・タイマ（BT）を備えています。このベーシック・インターバル・タイマは4種類のインターバル時間を設定することができ，このインターバル時間ごとにベーシック・インターバル・タイマ割り込み要求フラグ（IRQBT）をセットします。

ベーシック・インターバル・タイマは，ベーシック・インターバル・タイマ・モード・レジスタ（BTM）によって制御されます。BTMのフォーマットを図3－1に示します。

図3－1　ベーシック・インターバル・タイマ・モード・レジスタのフォーマット

アドレス	3	2	1	0	略号
F85H	BTM3	BTM2	BTM1	BTM0	BTM

(fx = 6.00 MHz)

			入力クロック指定	割り込みインターバル時間（スタンバイ解除時のウエイト時間）
0	0	0	$f_x/2^{12}$（1.46 kHz）	$2^{20}/f_x$（175 ms）
0	1	1	$f_x/2^{9}$（11.7 kHz）	$2^{17}/f_x$（21.8 ms）
1	0	1	$f_x/2^{7}$（46.9 kHz）	$2^{15}/f_x$（5.46 ms）
1	1	1	$f_x/2^{5}$（188 kHz）	$2^{13}/f_x$（1.37 ms）
上記以外			設定禁止	－

(fx = 4.19 MHz)

			入力クロック指定	割り込みインターバル時間（スタンバイ解除時のウエイト時間）
0	0	0	$f_x/2^{12}$（1.02 kHz）	$2^{20}/f_x$（250 ms）
0	1	1	$f_x/2^{9}$（8.18 kHz）	$2^{17}/f_x$（31.3 ms）
1	0	1	$f_x/2^{7}$（32.768 kHz）	$2^{15}/f_x$（7.81 ms）
1	1	1	$f_x/2^{5}$（131 kHz）	$2^{13}/f_x$（1.95 ms）
上記以外			設定禁止	－

ベーシック・インターバル・タイマ／ウォッチドッグ・タイマのスタート制御ビット

“1”を書き込むことによって，ベーシック・インターバル・タイマ／ウォッチドッグ・タイマのスタート（カウンタ，割り込み要求フラグのクリア）が行われる。
動作を開始すると自動的に“0”にリセットされる。

次に4種類のインターバル時間を設定した例を示します（すべてfx = 4.19 MHz動作時）。

（1）インターバル時間を250 msに設定する（250 msごとにIRQBTをセットする）。

```
SEL   MB15

MOV   A,#1000B
MOV   BTM,A
```

（2）インターバル時間を31.3 msに設定する（31.3 msごとにIRQBTをセットする）。

```
SEL   MB15

MOV   A,#1011B
MOV   BTM,A
```

（3）インターバル時間を7.81 msに設定する（7.81 msごとにIRQBTをセットする）。

```
SEL   MB15

MOV   A,#1101B
MOV   BTM,A
```

（4）インターバル時間を1.95 msに設定する（1.95 msごとにIRQBTをセットする）。

```
SEL   MB15

MOV   A,#1111B
MOV   BTM,A
```

3.2　ウォッチドッグ・タイマの応用

ベーシック・インターバル・タイマ／ウォッチドッグ・タイマは，ウォッチドッグ・タイマ許可フラグ（WDTM）に“1”をセットすると，ベーシック・インターバル・タイマ（BT）のオーバフローにより内部リセット信号を発生するウォッチドッグ・タイマとして動作します（なお，WDTMは一度“1”にセットすると，リセット以外にクリアすることはできません）。BTは，クロック発生回路からのクロックによって常にインクリメントされ，カウント動作を停止することはできません（図3－2参照）。

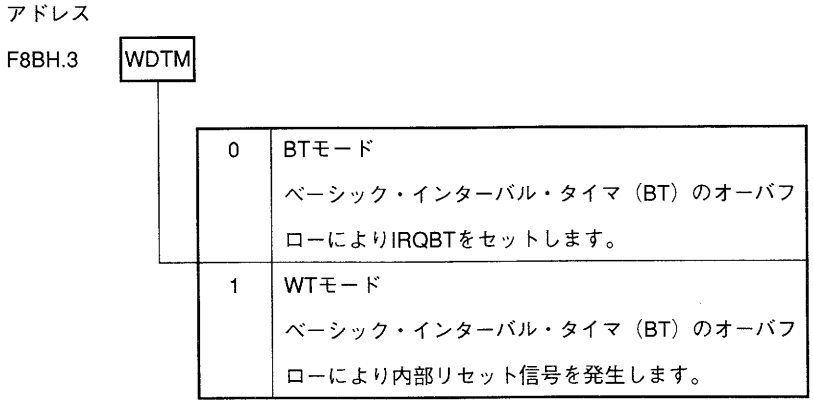
ウォッチドッグ・タイマ・モードでは，BTのオーバフローするインターバル時間を利用して，プログラムの暴走を検出します。このインターバル時間は，BTMのビット2-0の設定により4通りの時間が選択できます（図3－1参照）。これらの中からユーザ・システムに応じて暴走検出に必要な時間を決めてください。インターバル時間を設定しておいて，プログラムをその時間内に実行できる単位に分割し，それぞれの単位の最後でBTをクリアする命令を実行させるようにします。そうすると，設定時間内にこのBTクリアを実行する命令にたどりつかなければ（順調にプログラムの実行が進んでいなければ＝暴走）BTはオーバフローし，内部リセット信号が発生してプログラムを強制終了させてしまいます。この結果，内部リセットがかかったということはプログラムの暴走が起きたことを示し，その検出ができたことになります。

ウォッチドッグ・タイマの設定手順は，次のように行ってください（①，②の設定は同時に行ってもかまいません）。

① BTMにインターバル時間をセットする。
② BTMのビット3に“1”をセットする。
③ WDTMに“1”をセットする。
　　} 初期設定

④ ①～③を設定したあとは，インターバル時間以内にBTMのビット3に“1”をセットする。

図3－2　ウォッチドッグ・タイマ許可フラグのフォーマット



例　7.81 msのウォッチドッグ・タイマとして使用する（4.19 MHz動作時）。

プログラムをBTMの設定時間（7.81 ms）以内に処理が終了するいくつかのモジュールに分割し，各モジュールの終わりでBTをクリアする。暴走した場合，BTが設定時間内にクリアされないためオーバフローしてしまい，内部リセット信号が発生する。

初期設定：

```
SET1    MBE
SEL     MB15
MOV     A,#1101B
MOV     BTM,A          ;時間設定とスタート
SET1    WDTM           ;ウォッチドッグ・タイマを許可
  ⋮
```

（以後，7.81 msごとにBTMのビット3に“1”をセットする。）

モジュール1：

```
  ⋮
SET1    MBE
SEL     MB15
SET1    BTM.3
```

7.81 ms以内で処理完了

モジュール2：

```
  ⋮
SET1    MBE
SEL     MB15
SET1    BTM.3
```

7.81 ms以内で処理完了

```
  ⋮
```

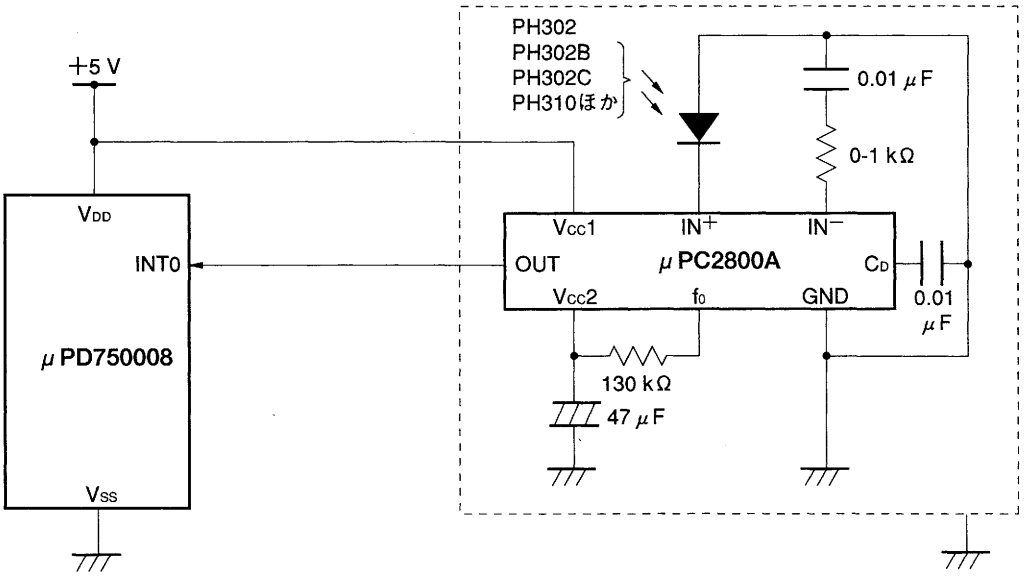
3.3　リモコン信号受信応用

ベーシック・インターバル・タイマを使用して，汎用赤外線リモート・コントローラ（以後，リモコンと略します）送信用デバイスμPD6122の送信データを受信するプログラムの例を紹介します。

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μPC2800Aを介して，P10/INT0端子から入力されます（図3－3参照）。

このプログラムは，そのリモコン信号のエッジ間の長さをINTBTでカウントして，コード化します。

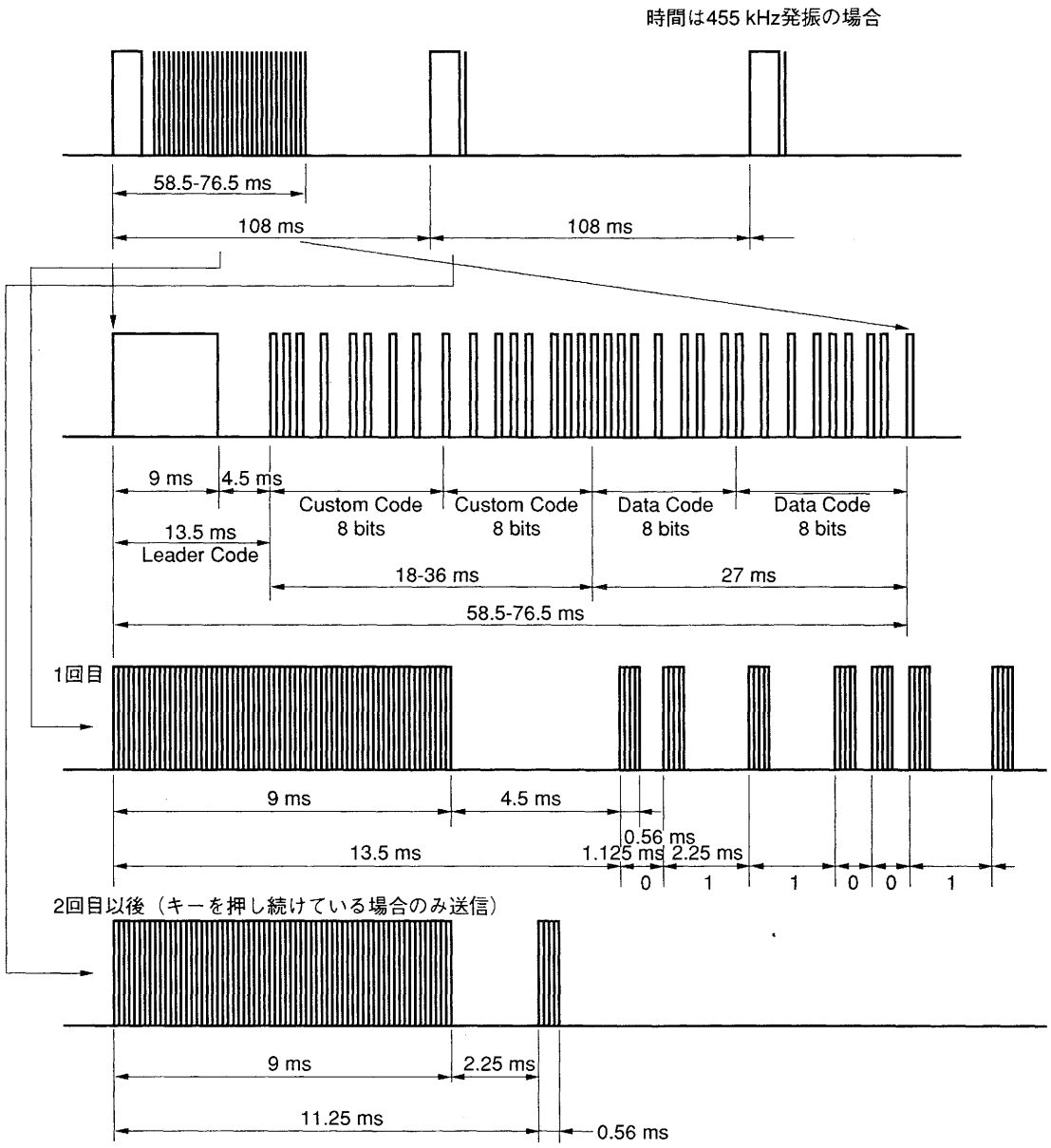
図3－3　リモコン信号受信回路例



リモコン信号は，リーダ・コード（Leader Code），カスタム・コード（Custom Code），データ・コード（Data Code），リピート・コード（Repeat Code）からなります。

ここで示されるリモコン信号は，NECのリモコン信号フォーマットに準拠しています。そのフォーマットを図3－4に示します。

図3－4　リモコン送信用IC出力信号

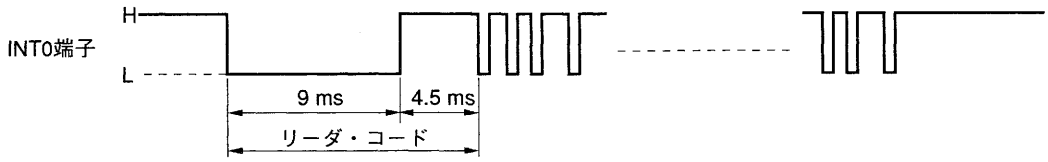


リモコン用受信プリアンプμPC2800Aの出力は，図3－5（a）のように，ロウ・アクティブです。

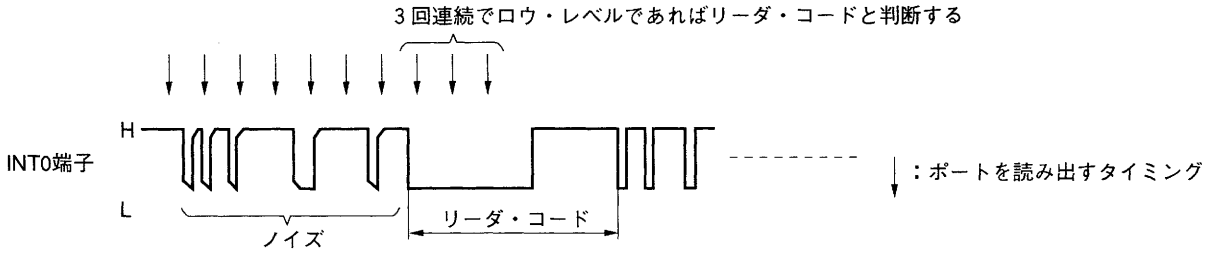
しかし，実際には外来光（蛍光灯など）のノイズがリーダ・コードの前に多数現れます。そこでこのプログラムでは，リーダ・コードの立ち下がりの確認は，ベーシック・インターバル・タイマの割り込み処理内で，ポートのレベルの読み出しにより行います（図3－5（b）参照）。

図3－5　受信プリアンプの出力波形

（a）理想の波形



（b）実際の波形



次の（1）プログラムの説明と（2）フロー・チャートでは，リモコン信号で正常と認識されたデータを“有効な”と表現しています。

（1）プログラムの説明

このプログラムは，**第11章　このアプリケーション・プログラムの応用例**で使用するものを，入出力インタフェース部を変更して説明しています。

〈使用するレジスタ〉

XA, B, HL

〈使用するRAM〉

RAMは，メモリ・バンク0の20H番地から配置できます。

WORK：	2ワード	；受信したデータを格納するワーク・エリア
RMDATA：	2ワード	；有効なデータ・コードを格納するエリア
LDCODE：	1ワード	；リーダ・コードのロウ・レベル時間をカウントする
MODEP：	1ワード	；どのエッジを検出中かを示すパラメータ
RPTIM：	1ワード	；有効なデータ入力から検出した時間をカウントする（200 msまで）
RPCODE：	1ワード	；有効なリピート・コードをカウントする
RMFLG：	1ワード	；フラグを格納する
REP_F：	1ビット	；1ならばRPTIMのカウントをスタートする

〈ネスティング〉

2レベル（16ワード）

〈使用するハードウエア〉

・ポート：Port1.0（INT0として使用）

・タイマ：ベーシック・インターバル・タイマ

〈使用する割り込み〉

INT0, INTBT

〈初期設定〉

・RAMの初期設定

・ハードウエアの初期設定

・INTBTの割り込み許可

〈起動方法〉

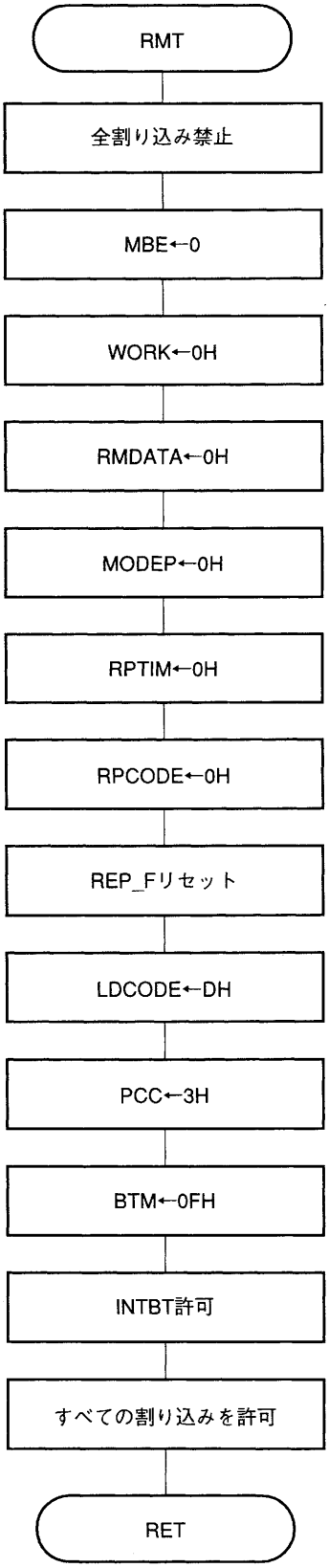
・初期設定処理（RMT：）の実行により，起動します。

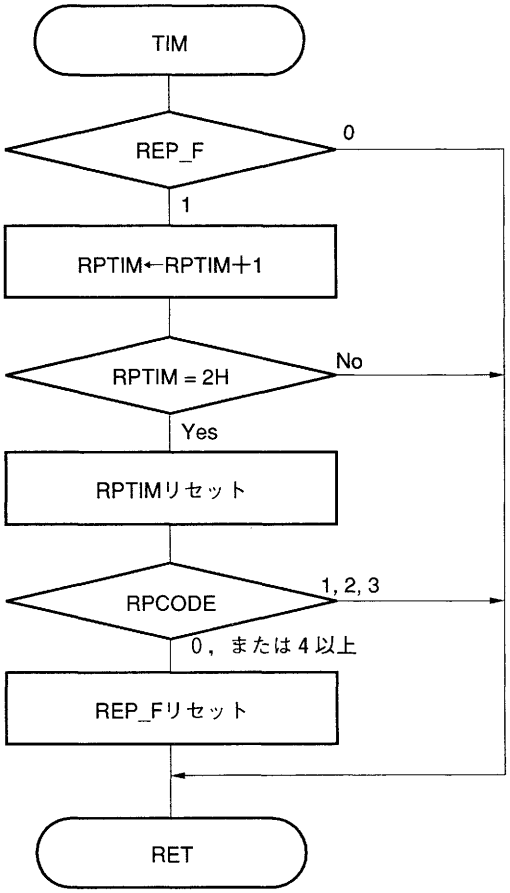
・有効なリモコン・コードが入力されると，REP_F＝1とし，RMDATAにコードが格納されます。そのあと200 msの間に1回もリピート・コードが入力されなかった場合（リモコンのキーが離されたと判定），また4回以上リピート・コードが入力された場合（2機以上のリモコン信号を受けたと判定）はREP_F＝0とします。

・リモコン・コードの受信を待機している間はMODEP＝0となります。

（2）フロー・チャート

初期設定



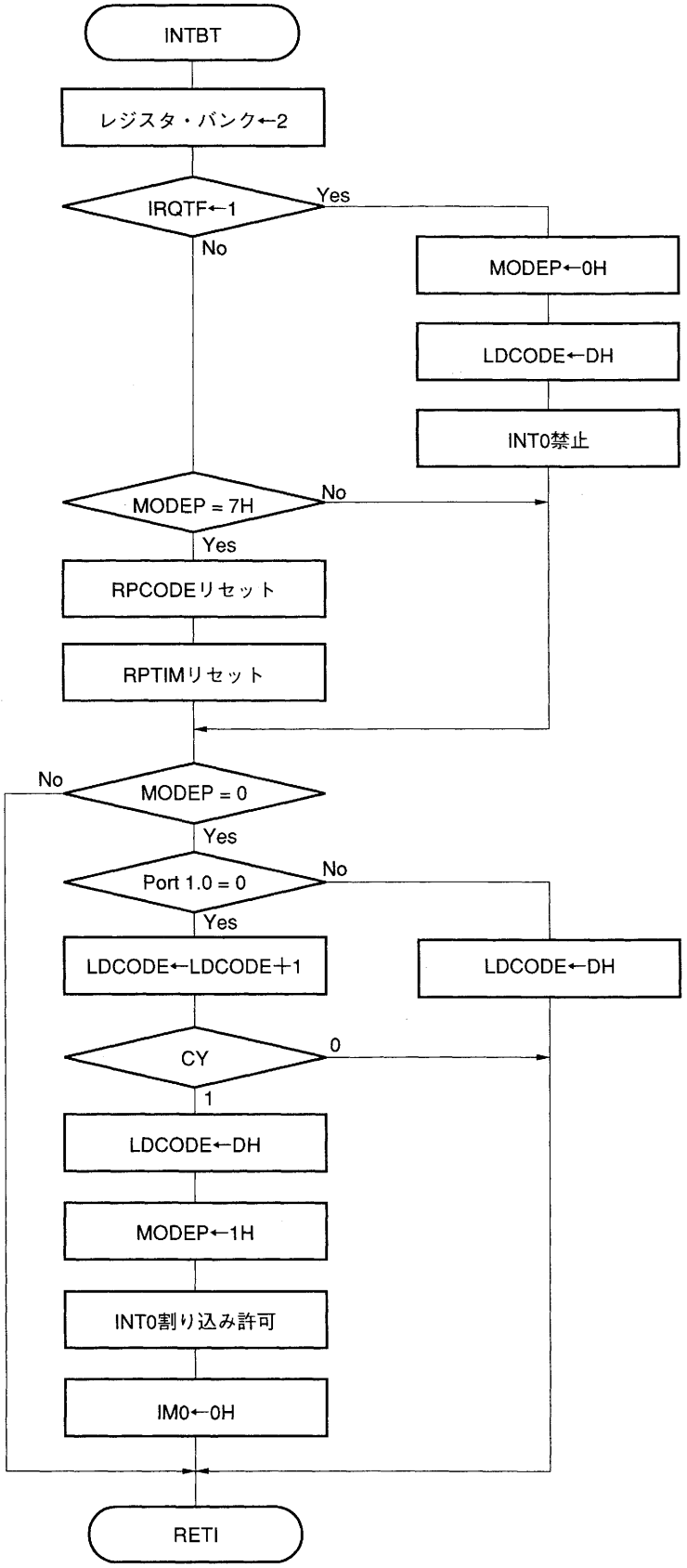


100 msごとに処理

200 ms経過したかをチェック

リピート・コードの回数をチェック

INTBT処理



2 msごとに処理

IRQTF = 1ならばモジュロ・レジスタのカウント終了

リピート・コード・カウンタをリセット

200 msのカウンタをリセット

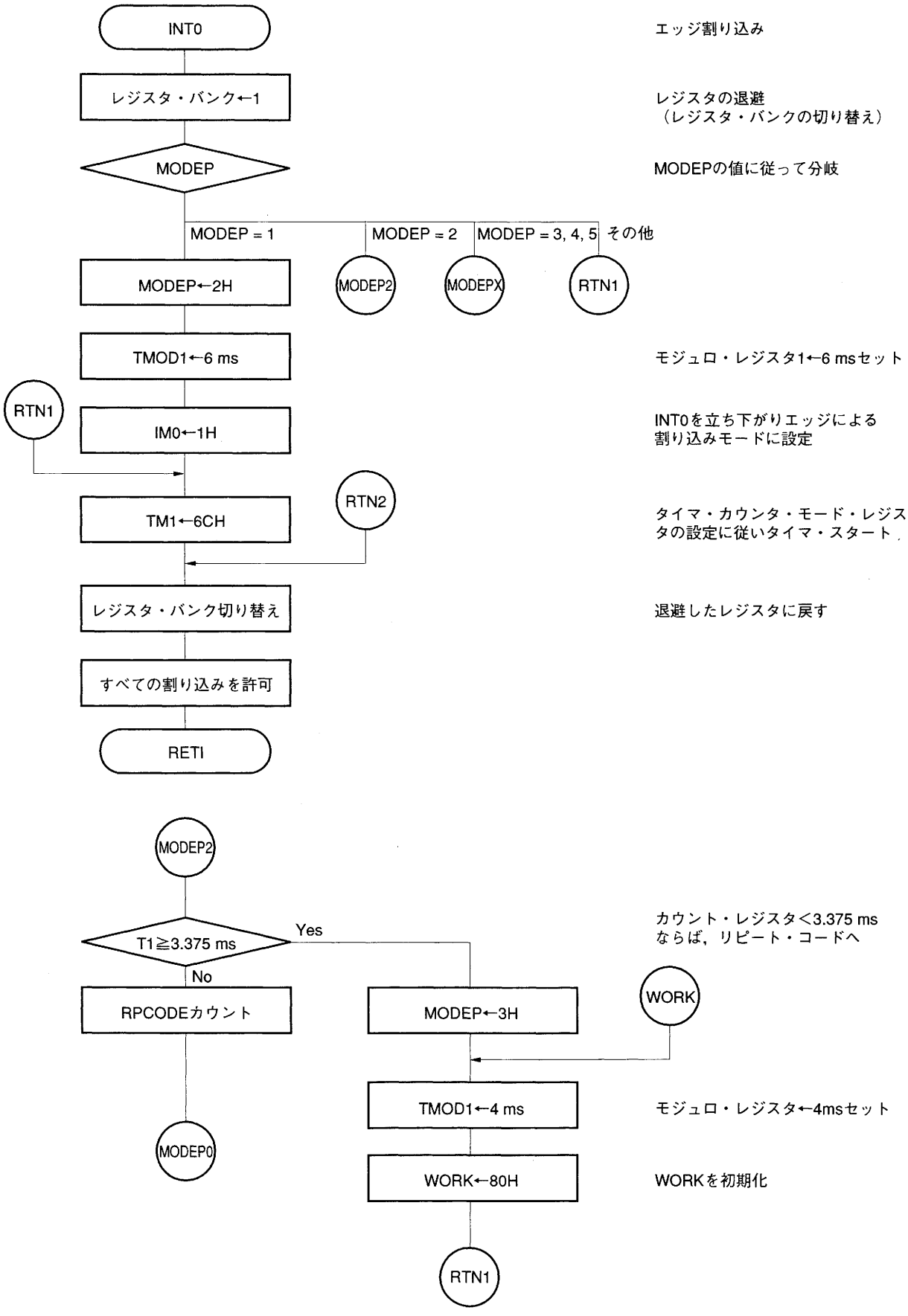
MODEP = 0のとき，リーダ・コードのロウ・レベル時間のチェック

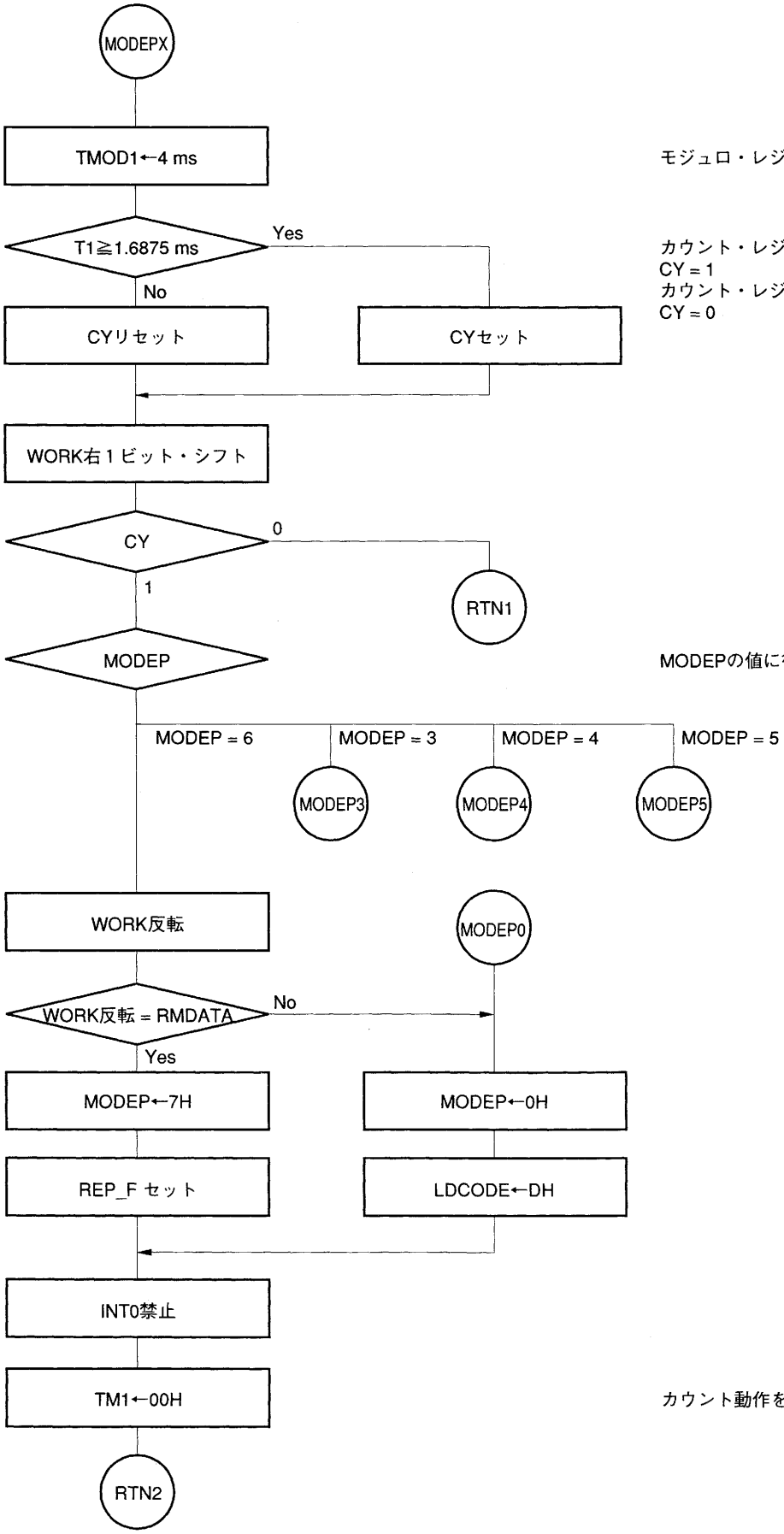
INT0がロウ・レベルのときLDCODEをカウント
ハイ・レベルのときはLDCODEを初期化

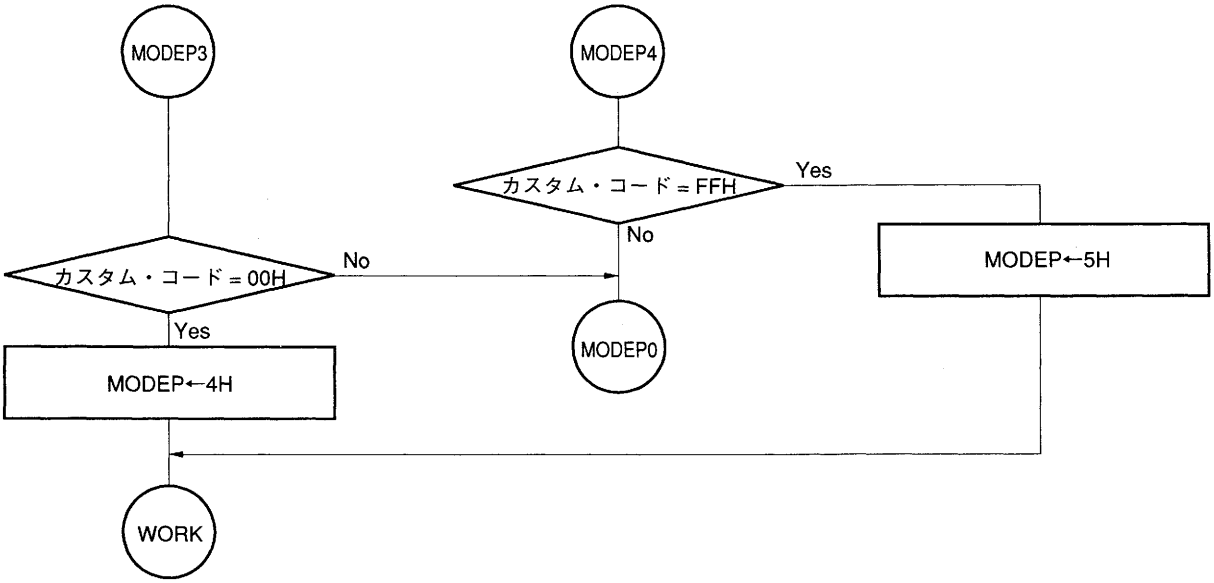
リーダ・コードのハイ・レベル時間のチェックへ

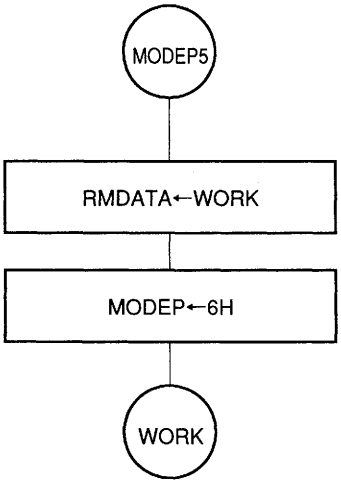
INT0を立ち上がりエッジの割り込みモードに設定

INT0処理


INT0
エッジ割り込み
レジスタ・バンク←1
レジスタの退避
（レジスタ・バンクの切り替え）
MODEP
MODEPの値に従って分岐
MODEP = 1
MODEP = 2
MODEP = 3, 4, 5
その他
MODEP←2H
MODEP2
MODEPX
RTN1
TMOD1←6 ms
モジュロ・レジスタ1←6 msセット
RTN1
IM0←1H
INT0を立ち下がりエッジによる
割り込みモードに設定
RTN2
TM1←6CH
タイマ・カウンタ・モード・レジス
タの設定に従いタイマ・スタート
レジスタ・バンク切り替え
退避したレジスタに戻す
すべての割り込みを許可
RETI
MODEP2
T1≧3.375 ms
Yes
No
カウント・レジスタ<3.375 ms
ならば，リピート・コードへ
RPCODEカウント
MODEP←3H
WORK
TMOD1←4 ms
モジュロ・レジスタ←4msセット
MODEP0
WORK←80H
WORKを初期化
RTN1


MODEPX
TMOD1←4 ms
モジュロ・レジスタ1←4 msセット
T1≧1.6875 ms
Yes
No
カウント・レジスタ≧1.6875 msならば，
CY = 1
カウント・レジスタ<1.6875 msならば，
CY = 0
CYリセット
CYセット
WORK右1ビット・シフト
CY
0
1
RTN1
MODEP
MODEPの値に従って分岐
MODEP = 6
MODEP = 3
MODEP = 4
MODEP = 5
MODEP3
MODEP4
MODEP5
WORK反転
MODEP0
WORK反転 = RMDATA
No
Yes
MODEP←7H
MODEP←0H
REP_F セット
LDCODE←DH
INT0禁止
TM1←00H
カウント動作を停止
RTN2


MODEP3
MODEP4
カスタム・コード＝FFH
Yes
No
カスタム・コード＝00H
No
MODEP←5H
Yes
MODEP←4H
MODEP0
WORK


