

お客様各位

カタログ等資料中の旧社名の扱いについて

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

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

2010年4月1日
ルネサスエレクトロニクス株式会社

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

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

ご注意書き

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

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

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

パーソナル無線用ソフトウェア作成技法



パーソナル無線用ソフトウェア作成技法

NEC 日本電気株式会社

目 次

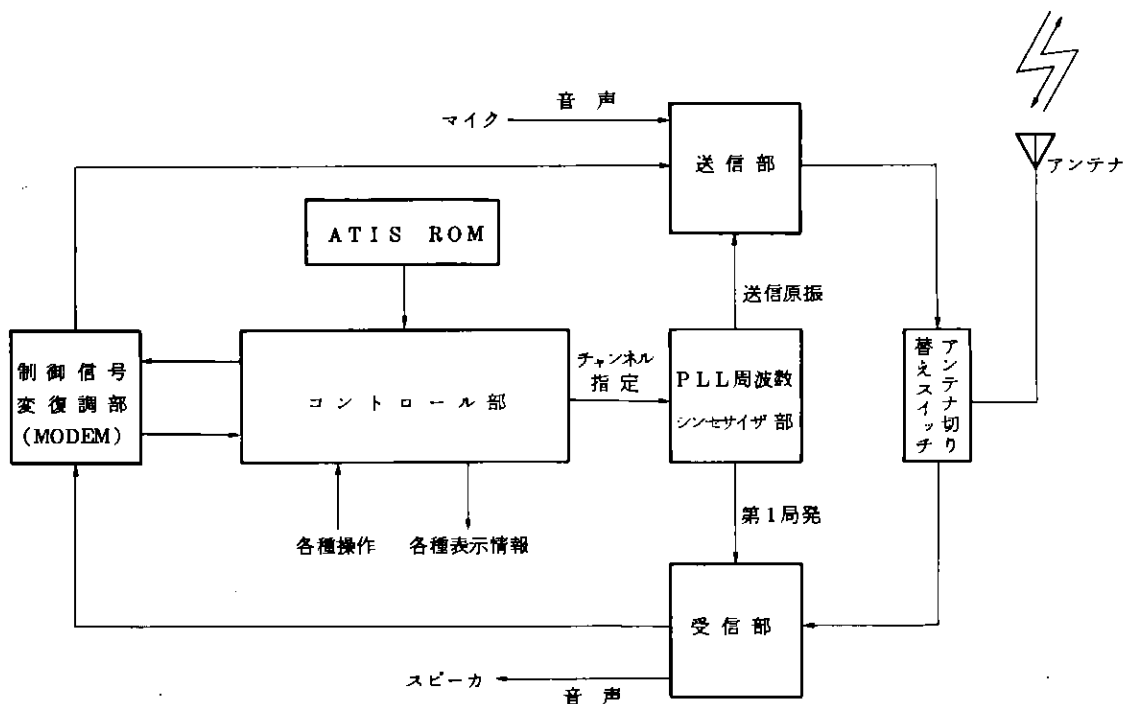
第1章 概 説	1
第2章 PLL周波数シンセサイザ・インタフェース	4
2.1 PLLデータ生成サブルーチン	7
2.1.1 受信時(SRPLL)	7
2.1.2 送信時(STPLL)	10
2.2 PLLデータ出力サブルーチン	10
第3章 LCDドライバ・インタフェース	15
3.1 LCDドライバ μ PD6320インタフェース・サブルーチン	15
3.2 LCDドライバ μ PD7225インタフェース・サブルーチン	20
第4章 MODEMインタフェース	28
4.1 送信処理	29
4.2 受信処理	43
4.2.1 受信ルーチン	43
4.2.2 ハーゲルバーガ復号化サブルーチン(SJD00)	54
第5章 ATIS ROMインタフェース・サブルーチン(SROM0)	61
第6章 参考資料	66

第1章 概 説

パーソナル無線は、周知のように900MHz帯の80チャンネルの周波数を利用した、空中線電力5W以下の周波数変調方式の簡易無線です。これは、周波数を制御するPLL^{注(1)}シンセサイザ回路のLSI化^{注(2)}MCAシステムの確立、各種ICおよびマイコン制御技術の発達により実現が可能となりました。また今後低消費電力化、機械の向上、小型化などの技術開発が進む事が予測されます。

さて、パーソナル無線は一般に図1.1に示すシステム構成となっています。本書では、パーソナル無線のコントロール部の基本的な制御処理であるPLL周波数シンセサイザ・インタフェース、LCDドライバ・インタフェース、MODEMインタフェース、ATIS ROM^{注(3)}インタフェースについて、マイクロコンピュータで実現する方法についてまとめました。なお、本書ではマイクロコンピュータとしてμPD7508を使用しています。このμPD7508を中心とした情報の流れを示すI/O構成図を図1.2に示します。

図1.1 パーソナル無線システム構成図



注1 PLL(Phase-Locked Loop:位相同期ループ)

発振器の周波数および位相がつかねに入力信号の周波数、位相と一致するように、位相差を検出してフィード・バックにより制御する回路構成

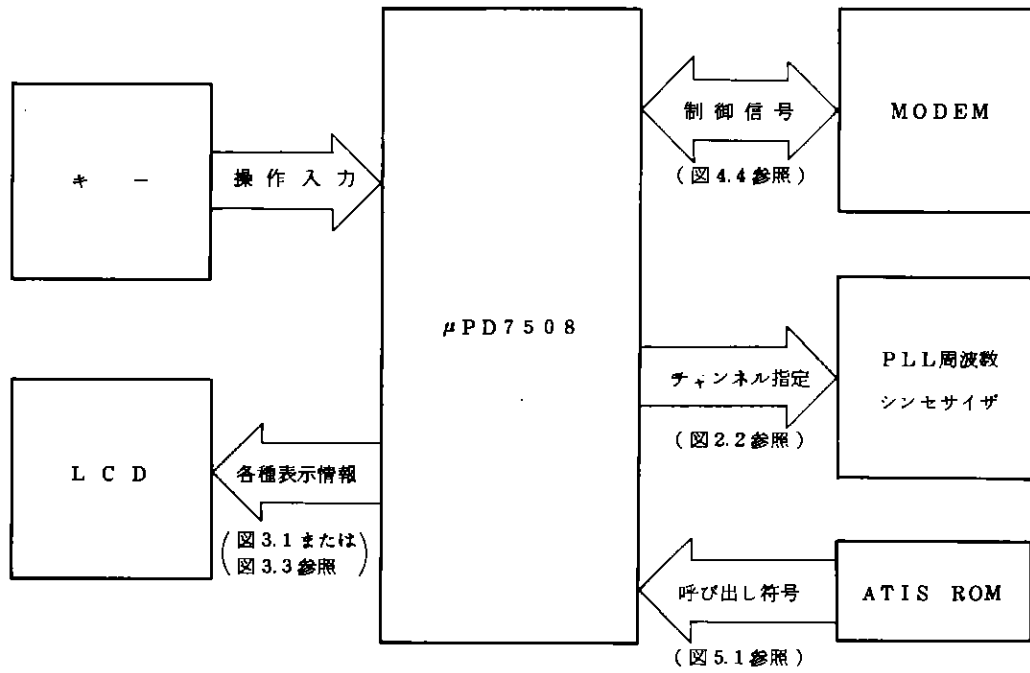
注2 MCA(Multichannel Access)システム

限られたチャンネルを不特定多数の局が共同利用できるシステム。

注3 ATIS(Automatic Transmitter Identification System:自動識別符号発射方式)

交信時に自動的に識別符号を発射する方式。

図 1.2 μ PD7508 I/O 構成図



次に、本書のプログラムに使用するデータ・メモリのレイアウトを図 1.3 に示します。なお、データ・メモリの使用方法は、各章のデータ・メモリ構成を参照してください。

図 1.3 データ・メモリ・レイアウト

L/H	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0												RTIM1 タイマ・カウンタ	RTIM2 ACC 経過エリフ	RINTT チャネル番号	RCH01	RCH02
1											RPLL1	RPLL2	RPLL3 PLL データ・エリフ	RPLL4	RPLL5	RPLL6
2														RLC71	RLC72	RLC73
3																
4																
5									RLCD1	RLCD2	RLCD3 μPD6320 用LCD	RLCD4	RLCD5	RLCD6 データ・エリフ	RLCD7	RLCD8
6	RINT1 RINT0 ACC 経過エリフ	ROUT1						送	信							
7	RROM0										RROMA				RTMNG	RBITL
8	RIN01	RIN02	RIN03	RIN04		RIN06 受	信					納				
9	RIN17							送	信	RIN24	RIOCH	RIOCL	RBITC	RADRH	RADRL	RTRNS
A	RFLG1	RFLG2				RDECD										RPT0
B																
C																
D																

第2章 PLL周波数シンセサイザ・インタフェース

図1.1に示すPLL周波数シンセサイザ部は、送信時に必要な送信原振80チャンネルと受信時に必要な第1局発周波数80チャンネルの合計160チャンネルの周波数を作り出します。

PLL周波数シンセサイザにはいくつかの方式がありますが、ここでは図2.1に示す二つの分周比を持つプリスケラ方式について説明します。

この方式では出力周波数 f_0 は、プリスケラの分周比 K とマイクロコンピュータからの N_p および A により(式2.1)にしたがって分周されます。
※参考資料(1)

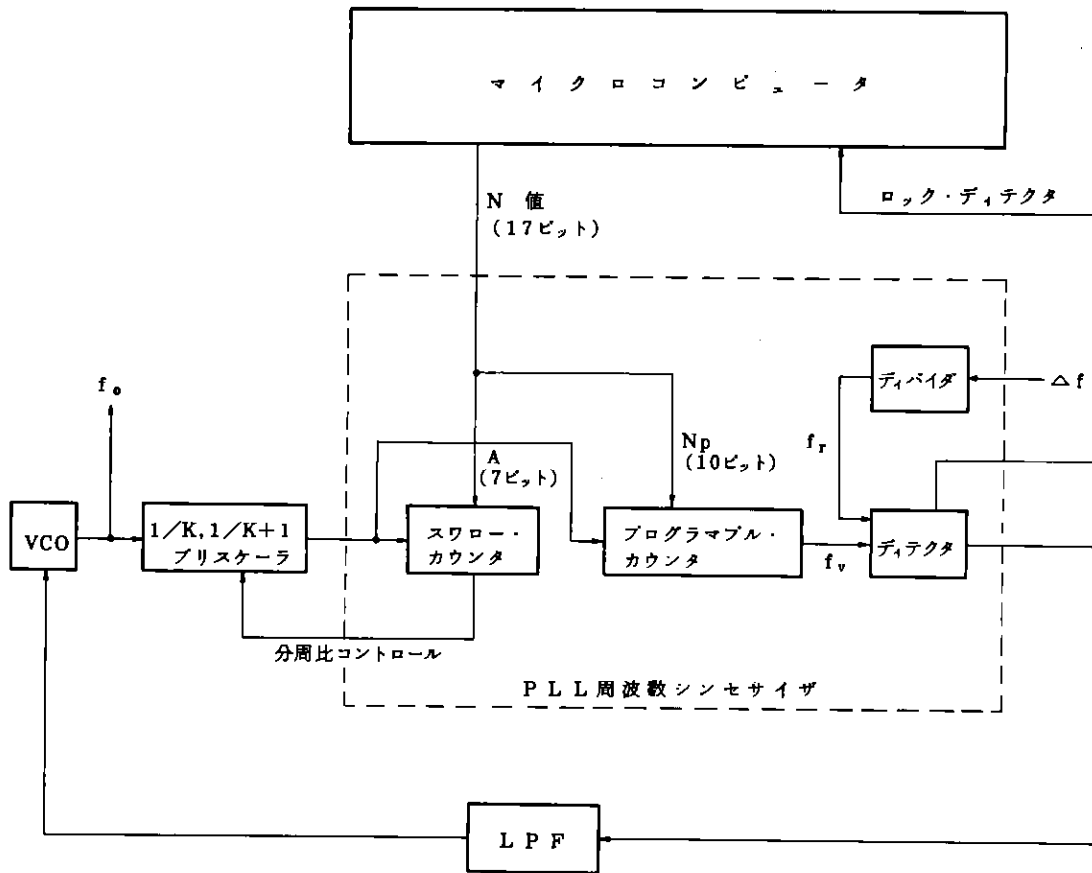
$$f_v = f_0 / (N_p \cdot K + A) \dots\dots\dots (式2.1)$$

- f_0 : 出力周波数
- Δf : チャンネル・スペース
- f_r : 基準周波数
- f_v : 分周周波数
- A : スワロー・カウンタの分周化
- N_p : プログラマブル・カウンタの分周化
- K : プリスケラの分周化

この分周周波数 f_v と基準周波数 f_r との位相差の量に応じて、 f_0 が漸次変化します。そして $f_v = f_r$ となった時に f_0 がロックします。以上の動作の詳細は、参考資料(1)を参照してください。

なお、 N_p と A をマイクロコンピュータで設定する場合には、 N_p を上位10ビット、 A を下位7ビットとして合成した N 値(17ビット)となります。この N 値の設定方法を受信時と送信時とにわけて以下に説明します。
※参考資料(2)
ただし、PLL周波数シンセサイザに μ PD2833C、プリスケラに μ PB566Cを使用し、 $K=128$ 、 $\Delta f=12.8\text{MHz}$ 、ディバイダの分周比 $1/1024$ 、 $f_r=0.0125\text{MHz}$ とします。

図 2.1 PLL周波数シンセサイザ・システム構成図



○受信時

※参考資料(6)

受信時の出力周波数 f_0 には (式 2.2) で示される値が必要です。

$$f_0 = \text{受信周波数} + 58.1125\text{MHz} \dots\dots\dots (\text{式 2.2})$$

なお、受信周波数の範囲は 0.025MHz 間隔で 903.0125MHz から 904.9875MHz であるため、 f_0 の範囲は 0.025MHz 間隔の 961.125MHz から 963.1MHz までになります。

この f_0 を (式 2.1) に代入して $f_v = f_r = 0.0125\text{MHz}$ となる時の N_p と A の値の求め方をチャンネル番号 = 1 ($f_0 = 961.125\text{MHz}$) を例にして以下に説明します。

(式 2.1) を A で解くと

$$\begin{aligned} A &= f_0 / f_v - N_p \cdot K \\ &= f_0 / 0.0125 - 128 \cdot N_p \dots\dots\dots (\text{式 2.3}) \end{aligned}$$

ここで A は 7 ビットであるから $0 \leq A \leq 127$

ゆえに

$$\begin{aligned} 0 &\leq f_0 / 0.0125 - 128 \cdot N_p \leq 127 \\ (80 \cdot f_0 - 127) / 128 &\leq N_p \leq f_0 / 1.6 \dots\dots\dots (\text{式 2.4}) \end{aligned}$$

(式 2.4) に $f_0 = 961.125\text{MHz}$ を代入すると

$$599.711 \leq N_p \leq 600.703$$

$$\therefore N_p = 600$$

この N_p と f_0 を (式 2.3) に代入すると

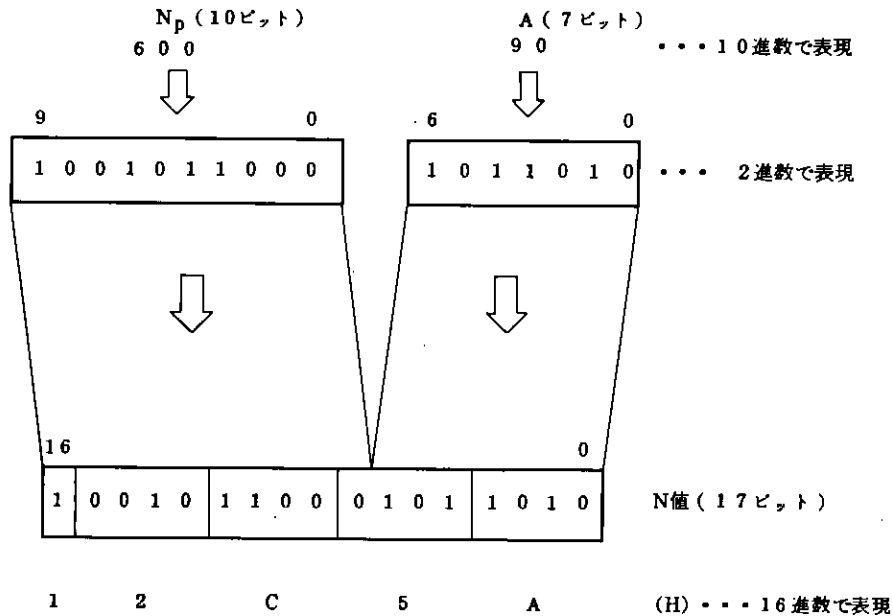
$$A = 90$$

となります。

また、この時の N_p を上位 10 ビット、 A を下位 7 ビットの 2 進数として合成した N 値は、図 2.2 に示すように 12C5AH となります。

一般に N 値が 1 変化すると出力周波数 f_0 は基準周波数 f_r 分変化します。ここでは $f_r = 0.0125\text{MHz}$ であるため、 0.025MHz 間隔のチャンネル番号を 1 変化させるためには、 N 値を 2 変化させる必要があります。ですからチャンネル番号 (CH) と N 値 (RxPLL) の関係は (式 2.5) のとおりになります。

図 2.2 N 値の合成



$$RxPLL = (1 \text{ チャンネル } N \text{ 値}) + 2(CH - 1)$$

$$= 12C5AH + 2(CH - 1) \dots \dots \dots \text{(式 2.5)}$$

○送信時

送信時の出力周波数 f_0 の範囲は、 0.025MHz 間隔の 903.0125MHz から 904.9875MHz までです。ここから受信時と同様にチャンネル番号 (CH) と N 値 (TxPLL) の関係を求めると (式 2.6) のとおりになります。

$$TxPLL = 11A31H + 2(CH - 1) \dots \dots \dots \text{(式 2.6)}$$

2.1 PLL データ生成サブルーチン

2.1.1 受信時 (SRPLL)

