MEGADRIVE用
サウンドツール資料

PRELIMINARY

YM2612

アプリケーションマニュアル

ヤマハ株式会社　半導体事業部

概　要

YAMAHA音源ＬＳＩ　YM２６１２は、Ｄ／Ａコンバータを内蔵したＦＭ方式の音源ＬＳＩであり、ＯＰＮ（ＹＭ２２０３）との互換性を保ちながら、更に６音同時発音を可能にしました。

特　徴

- ４オペレータ・６音同時発音。
- サイン波ＬＦＯ機能内蔵。
- ＹＭ２２０３とソフトウェアコンパチブル。
- ９ビットＤ／Ａコンバータ内蔵。２ｃｈ出力。
- マスタークロック・・・・８ＭＨｚ
- Ｎｃｈ-Ｓｉ　ｇａｔｅ　ＭＯＳ　ＬＳＩ
- ５Ｖ単一電源
- ２４ｐｉｎ　プラスチックＤＩＰ

A04062 9010 04

1. 構成と機能

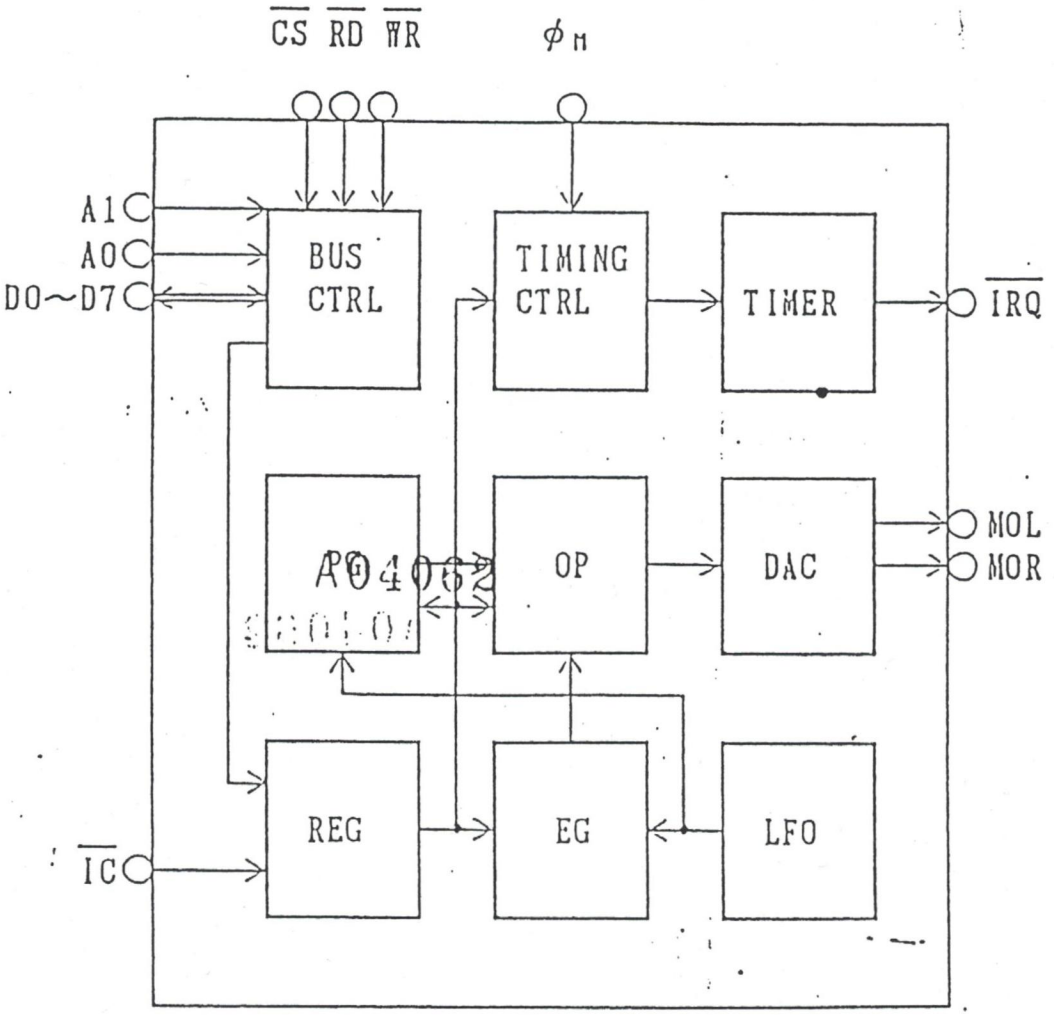
1－1．主要機能

ＹＭ２６１２の機能は、以下の通りです。
（基本的には、ＯＰＮ（ＹＭ２２０３）と同様）

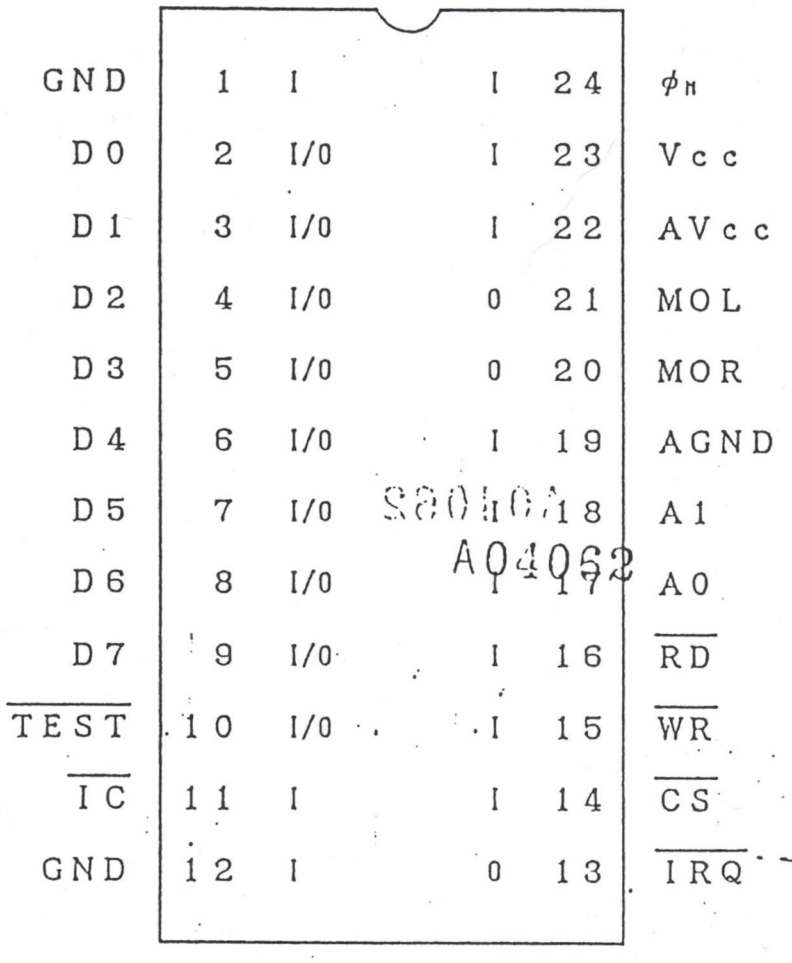
発音モード・・・・・・４オペレータＦＭ方式、6音同時発音。
アルゴリズム・・・・・８種類。
パラメータ・・・・・・レジスタアドレス、及びＦＭ音源部参照。
ＬＦＯ機能・・・・・・サイン波ＬＦＯ。ピッチ（ＰＭ）、及び振幅（ＡＭ）変調。
ＬＦＯ周波数可変。ＰＭＳ，ＡＭＳ制御と、オペレーク毎の
ＡＭ　ｏｎ／ｏｆｆ可能。
複合正弦波合成・・・・６音中１音可能。
タイマー機能・・・・・Ａ，Ｂの２種類のタイマー。
出力制御・・・・・・・９ビットＤ／Ａコンバーク内蔵。Ｌ及びＲのｏｎ／ｏｆｆ。
ＰＣＭ機能・・・・・・サンプリング・レート55.5kHz

AD1062

1－2．ブロックダイアグラム



1－3．端子配置図



S80A07
A04062

ϕ_M
マスタークロック入力です。

MOL・MOR
２チャンネルのアナログ出力です。ソースフォロワーで出力されます。

D0～D7
８ビットの双方向データバスです。プロセッサとデータのやり取りをします。

$\overline{CS}$・$\overline{RD}$・$\overline{WR}$・A1・A0
D0～D7のデータバスのコントロールをします。

CS	RD	WR	A1	A0	ｱﾄﾞﾚｽ 範囲	内容
0	1	0	0	0	$21～$2C	ﾀｲﾏｰ等のﾚｼﾞｽﾀ･ｱﾄﾞﾚｽを書き込みます。
					$30～$B6	ﾁｬﾝﾈﾙ1～3のﾚｼﾞｽﾀ･ｱﾄﾞﾚｽを書き込みます。
0	1	0	0	1	$21～$2C	ﾀｲﾏｰ等のﾚｼﾞｽﾀ･ﾃﾞｰﾀを書き込みます。
					$30～$B6	ﾁｬﾝﾈﾙ1～3のﾚｼﾞｽﾀ･ﾃﾞｰﾀを書き込みます。
0	1	0	1	0	$30～$B6	ﾁｬﾝﾈﾙ4～6のﾚｼﾞｽﾀ･ｱﾄﾞﾚｽを書き込みます。
0	1	0	1	1	$30～$B6	ﾁｬﾝﾈﾙ4～6のﾚｼﾞｽﾀ･ﾃﾞｰﾀを書き込みます。
0	0	1	0	0	$XX	ｽﾃｰﾀｽを読み出します。
1	X	X	X	X	$XX	D0～D7は高ｲﾝﾋﾟｰﾀﾞﾝｽになります。

A04062

$\overline{IRQ}$
二つのタイマーから出される割り込み信号です。タイマーにプログラムされた時間が経過すると、低レベルになります。オープンドレインで出力されます。

$\overline{IC}$
内部レジスタを初期化します。

$\overline{TEST}$
本ＬＳＩをテストするための端子です。どこにも接続しないで下さい。

GND・AGND
グランド端子です。

Vcc・AVcc
＋５Ｖ電源端子です。

1－5．データバスコントロール

レジスタアドレスの指定、及びデータのリード／ライトの為のデータバスコントロールは$\overline{CS}$・$\overline{WR}$・$\overline{RD}$・A1・A0の各信号で行われます。これらの信号の状態によってコントロール可能なアドレス範囲とデータの内容は、1－4．端子機能に示す表1．1のとおりです。

YM2612のレジスタは、アドレス$20～$B6を共用する2系列のレジスタバンクから構成されています。バンク指定は、バスコントロール信号A1によって行われます。この方法により、各レジスタのアクセスが可能になります。

A1によるレジスタアドレスの割り当ては次のとおりです。

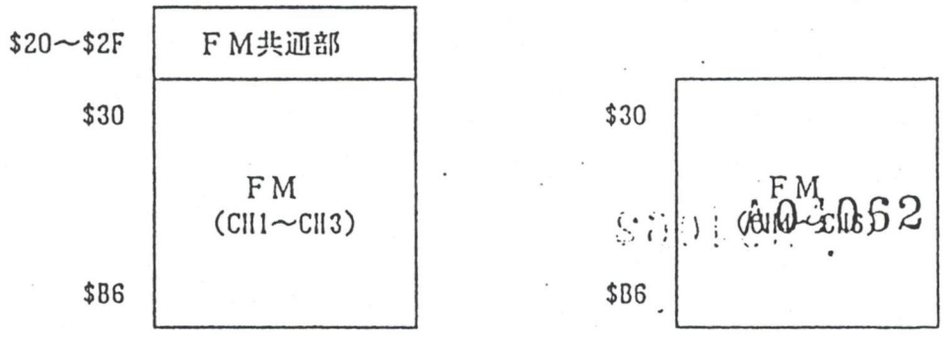


図1．1　レジスタアドレスの割り当て

$\overline{CS}$・$\overline{WR}$・$\overline{RD}$・A0・A1によるレジスタの制御モードは下表1．2のとおりです。

表1．2　リード／ライトモード選択

	$\overline{CS}$	$\overline{WR}$	$\overline{RD}$	A1	A0	モード
1	0	0	1	*	0	アドレス ライト モード
2	0	0	1	*	1	データ ライト モード
3	0	1	0	0	0	ステータス リード モード
5	1	×	×	×	×	インアクティブ モード

A1＝“*”は、バンク指定を行います。

1）アドレスライト　モード

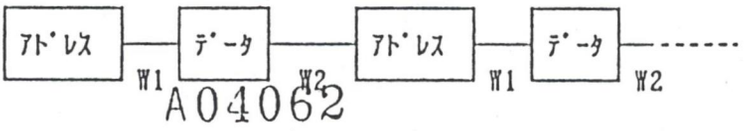
ＣＰＵからレジスタへデータを書き込む、又はレジスクからデークを読み出すためには、まずレジスタのアドレスを指定して、その後データの書き込み、読み出しを行います。

アドレス指定は、バスコントロール信号が＜アドレスライトモード＞の時、データバスＤ０～Ｄ７にアドレスデータを書き込みます。

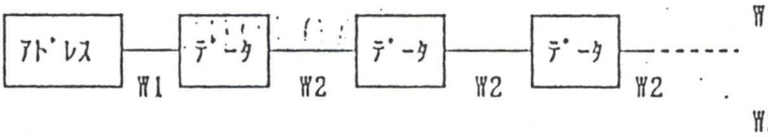
指定されたアドレスは、新たに次のアドレス指定が行われるまで保持されますので、同じアドレスを連続してアクセスするような場合、アドレスデータの書き込みは最初の１回だけでよくなり、その後は不要です。

2）データライト　モード

アドレス指定後、バスコントロール信号を＜データライトモード＞にして、データバス上のデータをレジスタに書き込みます。



A04062



W1：アドレスライト後の待ち時間

W2：データライト後の待ち時間

アドレス及びデータライトで注意しなければならないことは、書き込み終了後次の処理に移るまでに、所定の待ち時間を設定する必要があることです。これは、ＬＳＩ内部のデータ処理方法に起因しています。

従って、レジスタに正しくデータをセットするためには、待ち時間の設定を必ず行うようにして下さい。

なお、レジスタ書き込み時の待ち時間は、表１．３、１．４のとおりです。

3）ステータスリード　モード

バスコントロール信号を＜ステータスリードモード＞にした時、ステータスレジスクに発生するステータス情報がデータバス上に出力されます。

4）インアクティブ　モード

$\overline{\text{CS}}$が“１”の時、データバスＤ０～Ｄ７はハイインピーダンスになります。

《書き込みモード時の待ち時間》

表1．3　アドレスライト後

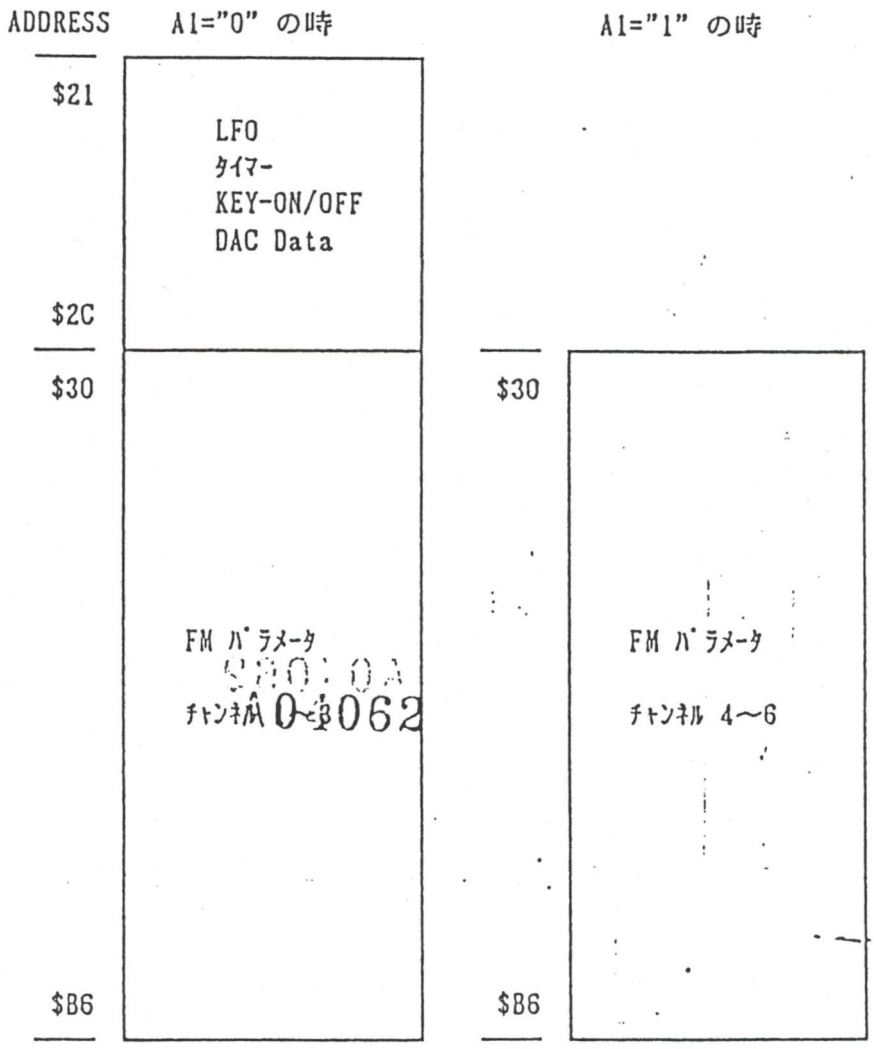
アドレス	待ちサイクル
$21～$B6	17

表1．4　データライト後

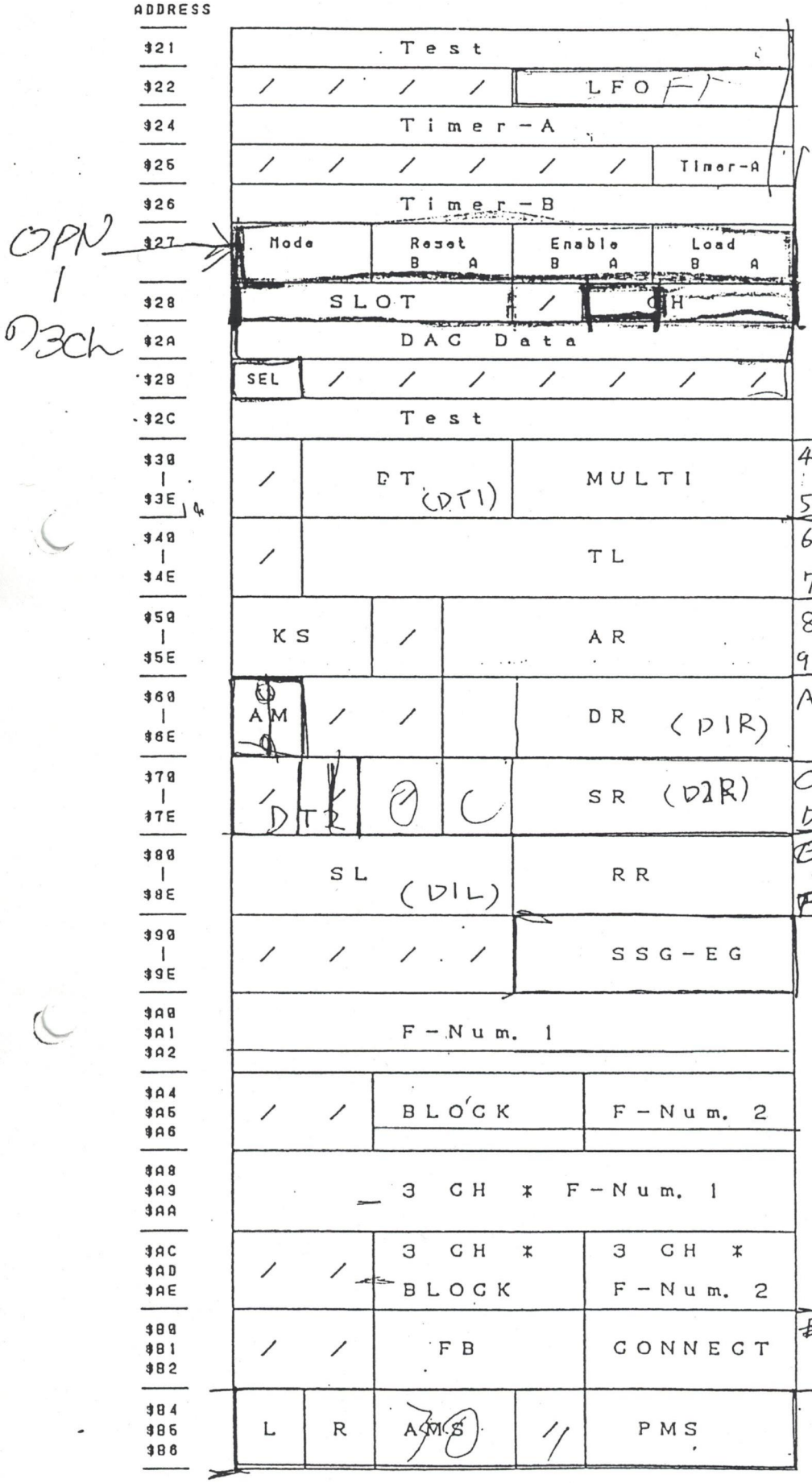
アドレス	待ちサイクル
$21～$9E	83
$A0～$A0	47

＊サイクル数は、マスタークロックφM のサイクル数です。

1－6．YM2612レジスタマップ



6. 2 WRITE DATA



1－7．ステータスフラグ

ADDRESS										COMMENT
								FLAG		
$XX	BUSY	/	/	/	/	/		A	B	Status

A04062

S30L0A

２．ＦＭ音源 諸機能

ＹＭ２６１２は、６音／６音色同時発音が可能な４オペレータＦＭ方式の音源ＬＳＩです。４オペレータ方式によるクオリティの高いＦＭサウンドに加えて、ＬＦＯ機能を内蔵したことにより、音色作りの可能性が飛躍的に向上しました。また、ＹＭ２２０３とソフトウェアの互換性がありますので、ＹＭ２２０３で作った音色が利用できます。

本章では、ＹＭ２６１２のレジスタ機能を中心に、オペレータを構成する各ブロックについて説明します。

S80108

２－１．レジスタ構成

オペレータを制御するパラメータは、＄２１～＄Ｂ６のレジスタに配置されています。ＦＭ音源の音創り及び発音制御は、各レジスタに適当なデータを書き込むことにより行われます。

A04052

２－１－１．共通レジスタ：＄２１～＄２８

音声出力の全チャンネルに共通な機能を配置したブロックです。ＬＦＯ、タイマー、チャンネル割当機能等で構成されています。

□ Ｔｅｓｔ：＄２１

ＹＭ２６１２テスト用レジスタです。ユーザーアプリケーションでは使用しません。

□ ＬＦＯ：＄２２　　（２－５．ＬＦＯ参照）

ＬＦＯのｏｎ／ｏｆｆ制御とＬＦＯのスピードを決める発振周波数の設定をします。

□ Ｔｉｍｅｒ－Ａ：＄２４，＄２５

タイマーは２種類のプリセッタブルタイマーと、各タイマーの始動、停止、フラグ制御を行うタイマーコントローラから構成されています。また、ステータス（リードモード）のＤ１，ＤＯに配置されたタイマーフラグが“１”になると同時にＩＲＱを発生して、ＣＰＵに対してタイマーインタラプトを行います。

タイマーＡは、＄２４、＄２５の１０ｂｉｔで作られる分解能９μＳ（φＭ=８MHz）のタイマーカウンターです。セット可能なタイムインターバルは、①式によって算出できます。

	D7	D6	D5	D4	D3	D2	D1	DO
$24	P9	P8	P7	P6	P5	P4	P3	P2
$25	／	／	／	／	／	／	P1	P0

tA ＝７２＊（１０２４－ＮＡ）／φＭ　・・・①

ＮＡ：０～１０２３
φＭ ：マスタークロック

（例）　φＭ ＝８MHz 時、

tA（MAX）＝ ９２１６μＳ
tA（MIN）＝ ９μＳ

□ Ｔｉｍｅｒ－Ｂ：＄２６

タイマーＢは、＄２６の８ｂｉｔで作られる分解能１４４μＳ（φＭ ＝８MHz ）のタイマーカウンターです。セット可能なタイムインターバルは、②式により算出できます。

	D7	D6	D5	D4	D3	D2	D1	DO
$26	P7	P6	P5	P4	P3	P2	P1	P0

tB ＝１１５２＊（２５６－ＮＢ）／φＭ　・・・②

ＮＢ：０～２５５
φＭ ：マスタークロック

□ Timer Control：$27

タイマーA及びBは$27のタイマーコントローラで制御されます。

	D7	D6	D5	D4	D3	D2	D1	D0
$27	MODE		RESET B	A	ENABLE B	A	LOAD B	A

・LOAD　：タイマーのスタート、ストップを制御します。

“1”でスタートします。
“0”でタイマーはストップします。

・ENABLE：ステータス（リードモード）のタイマーフラグを制御します。

“1”の時、タイマーカウンターがオーバーフローすると同時にタイマーフラグに“1”が立ちます。又、このタイマーフラグは$\overline{IRQ}$端子にインタラプト信号を発生します。
“0”の時、タイマーカウンターがオーバーフローしても、フラグは変化しません。

・RESET　：タイマーフラグをリセットします。

“1”の時、ステータスのタイマーフラグをリセットして、同時にこのｂｉｔ自身も“0”クリアされます。

・MODE　：チャンネル３のモードを設定します。

チャンネル３は、$27“D7、D6”によりモード設定が出来ます。

D7	D6	モード	機　　能
0	0	ノーマル	他のCHと同様、ノーマル発音をします。
0	1	CSM	CSM音声合成モードとなり、F-Numberは4スロット別々に設定できます。CSMモード時のKey-on/offはタイマーAを使って行います。
1	0	効果音	CSM同様、各スロットに別々のF-Numberを設定できます。

□ Ｋｅｙ　ｏｎ／ｏｆｆ：＄２８

チャンネル指定とスロットのｏｎ／ｏｆｆにより、チャンネル毎のＫｅｙアサインを行います。

	D7	D6	D5	D4	D3	D2	D1	D0
$28	SLOT				/	CH.		