MODEP5
RMDATA←WORK
MODEP←6H
WORK

（3）プログラム例

```
        VENT0   MBE=0,RBE=0,INIT        ;RESET
        VENT1   MBE=0,RBE=1,INTBT       ;INTBT/INT4
        VENT2   MBE=0,RBE=1,INT0        ;INT0

DSEG0   DSEG    0       AT      20H     ;メモリ・バンク0，アドレス20Hから格納

WORK:           DS      2               ;受信したデータを格納するワーク・エリア
RMDATA:         DS      2               ;有効なデータ・コードを格納
LDCODE:         DS      1               ;リーダ・コードのロウ・レベル時間をカウント
MODEP:          DS      1               ;どのエッジを検出中かを示すパラメータ
RPTIM:          DS      1               ;有効なデータ入力からの経過時間をカウント(200msまで)
RPCODE:         DS      1               ;有効なリピート・コードをカウント
RMFLG:          DS      1               ;フラグ

REP_F           EQU     RMFLG.0         ;1ならばRPTIMカウント・スタート

    ;〈サブルーチン　初期設定〉
INIT:
        DI                              ;すべての割り込みを禁止
        CLR1    MBE                     ;MBE←0

        MOV     XA,#00H                 ;RAMを設定
        MOV     WORK,XA
        MOV     RMDATA,XA
        MOV     MODEP,A
        MOV     RPTIM,A
        MOV     RPCODE,A
        MOV     RMFLG,A
        MOV     A,#0DH
        MOV     LDCODE,A                ;LDCODEを初期化

        MOV     A,#0011B
        MOV     PCC,A                   ;1マシン・サイクル←0.95μs，CPUは通常動作モードにする

        MOV     A,#1111B
        MOV     BTM,A                   ;インターバル時間←1.95ms

        EI      IEBT                    ;ベーシック・インターバル・タイマ(INTBT)の使用を許可
        EI                              ;すべての割り込みを許可
```

```
;〈サブルーチン　100 msタイマ処理〉

        SKT     REP_F
        RET
        INCS    RPTIM                   ;リモコン・リピート用タイマ＋1
        NOP
        MOV     A,RPTIM
        SKE     A,#2H                   ;リモコン・リピート用タイマ=200ms?
        RET
        MOV     A,#0H
        MOV     RPTIM,A                 ;RPTIMをクリア
        MOV     A,RPCODE
        ADDS    A,#0FH                  ;リモコン・リピート・コード・カウンタ=0?
        BR      TIM_52
        ADDS    A,#0FH                  ;リモコン・リピート・コード・カウンタ=1?
        RET
        ADDS    A,#0FH                  ;リモコン・リピート・コード・カウンタ=2?
        RET
        ADDS    A,#0FH                  ;リモコン・リピート・コード・カウンタ=3?
        RET

TIM_52:
        CLR1    REP_F                   ;REP_F←0
        RET
```

```
;〈サブルーチン　INTBT処理〉

        SEL     RB2             ;レジスタ・バンク←2
        SKTCLR  IRQT1           ;1ならば，モジュロ・レジスタ=カウント・レジスタ
        BR      MODEP7
        MOV     A,#0H
        MOV     MODEP,A         ;MODEP←0H
        MOV     A,#0DH
        MOV     LDCODE,A        ;LDCODEを初期化
        DI      IE0             ;INT0割り込みを禁止
        BR      RMCNT

MODEP7:
        MOV     A,MODEP
        SKE     A,#7H           ;MODEP=7?
        BR      RMCNT
        MOV     A,#0H
        MOV     MODEP,A         ;MODEP←0H
        MOV     RPCODE,A        ;リピート・コード・カウント・リセット
        MOV     RPTIM,A         ;200msカウント・リセット
RMCNT:
        MOV     A,MODEP
        SKE     A,#0H           ;MODEP=0?
        RETI
        SKF     PORT1.0
        BR      RMCNT_1
        INCS    LDCODE          ;リーダ・コード・スキャン・カウンタ＋1
        RETI
        MOV     A,#0DH          ;LDCODEを初期化
        MOV     LDCODE,A
        MOV     A,#1H
        MOV     MODEP,A         ;MODEP←1
        MOV     A,#0H
        MOV     IM0,A           ;立ち上がりエッジを指定
        EI      IE0             ;INT0割り込みを許可
        RETI

RMCNT_1:
        MOV     A,#0DH
        MOV     LDCODE,A        ;LDCODEを初期化
        RETI
```

```
;〈サブルーチン　INT0処理〉

        DI
        PUSH    BS
        SEL     RB1             ;レジスタ・バンク←1
        MOV     A,MODEP
        ADDS    A,#0FH          ;MODEP=0?
        BR      INT0_E
        ADDS    A,#0FH          ;MODEP=1?
        BR      INT0_P1
        ADDS    A,#0FH          ;MODEP=2?
        BR      INT0_P2
        ADDS    A,#09H          ;MODEP=3-6?
        BR      INT0_PX
        BR      INT0_E

INT0_P1:
        MOV     A,#2H
        MOV     MODEP,A         ;MODEP← 2
        MOV     XA,#62H
        MOV     TMOD1,XA        ;モジュロ・レジスタ←6ms
        MOV     A,#1H
        MOV     IM0,A           ;立ち下がりエッジ指定
        BR      INT0_E

INT0_P2:
        MOV     XA,T1
        ADDS    XA,#0C9H        ;カウント・レジスタ=3.375ms?
        BR      INT0_RP1
        MOV     A,#3H
        MOV     MODEP,A         ;MODEP← 3
        BR      INT0_W1

INT0_RP1:
        INCS    RPCODE          ;リモコン・リピート・コード・カウンタ＋1
        NOP
        BR      INT0_P0

INT0_PX:
        MOV     XA,T1
        ADDS    XA,#0E4H        ;カウント・レジスタ=1.6875ms?
        BR      INT0_CY0
        SET1    CY              ;CY← 1
        BR      INT0_L1

INT0_CY0:
        CLR1    CY              ;CY← 0
INT0_L1:
        MOV     XA,WORK
        MOV     B,A
        MOV     A,X
        RORC    A
        MOV     X,A
        MOV     A,B
        RORC    A
```

```
        MOV     WORK,XA             ;受信データ(WORK)を右1ビット・シフト
        SKT     CY
        BR      INT0_E
        MOV     A,MODEP
        ADDS    A,#0CH              ;MODEP=3?
        BR      INT0_P3
        ADDS    A,#0FH              ;MODEP=4?
        BR      INT0_P4
        ADDS    A,#0FH              ;MODEP=5?
        BR      INT0_P5
        MOV     XA,WORK             ;MODEP=6
        MOV     HL,#0FFH
        XOR     HL,XA
        MOV     XA,RMDATA
        SKE     XA,HL               ;有効なリモコン・データ=受信データを反転？
        BR      INT0_P0
        MOV     A,#7H
        MOV     MODEP,A             ;MODEP←7
        SET1    REP_F
        DI      IE0                 ;INT0割り込み禁止
        MOV     XA,#0H
        MOV     TM1,XA              ;タイマ・カウントを停止
        BR      INT0_E1

INT0_P3:
        MOV     XA,WORK
        MOV     HL,#00H
        SKE     XA,HL               ;カスタム・コード=00H?
        BR      INT0_P0
        MOV     A,#4H
        MOV     MODEP,A             ;MODEP←4
        BR      INT0_W1

INT0_P4:
        MOV     XA,WORK
        MOV     HL,#0FFH
        SKE     XA,HL               ;カスタム・コード=FFH?
        BR      INT0_P0
        MOV     A,#5H
        MOV     MODEP,A             ;MODEP←5
        BR      INT0_W1

INT0_P5:
        MOV     XA,WORK
        MOV     RMDATA,XA           ;有効なリモコン・データに受信データを入力
        MOV     A,#6H
        MOV     MODEP,A             ;MODEP←6
        BR      INT0_W1

INT0_P0:
        MOV     A,#0H
        MOV     MODEP,A             ;MODEP←0
        MOV     A,#0DH
        MOV     LDCODE,A            ;LDCODEを初期化
        DI      IE0                 ;INT0割り込みを禁止
        MOV     XA,#0H
```

```
        MOV     TM1,XA              ;タイマ・カウント停止
        BR      INT0_E1

INT0_W1:
        MOV     XA,#0FFH
        MOV     TMOD1,XA            ;モジュロ・レジスタ←4ms
        MOV     XA,#80H
        MOV     WORK,XA             ;WORKを初期化
INT0_E:
        MOV     A,MODEP
        ADDS    A,#0DH
        MOV     XA,#6CH
        MOV     XA,#7CH
        MOV     TM1,XA              ;タイマ・カウント・スタート
INT0_E1:
        POP     BS
        EI                          ;すべての割り込みを許可
        RETI
```

第4章　タイマ／イベント・カウンタの応用

4.1　インターバル・タイマの設定

タイマ設定時間は，タイマ・モード・レジスタの設定により選択されたカウント・パルス周波数と，モジュロ・レジスタの値によって決定します。

タイマ設定時間T（sec）は次の式で求められます。

$$T = \frac{N+1}{f_{CP}} = (N+1) \cdot (\text{分解能}) \quad \cdots\cdots ①$$

f_{CP}（Hz）：カウント・パルス周波数

N　　：モジュロ・レジスタの値（$N \neq 0$）

表4－1に，タイマ／イベント・カウンタの各カウント・パルスでの分解能と設定時間の範囲を示します。

（1）カウント・パルス周波数の決め方

設定しようとするタイマ設定時間が表4－1の設定時間内にあり，かつ最も分解能が高いモードを選択します。

（2）モジュロ・レジスタの値の決め方

①式より

$$N = \frac{T}{(\text{分解能})} - 1 \quad \cdots\cdots ②$$

となり，（分解能）に（1）で決まったモードの分解能を代入してモジュロ・レジスタに設定する値Nを決定します。

表4－1　分解能と最長設定時間（fx＝4.194304 MHz）

モード・レジスタ			分解能	最長時間
TM06	TM05	TM04		
1	0	0	244 μs	62.5 ms
1	0	1	61 μs	15.6 ms
1	1	0	15.3 μs	3.9 ms
1	1	1	3.8 μs	977 μs

次にタイマ・モード・レジスタとモジュロ・レジスタによってインターバル時間を設定した例を示します。例1，2では，インターバル時間ごとにタイマ／イベント・カウンタ割り込み要求フラグ（IRQT0）をセットします。

例1．10 msのインターバル時間を設定する（fx＝4.194304 MHz）。

最長設定時間が10 ms以上で，分解能が最も小さいモードを使用します。これは表4－1より最長設定時間が15.6 msのモードとなります。

モジュロ・レジスタに設定する値は，②式により次のようになります。

$$N = \frac{T}{(\text{分解能})} - 1$$

$$= \frac{10 \times 10^{-3}}{61 \times 10^{-6}} - 1$$

$$= 162.9 = 0A3H$$

```
TIMER0:
        SEL     MB15
        MOV     XA,#0A3H
        MOV     TM0D0,XA            ;モジュロ・レジスタを設定
        MOV     XA,#01011100B
        MOV     TM0,XA              ;タイマ・モード・レジスタを設定
```

例2．120 μsのインターバル時間を設定する（fx = 4.194304 MHz）。

最長設定時間が120 μs以上で，分解能が最も小さいモードを使用します。これは表4－1より最長設定時間が977 μsのモードとなります。

モジュロ・レジスタに設定する値は，②式より次のようになります。

$$N = \frac{T}{(\text{分解能})} - 1$$

$$= \frac{120\times10^{-6}}{3.8\times10^{-6}} - 1$$

$$= 30.5 \fallingdotseq 1FH$$

```
TIMER1：
        SEL     MB15
        MOV     XA,#1FH
        MOV     TMOD0,XA            ;モジュロ・レジスタを設定
        MOV     XA,#01111100B
        MOV     TM0,XA              ;タイマ・モード・レジスタを設定
```

4.2　PTO端子への出力例

μPD750008は，PTO0またはPTO1端子にタイマ／イベント・カウンタまたはタイマ・カウンタのTOUT F/Fの内容を出力することができます。

TOUT　F/Fは，モジュロ・レジスタとカウント・レジスタの値が一致するたびに反転するので，PTO0またはPTO1端子へ方形波を出力することができます。このときの出力周波数は，カウント・パルス周波数をモジュロ・レジスタにしたがって分周した周波数となります。

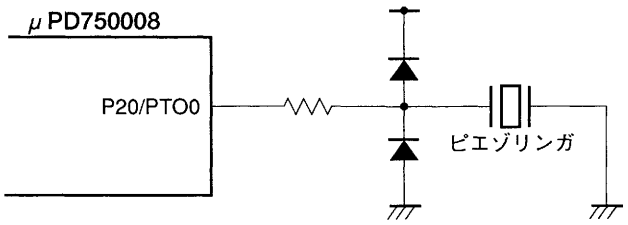
TOF0の内容をPTO端子に出力するために必要な操作を次に示します。

・TO許可フラグをセットする（TOE0←1）
・Port2.0の出力ラッチをリセットする（Port2.0←0）
・ポート2を出力ポート・モードにする（PM2←1）
・ポート2の内蔵プルアップ抵抗の接続を指定しない（POGA.2←0）

4.2.1　メロディ出力

タイマ／イベント・カウンタの出力であるPTO0端子を使用して，外部へメロディを出力するプログラム例を示します。

図4－1　PTO端子の使用例



P20／PTO0端子から出力する周波数は，タイマ／イベント・カウンタ・モード・レジスタによって，決定されるカウント・パルス周波数（CP）とモジュロ・レジスタに設定される値によって決定されます。

カウント・パルスとして$f_X/2^4$（f_{CP} = 262 kHz：f_X = 4.194304 MHz動作時）を選択したとき，各音階に対してモジュロ・レジスタに設定すべき値およびそれによって得られる周波数の各音階周波数に対する誤差を表4－2に示します。

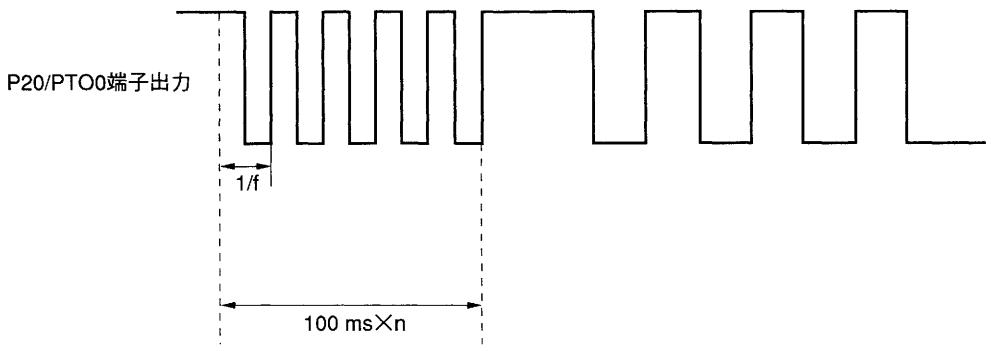
表4－2　各音階に対してモジュロ・レジスタに設定すべき値，およびそれによって得られる周波数の各音階周波数に対する誤差

平均律音階			PTO0端子出力		誤差（%）
音階		周波数（Hz）	TMOD1	周波数（Hz）	
ド	C5	523.25	F9H	523.75	+0.10
ド	C5＃	554.37	EBH	554.82	+0.08
レ	D5	587.33	DEH	587.16	−0.03
レ	D5＃	622.25	D1H	623.51	+0.20
ミ	E5	659.26	C6H	657.98	−0.19
ファ	F5	698.46	BAH	696.48	−0.28
ファ	F5＃	739.99	B0H	739.76	−0.03
ソ	G5	783.99	A6H	784.06	+0.01
ソ	G5＃	830.61	9CH	828.72	−0.23
ラ	A6	880.00	94H	878.78	−0.14
ラ	A6＃	932.33	8BH	928.63	−0.40
シ	B6	987.77	83H	984.49	−0.33
ド	C6	1046.50	7CH	1047.50	+0.10
ド	C6＃	1108.73	75H	1109.64	+0.08

メロディは，表4－2に示す各音階の周波数と時間長を変化させて出力します。時間長は，ベーシック・インターバル・タイマを使って作った100 msを基本時間として，その間に何回カウント動作を行うかで変化させます。

図4－2は，P20/PTO0端子からの出力信号の概念図です。

図4－2　P20／PTO0端子からの出力信号の概念図



このプログラム例では，図4－3　データ・フォーマットに従ったデータ・テーブルが，演奏データ・アドレス（MUS）から始まるRAMに格納されており，このデータ・テーブルに従ってメロディを出力します。このデータ・テーブルは，音階を決めるデータと音長を決めるデータの2つで構成されており，この組み合わせを変えることによって，さまざまなメロディを出力することができます。

図4－3　データ・フォーマット

データ	内　容
0XH-FXH	音階データ（表4－4参照）
X0H-XEH	音階データ（表4－5参照）
nFH	n×100 msの無音を出力後， 演奏をはじめから繰り返す

演奏用メモリ・エリアからデータを取り出すことにより，音階を出力できます。

音階出力

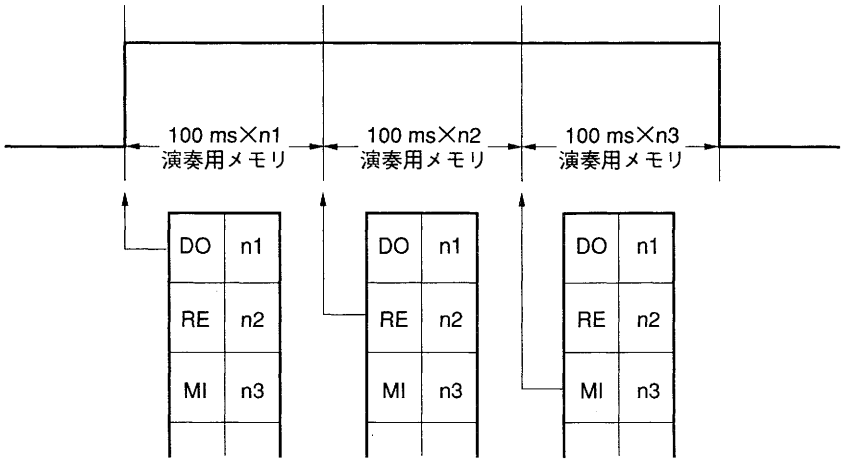


表4－3に音階データと音階，表示，およびPTO0端子出力の関係を，表4－4に音長データと音階出力の関係を示します。

表4－3　音階データと音階，表示，および PTO0端子出力の関係

音階	平均律音階		PTO0端子出力	
データ	音階	周波数(Hz)	TMOD1	周波数(Hz)
00H	ド　C5	523.25	F9H	523.75
01H	ド　C5＃	554.37	EBH	554.82
02H	レ　D5	587.33	DEH	587.16
03H	レ　D5＃	622.25	D1H	623.51
04H	ミ　E5	659.26	C6H	657.98
05H	ファ　F5	698.46	BAH	696.48
06H	ファ　F5＃	739.99	B0H	739.76
07H	ソ　G5	783.99	A6H	784.06
08H	ソ　G5＃	830.61	9CH	828.72
09H	ラ　A6	880.00	94H	878.78
0AH	ラ　A6＃	932.33	8BH	928.63
0BH	シ　B6	987.77	83H	984.49
0CH	ド　C6	1046.50	7CH	1047.50
0DH	ド　C6＃	1108.73	75H	1109.64
0EH	無音	0.0	FFH	0.0
0FH	無音	0.0	FFH	0.0

表4－4　音長データと音階出力の関係

音長	音階出力時間
データ	T＝100ms
00H	T×1
01H	T×2
02H	T×3
03H	T×4
04H	T×5
05H	T×6
06H	T×7
07H	T×8
08H	T×9
09H	T×10
0AH	T×11
0BH	T×12
0CH	T×13
0DH	T×14
0EH	T×15
0FH	リピート

（1）プログラムの説明

このプログラムは，**第11章　このアプリケーション・プログラムの応用**で使用するものを，入出力インタフェース部を変更して説明しています。

〈使用するレジスタ〉

XA，BC，HL

〈使用するRAM〉

RAMは，メモリ・バンク0の20H番地から配置できます。演奏用メロディ・データはメモリ・バンク1の00H番地からFFH番地に割り当てられます。

```
MUS：        2ワード        ；演奏データ・アドレス
TONE：       1ワード        ；音階データ
TONETIM：    1ワード        ；音長データ
TONEFLG：    1ワード        ；フラグ・エリア
TONE_F：     1ビットEQU     ；1ならば音階出力を許可
TMO_F：      1ビットEQU     ；1ならばタイマ・カウントをスタート
```

〈ネスティング〉

2レベル（12ワード）

〈使用するハードウエア〉

・ポート：Port2.0（PTO0出力兼用端子）

・タイマ：ベーシック・インターバル・タイマ，タイマ／イベント・カウンタ

〈初期設定〉

・演奏アドレスのクリア

・PTO0の出力許可

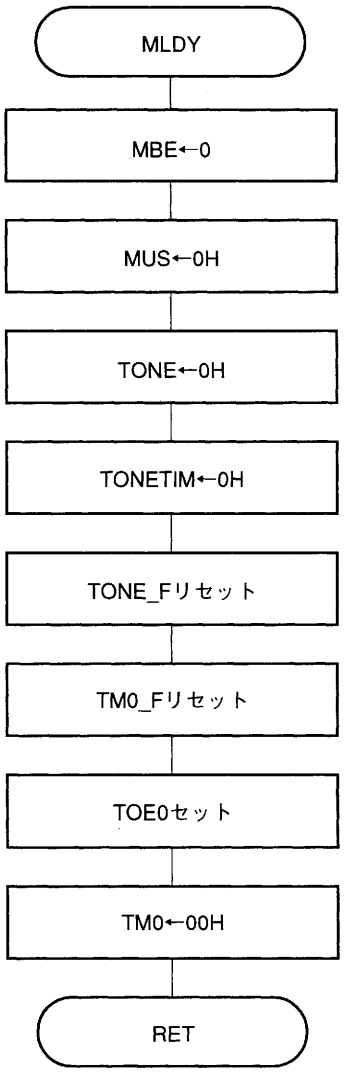
・タイマ／イベント・カウンタの停止

〈起動方法〉

TIM処理を100 msごとに実行することと，M_OUT処理を常に実行することを条件に，MLDYを初期化し，TONE_F＝1にするとメロディ演奏を開始します。

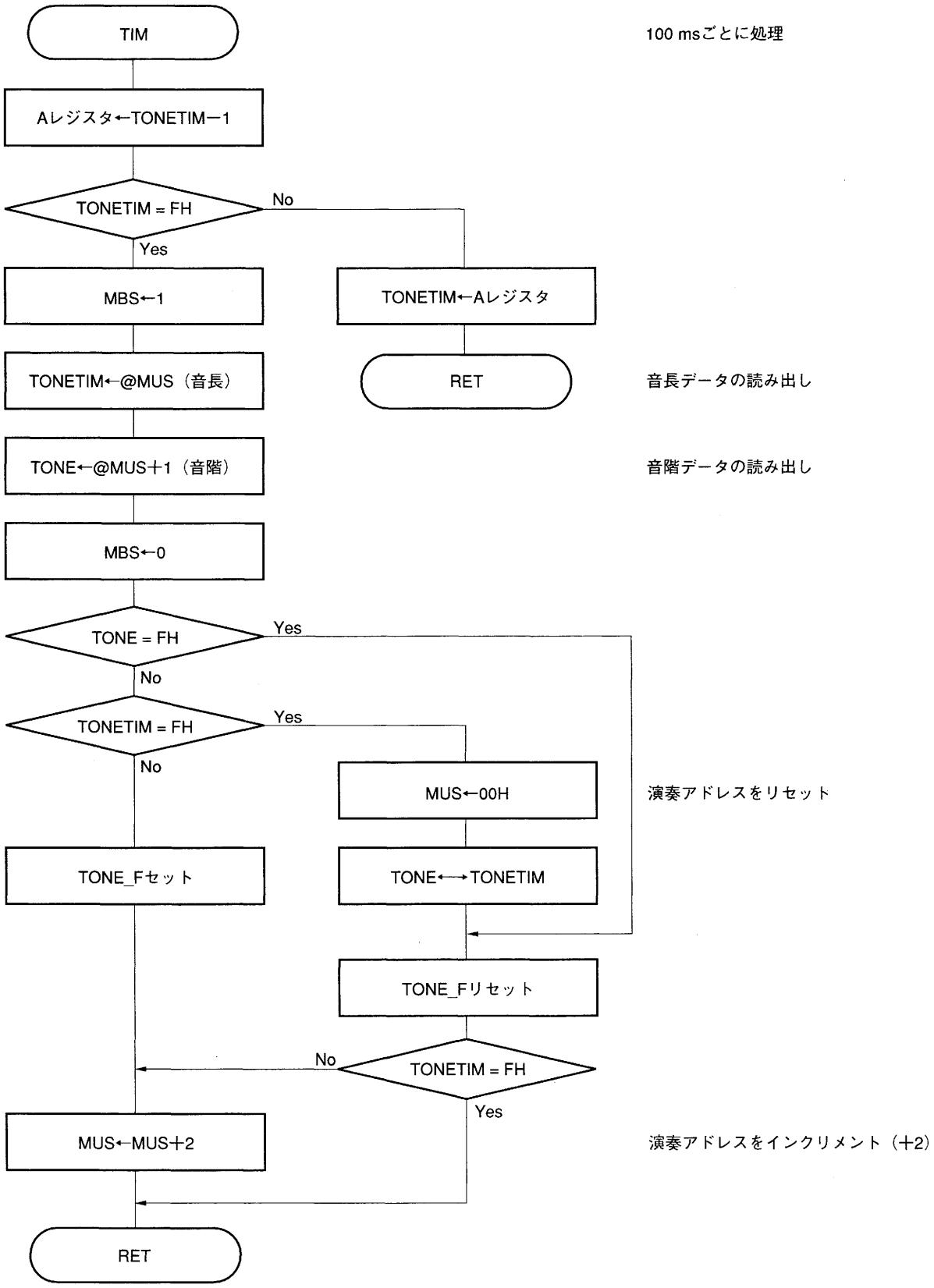
（2）フロー・チャート

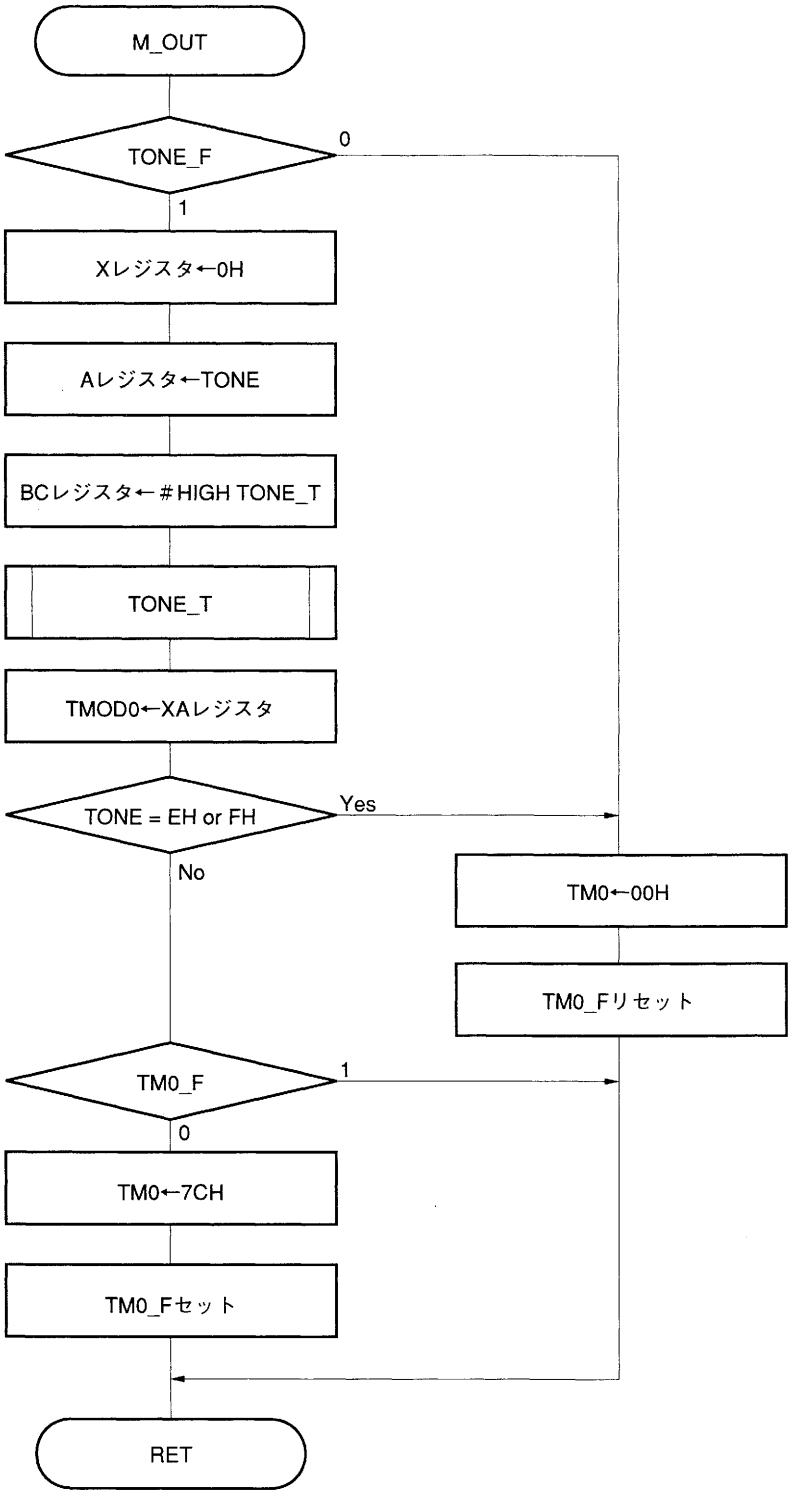
〈初期設定〉



出力許可フラグをセット

カウント動作を停止


TIM
100 msごとに処理
Aレジスタ←TONETIM−1
TONETIM = FH
No
Yes
MBS←1
TONETIM←Aレジスタ
TONETIM←@MUS（音長）
RET
音長データの読み出し
TONE←@MUS＋1（音階）
音階データの読み出し
MBS←0
TONE = FH
Yes
No
TONETIM = FH
Yes
No
MUS←00H
演奏アドレスをリセット
TONE_Fセット
TONE←→TONETIM
TONE_Fリセット
No
TONETIM = FH
Yes
MUS←MUS＋2
演奏アドレスをインクリメント（＋2）
RET



1ならば音階出力を許可

モジュロ・レジスタ0に音階データをセット

音階データで音階出力か，または無音出力かを判断

カウント動作スタート

（3）プログラム例

```
        VENT0   MBE=0,RBE=0,INIT        ;RESET

DSEG0   DSEG    0       AT      20H     ;メモリ・バンク0，アドレス20Hから格納

MUS:            DS      2               ;演奏データ・アドレス
TONE:           DS      1               ;音階データ
TONETIM:        DS      1               ;音長データ
TONEFLG:        DS      1               ;フラグ・エリア

TONE_F          EQU     TONEFLG.0       ;1ならば，音階出力を許可
TMO_F           EQU     TONEFLG.1       ;1ならばタイマ・カウント・スタート
```

；〈サブルーチン　初期設定〉

```
        CLR1    MBE                     ;MBE←0

        MOV     XA,#00H                 ;RAMの設定
        MOV     MUS,XA
        MOV     TONE,A
        MOV     TONETIM,A
        MOV     TONEFLG,A
        SET1    TOE0                    ;出力許可フラグをセット
        MOV     XA,#00H
        MOV     TM0,XA                  ;タイマ／イベント・カウンタを発振停止
```

```
;〈サブルーチン　タイマ処理〉

        MOV     A,TONETIM
        DECS    A
        BR      TIM_116
        MOV     XA,MUS

        SET1    MBE                     ;MBE←1
        SEL     MB1                     ;メモリ・バンク←1
        MOV     HL,XA
        MOV     A,@HL+
        NOP
        MOV     B,A
        MOV     A,@HL
        MOV     C,A
        SEL     MB0                     ;メモリ・バンク←0
        CLR1    MBE                     ;MBE←0

        MOV     A,B
        MOV     TONETIM,A               ;音長データ←@演奏データ・アドレス
        MOV     A,C
        MOV     TONE,A                  ;音階データ←@演奏データ・アドレス＋1
        MOV     A,#0H
        MOV     T_TIMC,A
        SKE     C,#0FH
        BR      TIM_115
        BR      TIM_114

TIM_115:
        SKE     B,#0FH
        BR      TIM_111
        BR      TIM_112

TIM_112:
        MOV     XA,#0H
        MOV     MUS,XA
        MOV     A,TONE
        MOV     TONETIM,A               ;TONETIM←リピート時間
        MOV     A,#0FH
        MOV     TONE,A                  ;TONE←FH
TIM_114:
        CLR1    TONE_F
        SKE     B,#0FH
        BR      TIM_113
        RET

TIM_111:
        SET1    TONE_F
TIM_113:
        MOV     XA,MUS
        ADDS    XA,#2H
        NOP
        MOV     MUS,XA                  ;演奏データ・アドレス＋2
        RET
```

```
TIM_116:
        MOV     TONETIM,A               ;音長データー1
        RET

    ;〈サブルーチン　メロディ出力処理〉

TONEOUT:
        SKT     TONE_F                  ;1ならば音階出力を許可
        BR      TONE0
        MOV     X,#0H
        MOV     A,TONE
        MOV     BC,#HIGH TONE_T
        CALL    !TABLE
        MOV     TMOD0,XA                ;モジュロ・レジスタ←音階データ
        MOV     A,TONE
        ADDS    A,#2H
        BR      TONE1
TONE0:
        MOV     XA,#0H                  ;タイマ・カウント停止
        CLR1    TM0_F                   ;0ならば，タイマ・カウント・スタート
        BR      TONE2
TONE1:
        SKF     TM0_F
        RET
        MOV     XA,#7CH                 ;タイマ・カウント・スタート
        SET1    TM0_F                   ;1ならば，タイマ・カウント停止
TONE2:
        MOV     TM0,XA
        RET
------------------------------------------------------------------------------
TABLE   CSEG    PAGE

        DB      0F9H        ;DO
        DB      0EBH        ;DO#
        DB      0DEH        ;RE
        DB      0D1H        ;RE#
        DB      0C6H        ;MI
        DB      0BAH        ;FA
        DB      0B0H        ;FA#
        DB      0A6H        ;SO
        DB      09CH        ;SO#
        DB      094H        ;RA+
        DB      08BH        ;RA#+
        DB      083H        ;SI+
        DB      07CH        ;DO+
        DB      075H        ;DO#+
        DB      0FFH        ;無音
        DB      0FFH        ;無音

TABLE:
        MOVT    XA,@BCXA
        RET
```

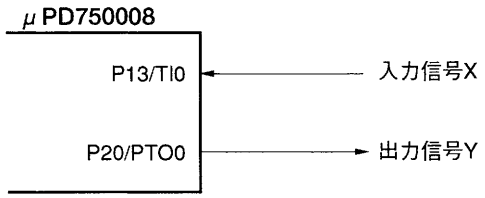
4.2.2 イベント・パルス分周出力

タイマ／イベント・カウンタのカウント・パルスにTI0端子からの入力パルスを選択し，N分周してP20/PTO0端子へ出力します。ただし，モジュロ・レジスタに設定する値をMとすると

N＝2（M＋1）

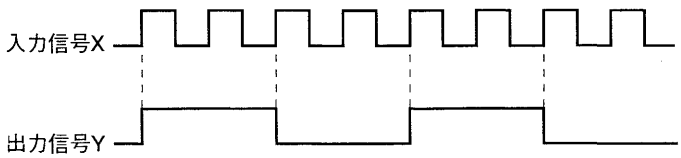
となります。

図4－4　イベント・パルス分周出力

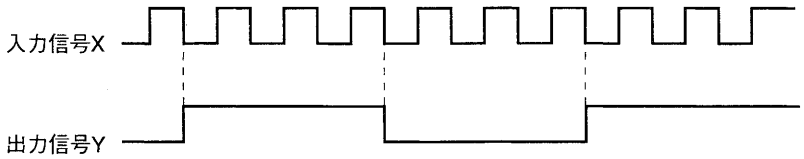


入力信号Xと出力信号Yとの関係は次のようになります。

（a）4分周の場合（M＝1，TM0←0CHのとき）



（b）6分周の場合（M＝2，TM0←1CHのとき）



次に，TI0端子からの入力パルスを立ち上がりエッジによりカウントし，6分周したパルスをP20/PTO0端子へ出力するプログラムを示します。

このプログラムは，**第11章　このアプリケーション・プログラムの応用例**には使用されていません。

（1）プログラムの説明

〈外部参照宣言シンボル〉

M

〈使用するレジスタ〉

XA

〈ネスティング〉

1レベル（4ワード）

〈使用するハードウエア〉

・ポート：Port2.0（PTO0出力兼用端子）

・タイマ：タイマ／イベント・カウンタ

〈初期設定〉

ポート・モード：ポート2←出力モード

〈起動方法〉

タイマ／イベント・カウンタをイベント・カウンタ動作モードに設定し，TOUTの出力を許可し，Port2.0←0とします。

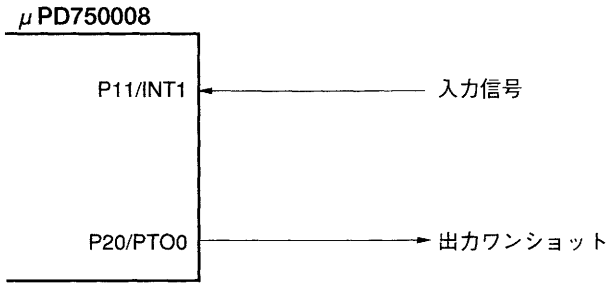
（2）プログラム例

```
PULSE   CSEG    INBLOCK

        CLR1    MBE
        MOV     XA,#M
        MOV     TMOD0,XA            ;モジュロ・レジスタ設定
        MOV     XA,#00001100B
        MOV     TM0,XA              ;タイマ・モード・レジスタ設定
        SET1    TOE0                ;TOUT出力を許可
        CLR1    PORT2.0             ;Port2.0←0
        RET
```