(1) 処理内容

チャンネル番号 (CH) より受信時 N 値 (RxPLL) を (式 2.3) にしたがって算出し, PLL データ・エリアへ格納します。ただし, CH は 01H ~ 50H の 2 桁の 16 進数とし, RxPLL は 5 桁の 16 進数とします。

(2) データ・メモリ構成

		COLUMN ADDRESS					
		A	B	C	D	E	F
ROW ADDRESS	0					RCH01	RCH02
	1		RPLL2	RPLL3	RPLL4	RPLL5	RPLL6

RCH01: チャンネル番号の上位桁

RCH02: チャンネル番号の下位桁

RPLL2: N 値の上位桁

}

} PLL データ・エリア

RPLL6: N 値の下位桁

(3) 入力条件

RCH01, RCH02 にチャンネル番号が格納されていること。

(4) 出力条件

N 値が RPLL2 ~ RPLL6 に格納されます。

(5) 変化するレジスタ

A, H, L, C

(6) 処理説明

a) 1 チャンネル・データ (12C5AH) を PLL データ・エリア (RPLL2 ~ RPLL6) にセットします。

b) チャンネル NO. (RCH01, RCH02) - 1 の 2 倍を PLL データ・エリアへ加算するために下記の操作を 2 回行ないます。

b)-1 (RCH02) をデクリメントします。

b)-2 その結果が 0FH であれば (RPLL5) をデクリメントします。

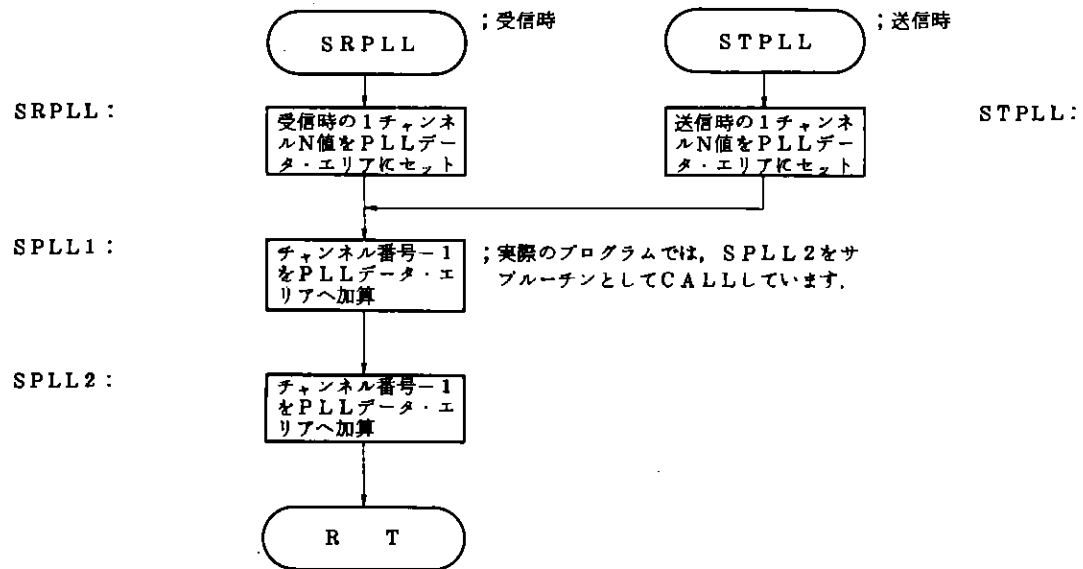
b)-3 (RPLL6) と (RCH02) を加算します。

b)-4 (RPLL5) と (RCH01) を加算します。この際, b)-3 による Carry を考慮します。

(7) フローチャート

PROGRAM LABEL

PROGRAM LABEL



(8) プログラム

バイト数 4 1

```
*****
;*
;*      SRPLL - GENERATE PLL DATA SUBROUTINE
;*      RXPLL = 12C5AH + 2(CH-1)
;*
;*      INPUTS   : RCH01,02 ... CHANNEL NO.
;*      OUTPUTS  : RPLL2-6 ... CONVERT PLL DATA
;*      DESTROYS : A,H,L,C
;*
*****
SRPLL:
    LHLI    RPLL6      ; SET 1 CHANNEL RXPLL DATA
    LAI     0AH        ;
    XAM     HL-        ; RPLL6 <-- 0AH
    LAI     5          ;
    XAM     HL-        ; RPLL5 <-- 5
    LAI     0CH        ;
    XAM     HL-        ; RPLL4 <-- 0CH
    LAI     2          ;
    XAM     HL-        ; RPLL3 <-- 2
    GJMP    SPL11      ;
;
*****
;*
;*      STPLL - GENERATE PLL DATA SUBROUTINE
;*      TXPLL = 11A31H + 2(CH-1)
;*
;*      INPUTS   : RCH01,02 ... CHANNEL NO.
;*      OUTPUTS  : RPLL2-6 ... CONVERT PLL DATA
;*      DESTROYS : A,H,L,C
;*
*****
STPLL:
    LHLI    RPLL6      ; SET 1 CHANNEL TXPLL DATA
    LAI     1          ;
    XAM     HL-        ; RPLL6 <-- 1
    LAI     3          ;
    XAM     HL-        ; RPLL5 <-- 3
    LAI     0AH        ;
    XAM     HL-        ; RPLL4 <-- 0AH
    LAI     1          ;
    XAM     HL-        ; RPLL3 <-- 1
;
SPLL1:
    SMB     0          ; RPLL2 <-- 1
    CALL    SPL12      ;
SPLL2:
    RC      ;
    LHLI    RPLL6      ;
    LADR    RCH02      ;
    AISC    0FH        ; ACC <-- RCH02 - 1
    DDRS    RPLL5      ;
SPLL3:
    ACSC    ;
    NOP     ;
    XAM     HL-        ;
    SKLEI   RPLL5 MOD 10H ;
    RT      ;
    LADR    RCH01      ;
    GJMP    SPL13      ;
```

2.1.2 送信時(STPLL)

(1) 処理内容

チャンネル番号(CH)より送信時N値(TxPLL)を(式2.4)にしたがって算出し、PLLデータ・エリアへ格納します。ただし、CHは01H~50Hの2桁の16進数とし、RxPLLは5桁の16進数とします。

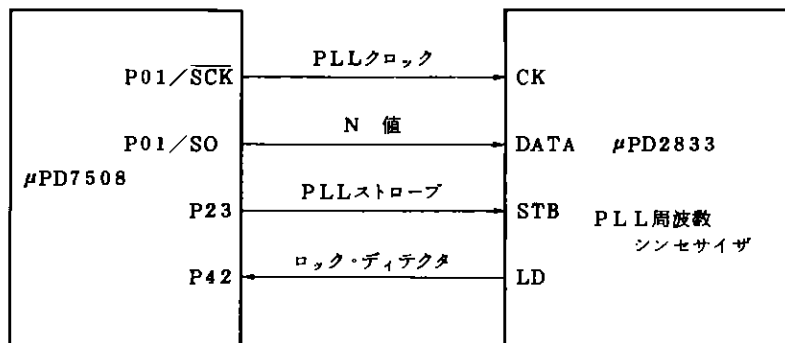
(2)~(8) は2.1.1を参照してください。

2.2 PLLデータ出力サブルーチン

(1) 処理内容

PLLデータ生成サブルーチンで求めたN値をμPD7508のシリアル・インタフェースを介してμPD2833に出力します。図2.3にμPD2833とμPD7508とのインタフェース構成を示します。

図2.3 PLL周波数シンセサイザμPD2833インタフェース構成図



このインタフェースにおけるμPD7508の各端子の機能は次のとおりです。

- P01/SCK : シリアル・クロックの出力に使用します。
- P02/SO : シリアル・データ(N値)の出力に使用します。
- P23 : ストロブ信号の出力に使用します。
- P42 : ロック・ディテクタ信号の入力に使用します。

(2) データ・メモリ構成

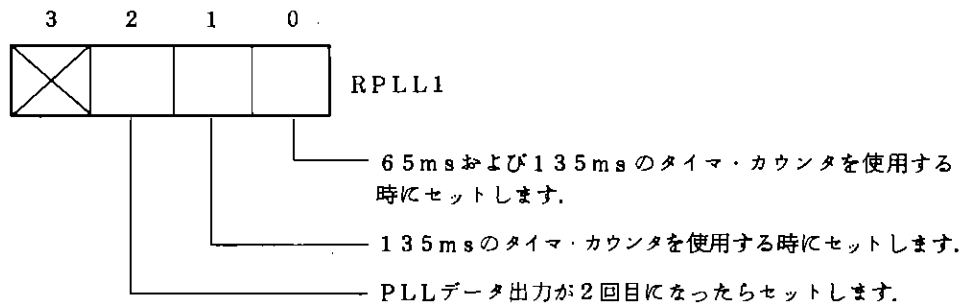
		COLUMN ADDRESS					
		A	B	C	D	E	F
ROW ADDRESS	0		RTIM1	RTIM2	RINTT		
	1	RPLL1	RPLL2	RPLL3	RPLL4	RPLL5	RPLL6

RTIM1: タイマ・カウンタ }
 RTIM2: タイマ・カウンタ } 5ms ごとのタイマ割込みでデクリメントされます。

RINTT: INTT発生時のAcc 逃避エリア

RPLL2: N値の上位桁 }
) } PLLデータ・エリア
 RPLL6: N値の下位桁 }

RPLL1: フラグ・エリア



(3) 入力条件

N値がPLLデータ・エリア(RPLL2~RPLL6)に格納されていること。

(4) 出力条件

RT : ディテクタがロックしなかった時。

RTS : ディテクタがロックした時。

(5) 変化するレジスタ

A, H, L

(6) 処理説明

- シリアル・インタフェースを介して、RPLL1~RPLL6のデータを3回に分けて出力します。
- PLLストロブ信号を出力します。
- 65ms ウェイトします。
- ロック・ディテクタ信号のチェックをし、ディテクタがロックしたならば(P42='H') RTSでメインに戻ります。
- c) から135ms経過するまでd)の判断を行いません。

f) この操作が2回目ならばRTでメインに戻ります。1回目ならばb)に戻ります。
ここで使用しているタイマ・カウンタ(RTIM1, RTIM2)は、5msごとのタイマ割込みにより次の方法でデクリメントされます。

① RPLL1₍₀₎^{*}がセットされていればRTIM1をデクリメントします。

その結果、Borrowが出なければメインに戻り、出ればb)の操作を行いません。

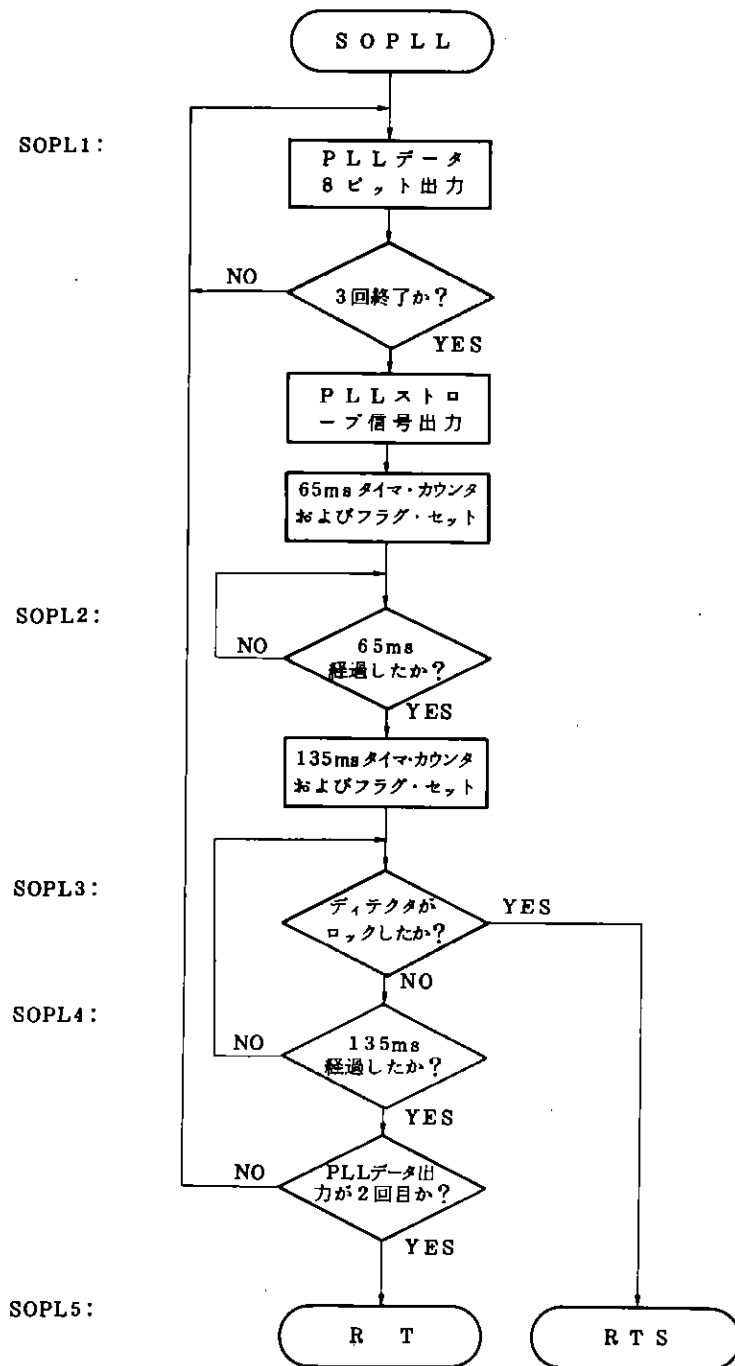
② RPLL1₍₁₎^{*}がセットされていなければ、RPLL1₍₀₎をリセットします。

またセットされていればRTIM2をデクリメントします。その結果Borrowが出ればRPLL1₍₀₎, RPLL1₍₁₎をリセットします。

* RPLL1₍₀₎またはRPLL₍₁₎の(0), (1)はビット数を示します。

(7) フローチャート

PROGRAM
LABEL



(8) プログラム

バイト数 5 5

```
*****
;*
;*      SOPLL - OUTPUT PLL DATA SUBROUTINE
;*      SCK ... OUTPUT PLL CLOCK
;*      SO ... OUTPUT PLL DATA
;*      P23 ... OUTPUT PLL STROBE
;*      P42 ... INPUT LOCK DETECTOR
;*
;*      INPUTS : RPLL1-6 ... GENERATE PLL DATA
;*      OUTPUTS : RT ... DETECTOR UNLOCK
;*               : RTS ... DETECTOR LOCK
;*      DESTROYS : A,H,L
;*
*****
;
SOPLL:
LHLI    RPLL1      ;
SOPL1:
LAM     HL+        ;
TAMSIO          ;
SIO          ; OUTPUT PLL DATA & CLOCK
NOP          ;
NOP          ; WAIT 6 MACHINE CYCLE
NOP          ;
ILS          ;
GJMP     SOPL1     ;
NOP          ; WAIT 8 MACHINE CYCLE
NOP          ;
NOP          ;
ORP      2.1000B   ; OUTPUT PLL STROBE
ANP      2,0111B   ;
SK1      0001B    ; RESET INTT RQF.
NOP          ;
LAI      12        ;
LADR     RTIM1     ; SET 65MS TIMER COUNTER
LHLI     RPLL1     ;
SMB      0         ; SET 65MS TIMER COUNTER FLAG
SOPL2:
SKMBF    0         ;
GJMP     SOPL2     ; LOOP 65MS
LHLI     RTIM2     ;
LAI      1         ;
XAM      HL-       ;
LAI      0AH       ;
ST       ; SET 135MS TIMER COUNTER
LHLI     RPLL1     ;
SMB      0         ;
SMB      1         ; SET 135MS TIMER COUNTER FLAG
SOPL3:
IP       4         ; INPUT LOCK DETECTOR
SKABT    2         ; SKIP IF UNLOCKED DETECTOR
GJMP     SOPL4     ;
RMB      0         ;
RMB      1         ; RESET 135MS TIMER COUNTER FLAG
RTS      ; DETECTOR LOCK
SOPL4:
SKMBF    1         ; SKIP IF RESET 135MS TIMER COUNTER FLAG
GJMP     SOPL3     ; LOOP 135MS
SKMBF    2         ; SKIP IF FIRST OUTPUT PLL
GJMP     SOPL5     ;
SMB      2         ; SET SECOND OUTPUT PLL FLAG
GJMP     SOPLL     ;
SOPL5:
RMB      2         ; RESET SECOND OUTPUT PLL FLAG
RT       ; DETECTOR UNLOCK
```

第3章 LCDドライバ・インタフェース

パーソナル無線には、5桁の群番号、待受、通話、モニタ、送信、メモリ等の表示要素があります。
※参考資料(3) ※参考資料(4)
 本書では、LCDドライバに μ PD6320または μ PD7225を使用した場合について説明します。

3.1 LCDドライバ μ PD6320インタフェース・サブルーチン

(1) 処理内容

群番号、待受、通話、モニタ、送信を表示するためのデータを μ PD7508のシリアル・インタフェースを介して μ PD6320に出力します。図3.1に μ PD7508と μ PD6320とのインタフェース構成を示します。図3.2にLCDパネルの結線方法を示します。