D4	ｽﾛｯﾄ1のON/OFF
D5	ｽﾛｯﾄ2のON/OFF
D6	ｽﾛｯﾄ3のON/OFF
D7	ｽﾛｯﾄ4のON/OFF

0	0	0	ﾁｬﾝﾈﾙ 1
0	0	1	ﾁｬﾝﾈﾙ 2
0	1	0	ﾁｬﾝﾈﾙ 3
1	0	0	ﾁｬﾝﾈﾙ 4
1	0	1	ﾁｬﾝﾈﾙ 5
1	1	0	ﾁｬﾝﾈﾙ 6

A04062

□ Ｔｅｓｔ：＄２Ｃ

ＹＭ２６１２テスト用レジスタです。ユーザーアプリケーションでは使用しません。

2－1－2．Ｄ／Ａ用レジスタ：＄２Ａ，＄２Ｂ

ＹＭ２６１２は、チャンネル６のＦＭ音の代わりに、ＣＰＵから書き込まれたデータを、直接Ｄ／Ａコンバータに送り込む事ができます。この機能を実現するためのレジスタです。

□ ＤＡＣ　Ｄａｔａ：＄２Ａ

Ｄ／Ａ変換をさせたいデータを与えます。データは、オフセット・バイナリにて作成して下さい。
但し、ｂｉｔ０がＬＳＢ、ｂｉｔ７がＭＳＢです。

□ ＤＡＣ　Ｓｅｌｅｃｔ：＄２Ｂ

チャンネル６に、ＦＭ音を出力するか、あるいは、レジスタ＄２Ａに書き込まれたデータを出力するかを選択するためのレジスタです。
このレジスタのビット７に”０”を立てた時は、ＦＭ音を、”１”を立てた時は、ＤＡＣ　Ｄａｔａを出力します。

2－1－3．チャンネルレジスタ：＄３０～＄Ｂ６

３チャンネル分の音色パラメーター、Ｆ－Ｎｕｍｂｅｒ等を配置したレジスタです。＄３０～＄９ＥはスロットＭ位に設定するパラメータ、＄Ａ０～＄Ｂ６はチャンネル単位に設定するパラメータです。
チャンネル選択は、バスコントロール信号：Ａ１で制御します。

Ａ１＝“０”の時、ＣＨ１～ＣＨ３のパラメータ制御。
Ａ１＝“１”の時、ＣＨ４～ＣＨ６の　　〃

レジスタアドレスとチャンネル・スロットの関係は表２．１の通りです。

表２．１　レジスタアドレスとチャンネル・スロットの関係

SLOT / パラメータ		CH1／CH4				CH2／CH5				CH3／CH6			
		1	3	2	4	1	3	2	4	1	3	2	4
DT/MULTI	3*	30	34	38	3C	31	35	39	3D	32	36	3A	3E
TL	4*	40	44	48	4C	41	45	49	4D	42	46	4A	4E
KS/AR	5*	50	54	58	5C	51	55	59	5D	52	56	5A	5E
AM/DR	6*	60	64	68	6C	61	65	69	6D	62	66	6A	CE
SR	7*	70	74	78	7C	71	75	79	7D	72	76	7A	7E
SL/RR	8*	80	84	88	8C	81	85	89	8D	82	86	8A	8E
SSG-EG	9*	90	94	98	9C	91	95	99	9D	92	96	9A	9E
F-Num1		A0				A1				A2			
Block/F-Num2		A4				A5				A6			
3CH*F-Num1 *1										A9	A8	AA	A2
3CH*(Block/F-Num2) *1										AD	AC	AE	A6
FB/Algorithm		B0				B1				B2			
L/R,AMS/PMS		B4				B5				B6			

＊１：＄Ａ８～＄ＡＡ，＄ＡＣ～＄ＡＥは、チャンネル３を効果音モード、又はＣＳＭ音声合成モードにした時の周波数（Ｂｌｏｃｋ，Ｆ－Ｎｕｍｂｅｒ）設定レジスタです。通常の発音モードでは使用しません。
チャンネル３のモード設定については、２－１－１を参照してください。

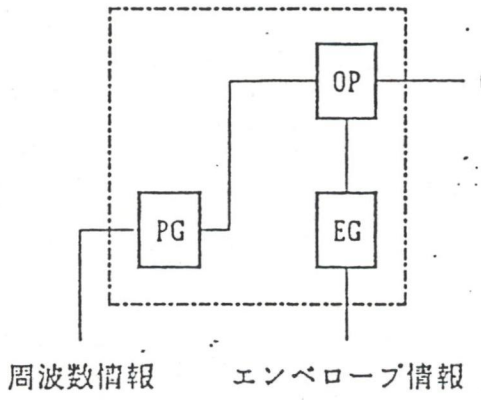
2－2．オペレータ

2－2－1．オペレータ

ＦＭ音源は複数のオペレータの組み合わせでさまざまな音色を創ることができます。しかし、１個のオペレータは、レジスタに書き込まれたデータにしたがって演算を行うだけの機能にすぎません。

> オペレータとは・・・与えられた周波数情報と出力レベル（エンベロープ）情報に応じてサイン波を発生する正弦波発振器である。

オペレータは下図のようなブロック図で表現されます。



ＯＰ：サインテーブル。

ＰＧ：フェイズジェネレータ。周波数（位相）情報発生回路。サインテーブルのデータ読み出し速度を発生する。

ＥＧ：エンベロープジェネレータ。サインテーブルの出力レベル、及び出力レベルの時間的変化を制御する。

つまり、ＰＧに周波数情報、ＥＧにエンベロープ情報を与えれば、オペレータはサイン波を出力します。しかし、これだけではサイン波しか得られませんから、余り面白くありません。そこで、複数のオペレータを接続することで、倍音成分を含んだ音色を作成できるようにしました。

> ＦＭの原理：・・サイン波でサイン波を変調することにより、複雑な倍音構成を持つ波形を作ることが出来る。

図２．１にＦＭ方式の基本となる２オペレータＦＭのブロック図を示します。

ＹＭ２６１２は、４個のオペレータを接続してＦＭ音色を作ります。この接続方法の事をアルゴリズムと言い、８種類のアルゴリズムを選択することができます。

→２－２－２アルゴリズム参照

> アルゴリズム・・・オペレータの接続形態（組合せ方）の事。４オペレータＦＭでは８種類ある。アルゴリズムにより、各オペレータの役割が決められる為に音色作成に際して、まず初めに決定するパラメータ。

図２．１のβはフィードバック変調の帰還率を表しています。フィードバックは、自分自身の出力を入力にフィードバックして、自己変調をかけます。この効果は、オペレータが無限に接続された事と等しく、倍音構成が整数次の高調波となることから、鋸歯状波（のこぎり波）的な波形，例えば、ストリングス系の音色等に効果的です。フィードバック機能は接続形態の一つと考えて良いでしょう。

→２－２－２アルゴリズム参照

オペレータは、変調する側をモジュレータ、被変調（有無にかかわらず）出力側をキャリアと呼び区別します。しかし、オペレータ本来の機能が変わる訳ではなく、アルゴリズムによりモジュレータにもキャリアにも成り得ます。

これはアルゴリズムを選択した際に、それぞれのオペレータがキャリアとして働くものか、モジュレータとして働くものかといった役割を表す名称にすぎません。

図２．１ではＯＰ１がモジュレータ、ＯＰ２がキャリアとなります。

これまでの事をまとめると、ＦＭ音源の音創りは以下の設定を行えばよいことになります。

> Ａ．アルゴリズムを選択して、各オペレータの役割を決める。
>
> Ｂ．ＰＧ（フェイズジェネレータ）のパラメータを設定して、各オペレータの出力周波数を決める。
>
> Ｃ．ＥＧ（エンベロープジェネレータ）のパラメータを設定して、各オペレータのエンベロープ及び出力レベルを決める。
>
> Ｄ．音色によりフィードバックＦＭを使用する。

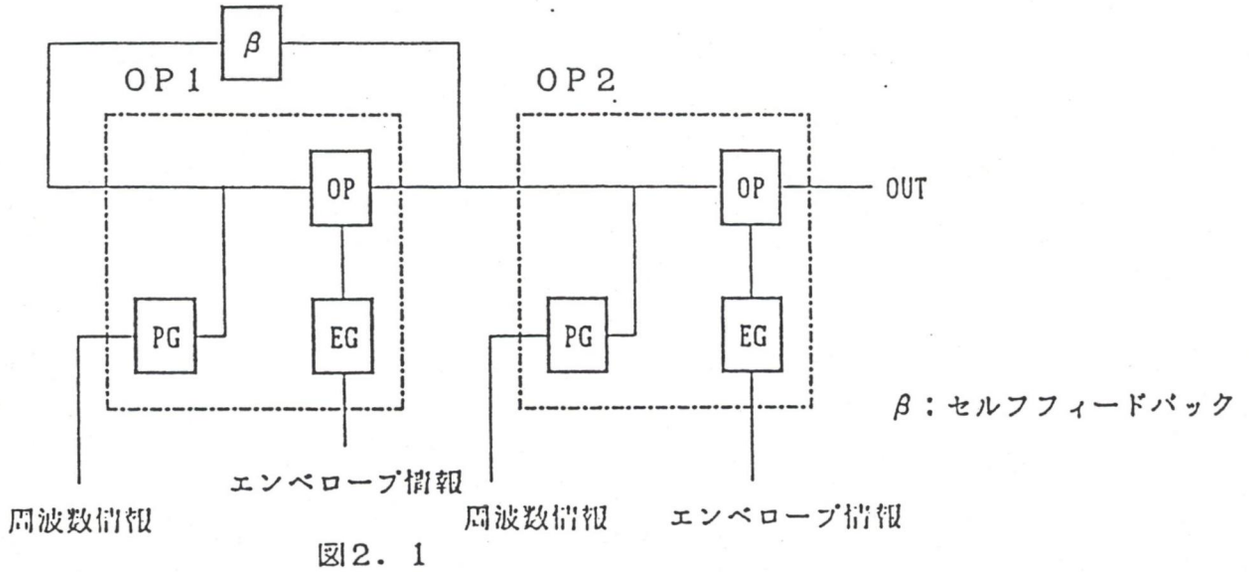


図２．１

2－2－2．アルゴリズム

オペレータの組合せ（接続形態）のことをアルゴリズムといいます。ＹＭ２６１２のような４オペレータＦＭ方式には、アルゴリズムは８種類用意されており、オペレータはアルゴリズムを選択することにより、キャリアあるいはモジュレータとして機能します。

但し、第４スロットはアルゴリズムに関係なく、必ずキャリアに設定されます。

アルゴリズムの選択は、ＦＭ音源の音創りの中では最も重要な要素となります。音創りの基本手順は、まず最初に目的の音色に最も適したアルゴリズムを選択することから始めます。その後、各スロット毎にパラメータを設定して、音色を決めていきます。

図２．２にアルゴリズムの形態を示します。さらに、以下に、各アルゴリズムの特徴を示します。

・各アルゴリズムの特徴　　2001-0A

１）シリアル４連モード

４個のスロットを直列接続して、多重変調を行います。多重変調方式は、連続的な変調の繰り返しの結果、最終のキャリア出力では非常に複雑な倍音構成が得られるます。
S4とS3で基本となる音色を作り、S2，S1では倍音成分を調整して微妙な音色の味付けをします。　　A04062

２）ダブル変調シリアル３連モード

S3をS2，S1の合成出力で変調します。１）同様、S4,S3で基本音色を作り、S2,S1のパラメータの設定で、さらに細かい音作りをします。

３）ダブル変調モード①

S4を２系列のモジュレータで変調しています。基本となる音色はS1とS4で作り、S2とS3で音色に自然感を与える付加音を加えます。

４）ダブル変調モード②

３）とよく似ていますすが、S3にはセルフフィードバックがかけられませんので、フルートのような木管楽器の音に適しています。S2，S1ではノイズ成分を作ります。

１）～４）は、キャリアが１個ですから、単一音色系となり、複雑な倍音成分を持ったソロ楽器の音色に向いています。

５）シリアル２連・２パラレルモード

２オペレータ２系列構成のアルゴリズムです。このモードは、倍音成分の多い音色にはやや不向きですが、音創りが比較的簡単なことと、２種類の音色を作成できることから、幅広い音作りに使用できます。

6）共通変調３パラレルモード

共通のモジュレータS1が、三個のキャリアS2，S3，S4を変調します。

7）シリアル２通＋２サインモード

一個の２オペレータＦＭと二個のサイン波の合成出力が得られます。

8）４パラレル・サイン波合成モード

四つのサイン波を合成した出力が得られます。ただし、S1はフィードバックをかける事で歪んだ音を作ることが可能です。

キャリアが複数のアルゴリズムでは、各キャリアの周波数情報に関するパラメータが音創りの決め手となります。

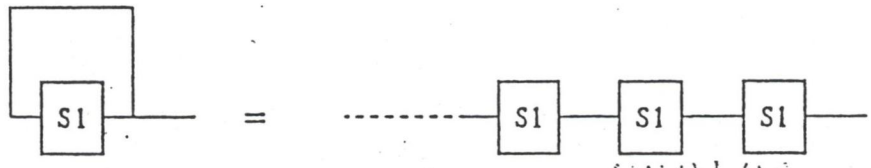
例えば、アルゴリズム“7”ではキャリア毎のマルチプルを違う値にすることで、オルガンサウンドのようなカプラー効果が得られます。またディチューンの設定値により、ピッチをわずかにずらすことで音にうねりが生じ、いわゆるコーラス（ディチューン）効果が作れます。

S30110A

2－2－3．フィードバック

各チャンネルの第１スロットは、セルフフィードバック機能を持っています。フィードバックとは、オペレータが自分自身の出力を変調信号として入力に帰還させることにより、自己変調を行う機能の事です。帰還率βは変調度を表し、０～７の８ステップの設定ができます。

フィードバックは、同じ周波数に設定された複数個のオペレータを、直列に接続したことと等しい状態になります。そして、この効果は、高調波成分が一様に分布された整数次の倍音構成、即ち、鋸歯状波に見られる高調波スペクトルとなり、ストリングス系の音色、あるいは変調度を深くすることでノイズ成分の発生等に適しています。



A04062

□ＦＢ／Ａｌｇｏｒｉｔｈｍ：$Ｂ０～$Ｂ２

セルフフィードバックの変調度、及びアルゴリズムを設定するレジスタです。

	D7	D6	D5	D4	D3	D2	D1	D0
$B0～$B2	／	／	Feedback			Algorithm		

表２．２にフィードバック変調度を示します。

表２．２　セルフフィードバックの変調度

Feedback	0	1	2	3	4	5	6	7
変調度	OFF	π/16	π/8	π/4	π/2	π	2π	4π

2－3．ＰＧ　（フェイズ・ジェネレータ）

オペレータの出力周波数（ＯＰのサインテーブルを、与えられた位相値で読み出す速度）は、ＰＧが発生する周波数（位相）情報によって決まります。つまり、位相値の増減を行うことで、オペレータは任意の出力周波数で発音が可能です。。

位相値即ち周波数情報は、Ｆ－Ｎｕｍｂｅｒ／Ｂｌｏｃｋ，Ｍｕｌｔｉｐｌｅ，Ｄｅｔｕｎｅの各パラメータで行います。

2－3－1．Ｆ－ＮｕｍｂｅｒとＢｌｏｃｋ

楽音の音階は、１オクターブ内の音程とオクターブの組み合わせで構成されています。そこで、Ｆ－Ｎｕｍｂｅｒで１オクターブ内の音程を作り、Ｂｌｏｃｋでオクターブ情報を設定すれば、８オクターブの音階列が簡単に作成できます。

A04062

１オクターブ内のＦ－Ｎｕｍｂｅｒ値は、マスタークロックと、必要とする音程の周波数が決まれば、次式を用いて算出することができます。

$$F\text{-Number} = (144 \times f_{note} \times 2^{20} / \phi M) / 2^{B-1}$$

fnote ：　発音周波数
φM　：　マスタークロック
B　　：　ブロックデータ

（例）φM=8ＭＨｚの時、A4（４４０Ｈｚ）のＦ－Ｎｕｍｂｅｒを求める。

$$F\text{-Number}(A4) = (144 \times 440 \times 2^{20} / 8 \times 10^{6}) / 2^{4-1}$$
$$= 1038.1$$

□　Ｆ－Ｎｕｍｂｅｒ／Ｂｌｏｃｋ：＄Ａ０～＄Ａ２／＄Ａ４～＄Ａ６

Ｆ－ＮｕｍｂｅｒとＢｌｏｃｋデータを設定するレジスタです。Ｆ－Ｎｕｍｂｅｒは下位８ｂｉｔ／上位３ｂｉｔの合計１１ｂｉｔ，Ｂｌｏｃｋは３ｂｉｔで構成されています。このデータは、チャンネル内の４個のオペレータに共通のデータとして働きます。

A4～A6 ／A0～A2	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
	／	／	Ｂｌｏｃｋ			F-Num2			F-Num1							

Ｆ－Ｎｕｍｂｅｒ／Ｂｌｏｃｋデータの設定は、必ず以下の手順で行って下さい。

①Ｂｌｏｃｋ／Ｆ－Ｎｕｍ2 のデータライト：＄Ａ４～＄Ａ６

②Ｆ－Ｎｕｍ1 のデータライト：＄Ａ０～＄Ａ２

2－3－2．Ｆ－Ｎｕｍｂｅｒテーブルの設定例

φＭ＝8ＭＨｚ、オクターブ：４（Ｃ4#～Ｃ5）、Ａ4＝４４０Ｈｚ．

Ｎｏｔｅ	音程 [Hz]	Ｆ－Ｎｕｍｂｅｒ	Ｆ11～Ｆ9	Ｆ8～Ｆ5	Ｆ4～Ｆ1	Key Code		
			D2D1D0	D7D6D5D4	D3D2D1D0	N4	N3	分割
Ｃ#	２７７．２	６５４．０	０１０	１０００	１１１０	０	０	０
Ｄ	２９３．７	６９２．８	０１０	１０１１	０１００	０	０	０
Ｄ#	３１１．１	７３４．０	０１０	１１０１	１１１０	０	０	０
Ｅ	３２９．６	７７７．７	０１１	００００	１００１	０	０	０
Ｆ	３４９．２	８２３．９	０１１	００１１	０１１１	０	０	０
Ｆ#	３７０．０	８７２．９	０１１	０１１０	１０　０	０	０	０
Ｇ	３９２．０	９２４．８	０１１	１００１	１１００	０	１	１
Ｇ#	４１５．３	９７９．８	０１１	１１０１	００１１	０	１	１
Ａ	４４０．０	１０３８．１	１００	００００	１１１０	１	０	２
Ａ#	４６６．２	１０９９．８	１００	０１００	１１００	１	０	２
Ｂ	４９３．９	１１６５．２	１００	１０００	１１０１	１	１	３
Ｃ	５２３．３	１２３４．５	１００	１１０１	００１０	１	１	３

2－3－3．Ｆ－ＮｕｍｂｅｒによるＫｅｙ　Ｃｏｄｅの作成

エンベロープジェネレータのパラメータの一つに、Ｋｅｙ－ｓｃａｌｅ機能があります。Ｋｅｙ－Ｓｃａｌｅは、発音する音程に対応してエンベロープレート（時間）が変化するスケーリングを持たせることができます。（2－4．ＥＧ参照）.

キースケーリング時に必要な１オクターブ内の分割は、Ｆ－Ｎｕｍｂｅｒ値を利用して行いますが、この分割データのことをＫｅｙ　Ｃｏｄｅといいます。

Ｆ－Ｎｕｍｂｅｒデータの上位４ｂｉｔ（Ｆ11～ Ｆ8）から、１オクターブ内を４分割します。

＊　１オクターブ内の周波数分割（Ｎ４，Ｎ３）

$$N4 = F11$$
$$N3 = F11 \cdot (F10 + F9 + F8) + \overline{F11} \cdot F10 \cdot F9 \cdot F8$$

さらに、Ｂｌｏｃｋデータ３ｂｉｔから、８オクターブ全音域における３２段階のＫｅｙ Ｃｏｄｅが作られます。

また、次項に説明するＤｅｔｕｎｅについても、このＫｅｙ　Ｃｏｄｅによって周波数分割行われます。

2－3－4. Multiple

Multipleは、F－Numberによって作られた周波数情報の倍率を設定するパラメータです。設定可能な倍率は、表2．3のとおりです。

2－3－5. Detune

Detuneは、F－Numberによって作られた周波数情報を、各スロット単位に微妙な周波数ずれを与えるパラメータです。また、DetuneはF－Numberから得られるKey Codeによって各周波数情報に対応する値をとります。

□ Detune／Multiple

	D7	D6	D5	D4	D3	D2	D1	D0
$30～$3E	／	Detune			Multiple			

表2．3 Multipleによる倍率

Multiple(H)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
倍 率	1/2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

A04062

表2．4 Detune

Detune	0	1	2	3	4	5	6	7
FD	0	1	2	3	0	-1	-2	-3

＊D6はサインビット。

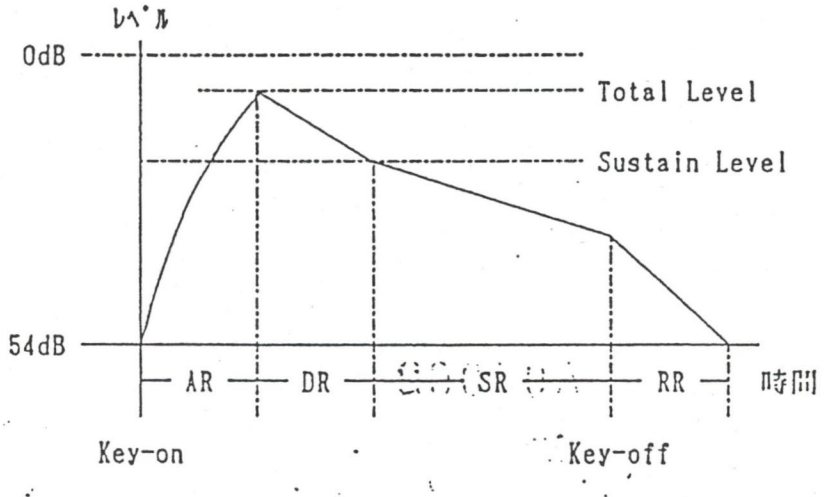
BLOCK	NOTE	FD=0	FD=1	FD=2	FD=3	BLOCK	NOTE	FD=0	FD=1	FD=2	FD=3
0	0	0.000	0.000	0.053	0.106	4	0	0.000	0.106	0.264	0.423
0	1	↑	↑	↑	↑	4	1	↑	0.159	0.317	0.423
0	2	↑	↑	↑	↑	4	2	↑	↑	↑	0.476
0	3	↑	↑	↑	↑	4	3	↑	↑	0.370	0.529
1	0	↑	0.053	0.106	↑	5	0	↑	0.212	0.423	0.582
1	1	↑	↑	↑	0.159	5	1	↑	↑	↑	0.635
1	2	↑	↑	↑	↑	5	2	↑	↑	0.476	0.688
1	3	↑	↑	↑	↑	5	3	↑	0.264	0.529	0.741
2	0	↑	↑	↑	0.212	6	0	↑	↑	0.582	0.846
2	1	↑	↑	0.159	↑	6	1	↑	0.317	0.635	0.899
2	2	↑	↑	↑	↑	6	2	↑	↑	0.688	1.005
2	3	↑	↑	↑	0.264	6	3	↑	0.370	0.741	1.058
3	0	↑	0.106	0.212	↑	7	0	↑	0.423	0.846	1.164
3	1	↑	↑	↑	0.317	7	1	↑	↑	↑	↑
3	2	↑	↑	↑	↑	7	2	↑	↑	↑	↑
3	3	0.000	↑	0.264	0.370	7	3	0.000	↑	↑	↑

２－４．ＥＧ　（エンベロープ・ジェネレータ）

ＥＧは、音の立ち上がりから消滅までの音量、音色の時間的変化を発生する回路です。
ＥＧは、エンベロープを発生するエンベロープジェネレータとエンベロープジェネレータのレベルを決める出力制御回路で構成されています。ＥＧを起動するためのエンベロープ情報は、レジスタに配置したＥＧ用のパラメータによって各オペレータごとに設定されます。

２－４－１．エンベロープジェネレータ

音の時間的変化を形作るエンベロープを発生します。エンベロープはアタック、ディケイ、サスティーン、リリースの四つのレートとサスティーンレベルによって表されます。
図２．３にエンベロープ波形を示します。



＊　エンベロープの波形は、アタック時は指数関数的に変化して、その他のレート時は直線的に変化します。

図２．３　エンベロープの波形と各パラメータ

□　ＡＲ（Ａｔｔａｃｋ　Ｒａｔｅ）：＄５０～＄５Ｅ

アタックレートとは、Ｋｅｙ－ｏｎした瞬間から最大レベルに到達する速さの事で、ＡＲはこのレートを決めるパラメータです。５ｂｉｔ；３２ステップで設定し、ＡＲが大きいほど立ち上がりは速くなります。また、“０”にするとアタックレートは無限大となり、ＥＧが起動しないためにエンベロープの変化は得られません。

	D7	D6	D5	D4	D3	D2	D1	D0
$50～$5E	KS		／		Attack Rate			

※ＫＳは次項参照。

□　ＤＲ（Ｄｅｃａｙ　Ｒａｔｅ）：＄６０～＄６Ｅ

ディケイレートとは、最大レベルからサスティーンレベルに減衰する速さの事で、ＤＲはこのレートを決めるパラメータです。５ｂｉｔ；３２ステップで設定し、ＤＲが大きいほど速く減衰します。また、“０”にするとディケイレートは無限大となり、最大レベルで持続音になります。

	D7	D6	D5	D4	D3	D2	D1	D0
$60～$6E	AMON	／	／		Decay Rate			

※ ＡＭＯＮは２－５．ＬＦＯ参照。

□ ＳＬ（Ｓｕｓｔａｉｎ　Ｌｅｖｅｌ）：＄８０～＄８Ｅ

サスティーンレベルとは、ディケイレートからサスティーンレートに切り替わる点のレベル（減衰量）の事で、ＳＬはこのレベルを決めるパラメータです。４ｂｉｔ；１６ステップで設定し、ＳＬが大きいほど減衰量は大きくなります。“０”にすると減衰量が０となり、ディケイによる減衰感が得られなくなります。各ｂｉｔの重み付けは表２．５のとおりです。

	D7	D6	D5	D4	D3	D2	D1	D0
$80～$8E	Sustain Level				Release Rate			

表２．５　ＳＬ各ｂｉｔの重み付け

	D7	D6	D5	D4
減衰量(dB)	24	12	6	3

※　Ｄ７～Ｄ４が全て“１”（１５）の場合、９３ｄＢとなります。

□ ＳＲ（Ｓｕｓｔａｉｎ　Ｒａｔｅ）：＄７０～＄７Ｅ

サスティーンレートとは、サスティーンレベルから減衰する速さの事で、ＳＲはこのレートを決めるパラメータです。５ｂｉｔ；３２ステップで設定し、ＳＲが大きいほど減衰は速くなります。また、“０”にするとサスティーンレベルが持続されます。

	D7	D6	D5	D4	D3	D2	D1	D0
$70～$7E	／	／	／	Sustain Rate				

□ ＲＲ（Ｒｅｌｅａｓｅ　Ｒａｔｅ）：＄８０～＄８Ｅ

リリースレートとは、Ｋｅｙ－ｏｆｆ後の減衰の速さの事で、ＲＲはこのレートを決めるパラメータです。４ｂｉｔ；１６ステップで設定し、ＲＲが大きいほど減衰は速くなります。

以上のＡ．Ｄ．Ｓ．Ｒ．パラメータの設定により、オペレータのエンベロープが作られます。しかし、これだけではオペレータの出力周波数とは無関係に、常に同じレートのエンベロープがオペレータに与えられることになり、音色によっては不自然な場合があります。
そこで、Ｆ－Ｎｕｍｂｅｒ／Ｂｌｏｃｋデータを利用する事によって、各レートが音程に応じた変化するようにしました。この機能をキースケーリングといいます。

□ ＫＳ（Ｋｅｙ－Ｓｃａｌｅ）：＄５０～＄５Ｅ

キースケールはエンベロープの時間を音程によって変化させる機能です。つまり、高音になるほど各レートは短くなります。２ｂｉｔ；４段階の設定が出来、０では効果は掛からず、３の時に（時間の）差が最も大きくなります。
表２．６にＫＳによるレートのキースケーリング値を示します。

エンベロープジェネレータの各レートは、A．D．S．R．パラメータの設定データとキースケーリング値によって最終的に決定されます。この値は次式により表されます。

Ｒａｔｅ＝2R+Ｒｋｓ　　；R＝0の場合、Ｒａｔｅ＝0

- RはA．D．S．R．の各パラメータの設定値。
  但し、RR（Ｒｅｌｅａｓｅ　Ｒａｔｅ）については
  （設定値×2＋1）をRとする。

- Ｒｋｓはキースケーリング値。

※Ｒａｔｅは６３を最大とし、計算結果が６３より大きな値の時は
全てＲａｔｅ＝６３とする。

表2．6　ＲａｔｅのＫｅｙ－Ｓｃａｌｉｎｇ

Block	0				1				2				3			
Note	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
0	0								1							
1	0				1				2				3			
2	0		1		2		3		4		5		6		7	
3	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Block	4				5				6				7			
Note	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
0	2								3							
1	4				5				6				7			
2	8		9		10		11		12		13		14		15	
3	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

2－4－2. ＳＳＧ－ｔｙｐｅ　Ｅｎｖｅｌｏｐｅ　Ｃｏｎｔｒｏｌ

エンベロープジェネレータのエンベロープは、あらかじめプリセットされたＳＳＧ－ｔｙｐｅエンベロープによってコントロールする事ができます。ＳＳＧ－ｔｙｐｅ　ＥｎｖｅｌｏｐｅはＳＳＧ音源に見られるエンベロープと同一の波形で構成されており、ＥＧ用パラメータの設定だけでは得られないエンベロープの変化をつけることが可能です。エンベロープの形状を図２．４に示します。

このエンベロープを使用する場合、ＥＧ用の各パラメータは下記のように対応します。

① ＡＲのデータ設定は“＄１Ｆ”に固定します。

② Ｋｅｙ－ｏｎ状態でのエンベロープの変化は、ＤＲ，ＳＲの各Ｒａｔｅ及びＳＬによるレベル設定で決まります。

③ ＲＲは通常のモードと同様に働き、Ｋｅｙ－ｏｆｆ後の減衰時間の設定となります。

□ＳＳＧ－ＥＧ：＄９０～＄９Ｅ

	D7	D6	D5	D4	D3	D2	D1	D0
$90～$9E	／	／	／	／	SSG-EG			

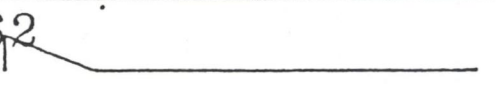
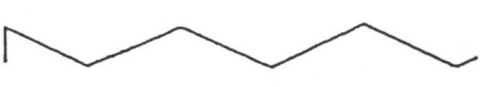
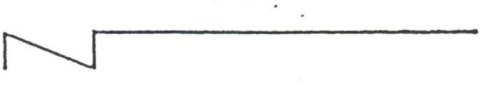
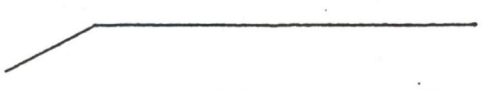
No	D3	D2	D1	D0	エンベロープ形状
0	1	0	0	0	
1	1	0	1	0	
2	1	0	1	0	
3	1	0	1	1	
4	1	1	0	0	
5	1	1	0	1	
6	1	1	1	0	
7	1	1	1	1	

図2．4　ＳＳＧ－ｔｙｐｅエンベロープ

２－４－３．出力制御回路

エンベロープジェネレータで作られたエンベロープは、出力制御回路でレベルを設定します。これによりオペレータの出力レベルが決まり、そのダイナミックレンジは９６ｄＢ、設定可能な分解能は０．７５ｄＢです。

ここで注意する点として、出力レベルは減衰量で表されているということです。つまり、オペレータの出力の最大値を０ｄＢとした時の減衰量を設定することになります。

□ＴＬ（Ｔｏｔａｌ　Ｌｅｖｅｌ）：＄４０～＄４Ｅ

出力レベルの設定はトータルレベルで行います。各ｂｉｔの重み付けは減衰量を示します。したがって、“00”で０ｄＢ（レベルｍａｘ）、“7F”で減衰量９６ｄＢ（レベルｍｉｎ）となります。