4.3　ワンショット・パルス出力

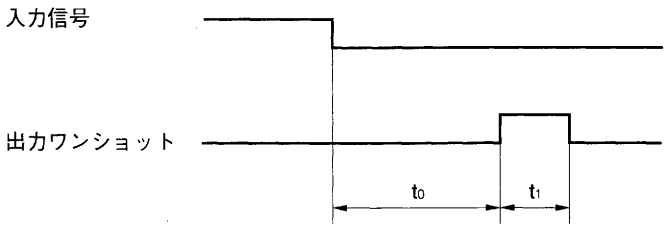
タイマ／イベント・カウンタのPTO端子をワンショット出力に応用します。出力のタイミングはP11/INT1端子のエッジ検出によって行います。

図4－5　ワンショット・パルス出力



このプログラムでは，INT1入力立ち下がりエッジによりPTO0端子へロウ・レベルを出力し，（1/16.4×10^3）×（N_0+1）秒後にハイ・レベルを出力します。出力をハイ・レベルに切り替えてから，（1/16.4×10^3）×（N_1+1）秒後に再びロウ・レベルを出力します（N_0 = 1-255，N_1 = 1-255）。

図4－6　ワンショット・パルス出力のタイミング（1）

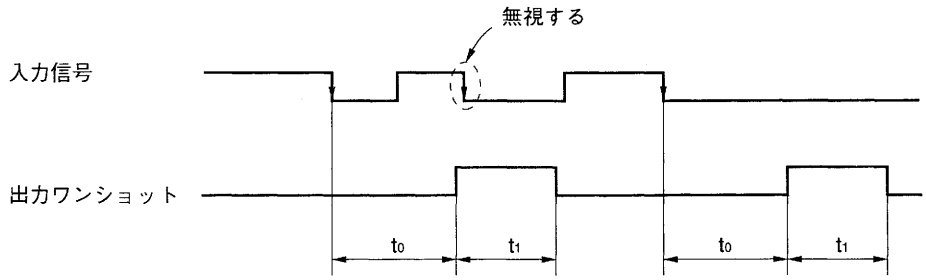


$$t_0 = \frac{1}{16.4} \times (N_0+1) \quad (ms)$$

$$t_1 = \frac{1}{16.4} \times (N_1+1) \quad (ms)$$

ただし，図4－7のように入力信号の立ち下がりエッジ検出からワンショット出力を終了するまでに，再び立ち下がりエッジを入力しても，このエッジは無視されます。

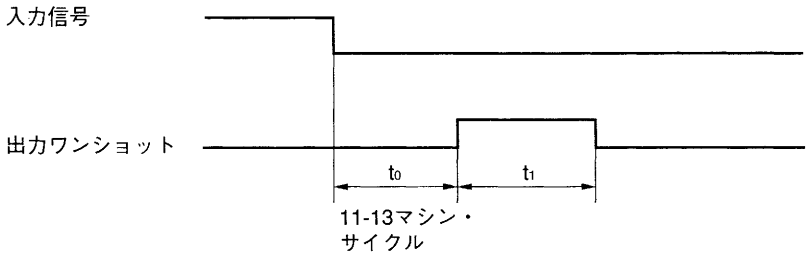
図4－7　ワンショット・パルス出力のタイミング（2）



（1）出力時間の性能

入力信号の立ち下がりエッジ検出から，実際にワンショットを出力するまでの時間は，モジュロ・レジスタに設定した値によって決まる時間よりも11マシン・サイクルから13マシン・サイクルの遅れがあります（図4－8参照）。また分解能は，$1/16.4\times10^{3}$ s $= 6.1\times10^{-5}$ sとなります。

図4－8　ワンショット・パルス出力のタイミング（3）



（2）プログラムの説明

このプログラムは，第11章　このアプリケーション・プログラムの応用では使用されていません。

〈外部参照宣言〉

N0，N1

〈使用するレジスタ〉

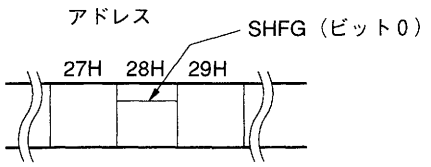
XA

〈使用するRAM〉

フラグ用に1ビットのRAMを使用し，メモリ・バンク0の08H-7FH番地に配置することができます。

SHFG：　1ビット　；ディレイ時間が終了し，PTO0端子にハイ・レベルが出力されるとセットされる。

図4－9　プログラム上のRAMのレイアウト



〈使用するハードウエア〉

・ポート：Port2.0（PTO0出力兼用端子），Port1.1（INT1兼用端子）

・タイマ：タイマ／イベント・カウンタ

〈割り込み〉

INT1，INTT0

〈初期設定〉

・RAM：SHFG←0

・ポート出力：Port2.0←0

・ポート・モード：ポート2←出力モード

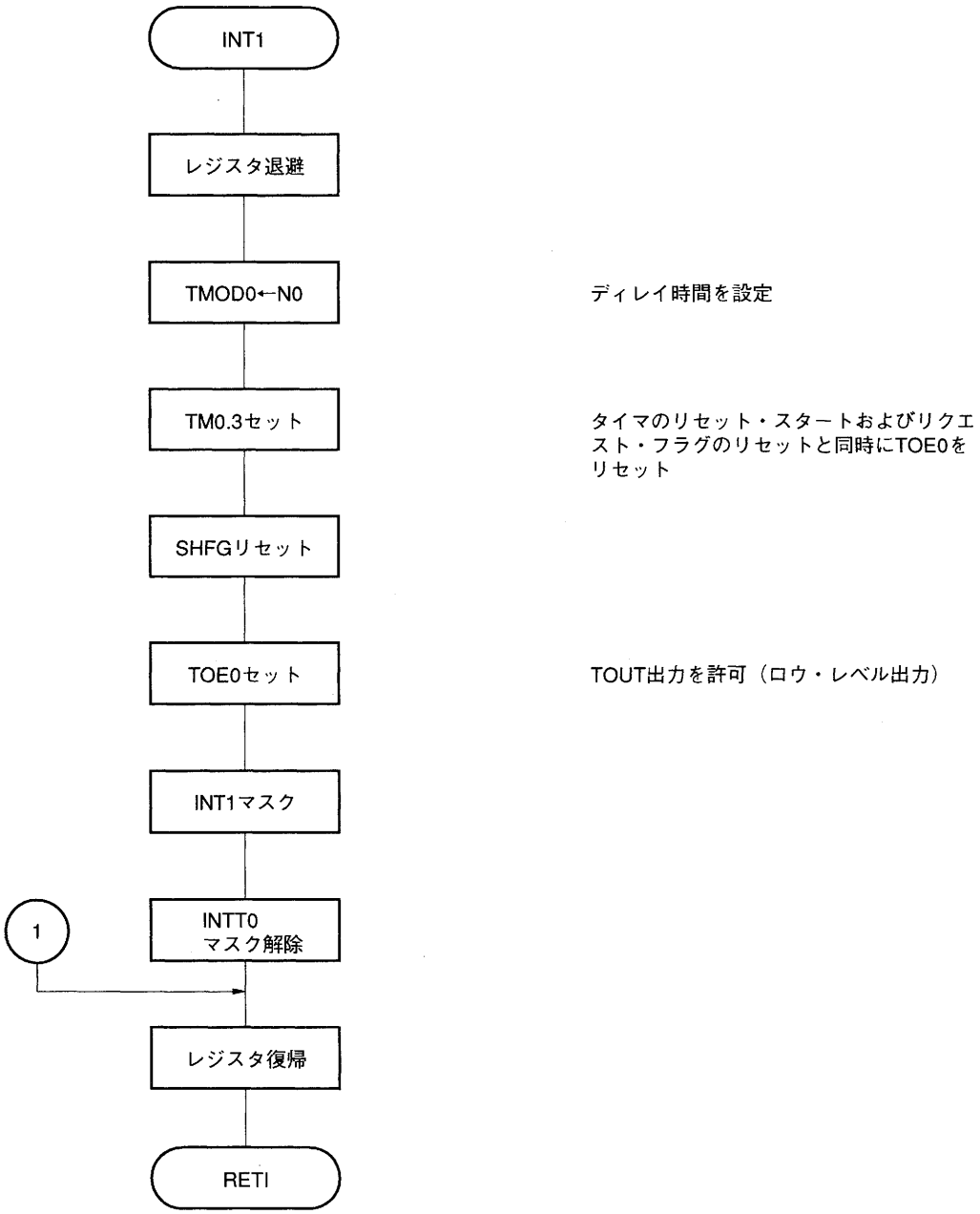
・タイマ：TM0←5CH

・割り込み：INT1割り込みを許可

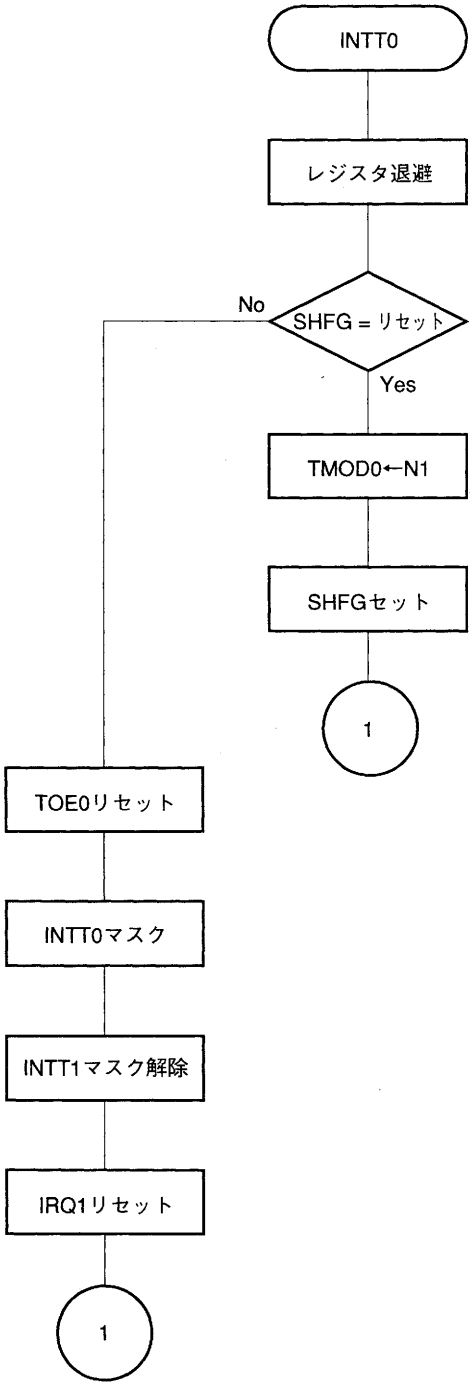
・INT1：立ち下がりエッジ検出モード

（3）フロー・チャート

INT1処理



INTT0処理


INTT0
レジスタ退避
No
SHFG＝リセット
Yes
TMOD0←N1
出力時間を設定
SHFGセット
1
TOE0リセット
TOUT出力を禁止（P20/PTO0←ロウ・レベル）
INTT0マスク
INTT1マスク解除
IRQ1リセット
ワンショット出力中にP11/INT1端子により検出した立ち下がりエッジをキャンセル
1

（4）プログラム例

```
        VENT3   MBE=0,RBE=0,INT1
        VENT4   MBE=0,RBE=0,INTT0
DSHOT   DSEG    AT      28H             ;データ・メモリを定義
FLAG:   DS      1H
SHFG    EQU     FLAG.0
        PUBLIC  SHFG
        EXTRN   N0,N1

    ;〈サブルーチン　INT1処理（MBE＝0）〉
INT1    CSEG    INBLOCK
        PUSH    XA
        MOV     XA,#N0
        MOV     TMOD0,XA                ;ディレイ時間を設定
        SET1    TM0.3
        CLR1    SHFG
        SET1    TOE0                    ;TOUT出力を許可
        DI      IE1                     ;INT1を禁止
        EI      IET0                    ;INTT0を許可
RETURN:
        POP     XA
        RETI

    ;〈サブルーチン　INTT0処理（MBE＝0）〉
INTT0:
        PUSH    XA
        SKF     SHFG
        BR      OVER
        MOV     XA,#N1
        MOV     TMOD0,XA                ;出力時間を設定
        SET1    SHFG
        BR      RETURN
OVER:                                   ;出力停止
        CLR1    TOE0                    ;TOUT出力を禁止
        DI      IET0                    ;INTT0を禁止
        EI      IE1                     ;INT1を許可
        CLR1    IRQ1
        BR      RETURN
```

第５章　時計用タイマの応用

μPD750008は，時計用タイマを１チャネル内蔵しており，次のような機能があります。

① 0.5 sの時間間隔でテスト・フラグ（IRQW）をセットします。またIRQWでスタンバイ・モードを解除することができます。
② メイン・システム・クロック，サブシステム・クロックのいずれでも0.5 s間隔を作ることができます。
③ 早送りモードにより1/128（3.91 ms）の時間間隔となり，プログラムのディバグや検査に便利です。
④ 固定周波数（2.048 kHz）をP23/BUZに出力することができ，ブザー音発生や，システム・クロック発振周波数のトリミングに使用できます。
⑤ 分周回路のクリアができますから，時計を０秒スタートできます。

5.1　時計プログラム

時計用タイマを使った時刻カウント・サブルーチンを紹介します。

このプログラムは，時計用タイマを通常時計モードとし，時計用タイマ割り込み要求フラグ（IRQW）がセットされるごとに秒カウンタをカウントし，120回カウントすると１分とします。

また，分カウンタをカウントするとリターン後スキップします。

（１）プログラムの説明

このプログラムは，**第11章　このアプリケーション・プログラムの応用例**では使用されていません。

〈パブリック宣言シンボル〉

CNT10：10進カウント・サブルーチン・エントリ・ラベル
CNT 6：６進カウント・サブルーチン・エントリ・ラベル
CNT 7：７進カウント・サブルーチン・エントリ・ラベル

〈外部参照宣言シンボル〉

SECD　：秒カウンタ・アドレス
MIND　：分カウンタ・アドレス
HOURD：時カウンタ・アドレス
DAYD　：曜日データ・アドレス

〈使用するレジスタ〉

XA, HL

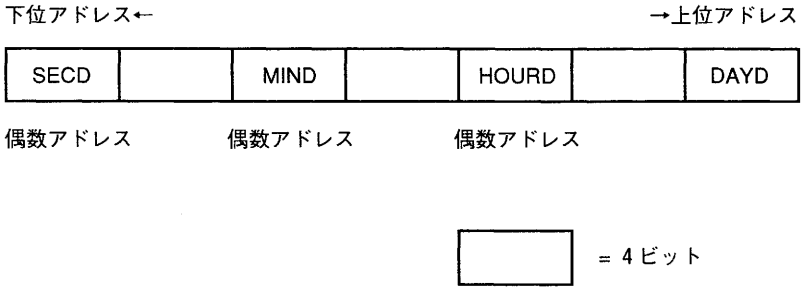
〈使用するRAM〉

アドレス〔H〕	名　　称	用　　途	初　期　値
外部参照	SECD （2×4ビット）	秒カウンタ （バイナリ表現で88H-FFHまでカウント）	88H
外部参照	MIND （2×4ビット）	分カウンタ （デシマル表現で0-59までカウント）	0
外部参照	HOURD （2×4ビット）	時カウンタ （デシマル表現で0-23までカウント）	0
外部参照	DAYD （1×4ビット）	曜日データ　日 月 火 水 木 金 土 0 1 2 3 4 5 6	0

RAMアドレス宣言時の条件

・各シンボルのロウ・アドレスは同じにしてください。また，カラム・アドレスの範囲は0H-EHにしてください。

・各シンボルは下の図のように配置してください。



〈ネスティング〉

2レベル（8ワード）

〈使用するハードウエア〉

時計用タイマ（初期設定：通常時計モード）

〈初期設定〉

・使用するRAMの初期値を設定

・時計用タイマ・モード設定およびクリア・スタート

〈起動方法〉

（ｉ）0.5 s以内に1回以上WATCHをコールします。

（ii）WATCHをコールするとIRQWの状態をチェックして，次の2つの処理に分かれます。

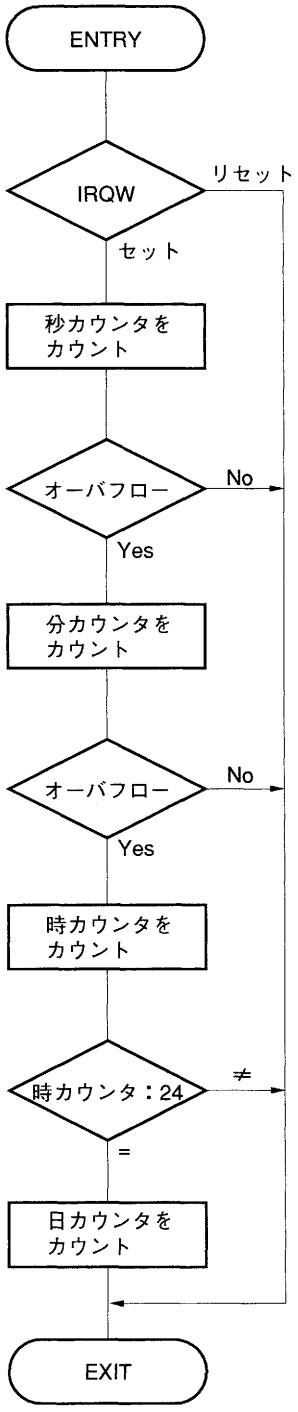
・IRQW = 0の場合

RET命令でユーザ・プログラムに戻ります。

・IRQW = 1の場合

IRQWをリセット（0）したあと時刻のカウントを行い，そのカウント値を秒，分，時間，曜日ごとに各エリアに格納し，RET命令でユーザ・プログラムに戻ります。ただし，分の桁が変化した場合はRETS命令でユーザ・プログラムに戻ります。

（2）フロー・チャート


ENTRY
IRQW
リセット
セット
秒カウンタを
カウント
オーバフロー
No
Yes
分カウンタを
カウント
オーバフロー
No
Yes
時カウンタを
カウント
時カウンタ：24
≠
=
日カウンタを
カウント
EXIT

（３）プログラム例

```
EXTRN     DATA(SECD,MIND,HOURD,DAYD)
PUBLIC    CNT10,CNT6,CNT7

      ;〈サブルーチン　時計カウント〉
WATCH     CSEG      SENT
          CLR1      MBE
          SKTCLR    IRQW              ;0.5秒チェック
          RET
          INCS      SECD              ;秒カウンタのカウント
          RET
          INCS      SECD + 1
          RET

          MOV       XA,#100H - 120
          MOV       SECD,XA           ;分カウンタのカウント
          MOV       HL,#MIND
          CALLF     !CNT10
          RETS
          CALLF     !CNT6
          RETS
          CALLF     !CNT10            ;時カウンタのカウント
          BR        WATCH2
          INCS      @HL
WATCH2：
          INCS      L
          MOV       XA,HOURD
          SKE       A,#4H
          RETS
          SKE       X,#2H
          RETS
          MOV       XA,#00H
          MOV       HOURD,XA
          CALLF     !CNT7             ;曜日を更新
          NOP
          RETS
```

```
      ;〈サブルーチン　N進カウント〉
CNTN      CSEG      SENT
CNT10:
          MOV       A,#(10H-0AH)        ;10進
CNT6:
          MOV       A,#(10H-6)          ;6進
CNT7:
          MOV       A,#(10H-7)          ;7進
;
          INCS      @HL
          NOP
          ADDS      A,@HL
          BR        NOMRET
          XCH       A,@HL
          INCS      L
          RETS
NOMRET:
          INCS      L
          RET
```

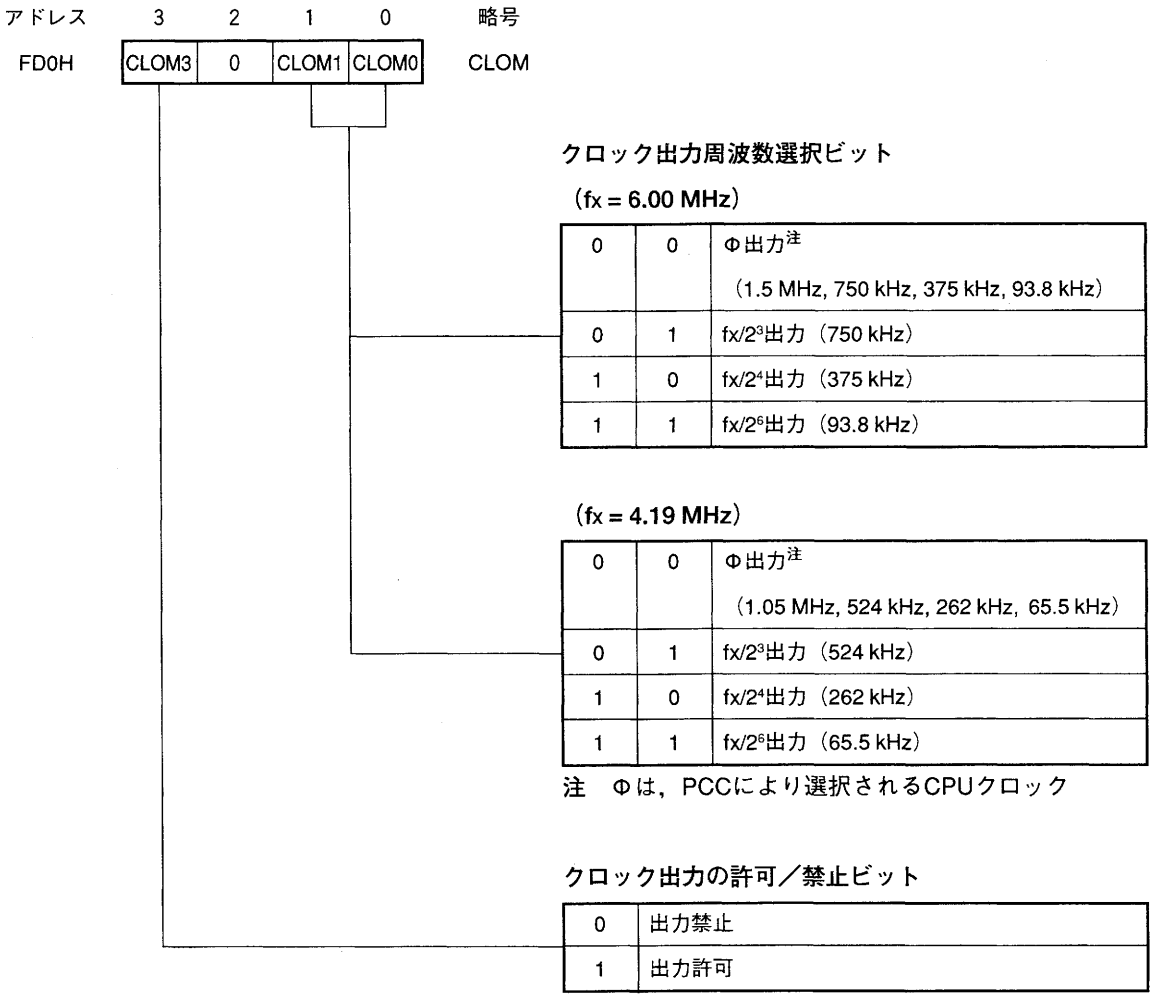
第6章　クロック出力回路の応用

6.1　PCLクロック出力

μPD750008はクロック出力回路を内蔵しており，P22/PCL端子から周辺LSI，あるいはμPD7500シリーズなどのスレーブ・マイコンにクロックを供給することができます。

クロック出力の周波数および出力の許可／禁止は，クロック出力モード・レジスタ（CLOM）によって設定します。クロック出力モード・レジスタのフォーマットを図6－1に示します。

図6－1　クロック出力モード・レジスタのフォーマット



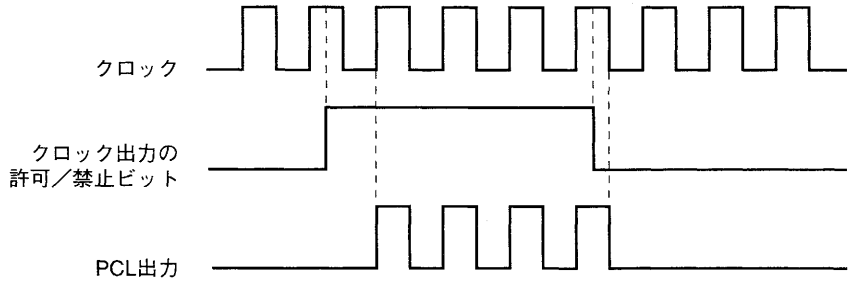
注意　CLOMのビット2には，必ず0を書き込むようにしてください。

次にP22/PCL端子にクロックを出力するために必要な操作を示します。

・クロック出力モード・レジスタによりクロック出力周波数を設定する。出力は禁止しておく。
・Port2.2の出力ラッチをリセットする。
・ポート2を出力ポート・モードにする。
・クロック出力を許可する。

また，μPD750008では，図6－2のようにクロック出力の許可／禁止切り替え時には幅の狭いパルスが出ないようになっています。

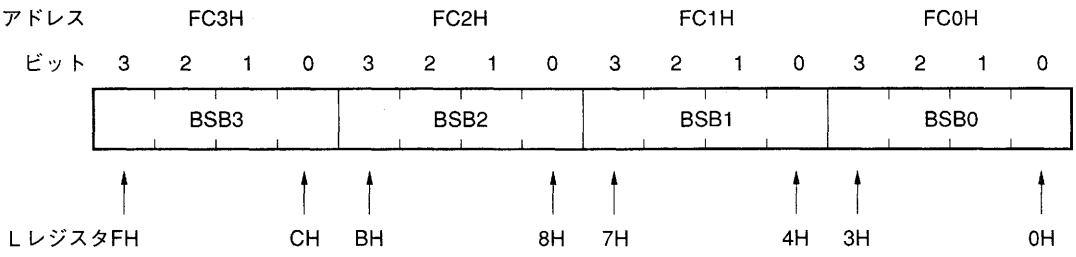
図6－2　クロック出力波形



第7章　ビット・シーケンシャル・バッファの応用

ビット・シーケンシャル・バッファ（BSB）はビット操作用の特殊データ・メモリで，Lレジスタによって間接ビット指定ができます。したがって，Lレジスタをインクリメントあるいはデクリメントするだけで，BSB内の指定ビットを更新することができます。

図7－1　ビット・シーケンシャル・バッファの間接アドレシング

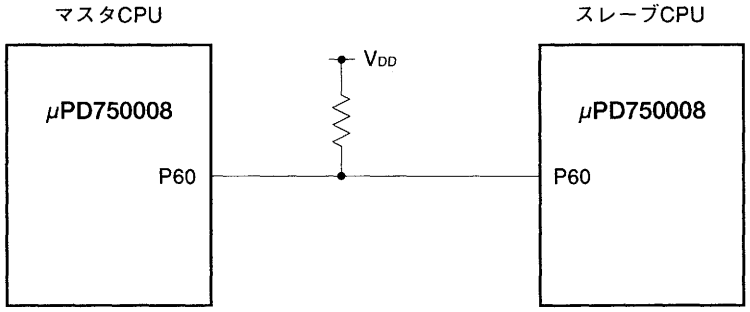


次にBSBの応用例を示します。

7.1　高速シリアル・データ転送

μPD750008で図7－2のようにマルチプロセッサ・システムを構成した場合，ビット・シーケンシャル・バッファを用いてシリアル・データ転送を行います。信号線は1本とし，スレーブCPUからの$\overline{BUSY}$信号出力と送信および受信を兼用します。

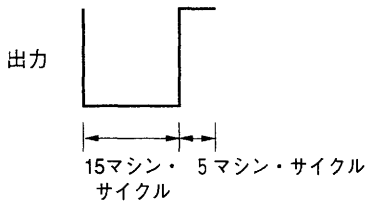
図7－2　マルチプロセッサ・システム構成図（高速シリアル・データ転送）



シリアル・データのデータ長はN（1-16）ビットとします。そのフォーマットを次に示します。

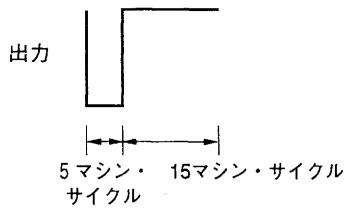
・データが0のとき

ロウ・レベルを15マシン・サイクル出力し，続けてハイ・レベルを5マシン・サイクル出力する。



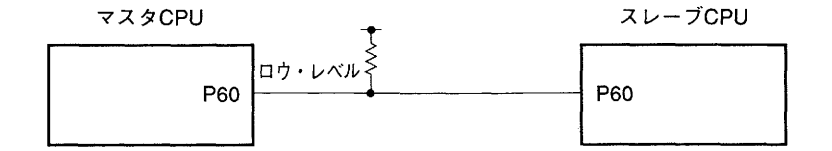
・データが1のとき

ロウ・レベルを5マシン・サイクル出力し，続けてハイ・レベルを15マシン・サイクル出力する。

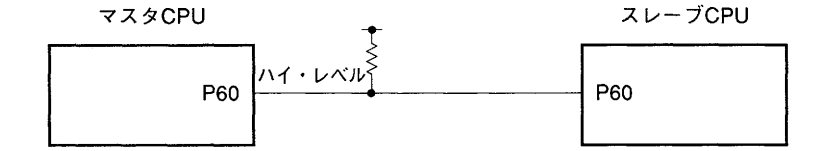


転送は，スレーブCPUからの$\overline{BUSY}$信号がオン（ハイ・レベル）のときに，マスタCPUからNビットのデータを送信し，続けてスレーブCPUからNビットのデータを送信します。転送データの取り込みは，先頭データの立ち下がりによって開始します。転送の流れを次に示します。

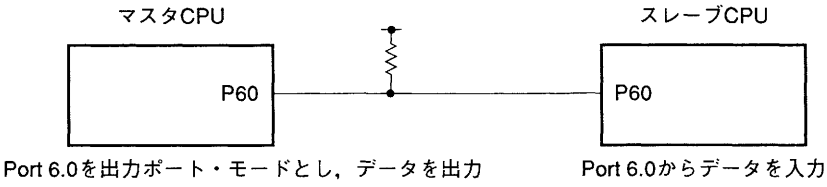
（a）スレーブCPUからの $\overline{\text{BUSY}}$ 信号オン



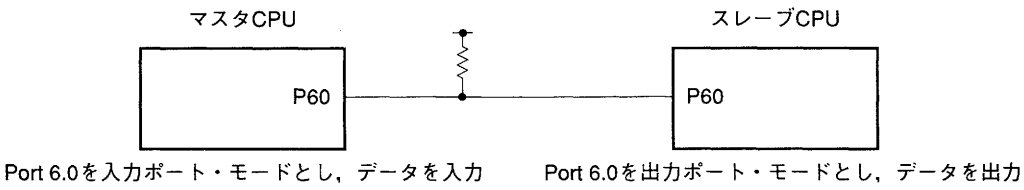
（b）スレーブCPUからの $\overline{\text{BUSY}}$ 信号オフ



（c）マスタCPUからNビットのデータを出力



（d）スレーブCPUからNビットのデータを出力



（1）プログラムの説明

このプログラムは，**第11章　このアプリケーション・プログラムの応用**では使用されていません。

〈パブリック宣言シンボル〉

TDATA, RDATA

〈使用するレジスタ〉

XA, L

〈使用するRAM〉

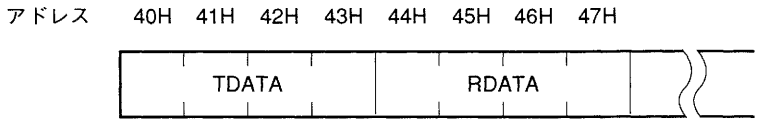
送信データ・エリアと受信データ・エリアを持ち，両エリアの下位アドレスからNビットに転送データを格納します。

RAM上には，メモリ・バンク0の20H-7FH番地に配置することができます。ただし，このときのアドレスは偶数アドレスとします。

TDATA：　4ワード　　；送信データ・エリア。下位アドレスからNビット分のデータを送信する。

RDATA：　4ワード　　；受信データ・エリア。受信したデータを下位Nビットに格納する。

図7－3　プログラム上のRAMのレイアウト



〈ネスティング〉

レベル1（4ワード）

〈使用するハードウエア〉

・ポート：Port6.0

・ビット・シーケンシャル・バッファ0-3

〈初期設定〉

・マスタ　PCC←3H

Port6.0←入力モード

・スレーブ　PCC←3H

Port6.0←出力モード

Port6.0←0

〈メイン処理〉

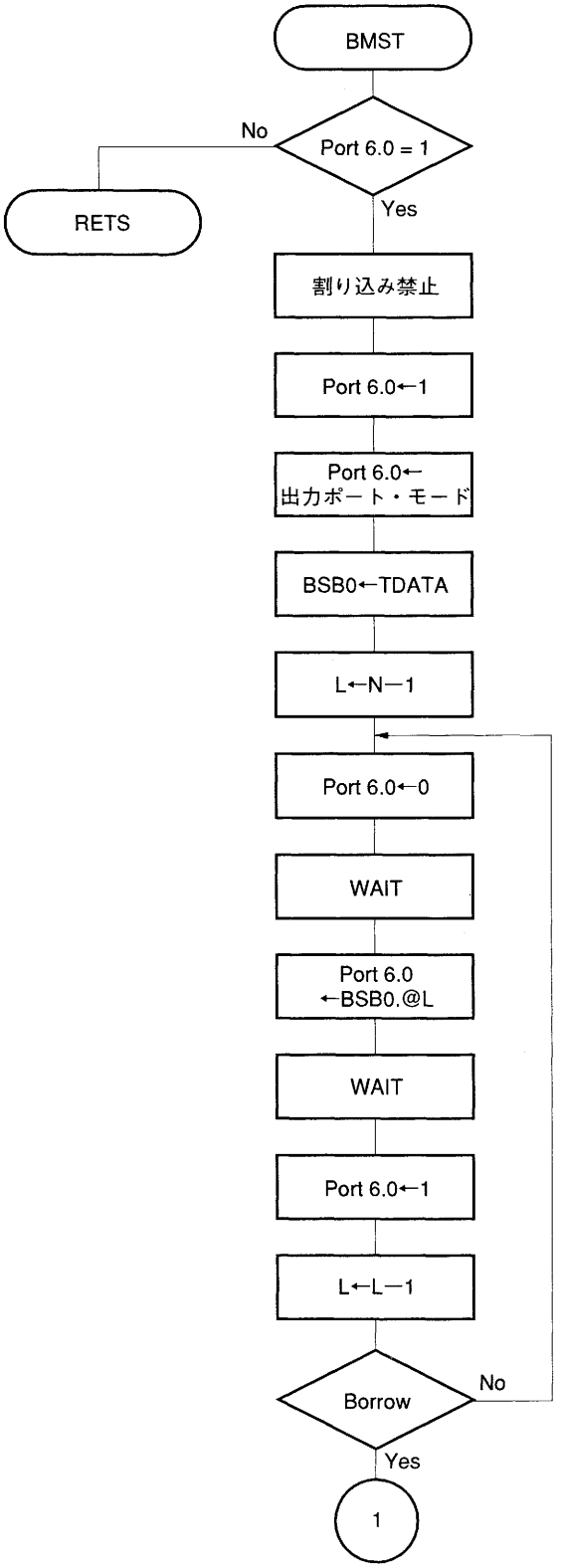
・マスタ　CALL　!BMST

・スレーブ　CALL　!BSLV

（2）マスタCPU

（a）フロー・チャート

転送処理

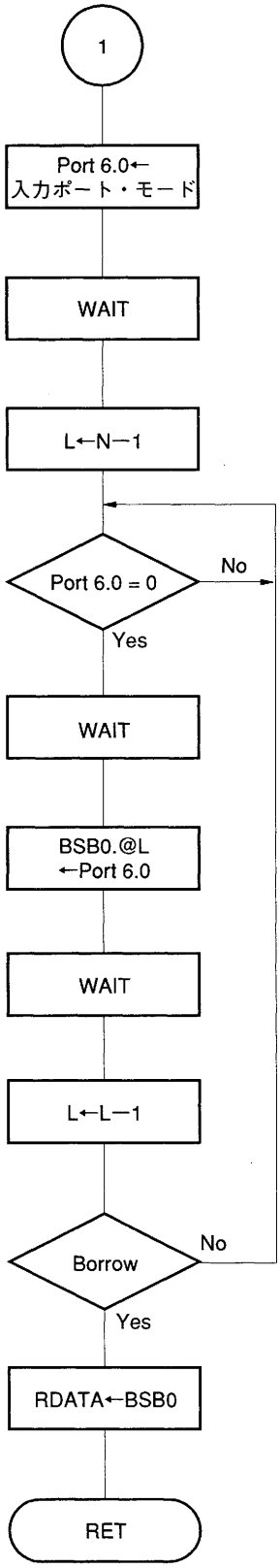


Port 6.0がハイ・レベルのとき転送可能

転送データをビット・シーケンシャル・バッファに設定

Port 6.0にロウ・レベルを出力してから5 μs後に送信データを出力

データ出力から10マシン・サイクル後にハイ・レベルを出力



Port 6.0からロウ・レベルを入力してから約10 μs後にデータ取り込み

入力データをRDATAに格納

（b）プログラム例

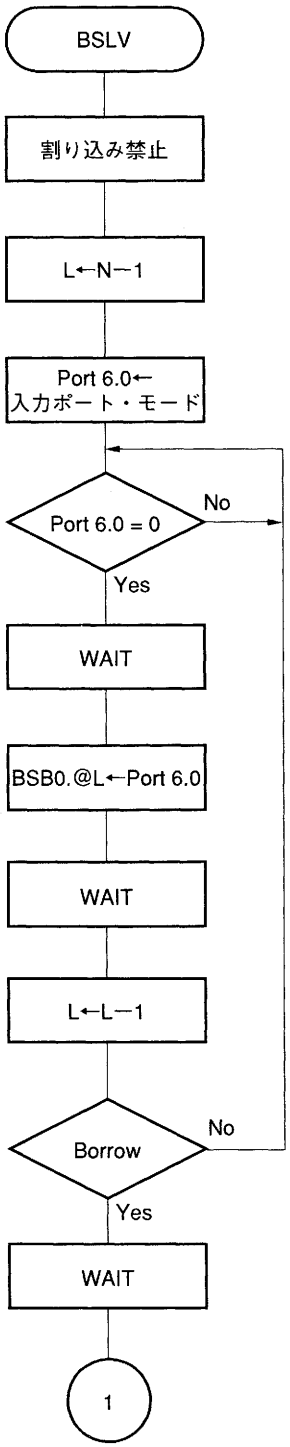
```
        PUBLIC  TDATA,RDATA

DBSIO   DSEG    0       AT      40H

TDATA:  DS      4H                      ;送信データ・エリア
RDATA:  DS      4H                      ;受信データ・エリア

N       EQU     8
```

```
BMST    CSEG    INBLOCK

        CLR1    MBE
        SKT     PORT6.0
        RETS

        DI
        SET1    PORT6.0             ;Port6.0←ハイ・レベル
        MOV     XA,#00010000B       ;Port6.0を出力モードに設定
        MOV     PMGA,XA
        MOV     XA,TDATA            ;送信データをBSBに設定
        MOV     BSB0,XA
        MOV     XA,TDATA+2
        MOV     BSB2,XA
        MOV     L,#N-1
LOOP:
        CLR1    PORT6.0             ;Port6.0←ロウ・レベル
        NOP
        SKF     BSB0.@L             ;Port6.0←BSB0.@L
        SET1    PORT6.0
        MOV     A,#0EH
WAIT:
        INCS    A
        BR      WAIT
        NOP
        SKT     PORT6.0
        SET1    PORT6.0             ;Port6.0←ハイ・レベル
        DECS    L
        BR      LOOP

        MOV     XA,#00000000B       ;Port6.0を入力モードに設定
        MOV     PMGA,XA

        NOP
        NOP
        NOP

        MOV     L,#N-1
CHECK:
        SKF     PORT6.0             ;受信データの立ち下がりをチェック
        BR      CHECK
        NOP
        SET1    BSB0.@L
        SKT     PORT6.0             ;データ取り込み
        CLR1    BSB0.@L
        NOP
        NOP
        NOP
        DECS    L
        BR      CHECK

        MOV     XA,BSB0             ;入力データをRDATAに格納
        MOV     RDATA,XA
        MOV     XA,BSB2
        MOV     RDATA+2,XA
        RET
```