図3.1 LCDドライバ μ PD6320インタフェース構成図

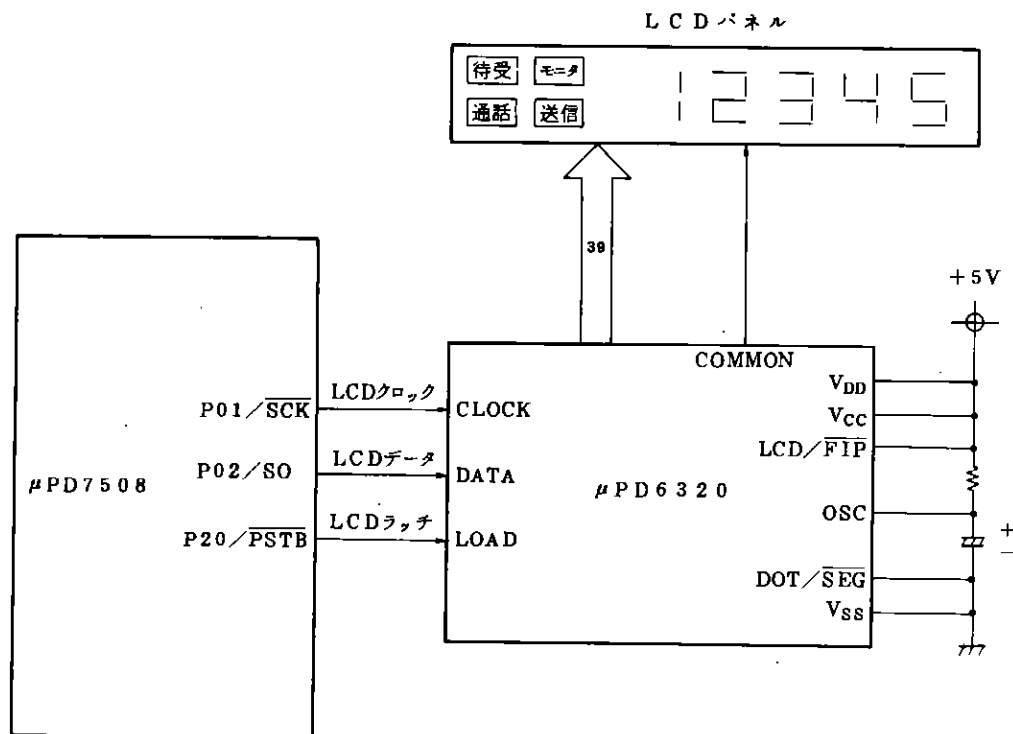
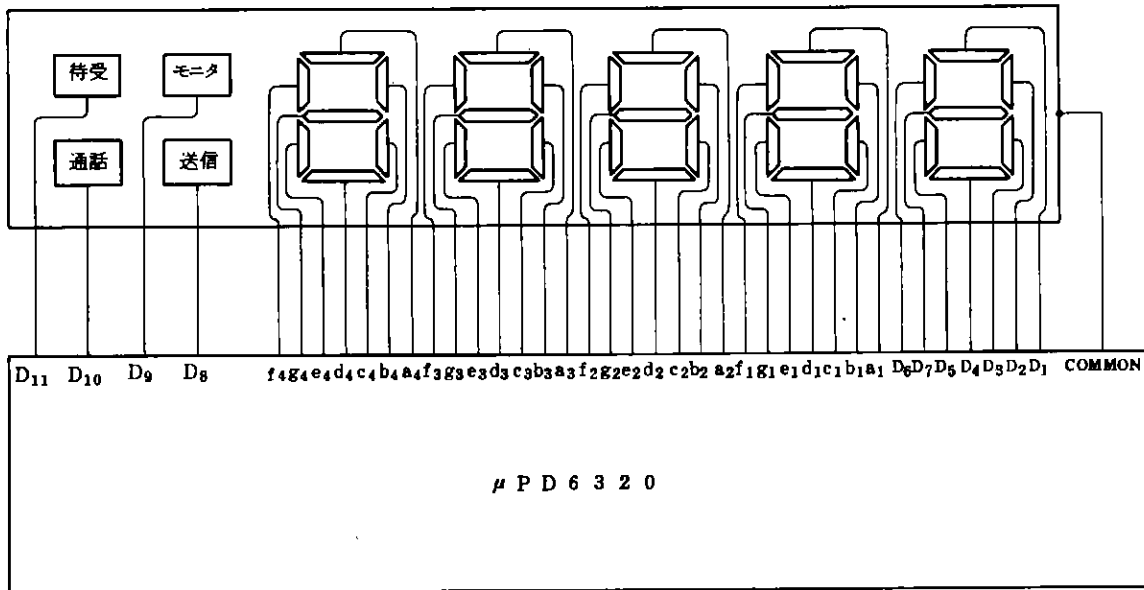


図 3.2 LCDパネル結線図



(2) データ・メモリ構成

		COLUMN ADDRESS							
		8	9	A	B	C	D	E	F
ROW ADDRESS	5	RLCD1	RLCD2	RLCD3	RLCD4	RLCD5	RLCD6	RLCD7	RLCD8

RLCD1

ビット3：μPD6320の第4桁の7SEGデコーダに'F'のデコーダでblank表示をさせるために'0'をセットします。
※参考資料(3)

ビット2：'待受'表示時にセットします。

ビット1：'通話'表示時にセットします。

ビット0：'モニタ'表示時にセットします。

RLCD2

ビット3：'送信'表示時にセットします。

RLCD3：群番号の10⁰位

}

RLCD7：群番号の10⁴位

※参考資料(3)
 なお、μPD6320にはデコーダが4桁しかないため、RLCD3を本サブルーチンでデコードして、RLCD2の0~2ビットおよびRLCD3に格納します。

RLCD8：表示には関係ないデータ

なお、 μ PD7508から μ PD6302への送信は、RLCD1からRLCD8まで32ビット行
 ないます。この送信データと表示の関係は下表の通りです。

MSB 1				LSB 32							
8		9		A	B	C	D	E	F	μ PD7508の データ・メモリの COLUMN ADDRESS	
0	待 受	通 話	モ ニ タ 信	送 信	群番号 10 ⁰ 位 (デコードされたデータ)	群番号 10 ¹ 位	群番号 10 ² 位	群番号 10 ³ 位	群番号 10 ⁴ 位	未使用	表示内容
	フ ラ グ				第1桁 7SEG デコーダ	第2桁 7SEG デコーダ	第3桁 7SEG デコーダ	第4桁 7SEG デコーダ	LED ドライバ		μ PD6320の 内部レジスタ名

(3) 入力条件

- RLCD3～RLCD7に群番号が格納されていること。
- RLCD1, RLCD2に表示する要素がセットされていること。

(4) 出力条件

なし。

(5) 変化するレジスタ

A, H, L

(6) 処理説明

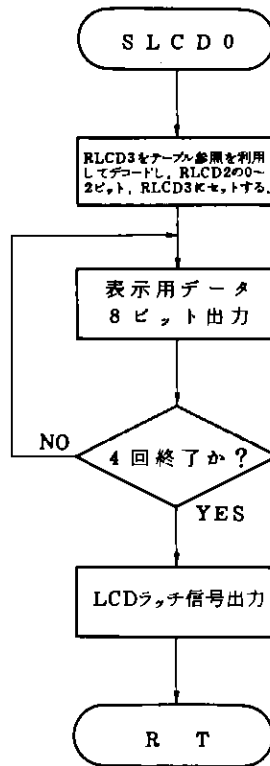
- a) RLCD3のデータをテーブル参照命令を利用してデコードし、RLCD2の0～2ビットおよびRLCD3にセットします。
- b) シリアル・インタフェースを介して、RLCD1～RLCD8のデータ32ビットを μ PD6320に送信します。
- c) LCDラッチ信号を出力します。

(7) フローチャート

PROGRAM
LABEL

SLCD0:

SLCD1:



(8) プログラム

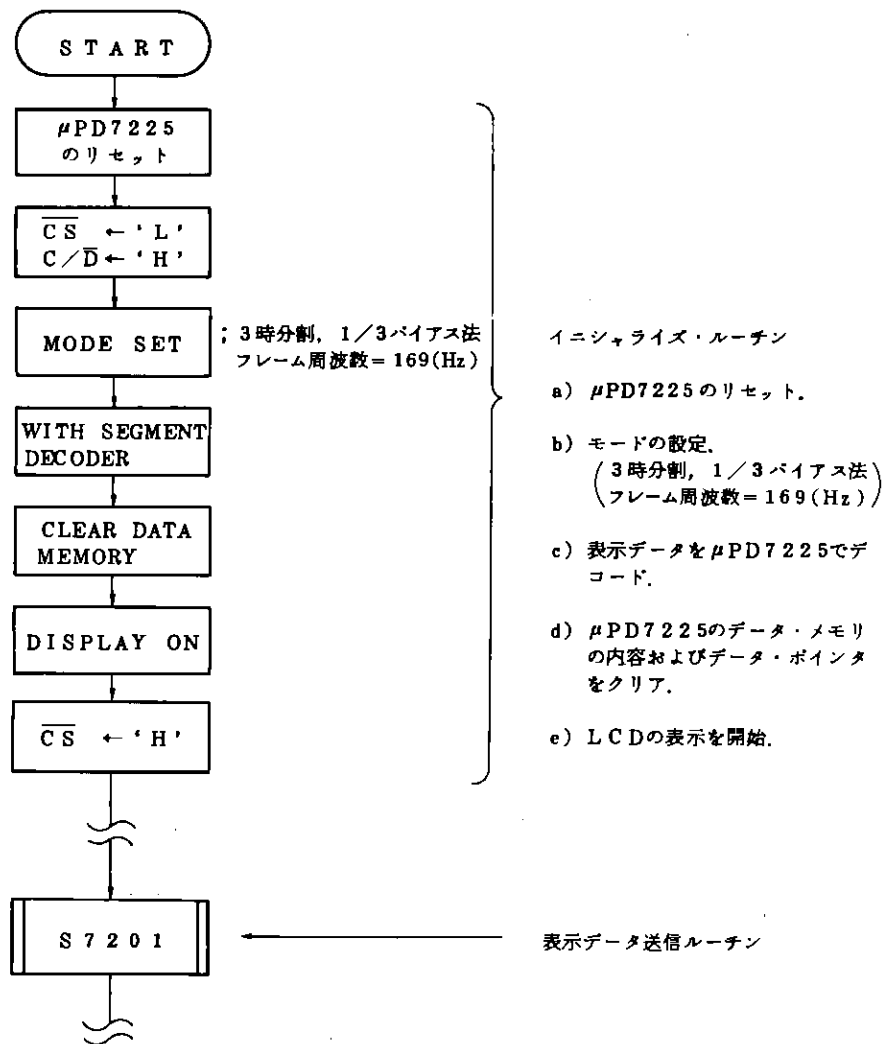
バイト数 45

```
*****
;*
;*      SLCD0 - LCD DRIVER UPD6320 INTERFACE SUBROUTINE      *
;*      SCK ... OUTPUT LCD CLOCK                               *
;*      SO  ... OUTPUT LCD DATA                               *
;*      P20 ... OUTPUT LCD LATCH                               *
;*
;*      INPUTS   : RLCD3-7 ... GROUP NO.                      *
;*                : RLCD1,2 ... STATE FLAG                    *
;*      OUTPUTS  : NOTHING                                     *
;*      DESTROYS : A,H,L                                       *
;*
*****
;
;      ORG      100H
;
;      DB      3FH          ; 0
;      DB      06H          ; 1
;      DB      5BH          ; 2
;      DB      4FH          ; 3
;      DB      66H          ; 4
;      DB      6DH          ; 5
;      DB      7DH          ; 6
;      DB      27H          ; 7
;      DB      7FH          ; 8
;      DB      6FH          ; 9
;      DB      77H          ; A
;      DB      7CH          ; B
;      DB      39H          ; C
;      DB      5EH          ; D
;      DB      79H          ; E
;      DB      00H          ; BLANK
;
; SLCD0:
;      LHLD    RLCD3      ;
;      LAI     0           ;
;      LAMTL   ; RLCD3 <-- DECODE RLCD3
;      DLS    ; HL <-- RLCD2
;      XAM    HL          ; RLCD2 <-- DECODE RLCD3
;      CMA    ;
;      SKABT  3           ;
;      SMB    3           ; RLCD2 <-- STATE FLAG
;
;
; SLCD1:
;      DLS    ; HL <-- RLCD1
;      LAM    HL+         ;
;      TAMSIO ;
;      SIO    ; OUTPUT LCD DATA & CLOCK
;      NOP    ;
;      NOP    ; WAIT 6 MACHINE CYCLE
;      NOP    ;
;      ILS    ;
;      GJMP   SLCD1      ;
;      NOP    ;
;      NOP    ; WAIT 8 MACHINE CYCLE
;      NOP    ;
;      ORP    2,0001B     ; OUTPUT LCD LATCH
;      ANP    2,1110B     ;
;      RT     ;
```


3.2 LCDドライバμPD7225 インタフェース・サブルーチン

ここでは、図3.3に示すμPD7225のコントロール手順の中で、イニシャライズ・ルーチンはすでに設定されているものとし、表示データ送信ルーチンのみ説明します。

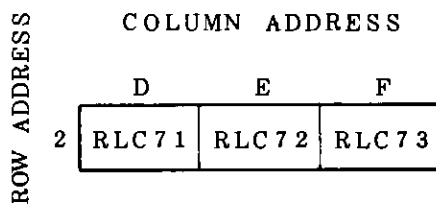
図3.3 μPD7225のコントロール手順



(1) 処理内容

群番号, 待受, 通話, モニタ, 送信, メモリを表示するためのデータをμPD7508のシリアル・インタフェースを介して, μPD7225に出力します。図3.4にμPD7508とμPD7225とのインタフェース構成を示します。

(2) データ・メモリ構成



RLC71：コマンド送信時に下位4ビットを格納します。

RLC72

ビット1：'通話'表示時にセットします。

ビット0：'待受'表示時にセットします。

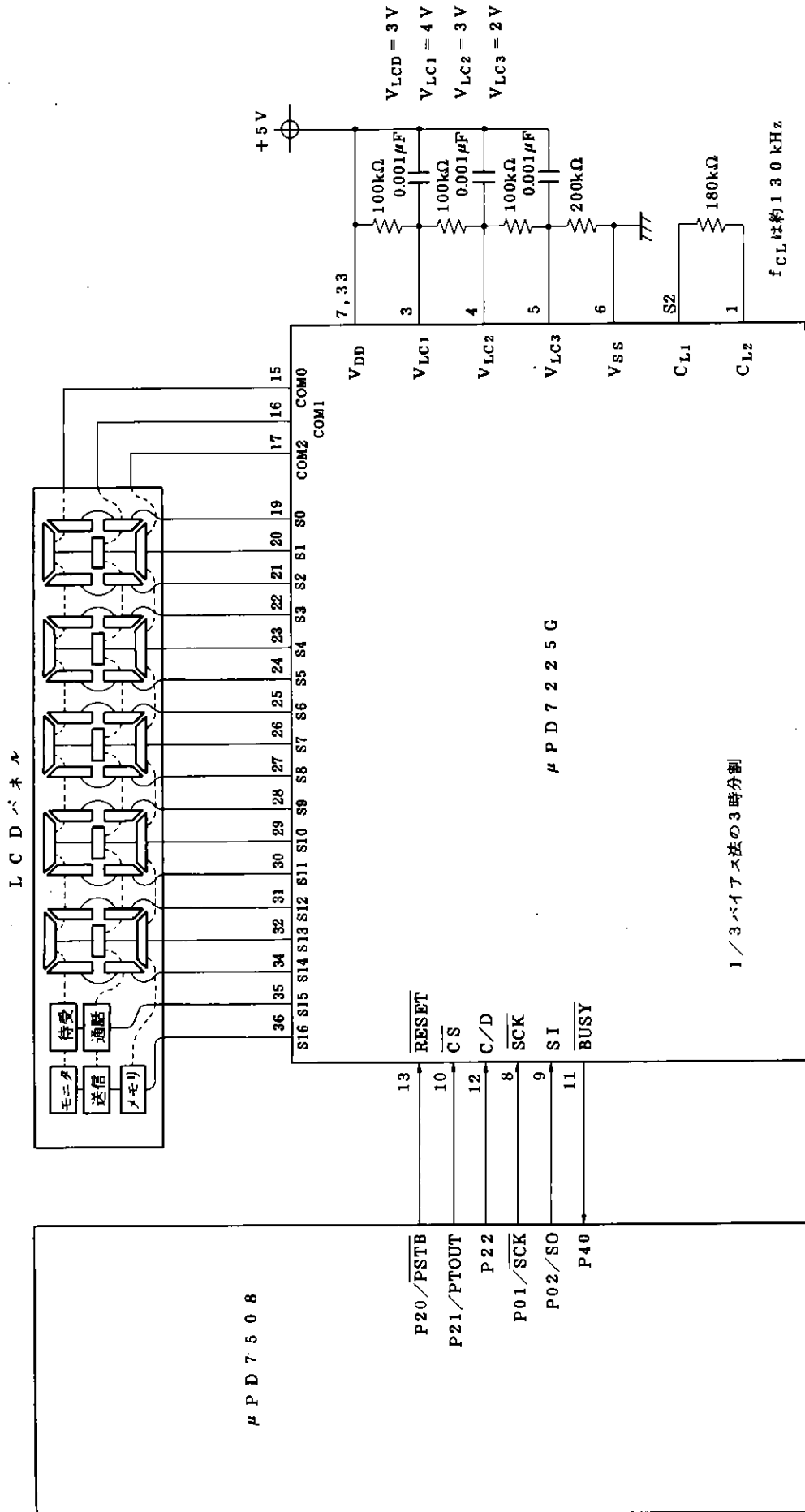
RLC73

ビット2：'メモリ'表示時にセットします。

ビット1：'送信'表示時にセットします。

ビット0：'モニタ'表示時にセットします。

図 3.4 LCDドライバ μ PD7225G インタフェース構成図



(3) 入力条件

- HLレジスタに表示する群番号の先頭アドレスがセットされていること。
- RLC72, RLC73に表示する要素がセットされていること。

(4) 出力条件

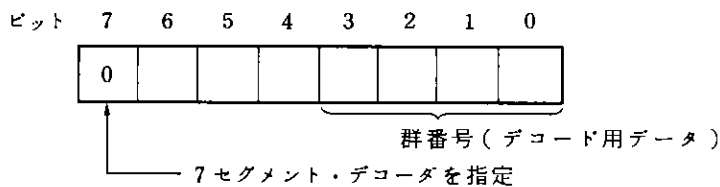
なし。

(5) 変化するレジスタ

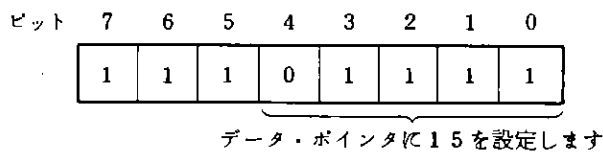
A・H・L・E

(6) 処理説明

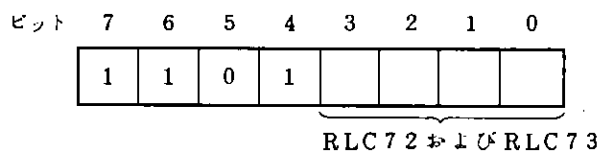
- μPD7225 の $\overline{\text{CS}}$ を'L'にして、シリアル・データの入力を可能にし、 $\text{C}/\overline{\text{D}}$ を'L'にしてシリアル・データ入力をデータ指定に設定します。
- $\overline{\text{BUSY}}$ のチェックをし、シリアル・データの入力が許可になるのを待ちます。
- 下記フォーマットで、シリアル・インタフェースを介して、群番号を1桁送信します。