但し、ＹＭ２６１２の場合、内蔵のＤ／Ａコンバークが９ｂｉｔであるため、実際のアナログ音声出力は、５４ｄＢ相当になります。従って、ＯＰＮと比較すると、全く同様なパラメータに設定した場合でも、アナログ音声出力がオーバーフローあるいはアンダーフローしてしまう場合があります。

	D7	D6	D5	D4	D3	D2	D1	D0
$40～$4E	／	Total Level						

表２．８　ＴＬ各ｂｉｔの重み付け

	D6	D5	D4	D3	D2	D1	D0
減衰量(dB)	48	24	12	6	3	1.5	0.75

2－5. LFO：Low Frequency Oscillator

ＬＦＯは、内蔵の低周波発振器の出力でオペレータを変調して、音に周期的な変化を与える機能です。ＹＭ２６１２のＬＦＯ波形はサイン波で、５種類のパラメータによって変調の制御をします。

□LFO FREQ. ：$22

ＬＦＯのｏｎ／ｏｆｆ制御とＬＦＯのスピードを決める発振周波数の設定をします。

	D7	D6	D5	D4	D3	D2	D1	D0
$22	／	／	／	／	ON	FREQ.CONT		

Ｄ３：“１”の時、ＬＦＯ　ｏｎ。
Ｄ２～Ｄ０：発振周波数の設定。

FREQ.CONT	0	1	2	3	4	5	6	7
freq(Hz)	3.98	5.56	6.02	6.37	6.88	9.63	48.1	72.2

□PMS（Phase Modulation Sensitivity）：$B4～$B6

Ｆ－Ｎｕｍｂｅｒ／Ｂｌｏｃｋで設定した周波数（位相）情報にＬＦＯを加える（変調する）ことにより、音程に周期的な変化が得られます。ＰＭＳは変調の深さ，位相変調度をチャンネル単位に設定するパラメータです。

□AMS（Amplitude Modulation Sensitivity）：$B4～B6

Ｔｏｔａｌ　ＬｅｖｅｌにＬＦＯを加えることにより、オペレータの出力レベルを周期的に変化させることができます。ＡＭＳは変調の深さ，振幅変調度をチャンネル単位に設定するパラメータです。

ＬＦＯによる振幅変調時の音への効果は、オペレータの役割によって違ってきます。即ち、キャリアが変調された時は音量の変化となり、モジュレータでは音色が変化します。

	D7	D6	D5	D4	D3	D2	D1	D0
$B4～$B6			AMS		／	PMS		

PMS	0	1	2	3	4	5	6	7
変調度(cent)	0	3.4	6.7	10	14	20	40	80

AMS	0	1	2	3
変調度(dB)	0	1.4	5.9	11.8

□AMON：＄６０～＄６Ｅ

振幅変調のｏｎ／ｏｆｆを各スロット毎に行うためのスイッチです。“１”の時ｏｎ。

	D7	D6	D5	D4	D3	D2	D1	D0
$60～$6E	AMON	／	／	Decay Rate				

以上のパラメータでＬＦＯの設定を行います。

内蔵ＬＦＯでは充分な効果が得られない場合、ソフトウエアＬＦＯ（鋸歯状波、矩形波、三角波、Ｓ／Ｈ等）が必要です。これは内蔵のタイマーを利用した割り込み処理で、ＬＦＯ波形に応じたデータをオペレータの各パラメータに与えてやればよいでしょう。

図２．５に、ＬＦＯ機能のブロック図を示します。

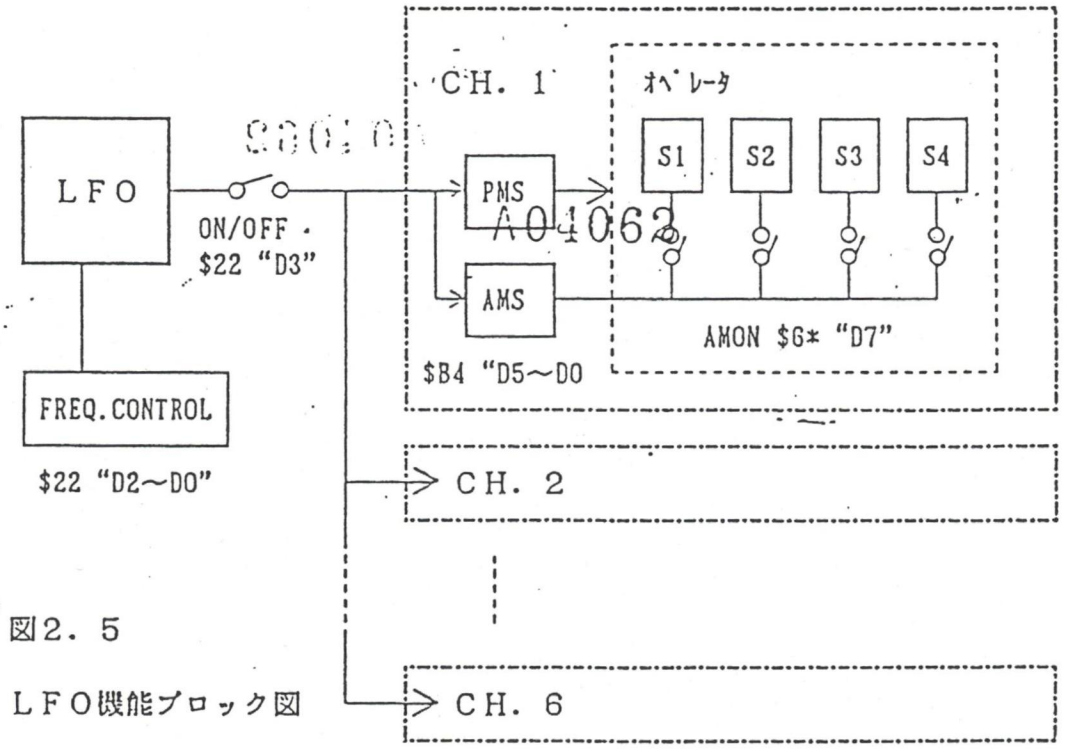


図２．５

ＬＦＯ機能ブロック図

【参　考】　ＬＦＯによる音の周期的な変化は、楽音上、次のような効果が得られます。

- 音程（ピッチ）の変化：ビブラート
- 音量（レベル）の変化：トレモロ
- 音色（トーン）の変化：ワウワウ

2－6．アキュムレータ

アキュムレータは、オペレータより送り込まれた各チャンネルの各スロット出力（９ビット）を累積加算し、Ｄ／Ａコンバータへ送り込みます。
従って、音作りを行う上で、特に、注意する必要はありません。

2－7．Ｄ／Ａコンバータ及び出力選択

2－7－1．Ｄ／Ａコンバータ

Ｄ／Ａコンバータは、アキュムレータで計算した結果をアナログ電圧に変換します。
また、ＹＭ２６１２は、アドレス＄２Ａレジスタに書き込まれたデータをＤ／Ａ変換する事が出来ます。これは、アナログ音声を８ビットのデータにコード化し、そのデータを＄２Ａレジスタに順次書き込む事により行います。
この時のサンプリング・レートは、５５．５ｋＨｚです。
→2－1－2．Ｄ／Ａ用レジスタ参照。

2－7－2．出力選択

ＹＭ２６１２は、ＭＯＬ，ＭＯＲという２チャンネルのアナログ出力を有しており、６チャンネル分のＦＭ音、あるいは、５チャンネル分のＦＭ音＋１ＰＣＭ音を、それぞれＭＯＬ，ＭＯＲに振り分ける事が出来ます。

□　Ｌ／Ｒ：＄Ｂ４～＄Ｂ６“Ｄ７，Ｄ６”

“１”で ON となり、そのＣＨに出力します。
但し、“０”と指定した場合でも、設定した音色パラメータ、トータル・レベルにより、若干は、聞こえることがあります。

3．電気的特性

3－1．絶対最大定格

項　目	定格値	単位
端子電圧	－0．3～7．0	V
動作周囲温度	0～70	℃
保存温度	－50～125	℃

3－2．推奨動作条件

項　目	記号	最小	標準	最大	単位
電源電圧	Vcc	4.75	5.0	5.25	V
	GND		0		V

3－3．直流特性　（動作周囲温度 T_a＝0～70℃）

項目		記号	条件	最小	標準	最大	単位
入力Highレベル電圧	全入力	V_{IH}		2.0		Vcc	V
入力Lowレベル電圧	全入力	V_{IL}		-0.3		0.8	V
クロック入力 Highレベル電圧	φM	V_{CH}		2.0		Vcc	V
クロック入力 Lowレベル電圧	φM	V_{CL}		-0.3		0.8	V
入力リーク電流	φM, $\overline{WR}$, $\overline{RD}$, A0, A1	I_L	V_{in}=0～5V	-10		10	μA
スリーステート入力電流 (オフ状態)	D0～D7	I_{TSL}	V_{in}=0～5V	-10		10	μA
出力Highレベル電圧	D0～D7	V_{OH1} V_{OH2}	I_{OH1}=0.4mA I_{OH2}=40μA	2.4 3.3			V V
出力Lowレベル電圧	D0～D7	V_{OL}	I_{OL}=2mA			0.4	V
出力リーク電流 (オフ状態)	$\overline{IRQ}$	I_{OL}	V_{OH}=0～5V	-10		10	μA
アナログ出力電圧	MOR, MOL	V_{OA}	最大音量 R_L=2.2kΩ	1.4	1.5	1.6	V_{PP}
		V_C	R_L=2.2kΩ		3.0		V
電源電流		I_{CC}				110	mA
プルアップ抵抗	$\overline{TEST}$, $\overline{IC}$, $\overline{CS}$	R_{PU}		60		600	kΩ
入力容量	全入力	C_I	f=1MHz			10	pF
出力容量	全出力	C_O				10	pF

A04052

3－4．交流特性　　（動作周囲温度T_a＝0～70℃）

項目		記号	条件	最小	標準	最大	単位
入力ｸﾛｯｸ周波数	φM	fc	(図3－1)		8.0		MHz
入力ｸﾛｯｸﾃﾞｭｰﾃｨ	φM				50		%
入力ｸﾛｯｸ立上り時間	φM	T_{CR}	(図3－1)			50	nS
入力ｸﾛｯｸ立下り時間	φM	T_{CF}	(図3－1)			50	nS
ｱﾄﾞﾚｽｾｯﾄｱｯﾌﾟ時間	A0,A1	T_{AS}	(図3－2，3)	10			nS
ｱﾄﾞﾚｽﾎｰﾙﾄﾞ時間	A0,A1	T_{AH}	(図3－2，3)	10			nS
ﾁｯﾌﾟｾﾚｸﾄﾗｲﾄ幅	$\overline{CS}$	T_{CSW}	(図3－2)	200			nS
ﾗｲﾄﾊﾟﾙｽ幅	$\overline{WR}$	T_{WW}	(図3－2)	200			nS
ﾗｲﾄﾃﾞｰﾀｾｯﾄｱｯﾌﾟ時間	D0～D7	T_{WDS}	(図3－2)	100			nS
ﾗｲﾄﾃﾞｰﾀﾎｰﾙﾄﾞ時間	D0～D7	T_{WDH}	(図3－2)	20			nS
ﾁｯﾌﾟｾﾚｸﾄﾘｰﾄﾞ幅	$\overline{CS}$	T_{CSR}	(図3－3)	350			nS
ﾘｰﾄﾞﾊﾟﾙｽ幅	$\overline{RD}$	T_{RW}	(図3－3)	350			nS
ﾘｰﾄﾞﾃﾞｰﾀｱｸｾｽ時間	D0～D7	T_{ACC}	(図3-3) C_L=100pF			250	nS
ﾘｰﾄﾞﾃﾞｰﾀﾎｰﾙﾄﾞ時間	D0～D7	T_{RDH}	(図3－3)	10			nS

A04062

※リセット

項目		記号	条件	最小	標準	最大	単位
ﾘｾｯﾄﾊﾟﾙｽ幅	$\overline{IC}$	T_{ICW}	(図3－4)		192		ｻｲｸﾙ

3－5．DAC特性

項目		記号	条件	最小	標準	最大	単位
最大出力振幅	MOR,MOL	V_{OUT}			2/5Vcc		V
分解能	MOR,MOL				9		BIT

タイミング図の設定は、ＶＨ＝２．０Ｖ、ＶＬ＝０．８Ｖを基準とする。

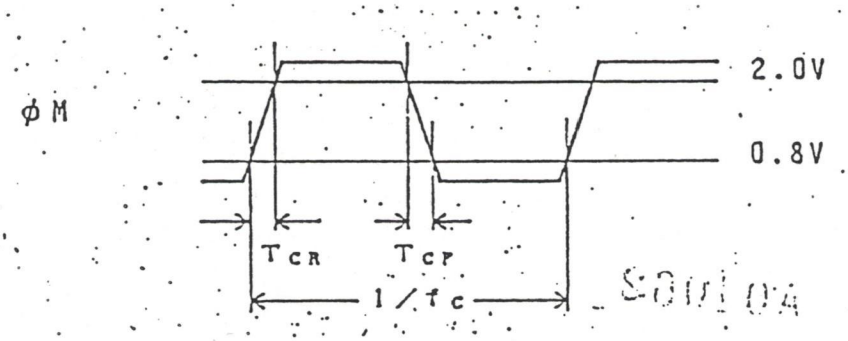
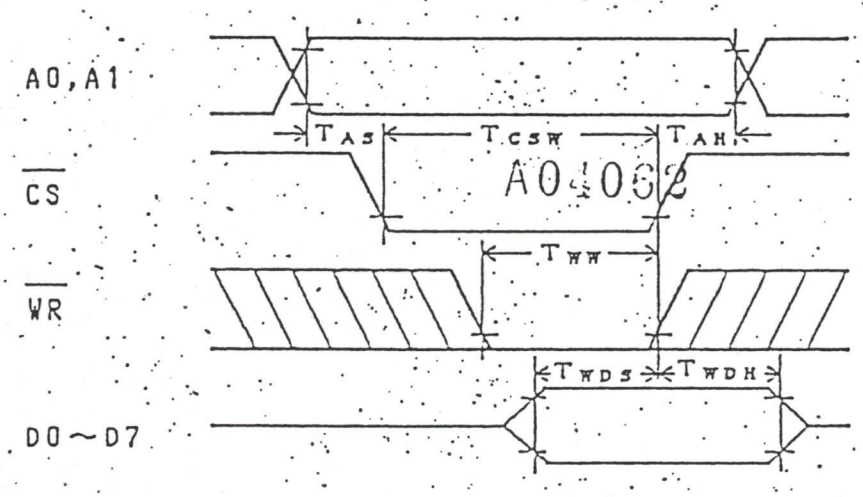


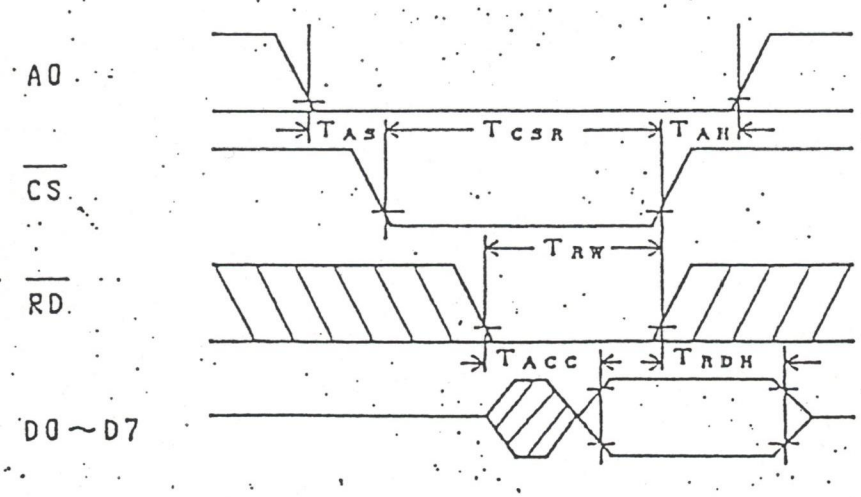
図３－１　　クロック・タイミング



（注）

TCSW，TWW，TWDS，TWDHは，$\overline{CS}$，$\overline{WR}$の何れかがＨＩＧＨレベルになる時を基準とする。

図３－２　　ライト・タイミング



（注）

TACCは，$\overline{CS}$，$\overline{RD}$の何れかが遅くＬＯＷレベルになる時を基準とする。

TCSR，TRW，TRDHは，$\overline{CS}$，$\overline{RD}$の何れかがＨＩＧＨレベルになる時を基準とする。

図３－３　　リード・タイミング

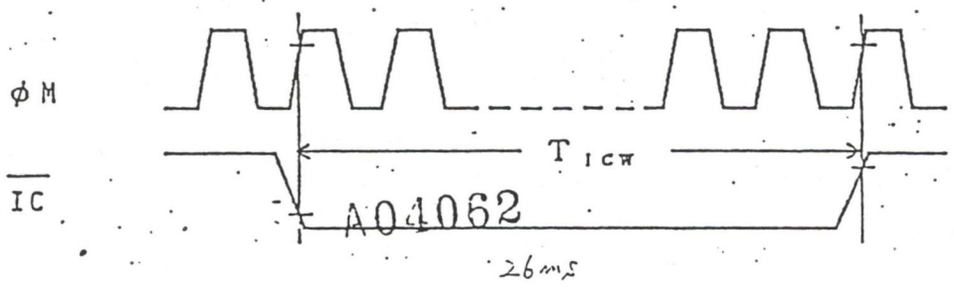


図3－4　　リセット・パルス幅

ＤＸ２１＊ＹＭ２１５１
対応表

ＢＹ　ＮＡＶＹ

FEED BACK / CONNECTION(ALGORISM)

CONNECTION

FEEDBACK	0	1	2	3	4	5	6	7
0	00	01	02	03	04	05	06	07
1	08	09	0A	0B	0C	0D	0E	0F
2	10	11	12	13	14	15	16	17
3	18	19	1A	1B	1C	1D	1E	1F
4	20	21	22	23	24	25	26	27
5	28	29	2A	2B	2C	2D	2E	2F
6	30	31	32	33	34	35	36	37
7	38	39	3A	3B	3C	3D	3E	3F

CMNOKY.ON

CONEECTION

ＤＸ２１	ＹＭ２１５１
ＯＰ１	4
ＯＰ２	3
ＯＰ３	2
ＯＰ４	1

MEGA DRIVE SOUND
開発マニュアル

A04062

目次

Ⅰ　Ｚ８０からみたメモリ構成
　①　Ｚ８０メモリマップ
　②　インタラプト

Ⅱ　６８ＫからＺ８０の制御
　①　Ｚ８０の起動
　②　Ｚ８０エリアへのデータの読み書き

Ⅲ　ＦＭ音源の制御
　①　６８ＫからＦＭチップへのアクセス
　②　Ｚ８０からＦＭチップへのアクセス

Ⅳ　ＰＳＧの制御
　①　６８ＫからＰＳＧへのアクセス
　②　Ｚ８０からＰＳＧへのアクセス

Ⅴ　Ｄ／Ａの制御

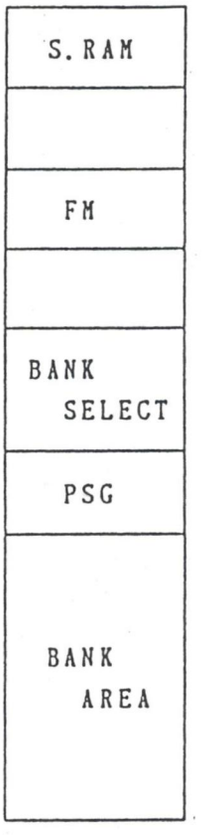
A04062

このマニュアルは、メガ・ドライブのサウンド開発のためのものです。特にメモリマップ、及び各エリアへのアクセス方法について記しました。ＦＭ音源、ＰＳＧの特徴については、別途マニュアルを用意しましたので、そちらを参考にして下さい。

Ｉ．　Ｚ８０からみたメモリ構成

①　Ｚ８０マップ

メモリマップを右に記す。メモリマップドＩ／Ｏになっているため、Ｉ／Ｏマップの割り当てなし。Ｚ８０は６８Ｋ側からみて、Ａ０００００Ｈ～ＡＯＦＦＦＦＨまでのエリアをカバーする。プログラム、バンク、メモリマップドＩ／Ｏの各エリアに分けて説明する。



１）プログラムエリア

プログラム、データ、ワークＲＡＭ等は０Ｈ～１ＦＦＦＨのＳ－ＲＡＭエリアを使用する。

２）ＢＡＮＫエリアおよびＢＡＮＫ切り替え

８０００Ｈ～ＦＦＦＦＨは、Ｂ.ＡＮＫ切り替えによって６８０００のメモリエリアを８０００Ｈごとにアクセスできる。これによってカートリッジのデータを読みだす事が可能となる。但し、６８Ｋ側のＲＡＭエリア（ＦＦ０００００Ｈ～ＦＦＦＦＦＦ(Ｈ)）はＡ書き込みのみが可能である。Ｚ８０側からＡ０８０９２モリのデータを読みだす場合は、ＢＵＳが競合しない様にハード的にチェックしている。先着優先となっている。ＢＡＮＫの切り替えは、Ｚ８０の６０００Ｈに６８側のアドレスＡ１５～Ａ２３の９ビットをＡ１５から順に１ビットずつ書き込んで行う。

３）メモリマップドＩ／Ｏエリア

・ＦＭ音源エリア

ＦＭ音源エリアは４０００Ｈ～４００３Ｈが割り当てられている。ＦＭ音源は３チャンネルずつの２部構成となっている。

ＦＭ１　レジスタＳＥＬＥＣＴ　４０００Ｈ
（１～３チャンネル）
　　データ　ＳＥＮＤ　　４００１Ｈ

ＦＭ２　レジスタＳＥＬＥＣＴ　４００２Ｈ
（４～６チャンネル）
　　データ　ＳＥＮＤ　　４００３Ｈ

・ＰＳＧエリア
ＰＳＧの制御は７Ｆ１１Ｈにデータを書き込んで行う。

例）Ｚ８０の８０００Ｈを６８ＫのＣ０００００Ｈに対応させる場合

A23								A15
1	1	0	0	0	0	0	0	0
C								0

A04062
S0010A

Ｚ８０側６０００Ｈに、Ａ１５から
０，０，０，０，０，０，０，１，１の順で書き込む。

注意
ＢＡＮＫの切り替えＤＡＴＡは、読みだすことができません。

② インタラプト
インタラプトは、Ｖインタラプトのみかかる。ＦＭチップのタイマー割込みはかからない。（ＶインタラプトはビデオのＶ－ＢＬＡＮＫの始まりから１ＬＩＮＥ分、約６４μｓ間の長さで発生する。）

Ⅱ．６８ＫからＺ８０の制御

① Ｚ８０の起動

Ｚ８０のＢＵＳ　ＲＥＱ要求、解除、Ｚ８０のリセット、リセット解除を６８Ｋ側から制御する。

(1) ＢＵＳ　ＲＥＱをかける。
(2) ＢＵＳ　ＲＥＳＥＴを解除する。
(3) Ｚ８０のＳ－ＲＡＭにプログラムを転送する。
(4) ＢＵＳ　ＲＥＳＥＴをかける。
(5) ＢＵＳ　ＲＥＱを解除する。
(6) ＢＵＳ　ＲＥＳＥＴを解除する。

ＢＵＳ　ＲＥＱＵＥＳＴ

・ＢＵＳ　ＲＥＱをかける場合
Ａ||１００Ｈに１００Ｈ（Ｗｏｒｄ）をかくとＢＵＳ　ＲＥＱがかかり、Ｚ８０は停止する。

・ＢＵＳ　ＲＥＱを解除する場合
Ａ||１００Ｈに０Ｈ（Ｗｏｒｄ）をかくと、解除される。

Ｚ８０リセット

・リセットする場合
Ａ||２００Ｈに０Ｈ（Ｗｏｒｄ）をかくと、Ｚ８０にリセットがかかり、プログラムカウンタ、レジスタ等クリアされる。
Ｚ８０は停止状態となる。

・リセットを解除する場合
Ａ||２００Ｈに１００Ｈ（Ｗｏｒｄ）をかくとリセットが解除され、Ｚ８０が動き出す。
（注）ＦＭ音源もリセットされる。　26msのリセット状態が必要

ＢＵＳ専有状態の確認
Ａ11００Ｈを読み出すことで確認できる。Ｂｙｔｅアクセスの場合、Ｂｉｔ０が１ならばＺ８０が専有している。０ならばＢＵＳ　ＲＥＱが受理され、６８Ｋがアクセス可能である。

②　Ｚ８０エリアへの読み書き

Ｚ８０エリア（Ａ０００００Ｈ～Ａ０７ＦＦＦＨ）のデータを読み書きする際、必ずＢＵＳ．ＲＥＱをかけてから行う。

Ｚ８０のデータＢＵＳは８Ｂｉｔであるため、バイトアクセスでしか正しいデータを読み書きできない。

Ⅲ　ＦＭ音源の制御

①　６８ＫからＦＭチップへのアクセス

ＦＭ音源メモリは、Ｚ８０側のエリアに存在する。そのため、ＢＵＳＲＥＱをかけてからＡ０４０００Ｈ～Ａ０４０００３Ｈをアクセスする。

A0402

②　Ｚ８０からＦＭチップへのアクセス

Ｚ８０の４０００Ｈ～４００３Ｈのメモリエリアをアクセスする。

Ⅳ　ＰＳＧの制御

①　６８ＫからＰＳＧのアクセス

ＶＤＰ制御エリアのＣ０００１１Ｈに書き込む。（Ｚ８０空間とは関係なくアクセス可能）

②　Ｚ８０からＰＳＧへのアクセス

ＶＤＰ制御エリアの７Ｆ１１Ｈに書き込む。

Ⅴ　　Ｄ／Ａの制御

ＦＭのＤ／Ａコンバータ（ＦＭのレジスタ　２ＡＨ）を使用するため、チャンネル６は使用不可となる。

次の手順を踏む。（Ｚ８０を想定している。）

① ＦＭレジスタ２ＢＨをＳＥＬＥＣＴし、データ８０Ｈを書き込み、チャンネル６をＤ／Ａモードにする。
② ＦＭレジスタ２ＡＨをＳＥＬＥＣＴし、Ｄ／Ａデータを書き込む。
③ ②をデータ数分繰り返す。
④ ＦＭレジスタ２ＢＨをＳＥＬＥＣＴし、データ０Ｈを書き込み、チャンネル６をＦＭモードに切り替える。

A01062

S80L0A

1989.10.4

ＳＥＧＡ　ＳＯＵＮＤ　開発マニュアル

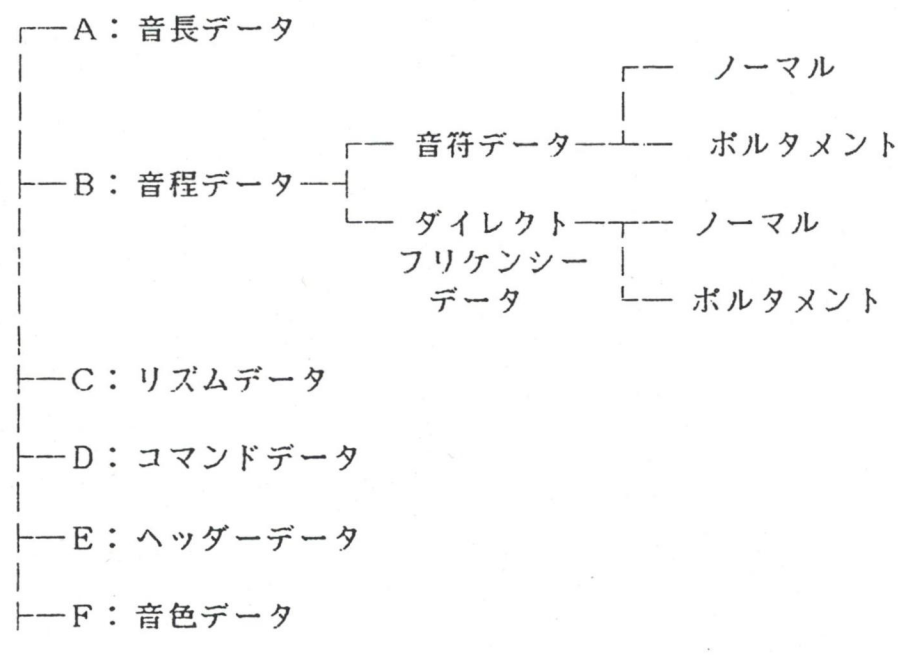
ＳＥＧＡエンタープライゼス
第二研究開発部
上保　徳彦

１：データフォーマット概略
　　Ａ：音長データ
　　Ｂ：音程データ
　　Ｃ：リズムデータ
　　Ｄ：コマンドデータ
　　Ｅ：ヘッダーデータ
　　Ｆ：音色データ
　　Ｇ：ビブラート、エンベロープデータ

２：データフォーマット詳細
　　Ａ：音長データ
　　Ｂ：音程データ
　　Ｃ：リズムデータ
　　Ｄ：コマンドデータ
　　Ｅ：ヘッダーデータ
　　Ｆ：音色データ
　　Ｇ：ビブラート、エンベロープデータ