（3）スレーブCPU

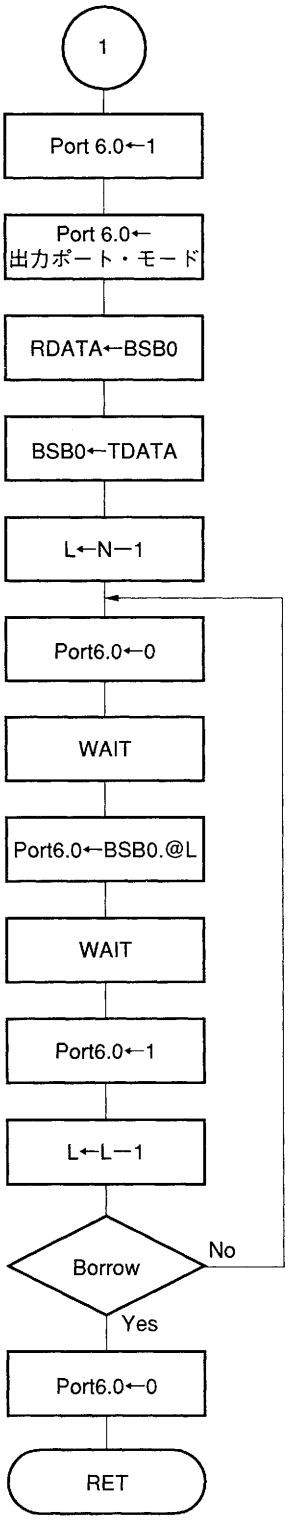
（a）フロー・チャート

転送処理



マスタからの信号の立ち下がりをチェック

Port 6.0からロウ・レベルを入力してから約1.0 μs後にデータ取り込み



入力データをRDATAに格納

転送データをビット・シーケンシャル・バッファに設定

Port6.0にロウ・レベルを出力してから5 μs後に
送信データを出力

$\overline{\text{BUSY}}$信号をオン

（b）プログラム例

```
        PUBLIC  TDATA,RDATA

DBSIOS  DSEG    0       AT      40H

TDATA:  DS      4H                      ;送信データ・エリア
RDATA:  DS      4H                      ;受信データ・エリア

N       EQU     8
```

```
BSLV    CSEG    INBLOCK

        DI
        CLR1    MBE
        MOV     L,#N-1
        MOV     XA,#00000000B       ;Port6.0を入力モードに設定
        MOV     PMGA,XA
CHECK:
        SKF     PORT6.0             ;受信データの立ち上がりチェック
        BR      CHECK
        NOP
        SET1    BSB0.@L             ;Port6.0からデータ取り込み
        SKT     PORT6.0
        CLR1    BSB0.@L
        NOP
        NOP
        NOP
        DECS    L
        BR      CHECK

        NOP
        NOP

        SET1    PORT6.0             ;Port6.0←ハイ・レベル
        MOV     XA,#10H             ;Port6.0を出力モードに設定
        MOV     PMGA,XA

        MOV     XA,BSB0             ;受信データをRDATAに格納
        MOV     RDATA,XA
        MOV     XA,BSB2
        MOV     RDATA+2,XA
        MOV     XA,TDATA
        MOV     BSB0,XA
        MOV     XA,TDATA+2
        MOV     BSB2,XA
        MOV     L,#N-1
LOOP:
        CLR1    PORT6.0             ;Port6.0←ロウ・レベル
        NOP
        SKF     BSB0.@L             ;Port6.0にデータ出力
        SET1    PORT6.0
        MOV     A,#0EH
WAIT:
        INCS    A
        BR      WAIT
        NOP
        SKT     PORT6.0
        SET1    PORT6.0             ;Port6.0←ハイ・レベル
        DECS    L
        BR      LOOP

        CLR1    PORT6.0
        RET
```

〔メ　モ〕

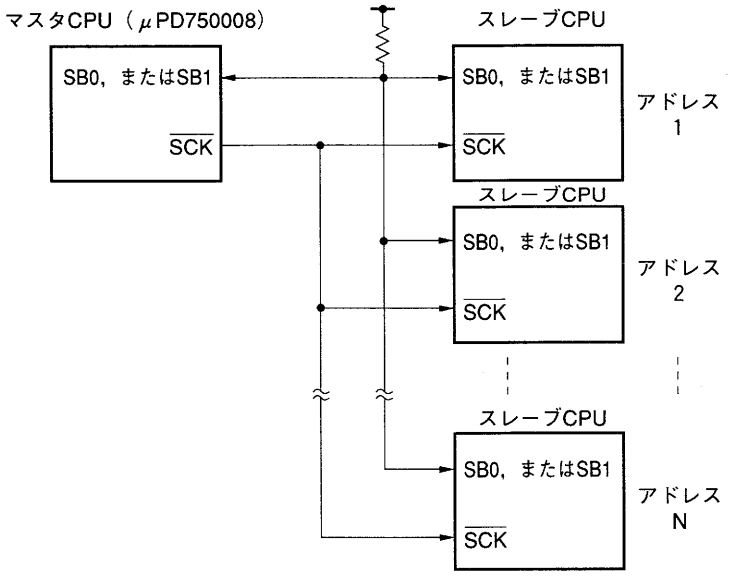
第8章　シリアル・インタフェースの応用

8.1　SBIモードによる応用例

μPD750008のシリアル・インタフェースをSBIモードで動作させる場合の応用例を示します。

図8－1に，SBIによるシリアル・バスの構成例を示します。

図8－1　シリアル・バスの構成例



（1）マスタCPUの応用

マスタCPUは，スレーブCPUに対して，次の処理を行います。

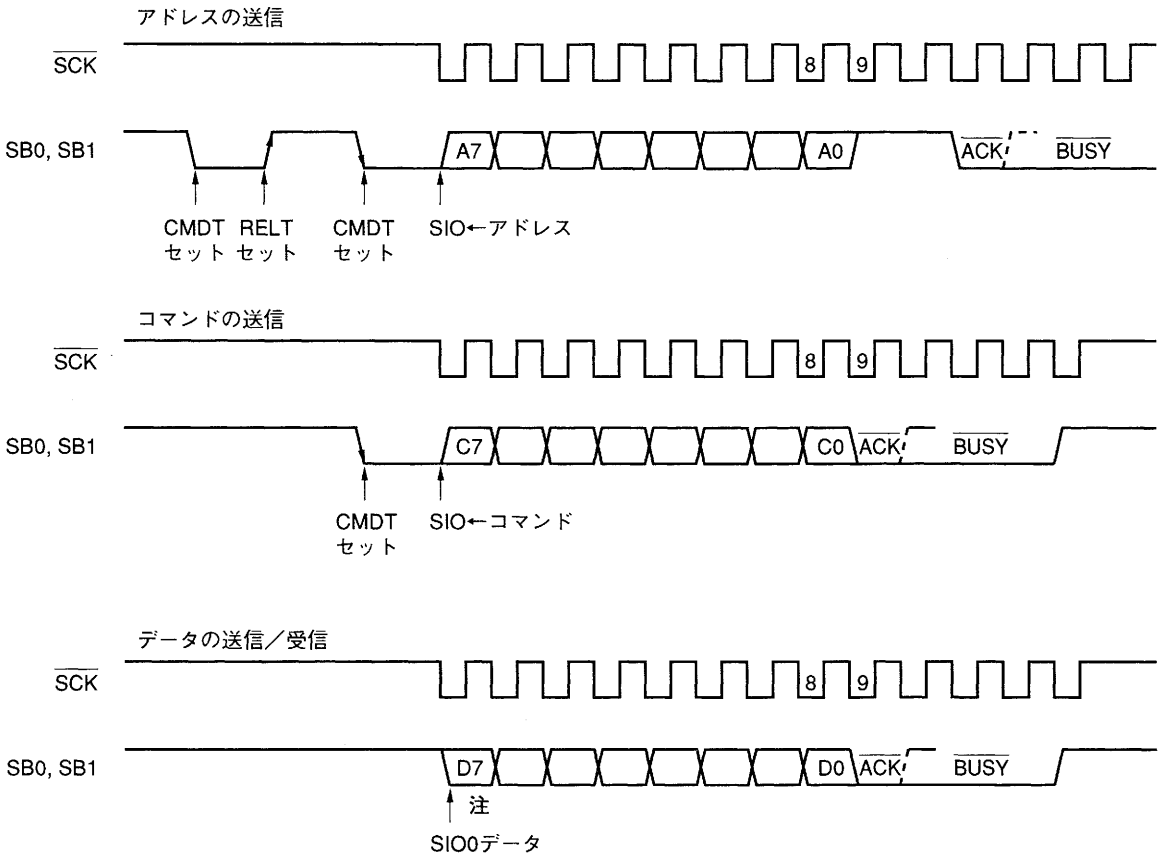
（a）アドレスの送信　……通信を行うスレーブCPUを選択します。

（b）コマンドの送信　｝……（a）により選択したスレーブCPUに対して，

（c）データの送信／受信　｝　コマンドおよびデータにより，通信します。

アドレス，コマンド，データのフォーマットを図8－2に示します。

図8－2　アドレス，コマンド，データのフォーマット

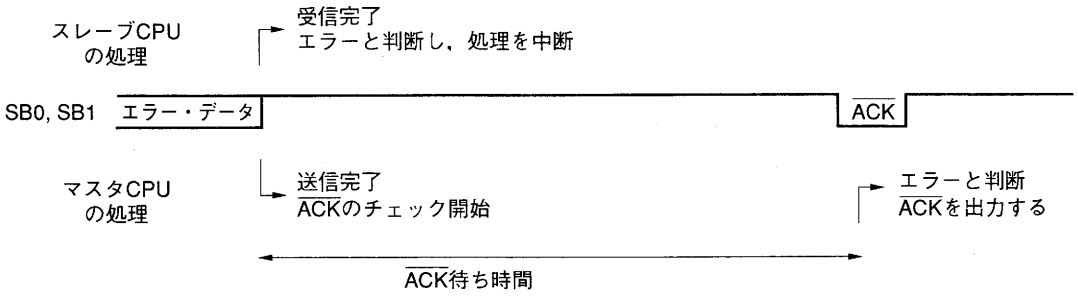


注　データ受信時には，0FFHを書き込む。

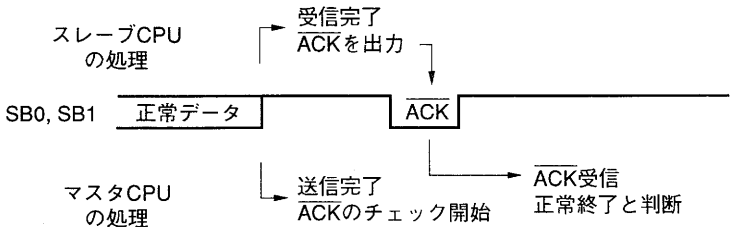
また，スレーブCPUから返してくるアクノリッジ信号（$\overline{ACK}$）により，スレーブCPU側で発生したエラーを判断します。

図8－3　アクノリッジ信号によるエラーの判断

・エラー時



・正常時



マスタCPUは，8ビット（1バイト）・データの転送が完了（INTCSIが発生）したあと，スレーブCPUからのアクノリッジ（$\overline{ACK}$）信号をチェックします。

転送が完了してから一定の時間内にスレーブCPUより$\overline{ACK}$信号が返されなければ，スレーブCPU側でエラーが発生したと判断します。

アドレス，コマンド，データの送信時には，SIOへデータを書き込むとともに，スレーブ・アドレス・レジスタ（SVA）にも同じデータを書き込み，送信データがバス・ライン上で変化した場合のエラー・チェックを行います。

（2）プログラムの説明

このプログラムは，音階や音長の表示データを表示ドライバに転送する例です。

このプログラムは，**第11章　このアプリケーション・プログラムの応用**で使用するものを，入出力インタフェース部を変更して説明しています。

〈使用するレジスタ〉

レジスタ・バンク0：XA

レジスタ・バンク3：XA，BC

〈使用するRAM〉

RAMは，メモリ・バンク0の20H番地から配置します。

MODESBI：	4ワード	；どのデータを転送中かを示す
LCD：	16ワード	；転送データ・エリア
REFRES_F：	1ビット	；1ならば転送開始，またはエラー

〈ネスティング〉

1レベル（6ワード）

〈使用するハードウエア〉

ポート：Port0.1（$\overline{SCK}$兼用端子），Port0.2（SB0兼用端子）

シリアル・インタフェース

〈割り込み〉

INTCSI

〈初期設定〉

・使用するポートの内蔵プルアップ抵抗の接続を指定

・シリアル動作モード

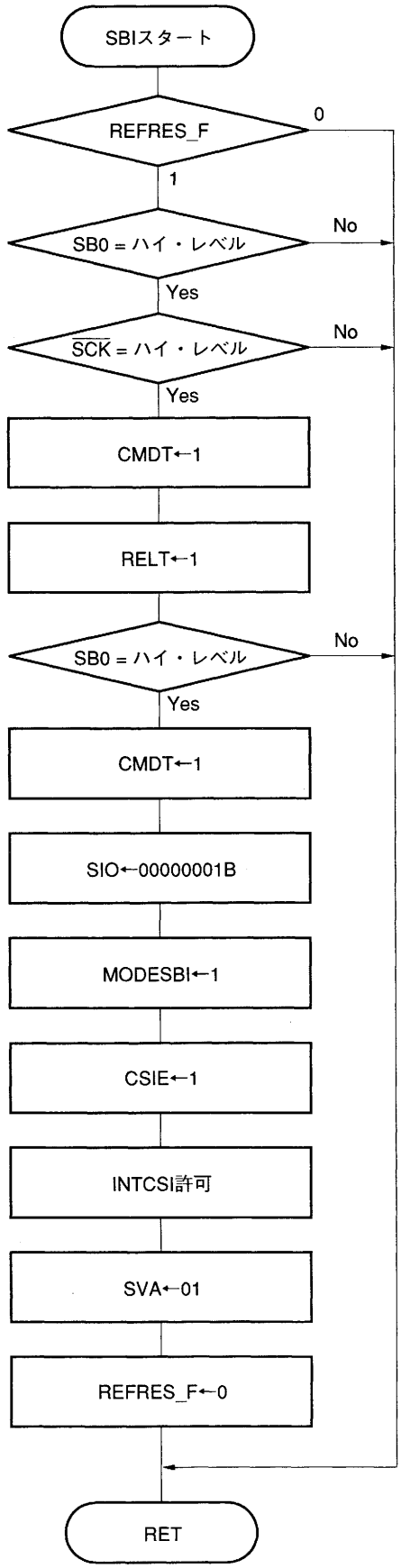
〈起動方法〉

REFRES_Fをセット（1）後，SBIをスタートすると，LCDの先頭から16ワード分のデータをスレーブCPUのアドレス1H番地に対して転送します。

転送が正常に終了すると，REFRES_Fがリセット（0）されます。

エラーが発生すると，REFRES_Fは1のまま，再度転送を行います。

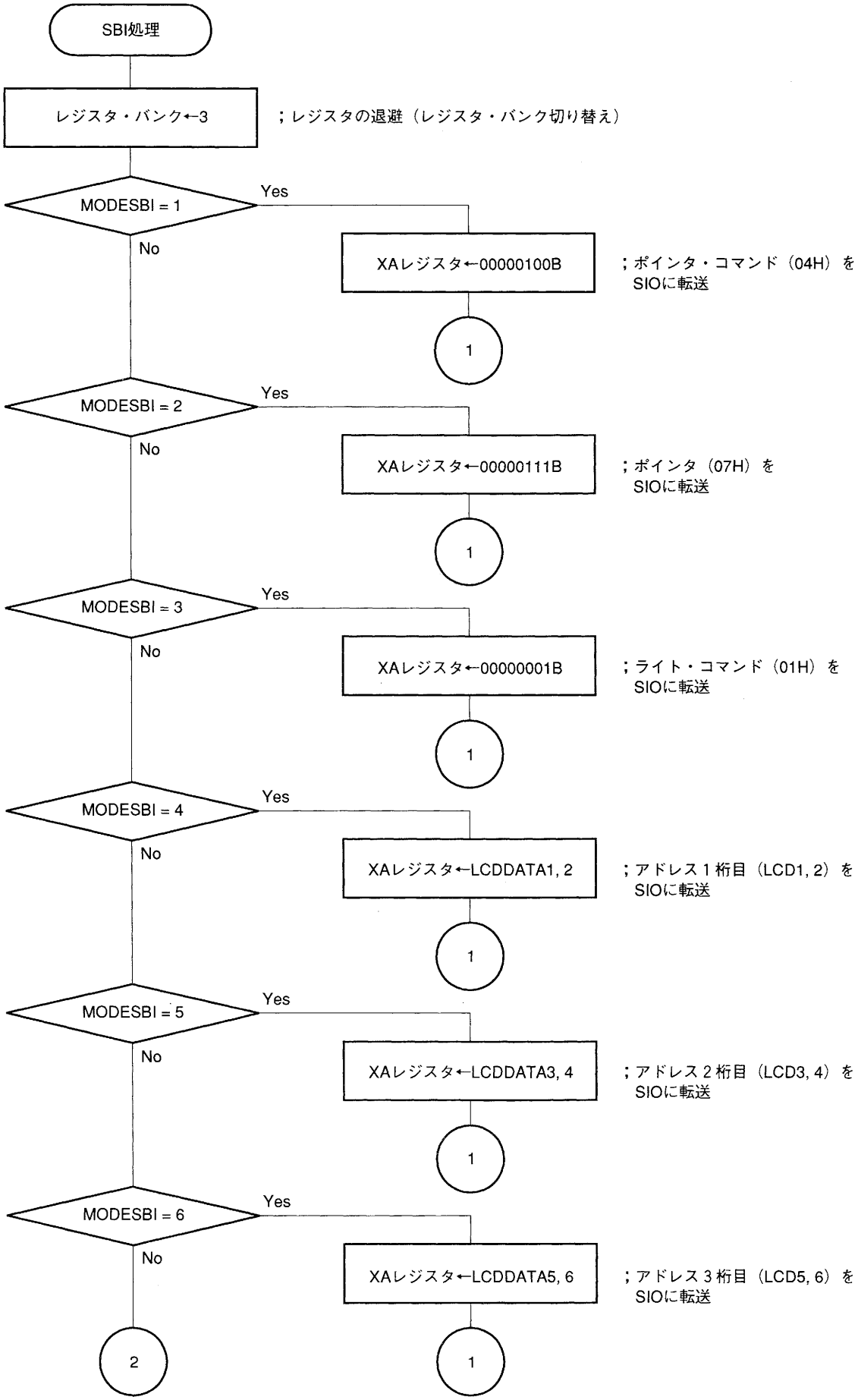
（3）フロー・チャート

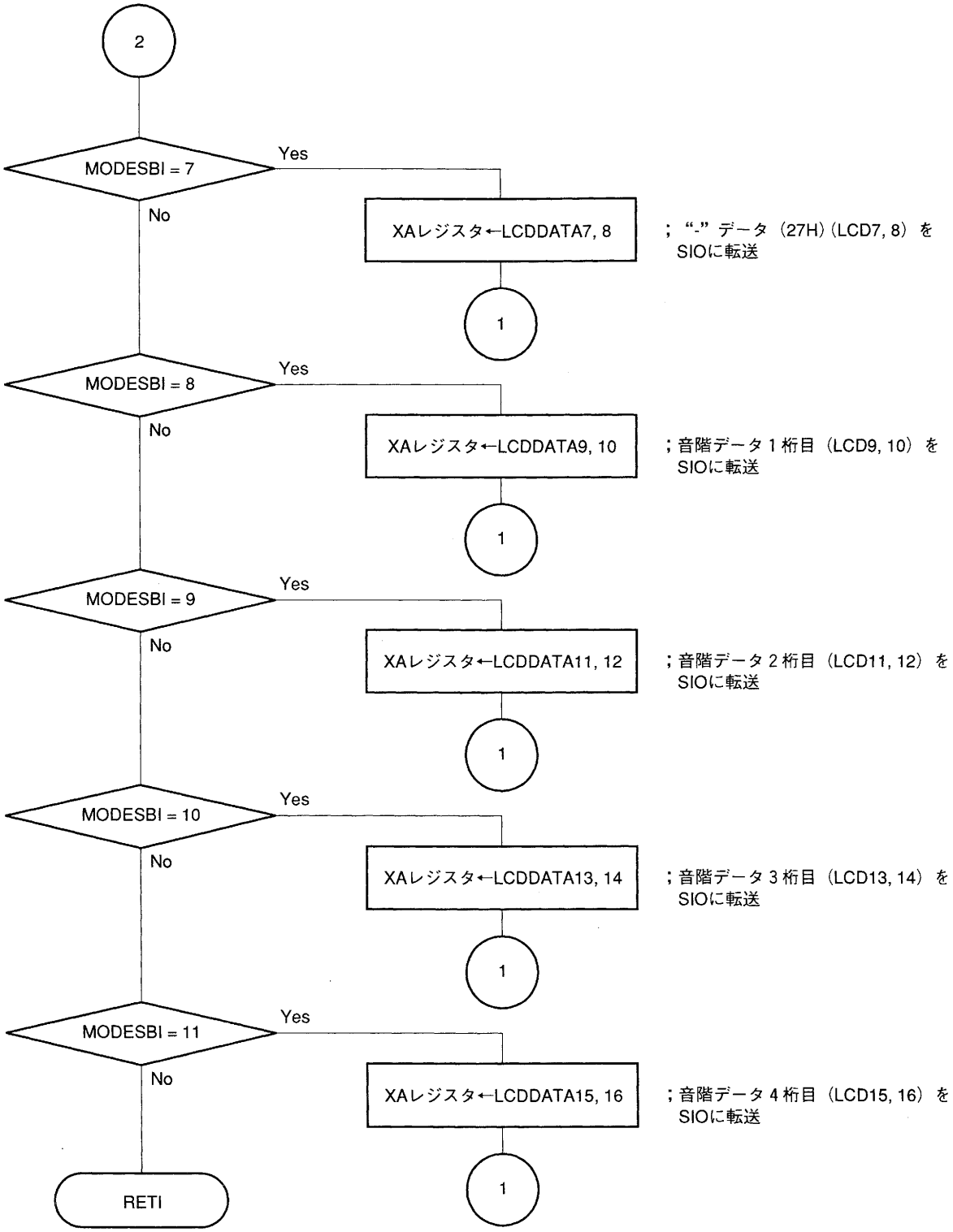


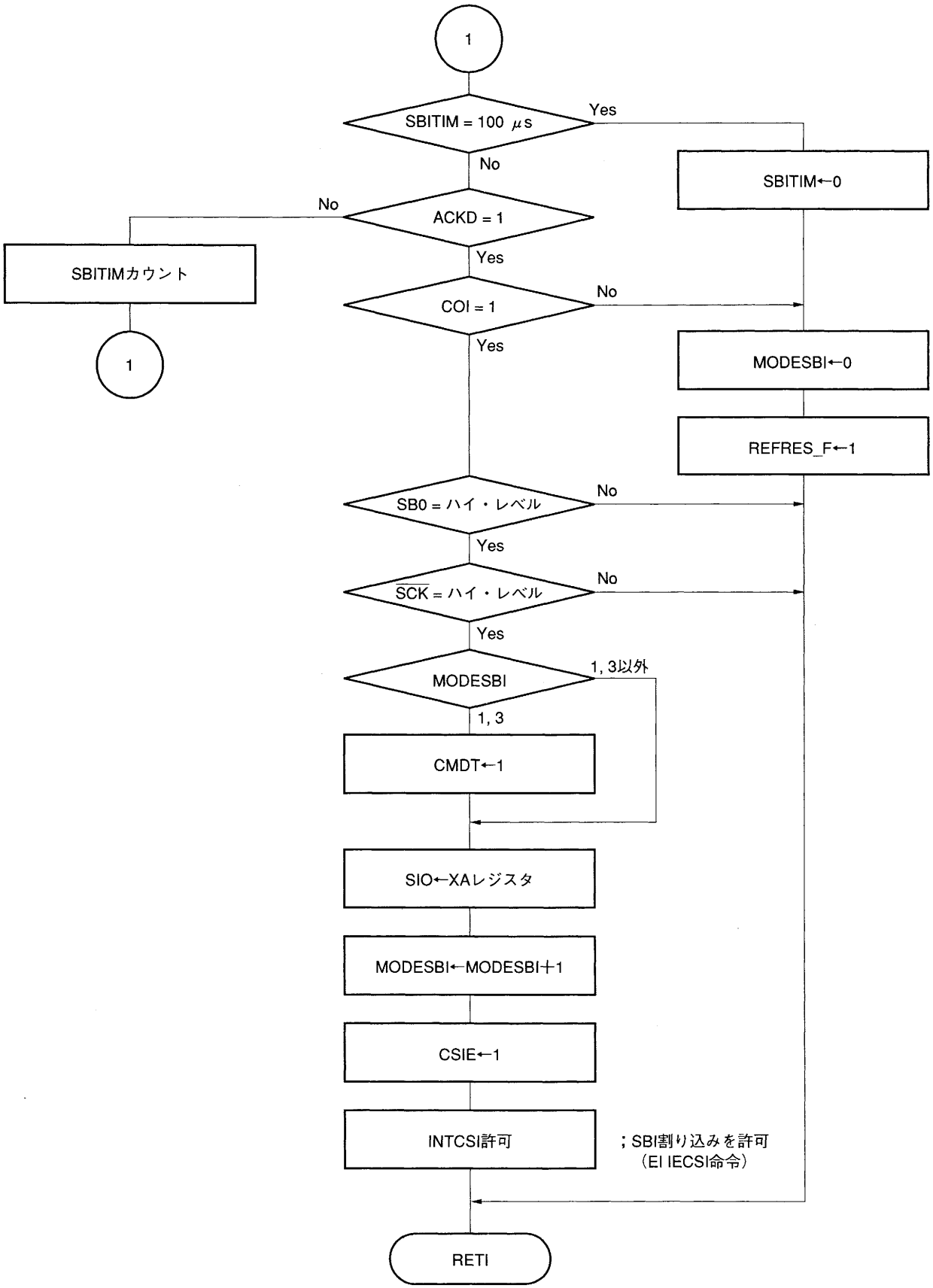
；スレーブ・アドレス（01H）をSIOに転送

；SBI割り込みを許可（EI IECSI命令）

；1ならばLCDをリフレッシュ


SBI処理
レジスタ・バンク←3
；レジスタの退避（レジスタ・バンク切り替え）
MODESBI = 1
Yes
No
XAレジスタ←00000100B
；ポインタ・コマンド（04H）をSIOに転送
1
MODESBI = 2
Yes
No
XAレジスタ←00000111B
；ポインタ（07H）をSIOに転送
1
MODESBI = 3
Yes
No
XAレジスタ←00000001B
；ライト・コマンド（01H）をSIOに転送
1
MODESBI = 4
Yes
No
XAレジスタ←LCDDATA1, 2
；アドレス1桁目（LCD1, 2）をSIOに転送
1
MODESBI = 5
Yes
No
XAレジスタ←LCDDATA3, 4
；アドレス2桁目（LCD3, 4）をSIOに転送
1
MODESBI = 6
Yes
No
XAレジスタ←LCDDATA5, 6
；アドレス3桁目（LCD5, 6）をSIOに転送
2
1


2
MODESBI = 7
Yes
No
XAレジスタ←LCDDATA7, 8
；“-” データ（27H）(LCD7, 8）を SIOに転送
1
MODESBI = 8
Yes
No
XAレジスタ←LCDDATA9, 10
；音階データ1桁目（LCD9, 10）を SIOに転送
1
MODESBI = 9
Yes
No
XAレジスタ←LCDDATA11, 12
；音階データ2桁目（LCD11, 12）を SIOに転送
1
MODESBI = 10
Yes
No
XAレジスタ←LCDDATA13, 14
；音階データ3桁目（LCD13, 14）を SIOに転送
1
MODESBI = 11
Yes
No
XAレジスタ←LCDDATA15, 16
；音階データ4桁目（LCD15, 16）を SIOに転送
1
RETI


1
SBITIM = 100 μs
Yes
No
SBITIM←0
No
ACKD = 1
SBITIMカウント
Yes
COI = 1
No
1
Yes
MODESBI←0
REFRES_F←1
SB0 = ハイ・レベル
No
Yes
SCK = ハイ・レベル
No
Yes
MODESBI
1, 3以外
1, 3
CMDT←1
SIO←XAレジスタ
MODESBI←MODESBI＋1
CSIE←1
INTCSI許可
；SBI割り込みを許可
（EI IECSI命令）
RETI

（4）プログラム例

```
        VENT4   MBE=0,RBE=1,SBI         ;SBI

DSEG0     DSEG 0        AT      20H     ;メモリ・バンク0，アドレス20Hから格納

MODESBI：       DS      1               ;どのデータを転送中かを示す
LCD：           DS      16              ;LCD表示用エリア

    ;〈サブルーチン　初期設定〉
        MOV     XA,#0C1H                ;ポート1，2，3は内蔵プルアップ抵抗の接続を指
                                         定しない
        MOV     POGA,XA                 ;ポート0，6，7は内蔵プルアップ抵抗の接続を指
                                         定する
        MOV     XA,#8AH
        MOV     CSIM,XA                 ;シリアル動作モードを設定
        SET1    RELT
        EI                              ;すべての割り込みを許可

    ;〈サブルーチン　SBIスタート〉
        SKT     REFRES_F                ;1ならばLCDをリフレッシュ
        RET
        SKT     PORT0.2                 ;SBO=ハイ・レベル？
        RET
        SKT     PORT0.1                 ;SCK=ハイ・レベル？
        RET
        SET1    CMDT                    ;CMDTセット
        NOP
        NOP
        SET1    RELT                    ;RELTセット
        NOP
        NOP
        SKT     PORT0.2                 ;SBO=ハイ・レベル？
        RET
        SET1    CMDT                    ;CMDTセット
        MOV     XA,#01H
        MOV     SIO,XA                  ;シフト・レジスタ←スレーブ・アドレス（01H）
        MOV     A,#1H
        MOV     MODESBI,A               ;MODESBI←1H
        EI      IECSI                   ;SBI割り込みを許可
        MOV     XA,#1H
        MOV     SVA,XA                  ;スレーブ・アドレス・レジスタ←1H
        CLR1    REFRES_F                ;REFRES_F← 0
        RET
        ;
```

```
;〈サブルーチン　SBI処理〉
SBI:
        SEL     RB3             ;レジスタ・バンク←3
        DI      IEBT            ;ベーシック・インターバル・タイマ（INTBT）の使用
                                  を禁止
        EI                      ;すべての割り込みを許可
        MOV     A,MODESBI
        ADDS    A,#0FH
        BR      SBI_E
        ADDS    A,#0FH
        BR      SBI1
        ADDS    A,#0FH
        BR      SBI2
        ADDS    A,#0FH
        BR      SBI3
        ADDS    A,#0FH
        BR      SBI4
        ADDS    A,#0FH
        BR      SBI5
        ADDS    A,#0FH
        BR      SBI6
        ADDS    A,#0FH
        BR      SBI7
        ADDS    A,#0FH
        BR      SBI8
        ADDS    A,#0FH
        BR      SBI9
        ADDS    A,#0FH
        BR      SBI10
        ADDS    A,#0FH
        BR      SBI11
        BR      SBI_E

SBI1:
        MOV     BC,#04H         ;シフト・レジスタ←ポインタ・コマンド（04H）
        BR      SBITIM

SBI2:
        MOV     BC,#07H         ;シフト・レジスタ←ポインタ（07H）
        BR      SBITIM

SBI3:
        MOV     BC,#01H         ;シフト・レジスタ←ライト・コマンド（01H）
        BR      SBITIM

SBI4:
        MOV     XA,LCD          ;シフト・レジスタ←アドレス1桁目
        MOV     BC,XA
        BR      SBITIM

SBI5:
        MOV     XA,LCD+2        ;シフト・レジスタ←アドレス2桁目
        MOV     BC,XA
        BR      SBITIM
```

```
SBI6:
        MOV     XA,LCD+4                ;シフト・レジスタ←アドレス３桁目
        MOV     BC,XA
        BR      SBITIM

SBI7:
        MOV     XA,LCD+6                ;シフト・レジスタ←“_”データ
        MOV     BC,XA
        BR      SBITIM

SBI8:
        MOV     XA,LCD+8                ;シフト・レジスタ←音階データ１桁目
        MOV     BC,XA
        BR      SBITIM

SBI9:
        MOV     XA,LCD+10               ;シフト・レジスタ←音階データ２桁目
        MOV     BC,XA
        BR      SBITIM

SBI10:
        MOV     XA,LCD+12               ;シフト・レジスタ←音階データ３桁目
        MOV     BC,XA
        BR      SBITIM

SBI11:
        MOV     XA,LCD+14               ;シフト・レジスタ←音階データ４桁目
        MOV     BC,XA
SBITIM:
        MOV     L,#1110B                ;100 μsカウンタの初期化
SBICNT:
        DECS    L
        BR      SBI_ACKD


    ;〈サブルーチン　エラー処理〉
SBI0:
        MOV     A,#0H
        MOV     MODESBI,A               ;MODESBI←0
        SET1    REFRES_F                ;１ならばLCD表示をリフレッシュ
        BR      SBI_E

SBI_ACKD:
        SKT     ACKD                    ;アクノリッジ検出フラグ=1 ?
        BR      SBICNT
        SKT     COI                     ;アドレス・コンパレータ信号=1 ?
        BR      SBI0
        SKT     PORT0.2                 ;SB0=ハイ・レベル ?
        BR      SBI_E
        SKT     PORT0.1                 ;SCK=ハイ・レベル ?
        BR      SBI_E
        MOV     A,MODESBI
        ADDS    A,#0FH                  ;MODESBI=0 ?
        BR      SBI_SIO
        ADDS    A,#0FH                  ;MODESBI=1 ?
        BR      SBI_CMDT
```