- b), c)を5回繰り返して、群番号を5桁送信します。
- $\overline{\text{BUSY}}$ のチェックをし、シリアル・データの入力が許可になるのを待ちます。
- $\text{C}/\overline{\text{D}}$ を'H'にしてシリアル・データ入力をコマンド指定に設定します。
- 下記フォーマットでシリアル・インタフェースを介して、LOAD DATA POINTERのコマンドを送信し、その後 $\overline{\text{BUSY}}$ のチェックをしシリアル・データの入力が可能になるのを待ちます。



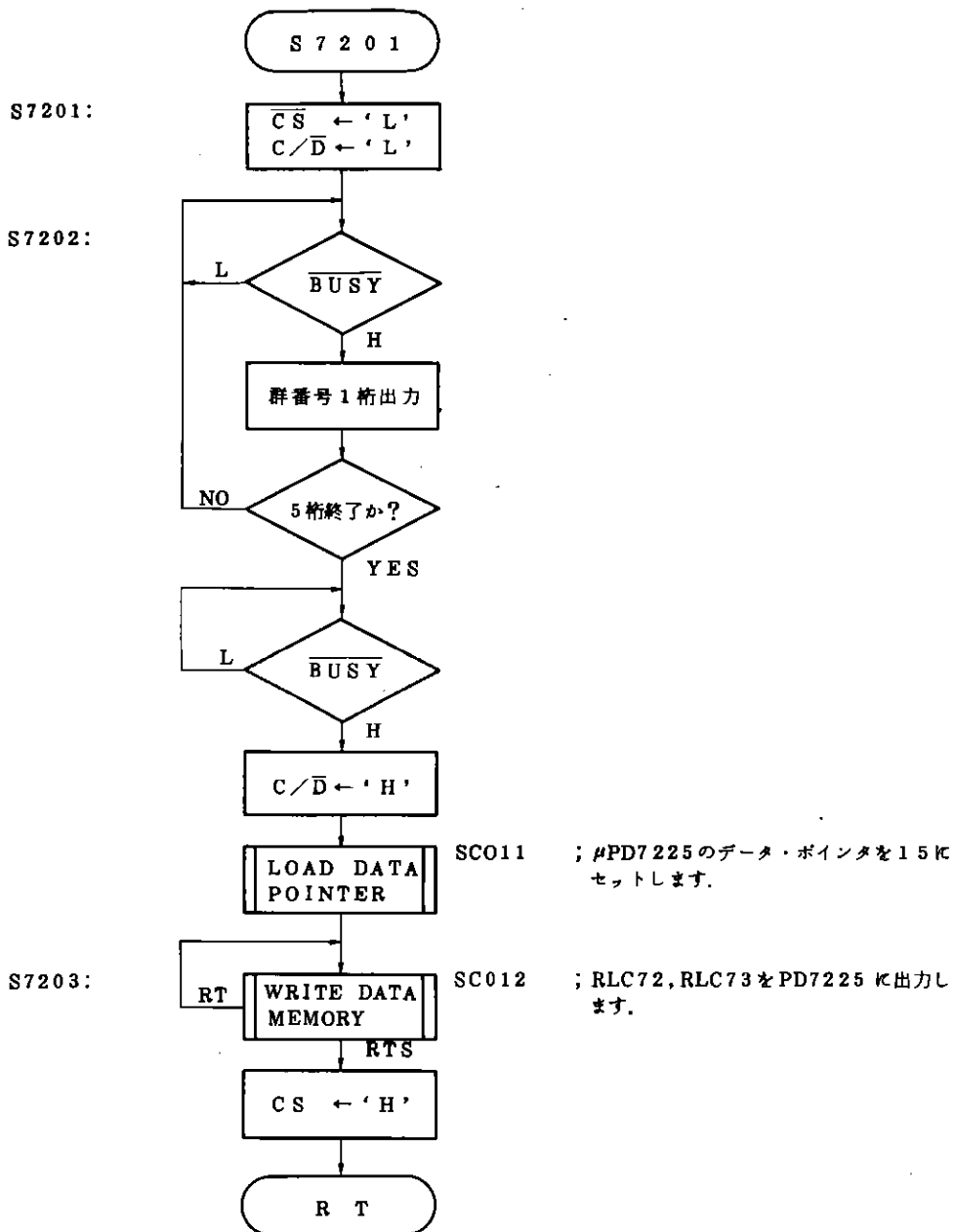
- 下記フォーマットで、シリアル・インタフェースを介して、WRITE DATA MEMORYのコマンドを送信し、その後 $\overline{\text{BUSY}}$ のチェックをしシリアル・データの入力が可能になるのを待ちます。



- ① データ・ポインタでアドレスされるデータ・メモリにRLC72をストアします。ただしデータ・ポインタは自動的にインクリメントされます。
 - ② データ・ポインタでアドレスされるデータ・メモリにRLC73をストアします。ただしデータ・ポインタは自動的にインクリメントされます。
- i) \overline{CS} を'H'にして表示データの送信を終了します。

(3) フローチャート

PROGRAM
LABEL

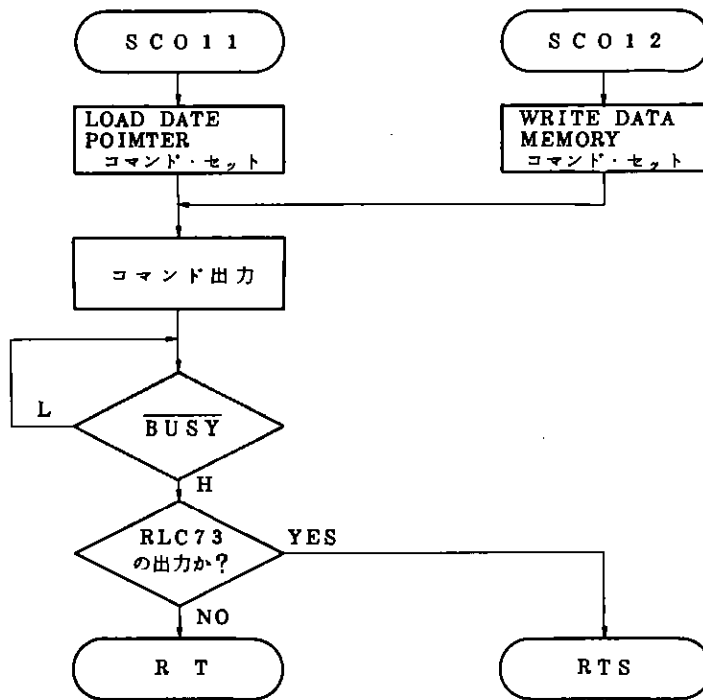


PROGRAM LABEL

PROGRAM LABEL

SCO11:

SCO12:



(8) プログラム
 バイト数 51

```

*****
;*
;*      S7201 - LCD DRIVER UPD7225 INTERFACE SUBROUTINE      *
;*      P20 ... OUTPUT LCD RESET                               *
;*      P21 ... OUTPUT CHIP SELECT (CS)                       *
;*      P22 ... OUTPUT COMMAND / DATA (C/D)                 *
;*      SCK ... OUTPUT LCD CLOCK                              *
;*      SO ... OUTPUT LCD DATA                               *
;*      P40 ... INPUT LCD BUSY                                *
;*
;*      INPUTS   : HL <-- GROUP NO. TOP ADDRESS              *
;*                : RLC72,73 <-- STATE FLAG                  *
;*      OUTPUTS  : NOTHING                                     *
;*      DESTROYS : A,H,L,E                                    *
;*
*****
:
S7201:
      LEI      4          ; SET GROUP NO. COUNTER
      ANP     2,1001B    ; CS <-- 'L' , C/D <-- C
S7202:
      CALL    S7202      ; WAIT UNTIL NOT BUSY
      LAI     0          ;
      CALL    SSERI      ; OUTPUT GROUP NO.
      ILS
      NOP
      DES
      GJMP   S7202      ; SKIP IF END OUTPUT GROUP NO.
:
      CALL    S7202      ; WAIT UNTIL NOT BUSY
      ORP    2,0100B    ; C/D <-- D
      LHLD   RLC71
      LAI    1111B
      ST
      CALL   SCO11      ; LOAD DATA POINTER
:
S7203:
      CALL    SCO12      ; WRITE DATA MEMORY
      GJMP   S7203      ;
:
      ORP    2,0010B    ; CS <-- 'H'
      RT
  
```

```

*****
*
*   SBUSY - JUDGE BUSY UPD7225 SUBROUTINE
*   P40 ... INPUT LCD BUSY
*
*   INPUTS   : NOTHING
*   OUTPUTS  : RT ... NOT BUSY UPD7225
*   DESTROYS : A
*
*****

```

```

SBUSY:
    IP      4           : INPUT LCD BUSY
    SKABT   0           : SKIP IF NOT BUSY UPD7225
    GJMP    SBUSY      : BUSY UPD7225
    RT      : NOT BUSY UPD7225

```

```

*****
*
*   SSERI - OUTPUT SERIAL INTERFACE SUBROUTINE
*   SCK ... OUTPUT LCD CLOCK
*   SO  ... OUTPUT LCD DATA
*
*   INPUTS   : A ... S07-4
*             : (HL) ... S03-0
*   OUTPUTS  : NOTHING
*   DESTROYS : NOTHING
*
*****

```

```

SSERI:
    TAMSIO           :
    SIO              : OUTPUT LCD DATA
    NOP              :
    NOP              :
    NOP              :
    RT               :

```

```

*****
*
*   SCO12 - OUTPUT WRITE DATA MEMORY COMMAND SUBROUTINE
*
*   INPUTS   : (HL) ... WRITE DATA
*   OUTPUTS  : RT ... L <-- L+1
*             : RTS ... L <-- L+1 , L=0
*   DESTROYS : A,H,L
*
*****

```

```

SCO12: LAI    1101B      ;

```

```

*****
*
*   SCO11 - OUTPUT LOAD DATA POINTER COMMAND SUBROUTINE
*
*   INPUTS   : (HL) ... DATA POINTER
*   OUTPUTS  : RT ... L <-- L+1
*             : RTS ... L <-- L+1 , L=0
*   DESTROYS : A,H,L
*
*****

```

```

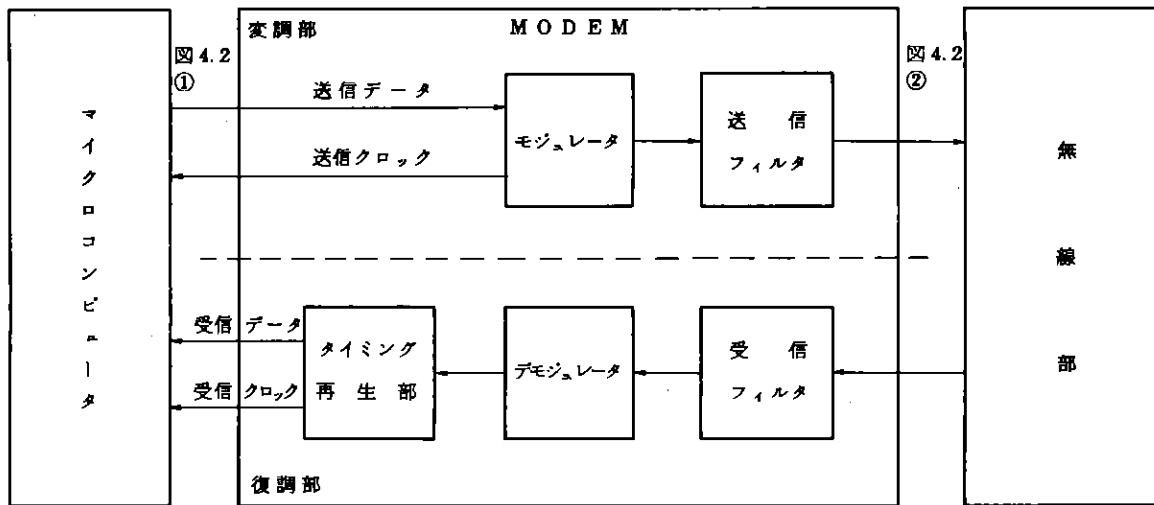
SCO11: LAI    1110B      ;
        CALL  SSERI      : OUTPUT COMMAND DATA
        CALL  SBUSY      : WAIT UNTIL NOT BUSY
        ILS
        RT
        RTS

```


第4章 MODEM インタフェース

MODEMは、マイクロコンピュータからの送信データを変調信号に変換する変調部と無線部からの変調信号を受信データに復調する復調部とから構成されている装置です。図4.1にMODEMのシステム構成を示します。

図4.1 MODEMシステム構成図



パーソナル無線では、制御信号の送受信をMODEMを介して行なっています。

送信時は、80ビットのデータ信号をマイクロコンピュータでハーゲルバーガ符号化して237ビットの制御信号に変えMODEMに送信します。この符号化は、受信時に誤り訂正をするために行なっています。MODEMでは、モジュレータ部で図4.2に示すように入力された制御信号をMSK変調します。

受信時は、無線部からの変調信号をデモジュレータでNRZ信号に復調し、タイミング再生部において図4.3に示すタイミングで受信クロックが発生します。

マイクロコンピュータでは、受信データをハーゲルバーガ復号化して、誤り訂正を行ないます。

注1 ハーゲルバーガ符号

制御信号の誤り訂正をするための符号。

注2 MSK (Minimum Shift Keying) 変調

デジタル信号の1と0に対応させて周波数を変化させるFSK変調の中で、変化幅を伝送速度の1/2に定めたもの。

注3 NRZ (Non Return to Zero) 信号

デジタル信号で、1ビットの情報に一定の時間長さを対応させる信号。

図 4.2 制御信号の変調波形

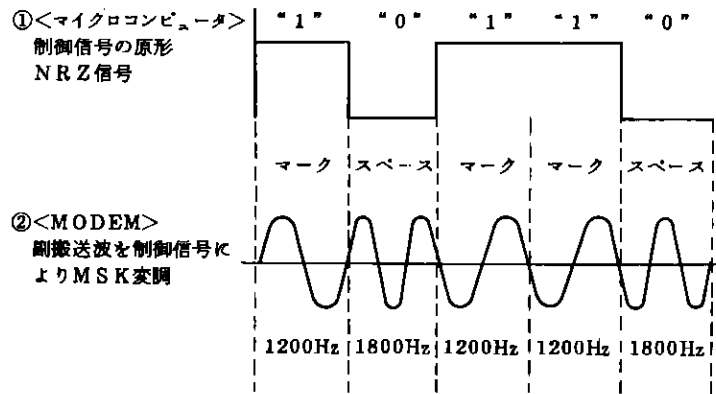
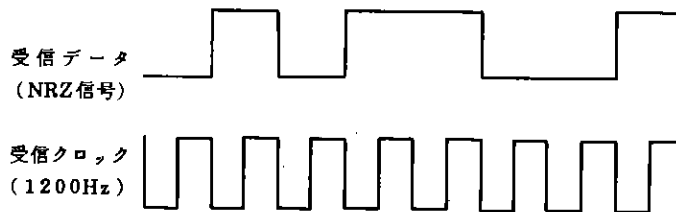


図 4.3 再生クロック・タイミングチャート



制御信号の送受信処理はMODEMからの送信クロックおよび受信クロックによりINT1、INT0で発生する割り込みルーチンで行なっています。この時、送信および受信クロックは共に1200Hzで送られてきます。

ですからINT1割り込みルーチンおよびINT0割り込みルーチンの処理が833 μ s(1/1200秒)以内に終了しないこと、次のデータの送受信ができなくなります。

※参考資料[5]

なお、本書ではMODEMに μ PD6302CAを使用した場合について説明します。

4.1 送信処理

(1) 処理内容

制御信号を μ PD6302CAに送信します。図4.4に μ PD7508と μ PD6302CAとのインタフェース構成を示します。

また、図4.5に制御信号の構成を示します。

図 4.4 MODEM μ PD6302CA インタフェース構成図

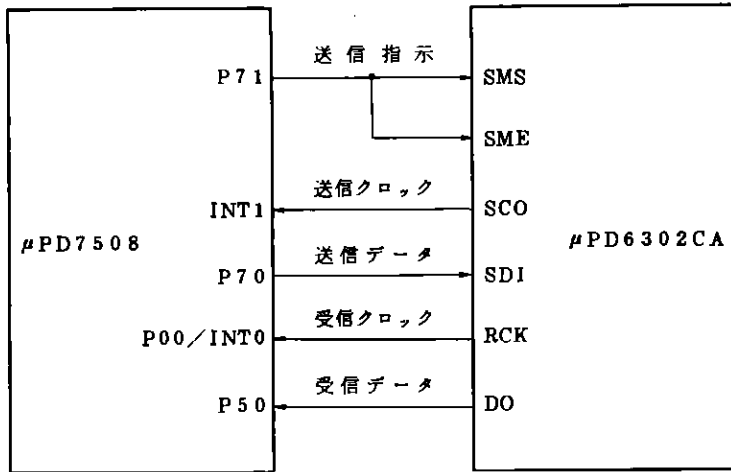
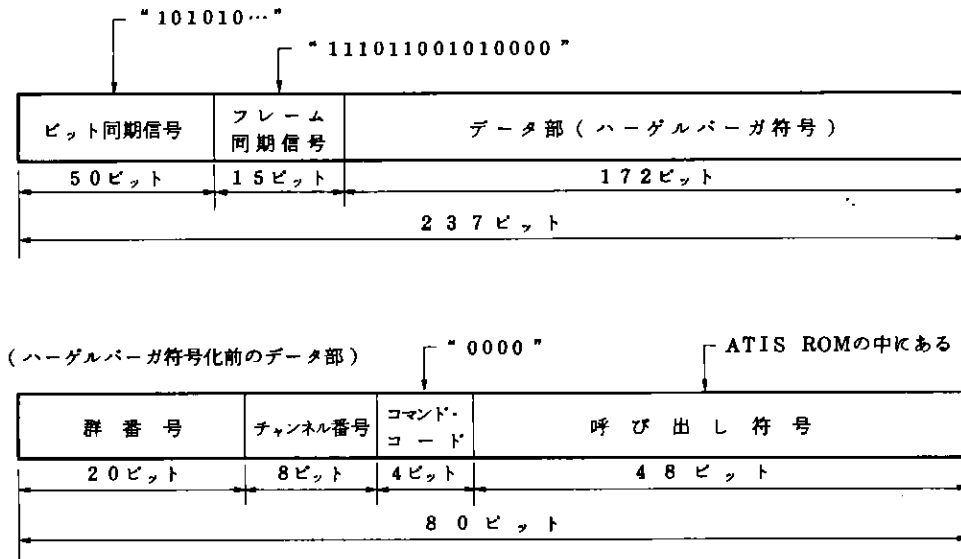