１：データフォーマット

データは次のように分別されます。



└──G：ビブラート、エンベロープデータ

以下に各データ形式の書式を簡略に記します。詳細は２章を参照してください。

1－A：音長データ

直値、またはアサインを用います。例えば

４分音符	Ｌ４
符点４分音符	ＬＦ４
３連符	Ｌ４＿３

1－B：音程データ

①音符ＮＯＲＭＡＬモード

書式： 音符データ，音長データ，音符データ，音長データ

音符データは、アサインで、ＣＯからＢＳ７まで８オクターブあります。

例： DB CO,L4,EO,LF8

前のデータと音程または音長が同じ時、省略することができます。

例： DB CO,L8,DO,L8,EO,L4 → DB CO,L8,DO,EO,L4

例： DB CO,L8,CO,L4,EO,L4 → DB CO,L8,L4,EO

ただし、休符のときは音長のみ省略できます。

例： DB NL,L1,NL,L1,NL,L8 → DB NL,L1,NL,NL,L8

②音符ポルタメントモード

書式： 音符データ，符号付変化量，音長データ

音程、音長共に、省略できません。

例： DB CO,0F0H,L8

③ダイレクトＦＲＥＱ ＮＯＲＭＡＬモード

書式： ＦＲＥＱデータ ＨＩＧＨ，ＬＯＷ，音長

音程、音長共に、省略できません。

例： DB 0,0F0H,L8

④ダイレクトＦＲＥＱ ポルタメントモード

書式： ＦＲＥＱデータ ＨＩＧＨ，ＬＯＷ，変化量，音長データ

音程、音長共に、省略できません。

例： DB 0,0F0H,020H,L8

1－C：リズムデータ

次のようなアサインをもっています。ＦＭとＰＳＧを同時に鳴らすことができます。

＊FM

S	Snare
HT	Hi Tom
MT	Mid Tom
LT	Low Tom
B	Bass Drum
H	HiHat

＊PSG

PH	PSG HiHat
PO	PSG Open HiHat

＊FM+PSG

OS	PSG Open + FM Snare
HS	PSG HiHat + FM Snare
HB	PSG HiHat + FM Bass Dr.
OB	PSG Open + FM BassDr
POH	PSG Open + FM HiHat

これらの組合せでリズムデータを表します。

書式： リズムデータ，音長データ
（音符データと同様に省略が可）
例： DB BH,L8,H,SBH,H

1－D：コマンドデータ

LRPAN　　L／R設定 (FM)

書式： LRPAN,<L/R DATA>
例： DB LRPAN,LSET

```
cf. LRSET  equ  C0h
    LSET   equ  80h
    RSET   equ  40h
```

FDT　　ディチューン (FM/PSG)

書式： FDT,<SIGN DATA>
例： DB FDT,1

SET_TFLG　　タイミングフラグＳＥＴ (FM/PSG)

書式： SET_TFLG,<DATA>
例： DB SET_TFLG,80H

CMTREND　　トータルリリースＯＦＦ (FM)

書式： CMTREND
例： DB CMTREND

```
ＡＵＴＯＰＡＮ          ＡＵＴＯＰＡＮ設定 （FM）
          書式： AUTOPAN,<PAN NO.>,<PAN TABLE NO.>,
                 <POINT START NO.>,<POINT LIMIT>,<LENGTH DATA>
            例：  DB      AUTOPAN,1,2,1,3,3

ＰＦＶＡＤＤ            ボリュームＡＤＤ （FM/PSG）
          書式： PFVADD,<PSG SIGN DATA>,<FM SIGN DATA>
            例：  DB      PFVADD,01H,07H

ＣＭＶＡＤＤ            ＦＭボリュームＡＤＤ （FM）
          書式： CMVADD,<SIGN DATA>
            例：  DB      CMVADD,-1

ＣＭＴＡＢ              タブ（前後の音をつなぐ） （FM/PSG）
          書式： CMTAB
            例：  DB      CMTAB

ＴＩＥ                  タイ （FM/PSG）
          書式： TIE
            例：  DB      TIE

ＳＲＥ                  スラー （FM/PSG）
          書式： SRE
            例：  DB      SRE

ＣＭＧＡＴＥ            ゲートタイム設定 （FM/PSG）
          書式： CMGATE,<DATA>
            例：  DB      CMGATE,8

ＬＦＯ                  ＬＦＯ設定 （FM）
          書式： LFO,<LFO DATA>,<PMS,AMS DATA>
            例：  DB      LFO,2,3FH

ＴＩＭＥＲ              タイマー設定 （FM/PSG）
          書式： TIMER,<A DATA(DW)>,<B DATA(DB)>
            例：  DB      TIMER
                  DW      120H
                  DB      0E6H

ＴＩＭＥＲ＿ＡＤＤ      タイマーＡＤＤ （FM/PSG）
          書式： TIMER_ADD,<A DATA(DW)>,<B DATA(DB)>
            例：  DB      TIMER_ADD
                  DW      2
                  DB      0

ＰＶＡＤＤ              ＰＳＧボリュームＡＤＤ（PSG）
          書式： PVADD,<SIGN DATA>
            例：  DB      PVADD,-2
```

```
REGSET          ＦＭレジスター設定（Channel Looking）（FM）
        書式：  REGSET,<REGISTER NO.>,<DATA>
          例：   DB      REGSET,040H,8

FMWRITE         ＦＭレジスター設定（Channel Not Look）（FM/PSG）
        書式：  FMWRITE,<REGSTER NO.>,<DATA>
          例：   DB      FMWRITE,22H,8

FEV             ＦＭ音色設定 （FM）
        書式：  FEV,<NO.(0-7Fh)>
        書式：  FEV,<NO.(80-FFh)>,<SONG NO.>
          例：   DB      FEV,0
          例：   DB      FEV,80H,81H

FVR             ＡＵＴＯビブラート設定 （FM/PSG）
        書式：  FVR,<DELAY>,<COUNTER>,<ADD DATA>,<LIMIT>
          例：   DB      FVR,5,1,2,4

PFVR            テーブルビブラート設定 （FM/PSG）
        書式：  PFVR,<PSG DATA>,<FM DATA>
          例：   DB      PFVR,2,8

CMEND           データテーブルＥＮＤ （FM/PSG）
        書式：  CMEND
          例：   DB      CMEND

CMNOIS          ＰＳＧノイズモード設定 （PSG）
        書式：  CMNOIS,<DATA>    ( 0E0H - 0E7H )
          例：   DB      CMNOIS,NOIS7    cf. NOIS7 equ 0E7H

VR              テーブルビブラート設定 （FM/PSG）
        書式：  VR,<TABLE NO.>
          例：   DB      VR,2

EV              エンベロープの設定 （PSG）
        書式：  EV,<TABLE NO.>
          例：   DB      EV,5

CMJUMP          テーブルジャンプ （FM/PSG）
        書式：  CMJUMP,<ADDRESS(DW)>
          例：   DB      CMJUMP
                 DW      ******

CMREPT          リピート設定 （FM/PSG）
          書式：  CMREPT,<REG>,<REPEAT TIME>,<ADDRESS(DW)>
          例：
           T810:                              ┌─┐ ┌──┐
                 DB    C3,L8                   │ │ │
                 DB    CMREPT,0,2        NEST 0    │
```

```
                DW    T810          ―┘       |
                DB    E3,L16                NEST1
                DB    CMREPT,1,8              |
                DW    T810          ―――――――┘
```

ＣＭＣＡＬＬ　　サブデータＣＡＬＬ　(FM/PSG)

書式：　CMCALL,<ADDRESS (DW)>

例：

```
        DB      CMCALL
        DW      SUB810
        DB      CMEND
SUB810:
        DB      C3,L4,CMRET
```

ＣＭＲＥＴ　　サブデータのリターン　(FM/PSG)

書式：　CMRET

例：　DB　　CMRET

ＣＭＢＡＳＥ　　テンポベース設定　(FM/PSG)

書式：　CMBASE,<ABS DATA>

例：　DB　　CMBASE,2

ＣＭＢＩＡＳ　　ＫＥＹバイアス設定　(FM/PSG)

書式：　CMBIAS,<SIGN DATA>

例：　DB　　CMBIAS,-4

ＣＭＰＯＲＴ　　ポルタメントモードＯＮ／ＯＦＦ　(FM/PSG)

書式：　CMPORT,<ON/OFF>

例：　DB　　CMPORT,ON

ＣＭＦＲＥＱ　　ダイレクトフリケンシーモードＯＮ／ＯＦＦ (FM/PSG)

書式：　CMFREQ,<ON/OFF>

例：　DB　　CMFREQ,ON

ＤＴ　　ＦＭチップのＳＥモードの設定　(FM)
及びディチューンＦＲＥＱの設定

書式：　DT,<SLOT1 DT(0-3)>,<SLOT2 DT(0-3)>,
<SLOT3 DT(0-3)>,<SLOT4 DT(0-3)>

例：　DB　　DT,0,1,0,3

ＤＴＤＲ　　ＤＴと同様

ＥＸＣＯＭ　　ＥＸＴＥＮＤ　ＣＯＭＭＡＮＤ (FM/PSG)

書式：　EXCOM,<EXTEND COMMAND>

ＣＭＩＮＴ　　インタラプト、タイマー使用の選択　(FM/PSG)

書式：　EXCOM,CMINT,<DATA>

例：　DB　　EXCOM,CMINT,80H

```
                                cf. 0H : int use
                                   40H : timer B use
                                   80H : timer A(song)/B(s.e.) use

TEMPO_CHG       テンポチェンジ (FM/PSG)
          書式: EXCOM,TEMPO_CHG,<DELAYCOUNT(ADD SIGN DATA)>
           例:  DB      EXCOM,TEMPO_CHG,1

    KEYSET          キーフラグSET (FM/PSG)
          書式: EXCOM,KEYSET,<SONG/S.E. NO.>
           例:  DB      EXCOM,KEYSET,81H

S_PSE           ポーズON/OFF (FM/PSG)
          書式: EXCOM,S_PSE,<ON/OFF>
           例:  DB      EXCOM,S_PSE,ON

WRITE_DATA      メモリへのデータ書き込み (FM/PSG)
          書式: EXCOM,WRTTE_DATA,<ADDR(DW)>,<BYTE NO.(DB)>,
                <DATA1>,<DATA2>,<DATA3>,..
           例:  DB      EXCOM,WRITE_DATA
                DW      1C00H
                DB      4,1,0,0,80H

SNG_BASE        曲の全チャンネルのテンポBASEの設定
                                                    (FM/PSG)
          書式: EXCOM,SNG_BASE,<ABS DATA>
           例:  DB      EXCOM,SNG_BASE,3

SSG             SSG TYPEの決定 (FM/PSG)
          書式: EXCOM,SSG,<SLOT1 DATA>,<SLOT2 DATA>,<SLOT3 DATA>
                ,<SLOT4 DATA>
           例:  DB      EXCOM,SSG,8,10,12,0

WOW             ワウの設定 (FM/PSG)
          書式: EXCOM,WOW,<ENVE TABLE NO.(0-0FFh)>,<FMREG BIT MA
                P>
           例:  DB      EXCOM,WOW,1,3
                cf.      bit 3  2  1  0
                             |  |  |  |
                           SL4 | SL2 |
                               SL3   SL1
```

追加事項：　　（株）AICOM　サウンド
　　塩谷　博光

上記のコマンドのうち、次にあげるものはなくなっています。

```
COMMAND:        TIMER_ADD       ;LFO,CMFREQ
```

このコマンドには以下のものが割与えられています。

Ｔ＿ＳＥＴ　　：ＩＮＴモード設定及びタイマＡ・Ｂデータ書込及び
　　：タイマＡ加減算処理

```
T_SET   EQU     0EBH    ; T_SET,<A DATA(DW)>  OR <A DATA(DW)>,<B DATA(DB)>
                        ;                                       Make Shioya
                        ; bit h7,h6, h5,h4(,h3,h2,)<h1,h0,l7---l0>
                        ;     |  |   |  |   |no use |<A DATA>
                        ;      --|    --|
                        ;        |      | 00:no use , 01:V interrupt
                        ;        |      | 10:timer b interrupt
                        ;        |      | 11:timer a b interrupt
                        ;        |      | <A DATA(DW)>,<B DATA(DB)> SET
                        ;        |
                        ;        | 00:Normal set , 01:Subtract set
                        ;        | 10:Add set
```

Ｔ＿ＳＥＴは２バイト分（ＤＷデータ）をデータとして取り込み，
上位１バイトのビット４，５，６，７をコマンドビットとして
各処理を行います。
コマンドはビット４，５はＩｎｔｅｒｒｕｐｔモード、ビット６，７は
タイマＡの加減算処理モードとなっており、各々の処理を行います。
なお，コマンドをＳＥＴしなかった場合，（００００Ｈ～０３ＦＦＨ）
そのまま１０ビットデータとしてタイマＡにＳＥＴされます。

＝＝＝＝　Ｉｎｔｅｒｒｕｐｔモード　＝＝＝＝

Ｔ＿ＳＥＴはＩｎｔｅｒｒｕｐｔモードの設定をＤＷデータの
上位１バイトのビット４，５の状態によって判断し，以下にあげる
処理を行います。（但し，後に述べる加減算処理との併用はできないので
注意して設定して下さい。又，タイマＡ・Ｂモードの時には，
ＤＷデータの他にＤＢデータが必要です。）

Ｔ＿ＳＥＴ，１０００Ｈ～１３ＦＦＨ　：ＣＰＵタイマモード

ＳＥＴされたデータは無視されます。

```
例：    DB      T_SET
        DW      11E0H           （１Ｅ０ＨはＳＥＴされるが，
                                意味を持たない。）
```

Ｔ＿ＳＥＴ，２０００Ｈ～２３ＦＦＨ　：ＦＭ音源のタイマＢモード

ＤＷデータの下位８ビットがＢデータとしてＳＥＴされます。

```
例：     DB      T_SET
         DW      20E6H   ;B DATA(E6H)
```

Ｔ＿ＳＥＴ，３０００００Ｈ～３３ＦＦＦＦＨ　：ＦＭのタイマＡ・Ｂモード
< DW ><DB> < DW ><DB>

上位２バイト（ＤＷデータ）中，下位１０ビットがタイマＡデータ，下位１バイト（ＤＢデータ）がタイマＢデータとして各々ＳＥＴされます。注意することは，タイマＡ・Ｂモードのときは２バイトデータだけでなく，もう１バイトデータが必要だということです。［例参照。］（タイマＡは曲用，タイマＢは効果音用として作用するのできちんと行うこと。）

```
例：     DB      T_SET
         DW      31E0H   ;A DATA
               DB      0E6H    ;B DATA
```

加減算処理はＤＷデータのビット６，７の状態によって判断します。
又，加減算処理はＦＭのタイマＡのみに働きます。
データ量は０００Ｈ～３ＦＦＨまでです。

Ｔ＿ＳＥＴ，４０００Ｈ～４３ＦＦＨ

現在，ＳＥＴされているデータから与えたデータを引きます。

Ｔ＿ＳＥＴ，８０００Ｈ～８３ＦＦＨ

現在，ＳＥＴされているデータに与えたデータを加えます。

以上ですが、各フォーマットは必ず守って下さい。
なお、データ量はタイマＡが０００Ｈ～３ＦＦＨ、タイマＢが００Ｈ～０ＦＦＨまでです。

現バージョンでは使用できないコマンド

Ｔ＿ＢＡＤＤ　：タイマＢの設定及び加減算処理

```
T_BADD  EQU     8       ; EXCOM,T_BADD,<B DATA(DB)>     ; Make Shioya
                        ; bit h7,h6,h5,h4,h3,h2,h1,h0
                        ;     |  |  |<B DATA>
                        ;      --|00:no operation, 01:subtract
                          ;       |10:add         , 11:normal set
```

Ｔ＿ＢＳＥＴ　：タイマＢ設定

```
T_BSET  EQU     9       ; EXCOM,T_BSET,<B DATA(DB)>     ;Make Hikoshi
```

Ｔ＿ＢＡＤＤは１バイト分（ＤＢデータ）をデータとして取り込み、
上位２ビットの状態よって，各モードを選択します。

Ｔ＿ＢＡＤＤ，００Ｈ～３ＦＨ　：Ｎｏ　ｏｐｅｒａｔｉｏｎ

何も行わないで，処理を終わります。

例　：DB　EXCOM,T_BADD,0FH
↑ＳＥＴされない。

Ｔ＿ＢＡＤＤ，４０Ｈ～７ＦＨ　：減算処理

現在，ＳＥＴされているデータから与えたデータを引きます。
（データ量００Ｈ～３ＦＨ）

例　：DB　EXCOM,T_BADD,54H
１４Ｈ分データを引きます。（４０ＨはＡＮＤされ消える。）

Ｔ＿ＢＡＤＤ，８０Ｈ～０ＢＦＨ　：加算処理

現在，ＳＥＴされているデータに与えたデータを加えます。
（データ量００Ｈ～３ＦＨ）

例　：DB　EXCOM,T_BADD,A6H
２６Ｈ分データを足します。（８０ＨはＡＮＤされ消える。）

Ｔ＿ＢＡＤＤ，０Ｃ０Ｈ～０ＦＦＨ：ノーマルＳＥＴ

直接データを書き込みます。

例　：DB　EXCOM,T_BADD,0DEH
０ＤＥＨを直接書き込む。

上記以外で直接データをＳＥＴする時（００Ｈ～０ＦＦＨ）には
Ｔ＿ＢＳＥＴを使用して下さい。

又，使用する際、***CMD.SRCの;をとって下さい。

***** ビブラート，エンベロープテーブル　コマンド. *****

ＴＢＲＥＰＴ　テーブルの先頭に戻る。
書式：　TBREPT

ＴＢＳＴＡＹ　現在の出力値を維持する。
書式：　TBSTAY

ＴＢＢＡＫ　　テーブルの先頭から指定のデータ数のデータに戻って繰り返す。

書式：　TBBAK,<DATA>

ＴＢＥＮＤ　　エンベロープの場合、音を止める。
ビブラートの場合ＴＢＳＴＡＹと同様。

書式：　TBEND

ＴＢＡＤＤ　　ＦＲＥＱ　ＭＵＬＴＩＰＬＥ　ＤＡＴＡ
(VIBRATE ONLY)

書式：　TBADD,<DATA>

１－Ｅ：ヘッダーデータ

曲およびＳ．Ｅ．１つにつきＦＭ，ＰＳＧ用のヘッダーをもつ。
曲またはＳ．Ｅ．がリクエストされると、曲またはＳ．Ｅ．を制御するためにヘッダーはＲＡＭに転送される。

例：　（リクエストＮＯ．８１Ｈ）

```
S81::
        DW      TIMB81                          ; voice top addr.

        DB      6,3,TP81,DL81                   ; FM use ch,PSG,use ch
                                                ; base,delay counter
        DEFW    TAB81D                              ; rythm table poin
ter
        DEFB    FB81D,FA81D                     ; bias,volm

        DEFW    TAB810                          ; FM 1ch table pointer
        DEFB    FB81M,FA81M                     ; bias,volm

        DEFW    TAB811                          ; FM 2ch table pointer
        DEFB    FB81B,FA81B                     ; bias,volm

        DEFW    TAB812                          ; FM 3ch table pointer
        DEFB    FB81K1,FA81K1                   ; bias,volm

        DEFW    TAB813                          ; FM 4ch table pointer
        DEFB    FB81K2,FA81K2                   ; bias,volm

        DEFW    TAB814                          ; FM 5ch table pointer
        DEFB    FB81K3,FA81K3                   ; bias,volm

        DEFW    TAB810P                         ; PSG 1ch table pointer
        DEFB    PB81M,PA81M,PV81M,PE81M         ; bias,volm,vibr,enve

        DEFW    TAB811P                         ; PSG 2ch table pointer
```

```
        DEFB    PB81B,PA81B,PV81B,PE81B         ; bias,volm,vibr,enve

        DEFW    TAB812P                         ; PSG 3ch table pointer
        DEFB    PB81K,PA81K,PV81K,PE81K         ; bias,volm,vibr,enve

TAB810::
        DB      C3,L8,.........



TIMB81::
;----------------< Piano >-------------------
fev8100::
        CNF     4,5                     ; algorithm , feedback
        MD      1,6,3,0,1,0,3,3         ; multiple , detune
        RSAR    1,31,2,20,1,31,2,20     ; key scale , attack rate
        D1R     5,5,5,7                 ; decay rate
        D2R     2,2,2,2                 ; sustain rate
           RRL     15,1,15,6,15,1,15,10    ; release rate , sustain leve

        TL      30,0,30,0               ; total level
;----------------< Bass >-------------------
fev8101::
        CNF     2,7
        MD      4,0,2,0,8,0,4,0
        RSAR    0,31,0,31,0,31,0,31
        D1R     16,15,9,8
        D2R     7,0,0,0
        RRL     15,3,15,0,15,0,15,4
        TL      20h,20h,20H,00H
```

CHANNEL NO.	＊使用するチャンネル数
FLAG	＊モード設定フラグ（音符モード、ボルタメントモード等）
CHANNEL	＊使用するチャンネルデータ
BASE	＊テンポ設定。BASE＊音長が実際の音長となる。
TABLE POINTER	＊テーブルデータのアドレスをさす。
BIAS	＊音符モードではKEYを決める。ダイレクトFREQモードでは音程データにプラスされる。
VOLM	＊VOLMデータ
VIBR	＊ビブラート NO.
ENVE	＊エンベロープ NO.

1－F：音色データ

２５バイトで１音色を構成します。つぎの並びになっています。

```
0 byte      :       bit 0-2 = algorithm   / 3-7 = feedback

 1          : slot 1 bit 0-3 = multiple    / 4-7 = detune
 2          : slot 2 bit 0-3 = multiple    / 4-7 = detune
 3          : slot 3 bit 0-3 = multiple    / 4-7 = detune
 4          : slot 4 bit 0-3 = multiple    / 4-7 = detune

 5          : slot 1 bit 0-4 = attack rate / 6-7 = key scale
 6          : slot 2 bit 0-4 = attack rate / 6-7 = key scale
 7          : slot 3 bit 0-4 = attack rate / 6-7 = key scale
 8          : slot 4 bit 0-4 = attack rate / 6-7 = key scale

 9          : slot 1 bit 0-4 = decay rate  / 7 = am on
10          : slot 2 bit 0-4 = decay rate  / 7 = am on
11          : slot 3 bit 0-4 = decay rate  / 7 = am on
12          : slot 4 bit 0-4 = decay rate  / 7 = am on

13          : slot 1 bit 0-4 = sustain rate
14          : slot 2 bit 0-4 = sustain rate
15          : slot 3 bit 0-4 = sustain rate
16          : slot 4 bit 0-4 = sustain rate

17          : slot 1 bit 0-3 = sustain level / 4-7 rerease rate
18          : slot 2 bit 0-3 = sustain level / 4-7 rerease rate
19          : slot 3 bit 0-3 = sustain level / 4-7 rerease rate
20          : slot 4 bit 0-3 = sustain level / 4-7 rerease rate

21          : slot 1 bit 0-6 = total level / 7 carier flag
22          : slot 1 bit 0-6 = total level / 7 carier flag
23          : slot 1 bit 0-6 = total level / 7 carier flag
24          : slot 1 bit 0-6 = total level / 7 carier flag
```

1－G：ビブラート、エンベロープデータ

ＭＭＴＢ１３．ＳＲＣにエンベロープテーブル、ビブラートテーブルがあります。各テーブルとも符号付きデータとなっています。

ビブラートは、サウンドルーチンが呼び出される毎に、曲またはＳ．Ｅのテーブルデータによって割り出された分周データに対して、テーブルデータがたしあわされます。ＦＭ／ＰＳＧともに有効です。

エンベロープは、サウンドルーチンが呼び出される毎に、ＶＯＬＭデータに対して、テーブルデータがたしあわされます。ＰＳＧのみ有効です。

書式：

```
LAVEL:
        DB      <data>,<data>, .... ,<table command>
```

なお、テーブルコマンドには、つぎのものがあります。

***** ビブラート，エンベロープテーブル　コマンド *****

ＴＢＲＥＰＴ　　テーブルの先頭に戻る。
書式：　TBREPT

ＴＢＳＴＡＹ　　現在の出力値を維持する。
書式：　TBSTAY

ＴＢＢＡＫ　　テーブルの先頭から指定のデータ数のデータに戻って繰り返す。
書式：　TBBAK,<DATA>

ＴＢＥＮＤ　　エンベロープの場合、音を止める。
ビブラートの場合ＴＢＳＴＡＹと同様。
書式：　TBEND

ＴＢＡＤＤ　　ＦＲＥＱ　ＭＵＬＴＩＰＬＥ　ＤＡＴＡ
(VIBRATE ONLY)
書式：　TBADD,<DATA>

２：データフォーマット詳細

２－Ｂ：音程データ

音符モードにおいて、音長、音符、コマンドはつぎのように割り振っています。

音長	１－７ＦＨ
音符	８０Ｈ－ＤＦＨ
コマンド	Ｅ０Ｈ－ＦＦＨ

①音符ノーマルモード

書式： 音符データ，音長データ，音符データ，音長データ

音符データは、ＣＯからＢＳ７まで８オクターブあります。

ＣＯはＰＳＧ周波数データで８５４Ｈ、ＦＭで２８４Ｈとしています。

移調はＢＩＡＳデータにより行うことができます。ＢＩＡＳデータは１が半音にあたります。符号付きデータなので、０ＦＦＨの場合半音低い音がでます。

②音符ポルタメントモード

書式： 音符データ，符号付変化量，音長データ

例： DB C3,-1,6

このモードでは初期音程から始まり、サウンドルーチンがＣＡＬＬされる毎に変化量を前の音程に足し合わせて出力する操作を音長分繰り返します。

テーブルデータが

```
DB    CO,010H,4
```

のとき、実際のデータは

	PSG FREQ	FM FREQ
1:	854H	284H
2:	864H	294H
3:	874H	2A4H
4:	884H	2B4H

となります。

ＰＳＧはデータが大きいほど低い音になるので、この場合音程は下がります。ＦＭの場合その逆で上がります。

③ダイレクトフリケンシーノーマルモード

書式： ＦＲＥＱデータ ＨＩＧＨ，ＬＯＷ，音長

テーブルのＦＲＥＱデータとＢＩＡＳ値を足して音程としてチップにセットします。

ＢＩＡＳが０、テーブルデータが

```
        DB     03H,050H,4
```

のとき、実際の出力データは

```
        1:      350H
        2:      350H
        3:      350H
        4:      350H
```

となります。

④ダイレクトフリケンシーボルタメントモード

書式：　ＦＲＥＱデータ　ＨＩＧＨ，ＬＯＷ，変化量，音長データ

このモードでは、初期音程（テーブルデータ＋ＢＩＡＳ値）から始まり、サウンドルーチンがＣＡＬＬされる毎に変化量を前の音程に足し合わせて出力する操作を音長分繰り返します。

テーブルデータが

```
        DB     2,0F0H,010H,4
```

のとき、実際のデータは

```
                FREQ
        1:      2F0H
        2:      300H
        3:      310H
        4:      320H
```

となります。

２－Ｃ：リズムデータ

ＦＭ及びＰＳＧのデータは　Ｍ５ＦＭＤＲ１３．ＳＲＣ内にあるテーブル（ＦＭは　ＤＲＴＢ，ＰＳＧはＰＳＧ＿ＤＲＴＢ）から曲データにしたがって逐次呼び出されます。

ＤＲＵＭのデータ形式はＳ．Ｅ．とほぼ同様です（ヘッダーが異なる）。

ＤＲＵＭ ビットアサインは次のようになっています。

```
        7 6 5 4 3 2 1 0
        | |     |--------- FM RYTHM NO.
        | |
        | |----------------PSG RYTHM NO.
        |------------------RYTHM FLAG
```

したがって、ＦＭのリズムは１５種（１－１５）、ＰＳＧは７種（１－７

）もつことができます。

テーブルデータが９２Ｈのとき、ＰＳＧ　ＮＯ．は１なのでＰＳＧ＿ＤＲＴＢをスキャンして、Ｐ＿ＨＩのアドレスを得ます。このデータによりＰＳＧのリズムが鳴ります。また、ＦＭ　ＮＯ．は２なのでＤＲＴＢをスキャンして、ＨＴＯＭのアドレスを得ます。このデータによりＦＭのリズムが鳴ります。ただしＦＭの音色はヘッダーに記されているＥＮＶＥ　ＮＯ．によってＤＲＶＴＢをスキャンして得られるアドレスのデータがセットされます。ＨＴＯＭの場合ＮＯ．が０なのでＶＳＮＲの音色がセットされます。（ＤＲＴＢ、ＰＳＧ＿ＤＲＴＢをスキャンする場合、－１した値でスキャンします。ＤＲＶＴＢの場合はそのままの値を用います。）

ＰＳＧ、ＦＭのリズムを追加、変更したい場合は、ＤＲＴＢ、ＤＲＶＴＢ、ＰＳＧ＿ＤＲＴＢ、およびテーブルデータを追加、変更してください。

```
DRTB::
        DW      SNR,HTOM,MTOM,LTOM,BSD,HHT,LTOM2,SNR2,KSDS,HTH
DRVTB::
        DW      VSNR,VBD,VHHT,VSNR2,VSDSTB,VHIHTTB
PSG_DRTB::
        DW      P_HI,P_OHI

;************************
;       HI TOM          *
;************************
HTOM::
        DW      HTOMT
        DB      00,10H,01H,0                    ; bias,volm,vibr,enve
HTOMT::
        DB      LRPAN,LSET,FHT,010,CMEND

VSNR::
        CNF     6,7
        MD      0,6,0,3,0,3,0,3
        RSAR    0,25,0,31,0,31,0,31
        D1R     21,17,17,12
        D2R     16,10,6,9
        RRL     15,4,15,5,15,10,15,8
        TL      0,2,3,0

;<<<<<<<<< psg hihat table >>>>>>>>>
P_HI::
        DW      P_HITB
        DB      00,04h,00H,1                    ; bias,volm,vibr,enve
P_HITB::
        DB      CMNOIS,0E7H
        DB      0C2H,8,CMEND
```