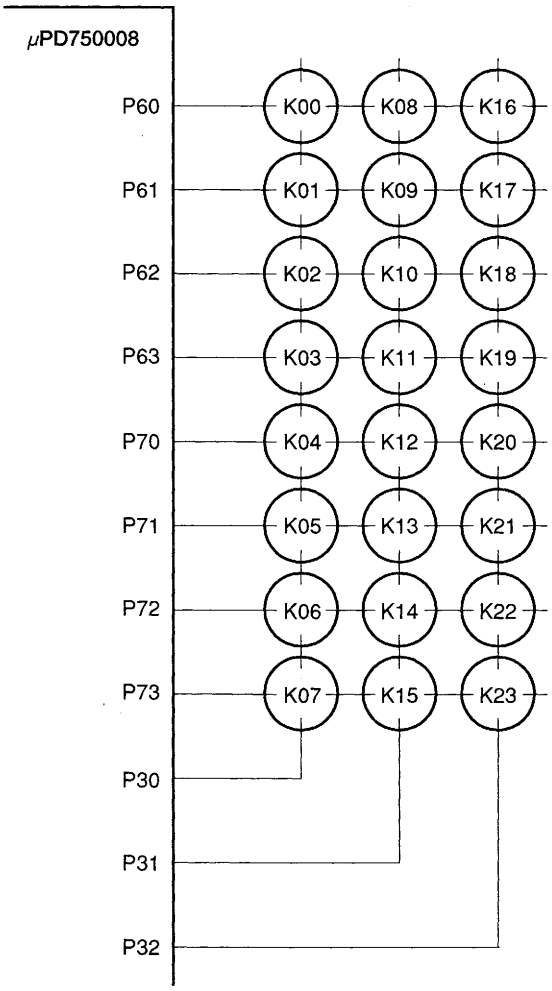
```
        ADDS    A,#0FH              ;MODESBI=2 ?
        BR      SBI_SIO
        ADDS    A,#0FH              ;MODESBI=3 ?
SBI_CMDT:
        SET1    CMDT                ;CMDTセット
SBI_SIO:
        MOV     XA,BC
        MOV     SIO,XA              ;シフト・レジスタ←BCレジスタ
        INCS    MODESBI             ;MODESBI＋1
        NOP
        SET1    CSIE                ;シリアル・インタフェース動作，シフト・レジスタの
                                     使用を禁止
        EI      IECSI               ;SBI割り込みを許可
SBI_E:
        RETI
        ;
```

第９章　キー入力サブルーチン

モメンタリ・キー（３×８）のキー入力を行います。

ここでは，キー・スキャン信号出力にポート３，リターン信号の取り込みにポート６，７を使用して，図９－１のようなキー・マトリクスを構成します。

図９－１　キー・マトリクスの構成



このプログラムでは，８msごとにキー・スキャンを行い，３回連続でキー・データが一致するとキーが確定（KEYCHKF＝1）します。

入力したキー・データは４ビットのキー・ソース・データ（KEYS）と８ビットのキー・リターン・データ（KEYR）に格納します。各キーとキー・データの関係を次に示します。

各キーとキー・データの関係

キー	KEYS	KEYR
K00	: 1110B	11111110B
K01	: 1110B	11111101B
K02	: 1110B	11111011B
K03	: 1110B	11110111B
K04	: 1110B	11101111B
K05	: 1110B	11011111B
K06	: 1110B	10111111B
K07	: 1110B	01111111B
K08	: 1101B	11111110B
K09	: 1101B	11111101B
K10	: 1101B	11111011B
K11	: 1101B	11110111B
K12	: 1101B	11101111B
K13	: 1101B	11011111B
K14	: 1101B	10111111B
K15	: 1101B	01111111B
K16	: 1011B	11111110B
K17	: 1011B	11111101B
K18	: 1011B	11111011B
K19	: 1011B	11110111B
K20	: 1011B	11101111B
K21	: 1011B	11011111B
K22	: 1011B	10111111B
K23	: 1011B	01111111B

（1）プログラムの説明

このプログラムは，**第11章　このアプリケーション・プログラムの応用**で使用するものを，入出力インタフェース部を変更して説明しています。

〈使用するレジスタ〉

XA, BC, HL

〈使用するRAM〉

RAMは，メモリ・バンク0の20H番地から配置します。

LEDDIG：	1ワード	；LED表示桁番号。2 msごとに次のように変化する
		；0111B→1011B→1101B→1110B
KEYCODE：	2ワード	；キー・ソース，キー・リターンをコード化する
KEYR：	2ワード	；キー・リターン
KEYRB：	2ワード	；前回のキー・リターン
KEYTIM：	2ワード	；キー・オン時間カウント用タイマ
KEYS：	1ワード	；キー・ソース
KEYSB：	1ワード	；前回のキー・ソース
KEYCATT：	1ワード	；キー・チャタリング・カウンタ
KEYF：	1ワード	；フラグ
KEYONF：	1ビット	；1ならばキー・オン
KEYON2SF：	1ビット	；2秒以上キーを押し続けると，1になる
HTKEYONF：	1ビット	；1ならば本体のキー・オン
KEYCHKF：	1ビット	；1ならばキー・スキャンを終了

〈ネスティング〉

1レベル（6ワード）

〈使用するハードウエア〉

ポート：ポート3，6，7

〈初期設定〉

・ポート3を出力モードにする。

・ポート6，7を入力モードにして，内蔵プルアップ抵抗の接続を指定する。

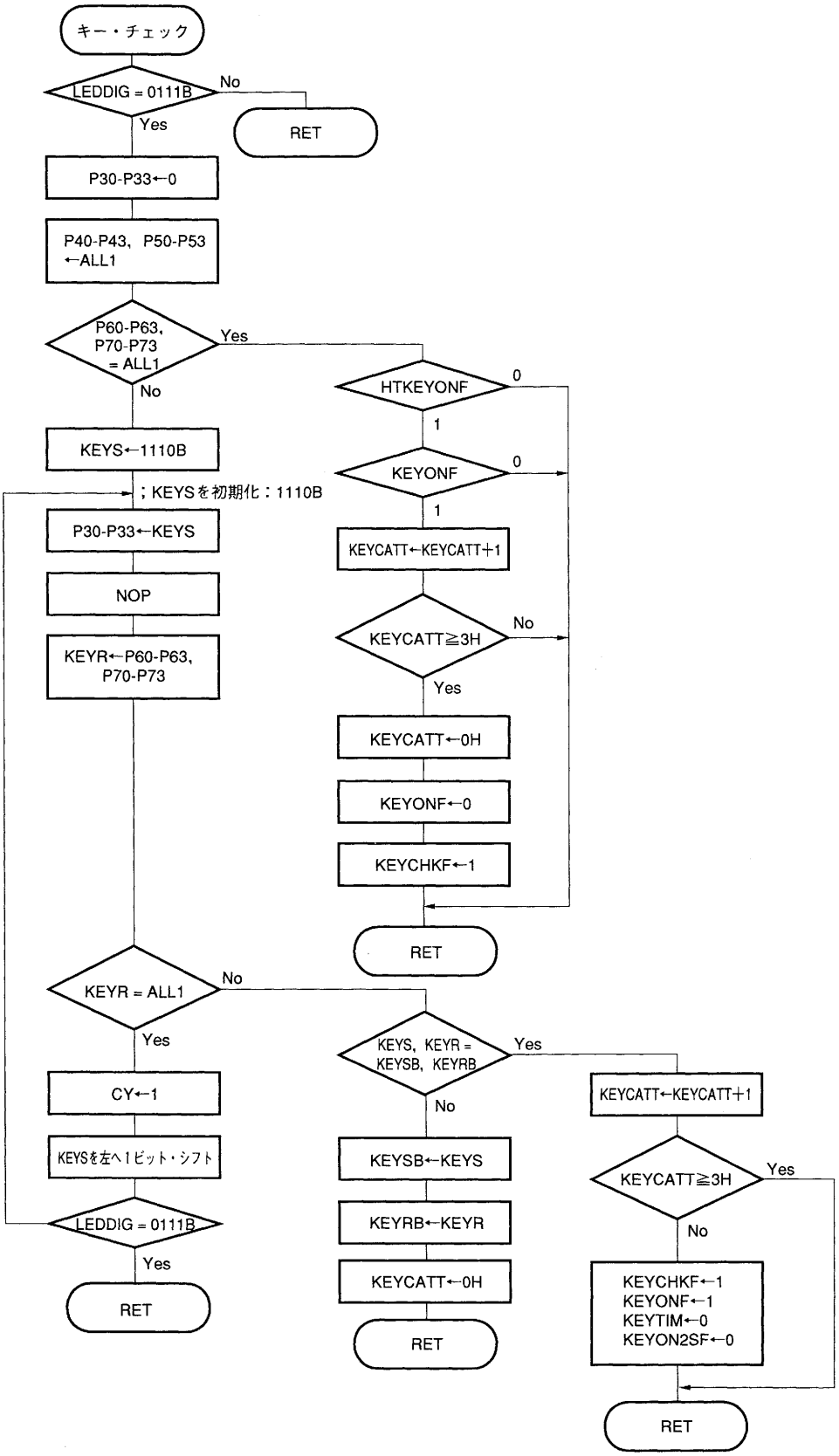
〈起動方法〉

ベーシック・インターバル・タイマ割り込みは2 msごとに発生するとします。

キー・チェック用の8 msのタイマはLED表示用の桁カウンタ（LEDDIG）を利用します。

メイン・ルーチンのキー・デコード処理でキー・スキャンの終了判定を行い，各キーを処理します。

（2）フロー・チャート


キー・チェック
LEDDIG = 0111B
No
RET
Yes
P30-P33←0
P40-P43, P50-P53 ←ALL1
P60-P63, P70-P73 = ALL1
Yes
No
HTKEYONF
0
1
KEYONF
0
1
KEYCATT←KEYCATT+1
KEYCATT≧3H
No
Yes
KEYCATT←0H
KEYONF←0
KEYCHKF←1
RET
KEYS←1110B
；KEYSを初期化：1110B
P30-P33←KEYS
NOP
KEYR←P60-P63, P70-P73
KEYR = ALL1
No
Yes
KEYS, KEYR = KEYSB, KEYRB
Yes
No
KEYCATT←KEYCATT+1
KEYCATT≧3H
Yes
No
KEYSB←KEYS
KEYRB←KEYR
KEYCATT←0H
RET
CY←1
KEYSを左へ1ビット・シフト
LEDDIG = 0111B
Yes
RET
KEYCHKF←1
KEYONF←1
KEYTIM←0
KEYON2SF←0
RET

（3）プログラム例

```
DSEG0       DSEG  0     AT     20H      ;メモリ・バンク0，アドレス20Hから格納

LEDDIG:           DS    1              ;LED表示桁番号

KEYCODE:          DS    2              ;キー・ソース，キー・リターンをコード化
KEYR:             DS    2              ;キー・リターン
KEYRB:            DS    2              ;前回のキー・リターン
KEYTIM:           DS    2              ;キー・オン時間カウント用タイマ
KEYS:             DS    1              ;キー・ソース
KEYSB:            DS    1              ;前回のキー・ソース
KEYCATT:          DS    1              ;キーのチャタリング・カウンタ


KEYFLG:           DS    1              ;フラグ
 KEYONF           EQU   KEYFLG.3       ;1ならばキー・オン
 KEYON2SF         EQU   KEYFLG.2       ;1ならば2秒以上キーを押し続けている
 HTKEYONF         EQU   KEYFLG.1       ;1ならば本体キー・オン
 KEYCHKF          EQU   KEYFLG.0       ;1ならばキー・スキャン終了

;*****  KEYRﾃﾞｰﾀ      *****

KEYR_1            EQU   11111110B
KEYR_2            EQU   11111101B
KEYR_3            EQU   11111011B
KEYR_4            EQU   11110111B
KEYR_5            EQU   11101111B
KEYR_6            EQU   11011111B
KEYR_7            EQU   10111111B
KEYR_8            EQU   01111111B

      ;〈サブルーチン　初期化〉
        MOV     A,#0FH
        OUT     PORT3,A
        MOV     XA,#0FH
        MOV     PMGA,XA             ;ポート3を出力モードに，ポート6を入力モードに
        MOV     XA,#34H
        MOV     PMGB,XA             ;ポート2，4，5を出力モードに，ポート7を入力モードに
        MOV     XA,#0C1H            ;ポート1，2，3に内蔵プルアップ抵抗を接続
        MOV     POGA,XA             ;ポート0，6，7に内蔵プルアップ抵抗を接続
```

```
      ;メイン・ルーチン
;++++++++++++++++++++++++++++++++
;         キー・デコード
;++++++++++++++++++++++++++++++++
KEYDEC:
         SKT      KEYCHKF                ;1ならばキー・スキャン終了
         BR       KEYEND

         CLR1     KEYCHKF
         SKT      KEYONF                 ;1ならばキー・オン
         BR       KEYOFF
;-------------------------
;         キー・オン処理
;-------------------------
         BR       KEYEND
         ;
KEYOFF:
;-------------------------
;         キー・オフ処理
;-------------------------

KEYEND:


      ;〈サブルーチン　100 msタイマ〉
;-------------------------
;   キー・オン時間のカウント
;-------------------------
         MOV      XA KEYTIM
         MOV      HL,XA
         INCS     HL
         NOP
         MOV      XA,HL
         MOV      KEYTIM,XA              ;キー・オン時間カウント用タイマ＋1
```

```
      ;〈サブルーチン　2 msのベーシック・インターバル・タイマ割り込み〉
;++++++++++++++++++++++++++++++++
;        キー・チェック
;++++++++++++++++++++++++++++++++
KEYCHK:
        MOV     A,LEDDIG
        SKE     A,#0111B                ;8 ms?
        RET

        MOV     A,#0H
        OUT     PORT3,A                 ;ポート3←0
        MOV     XA,#0FFH
        OUT     PORT4,XA                ;ポート4，5←ALL1
        IN      XA,PORT6
        MOV     BC,#0FFH
        SKE     XA,BC                   ;ポート6，7=ALL1?
        BR      KEYCHK1
        SKT     KEYONF                  ;1ならばキー・オン
        RET
        INCS    KEYCATT                 ;チャタリング・カウンタ＋1
        NOP
        MOV     A,KEYCATT
        ADDS    A,#0DH                  ;キーが3回一致?
        RET
        MOV     A,#0H
        MOV     KEYCATT,A               ;チャタリング・カウンタ←0H
        CLR1    KEYONF                  ;KEYONF←0
        SET1    KEYCHKF                 ;1ならばキー・スキャンを終了
        RET

KEYCHK1:
        MOV     A,#1110B
        MOV     KEYS,A                  ;キー・ソース←1110B
KEYCHK2:
        OUT     PORT3,A                 ;ポート3←1110B
        NOP                             ;キー・ソース安定時間待ち
        NOP
        NOP
        NOP
        NOP
        IN      XA,PORT6
        MOV     KEYR,XA                 ;キー・リターン←ポート6，7
        MOV     BC,#0FFH
        SKE     XA,BC                   ;ポート6，7=ALL1?
        BR      KEYCHK3
        SET1    CY                      ;CY←1
        MOV     A,KEYS
        ADDC    XA,XA
        MOV     KEYS,A                  ;キー・ソースを左へ1ビット・シフト
        SKE     A,#0111B
        BR      KEYCHK2
        RET
```

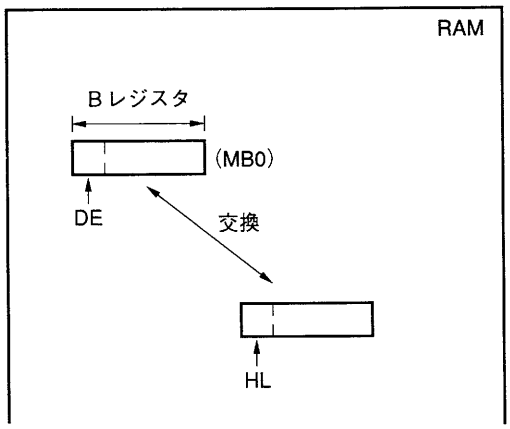
```
KEYCHK3:
        MOV     A,KEYS
        MOV     HL,#KEYSB
        SKE     A,@HL                   ;キー・ソース=KEYSB?
        BR      KEYCHK23
        MOV     A,KEYR
        MOV     HL,#KEYRB
        SKE     A,@HL                   ;キー・リターン=KEYRB?
        BR      KEYCHK23
        MOV     A,KEYR+1
        MOV     HL,#KEYRB+1
        SKE     A,@HL                   ;キー・リターン十1=KEYRB十1?
        BR      KEYCHK23
        INCS    KEYCATT                 ;チャリタング・カウンタ十1
        NOP
        MOV     A,KEYCATT
        ADDS    A,#0DH                  ;キーが3回一致?
        RET
        ;

    ;〈サブルーチン　チャタリング除去終了〉
        SET1    KEYONF
        CLR1    KEYON2SF
        MOV     XA,#0H
        MOV     KEYTIM,XA               ;2sタイマを起動
        SET1    KEYCHKF                 ;1ならばキー・スキャンを終了
        RET
        ;

    ;〈サブルーチン　前回とキーの内容が違う〉
KEYCHK23:
        MOV     A,KEYS
        MOV     KEYSB,A                 ;KEYSB←キー・ソース
        MOV     XA,KEYR
        MOV     KEYRB,XA                ;KEYRB←キー・リターン
        MOV     A,#0H
        MOV     KEYCATT,A               ;チャタリング・カウンタ←0
        RET
```

第10章　各種サブルーチン

10.1　データ転送

メモリ・バンク0とMBE・MBSによって選択されたメモリ・バンクとの間で最大16ワードまでのデータの交換を行います。

交換するデータの先頭アドレスをDEレジスタ（メモリ・バンク＝0）とHLレジスタ（メモリ・バンク＝MBE・MBS）で指定し，データ長（ワード数）をBレジスタで指定します。ただし，HLレジスタ，DEレジスタで指定するアドレスは，それぞれ2行にわたらないものとします。



プログラム例

このプログラムは，**第11章　このアプリケーション・プログラムの応用例**には使用されていません。

- **ステップ数**：9ステップ（9バイト）
- **使用するレジスタ**：A, E, DE, HL

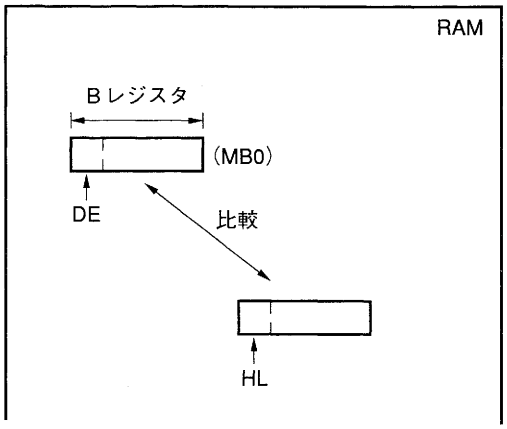
```
EXCH    CSEG    INBLOCK

EXCH:
        XCH     A,@DE
        XCH     A,@HL+
        XCH     A,@DE
        INCS    E
        NOP
        DECS    B
        BR      EXCH

        RET
```

10.2　データ比較

メモリ・バンク0とMBE・MBSによって選択されたメモリ・バンクとの間で最大16ワードまでのデータの比較を行います。

比較するデータの先頭アドレスをDEレジスタ（メモリ・バンク＝0）とHLレジスタ（メモリ・バンク＝MBE・MBS）で指定し，データ長（ワード数）をBレジスタで指定します。ただし，HLレジスタ，DEレジスタで指定するアドレスは，それぞれ2行にわたらないものとします。

比較した結果，2つのデータが等しいときはリターン・スキップします。



プログラム例

このプログラムは，**第11章　このアプリケーション・プログラムの応用例**には使用されていません。

- **ステップ数**：10ステップ（10バイト）
- **使用するレジスタ**：A, B, DE, HL

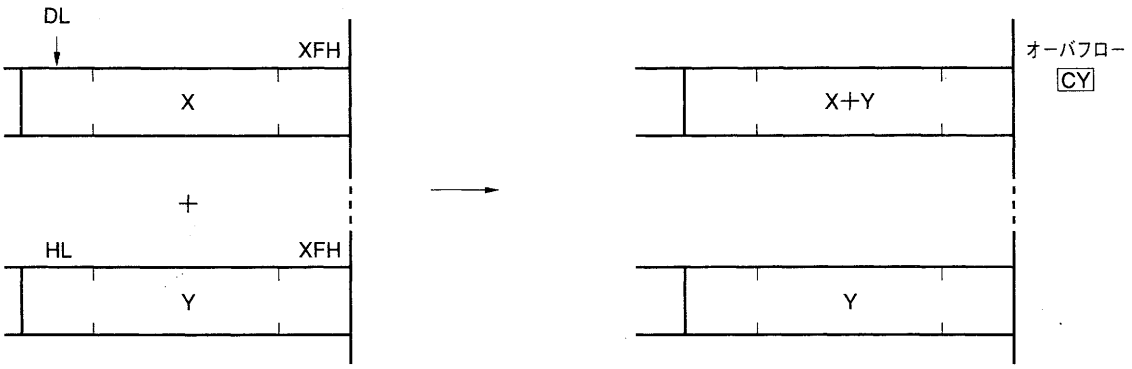
```
COMP    CSEG    INBLOCK

        MOV     A,@DE
        SKE     A,@HL
        RET
        INCS    L
        NOP
        INCS    E
        NOP
        DECS    B
        BR      COMP

        RETS
```

10.3　10進加算

メモリ・バンク0とMBE・MBSによって選択されたメモリ・バンクとの間で，最大16桁（16ワード）までの10進加算を行います。

10進加算するデータの最下位桁アドレスをDLレジスタとHLレジスタで指定し，桁数は10H-Lレジスタの値とします。このとき@DLで示されるデータ・メモリのアドレスは，MBE・MBSの値にかかわらず000H-0FFH（メモリ・バンク0）であり，@HLで示されるデータ・メモリのアドレスは，MBE・MBSの値で示されるメモリ・バンクの範囲となります。

10進加算した結果は，DLレジスタでアドレシングしたデータ・エリアに格納します。また，オーバフローはキャリー・フラグ（CY）に残します。



プログラム例

このプログラムは，**第11章　このアプリケーション・プログラムの応用例**には使用されていません。

・ステップ数：9ステップ（9バイト）
・使用するレジスタ：A, D, HL

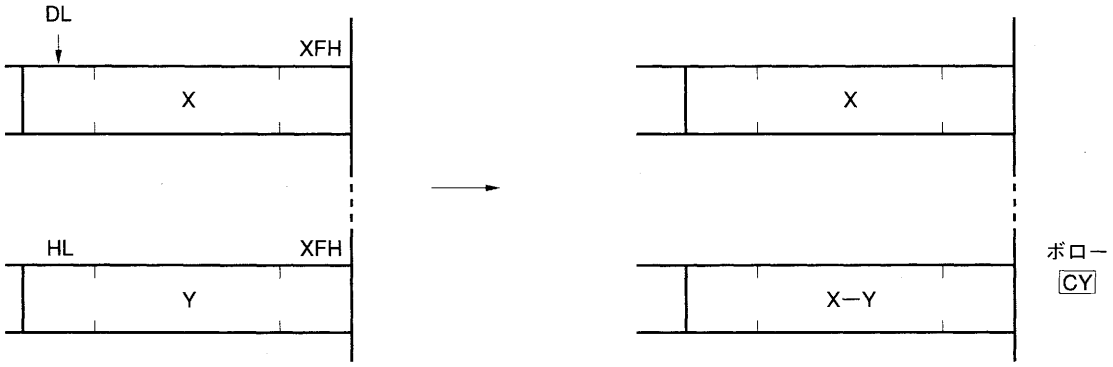
```
DECADD  CSEG    INBLOCK

        CLR1    CY
LOOP:
        MOV     A,@DL
        ADDS    A,#6
        ADDC    A,@HL
        ADDS    A,#0AH
        XCH     A,@DL
        INCS    L
        BR      LOOP

        RET
```

10.4　10進減算

メモリ・バンク0とMBE・MBSによって選択されたメモリ・バンクとの間で，最大16桁（16ワード）までの10進減算を行います。

10進減算の被減数を格納するデータ・エリアの最下位桁アドレスをDLレジスタで指定し，減数を格納するデータ・エリアの最下位桁アドレスをHLレジスタで指定します。桁数は10H-Lレジスタの値とします。このとき@DLで示されるデータ・メモリのアドレスは，MBE・MBSの値にかかわらず000H-0FFH（メモリ・バンク0）であり，@HLで示されるデータ・メモリのアドレスは，MBE・MBSの値で示されるメモリ・バンクの範囲となります。

10進減算した結果は，HLレジスタでアドレシングしたデータ・エリアに格納します。また，ボローはキャリー・フラグ（CY）に残します。



プログラム例

このプログラムは，**第11章　このアプリケーション・プログラムの応用例**には使用されていません。

・**ステップ数**：19ステップ（19バイト）

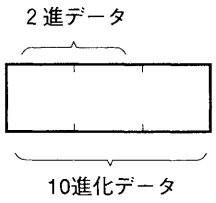
・**使用するレジスタ**：A, D, HL

・**使用するスタック**：2ワード

```
DECSUB  CSEG    INBLOCK

        PUSH    HL
        CLR1    CY
LOOP0:
        MOV     A,@DL
        SUBC    A,@HL
        ADDS    A,#0AH
        XCH     A,@HL
        INCS    L
        BR      LOOP0

        POP     HL
        SKT     CY
        RET

        CLR1    CY
LOOP1:
        MOV     A,#0H
        SUBC    A,@HL
        ADDS    A,#0AH
        XCH     A,@HL
        INCS    L
        BR      LOOP1

        RET
```

10.5　2進10進変換

データ・メモリ内の2進8ビット・データを2進化10進数に変換し，変換結果を変換前にデータがあったデータ・メモリとその上位1ワードに格納します。



2進8ビット・データは，下位桁をX，上位桁をYとすると

$X_{(2進)}+Y_{(2進)}$（10＋6）

となり，

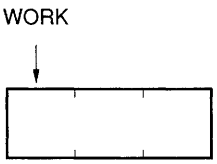
$(X_{(2進)}+Y_{(2進)}\times 10)+Y_{(2進)}\times 6$

と表すことができます。

ここで，XとYは2進数なので，まず（X＋Y×10）の部分のXとYを10進変換し，次にYの値の回数だけ6を加算します。このとき，加算するたびに10進補正を行います。

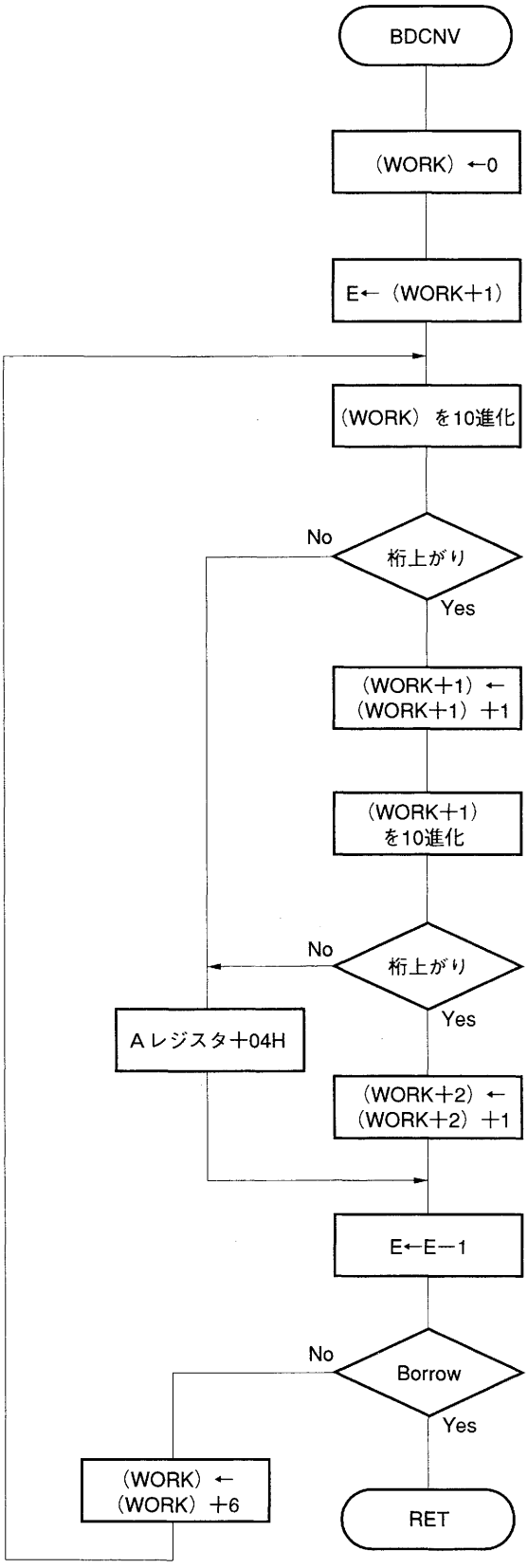
以上の処理によって，もとのデータを10進変換した値を得ることができます。

（1）使用するRAM



12ビット　：下位8ビットのデータを2進化10進数に変換した値を格納する。

（2）フロー・チャート



；1桁目を10進化し，桁上がりが起きたときは，
2桁目をインクリメント

；2桁目を10進化し，桁上がりが起きたときは，
3桁目をインクリメント

；2桁目の値の回数だけ6を加算

（3）プログラム例

このプログラムは，**第11章　このアプリケーション・プログラムの応用例**には使用されていません。

・ステップ数：24ステップ（32バイト）

・使用するレジスタ：A, E, HL

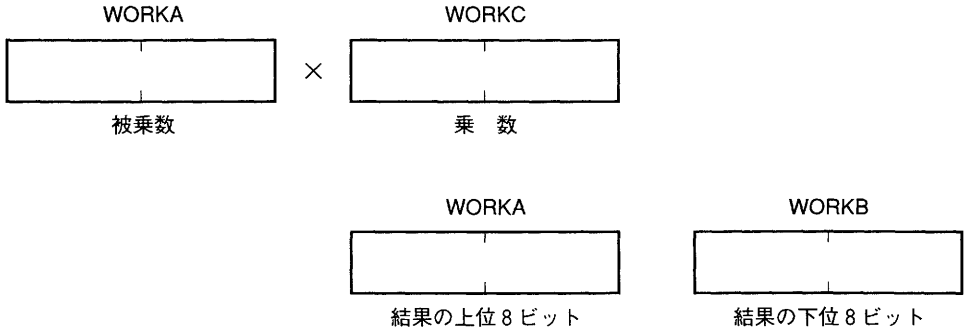
```
        PUBLIC  WORK

BDDATA  DSEG    AT      10H

WORK:   DS      3H
BDCNV   CSEG    INBLOCK

        MOV     A,#0H
        MOV     WORK + 2,A
        MOV     A,WORK + 1
        XCH     A,E

        MOV     A,WORK

BD:
        MOV     HL,#WORK            ;10進補正（1桁目）
        ADDS    A,#6H
        BR      ADD10

        MOV     @HL,A               ;2桁目をインクリメント
        INCS    L
        MOV     A,#7H
        ADDS    A,@HL               ;10進補正
        BR      ADD10

        INCS    WORK + 2            ;3桁目をインクリメント
CTLOOP:
        MOV     @HL,A               ;2桁目の値の回数，6を加算
        DECS    E
        BR      BDX2
        RET
BDX2:
        MOV     A,WORK
        ADDS    A,#6H
        BR      BD
ADD10:
        ADDS    A,#0AH
        NOP
        BR      CTLOOP
```

10.6　8ビット乗算

データ・メモリ内の8ビット・ワーク・エリアWORKAを被乗数とし，8ビット・ワーク・エリアWORKCを乗数として，8ビットの2進乗算を行います。

演算結果は，上位8ビットをWORKAに格納し，下位8ビットを8ビット・ワーク・エリアWORKBに格納します。

ただし，WORKA, WORKBは，下位アドレスから連続し，そのメモリ・バンクは，MBE・MBSによって選択されたメモリ・バンクとします。また，WORKCのメモリ・バンクは，0とします。



プログラム例

このプログラムは，**第11章　このアプリケーション・プログラムの応用例**には使用されていません。

- **ステップ数**：38ステップ（50バイト）
- **使用するレジスタ**：XA, BC, DE, HL
- **使用するスタック**：2ワード

```
        PUBLIC  WORKA,WORKB,WORKC

DMLT8   DSEG    AT      20H

WORKC:  DS      2H
WORKB:  DS      2H
WORKA:  DS      2H

MULT8   CSEG    INBLOCK

        MOV     XA,#00H
        MOV     WORKB,XA
        MOV     C,#1H
LOOP:
        MOV     B,#3H
        MOV     HL,#WORKB
        MOV     A,#0H
```

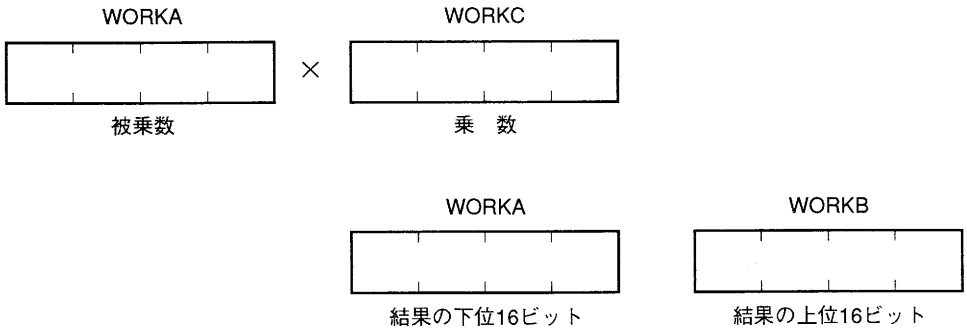
```
SHLOOP:
        XCH     A,@HL+
        NOP
        DECS    B
        BR      SHLOOP

        XCH     A,B
FIGURE:
        DECS    B
        BR      ADD

        DECS    C
        BR      LOOP

        RET
ADD:
        PUSH    BC
        CLR1    CY
        MOV     B,#1H
        MOV     DE,#WORKC
        MOV     HL,#WORKB
ADLOOP:
        MOV     A,@DE
        ADDC    A,@HL
        XCH     A,@HL+
        INCS    L
        NOP
        INCS    E
        NOP
        DECS    B
        BR      ADLOOP

        POP     BC
        MOV     HL,#WORKA
        NOT1    CY
        SKT     CY
INCA:
        INCS    @HL
        BR      FIGURE

        INCS    L
        NOP
        BR      INCA
```

10.7　16ビット乗算

データ・メモリ内の16ビット・ワーク・エリアWORKAを被乗数とし，16ビット・ワーク・エリアWORKCを乗数として，16ビットの２進乗数を行います。

演算結果は，上位16ビットをWORKBに格納し，下位16ビットを16ビット・ワーク・エリアWORKAに格納します。

ただし，WORKA, WORKBは，下位アドレスから連続し，そのメモリ・バンクは，MBE・MBSによって選択されたメモリ・バンクとします。また，WORKCのメモリ・バンクは，０とします。



プログラム例

このプログラムは，**第11章　このアプリケーション・プログラムの応用例**には使用されていません。

- **ステップ数**：39ステップ（51バイト）
- **使用するレジスタ**：XA, BC, DE, HL
- **使用するスタック**：２ワード

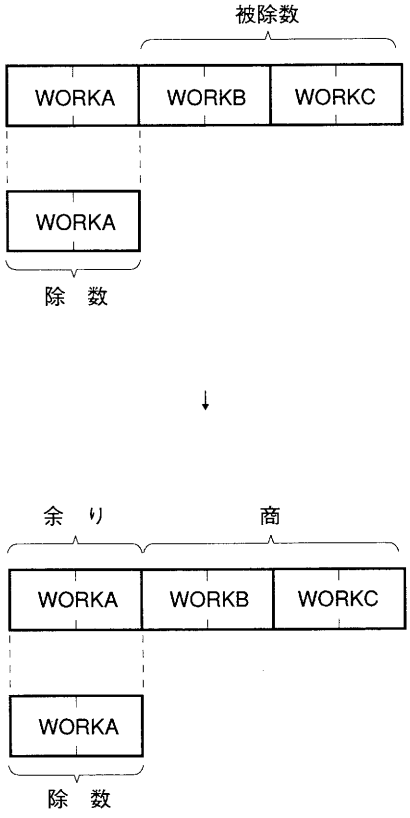
```
        PUBLIC  WORKA,WORKB,WORKC

DMLT16  DSEG    AT      20H

WORKC:  DS      4H
WORKB:  DS      4H
WORKA:  DS      4H

MULT16  CSEG    INBLOCK

        MOV     XA,#00H
        MOV     WORKB,XA
        MOV     WORKB+2,XA
        MOV     C,#3H
LOOP:
        MOV     B,#7H
        MOV     HL,#WORKB
        MOV     A,#0H
SHLOOP:
        XCH     A, @HL+
        NOP
        DECS    B
        BR      SHLOOP

        XCH     A,B
FIGURE:
        DECS    B
        BR      ADD
        DECS    C
        BR      LOOP
        RET
ADD:
        PUSH    BC
        CLR1    CY
        MOV     B,#3H
        MOV     DE,#WORKC
        MOV     HL,#WORKB
ADLOOP:
        MOV     A,@DE
        ADDC    A,@HL
        XCH     A,@HL+
        NOP
        INCS    E
        NOP
        DECS    B
        BR      ADLOOP

        POP     BC
        MOV     HL,#WORKA
        NOT1    CY
        SKT     CY
INCA:
        INCS    @HL
        BR      FIGURE

        INCS    L
        NOP
        BR      INCA
```