図 4.4 の P 7 1 の送信指示信号は、制御信号の送信中 'H' を出力します。なお、制御信号の送信は送信クロックの立上りにより発生する INT1 割込みルーチンで行ないます。

図 4.5 制御信号の構成



制御信号は、図 4.5 に示すようにビット同期信号、フレーム同期信号およびデータ部から構成されています。データ部には、群番号、チャンネル番号、コマンド・コード、呼び出し符号から成る 80 ビットを、受信時誤り訂正を行なうためハーゲルバーガ符号化して 172 ビットにしたデータが入っています。このハーゲルバーガ符号化の生成式は (式 4.1)、(式 4.2) であらわされます。

$$Y_{2i-1} \text{ (チェック・ビット)} = \begin{cases} \overline{X_i} & i = 1 \sim 3 \\ \overline{X_i \oplus X_{i-3}} & i = 4 \sim 80 \\ \overline{X_{i-3}} & i = 81 \sim 83 \\ 1 & i = 84 \sim 86 \end{cases} \quad (\text{式 4.1})$$

$$Y_{2i} \text{ (データ・ビット)} = \begin{cases} 0 & i = 1 \sim 6 \\ X_{i-6} & i = 7 \sim 86 \end{cases} \quad (\text{式 4.2})$$

ただし、 X_i は第 i 番目のデータで、 Y_{2i-1} 、 Y_{2i} は、それぞれ第 $2i-1$ 番目、第 $2i$ 番目のハーゲルバーガ符号です。

このようにハーゲルバーガ符号化では、誤り訂正を行なうために 1 ビットのデータに対して 1 ビットのチェック・ビットを作成しています。

なお、データ (80 ビット) の前後にそれぞれ 6 ビットのダミー・ビット 0 を付加したものを X_i ($i = 1 \sim 92$) とし、ハーゲルバーガ符号を Y_i ($i = 1 \sim 172$) とすると (式 4.1) (式 4.2) は、それぞれ (式 4.3)、(式 4.4) であらわされます。

$$Y_{2i-1} \text{ (チェック・ビット)} = \overline{X_{i+6} \oplus X_{i+3}} \quad i = 1 \sim 86 \dots (\text{式 4.3})$$

$$Y_{2i} \text{ (データ・ビット)} = X_i \quad i = 1 \sim 86 \dots (\text{式 4.4})$$

図 4.6 にハーゲルバーガ符号化の論理図を示します。

図 4.6 ハーゲルバーガ符号化論理図

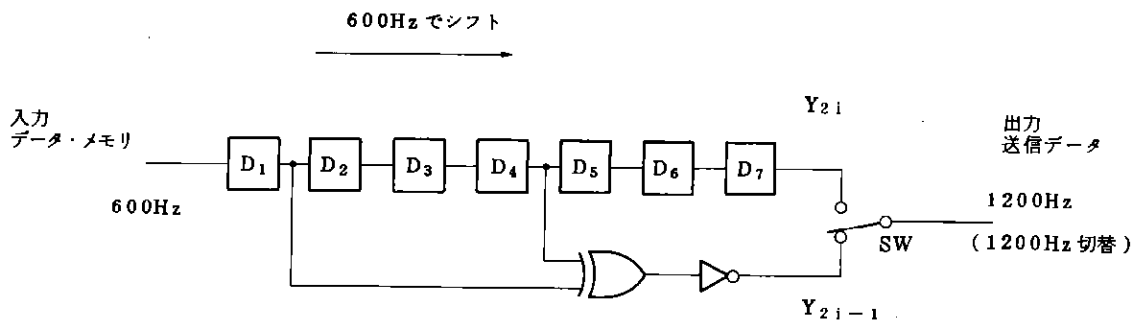


図 4.6 では、 D_7 が現在出力するデータ・ビットです。 D_7 に対するチェック・ビットを生成するには、後の 6 ビットが必要となります。

(2) データ・メモリ構成

		COLUMN ADDRESS															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
6	RINT1 Acc 送信 エリア	ROUT1 1010	(フレーム同期データ)						(送信群番号)			(チャンネル番号)		ROUTF (コメント)			
		0101	1101	1001	0100	0000	0000	送信データ			エリア						
7	RROM0	(ATIS ROM データ エリア)										RROMA	0000	0000	RBITL 送信 ビット 同期 カウンタ		
8																	
9						RIDCH 送信カウンタ	RIOCL	RBITC ビット カウンタ	RADRH 同期カウンタ	RADRL	RTRNS 送信 データ						
A	RFLG2 送信 フラグ レジスタ																

アドレス	変数名	ビット	フラグ名	説明
0A1H	RFLG2 (送信フラグ・ レジスタ)	3	FTRNS	送信フラグ。送信中にセットします。
		2	FTEND	送信データ終了フラグ。データが終了したらセットします。
		1	FCDGT	符号化要求フラグ。ハーゲルバーガ符号化する時にセットします。
		0	FENCD	データ部送信要求フラグ。データ部を送信する時にセットします。

(3) 入力条件

- 送信データ・エリアにデータが格納されていること。
- INT0の割込み(受信処理)が禁止されていること。

(4) 出力条件

制御信号出力が237ビットすべて終了すること

(5) 変化するレジスタ

A・H・L

(6) 処理説明

送信処理サブルーチン(サブルーチン名'STRNS')では送信処理に必要な各種フラグ、カウンタを設定し、INT1割込みを許可にして制御信号の送信を開始します。そして、制御信号の送信が237ビットすべて終了したら、INT1割込みを禁止にしてRTでメインに戻ります。

INT1割込みは、送信クロックの立上りで発生し、1ビットずつハーゲルバーガ符号化し

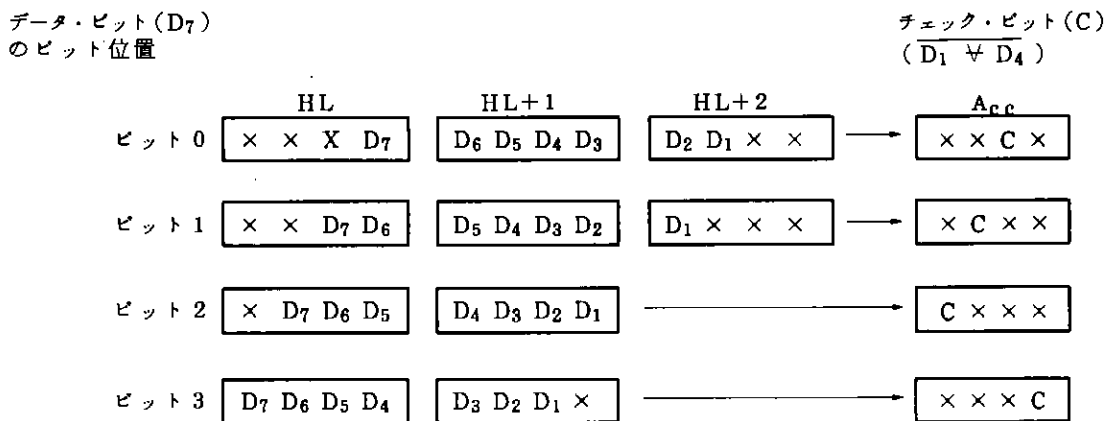
ながら制御信号を送信します。

以下にINT1割込みで行なっている制御信号の送信方法を説明します。

- a) ビット同期信号(全50ビット)を送信するため、データ・メモリ61Hのデータ(1010B)を送信ビット同期カウンタのカウンタ数(12)回繰り返し出力し、48ビットのビット同期信号を送信します。ただし、ビット同期信号の第1ビットめはINT1割込み発生前STRNSサブルーチン内で設定しますので、データ・メモリ61Hのデータ送信は第2ビット目から行ないます。
- b) ビット同期信号の残り2ビットとフレーム同期信号を出力するため、データ・メモリの62H^{*}~66H⁽²⁾の17ビットを送信します。
- c) データ部を送信するためデータ・メモリの66H⁽¹⁾~7DH⁽²⁾の92ビットをハーグルバーガ符号化し、172ビットにして出力します。

このハーグルバーガ符号化では符号化要求フラグを判定し、データ・ビットとチェックビットとを交互に生成します。チェック・ビットの生成($D_1 \oplus D_4$)は、D₇のビット位置(RBITC)により図4.7に示すように変化します。

図4.7 チェック・ビットの生成



X : チェック・ビットの生成には関係ないデータ

C : チェック・ビット (D₄のビット位置と同じビット位置にセットされます)

* データ・メモリ中の()内の数字はビット数を示します。

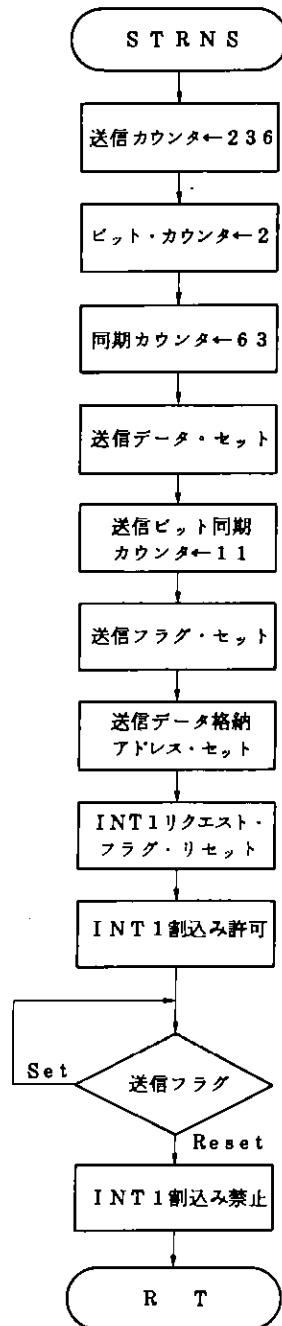
(7) フローチャート

F1/F5

PROGRAM LABEL

STRNS:

STRN1:



PROGRAM LABEL

PROGRAM LABEL

I1000:

I N T 1

Acc 退避

送信データ出力

送信データ格納
アドレス 退避

送信データ終了フラグが
セットか?

YES

送信カウンタ
デクリメント

カウント・アウトか?

YES

符号化要求
フラグがセット
か?

NO

I1001:

送信データ終了
フラグ・セット
送信データ・セット

符号化処理

データ処理

:ビット同期.
フレーム同期.
データ・ビット
処理.

I1030:

送信フラグ・
レジスタ・リセット

I N T 1 割込み禁止

I1020:

I1031:

送信データ格納
アドレス 復帰

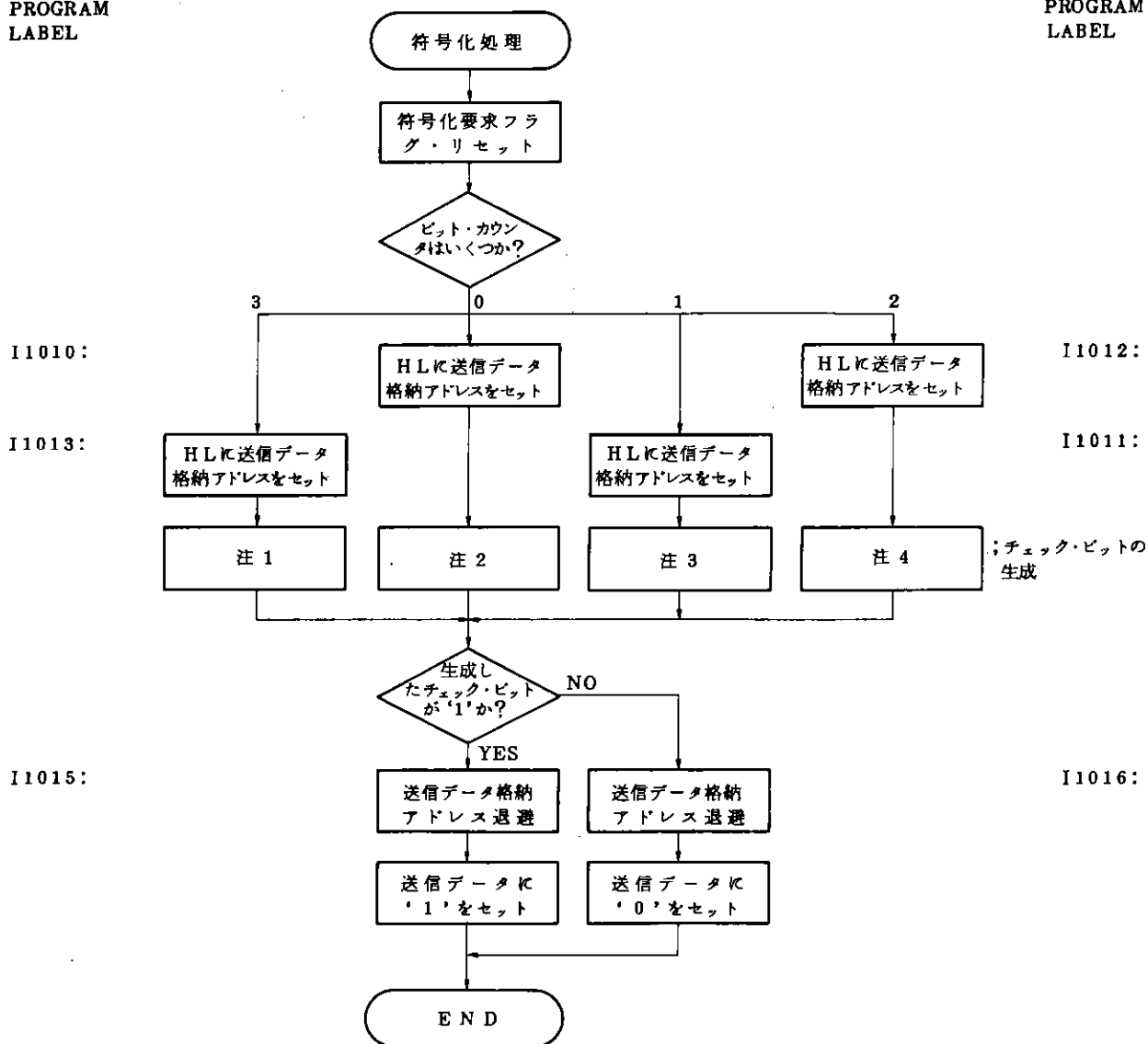
Acc 復帰

割込みマスタ・
イネーブル許可

R T P S W

PROGRAM LABEL

PROGRAM LABEL



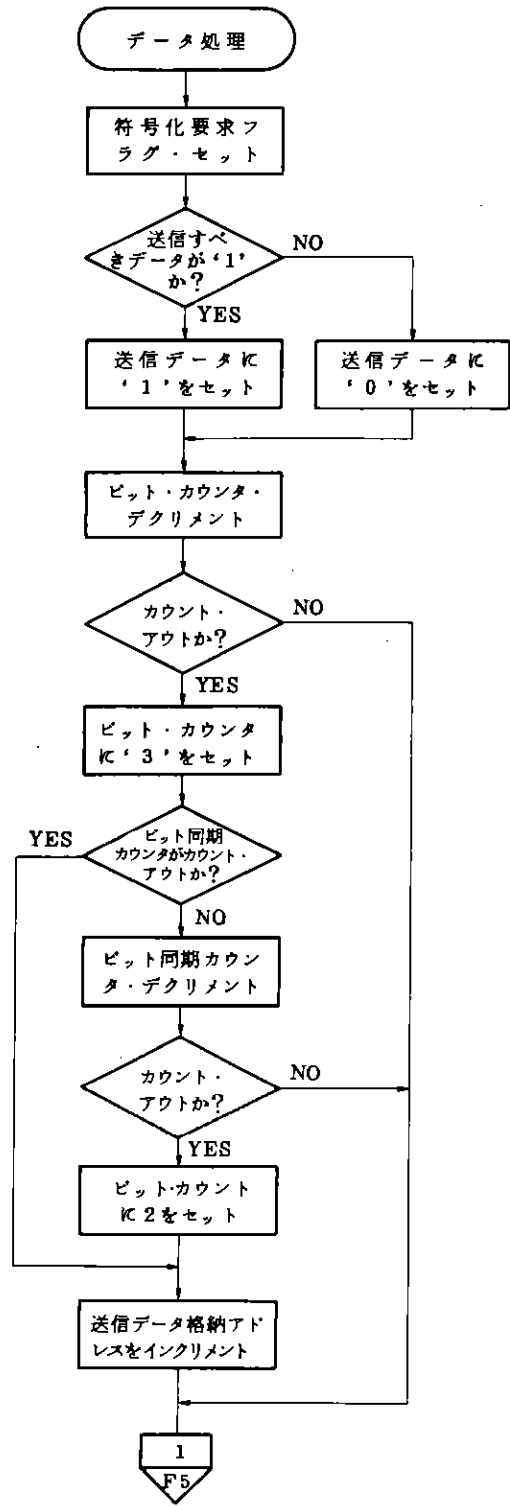
- 注1 $Acc(0) \leftarrow (HL+1)(1) \nabla (HL)(0)$
- 注2 $Acc(1) \leftarrow (HL+2)(1) \nabla (HL+1)(1)$
- 注3 $Acc(2) \leftarrow (HL+2)(2) \nabla (HL+1)(2)$
- 注4 $Acc(3) \leftarrow (HL+1)(3) \nabla (HL+1)(0)$

PROGRAM LABEL

I1020:

I1021:

I1022:



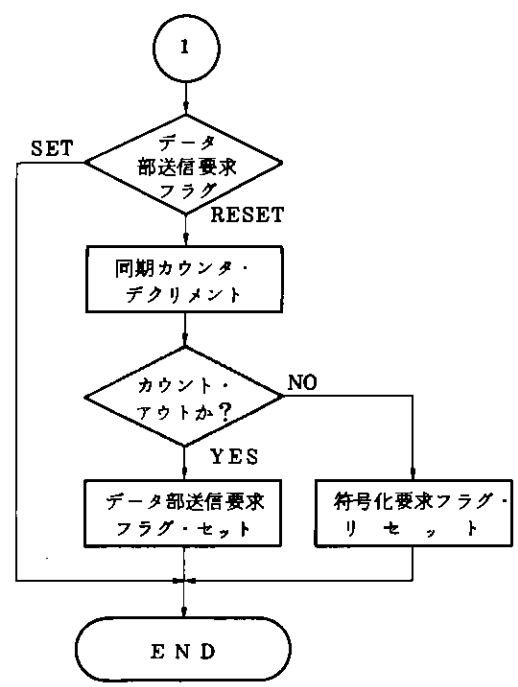
;送信データ格納アドレスで示されるデータ・メモリのビット・カウンタで示されるビットが'1'か?