２－Ｄ：コマンドデータ詳細

ＬＲＰＡＮ　　Ｌ／Ｒ設定　（FM）

書式：　LRPAN,<L/R DATA>

例：　DB　　LRPAN,LSET

cf. LRSET　equ　C0h
LSET　equ　80h
RSET　equ　40h

ＦＭチップのＬ／Ｒを設定します。ＣＯＨでＬ／Ｒ、８０ＨでＬ、４０ＨでＲが設定されます。

ＦＤＴ　　ディチューン　（FM/PSG）

書式：　FDT,<SIGN DATA>

例：　DB　　FDT,1

ＮＯＴＥ、またはダイレクトＦＲＥＱＵＥＮＣＹで与えた音程値にＦＤＴコマンドで設定した符号付きデータを足し合わせて、チップにセットします。

ＳＥＴ＿ＴＦＬＧ　　タイミングフラグＳＥＴ　（FM/PSG）

書式：　SET_TFLG,<DATA>

例：　DB　　SET_TFLG,80H

Ｔ＿ＦＬＧ（０Ａ０１Ｃ２２Ｈ）にデータ値をセットします。曲とキャラクタの同期をとるために使用します。

ＣＭＴＲＥＮＤ　　トータルリリースＯＦＦ　（FM）

書式：　CMTREND

例：　DB　　CMTREND

トータルレベル、及びリリースをきってＰＬＡＹを終了します。

ＡＵＴＯＰＡＮ　　ＡＵＴＯＰＡＮ設定　（FM）

書式：　AUTOPAN,<PAN NO.>,<PAN TABLE NO.>,
<POINT START NO.>,<POINT LIMIT>,<LENGTH DATA>

例：　DB　　AUTOPAN,1,2,1,3,3

ＡＵＴＯＰＡＮコマンドを設定すると、＜ＰＲＪ＞ＣＮＴ．ＳＲＣのＰＡＮ設定テーブルにしたがってＬ／Ｒの設定を行います。

ＡＵＴＯＰＡＮには３つのモードがあります。ＰＡＮ　ＮＯ．でを設定します。

モード１：（ＰＡＮ　ＮＯ．＝１）

まず、＜ＰＡＮ　ＴＡＢＬＥ　ＮＯ．＞に従って＜ＰＲＪ＞ＣＮＴ．ＳＲＣのｐａｎ＿ａｄｄｒ＿ｔｂをスキャンします。ＮＯ．が０の場合ｐａｎ＿０＿ｄａｔａのテーブルにしたがってＬ／Ｒの設定が行われます。＜ＰＯＩＮＴ　ＳＴＡＲＴ　ＮＯ．＞によって、テーブルの読みだし開始位置を決定します。０の場合、トップから、１の場合１バイト後からになります。そして、＜ＬＥＮＧＴＨ　ＤＡＴＡ＞分カウントダウンした後、つぎのテーブルデータを読みだしセットします。このモードのカウント単位は発音数です。したがって＜ＬＥＮＧＴＨ　ＤＡＴＡ＞音出力した後、つぎのＬ／Ｒがセットされます。なお、ＴＩＥコマンドでつなげられているデータは、１音とカウントします。テーブルデータをテーブルトップから＜ＰＯＩＮＴ　ＬＩＭＩＴ＞バイト分読み出したところで、テーブルのＴＯＰから再度データを読みだします。このとき、＜ＰＯＩＮＴ　ＳＴＡＲＴ　ＮＯ．＞は無視されます。

次に例を示します。
テーブルデータが

```
DB      AUTOPAN,1,2,1,3,2
DB      C3,L8,D3,E3,F3,G3,A3,B3
```

のとき、＜ＰＡＮ　ＴＡＢＬＥ　ＮＯ．＞は２なので、ＰＡＮ＿２＿ＤＡＴＡのテーブルよりデータを読みだします。

```
pan_2_data::
        db      0c0h
pan_3_data::
        dh      040h
        db      080h
        db      0c0h
```

また、＜ＰＯＩＮＴ　ＳＴＡＲＴ　ＮＯ．＞が１なので、テーブルＴＯＰから１バイト後ろの４０Ｈが設定され、Ｃ３の音がＲから出ます。＜ＬＥＮＧＴＨ　ＤＡＴＡ＞が２なので、Ｄ３の音までの２音がＲから出ます。Ｅ３、Ｆ３の音は、テーブルＴＯＰから２バイトめの８０Ｈが設定され、Ｌから出ます。Ｇ３、Ａ３の音は、テーブルＴＯＰから３バイトめのＣ０Ｈが設定され、Ｌ／Ｒから出ます。ここで、テーブルＴＯＰ＋＜ＰＯＩＮＴ　ＬＩＭＩＴ＞までのデータを設定したので、次のデータの読みだしは、テーブルのＴＯＰに戻り、Ｃ０Ｈが設定されます。

モード２：（ＰＡＮ　ＮＯ．＝２）

モード２とモード１の違いは、

Ａ）　カウント単位がサウンドルーチンのＣＡＬＬ数（インタラプト、またはＦＭタイマーによる）であること

Ｂ）　＜ＰＯＩＮＴ　ＬＩＭＩＴ＞後は、ループせずそのままの出力を保つ

Ｃ）　つぎの音を発音するときにイニシャライズされる

ことです。

次に例を示します。サウンドルーチンは、インタラプトでＣＡＬＬされているものとします。

テーブルデータが

```
DB      AUTOPAN,2,3,0,3,2
DB      C3,L8,D3,E3,F3,G3,A3,B3
```

のとき、＜ＰＡＮ　ＴＡＢＬＥ　ＮＯ．＞は３なのでｐａｎ＿３＿ｄａｔａのテーブルよりデータを読みだします。

また、＜ＰＯＩＮＴ　ＳＴＡＲＴ　ＮＯ．＞が０なので、テーブルＴＯＰの４０Ｈが設定され、Ｃ３の音がＲから出ます。＜ＬＥＮＧＴＨ　ＤＡＴＡ＞が２なので、２インタラプト後つぎのデータの８０Ｈが設定され、Ｌから音が出ます。さらに、２インタラプト後、つぎのデータのＣ０Ｈが設定され、Ｌ／Ｒから音が出ます。ここで、テーブルＴＯＰ＋＜ＰＯＩＮＴ　ＬＩＭＩＴ＞までのデータを設定したので、つぎの音程をセッするまでこの状態が保持されます。

Ｄ３の音程も、Ｃ３の音程同様に処理されます。

モード３：（ＰＡＮ　ＮＯ．＝３）

モード３は、モード２に準じます。モード２とモード３の違いは、＜ＰＯＩＮＴ　ＬＩＭＩＴ＞後は、ループして、テーブルＴＯＰのデータを設定することです。

ＰＦＶＡＤＤ　　ボリュームＡＤＤ　（FM/PSG）

書式：　PFVADD,<PSG SIGN DATA>,<FM SIGN DATA>

例：　DB　　PFVADD,01H,07H

ＰＳＧとＦＭのあるチャンネルが同じテーブルを読みだしている場合に、ＰＳＧとＦＭそれぞれの異なるＶＯＬＭ値の設定をするときに用います。

ＰＳＧ、ＦＭともに現在の設定値に＋／－されます。

ＣＭＶＡＤＤ　　ＦＭボリュームＡＤＤ　（FM）

書式：　CMVADD,<SIGN DATA>

例：　DB　　CMVADD,-1

ＦＭのＶＯＬＭ値の設定をするときに用います。現在の設定値に＋／－されます。ＰＳＧのチャンネルの場合、キャンセルされます。

ＣＭＴＡＢ　　タブ（前後の音をつなぐ）　（FM/PSG）

書式：　CMTAB

例：　DB　　CMTAB

ＴＩＥ　　タイ　（FM/PSG）

書式：　TIE

例：　DB　　TIE

ＳＲＥ　　スラー　（FM/PSG）

書式：　SRE
例：　DB　　SRE

ＣＭＴＡＢ、ＴＩＥ、ＳＲＥとも　ＮＡＭＥが異なるのみで、機は同一です。
これらは、ＫＥＹ　ＯＦＦ、ＫＥＹ　ＯＮをキャンセルします。したがってエンベロープが持続します。

ＣＭＧＡＴＥ　　ゲートタイム設定　（FM/PSG）

書式：　CMGATE,<DATA>
例：　DB　　CMGATE,8

ゲートタイム長分出力した後、ＫＥＹ　ＯＦＦします。ＢＡＳＥ値と掛け合わせたデータが実際のゲートＴＩＭＥ値となります。

ＬＦＯ　　ＬＦＯ設定　（FM）

書式：　LFO,<LFO DATA>,<PMS,AMS DATA>
例：　DB　　LFO,2,3FH

ＦＭ音源のＬＦＯおよび、コマンド実行チャンネルの　ＡＭＳ、ＰＭＳを設定します。

ＴＩＭＥＲ　　タイマー設定　（FM/PSG）

書式：　TIMER,<A DATA(DW)>,<B DATA(DB)>
例：　DB　　TIMER
DW　　120H
DB　　0E6H

ＦＭ音源のタイマーＡ、Ｂの値を設定します。タイマーＡの設定値は、０Ａ０１Ｃ０５Ｈ（ＷＯＲＤ（ＬＯＷ，ＨＩＧＨ））に、タイマーＢは、０Ａ０１Ｃ０７Ｈ（ＢＹＴＥ）にＳＴＯＲＥされます。

ＴＩＭＥＲ＿ＡＤＤ　　タイマーＡＤＤ　（FM/PSG）

書式：　TIMER_ADD,<A DATA(DW)>,<B DATA(DB)>
例：　DB　　TIMER_ADD
DW　　2
DB　　0

設定されているＦＭ音源Ａ、Ｂの値にデータ値を足し合わせたものをタイマー値として設定します。タイマーＡの設定値は、Ａ０１Ｃ０５（ＷＯＲＤ（ＬＯＷ，ＨＩＧＨ））に、タイマーＢは、Ａ０１Ｃ０７Ｈ（ＢＹＴＥ）にＳＴＯＲＥされます。

ＰＶＡＤＤ　　ＰＳＧボリュームＡＤＤ（PSG）

書式：　PVADD,<SIGN DATA>
例：　DB　　PVADD,-2

ＰＳＧのＶＯＬＭ値の設定をするときに用います。現在の設定値に＋／－されます。ＦＭのチャンネルの場合、キャンセルされます。

ＲＥＧＳＥＴ　　ＦＭレジスター設定（Channel Looking）（FM）

書式：　REGSET,<REGISTER NO.>,<DATA>
例：　DB　　REGSET,040H,8

チャンネルナンバーを参照して、ＦＭチップのレジスタＮＯ．を決定します。そのレジスタに対してデータを書き込みます。例では、コマンド設定チャンネルのＳＬＯＴ１のＴＬデータを８にしています。
なお、書き込んだデータは、一過性のものです。ソングでこのコマンドを使用しても、Ｓ．Ｅ．後のリカバーは行われません。

ＦＭＷＲＩＴＥ　　ＦＭレジスター設定（Channel Not Look）（FM/PSG）

書式：　FMWRITE,<REGSTER NO.>,<DATA>
例：　DB　　FMWRITE,22H,8

指定したＦＭレジスタに直接データを書き込みます。例では、ＬＦＯを８に設定しています。
このコマンドで音色を変えることも出来ますが、Ｓ．Ｅ．後のリカバーは行われません。

ＦＥＶ　　ＦＭ音色設定　（FM）

書式：　FEV,<NO.(0-7Fh)>
書式：　FEV,<NO.(80-FFh)>,<SONG NO.>
例：　DB　　FEV,0
例：　DB　　FEV,80H,81H

ＦＭ音源の音色を設定します。曲、Ｓ．Ｅ．のヘッダーＴＯＰで指定したアドレスから２５ＢＹＴＥで１音色となっています。したがって音色ＮＯ．を１とした場合、指定アドレスより２６ＢＹＴＥ目から始まる２５ＢＹＴＥのデータがセットされることになります。
音色ＮＯ．のＢＩＴ７をＯＮにしたＮＯ．を指定すると、他の曲の音色をチップにセットすることができます。例では、リクエストＮＯ．８１Ｈの曲の０番の音色を指定しています。

ＦＶＲ　　ＡＵＴＯビブラート設定　（FM/PSG）

書式：　FVR,<DELAY>,<COUNTER>,<ADD DATA>,<LIMIT>
例：　DB　　FVR,5,1,2,4

このコマンドを設定すると、つぎの処理を行ってビブラートをつけます。

A）　設定された音程を、<DELAY> データー１　カウント分出力します。カウント間隔は、インタラプトまたは、ＦＭタイマーの設定値で決まります。

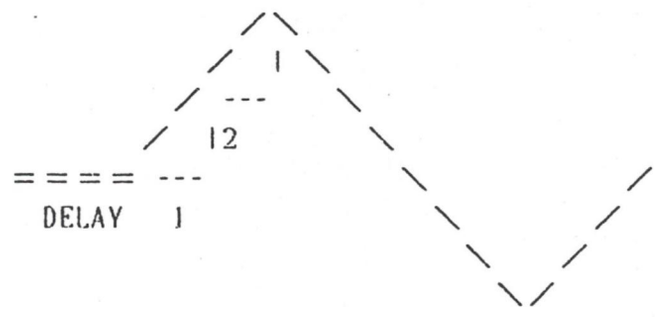
B）　＜ＣＯＵＮＴＥＲ＞値分カウントダウンした後、＜ＡＤＤ　ＤＡＴＡ＞値をいままで設定されている分周データに足し合わせます。＜ＡＤＤ　ＤＡＴＡ＞は符号付きデータです。

C）　＜ＬＩＭＩＴ＞データ／２回　Ｂ）の操作を繰り返した後、＜ＡＤＤ　ＤＡＴＡ＞の符号反転します。

D）　＜ＬＩＭＩＴ＞データ回　Ｂ）の操作を繰り返した後、＜ＡＤＤ　ＤＡＴＡ＞の符号反転します。

E）　　Ｄ）の操作を　ＫＥＹ　ＯＦＦされるまで繰り返します。

例のデータでは、つぎのビブラートがかかります。



このコマンドのキャンセルは、ＶＲコマンドを使用し、ＮＯ．０を設定します。

ＰＦＶＲ　　テーブルビブラート設定　（FM/PSG）

書式：　PFVR,<PSG DATA>,<FM DATA>

例：　DB　　PFVR,2,8

ＰＳＧとＦＭのあるチャンネルが同じテーブルを読みだしている場合に、ＰＳＧとＦＭそれぞれの異なるテーブルビフラートの設定をするときに用います。テーブルビフラートとは、ビブラートテーブルに記された符号付きデータを分周データに足し合わせることでビブラートを付けることを指します。テーブルビブラートは、ＦＶＲコマンドと異なり、分周データの累積はしません。

ビブラートテーブルは、＜ＰＲＪ＞ＴＢ．ＳＲＣのＶＢ１，ＶＢ２，．．．です。

ビブラートＮＯ．２が設定されると、ＶＢ２のテーブルからデータが読み出され、分周値に足されます。例えばＦＭチャンネルを使用し音程データがＣ１のとき、分周値は８２８４Ｈです。したがってビブラート処理後の分周値は、８２８４Ｈ，８２８６Ｈ，８２８８Ｈ，８２８Ａ，８２８ＣＨ，．．．となります。

データ読み出し間隔は、インタラプトまたは、ＦＭタイマーの設定値で決まります。

```
VB2::
        DB      000H,002H,004H,006H,008H,00AH,00CH,00EH
        DB          010H,012H,014H,018H,TBSTAY
```

CMEND　　データテーブルＥＮＤ　(FM/PSG)

書式：　CMEND
例：　DB　　CMEND

曲、Ｓ．Ｅ．データテーブルのＥＮＤを示します。

CMNOIS　　ＰＳＧノイズモード設定　(PSG)

書式：　CMNOIS,<DATA>　　(0E0H - 0E7H)
例：　DB　　CMNOIS,NOIS7　　cf. NOIS7 equ 0E7H

ＰＳＧのＮＯＩＳモードを指定します。
ＰＳＧのノイズは、つぎの種類があります。

同期ノイズ	０：	周波数１３．６５Ｈｚの固定ノイズ	(E0H)
	１：	周波数　６．８３Ｈｚの固定ノイズ	(E1H)
	２：	周波数　３．４１Ｈｚの固定ノイズ	(E2H)
	３：	周波数可変ノイズ	(E3H)
ホワイトノイズ	４：	周波数１３．６５Ｈｚの固定ノイズ	(E4H)
	５：	周波数　６．８３Ｈｚの固定ノイズ	(E5H)
	６：	周波数　３．４１Ｈｚの固定ノイズ	(E6H
	７：	周波数可変ノイズ	(E7H)

ノイズは、トーンのクロックを１６分周しています。したがってノイズ３を使用することで、ＴＯＮＥより低い音程を実現できます。

VR　　テーブルビブラート設定　(FM/PSG)

書式：　VR,<TABLE NO.>
例：　DB　　VR,2

テーブルビブラートのＮＯ．を設定します。ＮＯ．０を設定したときは、ビブラートの処理はキャンセルされます。ＦＶＲコマンドも同様です。
このコマンドは、ＰＳＧ、ＦＭともに有効です。
ＦＶＲコマンドとの共用はできません。

EV　　エンベロープの設定　(PSG)

書式：　EV,<TABLE NO.>
例：　DB　　EV,5

ＰＳＧエンベロープの設定を行います。
エンベロープテーブルに記された符号付きデータをＶＯＬＭデータに足し合わせることでエンベロープを付けています。
エンベロープテーブルは、＜ＰＲＪ＞ＴＢ．ＳＲＣのＥＶ１，ＥＶ２，．．．です。
エンベロープＮＯ．２が設定されると、ＥＶ２のテーブルからデータが読み出され、ＶＯＬＭに足されます。例えばＶＯＬＭ値が２の場合、エンベロープ処理後のアッテネーションデータは、１Ｈ，３Ｈ，５Ｈ，７Ｈ，９Ｈ，１１Ｈとなります。
データ読み出し間隔は、インタラプトまたは、ＦＭタイマーの設定値で決まります。
エンベロープＮＯ．０でエンベロープの処理はキャンセルされます。

```
EV2::
        DB              000H,002H,004H,006H,008H,010H,TBEND
```

ＣＭＪＵＭＰ　　テーブルジャンプ　(FM/PSG)

書式：　CMJUMP,<ADDRESS(DW)>

例：

```
        DB      CMJUMP
        DW      ******
```

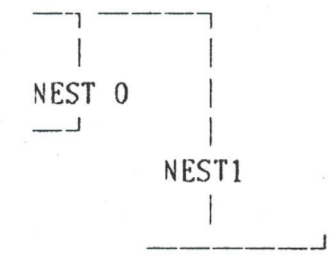
テーブル読み出し位置をＤＷ　で設定したアドレスにＪＵＭＰさせます。

ＣＭＲＥＰＴ　　リピート設定　(FM/PSG)

書式：　CMREPT,<REG>,<REPEAT TIME>,<ADDRESS(DW)>

例：

```
 T810:
        DB      C3,L8
        DB      CMREPT,0,2
        DW      T810
        DB      E3,L16
        DB      CMREPT,1,8
         DW     T810
```



ＣＭＲＥＰＴコマンドからＤＷで指定するアドレスまでを＜ＲＥＰＥＡＴ　ＴＩＭＥ＞回繰り返します。＜ＲＥＧ＞は、ネストの深さを指定します。

ＣＭＣＡＬＬ　　サブデータＣＡＬＬ　(FM/PSG)

書式：　CMCALL,<ADDRESS (DW)>

例：

```
        DB      CMCALL
        DW      SUB810
        DB      CMEND
```

```
SUB810:
        DB      C3,L4,CMRET
```

テーブル読み出し位置をＤＷ　で設定したアドレスにＪＵＭＰさせます。ＣＭＲＥＴコマンドにより、テーブル読み出し位置がコマンドのつぎのデータに戻ります。

ＣＭＲＥＴ　　サブデータのリターン　(FM/PSG)

```
書式：  CMRET
 例：   DB      CMRET
```

テーブル読み出し位置をＣＭＣＡＬＬコマンドのつぎのデータに戻します。

ＣＭＢＡＳＥ　　テンポベース設定　(FM/PSG)

```
書式：  CMBASE,<ABS DATA>
 例：   DB      CMBASE,2
```

チャンネルのテンポＢＡＳＥ値を設定します。データは絶対値です。データのＢＡＳＥ値は、音長データを読み込むときに、音長カウンターを音長データ＊ＢＡＳＥ値に設定します。したがって音長省略時にＣＭＢＡＳＥコマンドを実行しても、音長カウンターは更新されません。

ＣＭＢＩＡＳ　　ＫＥＹバイアス設定　(FM/PSG)

```
書式：  CMBIAS,<SIGN DATA>
 例：   DB      CMBIAS,-4
```

チャンネルのＫＥＹバイアス値を設定します。コマンド実行時に設定されているＫＥＹバイアス値に、ＤＡＴＡ値を足し合わせます。ＮＯＴＥモード時には、＋１で音程を半音あげます。－１で半音下げます。

バイアス値は、音程データを読み込むときに、音程データに足し合わされた後、周波数データが求められます。したがって音程省略時にＣＭＢＩＡＳコマンドを実行しても、周波数は更新されません。

ＣＭＰＯＲＴ　　ポルタメントモードＯＮ／ＯＦＦ　(FM/PSG)

```
書式：  CMPORT,<ON/OFF>
 例：   DB      CMPORT,ON
```

ポルタメントモードの設定、解除を指定します。

ＣＭＦＲＥＱ　　ダイレクトフリケンシーモードＯＮ／ＯＦＦ　(FM/PSG)

```
書式：  CMFREQ,<ON/OFF>
 例：   DB      CMFREQ,ON
```

ダイレクトフリケンシーモードの設定、解除を指定します。

ＤＴ　　ＦＭチップのＳＥモードの設定　(FM)
　　及びディチューンＦＲＥＱの設定

書式：　DT,<SLOT1 DT(0-3)>,<SLOT2 DT(0-3)>,
　　<SLOT3 DT(0-3)>,<SLOT4 DT(0-3)>

例：　DB　　DT,0,1,0,3

ＦＭチップの効果音モード使用を指定します。＜ＰＲＪ＞ＣＮＴ.ＳＲＣにあるｄｔｕｎｅ＿ｔｂのデータをコマンド指定データにより得、音程データによって得た分周データに加えることで、各スロットに設定する分周データを得ます。

例えば、音程データがＣ２のとき、分周値は８２８４Ｈです。例のようなＤＴコマンドの設定の場合、ＳＬＯＴ１のコマンドデータは０なので、ｄｔｕｎｅ＿ｔｂの０番目のデータ、０００Ｈが分周値に足されてＳＬＯＴ１にセットされます。同様にＳＬＯＴ２には１番目のデータ１３２Ｈが、ＳＬＯＴ３には０番目のデータ０００Ｈが、ＳＬＯＴ４には３番目のデータ１Ｅ４Ｈが、分周値に足されてセットされます。

```
dtune_tb::
        DW      000H,0132H,018EH,01E4H
```

ＤＴＤＲ　　ＤＴと同様

ＥＸＣＯＭ　　ＥＸＴＥＮＤ　ＣＯＭＭＡＮＤ　(FM/PSG)

書式：　EXCOM,<EXTEND COMMAND>

拡張用のコマンドです。

ＣＭＩＮＴ　　インタラプト、タイマー使用の選択　(FM/PSG)

書式：　EXCOM,CMINT,<DATA>

例：　DB　　EXCOM,CMINT,80H
　　cf. 0H : int use
　　40H : timer B use
　　80H : timer A(song)/B(s.e.) use

サウンドルーチンの同期方法を、Ａ）インタラプトで行う　Ｂ）ＦＭタイマーＢで行う　Ｃ）ＦＭタイマーＡ，Ｂで行う　の３種類のうちのいずれかを指定します。

データが０の場合、Ａ）インタラプト、４０Ｈの場合、Ｂ）ＦＭタイマーＢ、８０Ｈの場合、Ｃ）ＦＭタイマーＡ，Ｂ　が指定されます。

Ａ）インタラプト同期

サウンドルーチンは、インタラプト毎にＣＡＬＬされ、曲、Ｓ．Ｅ．の処理をします。

Ｂ）ＦＭタイマーＢ同期

サウンドルーチンは、ＴＩＭＥＲＢがオーバーフローする毎にＣＡＬＬされ、曲、Ｓ．Ｅの処理をします。
ＴＩＭＥＲＢの値は、Ｚ８０　ＳＲＡＭの１Ｃ０６Ｈ（ＢＹＴＥ）にセットされています。

Ｃ）ＦＭタイマーＡ、Ｂ同期

サウンドルーチンは、ＴＩＭＥＲＡ、Ｂがオーバーフローを監視し、Ａのオーバーフローで曲の処理を、Ｂのオーバーフローで Ｓ．Ｅ．の処理を行います。
タイマーＡの値は、Ｚ８０　ＳＲＡＭの１Ｃ０４Ｈ（ＷＯＲＤ（ＬＯＷ，ＨＩＧＨ））に、タイマーＢの値１Ｃ０６Ｈ（ＢＹＴＥ）にセットされています。

ＴＥＭＰＯ＿ＣＨＧ　　テンポチェンジ　(FM/PSG)

書式：　EXCOM,TEMPO_CHG,<DELAYCOUNT(ADD SIGN DATA)>
例：　DB　　EXCOM,TEMPO_CHG,1

ＤＥＬＡＹカウンター値を＋／－します。
ＤＥＬＡＹカウンターは、曲の音長を調整するために使用します。カウンター値をＮとすると、音長は　Ｎ／（Ｎ－１）倍されます。
Ｓ．Ｅ．には無関係です。

ＫＥＹＳＥＴ　　キーフラグＳＥＴ　(FM/PSG)

書式：　EXCOM,KEYSET,<SONG/S.E. NO.>
例：　DB　　EXCOM,KEYSET,81H

リクエストＮＯ．にあたる曲、Ｓ．Ｅ．を鳴らします。例ではリクエストＮＯ．８１Ｈの曲が鳴ります。

Ｓ＿ＰＳＥ　　ポーズＯＮ／ＯＦＦ　(FM/PSG)

書式：　EXCOM,S_PSE,<ON/OFF>
例：　DB　　EXCOM,S_PSE,ON

曲に対してＰＡＵＳＥをかけ、曲を一時的に止めます。ＯＦＦすると、ＰＡＵＳＥをかけた位置から曲が始まります。

ＷＲＩＴＥ＿ＤＡＴＡ　　メモリへのデータ書き込み　(FM/PSG)

書式：　EXCOM,WRTTE_DATA,<ADDR(DW)>,<BYTE NO.(DB)>,
<DATA1>,<DATA2>,<DATA3>,..
例：　DB　　EXCOM,WRITE_DATA

```
        DW      1C00H
        DB      4,1,0,0,80H
```

＜ＡＤＤＲ（ＤＷ）＞で指定したアドレスにから＜ＢＹＴＥ　ＮＯ．（ＤＢ）＞ＢＹＴＥ数のデータを書きます。例では１Ｃ００Ｈから１，０，０，８０Ｈの４バイトのデータを書き込んでいます。

ＳＮＧ＿ＢＡＳＥ　　曲の全チャンネルのテンポＢＡＳＥの設定　（FM/PSG）

書式：　EXCOM,SNG_BASE,<ABS DATA>
例：　DB　　EXCOM,SNG_BASE,3

曲全チャンネルのＢＡＳＥ値を設定します。データは絶対値です。

ＳＳＧ　　ＳＳＧ　ＴＹＰＥの決定　（FM/PSG）

書式：　EXCOM,SSG,<SLOT1 DATA>,<SLOT2 DATA>,<SLOT3 DATA>,<SLOT4 DATA>
例：　DB　　EXCOM,SSG,8,10,12,0

ＦＭチップのＳＳＧエンベロープの設定を行います。ＳＬＯＴデータのＢＩＴ４をＯＮにすることに注意してください。

ＷＯＷ　　ワウの設定　（FM/PSG）

書式：　EXCOM,WOW,<ENVE TABLE NO.(0-0FFh)>,<FMREG BIT MAP>
例：　DB　　EXCOM,WOW,1,3
cf.　　bit 3　2　1　0
　　|　|　|　|
　　SL4 | SL2 |
　　SL3　　SL1

ＷＯＷコマンドを設定すると、指定したスロットに対して、音色データで設定したトータルレベルに、エンベロープテーブルから読み出した値を足し合わせてセットする処理を行います。どのＳＬＯＴエンベロープテーブルデータを足しあわせるかは、＜ＦＭＲＥＧ　ＢＩＴ　ＭＡＰ＞で指定します。ＢＩＴがＯＮでそのスロットに対し処理を行います。