10.8　2進除算（16ビット÷8ビット）

データ・メモリ内の8ビット・ワーク・エリアWORKBおよびWORKCを被除数とし，8ビット・ワーク・エリアWORKDを除数として，2進除算を行います。

演算結果は，上位8ビットをWORKCに格納し，下位8ビットをWORKBに格納します。また，8ビット・ワーク・エリアWORKAに余りを格納します。

ただし，WORKA, WORKB, WORKCは同一のロウ・アドレス上にあり，下位アドレスから連続しているものとします。またWORKDのCOLUMNアドレスはWORKAと同じものとし，いずれのワーク・エリアもメモリ・バンク0とします。



プログラム例

このプログラムは，**第11章　このアプリケーション・プログラムの応用例**には使用されていません。

・ステップ数：42ステップ（59バイト）

・使用するレジスタ：XA, B, D, HL

```
        PUBLIC  WORKA,WORKB,WORKC,WORKD

DIV8D0  DSEG    AT      20H

WORKA:  DS      2H
WORKB:  DS      2H
WORKC:  DS      2H

DIV8D1  DSEG    AT      30H

WORKD:  DS      2H

DIV8    CSEG    INBLOCK

        CLR1    MBE
        MOV     B,#0H
        MOV     XA,#00H
        MOV     WORKA,XA
LOOP:
        MOV     HL,#WORKA
        SET1    CY
        SKT     (WORKA + 5).3
        CLR1    CY
ROTAT:
        MOV     A,@HL
        ADDC    A,@HL
        MOV     @HL,A
        INCS    L
        NOP
        SKE     L,#(WORKA + 6) AND 0FH
        BR      ROTAT

        CLR1    CY

        MOV     HL,#WORKD
        MOV     D,#WORKA SHR 4
```

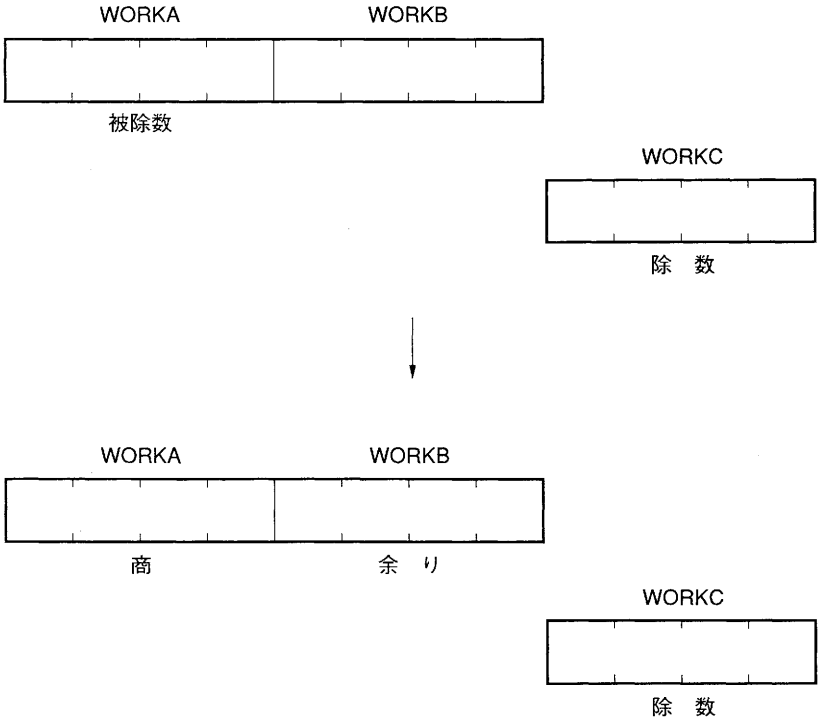
```
SUB:
        MOV     A,@DL
        SUBC    A,@HL
        XCH     A,@DL
        INCS    L
        SKE     L,#(WORKA + 2) AND 0FH
        BR      SUB

        SKT     CY
        BR      SETB0

        SKT     WORKB.0
        BR      ADD

FIGCNT:
        INCS    B
        BR      LOOP
        RET

ADD:
        CLR1    CY
        MOV     L,#WORKA AND 0FH
ADDLP:
        MOV     A,@DL
        ADDC    A,@HL
        XCH     A,@DL
        INCS    L
        SKE     L,#(WORKA + 2) AND 0FH
        BR      ADDLP
        BR      FIGCNT

SETB0:
        SET1    WORKB.0
        BR      FIGCNT
```

10.9　2進除算（16ビット÷16ビット）

データ・メモリ内の16ビット・ワーク・エリアWORKAを被除数とし，16ビット・ワーク・エリアWORKCを除数として，2進除算を行います。

演算結果は，WORKAに商を格納し，16ビット・ワーク・エリアWORKBに余りを格納します。

ただし，WORKA, WORKBは，下位アドレスから連続しているものとし，メモリ・バンクは0とします。また，WORKCのメモリ・バンクは，MBE・MBSによって選択されたメモリ・バンクとします。



プログラム例

このプログラムは，**第11章　このアプリケーション・プログラムの応用例**には使用されていません。

・ステップ数：49ステップ（63バイト）

・使用するレジスタ：A, B, DE, HL

```
        PUBLIC  WORKA,WORKB,WORKC

DDIV16  DSEG    AT      20H

WORKA:  DS      4H
WORKB:  DS      4H
WORKC:  DS      4H

DIV16   CSEG    INBLOCK

        MOV     DE,#WORKB
CLRB:
        MOV     A,#0H
        XCH     A,@DE
        INCS    E
        NOP

        SKE     E,#(WORKB + 4) AND 0FH
        BR      CLRB

        MOV     B,#0CH
LOOP:
        MOV     DE,#WORKA
        MOV     A,#0H
SHFT:
        XCH     A,@DE
        INCS    E
        NOP
        SKE     E,#(WORKA + 8) AND 0FH
        BR      SHFT
SUB:
        MOV     DE,#WORKB
        MOV     HL,#WORKC
        CLR1    CY
SLOOP:
        MOV     A,@DE
        SUBC    A,@HL
        XCH     A,@DE
        INCS    L
        NOP
        INCS    E
        NOP
        SKE     L,#(WORKA + 4) AND 0FH
        BR      SLOOP
```

```
        NOT1    CY
        SKT     CY
        BR      DO

        MOV     DE,#WORKA
        XCH     A,@DE
        ADDS    A,#1H
        XCH     A,@DE
        BR      SUB
DO:
        MOV     DE,#WORKB
        MOV     HL,#WORKC
ADD:
        MOV     A,@DE
        ADDC    A,@HL
        XCH     A,@DE
        INCS    L
        NOP
        INCS    E
        NOP
        SKE     E,#(WORKB + 4) AND 0FH
        BR      ADD
        INCS    B
        BR      LOOP

        RET
```

第11章　このアプリケーション・プログラムの応用例

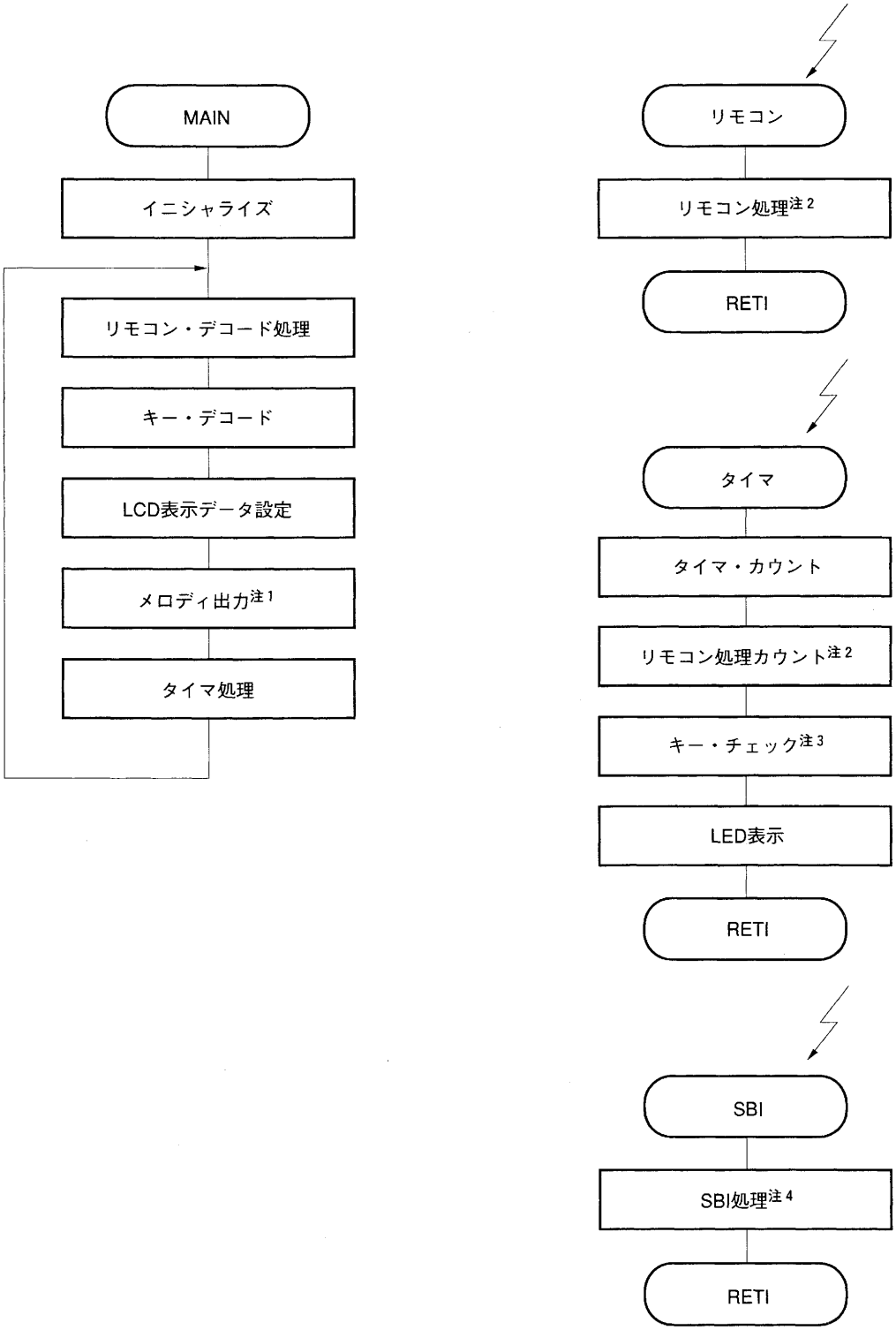
この章では，前章までのアプリケーション・プログラムを使用して作成したシステム・プログラムの例を示します。前章までのアプリケーション・プログラムのうち，使用したものを次に示します。

3.3　　リモコン信号受信応用
4.2.1　メロディ出力
8.1　　SBIモードによる応用例
第9章　キー入力サブルーチン

11.1　プログラム概要

このプログラムは，鍵盤に見立てたモメンタリ・キーまたはリモコン・キーにより，音階データとキーを押している時間による音長データを入力し，自動的に演奏を行います。

次に，全体のフロー・チャートを示します。



注1．4.2.1　メロディ出力のアプリケーション・ソフトを使用。

2．3.3　リモコン信号受信応用のアプリケーション・ソフトを使用。

3．第9章　キー入力サブルーチンのアプリケーション・ソフトを使用。

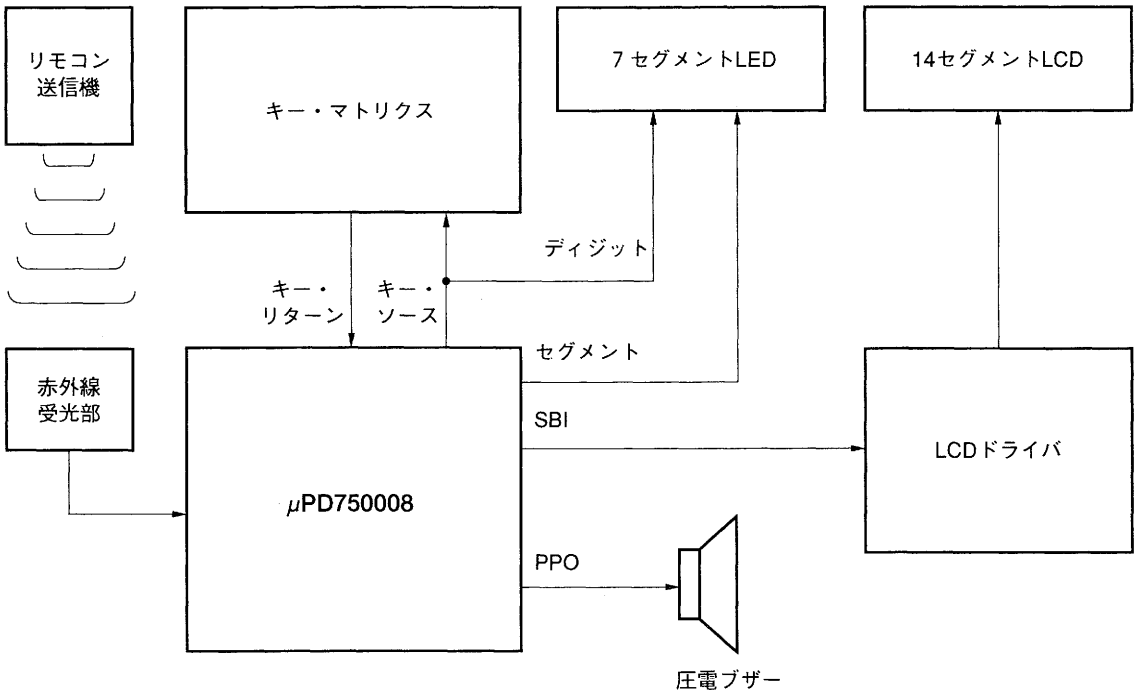
4．8.1　SBIモードによる応用例のアプリケーション・ソフトを使用。

11.2　システム構成

このシステムは，デバイスとしてμPD750008を使用し，キー・マトリクスまたはリモコン入力により，14セグメントLCDドライバ，7セグメントLEDドライバ，および圧電ブザーへの出力を制御します。

図11－1にシステム構成図を示します。

図11－1　システム構成図



キー・マトリクス　：MODE, UP, DOWN, END, WAIT，音階の各キーが配置されています。

赤外線受光部　：NECフォーマットの赤外線コードを受光します。
MODE, UP, DOWN, END, WAIT，音階の各キーがコード化されています。

7セグメントLED　：4桁の7セグメントLEDに“PROG”，“PLAY”，“PAUS”のモード，および100 ms-900 msまでの音長を表示します。ディジット信号は，キー・マトリクスのキー・ソース信号と共用します。

LCDドライバ　：SBIフォーマットのシリアル・データにより，8桁の14セグメントLCDに演奏データの内容，または出力中の音階を表示します。

圧電ブザー　：押されたキー，または演奏データを出力します。

11.3　ポート割り付け

表11－1　μPD750008のポート割り付け（1/2）

端子番号	端子名	入出力	兼用端子	機能	リセット時	アクティブ値	初期設定値
15	P00	入力	INT4	未使用（GNDに接続）	入力	－	入
14	P01	入出力	$\overline{SCK}$	シリアル・シフト・クロック		L	H
13	P02	入出力	SO/SB0	LCDドライバとのシリアル・バス・ライン		－	H
12	P03	入出力	SI/SB1	未使用（GNDに接続）		－	入
19	P10	入力	INT0	リモコン入力	入力	L	入
18	P11		INT1	未使用（GNDに接続）		－	入
17	P12		INT2	未使用（GNDに接続）		－	入
16	P13		TI0	未使用（GNDに接続）		－	入
25	P20	入出力	PTO0	圧電ブザー	入力	L	H
24	P21		PTO1	未使用（GNDに接続）		－	入
23	P22		PCL	未使用（GNDに接続）		－	入
22	P23		BUZ	未使用（GNDに接続）		－	入
9	P30	入出力	－	ディジット出力（LED7）／キー・ストローブ信号	入力	L	H
8	P31		－	ディジット出力（LED6）／キー・ストローブ信号		L	H
7	P32		－	ディジット出力（LED5）／キー・ストローブ信号		L	H
6	P33		－	ディジット出力（LED4）／キー・ストローブ信号		L	H
41	P40	入出力	－	セグメント出力（D.P.）	ハイ・レベル（プルアップ抵抗内蔵時）またはハイ・インピーダンス	L	H
40	P41			セグメント出力（g）		L	H
39	P42			セグメント出力（f）		L	H
38	P43			セグメント出力（e）		L	H
37	P50	入出力	－	セグメント出力（d）	ハイ・レベル（プルアップ抵抗内蔵時）またはハイ・インピーダンス	L	H
36	P51			セグメント出力（c）		L	H
35	P52			セグメント出力（b）		L	H
34	P53			セグメント出力（a）		L	H
33	P60	入出力	KR0	キー入力端子	入力	L	H
32	P61		KR1	キー入力端子		L	H
31	P62		KR2	キー入力端子		L	H
30	P63		KR3	キー入力端子		L	H
29	P70	入出力	KR4	キー入力端子	入力	L	H
28	P71		KR5	キー入力端子		L	H
27	P72		KR6	キー入力端子		L	H
26	P73		KR7	キー入力端子		L	H
11	P80	入出力	－	未使用（GNDに接続）	入力	－	入
10	P81		－	未使用（GNDに接続）		－	入

表11－1　μPD750008のポート割り付け（2/2）

端子番号	端子名	入出力	兼用端子	機能	リセット時	アクティブ値	初期設定値
4	X1, X2	入力	－	メイン・システム・クロック発振用	入力	－	－
5				クリスタル／セラミック接続端子		－	－
1	XT1	入力	－	未使用（GNDに接続）	入力	－	入
2	XT2	－		未使用（GNDに接続）	－	－	入
3	$\overline{\text{RESET}}$	入力	－	システム・リセット入力端子	－	L	－
20	IC	－	－	未使用（GNDに接続）	－	－	入
21	V_{DD}	－	－	未使用（GNDに接続）	－	－	入
42	V_{SS}	－	－	未使用（GNDに接続）	－	－	入

11.4　状態遷移表

表11－2　状態遷移表

モード ＼ キー入力		モード・キー		音階キー		ENDキー		UPキー	DOWNキー
		キー・オンが2秒以上	キー・オンが2秒未満	キー・オン	キー・オフ	キー・オン	キー・オフ		
PROG	機能	PAUSモードに切り替える。	データの入力を，先頭のデータから行うか，終わりのデータから行うかを切り替える。	音階キーを判別し，音階を出力する。また，音長をカウントする。	無音を出力し，音長カウントを停止する。音階，音長データを演奏アドレスに記憶し，次のアドレスへ進む。	無音を出力し，音長カウンタをカウントする。	音長カウントを停止し，リピート・コードと音長データ（リピート時間）を演奏アドレスに記憶する。アドレスは変化しない。	演奏アドレスを+1し，キー・オンの間は演奏データの音階を出力する。キー・オフしたら，無音を出力する。	演奏アドレスを−1し，キー・オンの間は演奏データの音階を出力する。キー・オフしたら，無音を出力する。
	LED表示	PAUS	Prog	Prog	Prog	Prog	Prog	Prog	Prog
	LCD表示	___END_	例：008-END_	例：000-DO__	例：001-RE__	例：127-WAIT	例：127-END_	例：004-SO__	例：006-SI_＋
PLAY	機能	PROGモードに切り替える。データの入力は演奏アドレスの続きから行う。	自動演奏を中断し，PAUSモードに切り替える。	受け付けない。		演奏アドレスを000番地に戻し，自動演奏を再開する。	受け付けない。	基本時間を＋1する。LEDは基本時間を1秒間表示したあと，モード表示に戻る。	基本時間を−1する。LEDは基本時間を1秒間表示したあと，モード表示に戻る。
	LED表示	Prog	PAUS	PLAY		PLAY		例：3_{00}→PLAY	例：1_{00}→PLAY
	LCD表示	例：007-DO_＋	___END_	例：006-SI__		例：000-DO__		例：000-DO__	
PAUS	機能	PROGモードに切り替える。データの入力は演奏アドレスの続きから行う。	自動演奏を再開し，PLAYモードに切り替える。	音階キーを判別し，音階を出力する。音長はカウントしない。	無音を出力する。データは記憶しない。また，アドレスも変化しない。	演奏アドレスを000番地に戻し，LCDは1秒間アドレスを表示する。	受け付けない。	基本時間を＋1する。LEDは基本時間を1秒間表示したあと，モード表示に戻る。	基本時間を−1する。LEDは基本時間を1秒間表示したあと，モード表示に戻る。
	LED表示	Prog	PLAY	PAUS	PAUS	PAUS		例：3_{00}→PAUS	例：1_{00}→PAUS
	LCD表示	例：003-FA__	例：004-SO__	例：___RA#1	___END_	例：000-END→（1秒後）→___END_		___END_	___END_

備考　リセット・スタート時は，PROGモード。

11.5　キーごとの機能説明

すべてのキーは，どのキーも押されていないときに押された場合，有効になります。別のキーが押されているときは，そのキーが離されるまで，ほかのキーを押しても（多重押し）有効とはなりません。

キー・マトリクス上のキーが押されていないときは，リモコン入力を受け付けます。

キー・マトリクスの構成については，**11.7　ハードウエア構成**を参照してください。

次に，各キーの説明を行います。

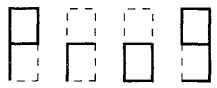
11.5.1　MODEキー

2秒以上キーを押すと，PAUSモードとPLAYモードを交互に切り替えます。

PROGモード時は，音階キーによる演奏データの入力が可能となります。また，MODEキーを2秒以下の間押すことにより，データの入力を先頭（演奏用データ・メモリの先頭）のデータから行うか，終わり（アドレスFnH）のデータから行うかを切り替えることができます。

PLAYモードまたはPAUSモード時は，PROGモードで入力した演奏データにより，自動演奏が可能になります。また，MODEキーの二重押しにより，演奏をスタートするか，ポーズするかを切り替えることができます。

次に，MODEキーの動作と表示を示します。

モード	MODEキー		7セグメントLED (LED4, LED5, LED6, LED7)表示
	2秒以上押し	2秒以下押し	
PROGモード	PAUSモードに切り替える	データの入力を始めから行うか終わりから行うかを切り替える	
PLAYモード	PROGモードに切り替える（データ入力は終わりから）	自動演奏を中断しPAUSモードとなる	
PAUSモード（ポーズ）		自動演奏を再開しPLAYモードとなる	

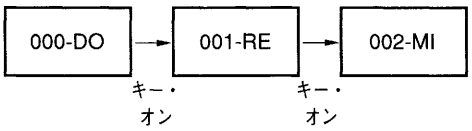
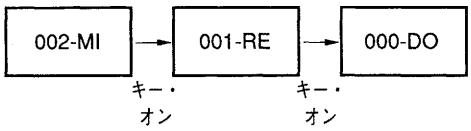
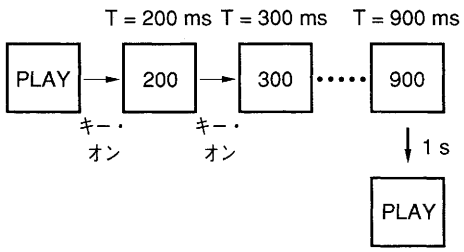
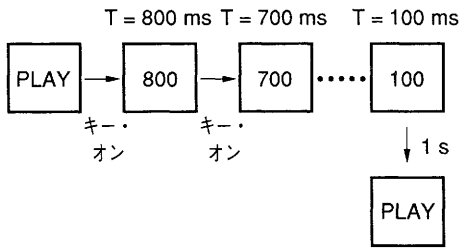
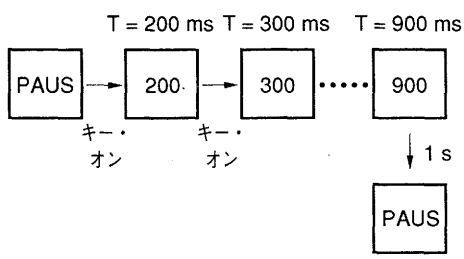
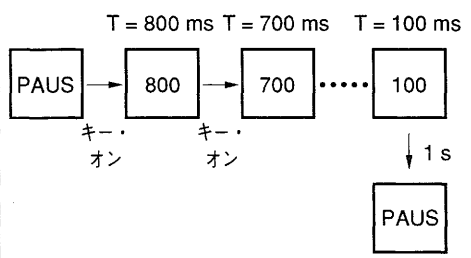
リセット・スタート時はPROGモードとなります。

11.5.2　UP/DOWNキー

PROGモード時は，キーを押すごとに演奏データのアドレスをアップ／ダウンし，キーを押している間は，そのデータの音階をLCDに表示出力します。

PLAYモードまたはPAUSモード時は，キーを押すごとに基本時間Tをアップ／ダウンしてLCDに表示し，表示が最大／最小になると，約1秒後にモード表示に戻ります。

次に，UP/DOWNキーの動作および表示を示します。

<table>
<tr><th>モード</th><th>UPキー</th><th>DOWNキー</th></tr>
<tr><td>PROGモード
演奏データの編集に使用</td><td>データ内容（DO RE MI）
14セグメントLCD
 
</td><td>データ内容（DO RE MI）
14セグメントLCD
 
</td></tr>
<tr><td>PLAYモード
基本時間の設定に使用</td><td>7セグメントLED　(Max.)
 
</td><td>7セグメントLED　(Min.)
 
</td></tr>
<tr><td>PAUSモード
基本時間の設定に使用</td><td>7セグメントLED　(Max.)
 
</td><td>7セグメントLED　(Max.)
 
</td></tr>
</table>

リセット・スタート時はT = 200 msとなります。

11.5.3　音階キー

キー・マトリクス上の，次の15のキーを音階キーといいます。

DO, RE, MI, FA, SO, RA＋, SI＋, DO＋, DO#, RE#, FA#, SO#, RA＋#, DO＋#, WAIT（無音）

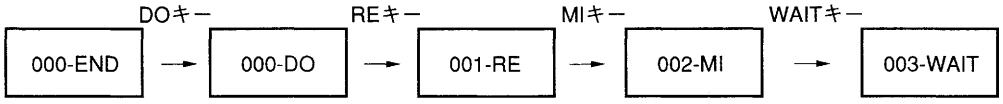
PROGモード，またはPAUSモード時のみ有効です。

PROGモードでは，キーを押している間，各キーに対応した音階または無音を出力し，また出力されている音階を表示します。キーが離されると，演奏用メモリ・エリアに出力した音階と音長のデータを書き込み，次のアドレスで出力される音階データを表示します。

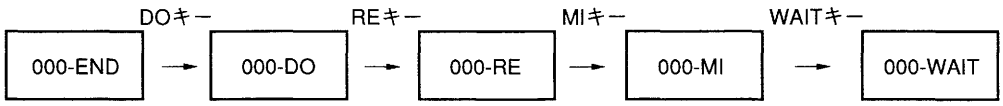
PAUSモードでは，キーを押している間，各キーに対応した音階または無音を出力し，出力されている音階を表示します。

次に，動作例を示します。

PROGモードの場合（14セグメントLCD）



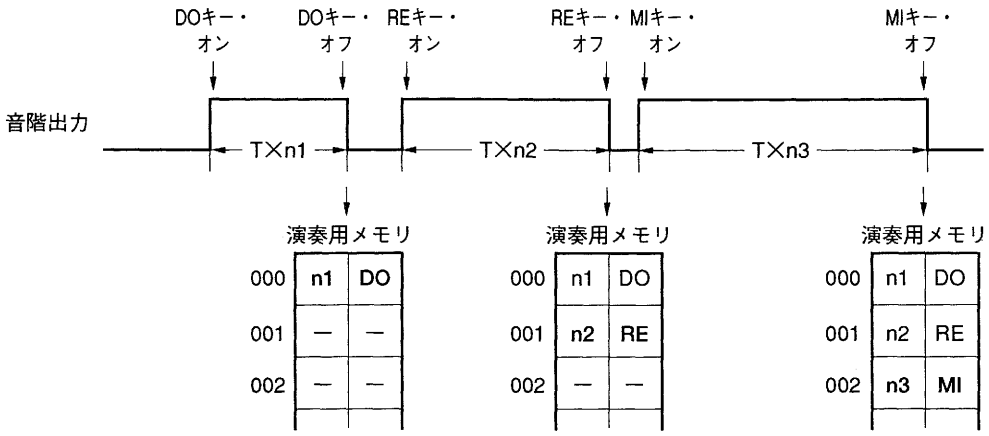
PAUSモードの場合（14セグメントLCD）



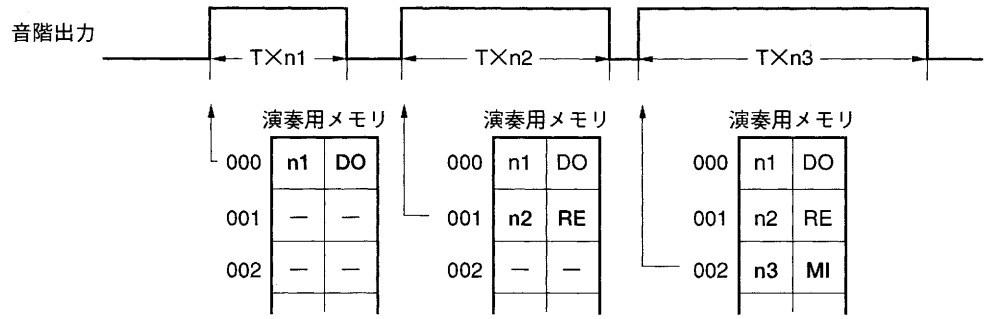
音階キーを押すことにより8ビットのデータを生成し，順に演奏用メモリ・エリアに読み出します。データの下位4ビットが音階データ，上位4ビットが音長データで，音長データは基本時間（T）の倍数になります。

音長データがF×Hのときは，演奏を初めから繰り返すリピート動作となります。

演奏データ・メモリには最大128個までのデータが記憶できます。



逆に，演奏用メモリ・エリアからデータを取り出し，再度演奏を行うこともできます。



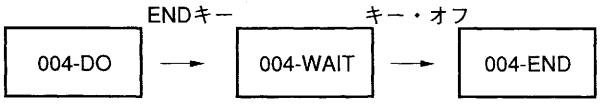
11.5.4　ENDキー

PROGモード時は，キーを押すごとに，リピート・コード（FnH）を書き込みます。nはキーを押している時間で決定します。

PLAYモードまたはPAUSモード時は，演奏データのアドレスをスタートに戻します。

次に，PROGモード時の動作例を示します。

14セグメントLCD



11.5.5　その他のキー

キーが押されても，何の動作も行いません。

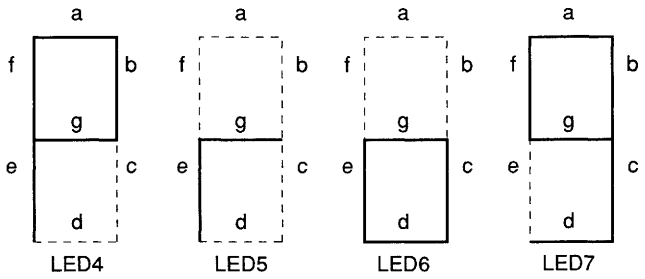
11.6　表示説明

11.6.1　LED表示

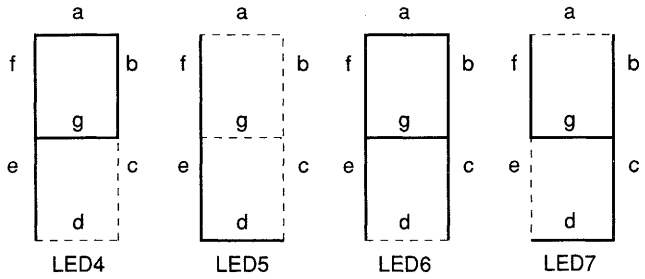
モード表示，および基準時間Tを設定するときの時間表示に使用します。

次に，表示パターンの例を示します。

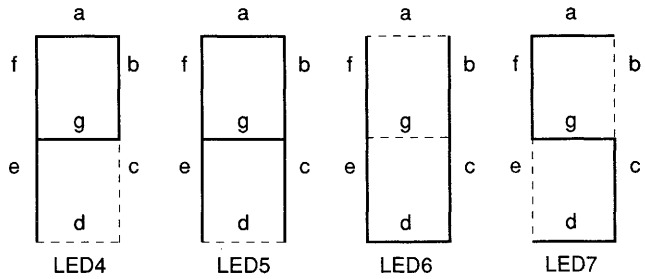
PROGモード表示



PLAYモード表示

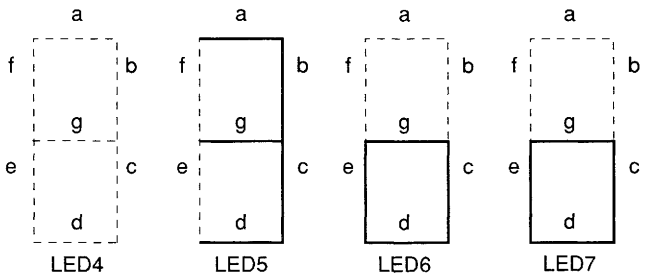


PAUSモード表示

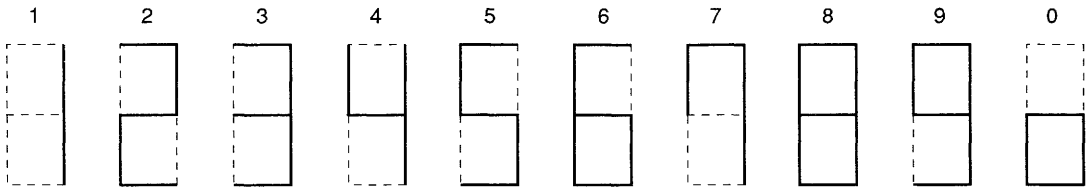


基本時間（T）表示

（300 ms設定）



数字表示パターン



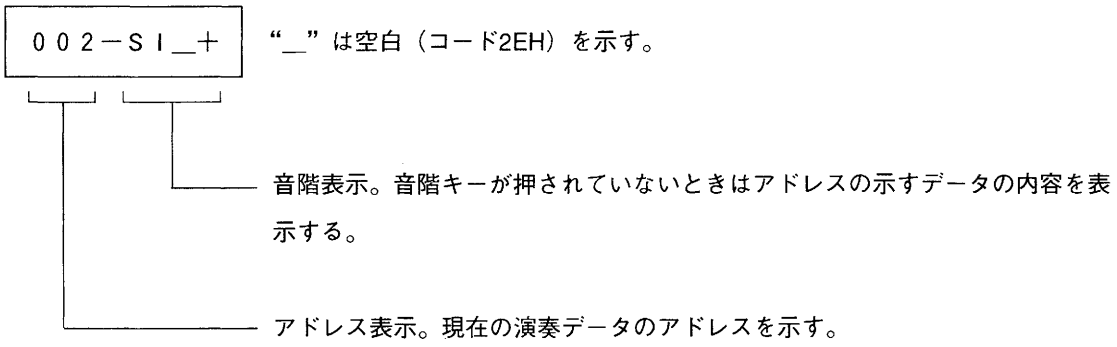
11.6.2　LCD表示

演奏データ，およびアドレスの表示に使用します。

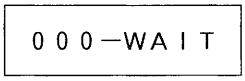
LCD表示は，LCDコントローラ／ドライバにSBIモードでキャラクタ・コードを転送することにより，行います。