PROGRAM LABEL

PROGRAM LABEL

I1024:

I1025:



(8) プログラム
 バイト数 207

```

:*****
;*
;*      STRNS - TRANSMIT SUBROUTINE
;*
;*      INPUT   : ROUT1- ... TRANSMIT DATA
;*      OUTPUTS : NOTHING
;*      DESTROYS : A,H,L
;*
:*****
;
STRNS:
    LHLI    RIOCH      ;
    LAI     0EH        ;
    XAM     HL+        ;
    LAI     0CH        ;
    XAM     HL+        ; SET TRANSMIT COUNTER
    LAI     2          ;
    XAM     HL+        ; SET BIT COUNTER
    LAI     3          ;
    XAM     HL+        ;
    LAI     0FH        ;
    XAM     HL+        ; SET SYNC COUNTER
    LAI     0111B     ;
    ST      ; SET TRANSMIT DATA
    LAI     0BH        ;
    XADR    RBITL      ; SET BIT SYNC COUNTER
    LHLI    RFLG2      ;
    SMB     FTRNS      ; SET TRANSMIT FLAG
    LHLI    ROUT1     ; SET TRANSMIT DATA ADDRESS
    SKI     100B       ;
    NOP     ;
    EI      100B       ; ENABLE INT1

STRN1:
    LADR    RFLG2      ;
    CMA     ;
    SKABT   FTRNS      ; SKIP IF RESET FTRNS
    GJMP    STRN1      ;
    DI      100B       ; DISABLE INT1
    RT      ;
  
```

```

:
:***** INT1 ROUTINE *****
:

```

```

    ORG    30H
    GJMP   I1000

```

```

:*****
:
:   SCDGT - EDIT CHECK DIGIT
:
:   INPUTS  : HL ... DATA START ADDRESS
:   OUTPUTS : A ... CHECK BIT
:   DESTROYS : A
:
:*****
:

```

```

SCDGT:
    PSHHL           ; SAVE TRANSMIT DATA ADDRESS
    ILS             ;
    GJMP    SCDGT2 ;
    LHLI    RROM0  ;
    GJMP    SCDGT2 ;
SCDGT1:
    PSHHL           ; SAVE TRANSMIT DATA ADDRESS
SCDGT2:
    ILS             ;
    GJMP    SCDGT3 ;
    LHLI    RROM0  ;
SCDGT3:
    LAM    HL-     ;
    GJMP    SCDGT4 ;
    LHLI    ROUTF  ;
SCDGT4:
    RAR           ;
    EXL           ; EX-OR D1:D4
    CMA           ;
    POPHL         ; LOAD TRANSMIT DATA ADDRESS
    RT            ;

```

```

:
:*****
:
:   INT1 ROUTINE (TRANSMIT)
:
:*****
:

```

```

I1000:
    XADR    RINT1   ; SAVE ACC
    LADR    RTRNS   ;
    OP      7       ; OUTPUT TRANSMIT DATA
    PSHHL           ; SAVE TRANSMIT DATA ADDRESS
    LHLI    RFLG2   ;
    SKMBF   FTEND   ; SKIP IF RESET FLAG TRANSMIT DATA END
    GJMP    I1030   ; END TRANSMIT
    DDERS   RIOCL   ; DECREMENT TRANSMIT COUNTER
    GJMP    I1001   ; NOT COUNT OUT
    DDERS   RIOCH   ;
    GJMP    I1001   ; NOT COUNT OUT
    SMB     FTEND   ; SET FLAG TRANSMIT DATA END
    LAI    0100B   ;
    XADR    RTRNS   ; SET TRANSMIT DATA
    GJMP    I1031   ;
I1001:
    SKMBT   FCDGT   ; SKIP IF SET FLAG CHECK DIGIT
    GJMP    I1020   ; EDIT DATA PROCESS

```


4.2 受信処理

4.2.1 受信ルーチン

(1) 処理内容

制御信号をμPD6302CAからの受信クロックにより発生するINT0割込みで取込みます。制御信号のビット同期信号(50ビット)とフレーム同期信号(15ビット)はチェックのみを行ない、データ部のうち84ビットはデータ・メモリに格納し、残り88ビットは無視します。

なお、ハード構成は図4.4を参照してください。

(2) データ・メモリ構成

		COLUMN ADDRESS																						
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F							
ROW ADDRESS	6	RINT0		(フレーム同期データ)																				
				1	1101	1001	0100	00																
	7																				RTMNG 受信 ビット 同期 カウンタ			
	8	RIN01	RIN02	RIN03	RIN04 (ダミー・ビット)	RIN06	(受信群番号)																	
				受信データ格納エリア																				
9	RIN17	(チャンネル番号)			(コマンド)		(呼び出し符号)	RIN2	RIOCH	RIOCL	RBITC	RADRH	RADRL				RPO70	RFRMC						
								受信カウンタ	ビット・カウンタ	受信データアドレス	格納	受信データ					受信データセーフエリア	フレーム同期カウンタ						
A	RFLG1 受信 フラグ レジスタ					RDECD	デコード・エリア																	
																		(チャンネル番号)			(コマンド)			

アドレス	変数名	ビット	フラグ名	説明
0A0H	RFLG1	3	FRECV	データ受信フラグ。データを受信する時にセットします。
		2	FFRAM	フレーム同期受信フラグ。フレーム同期を受信する時にセットします。
		1	FRCHK	ビット同期、フレーム同期チェック・データ・フラグ。
		0	FDECD	復号化要求フラグ。受信データをハーゲルパーガ復号化する時にセットします。

(3) 入力条件

受信処理を開始する前にRTMNGに9をセットしてからINT0を許可にすること。

(4) 出力条件

なし

(5) 変化するレジスタ

なし

(6) 処理説明

a) ビット同期入力

入力データのチェックのみを行ない、データ・メモリには格納しません。10ビット以上'1'、'0'が交互に続いた時、ビット同期入力中と判断してフレーム同期が入力されるのを待ちます。

b) フレーム同期入力

入力データのチェックのみを行ない、データ・メモリには格納しません。フレーム同期入力待ちの間に入力データが'1'、'1'と続いたらフレーム同期入力と判断し、フレーム同期データ・エリアのデータをもとに残り13ビットのチェックを行ないます。フレーム同期に2ビット以上のエラーがあった場合には、再度ビット同期のチェックからやり直します。

c) データ部入力

フレーム同期に2ビット以上のエラーがなかった時ダミー・ビットを含めて84ビット（呼び出し符号の先頭8ビットまで）を入力し、データ・メモリへ格納します。チェックビットは反転して格納します。

d) その他

データ部入力が84ビット終了すると復号化要求フラグをセットします。このフラグがセットされている時は、INT0割込みが発生しても何も処理を行わず、メインに戻ります。

(7) フローチャート

F 1 / F 4

PROGRAM LABEL

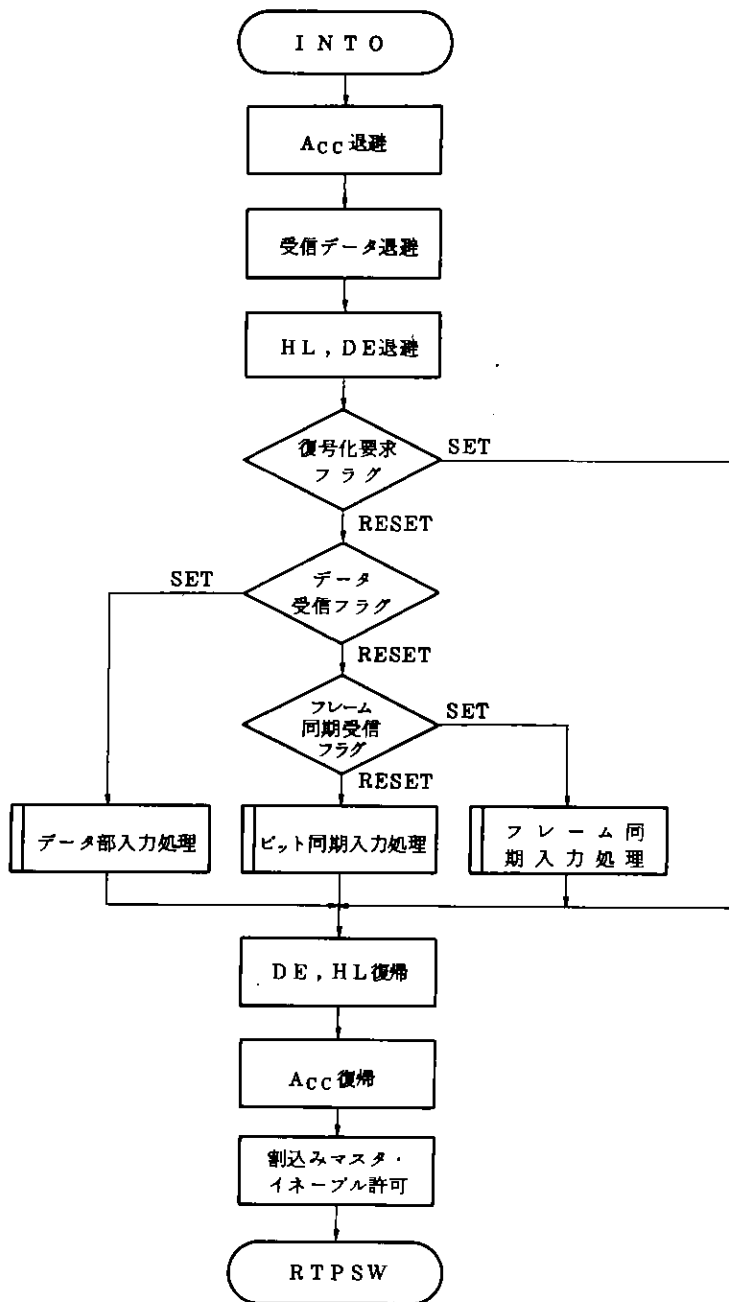
PROGRAM LABEL

10000:

10030:

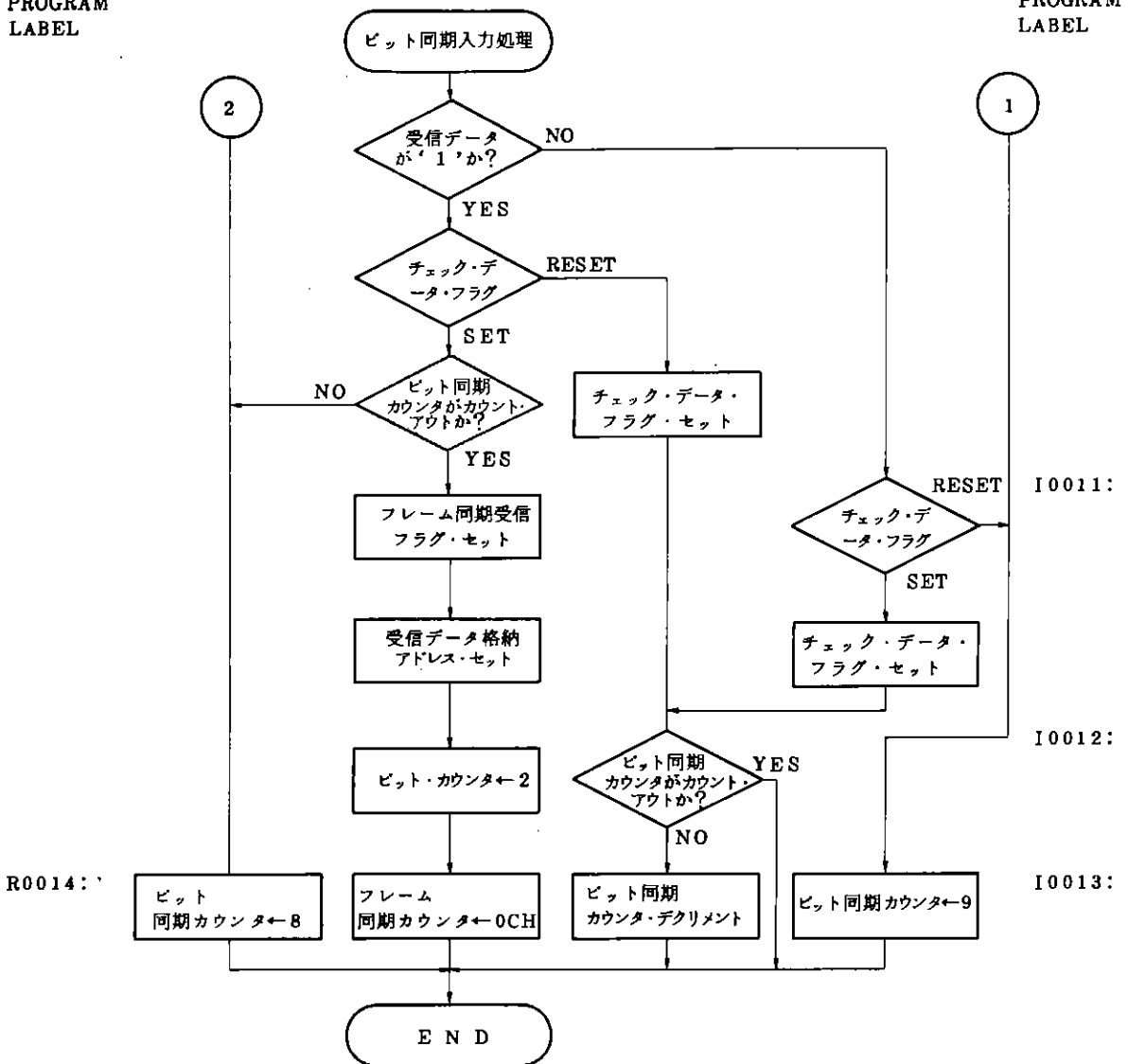
10040:

10Q20:



PROGRAM LABEL

PROGRAM LABEL



PROGRAM LABEL

PROGRAM LABEL

I0021:

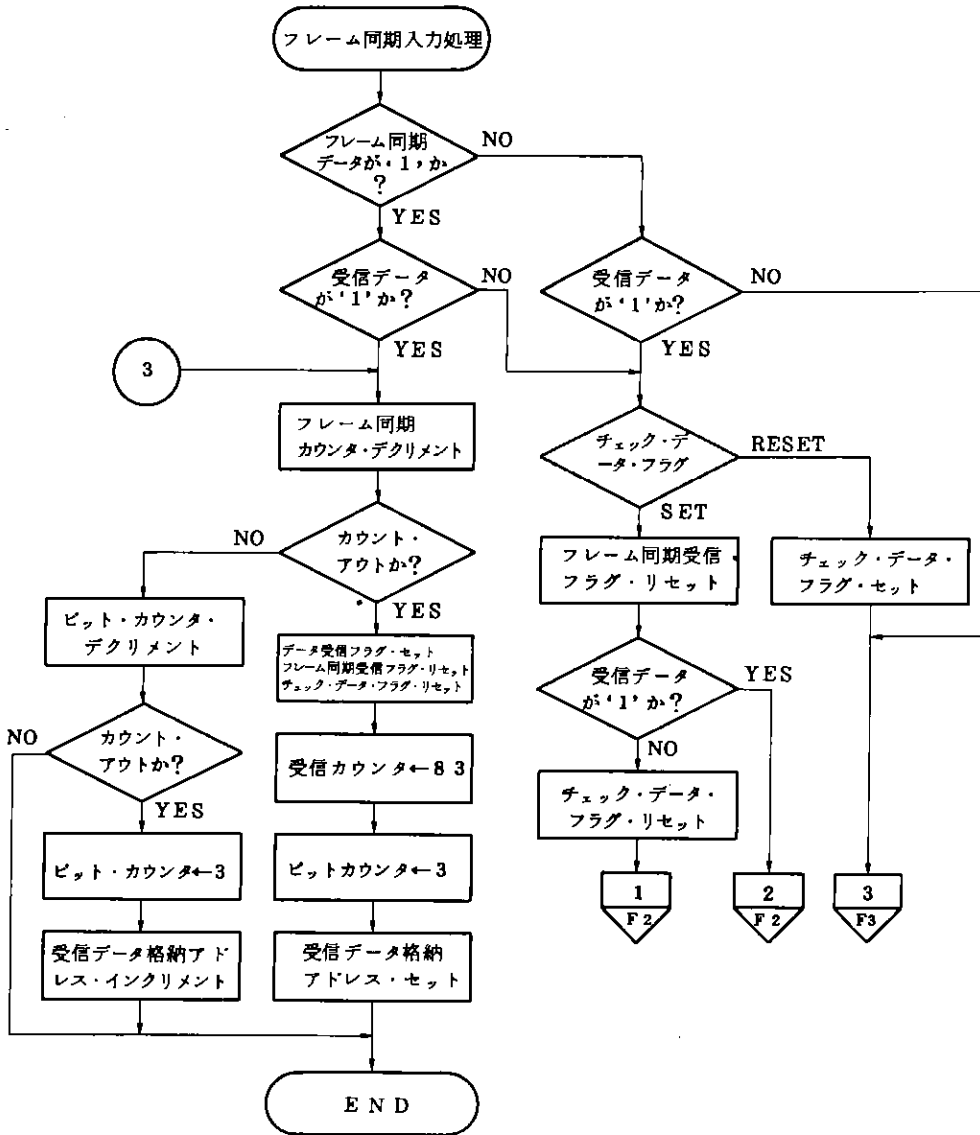
I0022:

I0023:

I0024:

I0025:

I0027:



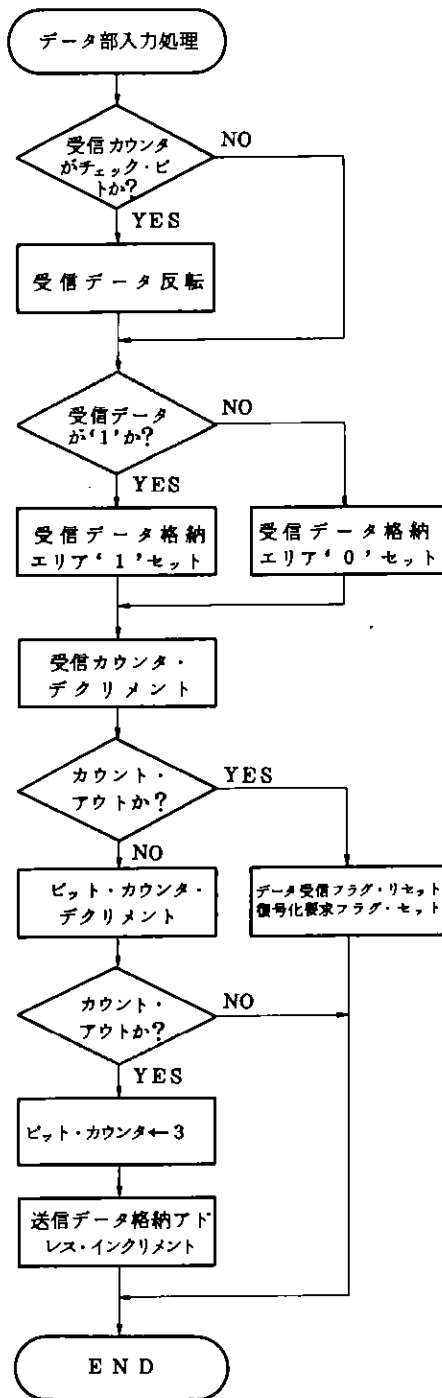
PROGRAM LABEL

PROGRAM LABEL

I0030:

I0031:

I0032:



(8) プログラム

バイト数 2 1 4

```

;
;***** INT0 ROUTINE *****
;
;   ORG      20H
;
;   GJMP    10000
;
;*****
;*
;*   SRMBT - RESET MEMORY BIT
;*
;*           INPUTS   : HL      ... SET DATA RAM ADDRESS
;*                   : RBITC ... SET BIT POSITION
;*           OUTPUTS  : RTS
;*           DESTROYS : A
;*
;*****
;
SRMBT:
    LADR    RBITC      ;
    AISC    0FH        ;
    GJMP    SRMB0      ; BIT 0
    AISC    0FH        ;
    GJMP    SRMB1      ; BIT 1
    AISC    0FH        ;
    GJMP    SRMB2      ; BIT 2
    GJMP    SRMB3      ; BIT 3
SRMB1: LAI    1101B    ;
SRMB2: LAI    1011B    ;
SRMB3: LAI    0111B    ;
SRMB0: LAI    1110B    ;
        ANL          ; RESET BIT
        ST           ;
        RTS          ;

```

```

:
:*****
:
:      SSMBT - SET MEMORY BIT
:
:      INPUTS   : HL   ... SET DATA RAM ADDRESS
:               : RBITC ... SET BIT POSITION
:      OUTPUTS  : RT
:      DESTROYS : A
:*****
:

```

```

SSMBT:
      LADR      RBITC      :
      AISC      0FH        :
      GJMP      SSMB0      : BIT 0
      AISC      0FH        :
      GJMP      SSMB1      : BIT 1
      AISC      0FH        :
      GJMP      SSMB2      : BIT 2
      GJMP      SSMB3      : BIT 3
SSMB1: LAI      0010B      :
SSMB2: LAI      0100B      :
SSMB3: LAI      1000B      :
SSMB0: LAI      0001B      :
      ORL              : SET BIT
      ST
      RT

```

```

:
:*****
:
:      INT0 ROUTINE (RECEIVE)
:*****
:

```

```

10000:
      XADR      RINT0      : SAVE ACC
      IP        5          :
      XADR      RPOT0      : SAVE RECEIVE DATA
      PSHHL     : SAVE HL
      PSHDE     : SAVE DE
      LADR      RPOT0      :
      LHLD     RFLG1      :
      SKMBF     FDECD      : SKIP IF RESET FLAG DECODE REQ.
      GJMP      I0040      : DECODE REQ.
      SKMBF     FRECV      : SKIP IF RESET FLAG DATA RECEIVE
      GJMP      I0030      : RECEIVE DATA
      SKMBF     FFRAM      : SKIP IF RESET FLAG FLAME SYNC RECEIVE
      GJMP      I0020      : FRAME CHECK

```


4.2.2 ハーゲルバーガ復号化サブルーチン(SJD00)

(1) 処理内容

MODEMインタフェースの章で説明したように、制御信号をハーゲルバーガ符号化して送信するため、受信時には入力信号をハーゲルバーガ符号化する必要があります。本書では、データ・メモリの受信データ格納エリアに格納した制御信号の内、ダミー・ビットからコマンドまでの18ワードをハーゲルバーガ復号化してデコード・エリアに格納します。

図4.7にこのハーゲルバーガ復号化の論理図を示します。

図4.7 ハーゲルバーガ復号化論理図

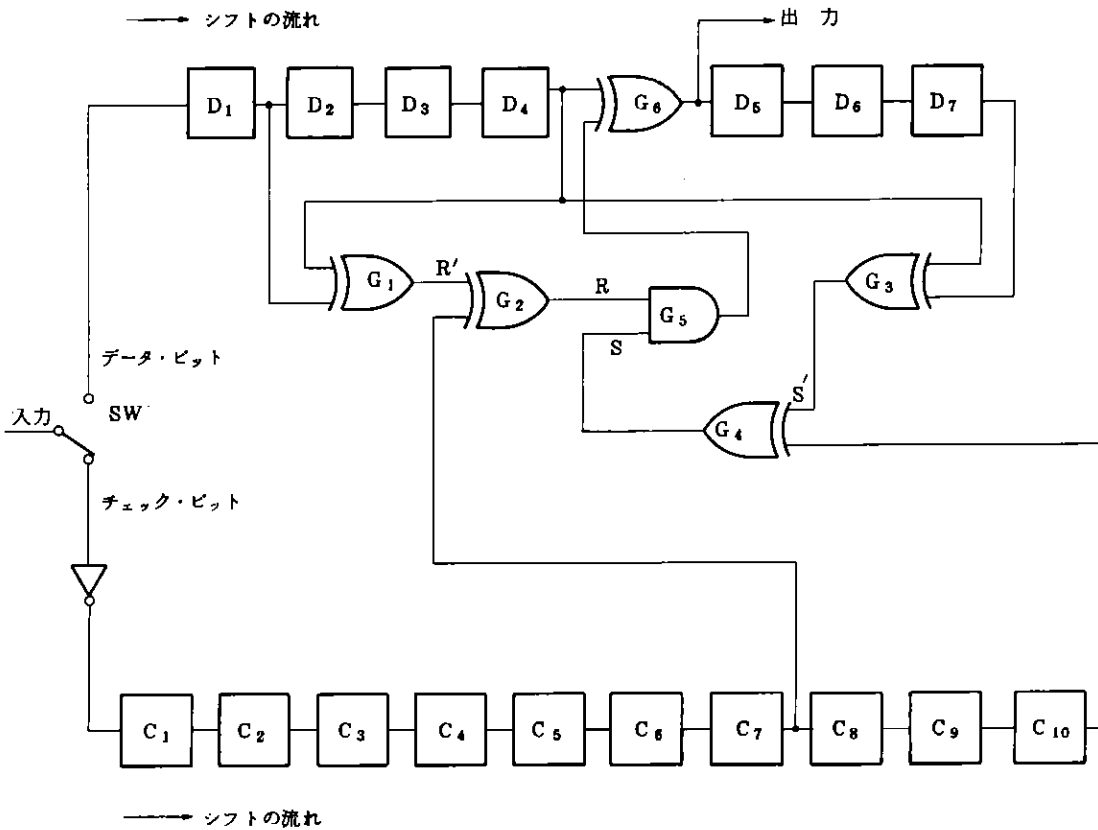


図4.7のようにハーゲルバーガ符号化した制御信号からは、チェック・ビットとデータ・ビットとが交互に入力されています。この時、ハーゲルバーガ復号化の誤り訂正は、図4.7のRとSのANDをとり、それとD₄とのEXORをとることにより行ないます。この時RとSとのAND出力が'1'だったらD₄は訂正されます。この訂正方法では、単独でのバースト誤りを6ビットまで訂正可能です。しかし、2箇所以上にバースト誤りが発生した時には、誤りと誤りの間に正しいデータが19ビット以上ないと誤り訂正ができません。しかし、誤りと誤りの間隔が6ビット以内なら一つのバースト誤りとして処理できます。

(2) データ・メモリ構成

4.2.1(2)参照のこと。

(3) 入力条件

なし

(4) 出力条件

RT : 復号化要求フラグがセットされていない時.

RTS : 復号化処理終了

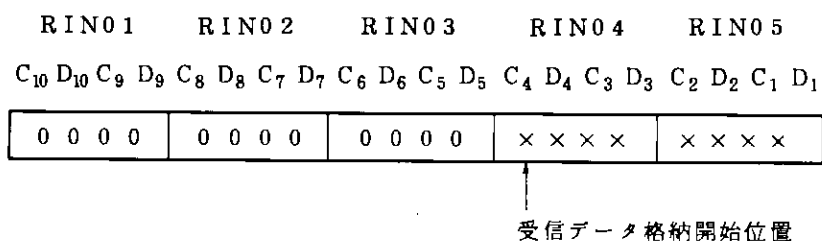
(5) 変化するレジスタ

A・H・L

(6) 処理説明

復号化要求フラグ(FDECD)がセットされていれば, 受信格納エリア(RIN01~RIN24)のデータをデコード・エリアを使用して, 以下の手順で復号化します.

- a) 入力した受信データの前に'0'を12ビット付加し, C₁₀ D₁₀~C₅ D₅と想定して復号化を開始します.



b) カウンタ・セット

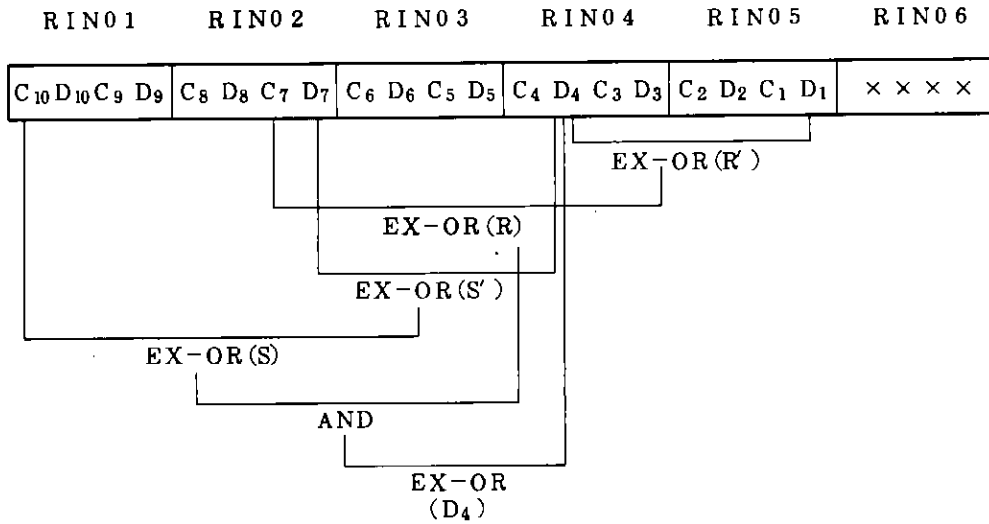
RIOCH, RIOCL : 2種類の復号化処理を一組とした繰返し回数(データ・カウンタ)

RADRH, RADRL : 復号化後データの格納アドレス

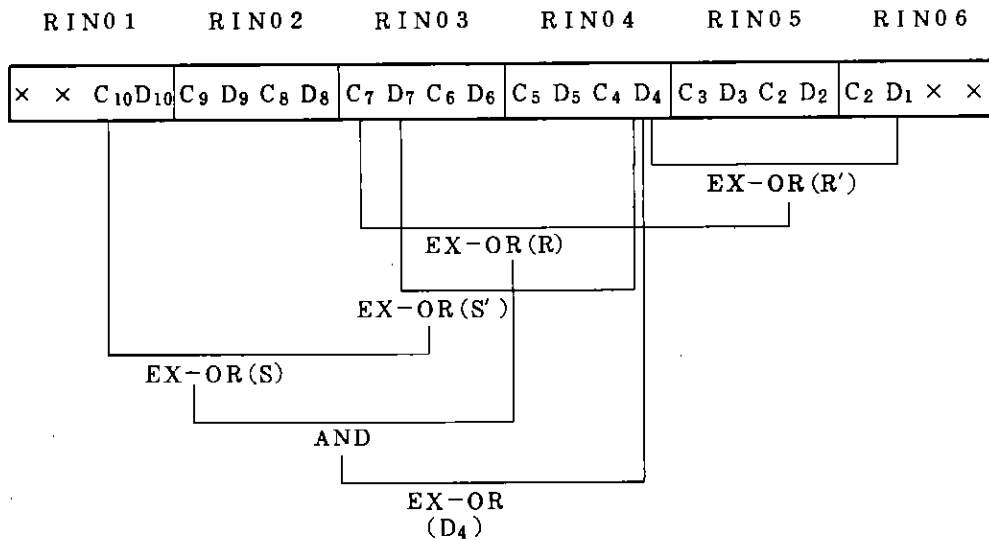
RBITC : 復号化後データの格納ビット位置(ビット・カウンタ)

なお, μ PD7508のデータ・メモリの1ワードは4ビットですので, データ・メモリの1ワードには復号化するデータ・ビットが2ビットあります. そのため, 1ワードのデータ・メモリ内のデータ・ビットを復号化する処理は, そのビット位置によってc), d)の2種類に分かれます.

c) 復号化処理 1



d) 復号化処理 2

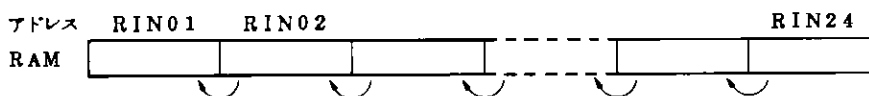


c) , d) の復号化処理によって D_4 は (式 4.5) に示すようになります。

$$\begin{aligned}
 D_4 &= D_4 \vee ((C_7 \vee (D_4 \vee D_1)) \wedge (C_{10} \vee (D_7 \vee D_4))) \dots \text{(式 4.5)} \\
 &= D_4 \vee ((C_7 \vee R') \wedge (C_{10} \vee S')) \\
 &= D_4 \vee (R \wedge S)
 \end{aligned}$$

e) シフト

繰返し回数カウンタにより復号化終了を判定し、終了でない場合は受信データ格納エリアを1ワード(4ビット)ずつ左シフトして復号化処理を繰返します。終了の場合は、RTSでメインに戻ります。



(7) フローチャート

F1/F2

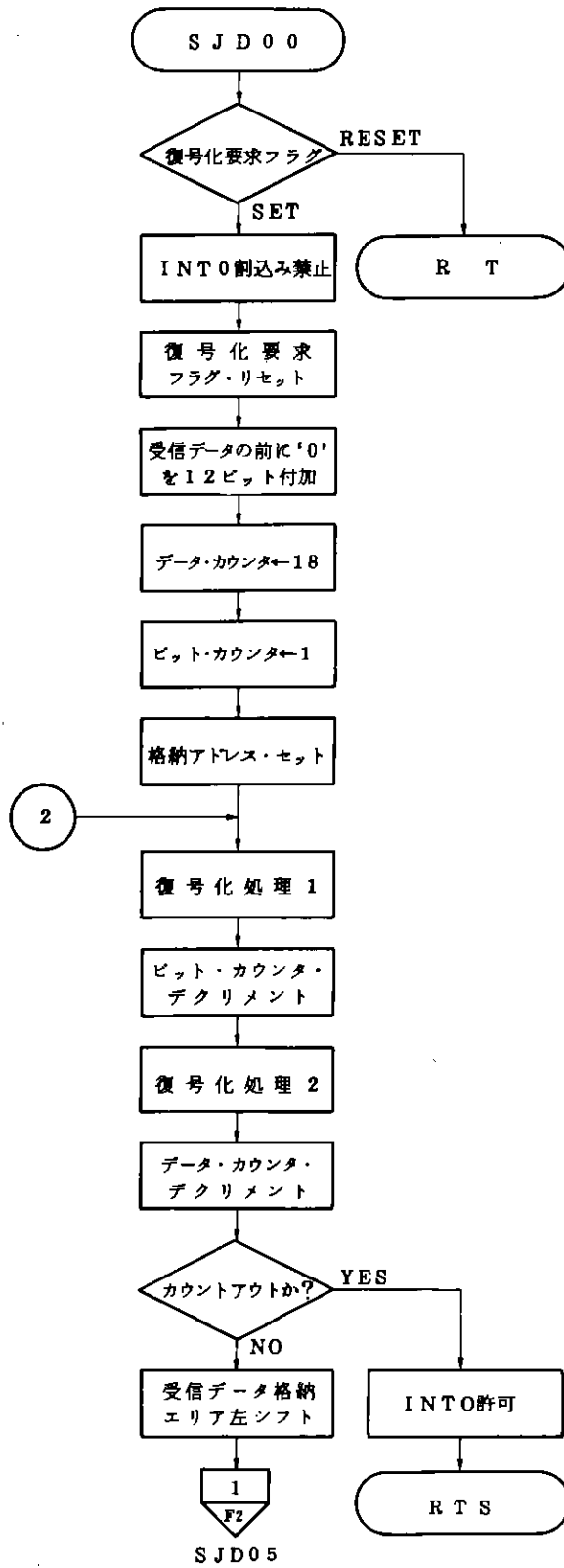
PROGRAM LABEL

SJD00:

SJD01:

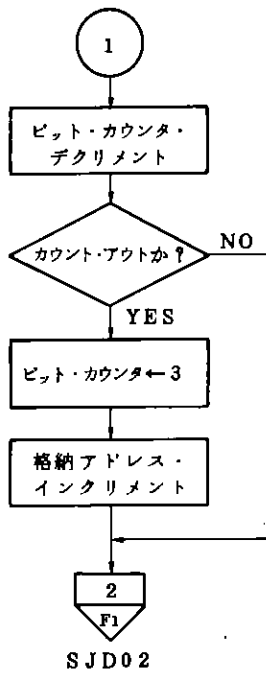
SJD02:

SJD03:



PROGRAM LABEL

SJD05:



(8) プログラム
バイト数 155

```

;
;*****
;*
;*      SJD00 - DECODE RECEIVE DATA SUBROUTINE
;*
;*      INPUTS   : NOTHING
;*      OUTPUTS  : RT ... INPUT FDECD RESET
;*                : RTS ... INPUT FDECD SET
;*      DESTROYS : A,H,L
;*
;*****
;

```

```

SJD00:
LHLI    RFLG1      ;
SKMBT   FDECD     ; SKIP IF SET FLAG DECODE REQ.
RT
DI       010B     ; DISABLE INT0
RMB     FDECD     ; RESET FLAG DECODE REQ.
LHLI    RIN03     ;

SJD01:
LAI     0          ;
XAM     HL-        ;
GJMP    SJD01     ;
LHLI    RIOCH     ;
LAI     1          ;
XAM     HL+        ;
LAI     2          ;
XAM     HL+        ; DATA COUNT
LAI     1          ;
XAM     HL+        ; BIT COUNT
LAI     (RDECD+1)/10H ;
XAM     HL+        ;
LAI     (RDECD+1) MOD 10H ;
ST      ; STORE ADDRESS

```

;
SJD02:

```

LHLI RIN04      :
LAM    HL+      :
CALL   SEXOR    : EX-OR D1:D4
LHLI   RIN02    :
EXL    : EX-OR C7 = R
XADR   RDECD    : SAVE
LADR   RIN04    :
CALL   SEXOR    : EX-OR D4:D7
LHLI   RIN01    :
EXL    : EX-OR C10 = S
LHLI   RDECD    :
CALL   SDECD    : AND R:S
LADR   RIN04    :
RAR    :
EXL    : DECODE
LHLI   RIN04    :
SMB    2        :
SKABT  1        :
RMB    2        :

```

```

:
XHDR   RADRH    :
XLDR   RADRL    :
SKABT  1        :
CALL   SRMBT    : SET '0' DATA
CALL   SSMBT    : SET '1' DATA
XHDR   RADRH    :
XLDR   RADRL    :
DDRS   RBITC    :

```

```

:
LHLI   RIN04    :
LADR   RIN06    :
CALL   SEXOR    : EX-OR D1:D4
LHLI   RIN03    :
EXL    : EX-OR D7 = R
XADR   RDECD    : SAVE
LAM    HL+      :
CALL   SEXOR    : EX-OR D4:D7
LHLI   RIN01    :
EXL    : EX-OR C10 = S
LHLI   RDECD    :
XAM    HL       :
CALL   SDECD    : AND R:S
LHLI   RIN04    :
RAR    :
EXL    : DECODE
LHLI   RIN04    :
SMB    0        :
SKABT  0        :
RMB    0        :

```

```

:
XHDR   RADRH    :
XLDR   RADRL    :
SKABT  0        :
CALL   SRMBT    : SET '0' DATA
CALL   SSMBT    : SET '1' DATA
XHDR   RADRH    :
XLDR   RADRL    :
DDRS   RIOCL    :
GJMP   SJD03    :
DDRS   RIOCH    :
GJMP   SJD03    :
EI     010B     : ENABLE INT0
RTS    : END DECODE

```



```

SJD03:  LHLI    RIN24      :
SJD04:  XAM     HL-       :
        GJMP    SJD04     :
        SKHEI   RIN17/10H :
        GJMP    SJD05     :
        LHI     RIN01/10H :
        GJMP    SJD04     :
SJD05:  DDRS    RBITC     :
        GJMP    SJD02     : LOOP
        LAI     3         :
        XADR    RBITC     :
        IDRS    RADRL     :
        GJMP    SJD02     : LOOP

```

```

:*****
:*
:*      SEXOR - EXCLUSIVE-OR LOGIC
:*
:*      INPUTS   : A(2)    ... DATA
:*               : (HL)(0) ... DATA
:*
:*      OUTPUTS  : A(1,3)
:*
:*      DESTROYS : A
:*
:*****

```

```

SEXOR:
        SKMBT   0         :
        CMA     :         :
        SKABT   2         :
        LAI     1010B     :
        LAI     0000B     :
        RT      :

```

```

:*****
:*
:*      SDECD - AND LOGIC (DECODER)
:*
:*      INPUTS   : A(3)    ... DATA
:*               : (HL)(1) ... DATA
:*
:*      OUTPUTS  : A(1)
:*               : (HL)(1)
:*
:*      DESTROYS : A
:*
:*****

```

```

SDECD:
        RAR     :
        RAR     :
        ANL     :
        ST      :
        RT      :

```

第5章 ATIS ROM インタフェース・サブルーチン

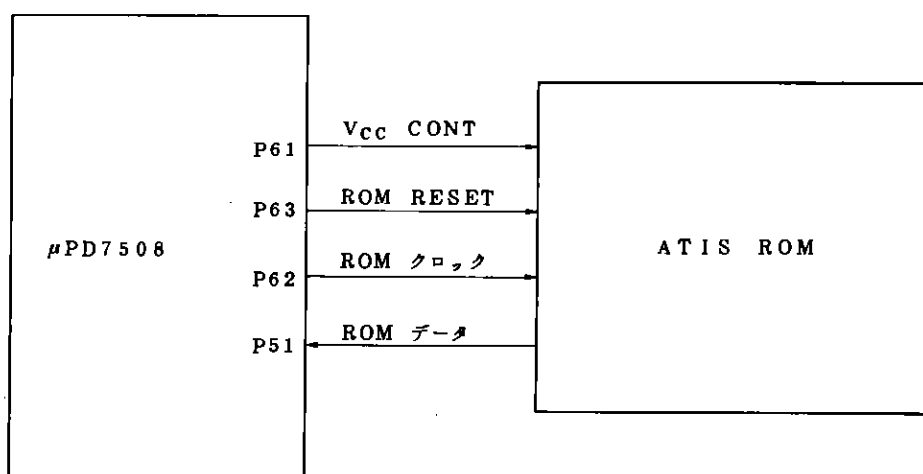
(SROM0)

(1) 処理内容

ATIS ROMは、制御信号の呼び出し符号部分(48ビット)を記憶しています。この呼び出し符号は、電波の監視をするための符号で、地域分け符号・クラス分け符号・免許番号符号・スクランブル符号・予備符号から構成されています。

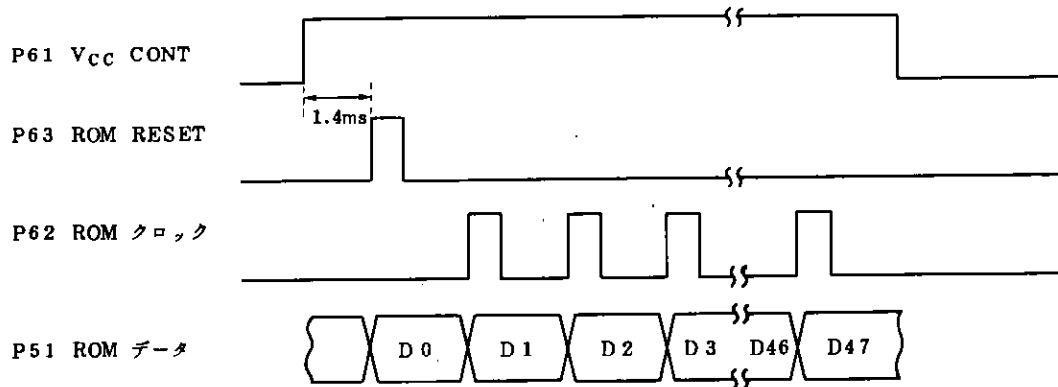
図5.1にATIS ROMと μ PD7508とのインタフェース構成を示します。

図5.1 ATIS ROMインタフェース構成図



ATIS ROMが記憶している呼び出し符号をROMデータとして図5.2に示すタイミングで入力し、データ・メモリのATIS ROMデータ・エリアに格納して、同時に電波管理局が指定するデータが書き込まれているかを判定します。この判定では、最初の4ビット(0001)およびD41~D43(101)の3ビットを検査します。ATIS ROMからのデータ入力は、Vcc CONT信号によってATIS ROMに電源を投入してからATIS ROMが安定な状態になるまで1.4 ms 待ってから行なっています。

図 5.2 ROMデータ入力のタイミング・チャート



(2) データ・メモリ構成

4.1(2)参照のこと.

(3) 入力条件

なし

(4) 出力条件

RT : ATIS ROMに指定されたデータが書き込まれていない時.

RTS : ATIS ROMに指定されたデータが書き込んである時.

(5) 変化するレジスタ

A, H, L

(6) 処理説明

- a) V_{CC} CONT信号を'H'にして1.4msの間ATIS ROMが安定状態になるのを待ちます.
- b) ROM RESET信号を出力します.
- c) ROMデータを入力して, ATIS ROMデータ・エリアに格納します. その際ビット・カウンタによってビット位置を決めています.
- d) ATIS ROMデータ格納エリア(70H~7BH)への格納が終了するまでROMクロック信号を出力してc)の操作を繰り返します.
- e) V_{CC} CONT信号を'L'にします.
- f) ROMデータに指定されたデータが書き込んであるかどうかの判断を行ない, 書き込んでなければRTでメインに戻り, 書き込んであればRTSでメインに戻ります.

(7) フローチャート

F1 / F2

PROGRAM LABEL

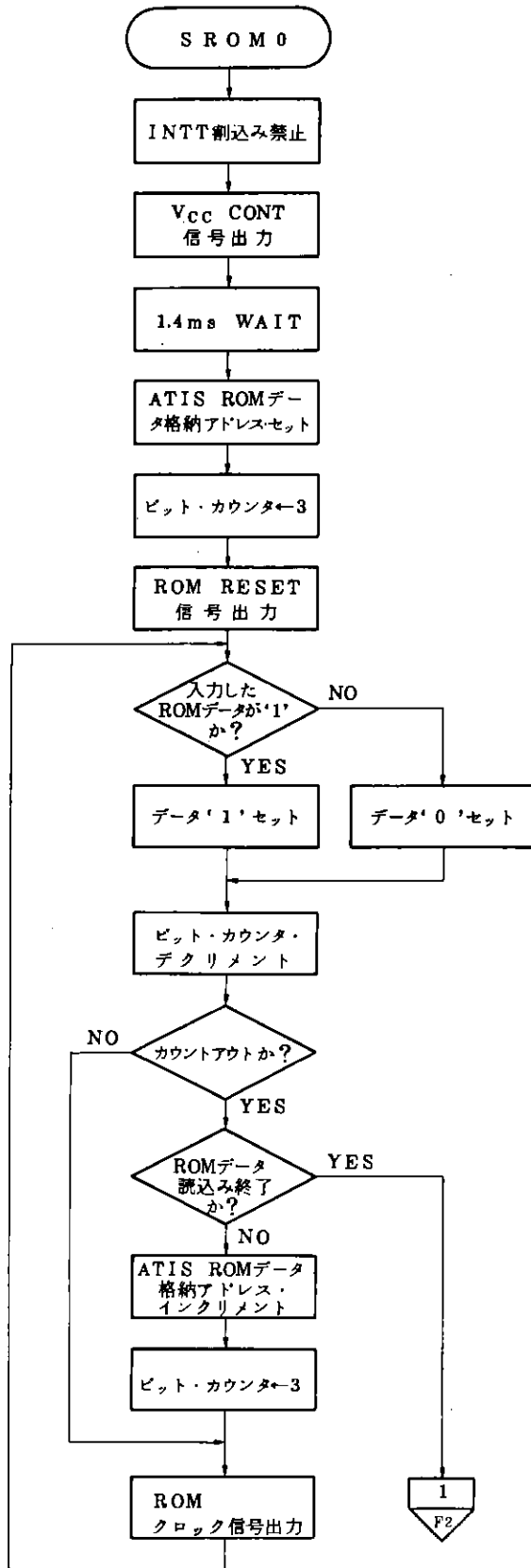
SROM0:

SROM2:

SROM3:

SROM4:

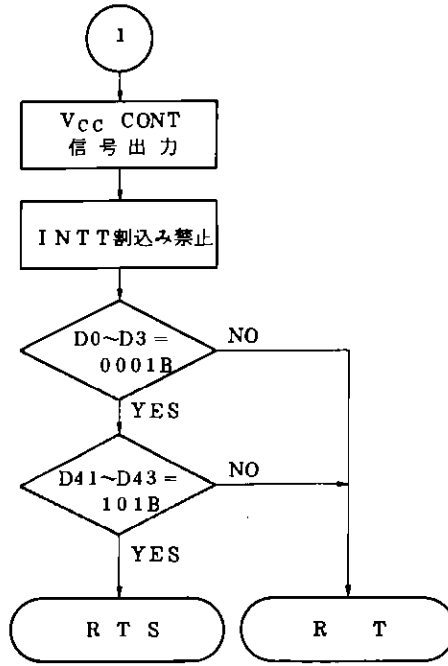
SROM5:



PROGRAM
LABEL

F 2 / F 2

SROM6 :



(8) プログラム

バイト数 60

```
*****
;*
;*      SROM0 - ATIS ROM INTERFACE SUBROUTINE
;*      P61 ... OUTPUT VCC CONT
;*      P63 ... OUTPUT ATIS ROM RESET
;*      P62 ... OUPUT ATIS ROM CLOCK
;*      P51 ... INPUT ATIS ROM DATA
;*
;*      INPUTS   : NOTHING
;*      OUTPUTS  : RT   ... ERROR READ ATIS ROM
;*                : RTS ... NORMAL READ ATIS ROM
;*      DESTROYS : A,H,L
;*
*****
;
SROM0:
    DI      1          ; DISABLE INTERRUPT TIMER
    ORP     6.0010B   ; VCC CONT <-- 'H'
    LAI     0CH       ;
    LLI     5          ; 1MS COUNT
SROM2:
    ILS
    GJMP    SROM2     ; LOOP
    AISC    1         ;
    GJMP    SROM2     ; LOOP
    LHLD   RROM0
    LAI     3         ;
    XADR    RBITC     ; SET BIT COUNTER
    ORP     6.1000B   ; ATIS ROM RESET
    ANP     6.0111B   ;
SROM3:
    IP      5          ; INPUT ATIS ROM DATA
    SKABT   1         ;
    CALL    SRMBT     ; SET '0' DATA
    CALL    SSMBT     ; SET '1' DATA
    DDRS    RBITC     ; DECREMENT BIT COUNTER
    GJMP    SROM5
    SKLEI   0BH       ;
    GJMP    SROM4
    GJMP    SROM6
SROM4:
    ILS
    LAI     3         ;
    XADR    RBITC     ; SET BIT COUNTER
SROM5:
    ORP     6.0100B   ; OUTPUT ATIS ROM CLOCK
    ANP     6.1011B   ;
    GJMP    SROM3     ; LOOP
;
SROM6:
    ANP     6.1101B   ; VCC CONT <-- 'L'
    EI      1         ; ENABLE INTERRUPT TIMER
    LADR    RROM0     ; ERROR CHECK
    SKAEI   0001B    ;
    RT      ; ERROR
    LHLD   RROMA
    SKMBT   0         ;
    LAI     1010B     ;
    LAI     1011B     ;
    SKAEM
    RT      ; ERROR
    RTS     ; NORMAL
```

第6章 参考資料

- (1) 「PLLを使った周波数シンセサイザ回路の基礎」
集積回路技術資料 IEB-550
- (2) 「μPD2833C PLL周波数シンセサイザ用LSI」
IC-6311
- (3) 「μPD6320G, 6321G FIP/LCDスタティック表示ドライバ」
新製品速報 IN-5997
- (4) 「μPD7225GプログラマブルLCDコントローラ/ドライバ」
新製品速報 IN-5825
- (5) 「μPD6302CA MSK MODEM LSI」
開発速報 ID-6316
- (6) 「パーソナル無線ガイドブック」
CQ ham radio 別冊
昭和58年1月15日 発行

なお、(1)～(5)は、弊社の技術資料および製品資料です。

○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、冗長対策設計、誤動作防止設計等安全設計に十分ご注意ください。

○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。

M4 04.11