エンベロープテーブルデータの読み出しは、ＥＶコマンド指定時と同様に行っています。

***** ビブラート、エンベロープテーブル　コマンド *****

エンベロープ、ビブラートテーブル記述のためのコマンドが４つあります。

ＴＢＲＥＰＴ　　テーブルの先頭に戻る。

書式：　TBREPT

データ読み出し位置をテーブル先頭に移します。

ＴＢＳＴＡＹ　　現在の出力値を維持する。

書式：　TBSTAY

データ読み出しを停止し、現在の出力を保ちます。

ＴＢＢＡＫ　　テーブルの先頭から指定のデータ数のデータに戻って繰り返す。

書式：　TBBAK,<DATA>

データ読み出し位置をテーブル先頭＋<DATA>位置に移します。

ＴＢＥＮＤ　　エンベロープの場合、音を止める。
ビブラートの場合ＴＢＳＴＡＹと同様。

書式：　TBEND

エンベロープテーブルの場合、データ読み出しを停止し出力をＯＦＦします。
ビブラートテーブルの場合ＴＢＳＴＡＹと同様です。

ＴＢＡＤＤ　　ＦＲＥＱ　ＭＵＬＴＩＰＬＥ　ＤＡＴＡ
(VIBRATE ONLY)

書式：　TBADD,<SIGN DATA>

ビブラートテーブルの場合のみ有効です。ビブラートの深さを＋／－します。
このコマンドが設定されるとそれまでのビブラートＡＤＤデータに＜ＳＩＧＮ　ＤＡＴＡ＞が足し合わされ、新しいビブラートＡＤＤデータとなります。分周データに足し合わされるビブラートデータは、ビブラートＡＤＤデータ＊ビブラートテーブルデータとなります。
ビブラートＡＤＤデータは、ＫＥＹ　ＯＮ毎にクリアーされます。

1－Ｆ：音色データ

２５バイトで１音色を構成します。つぎの並びになっています。
キャリアのトータルレベルデータは、ｂｉｔ７をＯＮにします。このことで、ＶＯＬＭデータが足された値がセットされます。

1－G：ビブラート、エンベロープデータ

M5TB13．SRCにエンベロープテーブル、ビブラートテーブルがあります。各テーブルとも符号付きデータとなっています。

エンベロープ、ビブラートとも、ENVTB：、VIBTB：のアドレステーブルによって指定されます。例えばビブラートNO．として1が指定された場合、VIBTB：の1番目に指定されているテーブルが使用さ

1－Ｇ：ビブラート、エンベロープデータ

Ｍ５ＴＢ１３．ＳＲＣにエンベロープテーブル、ビブラートテーブルがあります。各テーブルとも符号付きデータとなっています。

エンベロー

1989.10.4

エディタの使用方法について

ＳＥＧＡエンタープライゼス
第二研究開発部
上保　徳彦

１：起動
２：モード
２－Ａ）ＦＬＡＧ　ＭＯＤＥ
２－Ｂ）ＤＡＴＡ　ＭＯＤＥ
２－Ｃ）ＬＯＤＥＲ　ＭＯＤＥ
２－Ｄ）ＳＥＮＤ　ＭＯＤＥ
２－Ｅ）ＴＥＳＴ　ＭＯＤＥ
３：ｚ８０サウンドコントロールのインストール

１：起動

このエディタは、メガドライブ　サウンドコントロール（ＶＥＲ１．３）用のエディタです。また、別記のＲＯＭ＿ＲＡＭボードの使用を前提としています。ただし、データエディット時、データロード時に、Ｚ８０のＳＲＡＭ領域以外をアクセスしなければ、ＲＯＭボードでも使用可能です。

パワーＯＮ、で起動されます。起動時には、ボードのプロテクトを以前指定したモードにするとともに、サウンドコントロール（ＶＥＲ１３）をＺ８０　ＳＲＡＭに転送します。また、バンクアドレス、データスタートアドレスも以前設定したものを設定します。リセット時には、バンクアドレス、データスタートアドレスのみ設定します（バンクアドレス、データスタートアドレスについては、ＶＥＲ１３のマニュアル参照のこと）。これらの操作のためにＲＯＭ＿ＲＡＭボードのＲＡＭ４つのうち、最後のＲＡＭ８０００Ｈの最後の８バイトを使用しています。

２：モード

ＳＥＬＥＣＴ　ＭＯＤＥ、ＦＬＡＧ　ＭＯＤＥ、ＤＡＴＡ　ＭＯＤＥ、ＬＯＤＥ　ＭＯＤＥ、ＳＥＮＤ　ＭＯＤＥ、ＴＥＳＴ　ＭＯＤＥの６つのモードがあります。

２－Ａ）ＳＥＬＥＣＴ　ＭＯＤＥ

各モードのセレクトを行います。

パワーＯＮ、リセット、他のモードでＳＴＡＲＴボタンを押すことによって、ＳＥＬＥＣＴモードになります。

パッドの上下でカーソル移動、Ｃボタンで決定します。

２－Ｂ）ＦＬＡＧ　ＭＯＤＥ

サウンドコントロールのＷＯＲＫＲＡＭをリアルタイム表示します。

画面は下図のようになっています。
バッド１の上下左右でカーソル移動、Ａボタンで数字のインクリメント、Ｂボタンでデクリメントです。Ｃボタンを押すと画面下から２行めで指定されているリクエストナンバーがセットされ、音がでます。
バッド２の左右で画面がスクロールします。

```
----------------------------------------
        sound editor                    |
                flag mode 0             |
flag  base    bias  enve  stac  lcon  f |
   chan   tb p   vibr  volm  endc lcstr |                  ｱﾄﾞﾚｽ
 80 00 02 3020 f0 01 03 10 30 10 08 00  |   RYTHM FMCH 2  01C40H-
 80 01 02 3020 f0 01 03 10 30 10 08 00  |   SONG  FMCH 0  01C70H-
 80 02 02 3020 f0 01 03 10 30 10 08 00  |   SONG  FMCH 1  01CA0H-
 80 04 02 3020 f0 01 03 10 30 10 08 00  |   SONG  FMCH 4  01CD0H-
 80 05 02 3020 f0 01 03 10 30 10 08 00  |   SONG  FMCH 5  01D00H-
 80 06 02 3020 f0 01 03 10 30 10 08 00  |   SONG  FMCH 6  01D30H-
 00 00 02 3020 f0 01 03 10 30 10 08 00  |   RYTHM WORK    01D60H-
 00 00 02 3020 f0 01 03 10 30 10 08 00  |   PSG CH 0      01D90H-
 00 00 02 3020 f0 01 03 10 30 10 08 00  |   PSG CH 1      01DC0H-
 00 00 02 3020 f0 01 03 10 30 10 08 00  |   PSG CH 2      01DF0H-
 00 00 02 3020 f0 01 03 10 30 10 08 00  |   BACK S.E FCH5 01E20H-
 00 00 02 3020 f0 01 03 10 30 10 08 00  |   BACK S.E FCH6 01E50H-
 00 00 00 0000 00 00 00 00 00 00 00 00  |   S.E CH 2      01E80H-
 00 00 00 0000 00 00 00 00 00 00 00 00  |   S.E CH 5      01EB0H-
 00 00 00 0000 00 00 00 00 00 00 00 00  |   S.E CH 6      01EE0H-
 00 00 00 0000 00 00 00 00 00 00 00 00  |   S.E PSG CH 0  01F10H-
 00 00 00 0000 00 00 00 00 00 00 00 00  |   S.E PSG CH 1  01F40H-
 00 00 00 0000 00 00 00 00 00 00 00 00  |   S.E PSG CH 2  01F70H-
                                        |
 81 81                                  |   kyflag
 00 00 00 00 00 00 00 00 00 00          |   ram 1c00h -
----------------------------------------
```

２－Ｃ）ＤＡＴＡ　ＭＯＤＥ

```
|--------------------------------------|
|  sound editor                        |
|          data&voice mode             |
|                                      |
|HEADDER                               |
|     flag  base    bias  enve         |
|        chan   tb p   vibr  volm      |
|      80 03 02 8002 00 00 03 00       |
| ADDR 8002                            |
| CHAN 00       VOICE NO: 03           |
```

```
| EF 00 C1 08     FBC B4                  |
| C2 06 C3 C5     MD  38 02 34 11         |
| 80 12 EF 01     AR  48 10 44 1B         |
| C1 08 0C 0C     D1R 58 14 54 18         |
| 0E C3 D0 12     D2R 68 20 64 07         |
| C2 10 D2 D2     RLR 78 03 74 18         |
| C3 10 D1 D1     TL  88 02 84 13         |
| D5 D4 D2 10                             |
| D3 D1 08 D2                             |
| F2                                      |
|                                         |
|                                         |
|                                         |
|                                         |
|  81 81                                  |
| RAM TOP   00 00 00 00 00 00 00 00 00    |
|                                         |
-------------------------------------------
```

このモードでは、ＫＥＹ　ＮＯ．に対応するデータの、ある１つのチャンネルのヘッダー、音データ、ＶＯＩＣＥデータを表示します。
音データは左４列、ＶＯＩＣＥデータは右４列で表示しています。
表示チャンネル、ＶＯＩＣＥ　ＮＯ．、データトップアドレスはカーソルを合わせ、Ａ／Ｂボタンで任意に設定可能です。ただしＫＥＹ　ＮＯ．の設定を変えると初期化され、表示チャンネルは0、ＶＯＩＣＥ　ＮＯ．０、データトップアドレスはヘッダーのテーブルポインタ値となります。

イ：ＶＯＩＣＥデータの表示

ＶＯＩＣＥ　ＮＯ．で指定されたＶＯＩＣＥデータを表示します。
各列左からＳＬＯＴ１、ＳＬＯＴ２、ＳＬＯＴ３、ＳＬＯＴ４、となっています。

ウ：ＨＥＡＤＤＥＲデータの表示

ＫＥＹ　ＮＯ．で指定されたＮＯ．のヘッダーのうちＣＨＡＮ　ＮＯ．で指定されたチャンネルのデータを表示します。

２－Ｃ）ＬＯＤＥ　ＭＯＤＥ

ＬＯＡＤＥＲを用いてデータをダウンロードします。データフォーマットは、ＩＮＴＥＬ　ＨＥＸ、モトローラＳ２８、Ｓ３８いずれでも可です。
ＩＮＴＥＬ　ＨＥＸデータの場合、アドレス０－１ＦＦＦＨまでのデータは、Ｚ８０　ＳＲＡＭに、８０００Ｈ以降のデータは、バンク側に送ります。したがってＢＡＮＫアドレスおよび、ボードのプロテクトＯＮ／ＯＦＦに注意してください。とくにボードのバンク使用可能範囲を越えてバンクアドレスを設定してロードすると暴走します。

ボードのバンク使用可能範囲は、ＲＯＭ＿ＲＡＭボードに付いているロータリーＳＷで行います。バンクスタートアドレスは、８０００Ｈ＊ロータリーＳＷ値です。０００００Ｈから７ＦＦＦＦＨのエリアをカバーします。なお、ロータリーＳＷをＦにした場合、２００００００から２１ＦＦＦＦＨをカバーします。

ボードのプロテクトＯＮ／ＯＦＦは、Ａ／Ｂボタンで設定します。

バンクアドレスは、Ａ／Ｂボタンで書換えます。書換えがうまくいかない場合、Ｚ８０側の状態をチェックしてください。

ロードのスタートは、ＳＴＡＲＴ表示にカーソルをあわせ、Ｃボタンを押すことで設定します。
ＲＥＡＤＹ表示が出たのち、ＰＣ側のプリンタポートからデータを送ります。
ＣＯＭＰＬＥＴＥ表示が出たらば終了です。
このとき、Ｃボタンを押せば、再度ロードすることができます。

２－Ｄ）ＳＥＮＤ　ＭＯＤＥ

エディタに組み込まれたサウンドコントロールをＺ８０　ＳＲＡＭに転送します。Ｃボタンで転送を開始します。

２－Ｅ）ＴＥＳＴ　ＭＯＤＥ

ＲＯＭ＿ＲＡＭボードのＲＡＭアドレスが書き変わらないかをチエックするためのものです。ＴＥＳＴモードをＯＮにすると、アドレス１０００２Ｈから４ＢＹＴＥを画面左上に表示します。
ＴＥＳＴモードのＯＮ/ＯＦＦは、Ａ／Ｂボタンで行います。

３：ｚ８０サウンドコントロールのインストール

＜ｄａｔａｅｄ１０＞ｓｎｄｉｎｉｔ．ａｓｍに、ｚ８０　ｓｒａｍに転送するデータおよび転送プログラムがあります。
ｉｎｃｌｕｄｅ　さりている　．ｈｈｈファイル名を書換え、ｅｄ．ｂａｔを起動してアセンブル、リンクを行うことで、ｐｏｗｅｒ　ｏｎ時にｚ８０　ｓｒａｍに転送されるデータを変更することができます。

＜＜＜＜＜＜＜＜＜＜＜＜＜＜＜＞＞＞＞＞＞＞＞＞＞＞＞＞＞

５：バンク設定の方法

Ｚ８０をＧＯするまえに　ＢＡＮＫアドレス、およびデータスタートオフセット値をＺ８０側に与えて下さい。

ＢＡＮＫアドレス

0a01c00h　==>　ＢＩＴ　１６－２３
0a01c01h　==>　ＢＩＴ　８－１５

ＥＸ．　ＢＡＮＫが　３８０００Ｈのとき

1c00H	1c01
3	80

```
BANKHI          EQU     1c00H           ; HI BYTE
BANKLOW         EQU     1c01H           ; LOW BYTE
```

＜＜＜＜＜＜＜＜＜＜＜＜＜＜＜＞＞＞＞＞＞＞＞＞＞＞＞＞＞

６：データトップアドレスの与え方

データトップアドレスとは、Ｚ８０エリアから見たＳＯＮＧデータのトップアドレスです。バンクのトップに置くときは、８０００Ｈとなます。つぎのように設定して下さい。

1c02H	1c03
00	80

＜＜＜＜＜＜＜＜＜＜＜＜＜＜＜＞＞＞＞＞＞＞＞＞＞＞＞＞＞

７：Ｚ８０へのプログラムの転送

Ａ：曲を鳴らすとき

３：で述べたＭ５ＣＮＴ．ＨＨＨにあたるデータをＡ００００００Ｈから送ります。

また、Ｍ５ＳＥ．ＨＨＨをＡ０１０００Ｈからおくります。

Ｚ８０をスタートさせるまえに、３：で述べたようにＢＡＮＫアドレス、およびデータスタートオフセット値をＺ８０側に与えて下さい。

Ｚ８０をスタートすると、ｋｙｆｌａｇの値はクリアします。

＜＜＜＜＜＜＜＜＜＜＜＜＜＜＜＞＞＞＞＞＞＞＞＞＞＞＞＞＞

８：ＰＡＵＳＥについて

ＰＡＵＳＥのセット

ｐａｕｓｅ＿ｆｌ（Ａ０１Ｃ０ＢＨ）に１をセットします。

ＰＡＵＳＥの解除

ｐａｕｓｅ＿ｆｌに８０Ｈをセットします。

＜＜＜＜＜＜＜＜＜＜＜＜＜＜＞＞＞＞＞＞＞＞＞＞＞＞＞＞

９：フェイドアウトについて

ｋｙｆｌａｇに　Ｅ０ｈをセットすると、フェイドアウトがかかります。

フェイドアウトのスピードおよびＶＯＬＭのレベルを任意に指定したいときは次のようにしてください。

ａ：現在のｖｏｌｍより何段階ｖｏｌｍを落としたいかをＦＯＵＴＦ（Ａ０１Ｃ０８Ｈ）にいれる。範囲は１から７ＦＨ。ただしｂｉｔ７をたてる。

ｂ：ｖｏｌｍを１段階落とすのに、何インタラプトかけるかをＦＯＵＴＴＭ＿ＳＴＲ（Ａ０１Ｃ０９Ｈ），ＦＯＵＴＴＭ（Ａ０１Ｃ０ＡＨ）にいれる。

フェイドアウトタイムは、インタラプトで同期をとっている場合ａ＊ｂ＊１６ｍｓｅｃとなります。

＜＜＜＜＜＜＜＜＜＜＜＜＜＜＞＞＞＞＞＞＞＞＞＞＞＞＞＞

１０：音色データのＳＥＴについて

```
FMREG_TB
FMREG_RRTB      :   リリースのレジスタテーブル
fmreg_tl_tb     :   トータルレベルのレジスタテーブル
```

ＦＭデータは、上記のテーブルより順にレジスタＮＯ．を参照してチップにＳＥＴされます。

トータルレベルは、キャリアにｂｉｔ７を立てることによってｖｏｌｍを変化させるレジスタを判別しています。キャリアデータのｂｉｔ７を立てることはマクロによっておこなっています。マクロを使用せず、ＦＭデータを直値で与える場合は、キャリアデータのｂｉｔ７を立てる必要があります。

＜＜＜＜＜＜＜＜＜＜＜＜＜＜＞＞＞＞＞＞＞＞＞＞＞＞＞＞

１１：ＴＩＭＥＲ

サウンドルーチンの同期を取る方法として、Ａ）インタラプトで行う　Ｂ）ＦＭタイマーＢで行う　Ｃ）ＦＭタイマーＡ，Ｂで行う　の３種類があります。

Ｚ８０　ＳＲＡＭの１Ｃ０７Ｈ（ＢＹＴＥ）の値でモード設定を行います。

```
;----------------< Piano >-------------------
fev8100::
        CNF     4,5                     ; algorithm , feedback
        MD      1,6,3,0,1,0,3,3         ; multiple , detune
        RSAR    1,31,2,20,1,31,2,20     ; key scale , attack rate
        D1R     5,5,5,7                 ; decay rate
        D2R     2,2,2,2                 ; sustain rate
        RRL     15,1,15,6,15,1,15,10    ; release rate , sustain level
        TL      30,0,30,0               ; total level
;----------------< Bass >-------------------
fev8101::
        CNF     2,7
        MD      4,0,2,0,8,0,4,0
        RSAR    0,31,0,31,0,31,0,31
        D1R     16,15,9,8
        D2R     7,0,0,0
        RRL     15,3,15,0,15,0,15,4
        TL      20h,20h,20H,00H
```

S．E．の例

チャンネルナンバー　0から6（3は除く）はFM音源を使用することを示しています。80H、0A0H、0C0Hは　PSG1、2、3チャンネルです。PSGでノイズを使用する場合は、チャンネルナンバーを0C0Hにして、CMNOISコマンドを使用してください。

```
;*********************************
;   DOOR
;*********************************
S91::

        DW      TIMB91                  ; voice top addr.

        DB      1,2                     ; base,use chan no

        DB      0A0H,06H                ; flag,chan
        DW      TAB910P                 ; FM 1ch table pointer
        DEFB    000H,000H               ; bias,volm

        DB      0A8H,0C0H               ; flag,chan
        DW      TAB911P                 ; FM 1ch table pointer
        DEFB    000H,004H               ; bias,volm
```

```
TAB910P::
        DB      FEV,0
            DB      93H,10H,3
        DB      93H,0E0H,10
        DB      CMEND
TAB911P::
        db      CMNOIS,7
        DB      0,10H,1,3
        DB      0,013H,10H,10
        DB      CMEND

TIMB91::
        CNF     2,7
        MD      0,2,2,6,4,2,1,2
        RSAR    0,27,0,20,0,25,0,21
        D1R     16,8,17,16
        D2R     0,16,14,7
        RRL     15,7,15,6,15,5,15,5
        TL      0,4,27,0
```

<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>

4：WORK RAM

つぎのように割り当てています。

```
RAM     EQU     1C00H
TB20CH  EQU     0               ; RYTHM FMCH 2  01C40H
TB21CH  EQU     TB20CH+FLGVOL   ; SONG  FMCH 0  01C70H
TB22CH  EQU     TB21CH+FLGVOL   ; SONG  FMCH 1  01CA0H
TB23CH  EQU     TB22CH+FLGVOL   ; SONG  FMCH 4  01CD0H
TB24CH  EQU     TB23CH+FLGVOL   ; SONG  FMCH 5  01D00H
TB25CH  EQU     TB24CH+FLGVOL   ; SONG  FMCH 6  01D30H
TB26CH  EQU     TB25CH+FLGVOL   ; RYTHM WORK    01D60H
TB27CH  EQU     TB26CH+FLGVOL   ; PSG CH 0      01D90H
TB28CH  EQU     TB27CH+FLGVOL   ; PSG CH 1      01DC0H
TB29CH  EQU     TB28CH+FLGVOL   ; PSG CH 2      01DF0H
TB2ACH  EQU     TB29CH+FLGVOL   ; BACK S.E FCH5 01E20H
TB2BCH  EQU     TB2ACH+FLGVOL   ; BACK S.E FCH6 01E50H
TB2CCH  EQU     TB2BCH+FLGVOL   ; S.E CH 2      01E80H
TB2DCH  EQU     TB2CCH+FLGVOL   ; S.E CH 5      01EB0H
TB2ECH  EQU     TB2DCH+FLGVOL   ; S.E CH 6      01EE0H
TB2FCH  EQU     TB2ECH+FLGVOL   ; S.E PSG CH 0  01F10H
TB30CH  EQU     TB2FCH+FLGVOL   ; S.E PSG CH 1  01F40H
TB31CH  EQU     TB30CH+FLGVOL   ; S.E PSG CH 2  01F70H
```

```
        dw      setb##
```

とすればスムーズにはこびます。

ただし　エディタは、現在００００Ｈから１ＦＦＦＨ（Ｚ８０　ＳＲＡＭエリア）にあるデータしか表示できません。ＲＡＭボード、またはＲＯＭ＿ＲＡＭボードを使用することで解決できます。）

３－Ｄ）　実行ファイルのメモリ割当について

上記をまとめると、つぎのような割当になります。

SOUND CONTROL	M5CNT13.HEX
S.E DATA	1000H M5SE.HEX
SONG DATA	任意（1C02H(WORD) で指定 M5SNG13.HEX
WORK RAM	1C00H
FM I/O	4000H - 4003H
BANK I/O	6000H
SONG DATA	8000H (BANK AREA) M5BNK13.HEX

３－Ｅ）：ソングファイルについて

ａ：　１曲づつのファィルをもちます。そこに、ヘッダーデータ、音色データ、ＳＯＮＧデータを書き込みます。

ｂ：　ヘッダーのデータフォーマットは以下の例のようになっています。
２行目で指定するチャンネル数とヘッダーの数は一致しているなくてはなりません。
リズムを使用せずＦＭ６チャンネルスベテをトーンに使用する場合、チャンネル数を７にし、リズムテーブルをＣＭＥＮＤにします。

S．E．とSONGのヘッダーのフォーマットが違うので注意が必要です。

c：　音色データは、例のようになっています。マクロについては、マクロファイルを参照してください。各データの順番をいれかえることはできません。また、キャリアのトータルレベルは、ＢＩＴ７がＯＮでなくてはなりません。

---------- データ例 ----------

```
S81::
        DW      TIMB81                          ; voice top addr.

        DB      6,3,TP81,DL81 .                 ; FM use ch,PSG,use ch
                                                ; base,delay counter
        DEFW    TAB81D                          ; rythm table pointer
        DEFB    FB81D,FA81D                     ; bias,volm

        DEFW    TAB810                          ; FM 1ch table pointer
        DEFB    FB81M,FA81M                     ; bias,volm

        DEFW    TAB811                          ; FM 2ch table pointer
        DEFB    FB81B,FA81B                     ; bias,volm

        DEFW    TAB812                          ; FM 3ch table pointer
        DEFB    FB81K1,FA81K1                   ; bias,volm

        DEFW    TAB813                          ; FM 4ch table pointer
        DEFB    FB81K2,FA81K2                   ; bias,volm

        DEFW    TAB814                          ; FM 5ch table pointer
        DEFB    FB81K3,FA81K3                   ; bias,volm

        DEFW    TAB810P                         ; PSG 1ch table pointer
        DEFB    PB81M,PA81M,PV81M,PE81M         ; bias,volm,vibr,enve

        DEFW    TAB811P                         ; PSG 2ch table pointer
        DEFB    PB81B,PA81B,PV81B,PE81B         ; bias,volm,vibr,enve

        DEFW    TAB812P                         ; PSG 3ch table pointer
        DEFB    PB81K,PA81K,PV81K,PE81K         ; bias,volm,vibr,enve

TAB810::
        DB      C3,L8,.........


TIMB81::
```

てはいけません。

－－－－－－－　実行ファイルＤＢデータ　－－－－－－－

実行ファイルをＤＢテーブルデータに変換したものです。
ゲーム組み込み時に使用します。

m5cnt.hhh	s-ram prg & data
m5sng.hhh	bank data
m5se.hhh	s.e data

３－Ｂ）ＢＡＴファイルについて

Ｍ５１３．ＢＡＴのオプションにはつぎのものがあります。

①Ａ／ａ　　アセンブル、リンクをし、Ｍ５ＣＮＴ１３．ＨＥＸ、Ｍ５ＳＮＧ１３．ＨＥＸを作る。

②Ｚ／ｚ　　ＺＩＣＥを起動する。

③ＡＬＬ／ａｌｌ　　アセンブル、リンクをし、Ｍ５ＢＮＫ１３．ＨＥＸをつくる。

④Ｂ／ｂ　　ＨＥＸファイルをＤＢに変換し、拡張子をＨＨＨにしてＢドライブにコピーする。

３－Ｃ）実行ファイルについて

前述のように、実行ファイルには

①＜ＰＲＪ＞ＳＮＧ．ＨＥＸ　　（アドレス　１０００Ｈ　以降任意）
②＜ＰＲＪ＞ＳＥ．ＨＥＸ　　（アドレス　１０００Ｈ　以降任意）
③＜ＰＲＪ＞ＣＮＴ．ＨＥＸ　　（アドレス　０００Ｈ　から）
④＜ＰＲＪ＞ＢＮＫ．ＨＥＸ　　（アドレス　８０００Ｈ　から）

の４つがあります。

３－Ｃ－①）　＜ＰＲＪ＞ＳＮＧ．ＨＥＸ

＜ＰＲＪ＞ＳＮＧ．ＨＥＸ　は、１曲だけをアセンブル、リンクするものです。リンク時にエラーがでますが、構いません。このエラーは、ダミーのラベルを持つことで解決できますが、その場合、そのままでは、全曲をリンクして＜ＰＲＪ＞ＢＮＫ．ＨＥＸを作る場合にＭＵＬＴＩ　ＤＥＦＩＮＤ　ＥＲＲＯＲが起きるので、マクロのＩＦ文等でとりはずしできるようにしておくことが必要です。

＜ＰＲＪ＞ＳＮＧ．ＨＥＸのスタートアドレスは、＜ＰＲＪ＞．ＬＮＫというリンクファイル内で与えています。Ｐ：１４００　がそれです。この場合、スタートアドレスが１４００Ｈとしてリンクします。このときＺ８０メモリの１Ｃ０２Ｈを００Ｈ、１Ｃ０３Ｈを１４Ｈとします。（４：データトップアドレスの与え方参照）

（ＷＯＲＤで　１４００Ｈ。サウンドコントロールは、このアドレ

スを参照して、曲、Ｓ．Ｅ．、ＥＮＶＥ、ＶＩＢＲ等のデータアドレスを得ます。また、エディタも同様です。）

また、１Ｃ００Ｈ以降は、プログラムのＷＯＲＫ　ＲＡＭなので、＜ＰＲＪ＞ＳＮＧ．ＨＥＸは、１ＢＦＦＨまでに納める必要があります。

３－Ｃ－②）＜ＰＲＪ＞ＳＥ．ＨＥＸ

＜ＰＲＪ＞ＳＥ．ＨＥＸ　は、ＳＥデータをリンクしたものです。スタートアドレスは　＜ＰＲＪ＞．ＭＫＦのＬＮＫコマンドで与えています。このアドレスは、つぎに述べる”Ｓ．Ｅ．データのスタートアドレス”と同一でなくてはなりません。

Ｓ．Ｅ．データのスタートアドレスは、＜ＰＲＪ＞ＴＢ．ＬＩＢで

```
setop_adr       equ     1000h
```

として、プログラムに与えます。このアサインは＜ＰＲＪ＞ＴＢ．ＳＲＣのＡＤＤＲＴＢ：：６ＢＹＴＥ目の

```
dw      setop_adr
```

に反映されます。サウンドコントロールプログラムおよびエディタは、このデータからＳ．Ｅ．のスタートアドレスを知ることができます。

＜ＰＲＪ＞ＴＢ．ＳＲＣはソングファイルとリンクされ、＜ＰＲＪ＞ＳＮＧ．ＨＥＸ、および＜ＰＲＪ＞ＢＮＫ．ＨＥＸをつくります。したがってＡＤＤＲＴＢ：：６ＢＹＴＥ目の

```
dw      setop_adr
```

は、＜ＰＲＪ＞ＳＮＧ．ＨＥＸおよび＜ＰＲＪ＞ＢＮＫ．ＨＥＸのスタートアドレス＋６に存在します。

３－Ｃ－③＜ＰＲＪ＞ＣＮＴ．ＨＥＸ　（アドレス　０００００Ｈ　から）

サウンドコントロール部です。１Ｃ０２Ｈ（ＷＯＲＤ）に書かれているアドレスをもとに、テーブル参照する方法をとっています。１Ｃ０２Ｈに書き込むデータスタートアドレスの指定は、的確に行う必要があります。