次に，表示例を示します。

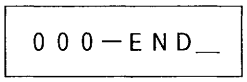
SI＋キー・オン



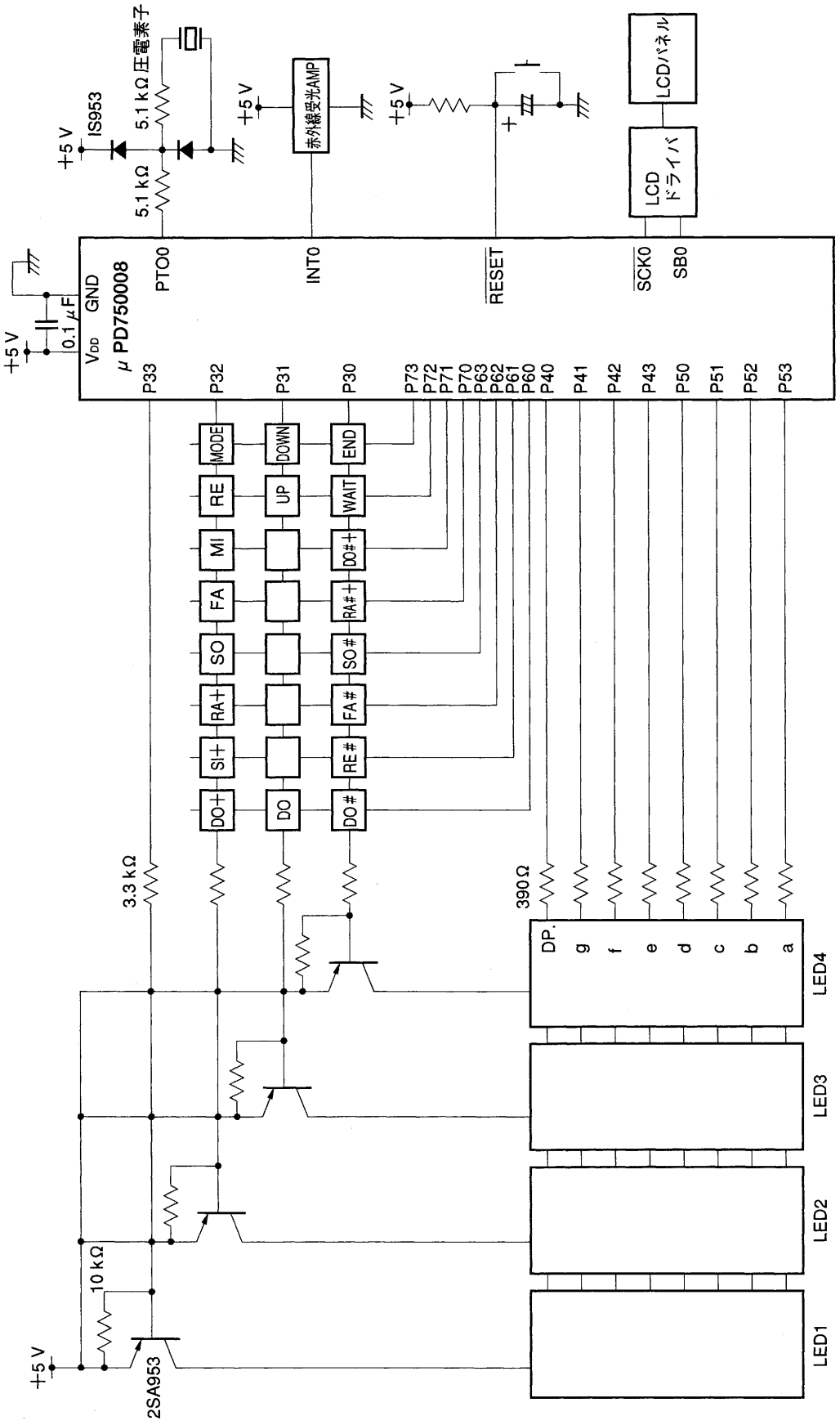
WAITキー・オン



演奏データがリピート・コードのとき



11.7　ハードウエア構成



11.8　アプリケーション・プログラム

```
;*************************************************************
;***                                                       ***
;***            アプリケーション・プログラム               ***
;***                                                       ***
;*************************************************************

        NAME    MAIN                    ;ﾓｼﾞｭｰﾙ名
        STKLN   40                      ;ｽﾀｯｸ･ｻｲｽﾞ指定

    VENT0   MBE=0,RBE=0,INIT            ;RESET
    VENT1   MBE=0,RBE=1,INTBT           ;INTBT/INT4
    VENT2   MBE=0,RBE=1,INT0            ;INT0
    VENT4   MBE=0,RBE=1,SBI             ;SBI

DSEG0   DSEG    0       AT  20H         ;ﾒﾓﾘ･ﾊﾞﾝｸ0,ｱﾄﾞﾚｽ20Hから格納

KEYCODE:        DS      2               ;ｷｰ･ｿｰｽ,ｷｰ･ﾘﾀｰﾝをｺｰﾄﾞ化
KEYR:           DS      2               ;ｷｰ･ﾘﾀｰﾝ
KEYRB:          DS      2
MUS:            DS      2               ;演奏ﾃﾞｰﾀ･ｱﾄﾞﾚｽ
KEYTIM:         DS      2               ;ｷｰ･ｵﾝ時間ｶｳﾝﾄ用ﾀｲﾏ
LCD:            DS      16              ;LCD表示用ｴﾘｱ
WORK:           DS      2               ;ﾘﾓｺﾝ受信したﾃﾞｰﾀを格納
RMDATA:         DS      2               ;有効なﾘﾓｺﾝ･ﾃﾞｰﾀ･ｺｰﾄﾞを格納
KEYS:           DS      1               ;ｷｰ･ｿｰｽ
KEYSB:          DS      1
KEYCATT:        DS      1               ;ｷｰのﾁｬﾀﾘﾝｸﾞ･ｶｳﾝﾀ
KEYFLG:         DS      1               ;ﾌﾗｸﾞ
TONE:           DS      1               ;音階ﾃﾞｰﾀ
TONEFLG:        DS      1               ;ﾌﾗｸﾞ
B_TIM:          DS      1               ;ﾍﾞｰｽ･ﾀｲﾏ(1.95msｶｳﾝﾄ･ｱｯﾌﾟ)
B_TIMC:         DS      1               ;50ms,100msｶｳﾝﾄ･ﾀｲﾏ(9.75msｶｳﾝﾄ･ｱｯﾌﾟ)
DISPTIM:        DS      1               ;LED表示用ﾀｲﾏ
TONETIM:        DS      1               ;音長ﾃﾞｰﾀ
T_TIM:          DS      1               ;基本時間
T_TIMC:         DS      1               ;基本時間用ｶｳﾝﾀ
RPTIM:          DS      1               ;ﾘﾓｺﾝ･ﾘﾋﾟｰﾄ用ﾀｲﾏ
RPCODE:         DS      1               ;ﾘﾓｺﾝ･ﾘﾋﾟｰﾄ･ｺｰﾄﾞ･ｶｳﾝﾀ
LEDDIG:         DS      1               ;LED表示桁番号
MODE:           DS      1               ;ﾓｰﾄﾞ設定用  0:PROG,1:PLAY,2:PAUS
DEC:            DS      3               ;HEX  →  DEC
LCDTIM:         DS      1               ;1sｶｳﾝﾀ
LCDFLG:         DS      1               ;ﾌﾗｸﾞ
MODESBI:        DS      1               ;どのﾃﾞｰﾀを転送中かを示す
LDCODE:         DS      1               ;ﾘｰﾀﾞ･ｺｰﾄﾞ･ｽｷｬﾝ用ｶｳﾝﾀ
```

```
MODEP:          DS      1           ;どのｴｯｼﾞを検出中かを示す
RMFLG:          DS      1           ;ﾌﾗｸﾞ
KEYONF          EQU     KEYFLG.3    ;1ならばｷｰ･ｵﾝ
KEYON2SF        EQU     KEYFLG.2    ;1ならば2秒以上ｷｰを押し続けている
HTKEYONF        EQU     KEYFLG.1    ;1ならば本体ｷｰ･ｵﾝ
KEYCHKF         EQU     KEYFLG.0    ;1ならばｷｰ･ｽｷｬﾝ終了

TONE_F          EQU     TONEFLG.0   ;1ならば音階出力を許可

TM0_F           EQU     TONEFLG.1   ;1ならばﾀｲﾏ･ｶｳﾝﾄ･ｽﾀｰﾄ

REFRES_F        EQU     LCDFLG.0    ;1ならばLCDをﾘﾌﾚｯｼｭ

REP_F           EQU     RMFLG.0     ;1ならばﾘﾓｺﾝ･ﾘﾋﾟｰﾄ用ﾀｲﾏ･ｶｳﾝﾄをｽﾀｰﾄ

;*****      LED文字ﾃﾞｰﾀ      *****

LED_P           EQU     00110001B   ;"P"
LED_R           EQU     11110101B   ;"R"
LED_0           EQU     11000101B   ;"0"
LED_G           EQU     00001001B   ;"G"
LED_L           EQU     11100011B   ;"L"
LED_A           EQU     00010001B   ;"A"
LED_Y           EQU     10001001B   ;"Y"
LED_U           EQU     10000011B   ;"U"
LED_S           EQU     01001001B   ;"S"

LED_0           EQU     11000101B   ;"0"
LED__           EQU     11111111B   ;"_"

;*****      KEYRﾃﾞｰﾀ     *****

KEYR_1          EQU     11111110B
KEYR_2          EQU     11111101B
KEYR_3          EQU     11111011B
KEYR_4          EQU     11110111B
KEYR_5          EQU     11101111B
KEYR_6          EQU     11011111B
KEYR_7          EQU     10111111B
KEYR_8          EQU     01111111B

;*****      MAIN  ROUTINE    *****
MAIN_C  CSEG    INBLOCK

INIT:
;**********************************
;           初期設定
;**********************************
        DI                          ;すべての割り込みを禁止
        CLR1    MBE                 ;MBE←0
```

```
;++++++++++++++++++++++++++++++++
;            ｼｽﾃﾑ･ｸﾛｯｸ設定
;++++++++++++++++++++++++++++++++
        MOV     A,#0011B
        MOV     PCC,A               ;0.95μs,通常動作ﾓｰﾄﾞ

;++++++++++++++++++++++++++++++++
;            ｽﾀｯｸ･ﾎﾟｲﾝﾀ設定
;++++++++++++++++++++++++++++++++
        MOV     XA,#00H
        MOV     SP,XA               ;ｽﾀｯｸ･ﾎﾟｲﾝﾀ←00H
        MOV     SBS,A               ;ﾒﾓﾘ･ﾊﾞﾝｸ←0,MkⅡﾓｰﾄﾞ

;++++++++++++++++++++++++++++++++
;            入出力ﾎﾟｰﾄ設定
;++++++++++++++++++++++++++++++++
        MOV     A,#0H
        OUT     PORT2,A
        OUT     PORT8,A

        MOV     A,#0FH
        OUT     PORT3,A
        OUT     PORT4,A
        OUT     PORT5,A
        OUT     PORT6,A
        OUT     PORT7,A

        MOV     XA,#0FH
        MOV     PMGA,XA             ;ﾎﾟｰﾄ3を出力ﾓｰﾄﾞ,ﾎﾟｰﾄ6を入力ﾓｰﾄﾞに

        MOV     XA,#34H
        MOV     PMGB,XA             ;ﾎﾟｰﾄ2,4,5を出力ﾓｰﾄﾞ,ﾎﾟｰﾄ7を入力ﾓｰﾄﾞに

        MOV     XA,#00H
        MOV     PMGC,XA             ;ﾎﾟｰﾄ8を入力ﾓｰﾄﾞに

        MOV     XA,#0C1H            ;ﾎﾟｰﾄ1,2,3は内蔵ﾌﾟﾙｱｯﾌﾟ抵抗の接続を指定しない
        MOV     POGA,XA             ;ﾎﾟｰﾄ0,6,7は内蔵ﾌﾟﾙｱｯﾌﾟ抵抗の接続を指定する

        MOV     XA,#0H
        MOV     POGB,XA             ;ﾎﾟｰﾄ8は内蔵ﾌﾟﾙｱｯﾌﾟ抵抗の接続を指定しない

;++++++++++++++++++++++++++++++++
;            ﾊｰﾄﾞｳｴｱ設定
;++++++++++++++++++++++++++++++++
        MOV     XA,#00H
        MOV     WM,XA               ;時計ﾓｰﾄﾞを禁止
        MOV     TM0,XA              ;ﾀｲﾏ/ｲﾍﾞﾝﾄ･ｶｳﾝﾀの発振を停止
        MOV     TM1,XA              ;ﾀｲﾏ･ｶｳﾝﾀの発振を停止
```

```
        SET1    T0E0                ;出力許可ﾌﾗｸﾞ←1
        MOV     A,#0000B
        MOV     IPS,A               ;高位の割り込みを設定
        MOV     XA,#8AH
        MOV     CSIM,XA             ;ｼﾘｱﾙ動作ﾓｰﾄﾞを設定
        SET1    RELT

;++++++++++++++++++++++++++++++++
;           RAMｸﾘｱ設定
;++++++++++++++++++++++++++++++++
        SET1    MBE                 ;MBE←1
        SEL     MB0                 ;ﾒﾓﾘ･ﾊﾞﾝｸ←0
        MOV     A,#0H
        MOV     HL,#04H
LOOP1:
        MOV     @HL,A
        INCS    L
        BR      LOOP1

        INCS    H
        BR      LOOP1

        SEL     MB1                 ;ﾒﾓﾘ･ﾊﾞﾝｸ←1
        MOV     A,#0FH
LOOP2:
        MOV     @HL,A
        INCS    L
        BR      LOOP2

        INCS    H
        BR      LOOP2

        SEL     MB0                 ;ﾒﾓﾘ･ﾊﾞﾝｸ←0
        CLR1    MBE                 ;MBE←0

;++++++++++++++++++++++++++++++++
;           RAM設定
;++++++++++++++++++++++++++++++++
        MOV     A,#0DH
        MOV     LDCODE,A            ;LDCODEを初期化
        MOV     A,#0111B
        MOV     LEDDIG,A            ;LEDDIGを初期化
        MOV     A,#2H
        MOV     T_TIM,A             ;T_TIMを初期化

;++++++++++++++++++++++++++++++++
;       ﾍﾞｰｼｯｸ･ｲﾝﾀｰﾊﾞﾙ･ﾀｲﾏ設定
;++++++++++++++++++++++++++++++++
        MOV     A,#1111B
        MOV     BTM,A               ;ﾀｲﾏ･ｽﾀｰﾄ
```

```
        EI      IEBT                    ;ﾍﾞｰｼｯｸ･ｲﾝﾀｰﾊﾞﾙ･ﾀｲﾏ割り込み(INTBT)を許可
        EI                              ;すべての割り込みを許可

;*********************************
;           ﾒｲﾝ･ﾙｰﾁﾝ
;*********************************
MAIN:
        CALL    !RMDEC                  ;ﾘﾓｺﾝ･ﾃﾞｺｰﾄﾞ処理
        CALL    !KEYDEC                 ;ｷｰ･ﾃﾞｺｰﾄﾞ
        CALL    !LCDDATA                ;LCD表示ﾃﾞｰﾀ設定
        CALL    !TONEOUT                ;ﾒﾛﾃﾞｨ出力
        CALL    !TIM                    ;ﾀｲﾏ処理
        BR      MAIN

;*****  TABLE  LOOK  UP      *****
$IC=TABLE.INC

;*****       SUBROUTINESE     *****
$IC=SUB.INC

;+++++++++++++++++++++++++++++++++
;       ﾘﾓｺﾝ･ﾃﾞｺｰﾄﾞ処理
;+++++++++++++++++++++++++++++++++
RMDEC:
        SKF     HTKEYONF                ;1ならば本体ｷｰ･ｵﾝ
        RET

        SKTCLR  IRQT1                   ;1ならばﾓｼﾞｭﾛ･ﾚｼﾞｽﾀ=ｶｳﾝﾄ･ﾚｼﾞｽﾀ
        BR      MODEP7
        MOV     A,#0H
        MOV     MODEP,A                 ;MODEP←0H
        MOV     A,#0DH
        MOV     LDCODE,A                ;LDCODEを初期化
        DI      IE0                     ;INT0割り込みを禁止
        RET

MODEP7:
        MOV     A,MODEP
        SKE     A,#7H                   ;MODEP=7?
        RET
        MOV     A,#0H
        MOV     MODEP,A                 ;MODEP←0H
        MOV     RPCODE,A                ;ﾘﾋﾟｰﾄ･ｺｰﾄﾞ回数をｸﾘｱ
        MOV     RPTIM,A                 ;200msｶｳﾝﾀをｸﾘｱ
        MOV     XA,RMDATA
        MOV     HL,XA
        ADDS    XA,#0EDH
        BR      MODEP7_1
        RET
```

```
MODEP7_1:
        MOV     XA,HL
        MOV     BC,#HIGH  RM_T
        CALL    !TABLE
        MOV     BC,XA                   ;BCﾚｼﾞｽﾀ←RMDATAをｺｰﾄﾞ化
        SKF     KEYONF
        RET
        SET1    KEYONF
        MOV     XA,#10H
        SKE     XA,BC
        BR      MODEP7_2
        CLR1    KEYON2SF
        MOV     XA,#0H
        MOV     KEYTIM,XA
MODEP7_2:
        MOV     XA,BC
        MOV     KEYCODE,XA
        SET1    KEYCHKF
        RET

;++++++++++++++++++++++++++++++++
;           ｷｰ･ﾃﾞｺｰﾄﾞ
;++++++++++++++++++++++++++++++++
KEYDEC:
        SKT     KEYCHKF                 ;1ならばｷｰ･ｽｷｬﾝを終了
        RET
        CLR1    KEYCHKF
        SKT     KEYONF                  ;1ならばｷｰ･ｵﾝ
        BR      KEYOFF
        MOV     XA,KEYCODE
        ADDS    XA,#0F0H                ;音階ｷｰ･ｵﾝ?
        BR      ONKAI1
        ADDS    A,#0FH
        BR      MODE1                   ;0ならばMODEｷｰ･ｵﾝ
        ADDS    A,#0FH
        BR      UPKEY1                  ;1ならばUPｷｰ･ｵﾝ
        ADDS    A,#0FH
        BR      DOWN1                   ;2ならばDOWNｷｰ･ｵﾝ
        BR      ENDKEY1                 ;3ならばENDｷｰ･ｵﾝ

;--------------------------
;       音階ｷｰ･ｵﾝ
;--------------------------
ONKAI1:

        MOV     A,MODE                  ;PROGﾓｰﾄﾞ?
        SKE     A,#0H
        BR      ONKAI1_1
        MOV     A,#0H                   ;音長ﾃﾞｰﾀ←0H
```

```
        MOV     T_TIMC,A
        BR      ONKAI1_2

ONKAI1_1:
        SKE     A,#2H                   ;PAUSﾓｰﾄﾞ?
        RET
        MOV     A,#0FH                  ;音長ﾃﾞｰﾀ←FH
ONKAI1_2:
        MOV     TONETIM,A
        MOV     A,KEYCODE               ;TONE←KEYCODE
        MOV     TONE,A
        SET1    TONE_F
        RET

;-------------------------
;       MODEｷｰ(≧2s)
;-------------------------
MODE1:
        SKT     KEYON2SF
        RET
        MOV     A,MODE
        SKE     A,#0H                   ;PROGﾓｰﾄﾞ?
        BR      MODE1_1
        MOV     A,#2H
        MOV     MODE,A                  ;MODE←PAUSﾓｰﾄﾞ
        RET

MODE1_1:
        MOV     A,#0H
        MOV     MODE,A                  ;MODE←PROGﾓｰﾄﾞ
        CLR1    TONE_F
        RET

;-------------------------
;       UPｷｰ･ｵﾝ
;-------------------------
UPKEY1:
        MOV     A,MODE
        SKE     A,#0H                   ;PROGﾓｰﾄﾞ?
        BR      UPKEY1_1
        MOV     XA,MUS
        ADDS    XA,#2H
        NOP
        MOV     MUS,XA                  ;演奏ﾃﾞｰﾀ･ｱﾄﾞﾚｽ+2H
        BR      DOWN1_1

UPKEY1_1:
        MOV     A,T_TIM
        SKE     A,#9H
        INCS    T_TIM                   ;基本時間+1H
```

```
        NOP
        BR      DOWN1_3

;-------------------------
;       DOWNｷｰ･ｵﾝ
;-------------------------
DOWN1:
        MOV     A,MODE
        SKE     A,#0H               ;PROGﾓｰﾄﾞ?
        BR      DOWN1_2
        MOV     XA,MUS
        DECS    XA
        NOP
        DECS    XA
        NOP
        MOV     MUS,XA              ;演奏ﾃﾞｰﾀ･ｱﾄﾞﾚｽ-2
DOWN1_1:
        MOV     XA,MUS
        MOV     HL,XA
        INCS    HL
        NOP

        SET1    MBE                 ;MBE←1
        SEL     MB1                 ;ﾒﾓﾘ･ﾊﾞﾝｸ←1
        MOV     A,@HL
        NOP
        SEL     MB0                 ;ﾒﾓﾘ･ﾊﾞﾝｸ←0
        CLR1    MBE                 ;MBE←0

        MOV     TONE,A              ;音階ﾃﾞｰﾀ←@演奏ﾃﾞｰﾀ･ｱﾄﾞﾚｽ+1
        SET1    TONE_F              ;ｷｰ･ｵﾝの間演奏ﾃﾞｰﾀを出力
        RET

DOWN1_2:
        MOV     A,T_TIM
        SKE     A,#1H
        DECS    A
        NOP
        MOV     T_TIM,A             ;基本時間-1

DOWN1_3:
        MOV     A,#0AH
        MOV     DISPTIM,A           ;基本時間表示のｶｳﾝﾄ･ｽﾀｰﾄ
        RET
```

```
;-------------------------
;       ENDｷｰ･ｵﾝ
;-------------------------
ENDKEY1:
        MOV     A,MODE
        SKE     A,#0H               ;PROGﾓｰﾄﾞ?
        BR      END2
        MOV     A,#0H
        MOV     TONETIM,A           ;音長ｶｳﾝﾄ･ｽﾀｰﾄ
        MOV     T_TIMC,A
        BR      ONKAI2_1

END2:
        CLR1    TONE_F
        MOV     A,MODE
        SKE     A,#1H               ;PLAYﾓｰﾄﾞ?
        BR      END2_2
        MOV     A,#0H
        MOV     TONETIM,A
        MOV     T_TIMC,A
END2_1:
        MOV     XA,#0H
        MOV     MUS,XA              ;演奏ﾃﾞｰﾀ･ｱﾄﾞﾚｽ←0H
        RET

END2_2:
        MOV     A,#0AH
        MOV     LCDTIM,A            ;LCD表示用ﾀｲﾏ(1s)のｶｳﾝﾄ･ｽﾀｰﾄ
        BR      END2_1

;-------------------------
;       ｷｰ･ｵﾌ
;-------------------------
KEYOFF:
        MOV     XA,KEYCODE
        ADDS    XA,#0F0H            ;音階ｷｰ･ｵﾌ?
        BR      ONKAI2
;-------------------------
;       ENDｷｰ･ｵﾌ
;-------------------------
        SKE     A,#3H               ;ENDｷｰ･ｵﾌ?
        BR      MODE2
        MOV     A,MODE
        SKE     A,#0H               ;PROGﾓｰﾄﾞ?
        RET
        MOV     XA,MUS
        MOV     HL,XA
        MOV     A,TONETIM
        MOV     B,A
```

```
        SET1    MBE                     ;MBE←1
        SEL     MB1                     ;ﾒﾓﾘ･ﾊﾞﾝｸ←1
        MOV     A,#0FH
        MOV     @HL,A                   ;@演奏ﾃﾞｰﾀ･ｱﾄﾞﾚｽ←FH
        INCS    HL                      ;演奏ﾃﾞｰﾀ･ｱﾄﾞﾚｽ+1H
        NOP
        MOV     A,B
        MOV     @HL,A                   ;@演奏ﾃﾞｰﾀ･ｱﾄﾞﾚｽ+1←音長ﾃﾞｰﾀ
        SEL     MB0                     ;ﾒﾓﾘ･ﾊﾞﾝｸ←0
        CLR1    MBE                     ;MBE←0
        RET

;-------------------------
;       音階ｷｰ･ｵﾌ
;-------------------------
ONKAI2:
        MOV     A,MODE
        SKE     A,#0H                   ;PROGﾓｰﾄﾞ?
        BR      ONKAI2_2
        MOV     A,TONETIM               ;Aﾚｼﾞｽﾀ←TONETIM
        CALL    !TONE_S
        MOV     XA,MUS
        ADDS    XA,#2H
        NOP
        MOV     MUS,XA                  ;演奏ﾃﾞｰﾀ･ｱﾄﾞﾚｽ+2H
        INCS    HL                      ;演奏ﾃﾞｰﾀ･ｱﾄﾞﾚｽ+1H
        NOP
        MOV     A,TONE                  ;Aﾚｼﾞｽﾀ←TONE

        SET1    MBE                     ;MBE←1
        SEL     MB1                     ;ﾒﾓﾘ･ﾊﾞﾝｸ←1
        MOV     @HL,A                   ;@演奏ﾃﾞｰﾀ･ｱﾄﾞﾚｽ+1H←TONE
        SEL     MB0                     ;ﾒﾓﾘ･ﾊﾞﾝｸ←0
        CLR1    MBE                     ;MBE←0

ONKAI2_1:
        CLR1    TONE_F                  ;無音を出力
        RET

ONKAI2_2:
        SKE     A,#2H                   ;PAUSﾓｰﾄﾞ?
        RET
        BR      ONKAI2_1
```

```
;--------------------------
;       MODEｷｰ(<2s)
;--------------------------
MODE2:
        SKE     A,#0H               ;MODEｷｰ･ｵﾌ?
        BR      ONKAI2_1
        SKF     KEYON2SF
        RET
        MOV     A,MODE
        SKE     A,#0H               ;PROGﾓｰﾄﾞ?
        BR      MODE2_4
        MOV     XA,MUS
        ADDS    XA,#0FFH            ;演奏ﾃﾞｰﾀ･ｱﾄﾞﾚｽ=00H?
        BR      MODE2_2
        BR      END2_1

MODE2_1:
        MOV     MUS,XA
MODE2_2:
        MOV     XA,MUS
        MOV     HL,XA

        SET1    MBE
        SEL     MB1
        MOV     A,@HL
        SEL     MB0
        CLR1    MBE

        SKE     A,#0FH              ;@演奏ﾃﾞｰﾀ･ｱﾄﾞﾚｽ=FH?
        BR      MODE2_3
        RET

MODE2_3:
        MOV     XA,MUS
        ADDS    XA,#2H              ;演奏ﾃﾞｰﾀ･ｱﾄﾞﾚｽ+2H>256?
        BR      MODE2_1
        BR      END2_1

MODE2_4:
        SKE     A,#1H               ;PLAYﾓｰﾄﾞ?
        BR      MODE2_5
        MOV     A,#2H
        MOV     MODE,A              ;MODE←PAUSﾓｰﾄﾞ
        BR      ONKAI2_1

MODE2_5:
        MOV     A,#1H
        MOV     MODE,A              ;MODE←PLAYﾓｰﾄﾞ
        MOV     A,#0H
```

```
        MOV     TONETIM,A
        MOV     T_TIMC,A

;++++++++++++++++++++++++++++++++
;       LCD表示ﾃﾞｰﾀ設定
;++++++++++++++++++++++++++++++++
LCDDATA:
        SKT     REFRES_F            ;1ならばLCDをﾘﾌﾚｯｼｭ
        RET
        CLR1    REFRES_F            ;REFRES_F←0
;-------------------------
;   ｱﾄﾞﾚｽHEX-->DEC変換
;-------------------------
        MOV     A,#0H
        MOV     DEC,A               ;DEC←0H
        MOV     A,MUS
        XCH     A,E                 ;Eﾚｼﾞｽﾀ←MUS

        MOV     A,MUS+1
LCDHEX1:
        MOV     HL,#DEC+1           ;10進補正(1桁目)
        ADDS    A,#6H
        BR      LCDHEX4

        MOV     @HL,A               ;2桁目をｲﾝｸﾘﾒﾝﾄ
        INCS    L
        MOV     A,#7H
        ADDS    A,@HL               ;10進補正
        BR      LCDHEX4

        INCS    DEC                 ;3桁目をｲﾝｸﾘﾒﾝﾄ
LCDHEX2:
        MOV     @HL,A               ;2桁目の値の回数に,6を加算
        DECS    E
        BR      LCDHEX3
        BR      LCD1

LCDHEX3:
        MOV     A,DEC+2
        ADDS    A,#6H
        BR      LCDHEX1

LCDHEX4:
        ADDS    A,#0AH
        NOP
        BR      LCDHEX2

LCD1:
        MOV     A,MODE
        SKE     A,#2H               ;PAUSﾓｰﾄﾞ?
```

```
        BR      LCD2
        MOV     A,LCDTIM
        SKE     A,#0H                   ;LCD表示用ﾀｲﾏ(1s)=0?
        BR      LCD2
        MOV     XA,#2EH
        MOV     LCD,XA                  ;ｱﾄﾞﾚｽの1桁目を設定
        MOV     LCD+2,XA                ;ｱﾄﾞﾚｽの2桁目を設定
        MOV     LCD+4,XA                ;ｱﾄﾞﾚｽの3桁目を設定
        BR      LCD3

LCD2:
        MOV     X,#0H
        MOV     A,DEC
        MOV     LCD,XA                  ;ｱﾄﾞﾚｽの1桁目を設定
        MOV     A,DEC+1
        MOV     LCD+2,XA                ;ｱﾄﾞﾚｽの2桁目を設定
        MOV     A,DEC+2
        MOV     LCD+4,XA                ;ｱﾄﾞﾚｽの3桁目を設定
LCD3:
        MOV     XA,#27H
        MOV     LCD+6,XA                ;"-"ﾃﾞｰﾀ設定
        MOV     XA,MUS
        MOV     HL,XA
        MOV     XA,@HL                  ;XAﾚｼﾞｽﾀ←@演奏ﾃﾞｰﾀ･ｱﾄﾞﾚｽ
        ADDS    XA,XA
        ADDS    XA,XA                   ;XAﾚｼﾞｽﾀ×4
        MOV     DE,XA
        MOV     BC,#HIGH  LCD_T
        CALL    !TABLE
        MOV     LCD+8,XA                ;音階ﾃﾞｰﾀの1桁目を設定
        CALL    !TABLEINC
        MOV     LCD+10,XA               ;音階ﾃﾞｰﾀの2桁目を設定
        CALL    !TABLEINC
        MOV     LCD+12,XA               ;音階ﾃﾞｰﾀの3桁目を設定
        CALL    !TABLEINC
        MOV     LCD+14,XA               ;音階ﾃﾞｰﾀの4桁目を設定
        SKT     PORT0.2                 ;SBO=ﾊｲ･ﾚﾍﾞﾙ?
        RET
        SKT     PORT0.1                 ;SCK=ﾊｲ･ﾚﾍﾞﾙ?
        RET
        SET1    CMDT                    ;CMDTｾｯﾄ
        NOP
        NOP
        SET1    RELT                    ;RELTｾｯﾄ
        NOP
        NOP
        SKT     PORT0.2                 ;SBO=ﾊｲ･ﾚﾍﾞﾙ?
        RET
        SET1    CMDT                    ;CMDTｾｯﾄ
        MOV     XA,#01H
```

```
        MOV     SIO,XA              ;ｼﾌﾄ･ﾚｼﾞｽﾀ←ｽﾚｰﾌﾞ･ｱﾄﾞﾚｽ(01H)
        MOV     A,#1H
        MOV     MODESBI,A           ;MODESBI←1H
        EI      IECSI               ;SBI割り込みを許可
        MOV     XA,#1H
        MOV     SVA,XA              ;ｽﾚｰﾌﾞ･ｱﾄﾞﾚｽ･ﾚｼﾞｽﾀ←1H
        CLR1    REFRES_F            ;REFRES_F←0
        RET

;++++++++++++++++++++++++++++++++
;           ﾒﾛﾃﾞｨ出力
;++++++++++++++++++++++++++++++++
TONEOUT:
        SKT     TONE_F              ;1ならば音階出力を許可
        BR      TONE0
        MOV     X,#0H
        MOV     A,TONE
        MOV     BC,#HIGH  TONE_T
        CALL    !TABLE
        MOV     TMOD0,XA            ;ﾓｼﾞｭﾛ･ﾚｼﾞｽﾀ←音階ﾃﾞｰﾀ
        MOV     A,TONE
        ADDS    A,#2H
        BR      TONE1
TONE0:
        MOV     XA,#0H              ;ﾀｲﾏ･ｶｳﾝﾀを停止
        CLR1    TM0_F               ;0ならばﾀｲﾏ･ｶｳﾝﾄ･ｽﾀｰﾄ
        BR      TONE2

TONE1:
        SKF     TM0_F
        RET
        MOV     XA,#7CH             ;ﾀｲﾏ･ｶｳﾝﾄ･ｽﾀｰﾄ
        SET1    TM0_F               ;1ならばﾀｲﾏ･ｶｳﾝﾀを停止
TONE2:
        MOV     TM0,XA
        RET

;++++++++++++++++++++++++++++++++
;           ﾀｲﾏ処理
;++++++++++++++++++++++++++++++++
TIM:
        MOV     A,B_TIMC
        SKE     A,#0AH              ;B_TIMC≧100ms?
        BR      TIM_50
;=========================
;       100ms通過
;=========================
        MOV     A,#0H
        MOV     B_TIMC,A
        SKT     REP_F
```

```
        BR      TIM_51
        INCS    RPTIM                   ;ﾘﾓｺﾝ･ﾘﾋﾟｰﾄ用ﾀｲﾏ+1
        NOP
        MOV     A,RPTIM
        SKE     A,#2H                   ;ﾘﾓｺﾝ･ﾘﾋﾟｰﾄ用ﾀｲﾏ=200ms?
        BR      TIM_51
        MOV     A,#0H
        MOV     RPTIM,A                 ;PRTIMをｸﾘｱ
        MOV     A,RPCODE
        ADDS    A,#0FH                  ;ﾘﾓｺﾝ･ﾘﾋﾟｰﾄ･ｺｰﾄﾞ･ｶｳﾝﾀ=0
        BR      TIM_52
        ADDS    A,#0FH                  ;ﾘﾓｺﾝ･ﾘﾋﾟｰﾄ･ｺｰﾄﾞ･ｶｳﾝﾀ=1
        BR      TIM_54
        ADDS    A,#0FH                  ;ﾘﾓｺﾝ･ﾘﾋﾟｰﾄ･ｺｰﾄﾞ･ｶｳﾝﾀ=2
        BR      TIM_54
        ADDS    A,#0FH                  ;ﾘﾓｺﾝ･ﾘﾋﾟｰﾄ･ｺｰﾄﾞ･ｶｳﾝﾀ=3
        BR      TIM_54

TIM_52:
        SKF     HTKEYONF
        BR      TIM_51
        SET1    KEYCHKF                 ;ﾘﾓｺﾝ･ﾘﾋﾟｰﾄ･ｺｰﾄﾞ･ｶｳﾝﾀ=0  or  ≧4
        CLR1    KEYONF                  ;KEYONF←0
        CLR1    REP_F                   ;REP_F←0
TIM_54:
        MOV     A,#0H
        MOV     RPCODE,A                ;RPCODEをｸﾘｱ
TIM_51:
        MOV     XA,KEYCODE
        MOV     HL,#10H
        SKE     XA,HL
        BR      TIM_101
        SKF     KEYON2SF
        BR      TIM_101
        MOV     XA,KEYTIM
        MOV     HL,#14H
        SKE     XA,HL                   ;ｷｰを2秒以上押し続けている?
        BR      TIM_105
        SET1    KEYON2SF                ;1ならばｷｰを2秒以上押し続けている
        SET1    KEYCHKF
TIM_101:
        MOV     A,DISPTIM
        SKE     A,#0H                   ;LED表示用ﾀｲﾏ(1s)をｶｳﾝﾄ中?
        BR      TIM_107
TIM_102:
        MOV     A,LCDTIM
        SKE     A,#0H                   ;LCD表示用ﾀｲﾏ(1s)をｶｳﾝﾄ中?
        BR      TIM_108
TIM_103:
        MOV     A,MODE
```

```
        SKE     A,#0H               ;PROGﾓｰﾄﾞ?
        BR      TIM_109
        INCS    T_TIMC              ;基本時間用ｶｳﾝﾀ+1
        NOP
        MOV     A,T_TIM
        MOV     L,A
        MOV     A,T_TIMC
        SKE     A,L                 ;基本時間=基本時間用ｶｳﾝﾀ?
        BR      TIM_53
        MOV     A,#0H
        MOV     T_TIMC,A
        MOV     A,TONETIM
        SKE     A,#0EH              ;音長ﾃﾞｰﾀ=EH?
        BR      TIM_104
        BR      TIM_53

TIM_104:
        INCS    TONETIM             ;音長ﾃﾞｰﾀ+1
        NOP
        BR      TIM_53

;-------------------------
;       ｷｰ･ｵﾝ時間ｶｳﾝﾄ
;-------------------------
TIM_105:
        MOV     XA,KEYTIM
        MOV     HL,XA
        INCS    HL
        NOP
        MOV     XA,HL
        MOV     KEYTIM,XA           ;ｷｰ･ｵﾝ時間ｶｳﾝﾄ用ﾀｲﾏ+1
        BR      TIM_101

;-------------------------
;   LED表示用ﾀｲﾏ(1s)ｶｳﾝﾄ
;-------------------------
TIM_107:
        MOV     A,DISPTIM
        DECS    A
        NOP
        MOV     DISPTIM,A           ;LED表示用(1s)ﾀｲﾏ-1
        BR      TIM_102

;-------------------------
;   LCD表示用ﾀｲﾏ(1s)ｶｳﾝﾄ
;-------------------------
TIM_108:
        MOV     A,LCDTIM
        DECS    A
        NOP
```

```
        MOV     LCDTIM,A            ;LCD表示用(1s)ﾀｲﾏ-1
        BR      TIM_103

TIM_109:
        SKE     A,#1H               ;PLAYﾓｰﾄﾞ?
        BR      TIM_53
        INCS    T_TIMC              ;基本時間用ｶｳﾝﾀ+1
        NOP
TIM_110:
        MOV     A,T_TIM
        MOV     L,A
        MOV     A,T_TIMC
        SKE     A,L                 ;基本時間=基本時間用ｶｳﾝﾀ?
        BR      TIM_53
        MOV     A,#0H
        MOV     T_TIMC,A
        MOV     A,TONETIM
        DECS    A
        BR      TIM_116
        MOV     XA,MUS

        SET1    MBE                 ;MBE←1
        SEL     MB1                 ;ﾒﾓﾘ･ﾊﾞﾝｸ←1
        MOV     HL,XA
        MOV     A,@HL+
        NOP
        MOV     B,A
        MOV     A,@HL
        MOV     C,A
        SEL     MB0                 ;ﾒﾓﾘ･ﾊﾞﾝｸ←0
        CLR1    MBE                 ;MBE←0

        MOV     A,B
        MOV     TONETIM,A           ;音長ﾃﾞｰﾀ←@演奏ﾃﾞｰﾀ･ｱﾄﾞﾚｽ
        MOV     A,C
        MOV     TONE,A              ;音階ﾃﾞｰﾀ←@演奏ﾃﾞｰﾀ･ｱﾄﾞﾚｽ+1
        MOV     A,#0H
        MOV     T_TIMC,A
        SKE     C,#0FH
        BR      TIM_115
        BR      TIM_114

TIM_115:
        SKE     B,#0FH
        BR      TIM_111
        BR      TIM_112

TIM_112:
        MOV     XA,#0H
        MOV     MUS,XA
```

```
        MOV     A,TONE
        MOV     TONETIM,A           ;TONETIM←ﾘﾋﾟｰﾄ時間
        MOV     A,#0FH
        MOV     TONE,A              ;TONE←FH
TIM_114:
        CLR1    TONE_F
        SKE     B,#0FH
        BR      TIM_113
        BR      TIM_53

TIM_111:
        SET1    TONE_F
TIM_113:
        MOV     XA,MUS
        ADDS    XA,#2H
        NOP
        MOV     MUS,XA              ;演奏ﾃﾞｰﾀ･ｱﾄﾞﾚｽ+2
        BR      TIM_53

TIM_116:
        MOV     TONETIM,A           ;音長ﾃﾞｰﾀ-1
        BR      TIM_53

;=========================
;       50ms通過
;=========================
TIM_50:
        SKE     A,#5H               ;B_TIMC≧50ms?
        RET
;-------------------------
;   ﾘﾓｺﾝ･ﾘﾋﾟｰﾄ用ﾀｲﾏ･ｶｳﾝﾄ
;-------------------------
TIM_53:
        SET1    REFRES_F            ;1ならばLCD表示をﾘﾌﾚｯｼｭ
        RET

;+++++++++++++++++++++++++++++++++
;           ﾘﾓｺﾝ処理
;+++++++++++++++++++++++++++++++++
INT0:
        DI
        PUSH    BS
        SEL     RB1                 ;ﾚｼﾞｽﾀ･ﾊﾞﾝｸ←1
        MOV     A,MODEP
        ADDS    A,#0FH              ;MODEP=0?
        BR      INT0_E
        ADDS    A,#0FH              ;MODEP=1?
        BR      INT0_P1
        ADDS    A,#0FH              ;MODEP=2?
        BR      INT0_P2
```

```
        ADDS    A,#09H              ;MODEP=3-6?
        BR      INT0_PX
        BR      INT0_E

INT0_P1:
        MOV     A,#2H
        MOV     MODEP,A             ;MODEP←2
        MOV     XA,#62H
        MOV     TMOD1,XA            ;ﾓｼﾞｭﾛ･ﾚｼﾞｽﾀ←6ms
        MOV     A,#1H
        MOV     IM0,A               ;立ち下がりｴｯｼﾞを指定
        BR      INT0_E

INT0_P2:
        MOV     XA,T1
        ADDS    XA,#0C9H            ;ｶｳﾝﾄ･ﾚｼﾞｽﾀ=3.375ms?
        BR      INT0_RP1
        MOV     A,#3H
        MOV     MODEP,A             ;MODEP←3
        BR      INT0_W1

INT0_RP1:
        INCS    RPCODE              ;ﾘﾓｺﾝ･ﾘﾋﾟｰﾄ用ｺｰﾄﾞ･ｶｳﾝﾀ+1
        NOP
        BR      INT0_P0

INT0_PX:
        MOV     XA,T1
        ADDS    XA,#0E4H            ;ｶｳﾝﾄ･ﾚｼﾞｽﾀ=1.6875ms?
        BR      INT0_CY0
        SET1    CY                  ;CY←1
        BR      INT0_L1

INT0_CY0:
        CLR1    CY                  ;CY←0
INT0_L1:
        MOV     XA,WORK
        MOV     B,A
        MOV     A,X
        RORC    A
        MOV     X,A
        MOV     A,B
        RORC    A
        MOV     WORK,XA             ;受信ﾃﾞｰﾀ(WORK)を右へ1ﾋﾞｯﾄ･ｼﾌﾄ
        SKT     CY
        BR      INT0_E
        MOV     A,MODEP
        ADDS    A,#0CH              ;MODEP=3?
        BR      INT0_P3
        ADDS    A,#0FH              ;MODEP=4?
```

```
        BR      INT0_P4
        ADDS    A,#0FH              ;MODEP=5?
        BR      INT0_P5
        MOV     XA,WORK             ;MODEP=6
        MOV     HL,#0FFH
        XOR     HL,XA
        MOV     XA,RMDATA
        SKE     XA,HL               ;有効なﾘﾓｺﾝ･ﾃﾞｰﾀ=受信ﾃﾞｰﾀ反転?
        BR      INT0_P0
        MOV     A,#7H
        MOV     MODEP,A             ;MODEP←7
        SET1    REP_F
        DI      IE0                 ;INT0割り込みを禁止
        MOV     XA,#0H
        MOV     TM1,XA              ;ﾀｲﾏ･ｶｳﾝﾀを停止
        BR      INT0_E1

INT0_P3:
        MOV     XA,WORK
        MOV     HL,#00H
        SKE     XA,HL               ;ｶｽﾀﾑ･ｺｰﾄﾞ=00H?
        BR      INT0_P0
        MOV     A,#4H
        MOV     MODEP,A             ;MODEP←4
        BR      INT0_W1

INT0_P4:
        MOV     XA,WORK
        MOV     HL,#0FFH
        SKE     XA,HL               ;ｶｽﾀﾑ･ｺｰﾄﾞ=FFH?
        BR      INT0_P0
        MOV     A,#5H
        MOV     MODEP,A             ;MODEP←5
        BR      INT0_W1
INT0_P5:
        MOV     XA,WORK
        MOV     RMDATA,XA           ;有効なﾘﾓｺﾝ･ﾃﾞｰﾀ←受信ﾃﾞｰﾀ
        MOV     A,#6H
        MOV     MODEP,A             ;MODEP←6
        BR      INT0_W1

INT0_P0:
        MOV     A,#0H
        MOV     MODEP,A             ;MODEP←0
        MOV     A,#0DH
        MOV     LDCODE,A            ;LDCODEを初期化
        DI      IE0                 ;INT0割り込みを禁止
        MOV     XA,#0H
        MOV     TM1,XA              ;ﾀｲﾏ･ｶｳﾝﾀを停止
        BR      INT0_E1
```

```
INT0_W1:
        MOV     XA,#42H
        MOV     TMOD1,XA            ;ﾓｼﾞｭﾛ･ﾚｼﾞｽﾀ←4ms
        MOV     XA,#80H
        MOV     WORK,XA             ;WORKを初期化
INT0_E:
        MOV     XA,#6CH
        MOV     TM1,XA              ;ﾀｲﾏ･ｶｳﾝﾄ･ｽﾀｰﾄ
INT0_E1:
        POP     BS
        EI                          ;すべての割り込みを許可
        RETI

;++++++++++++++++++++++++++++++++
;               ﾀｲﾏ
;++++++++++++++++++++++++++++++++
INTBT:
        SEL     RB2                 ;ﾚｼﾞｽﾀ･ﾊﾞﾝｸ←2
        DI      IECSI               ;SBI割り込みを禁止
        EI                          ;すべての割り込みを許可
        CALL    !TIMCNT             ;ﾀｲﾏ･ｶｳﾝﾄ
        CALL    !RMCNT              ;ﾘﾓｺﾝ処理ｶｳﾝﾄ
        CALL    !KEYCHK             ;ｷｰ･ﾁｪｯｸ
        CALL    !LED1               ;LED表示
        EI      IECSI               ;SBI割り込みを許可
        RETI

;++++++++++++++++++++++++++++++++
;           SBI処理
;++++++++++++++++++++++++++++++++
SBI:
        SEL     RB3                 ;ﾚｼﾞｽﾀ･ﾊﾞﾝｸ←3
        DI      IEBT                ;ﾍﾞｰｼｯｸ･ｲﾝﾀｰﾊﾞﾙ･ﾀｲﾏ割り込み(INTBT)を禁止
        EI                          ;すべての割り込みを許可
        MOV     A,MODESBI
        ADDS    A,#0FH
        BR      SBI_E
        ADDS    A,#0FH
        BR      SBI1
        ADDS    A,#0FH
        BR      SBI2
        ADDS    A,#0FH
        BR      SBI3
        ADDS    A,#0FH
        BR      SBI4
        ADDS    A,#0FH
        BR      SBI5
        ADDS    A,#0FH
        BR      SBI6
```

```
        ADDS    A,#0FH
        BR      SBI7
        ADDS    A,#0FH
        BR      SBI8
        ADDS    A,#0FH
        BR      SBI9
        ADDS    A,#0FH
        BR      SBI10
        ADDS    A,#0FH
        BR      SBI11
        BR      SBI_E

SBI1:
        MOV     BC,#04H         ;ｼﾌﾄ･ﾚｼﾞｽﾀ←ﾎﾟｲﾝﾀ･ｺﾏﾝﾄﾞ(04H)
        BR      SBITIM

SBI2:
        MOV     BC,#07H         ;ｼﾌﾄ･ﾚｼﾞｽﾀ←ﾎﾟｲﾝﾀ(07H)
        BR      SBITIM

SBI3:
        MOV     BC,#01H         ;ｼﾌﾄ･ﾚｼﾞｽﾀ←ﾗｲﾄ･ｺﾏﾝﾄﾞ(01H)
        BR      SBITIM

SBI4:
        MOV     XA,LCD          ;ｼﾌﾄ･ﾚｼﾞｽﾀ←ｱﾄﾞﾚｽ1桁目
        MOV     BC,XA
        BR      SBITIM

SBI5:
        MOV     XA,LCD+2        ;ｼﾌﾄ･ﾚｼﾞｽﾀ←ｱﾄﾞﾚｽ2桁目
        MOV     BC,XA
        BR      SBITIM

SBI6:
        MOV     XA,LCD+4        ;ｼﾌﾄ･ﾚｼﾞｽﾀ←ｱﾄﾞﾚｽ3桁目
        MOV     BC,XA
        BR      SBITIM

SBI7:
        MOV     XA,LCD+6        ;ｼﾌﾄ･ﾚｼﾞｽﾀ←"_"ﾃﾞｰﾀ
        MOV     BC,XA
        BR      SBITIM

SBI8:
        MOV     XA,LCD+8        ;ｼﾌﾄ･ﾚｼﾞｽﾀ←音階ﾃﾞｰﾀ1桁目
        MOV     BC,XA
        BR      SBITIM
```

```
SBI9:
        MOV     XA,LCD+10           ;ｼﾌﾄ･ﾚｼﾞｽﾀ←音階ﾃﾞｰﾀ2桁目
        MOV     BC,XA
        BR      SBITIM

SBI10:
        MOV     XA,LCD+12           ;ｼﾌﾄ･ﾚｼﾞｽﾀ←音階ﾃﾞｰﾀ3桁目
        MOV     BC,XA
        BR      SBITIM
SBI11:
        MOV     XA,LCD+14           ;ｼﾌﾄ･ﾚｼﾞｽﾀ←音階ﾃﾞｰﾀ4桁目
        MOV     BC,XA
SBITIM:
        MOV     L,#1110B            ;100μsｶｳﾝﾀを初期化
SBICNT:
        DECS    L
        BR      SBI_ACKD

SBI0:
        MOV     A,#0H
        MOV     MODESBI,A           ;MODESBI←0
        SET1    REFRES_F            ;1ならばLCD表示をﾘﾌﾚｯｼｭ
        BR      SBI_E

SBI_ACKD:
        SKT     ACKD                ;ｱｸﾉﾘｯｼﾞ検出ﾌﾗｸﾞ=1?
        BR      SBICNT
        SKT     COI                 ;ｱﾄﾞﾚｽ･ｺﾝﾊﾟﾚｰﾀ信号=1?
        BR      SBI0
        SKT     PORT0.2             ;SBO=ﾊｲ･ﾚﾍﾞﾙ?
        BR      SBI_E
        SKT     PORT0.1             ;SCK=ﾊｲ･ﾚﾍﾞﾙ?
        BR      SBI_E
        MOV     A,MODESBI
        ADDS    A,#0FH              ;MODESBI=0?
        BR      SBI_SIO
        ADDS    A,#0FH              ;MODESBI=1?
        BR      SBI_CMDT
        ADDS    A,#0FH              ;MODESBI=2?
        BR      SBI_SIO
        ADDS    A,#0FH              ;MODESBI=3?
SBI_CMDT:
        SET1    CMDT                ;CMDTｾｯﾄ
SBI_SIO:
        MOV     XA,BC
        MOV     SIO,XA              ;ｼﾌﾄ･ﾚｼﾞｽﾀ←BCﾚｼﾞｽﾀ
        INCS    MODESBI             ;MODESBI+1
        NOP
        SET1    CSIE                ;ｼﾘｱﾙ･ｲﾝﾀﾌｪｰｽ動作,ｼﾌﾄ･ﾚｼﾞｽﾀを禁止
        EI      IECSI               ;SBI割り込みを許可
```

```
SBI_E:
        EI          IEBT                ;ﾍﾞｰｼｯｸ･ｲﾝﾀｰﾊﾞﾙ･ﾀｲﾏ割り込み(INTBT)を許可
        RETI

END
```

```
$NOLIST
;*************************************************
;***                サブルーチン                 ***
;*************************************************
$LIST
$TT='SUBROUTINE'
TONE_S:
;++++++++++++++++++++++++++
;+++    音長ﾃﾞｰﾀ        +++
;++++++++++++++++++++++++++
        MOV     B,A
        MOV     XA,MUS
        MOV     HL,XA
        SET1    MBE                 ;MBE←1
        SEL     MB1                 ;ﾒﾓﾘ･ﾊﾞﾝｸ←1
        MOV     A,B
        MOV     @HL,A
        CLR1    MBE                 ;MBE←0
        RET

;++++++++++++++++++++++++++++++++
;           ﾀｲﾏ･ｶｳﾝﾄ
;++++++++++++++++++++++++++++++++
TIMCNT:
        INCS    B_TIM               ;ﾍﾞｰｽ･ﾀｲﾏ+1
        NOP
        MOV     A,B_TIM
        ADDS    A,#0BH              ;ﾍﾞｰｽ･ﾀｲﾏ>10ms?
        RET
        INCS    B_TIMC              ;100msｶｳﾝﾄ用ﾀｲﾏ+1
        NOP
        MOV     A,#0H
        MOV     B_TIM,A             ;ﾍﾞｰｽ･ﾀｲﾏ←0H
        RET

;++++++++++++++++++++++++++++++++
;           ﾘﾓｺﾝ処理ｶｳﾝﾄ
;++++++++++++++++++++++++++++++++
RMCNT:

        SKF     HTKEYONF
        BR      RMCNT_1
        MOV     A,MODEP
        SKE     A,#0H               ;MODEP=0?
        RET
        SKF     PORT1.0
        BR      RMCNT_1
        INCS    LDCODE              ;ﾘｰﾀﾞ･ｺｰﾄﾞ･ｽｷｬﾝ用ｶｳﾝﾀ+1
        RET
        MOV     A,#0DH              ;LDCODEを初期化
```

```
        MOV     LDCODE,A
        MOV     A,#1H
        MOV     MODEP,A             ;MODEP←1
        MOV     A,#0H
        MOV     IM0,A               ;立ち上がりｴｯｼﾞを指定
        MOV     XA,#62H             ;ﾓｼﾞｭﾛ･ﾚｼﾞｽﾀ←6ms
        MOV     TMOD1,XA
        MOV     XA,#6CH
        MOV     TM1,XA              ;ﾀｲﾏ･ｶｳﾝﾄ･ｽﾀｰﾄ
        CLR1    IRQ0
        EI      IE0                 ;INT0割り込みを許可
        RET

RMCNT_1:
        MOV     A,#0DH
        MOV     LDCODE,A            ;LDCODEを初期化
        RET

;++++++++++++++++++++++++++++++++
;           ｷｰ･ﾁｪｯｸ
;++++++++++++++++++++++++++++++++
KEYCHK:
        MOV     A,LEDDIG
        SKE     A,#0111B            ;LEDDIG=0111B?
        RET
        MOV     A,#0H
        OUT     PORT3,A             ;ﾎﾟｰﾄ3←0
        MOV     XA,#0FFH
        OUT     PORT4,XA            ;ﾎﾟｰﾄ4,5←ALL1
        IN      XA,PORT6
        MOV     BC,#0FFH
        SKE     XA,BC               ;ﾎﾟｰﾄ6,7=ALL1?
        BR      KEYCHK1
        SKT     HTKEYONF            ;1ならば本体ｷｰ･ｵﾝ
        RET
        SKT     KEYONF              ;1ならばｷｰ･ｵﾝ
        RET
        INCS    KEYCATT             ;ﾁｬﾀﾘﾝｸﾞ･ｶｳﾝﾀ+1
        NOP
        MOV     A,KEYCATT
        ADDS    A,#0DH              ;ｷｰが3回一致?
        RET
        MOV     A,#0H
        MOV     KEYCATT,A           ;ﾁｬﾀﾘﾝｸﾞ･ｶｳﾝﾀ←0H
        CLR1    KEYONF              ;KEYONF←0
        SET1    KEYCHKF             ;1ならばｷｰ･ｽｷｬﾝを終了
        CLR1    HTKEYONF
        RET
```

```
KEYCHK1:
        MOV     A,#1110B
        MOV     KEYS,A              ;ｷｰ･ｿｰｽ←1110B
KEYCHK2:
        OUT     PORT3,A             ;ﾎﾟｰﾄ3←1110B
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        IN      XA,PORT6
        MOV     KEYR,XA             ;ｷｰ･ﾘﾀｰﾝ←ﾎﾟｰﾄ6,7
        MOV     BC,#0FFH
        SKE     XA,BC               ;ﾎﾟｰﾄ6,7=ALL1?
        BR      KEYCHK3
        SET1    CY                  ;CY←1
        MOV     A,KEYS
        ADDC    XA,XA
        MOV     KEYS,A              ;ｷｰ･ｿｰｽを左へ1ﾋﾞｯﾄ･ｼﾌﾄ
        SKE     A,#0111B
        BR      KEYCHK2
        RET

KEYCHK3:
        MOV     A,KEYS
        MOV     HL,#KEYSB
        SKE     A,@HL               ;ｷｰ･ｿｰｽ=KEYSB?
        BR      KEYCHK23
        MOV     A,KEYR
        MOV     HL,#KEYRB
        SKE     A,@HL               ;ｷｰ･ﾘﾀｰﾝ=KEYRB?
        BR      KEYCHK23
        MOV     A,KEYR+1
        MOV     HL,#KEYRB+1
        SKE     A,@HL               ;ｷｰ･ﾘﾀｰﾝ+1=KEYRB+1?
        BR      KEYCHK23
        INCS    KEYCATT             ;ﾁｬﾀﾘﾝｸﾞ･ｶｳﾝﾀ+1
        NOP
        MOV     A,KEYCATT
        ADDS    A,#0DH              ;ｷｰが3回一致?
        RET
        MOV     A,KEYS
        SKE     A,#1110B            ;ｷｰ･ｿｰｽ=1110B?
        BR      KEYCHK11
        MOV     XA,KEYR
```

```
        MOV     HL,#KEYR_1
        SKE     XA,HL
        BR      KEYCHK4
        MOV     BC,#01H
        BR      KEYCHK22

KEYCHK4:
        MOV     HL,#KEYR_2
        SKE     XA,HL
        BR      KEYCHK5
        MOV     BC,#03H             ;KEYCODE←03H
        BR      KEYCHK22

KEYCHK5:
        MOV     HL,#KEYR_3
        SKE     XA,HL
        BR      KEYCHK6
        MOV     BC,#06H             ;KEYCODE←06H
        BR      KEYCHK22

KEYCHK6:
        MOV     HL,#KEYR_4
        SKE     XA,HL
        BR      KEYCHK7
        MOV     BC,#08H             ;KEYCODE←08H
        BR      KEYCHK22

KEYCHK7:
        MOV     HL,#KEYR_5
        SKE     XA,HL
        BR      KEYCHK8
        MOV     BC,#0AH             ;KEYCODE←0AH
        BR      KEYCHK22

KEYCHK8:
        MOV     HL,#KEYR_6
        SKE     XA,HL
        BR      KEYCHK9
        MOV     BC,#0DH             ;KEYCODE←0DH
        BR      KEYCHK22

KEYCHK9:
        MOV     HL,#KEYR_7
        SKE     XA,HL
        BR      KEYCHK10
        MOV     BC,#0FH             ;KEYCODE←0FH
        BR      KEYCHK22

KEYCHK10:
        MOV     HL,#KEYR_8
```

```
        SKE     XA,HL
        RET
        MOV     BC,#13H             ;KEYCODE←13H
        BR      KEYCHK22

KEYCHK11:
        SKE     A,#1101B            ;ｷｰ･ｿｰｽ=1101B?
        BR      KEYCHK14
        MOV     XA,KEYR
        MOV     HL,#KEYR_1
        SKE     XA,HL
        BR      KEYCHK12
        MOV     BC,#00H             ;KEYCODE←00H
        BR      KEYCHK22

KEYCHK12:
        MOV     HL,#KEYR_7
        SKE     XA,HL
        BR      KEYCHK13
        MOV     BC,#11H             ;KEYCODE←11H
        BR      KEYCHK22

KEYCHK13:
        MOV     HL,#KEYR_8
        SKE     XA,HL
        RET
        MOV     BC,#12H             ;KEYCODE←12H
        BR      KEYCHK22

KEYCHK14:
        SKE     A,#1011B            ;ｷｰ･ｿｰｽ=1011B?
        RET
        MOV     XA,KEYR
        MOV     HL,#KEYR_1
        SKE     XA,HL
        BR      KEYCHK15
        MOV     BC,#0CH             ;KEYCODE←0CH
        BR      KEYCHK22

KEYCHK15:
        MOV     HL,#KEYR_2
        SKE     XA,HL
        BR      KEYCHK16
        MOV     BC,#0BH             ;KEYCODE←0BH
        BR      KEYCHK22

KEYCHK16:
        MOV     HL,#KEYR_3
        SKE     XA,HL
        BR      KEYCHK17
```

```
        MOV     BC,#09H             ;KEYCODE←09H
        BR      KEYCHK22

KEYCHK17:
        MOV     HL,#KEYR_4
        SKE     XA,HL
        BR      KEYCHK18
        MOV     BC,#07H             ;KEYCODE←07H
        BR      KEYCHK22

KEYCHK18:
        MOV     HL,#KEYR_5
        SKE     XA,HL
        BR      KEYCHK19
        MOV     BC,#05H             ;KEYCODE←05H
        BR      KEYCHK22

KEYCHK19:
        MOV     HL,#KEYR_6
        SKE     XA,HL
        BR      KEYCHK20
        MOV     BC,#04H             ;KEYCODE←04H
        BR      KEYCHK22

KEYCHK20:
        MOV     HL,#KEYR_7
        SKE     XA,HL
        BR      KEYCHK21
        MOV     BC,#02H             ;KEYCODE←02H
        BR      KEYCHK22

KEYCHK21:
        MOV     HL,#KEYR_8
        SKE     XA,HL
        RET
        MOV     BC,#10H             ;KEYCODE←10H
KEYCHK22:
        SKF     HTKEYONF
        RET
        SET1    HTKEYONF            ;1ならば本体ｷｰ･ｵﾝ
        SET1    KEYONF
        MOV     XA,#10H
        SKE     XA,BC
        BR      KEYCHK24
        CLR1    KEYON2SF
        MOV     XA,#0H
        MOV     KEYTIM,XA
KEYCHK24:
        MOV     XA,BC
        MOV     KEYCODE,XA
```

```
        SET1    KEYCHKF             ;1ならばｷｰ･ｽｷｬﾝを終了
        RET

KEYCHK23:
        MOV     A,KEYS
        MOV     KEYSB,A             ;KEYSB←ｷｰ･ｿｰｽ
        MOV     XA,KEYR
        MOV     KEYRB,XA            ;KEYRB←ｷｰ･ﾘﾀｰﾝ
        MOV     A,#0H
        MOV     KEYCATT,A           ;ﾁｬﾀﾘﾝｸﾞ･ｶｳﾝﾀ←0
        RET

;++++++++++++++++++++++++++++++++
;           LED表示
;++++++++++++++++++++++++++++++++
LED1:
        SET1    CY
        MOV     A,LEDDIG
        RORC    A                   ;LEDDIGを右へ1ﾋﾞｯﾄ･ｼﾌﾄ
        MOV     LEDDIG,A
        SKT     CY
        BR      LED2

LEDMODE0:
        MOV     A,MODE
        SKE     A,#0H
        BR      LEDMODE1
        SKT     LEDDIG.3            ;"PROG"ﾓｰﾄﾞを表示
        BR      LED4P
        SKT     LEDDIG.2
        BR      LED5R
        SKT     LEDDIG.1
        BR      LED60
        MOV     HL,#LED_G           ;LED7←"G"を表示
LED4P:
        MOV     HL,#LED_P           ;LED4←"P"を表示
LED5R:
        MOV     HL,#LED_R           ;LED5←"R"を表示
LED60:
        MOV     HL,#LED_0           ;LED6←"0"を表示
        BR      LED_OUT

LED2:
        MOV     A,#0111B
        MOV     LEDDIG,A            ;LEDDIGを初期化
        BR      LEDMODE0

LEDMODE1:
        SKE     A,#1H
        BR      LEDMODE2
```

```
        SKT     LEDDIG.3            ;"PLAY"ﾓｰﾄﾞを表示
        BR      LED4P1
        SKT     LEDDIG.2
        BR      LED5L
        SKT     LEDDIG.1
        BR      LED6A
        MOV     HL,#LED_Y           ;LED7←"Y"を表示
LED4P1:
        MOV     HL,#LED_P           ;LED4←"P"を表示
LED5L:
        MOV     HL,#LED_L           ;LED5←"L"を表示
LED6A:
        MOV     HL,#LED_A           ;LED6←"A"を表示
        BR      LED_CNT

LEDMODE2:
        SKT     LEDDIG.3            ;"PAUS"ﾓｰﾄﾞを表示
        BR      LED4P1
        SKT     LEDDIG.2
        BR      LED5A
        SKT     LEDDIG.1
        BR      LED6U
        MOV     HL,#LED_S           ;LED7←"S"を表示
LED5A:
        MOV     HL,#LED_A           ;LED5←"A"を表示
LED6U:
        MOV     HL,#LED_U           ;LED6←"U"を表示
LED_CNT:
        MOV     A,DISPTIM
        ADDS    A,#0FH
        BR      LED_OUT
        SKT     LEDDIG.3            ;基本時間を表示
        BR      LED40
        SKT     LEDDIG.2
        BR      LED5TIM
        MOV     HL,#LED_0           ;LED6,7←"0"を表示
LED40:
        MOV     HL,#LED__           ;LED4←"_"を表示
        BR      LED_OUT

LED5TIM:
        MOV     X,#0H               ;LED5←時間の100の位を表示
        MOV     A,T_TIM
        MOV     BC,#HIGH LED_T
        CALL    !TABLE
        MOV     HL,XA
LED_OUT:
        MOV     A,#1111B
        OUT     PORT3,A             ;ﾃﾞｨｼﾞｯﾄ出力をｵﾌ
        MOV     XA,HL
```

```
OUT     PORT4,XA            ;ｾｸﾞﾒﾝﾄを出力
MOV     A,LEDDIG
OUT     PORT3,A             ;ﾃﾞｨｼﾞｯﾄ出力
RET
```

```
$TT='TABLE'
;**********************************
;***     テーブル参照エリア        ***
;**********************************

RM          CSEG    PAGE

RM_T:
;++++++++++++++++++++++++++++++++++
;+++      RMDATA-->KEYCODE        +++
;++++++++++++++++++++++++++++++++++
DB      10H                     ;MODE
DB      11H                     ;UP
DB      12H                     ;DOWN
DB      13H                     ;END
DB      0FH                     ;WAIT
DB      0DH                     ;DO#  +
DB      0AH                     ;RA#  +
DB      08H                     ;SO#
DB      06H                     ;FA#
DB      03H                     ;RE#
DB      01H                     ;DO#
DB      0CH                     ;DO   +
DB      0BH                     ;SI   +
DB      09H                     ;RA   +
DB      07H                     ;SO
DB      05H                     ;FA
DB      04H                     ;MI
DB      02H                     ;RE
DB      00H                     ;DO


TONE_C      CSEG    PAGE

TONE_T:
;++++++++++++++++++++++++++
;+++    音階ﾃﾞｰﾀ         +++
;++++++++++++++++++++++++++
DB      0F9H                    ;DO
DB      0EBH                    ;DO#
DB      0DEH                    ;RE
DB      0D1H                    ;RE#
DB      0C6H                    ;MI
DB      0BAH                    ;FA
DB      0B0H                    ;FA#
DB      0A6H                    ;SO
DB      09CH                    ;SO#
DB      094H                    ;RA   +
DB      08BH                    ;RA#  +
DB      083H                    ;SI   +
```

```
DB      07CH                    ;DO  +
DB      075H                    ;DO#  +
DB      0FFH                    ;無音
DB      0FFH                    ;無音


LCD_C       CSEG    PAGE

LCD_T:
;++++++++++++++++++++++++++++++++++
;+++         LCD表示ﾊﾟﾀｰﾝ          +++
;++++++++++++++++++++++++++++++++++
DB      0EH                     ;"DO  "
DB      19H
DB      2EH
DB      2EH

DB      0EH                     ;"DO#  "
DB      19H
DB      25H
DB      2EH

DB      1CH                     ;"RE  "
DB      0FH
DB      2EH
DB      2EH

DB      1CH                     ;"RE#  "
DB      0FH
DB      25H
DB      2EH

DB      17H                     ;"MI  "
DB      13H
DB      2EH
DB      2EH

DB      10H                     ;"FA  "
DB      0BH
DB      2EH
DB      2EH

DB      10H                     ;"FA#  "
DB      0BH
DB      25H
DB      2EH

DB      1DH                     ;"SO  "
DB      19H
DB      2EH
DB      2EH
```

```
DB      1DH                     ;"SO#  "
DB      19H
DB      25H
DB      2EH

DB      1CH                     ;"RA  +"
DB      0BH
DB      2EH
DB      26H

DB      1CH                     ;"RA#  +"
DB      0BH
DB      25H
DB      26H

DB      1DH                     ;"SI  +"
DB      13H
DB      2EH
DB      26H

DB      0EH                     ;"DO  +"
DB      19H
DB      2EH
DB      26H

DB      0EH                     ;"DO#  +"
DB      19H
DB      25H
DB      26H

DB      21H                     ;"WAIT  "
DB      0BH
DB      13H
DB      1EH

DB      0FH                     ;"END  "
DB      18H
DB      0EH
DB      2EH

LED_C       CSEG    PAGE

LED_T:
```

```
;++++++++++++++++++++++++++
;+++    LEDｾｸﾞﾒﾝﾄ        +++
;++++++++++++++++++++++++++
;Address    abcdefg.         Display
DB          11000101B        ;"0"
DB          10011111B        ;"1"
DB          00100101B        ;"2"
DB          00001101B        ;"3"
DB          10011001B        ;"4"
DB          01001001B        ;"5"
DB          01000001B        ;"6"
DB          00011011B        ;"7"
DB          00000001B        ;"8"
DB          00001001B        ;"9"


;++++++++++++++++++++++++++++++++++
;+++         ﾃｰﾌﾞﾙ参照ｻﾌﾞﾙｰﾁﾝ     +++
;++++++++++++++++++++++++++++++++++
TABLEINC:
        INCS    DE          ;Table Address  +1
        MOV     XA,DE
TABLE:
        MOVT    XA,@BCXA
        RET
```