３－Ｃ－④）＜ＰＲＪ＞ＢＮＫ．ＨＥＸ　（アドレス　８０００Ｈ　から）

＜ＰＲＪ＞ＢＮＫ．ＨＥＸは、ソングデータ等、Ｍｅｇａ　Ｄｒｉｖｅのバンクエリア（カートリッジ）からＺ８０がよみだすデータをリンクしてつくります。スタートアドレス及びリンクするファイル名は、ＡＬＬ．ＬＮＫファイルで与えます。このデータで曲を鳴らしたいときは、１Ｃ０２Ｈ、１Ｃ０３Ｈにデータスタートアドレスを書き込んでおきます。通常８０００Ｈ（バンクトップ）をスタートアドレスとします。

Ｓ．Ｅ．のデータが多く１ＢＦＦＨまでに納まらない場合、ソングデータとリンクしバンクデータとします。この場ＡＤＤＲＴＢ：：の６ＢＹＴＥ目の

```
dw      setop_adr
```

を、

<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>

3：ｆｉｌｅ

つぎに、ＶＥＲ１．３のファイルを書き出します。

```
－－－－－－－  アサインファイル  －－－－－－－
m5eql3.lib              アサイン
m5tbl3.lib              アサイン

－－－－－－－  プログラム  ソースファイル  －－－－－－－
m5intl3.src             インタラプト制御ルーチン
                    include m5eql3.lib

m5cntl3.src             サウンドコントロール
                  include M5EQ13.LIB
                  include M5mcr13.lib

m5cmdl3.src             コマンドコントロール
                  include M5EQ13.LIB
                  include M5mcr13.lib

m5fmdrl3.src            リズムコントロール
                  include m5eql3.lib
                  include m5tbl3.lib
                  include M5mcr13.lib

m5psgl3.src             ＰＳＧコントロール
                  include m5eql3.lib

m5tbl3.src              ＥＮＶＥ，ＶＩＢＲ，プライオリティテーブル
                  include m5eql3.lib

m5setbl3.src            Ｓ．Ｅ．アドレステーブル
                  included in m5sel3.s

－－－－－－－  マクロライブラリ  ファイル  －－－－－－－
m5mcr13.lib             マクロ

－－－－－－－  バッチファイル  －－－－－－－
m513.bat                バッチ
m513z.bat               ＺＩＣＥ（ＺＡＸ）用バッチ

－－－－－－－  メイクファイル  －－－－－－－
m513.mkf                ＭＡＫＥファイル
MAKELURE.DEF            ＭＡＫＥ定義ファイル

－－－－－－－  リンクファイル  －－－－－－－
```

M513.LNK　　M5SNG.HEX　作成用リンクファイル
ALL.LNK　　M5BNK.HEX　作成用リンクファイル

－－－－－－－　Ｓ．Ｅデータ　ソースファイル　－－－－－－－
m5se13.s　　Ｓ．Ｅ．データファイル

－－－－－－－　ＳＯＮＧデータ　ソースファイル　－－－－－－－
m5sng113.s　　ＳＯＮＧデータファイル　Ｓ８１
m5sng213.s　　ＳＯＮＧデータファイル　Ｓ８２
m5sng313.s　　ＳＯＮＧデータファイル　Ｓ８３
m5sng413.s　　ＳＯＮＧデータファイル　Ｓ８４
m5sng513.s　　ＳＯＮＧデータファイル　Ｓ８５
m5sng613.s　　ＳＯＮＧデータファイル　Ｓ８６
m5sng713.s　　ＳＯＮＧデータファイル　Ｓ８７
m5sng813.s　　ＳＯＮＧデータファイル　Ｓ８８
m5sng913.s　　ＳＯＮＧデータファイル　Ｓ８９
m5snga13.s　　ＳＯＮＧデータファイル　Ｓ８Ａ
m5sngb13.s　　ＳＯＮＧデータファイル　Ｓ８Ｂ
m5sngc13.s　　ＳＯＮＧデータファイル　Ｓ８Ｃ
m5sngd13.s　　ＳＯＮＧデータファイル　Ｓ８Ｄ
m5snge13.s　　ＳＯＮＧデータファイル　Ｓ８Ｅ
m5sngf13.s　　ＳＯＮＧデータファイル　Ｓ８Ｆ

－－－－－－－　実行ファイル　－－－－－－－
m5cnt13.hex　　サウンドコントロールプログラム
　ＭＳＩＮＴ，Ｍ５ＣＮＴ，Ｍ５ＦＭＤＲ，Ｍ５ＣＭＤ，Ｍ５ＰＳＧをリンクしてできます。
　Ｚ８０のＳＲＡＭエリアトップ（０Ａ００００Ｈ）よりＳＴＡＹします。

m5se13.hex　　Ｓ．Ｅ．データ
　Ｍ５ＳＥＴＢ．ｒｅｌとＳ．Ｅ．データで構成されます。
　Ｚ８０のＳＲＡＭエリア（０Ａ０１０００Ｈ）よりＳＴＡＹします。

M5BNK13.HEX　　バンクデータ
　Ｍ５ＴＢ．ｒｅｌとＳＯＮＧファイルをリンクしてできます。
　Ｚ８０のバンクエリア（４Ｍのゲームの場合０００００Ｈから７ＦＦＦＦＨの間）に置きます。
　Ｚ８０からみたアドレスは、８０００Ｈ以降になります。

M5SNG.HEX　　ＳＯＮＧデータ
　Ｍ５ＴＢ．ｒｅｌとＳＯＮＧファイル１つをリンクしてできます。
　Ｚ８０のＳＲＡＭエリアにＳＴＡＹします。ＳＴＡＹアドレスはｍ５１３．ｌｎｋのリンカーのオプションで与えます。６８０００からみたアドレスは０Ａ００００Ｈ＋リンカーのオプションで与えたアドレスになります。
　なお、ｚ８０ＳＲＡＭの１Ｃ００ＨよりＷＯＲＫ　ＲＡＭとして使用しているので、この実行ファイルのＥＮＤは１ＢＦＦＨを越え

１－Ｃ）実行ファイルの生成　（アセンブル、リンク）

プロジェクトファイルのアセンブル、リンクには、＜ＰＲＪ＞．ＢＡＴを起動して行います。Ａオプションでコントロール、曲、Ｓ．Ｅ．の実行ファイルを作成します。ＡＬＬオプションでバンクデータ（全ての曲をつなげたもの）を作成します。

例　TST A

＜ＰＲＪ＞ＳＮＧ．ＨＥＸを作成時にリンクエラーがでますが、差し支えありません。
アセンブル、リンク時にはＴＯＯＬＳ－１，２，３のＭＡＫＥ．ＥＸＥを使用して履歴の管理をしています。

－－－－ ＺＩＣＥ（　ＺＡＸ　）使用の場合　－－－－

アセンブル、リンクを終了すると、ＩＣＥに入ります。ＩＣＥに入らない場合、１度エディタをＦＬＡＧ　ＭＯＤＥ画面に切り替えて、メニュー画面に戻してください（ＳＴＡＲＴ　ボタンで戻る）。ＩＣＥのプロンプトがでている状態で、

ＳＥＴＵＰ

と入力すると、ＩＣＥのマッピングをし、ＲＡＭの初期設定をし、

＜ＰＲＪ＞ＣＮＴ．ＨＥＸ　（アドレス　０００００Ｈ　から）
＜ＰＲＪ＞ＳＥ．ＨＥＸ　（アドレス　１０００Ｈ　以降任意）
＜ＰＲＪ＞ＳＮＧ．ＨＥＸ　（アドレス　１０００Ｈ　以降任意）

をロードします。

＜ＰＲＪ＞ＢＮＫ．ＨＥＸ　（アドレス　８０００Ｈ　から）

をロードしたい場合　ＢＮＫと入力します。このようなＩＣＥのマクロは、＜ＰＲＪ＞Ｚ．ＭＡＣに登録してあります。
Ｇ　０で実行します。

１－Ｄ）エディタの起動

ＲＯＭ＿ＲＡＭボードにＳＯＵＮＤエディタのＲＯＭをさし、ボードのロータリーＳＷを１にします。ボードをメガドライブにさし、メガドライブをＰＯＷＥＲ　ＯＮして、エディタを起動します。このとき、Ｚ８０ＳＲＡＭには、サウンドコントロールプログラムが転送されます。

１－Ｅ）実行ファイルのダウンロード

パッドでカーソルを　”ＬＯＡＤ　ＭＯＤＥ”にあわせ、Ｃボタンを押します。すると、画面がＬＯＡＤＥＲ　ＭＯＤＥに切り替わります。
バンクアドレスを１００００に設定し、”ＳＴＡＲＴ”にカーソルをあわせてＣボタンを押すと”ＲＥＡＤＹ”と表示されます。

つぎにＰＣ－９８側でＩＣＤ＿ＬＤＲ．ＣＯＭを起動してソングデータを転送します。

例　　ICD_LDR TSTSNG.HEX

”ＣＯＭＰＬＥＴＥ”表示がでたら、Ｃボタン２回を押してＳ．Ｅ．データを受け取る準備をします。そしてＰＣ－９８よりデータを転送します。

例　　ICD_LDR TSTSE.HEX

１－Ｆ）データスタートアドレスの指定

バッドのＳＴＡＲＴボタンを押してＳＥＬＥＣＴ画面に戻ります。そして、”ＦＬＡＧ　ＭＯＤＥ”をセレクトしてください。

”ＦＬＡＧ　ＭＯＤＥ”の画面最下は、つぎのデータ列が表示されているはずです。

```
01   00   00 . 00   00   00    00      00      00      80
|         |         |         |       |       |       |
BANK     DATA START TIMER A   TIMER B  TIMER   NO USE   KY FLAG
ADDR.(2) ADDR.(2)    (2)      (1)     FLAG(1)    (1)    (1)
```

カーソルを最下行に移動させ、データスタートアドレスを入力します。１－Ｃでできたソングの実行データのスタートアドレスは１４００Ｈなので、最下行を

01 00 00 14 00 00 00 00 00

とします。データはＡボタンでＩＮＣ、ＢボタンでＤＥＣします。

１－Ｇ）ＳＯＮＧ、Ｓ．Ｅ．を鳴らす。

つぎに、下から２行めの　”８０”の表示を”８１”に変えてＣボタンを押します。すると簡単なリズムが鳴ります。

Ｓ．Ｅ．を鳴らすときは下から２行めを９０Ｈから９ＢＨのデータを入力してＣボタンを押します。

＜＜＜＜＜＜＜＜＜＜＜＜＜＜＞＞＞＞＞＞＞＞＞＞＞＞＞＞

２：Ｚ８０ＩＣＥとの併用について

Ｚ８０ＩＣＥとＲＯＭ＿ＲＡＭボード、エディタを併用する場合、次の点に注意してください。

Ａ）電源の強化

１２Ｖ　１．３Ａのスイッチングレギュレーターで電源を供給することをおすすめします。

Ｂ）プログラムの転送

エディタに組み込んだルーチンをＺ８０側に転送して使用する場合、全メモリをユーザー側にマッピングした後、エディタのＳＥＮＤ　ＭＯＤＥを使用して転送してください。

ディスク上の実行ファイルをローダーを使用して転送する場合、ＩＣＥをブレーク状態にして行うことをおすすめします。

1989.10.4

MEGA DRIVE Z80 SOUND FILE
VER 1.3 MANUAL

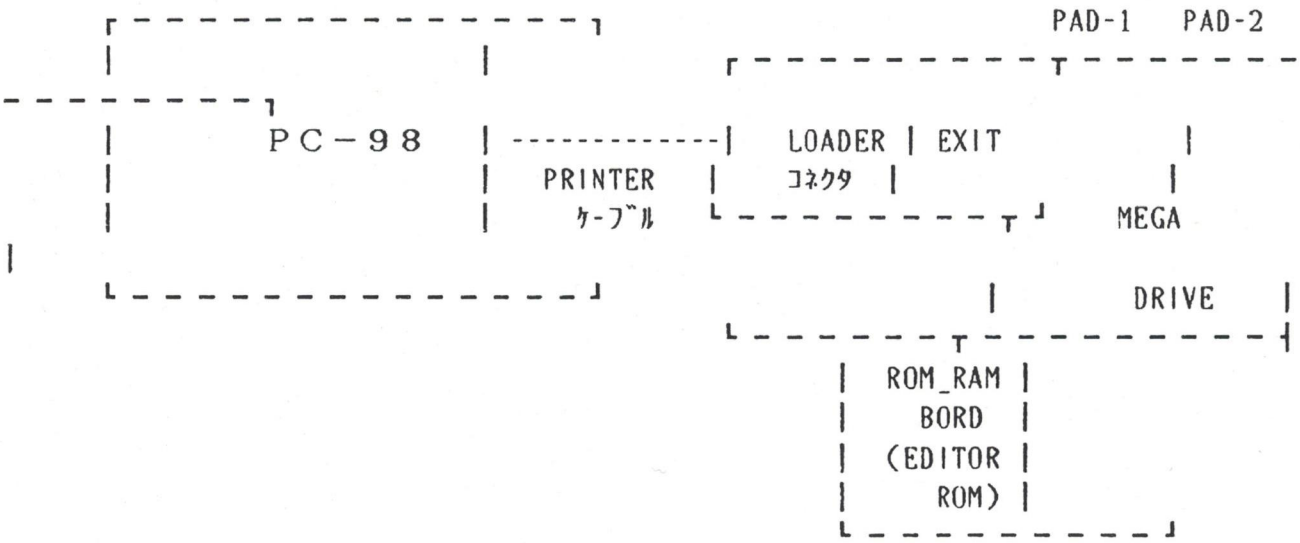
ＳＥＧＡエンタープライゼス
第二研究開発部
上保　徳彦

１：概要
２：Ｚ８０ＩＣＥとの併用について
３：ＦＩＬＥ
４：ＷＯＲＫ　ＲＡＭ
５：バンク設定の方法
６：データトップアドレスの与え方
７：Ｚ８０へのプログラムの転送
８：ＰＡＵＳＥについて
９：フェイドアウトについて
１０：音色データのＳＥＴについて
１１：ＴＩＭＥＲ
１２：プライオリティ

＜＜＜＜＜＜＜＜＜＜＜＜＜＜＜＞＞＞＞＞＞＞＞＞＞＞＞＞＞

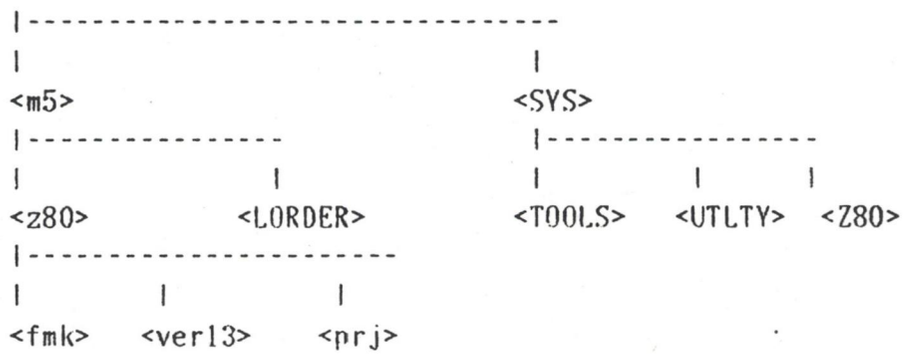
１：概要

１－Ａ）システム
次のシステムを構成してください。



１－Ｂ）ディレクトリ及びユーティリティ

つぎのディレクトリ構成を作り、マスターファイルを移して下さい。



＜ＴＯＯＬＳ＞には、ＡＳＣＩＩ出版から出されているＴＯＯＬＳ－１，２，３をいれます。必要なツールは

ＳＥＤ．ＥＸＥ
ＭＡＫＥ．ＥＸＥ
ＨＥＡＤ．ＥＸＥ
ＴＡＩＬ．ＥＸＥ

です。

＜ＵＴＩＬＴＹ＞には、ＣＦ．ＥＸＥをいれます。このツールは、ＨＥＸファイルをＤＢテーブルファイルに変換するために用います。

１－Ｃ）プロジェクトディレクトリ、ファイルの作成

つぎのようにして、プロジェクトディレクトリおよびプロジェクトファイルを作ります。

１　＜ｆｍｋ＞のディレクトリにはいる。

２　ｆｍｋ．ｂａｔの中のパラメータ（ｙｅａｒ，ｍｏｎｔｈ，ｄａｙ，ｐｒｏｊ－ｎａｍｅ，ｐｅｒｓｏｎ）を書き換える。

例　1989.9.28 SOUND-TEST SEGA

３　ｆｍｋ　＜プロジェクト名（３文字まで）＞と入力する。

例　FMK TST

このｂａｔで＜ｖｅｒ１３＞のマスターファイルをもとにプロジェクトファイルがプロジェクトディレクトリにつくられます。プロジェクトファイルにはｆｍｋ．ｂａｔの中のパラメータ（ｙｅａｒ，ｍｏｎｔｈ，ｄａｙ，ｐｒｏｊ－ｎａｍｅ，ｐｅｒｓｏｎ）が書き込まれ、ｉｎｃｌｕｄｅファイル名が書き換えられます。

作業には、ＴＯＯＬＳ１－２－３（ＡＳＣＩＩ出版）が必要です。

また、ＥＸＣＯＭ、ＩＮＴコマンドで設定可能です。

１１－Ａ）インタラプトで行う場合

Ｚ８０　ＳＲＡＭの１Ｃ０７Ｈを０にします。サウンドルーチンは、インタラプト毎にＣＡＬＬされ、ＳＯＮＧ、Ｓ．Ｅ．の処理をします。

１１－Ｂ）ＦＭタイマーＢで行う場合

Ｚ８０　ＳＲＡＭの１Ｃ０７Ｈを４０Ｈにします。また、タイマーＢの値をＺ８０　ＳＲＡＭの１Ｃ０６Ｈ（ＢＹＴＥ）にセットします。ＴＩＭＥＲコマンドを使用することでセットできます。

サウンドルーチンは、ＴＩＭＥＲＢがオーバーフローする毎にＣＡＬＬされ、ＳＯＮＧ、Ｓ．Ｅ．の処理をします。

１１－Ｃ）ＦＭタイマーＡ、Ｂで行場合

Ｚ８０　ＳＲＡＭの１Ｃ０７Ｈを８０Ｈにします。また、タイマーＡの値をＺ８０　ＳＲＡＭの１Ｃ０４Ｈ（ＷＯＲＤ，ＬＯＷ、ＨＩＧＨ）に、タイマーＢの値１Ｃ０６Ｈ（ＢＹＴＥ）にセットします。ＴＩＭＥＲコマンドを使用することでセットできます。

サウンドルーチンは、ＴＩＭＥＲＡ、Ｂがオーバーフローを監視し、Ａのオーバーフローで曲の処理を、Ｂのオーバーフローで Ｓ．Ｅ．の処理を行います。

注意）

マニュアルのタイマー値を求める式定数には誤りがあり、２倍の定数が正しい値です。したがって、ＴＩＭＥＲ　Ｂを１６ｍｓｅｃに設定するタイマー値は０ＣＢＨ、８ｍｓｅｃは０Ｅ６Ｈです。同様にＴＩＭＥＲ　Ａ　１６ｍｓｅｃは　０ｂ４Ｈ、８ｍｓｅｃは２５ＡＨです。

＜＜＜＜＜＜＜＜＜＜＜＜＜＜＜＞＞＞＞＞＞＞＞＞＞＞＞＞＞＞

１２：プライオリティ

プライリティは、＜ｐｒｊ＞ｔｂ．ｓｒｃのｐｌｔｂに記されたデータによっています。実際の記述は、＜ｐｒｊ＞ｔｂ．ｌｉｂのアサインで行います。

プライオリティデータは次のようになっています。

０－７ＦＨ　：　プライオリティのチエック、登録が行われＪＲ。
０がＭＩＮＩＭＡＭ、７ＦがＭＡＸです。

８０Ｈ－ＦＦＨ：　最優先されます。しかし、プライオリティの登録は行われません。

1989.10.4

ＲＯＭ－ＲＡＭボードについて

ＳＥＧＡエンタープライゼス
第二研究開発部
上保　徳彦

Ａ：ＲＯＭ－ＲＡＭボードの概要
Ｂ：ロータリーＳＷとＲＡＭアドレスの関係について
Ｃ：コントロールポート（０Ａ１３０Ｅ０Ｈ－ＷＯＲＤ）の仕様
Ｄ：ＤＩＰ　ＳＷの仕様
Ｅ：ＲＥＳＥＴ後の初期設定

Ａ：ＲＯＭ－ＲＡＭボードの概要

ＲＯＭとＲＡＭを同一基盤上にのせることができます。
ＲＯＭサイズは、８００００Ｈ（５１２Ｋバイト）で、０００００ＯＨから７ＦＦＦＦＨまでのエリアをカバーします。
ＲＯＭの種類は、ボードのディップＳＷにより、２Ｍ／１Ｍ、ＭＡＳＫ／ＪＥＤＥＣの切り替えができます。

ＲＡＭサイズ　２００００Ｈ（１２８Ｋバイト）で、ボードに付いているロータリーＳＷの指定で、００００ＯＨから７ＦＦＦＨまたは、２００００ＯＨから２１ＦＦＦＦＨまでの任意のエリアをカバーします。ＲＯＭと重複したエリアはＲＡＭが優先されます。
ＲＡＭのデータは、バックアップされます。

ボードは、コントロールポート（０Ａ１３０Ｅ０Ｈ－ＷＯＲＤ）をもち、ここにデータを書き込むことで、ＲＡＭのＷＲＩＴＥプロテクトの指定、バンクアドレスの表示／非表示の指定、ができます。また、ＲＥＡＤすることで、上記の設定状態およびロータリーＳＷの状態が判別できます。

Ｂ：ロータリーＳＷとＲＡＭアドレスの関係について

ロータリーＳＷとＲＡＭアドレスの関係は、つぎの通りです。

ＳＷ	０	00000H - 1FFFFH
	１	08000H - 27FFFH

2	10000H - 2FFFFH
3	18000H - 37FFFH
4	20000H - 3FFFFH
5	28000H - 47FFFH
6	30000H - 4FFFFH
7	38000H - 57FFFH
8	40000H - 5FFFFH
9	48000H - 57FFFH
A	50000H - 6FFFFH
B	58000H - 77FFFH
C	60000H - 7FFFFH
D	68000H - 7FFFFH
E	70000H - 7FFFFH
F	200000H -21FFFFH

ボードには３２ＫのＲＡＭが４つ載っています。ロータリーＳＷを回すと４つのＲＡＭのアドレスがシフトされたかの印象を受けますが、実際は異なります。図－２）のようにローテイトシフトされる形になります。

Ｃ：コントロールポート（０Ａ１３０Ｅ０Ｈ－ＷＯＲＤ）の仕様

コントロールポート（０Ａ１３０Ｅ０Ｈ）の仕様は、つぎのとおりです。

①ボードのプロテクトの設定
ＢＩＴ５＝１　で　ＷＲＩＴＥ不可／　０で可となります。

②バンクアドレス表示、非表示の設定
４つのＲＡＭそれぞれの最終ＷＯＲＤにＺ８０が参照できるバンクアドレスの表示が可能です。
コントロールポートのＢＩＴ０から３までがＲＡＭの０から３に対応しています。ビットを１にすることでバンクアドレスが表示されます。表示状態ではそのＲＡＭの最終ＷＯＲＤにデータを書き込むことはできません。
バンクアドレスは、ＢＩＴ１５からＢＩＴ２３までが表示されます。
図－３）にバンクアドレスの表示フォーマットを示します。

③各設定の読み出し
コントロールポートを読み出すことで、ボードのプロテクトの設定状態、バンクアドレス表示、非表示の設定状態、ロータリーＳＷの状態が解ります。
図－４）にコントロールポートのビットマップを示します。

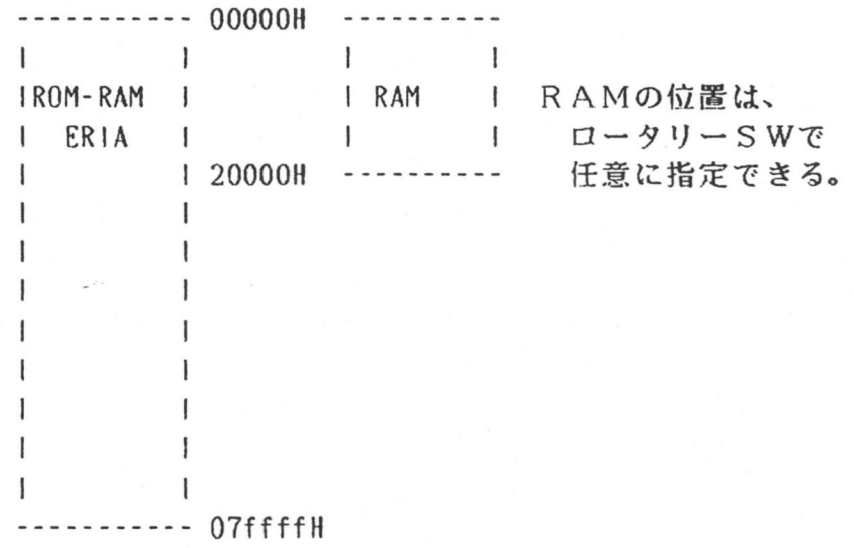
Ｄ：ＤＩＰ　ＳＷの仕様

ボードのディップＳＷによりＲＯＭの種類を、２Ｍ／１Ｍ、
ＭＡＳＫ／ＪＥＤＥＣに切り替えることができます。
図－４）にＳＷのフォーマットを記します。

Ｅ：ＲＥＳＥＴ後の初期設定

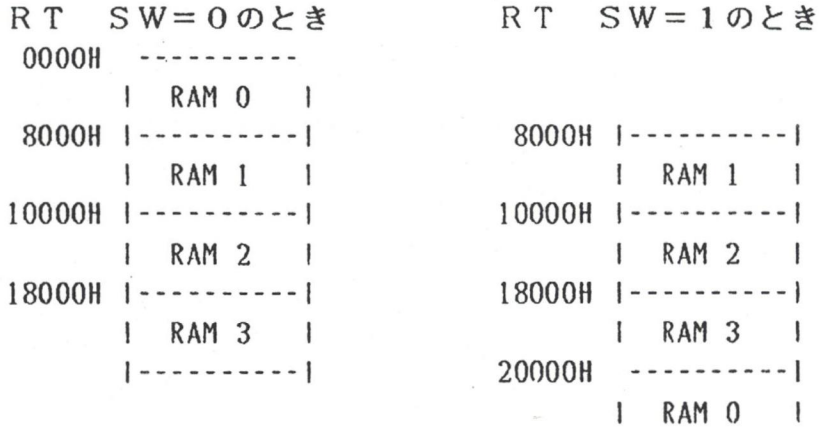
本体のＲＥＳＥＴボタンが押されると、ＲＯＭ＿ＲＡＭボードは、
ＷＲＩＴＥプロテクト、バンクアドレスのマスク状態になります。

図－１）



図－２）ロータリーＳＷとＲＡＭの関係

ＲＡＭは、シフトするのではなく、ローテイトします。



図－３）

Ｚ８０が参照できるバンクアドレスを読み出すことができます。
コントロールポートのＢＩＴ０からＢＩＴ３をＯＮにすること
で、ＲＡＭ０からＲＡＭ３までの後ろ２バイトにバンクアドレス

が現れます。フォーマットは次のとおりです。



RAM ERIA	FE								FF							
ｱﾄﾞﾚｽBIT	A23	A22	A21	A20	A19	A18	A17	A16	A15	0	0	0	0	0	0	0
DATA(ﾊﾞﾝｸ 38000H)	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0

図－４）コントロールポート

0A130E0H　BIT MAP

	ﾛｰﾀﾘｰ SW				WRITE PROTECT				RAM EREA BANK DISP			
					1:ON /0:OFF				1:DISP/0:MASK			
F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
															RAM0
										PROTECT				RAM1	
													RAM2		
												RAM3			

図－５）ＤＩＰ　ＳＷ

	ON	OFF
1	2M ROM	1M ROM
2	MASK ROM	JEDEC ROM
3	未使用	
4	未使用	

ＰＳＧマニュアル

１９８９年１０月　４日　改訂

ＰＳＧ（Ｐｒｏｇｒａｍａｂｌｅ　Ｓｏｕｎｄ　Ｇｅｎｅｒａｔｏｒ）は、３個のトーンシェネレータと１個のノイズジェネレータを内臓しています。

ＣＰＵからのアクセスはＩ／Ｏエリア７ＦＨへのライトです。基本クロックは、ＮＴＳＣのときは３．５７９５４５ＭＨｚ、ＰＡＬのときは３．５４６８９５ＭＨｚです。この違いは１％ぐらいですので考慮しなくてもよいですが、インタラプトの間隔が違いますので注意して下さい。ＣＰＵより送られるデータはＰＳＧ内部でただちにラッチされますのでウエイトの必要はありません。パワーオンリセットでサウンドの出力はＯＦＦになりますが、ソフト的にもプログラムの始めのほうで出力ＯＦＦをして下さい。