〔メ　モ〕

アンケート記入のお願い

お手数ですが，このドキュメントに対するご意見をお寄せください。今後のドキュメント作成の参考にさせていただきます。

［ドキュメント名］ μPD750008サブシリーズ　アプリケーション・ノート
（U10452JJ1V0AN00（第1版））

［お名前など］（さしつかえのない範囲で）

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（　　）

1．ご評価（各欄に○をご記入ください）

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（　　）					
（　　）					

2．わかりやすい所（第　章，第　章，第　章，第　章，その他　　）
理由［　　］

3．わかりにくい所（第　章，第　章，第　章，第　章，その他　　）
理由［　　］

4．ご意見，ご要望

5．このドキュメントをお届けしたのは
NEC販売員，特約店販売員，NEC半導体ソリューション技術本部員，
その他（　　）

ご協力ありがとうございました。
下記あてにFAXで送信いただくか，最寄りの販売員にコピーをお渡しください。

NEC半導体インフォメーションセンター
FAX：（044）548-7900

キリトリ

お問い合わせは，最寄りのNECへ

【営業関係お問い合わせ先】

半導体第一販売事業部 半導体第二販売事業部 半導体第三販売事業部	〒108-01	東京都港区芝五丁目7番1号（NEC本社ビル）	東　京 (03)3454-1111 (大代表)
中部支社 半導体販売部	〒460	名古屋市中区錦一丁目17番1号（NEC中部ビル）	名古屋 (052)222-2170
関西支社 半導体第一販売部 関西支社 半導体第二販売部 関西支社 半導体第三販売部	〒540	大阪市中央区城見一丁目4番24号（NEC関西ビル）	大　阪 (06) 945-3178 大　阪 (06) 945-3200 大　阪 (06) 945-3208

北海道支社	札　幌 (011)231-0161	小山支店	小　山 (0285)24-5011	三重支店	津 (0592)25-7341
東北支社	仙　台 (022)261-5511	長野支社	長　野 (0262)35-1444	京都支社	京　都 (075)344-7824
岩手支店	盛　岡 (0196)51-4344	松本支店	松　本 (0263)35-1666	神戸支社	神　戸 (078)333-3854
山形支店	山　形 (0236)23-5511	上諏訪支店	諏　訪 (0266)53-5350	中国支社	広　島 (082)242-5504
郡山支店	郡　山 (0249)23-5511	甲府支店	甲　府 (0552)24-4141	鳥取支店	鳥　取 (0857)27-5311
いわき支店	いわき (0246)21-5511	埼玉支社	大　宮 (048)641-1411	岡山支店	岡　山 (086)225-4455
長岡支店	長　岡 (0258)36-2155	立川支社	立　川 (0425)26-5981	四国支社	高　松 (0878)36-1200
土浦支店	土　浦 (0298)23-6161	千葉支社	千　葉 (043)238-8116	新居浜支店	新居浜 (0897)32-5001
水戸支店	水　戸 (0292)26-1717	静岡支社	静　岡 (054)255-2211	松山支店	松　山 (0899)45-4111
神奈川支社	横　浜 (045)324-5511	浜松支店	浜　松 (053)452-2711	九州支社	福　岡 (092)271-7700
高崎支店	高　崎 (0273)26-1255	北陸支社	金　沢 (0762)23-1621	北九州支店	北九州 (093)541-2887
太田支店	太　田 (0276)46-4011	福井支店	福　井 (0776)22-1866		
宇都宮支店	宇都宮 (0286)21-2281	富山支店	富　山 (0764)31-8461		

【本資料に関する技術お問い合わせ先】

半導体ソリューション技術本部 マイクロコンピュータ技術部	〒210	川崎市幸区塚越三丁目484番地	川　崎	(044)548-7923
半導体販売技術本部 東日本販売技術部	〒108-01	東京都港区芝五丁目7番1号（NEC本社ビル）	東　京	(03)3798-9619
半導体販売技術本部 中部販売技術部	〒460	名古屋市中区錦一丁目17番1号（NEC中部ビル）	名古屋	(052)222-2125
半導体販売技術本部 西日本販売技術部	〒540	大阪市中央区城見一丁目4番24号（NEC関西ビル）	大　阪	(06) 945-3383

半導体
インフォメーションセンター
FAX(044)548-7900
（FAXにてお願い致します）

C95.9