（１）トーンジェネレータ

各トーンジェネレータは、周波数設定部（プログラマブルカウンタ）及びレベル設定部（プログラマブルアッテネータ）より構成されます。

①１０ｂｉｔ分周比ｎの算出方法

周波数設定部では、基本クロックが分周回路により１／３２に分周され、それがＦ９（ＭＳＢ：最上位ビット）～Ｆ０（ＬＳＢ：最下位ビット）の１０ｂｉｔで設定されるトーンカウンタで分周されます。従って、基本クロック周波数を３２で割り、さらに希望周波数で割った値をＦ９～Ｆ０に設定すれば、希望の周波数を出力させることができます。

$$n = N / (32 \times f)$$

ここで　ｎ：１０ｂｉｔ分周比（Ｆ９～Ｆ０）
Ｎ：基本クロック
ｆ：希望周波数

②トーン周波数設定

トーンカウンタに１０ｂｉｔ分周比ｎ（Ｆ９～Ｆ０）をセットして希望の周波数に設定します。１ｓｔ　　バイト、２ｎｄバイトの区別は最上位ビットで行なわれます。

１ｓｔバイト

D7	D6	D5	D4	D3	D2	D1	D0
＊	REG. ADDR.			n			
1	R2	R1	R0	F3	F2	F1	F0

２ｎｄバイト

D7	D6	D5	D4	D3	D2	D1	D0
＊				n			
0	×	F9	F8	F7	F6	F5	F4

R2	R1	R0	コントロールレジスタ割り当て	1st
0	0	0	トーンジェネレータ1	8×
0	1	0	トーンジェネレータ2	A×
1	0	0	トーンジェネレータ3	C×

③周波数設定の例

例として基本クロック周波数３．５７９５４５ＭＨｚ（ＮＴＳＣ）希望周波数４４０Ｈｚ（音楽では　　中央の”ラ”の音になります）をトーン１から出力させることを考えてみます。

ａ．分周比ｎの計算

ｎ＝Ｎ／（３２×ｆ）
＝３５７９５４５／（３２×４４０）
≒２５４．２２９

ｎは１０ｂｉｔの整数ですので、四捨五入してｎ＝２５４とします。

従って、実際に出力される周波数は

ｆ＝Ｎ／（３２×ｎ）
＝３５７９５４５／（３２×２５４）
≒４４０．３９７（Ｈｚ）

ここで、セント誤差△Ｃは下記の式で求まります。

△Ｃ＝｛（ｆ’－ｆ）／ｆ｝／（1200√２－１）
≒｛（４４０．３９７－４４０）／４４０｝／（1200√２－１）
≒（０．３９７／４４０）／０．０００５７８
≒１．５６
ｆ：真の周波数
ｆ’：実際の周波数　　1200√２：２の１２００乗根

ｂ．ＰＳＧに送るデータ

ｎ＝２５４
＝００１１１１１１１０Ｂ

１stバイト

＊	REG. ADDR.			n			
1	R2	R1	R0	F3	F2	F1	F0

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	1	1	1	0

２ndバイト

＊		n					
0	×	F9	F8	F7	F6	F5	F4

D7	D6	D5	D4	D3	D2	D1	D0
0	×	0	0	1	1	1	1

④トーンレベル設定

トーンジェネレータで設定された周波数はレベル設定部へ送られ、その音量が設定されます。レベル設定部はプログラマブルアッテネータで、４ｂｉｔのアッテネーション値によって０［ｄｂ］から０ＦＦまで１６ステップの音量設定ができます。

１stバイトのみ

D7	D6	D5	D4	D3	D2	D1	D0

＊	REG. ADDR.			ATT. DATA			
1	R2	R1	R0	A3	A2	A1	A0

R2	R1	R0	コントロールレジスタ割り当て	HEX
0	0	1	トーン１　アッテネーション	９×
0	1	1	トーン２　アッテネーション	Ｂ×
1	0	1	トーン３　アッテネーション	Ｄ×

アッテネーション

[db]	A3 A2 A1 A0	HEX
0	0 0 0 0	×0
2	0 0 0 1	×1
4	0 0 1 0	×2
6	0 0 1 1	×3
8	0 1 0 0	×4
10	0 1 0 1	×5
12	0 1 1 0	×6
14	0 1 1 1	×7

[db]	A3 A2 A1 A0	HEX
16	1 0 0 0	×8
18	1 0 0 1	×9
20	1 0 1 0	×A
22	1 0 1 1	×B
24	1 1 0 0	×C
26	1 1 0 1	×D
28	1 1 1 0	×E
OFF	1 1 1 1	×F

（２）ノイズジェネレータ

ノイズジェネレータは、ノイズ発生回路とレベル設定部より構成されます。ノイズ発生回路より供給されるノイズのソースはＥＸ－ＯＲフィードバック付きのシフトレジスタです。ノイズコントロールレジスタが変更されるたびにシフトレジスタはクリアされます。

このシフトレジスタのシフトクロックは、ＮＦ０とＮＦ１で決まる４種類のモードで決定されます。例えば、ＮＦ０＝ＮＦ１＝０のときはシフトクロックは（Ｎ／３２）／５１２となり、ここでＦＢ＝０の場合はこのシフトクロックが１６分周されて周波数がＮ／（３２×５１２×１６）の同期ノイズになり、ＦＢ＝１の場合はこのシフトクロックによってＥＸ－ＯＲフィードバック付きでシフトレジスタが駆動されてホワイトノイズがつくられます。

①ノイズ発生回路コントロール

１stバイトのみ

D7	D6	D5	D4	D3	D2	D1	D0
＊	REG. ADDR.			SHIFT			
1	1	1	0	×	FB	NF1	NF0

FB	ノイズ選択

NF1	NF0	シフトクロック	K

0	同期ノイズ
1	ホワイトノイズ

0	0	（N／32）／K	512
0	1	（N／32）／K	1024
1	0	（N／32）／K	2048
1	1	トーンジェネレータ3	

↑

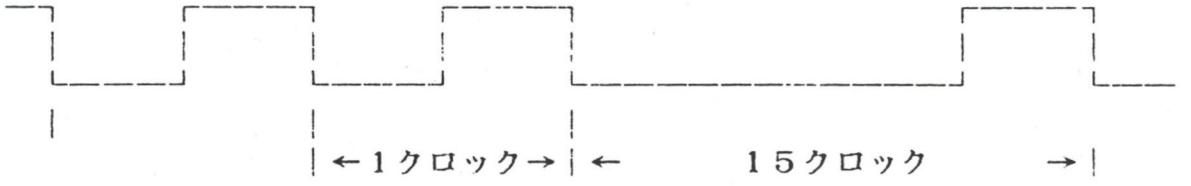
このときノイズ周波数の連続可変が可能です。

N：基本クロック

a．同期ノイズ（ＦＢ＝０）

○ＮＦ０＝ＮＦ１＝１以外のとき

Ｎ／（３２×Ｋ）



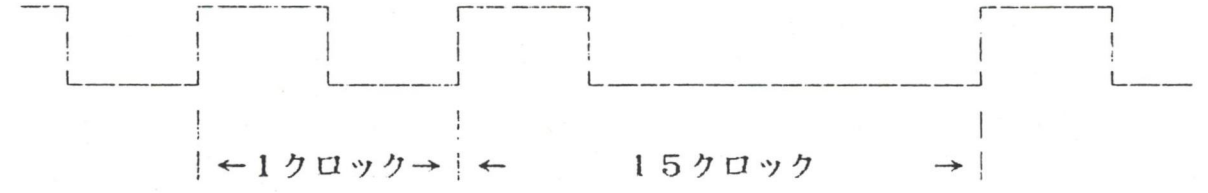
ノイズ



※ノイズの周波数は
（Ｎ／（３２×Ｋ））／１６＝Ｎ／（３２×Ｋ×１６）

○ＮＦ０＝ＮＦ１＝１のとき（トーン３によるコントロール）

トーン３



ノイズ



※ノイズの周波数は
ＴＯＮＥ３の周波数／１６＝Ｎ／（３２×ｎ×１６）

b．ホワイトノイズ（ＦＢ＝１）

ＮＦ１＝０，ＮＦ＝１，ｎ＝１のときのスペクトラム

②ノイズレベル設定

1 stバイトのみ

D7	D6	D5	D4	D3	D2	D1	D0

＊	REG. ADDR.			ATT. DATA			
1	1	1	1	A3	A2	A1	A0

※アッテネーションのコントロールはトーンの場合と同じです。

（３）レジスタアドレスフィールド

ＰＳＧは１ＳＴバイトのＲ２～Ｒ０の３ビットによってどのコントロールレジスタのデータが送られたのかを判別します。これは下の表のようになります。

R2	R1	R0	コントロールレジスタ割り当て	1st
0	0	0	トーン１　分周比	８×
0	0	1	トーン１　アッテネーション	９×
0	1	0	トーン２　分周比	Ａ×
0	1	1	トーン２　アッテネーション	Ｂ×
1	0	0	トーン３　分周比	Ｃ×
1	0	1	トーン３　アッテネーション	Ｄ×
1	1	0	ノイズ発生回路コントロール	Ｅ×
1	1	1	ノイズ　アッテネーション	Ｆ×

（４）音の要素とＰＳＧとの対応

音の要素	物理的要素	ＰＳＧへの対応
音の高低	周波数	(1)－②トーン周波数設定、(2)－①ノイズ発生回路コントロール
音色	倍音構成	主に波形的なものですが、ＰＳＧはトーンではデューティ５０％のパルス波形のみで、３つの周波数を同時に出力できることによりアッテネーションコントロールとの併用で倍音構成のコントロールが可能です。同期ノイズモードではデューティ６．２５％のトーンを出力します。
音の強弱	振幅	３つのトーンとノイズは、(1)－③、(2)－②のように４ビットデータでアッテネーションをコントロールすることが可能です。
音の鳴り方	エンベロープ	各々アッテネーションを外部データによりコントロールすることで実現できます。アッテネーションの４ビットデータを各々のステップで書き換え、エンベロープのシーケンスコントロールを行う範囲において可能です。周波数については、トーン

		周波数、ノイズ周波数を成分コントロールする範囲において可能です。

１２平均律のときの音程と分周比の関係
（基本クロック：３．５７９５４５ＭＨｚ）

音程	分周比	HEX		PSG 出力 [Hz]	実際の 周波数 [Hz]
		1st	2nd		
A 2	1017	X9	3F	109.991	110.000
A#2	960	X0	3C	116.522	116.541
B 2	906	XA	38	123.467	123.471
C 3	855	X7	35	130.832	130.813
C#3	807	X7	32	138.613	138.591
D 3	762	XA	2F	146.799	146.832
D#3	719	XF	2C	155.578	155.563
E 3	679	X7	2A	164.744	164.814
F 3	641	X1	28	174.510	174.614
F#3	605	XD	25	184.894	184.997
G 3	571	XB	23	195.904	195.998
G#3	539	XB	21	207.534	207.652
A 3	508	XC	1F	220.199	220.000
A#3	480	X0	1E	233.044	233.082
B 3	453	X5	1C	246.934	246.942
C 4	428	XC	1A	261.357	261.626
C#4	404	X4	19	276.884	277.183
D 4	381	XD	17	293.598	293.665
D#4	360	X8	16	310.725	311.127
E 4	339	X3	15	329.973	329.628
F 4	320	X0	14	349.565	349.228
F#4	302	XE	12	370.400	369.994
G 4	285	XD	11	392.495	391.995
G#4	269	XD	10	415.840	415.305
A 4	254	XE	0F	440.397	440.000
A#4	240	X0	0F	466.087	466.164
B 4	226	X2	0E	494.960	493.883
C 5	214	X6	0D	522.715	523.251
C#5	202	XA	0C	553.767	554.365
D 5	190	XE	0B	588.742	587.330
D#5	180	X4	0B	621.450	622.254
E 5	170	XA	0A	658.005	659.255
F 5	160	X0	0A	699.131	698.456
F#5	151	X7	09	740.801	739.989
G 5	143	XF	08	782.244	783.991
G#5	135	X7	08	828.600	830.609
A 5	127	XF	07	880.795	880.000
A#5	120	X8	07	932.174	932.328
B 5	113	X1	07	989.920	987.767

C 6	107	XB	06	1045.429	1046.502
C#6	101	X5	06	1107.534	1108.731
D 6	95	XF	05	1177.484	1174.659
D#6	90	XA	05	1242.899	1244.508
E 6	85	X5	05	1316.011	1318.510
F 6	80	X0	05	1398.262	1396.913
F#6	76	XC	04	1471.854	1479.978
G 6	71	X7	04	1575.506	1567.982
G#6	67	X3	04	1669.566	1661.219
A 6	64	X0	04	1747.827	1760.000
A#6	60	XC	03	1864.349	1864.655
B 6	57	X9	03	1962.473	1975.533

※f＝3579545／（32×n）　　n：10bit分周比
最高周波数はn＝1

ＰＳＧマニュアル　ＥＮＤ

	8
C	3A69H
C #	3A8DH
D	3AB4H
D #	3ADDH
E	3B09H
F	3B37H
F #	3B68H
G	3B9CH
G #	3BD3H
A	3C0DH
A #	3C4BH
B	3C8CH

・ポルタメント
1 OCT間 (Ca - Ca-1)
を0269Hとしてかんがえる

平均 100cent = 033H

・効果音モード 03 DF

YM2151 のモデルで DT2が
使用してあった場合。

DT2 = 1

基本周波数に +132H (TONE, 約4度)

DT2 = 2

基本周波数に +187H (TONE, 約5度)

DT2 = 3

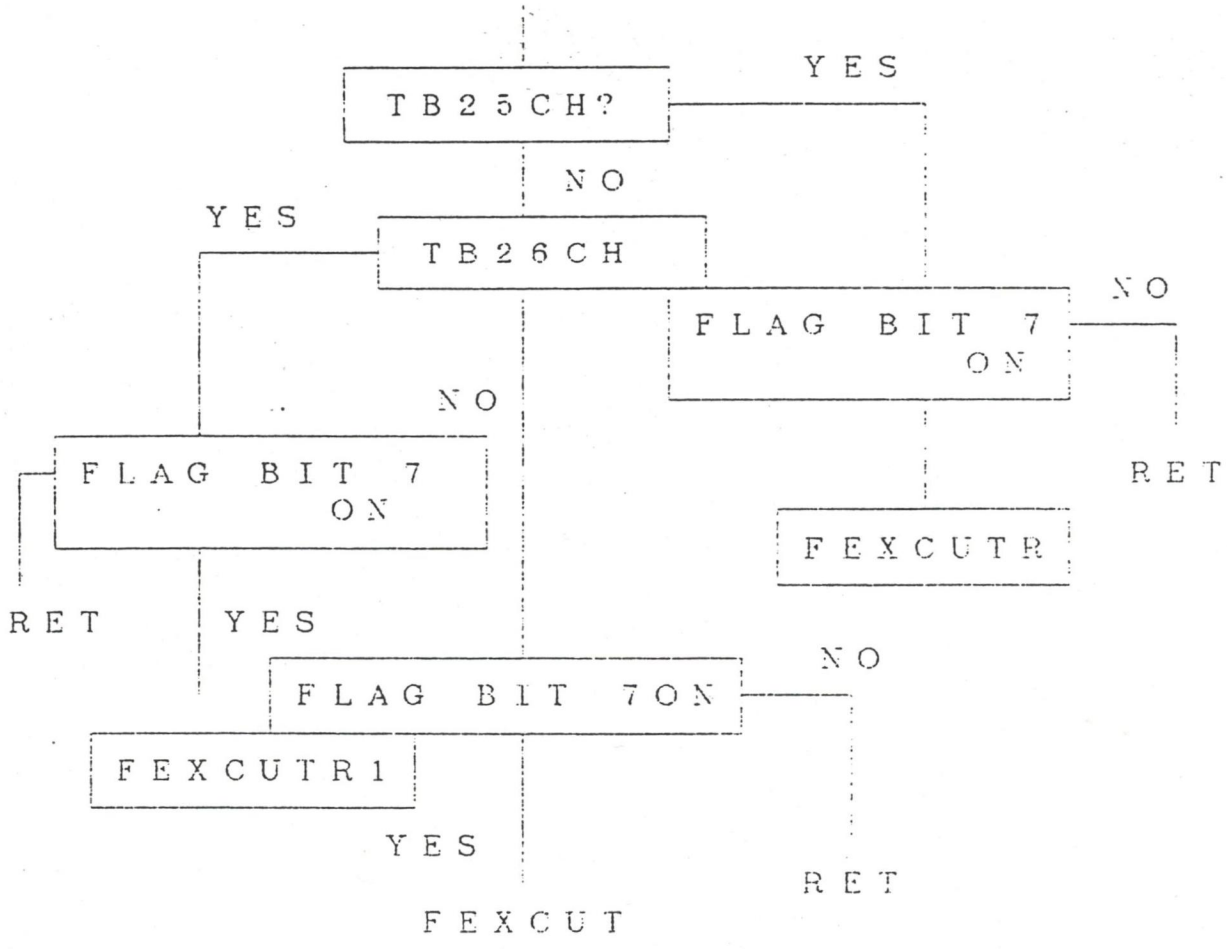
基本周波数に +1DBH (TONE,
減7度 7度の間)

する。

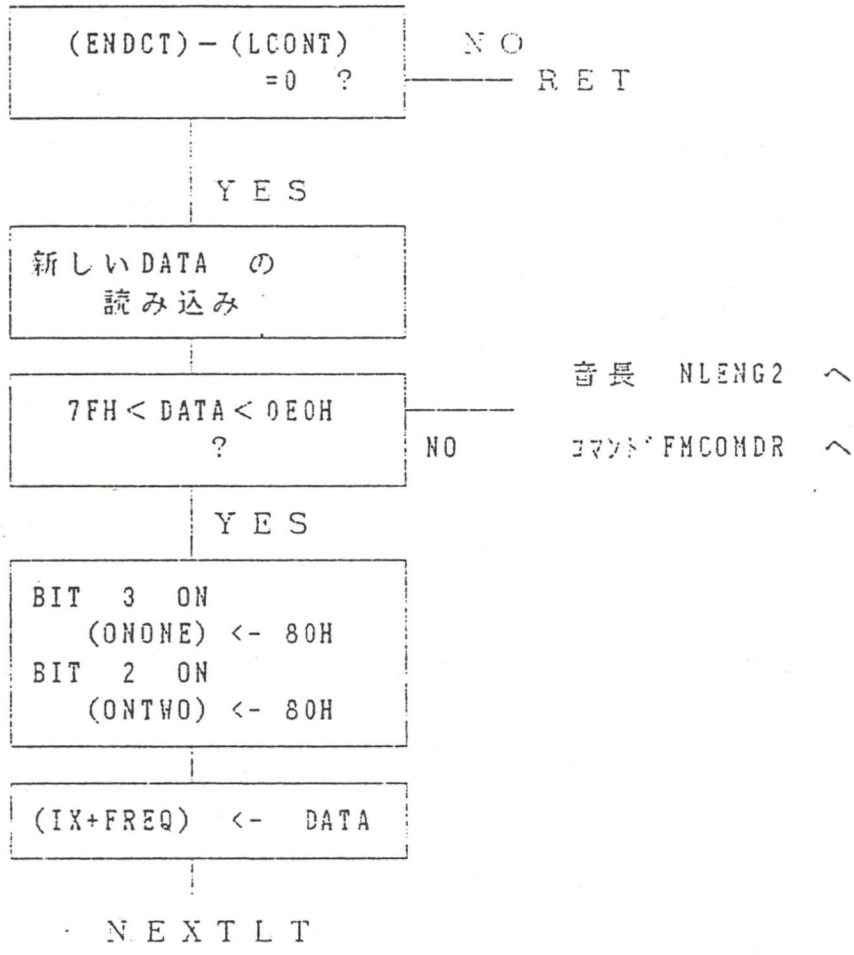
FM DIRECT PROG

	0	1	2	3	4	5
C	0A69H	1269H	1A69H	2269H	2A69H	3269H
C #	0A8DH	128DH	1A8DH	228DH	2A8DH	328DH
D	0AB4H	12B4H	1AB4H	22B4H	2AB4H	32B4H
D #	0ADDH	12DDH	1ADDH	22DDH	2ADDH	32DDH
E	0B09H	1309H	1B09H	2309H	2B09H	3309H
F	0B37H	1337H	1B37H	2337H	2B37H	3337H
F #	0B68H	1368H	1B68H	2368H	2B68H	3368H
G	0B96H	1396H	1B96H	2396H	2B96H	3396H
G #	0BD3H	13D3H	1BD3H	23D3H	2BD3H	33D3H
A	0C0DH	140DH	1C0DH	240DH	2C0DH	340DH
A #	0C48H	1448H	1C48H	2448H	2C48H	3448H
B	0C8CH	148CH	1C8CH	248CH	2C8CH	348CH

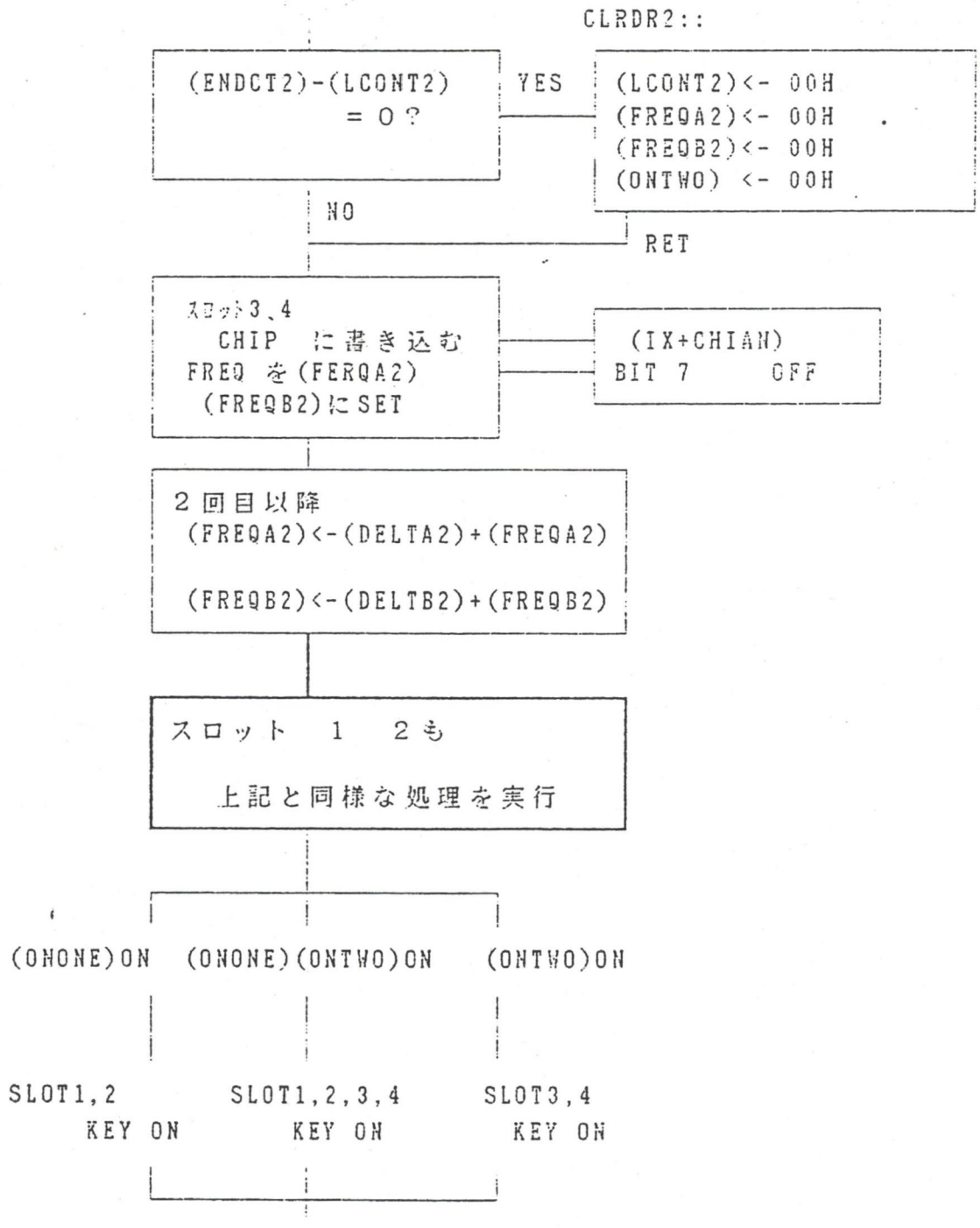
MAIN: FMSCAN

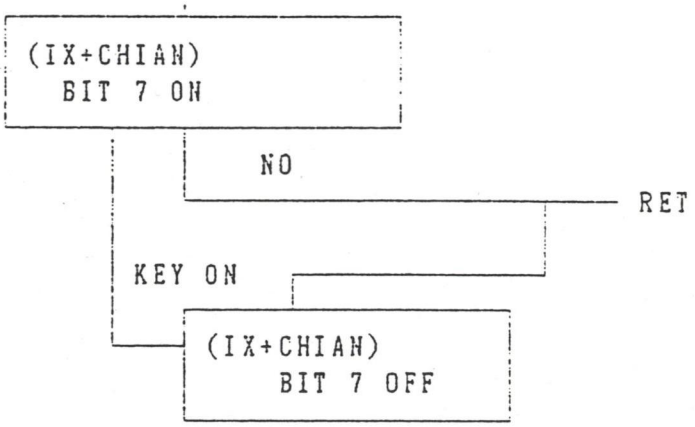

TB25CH?
YES
NO
TB26CH
YES
NO
FLAG BIT 7 ON
NO
RET
FEXCUTR
FLAG BIT 7 ON
RET
YES
NO
FLAG BIT 7ON
FEXCUTR1
YES
RET
FEXCUT

FEXCUTR:

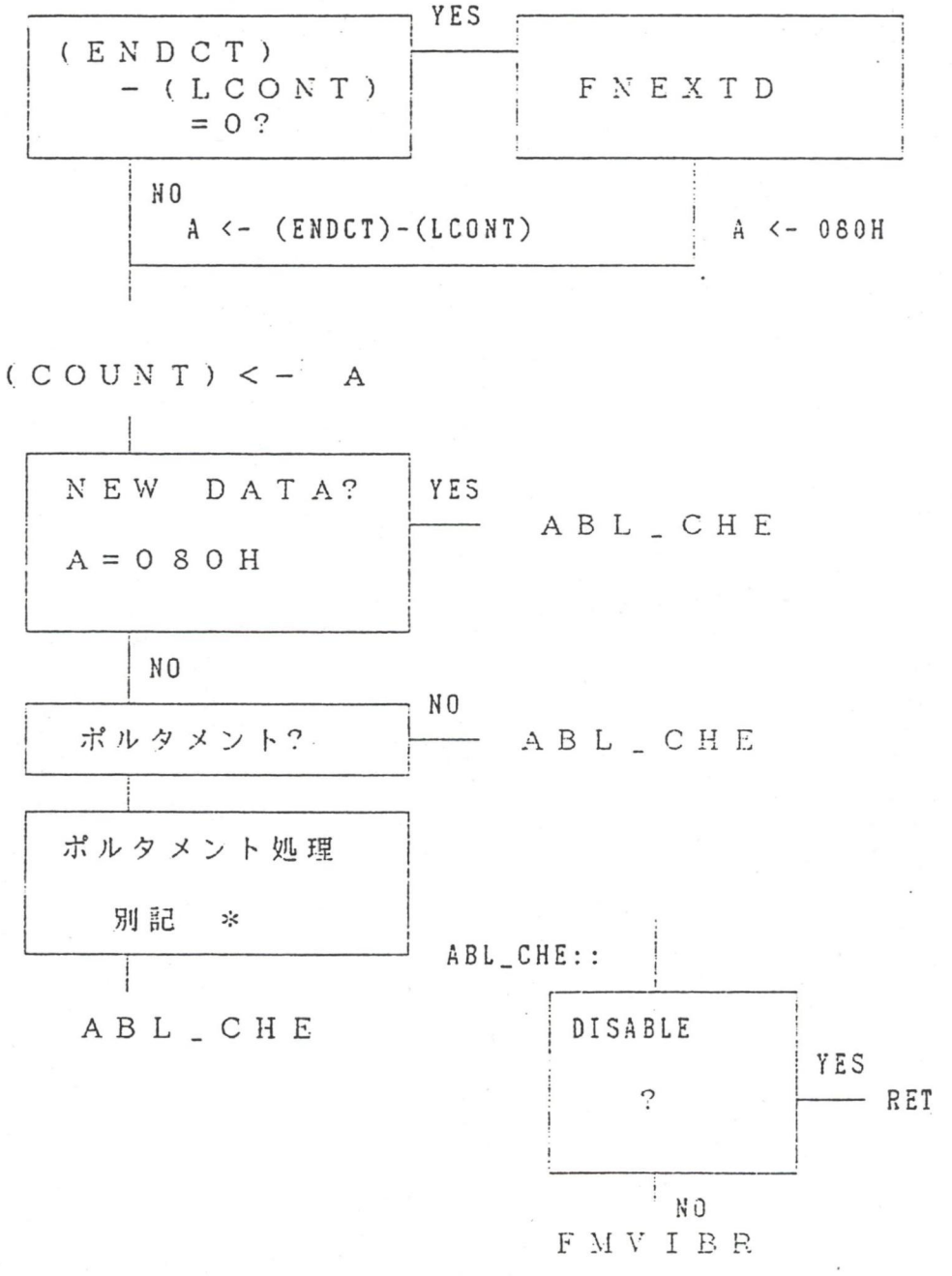

(ENDCT)－(LCONT)
=0 ?
NO
RET
YES
新しいDATA の
読み込み
7FH＜DATA＜0E0H
?
音長 NLENG2 へ
NO
ｺﾏﾝﾄﾞ FMCOMDR へ
YES
BIT 3 ON
(ONONE) <- 80H
BIT 2 ON
(ONTWO) <- 80H
(IX+FREQ) <- DATA
NEXTLT

FEXCUTR1:

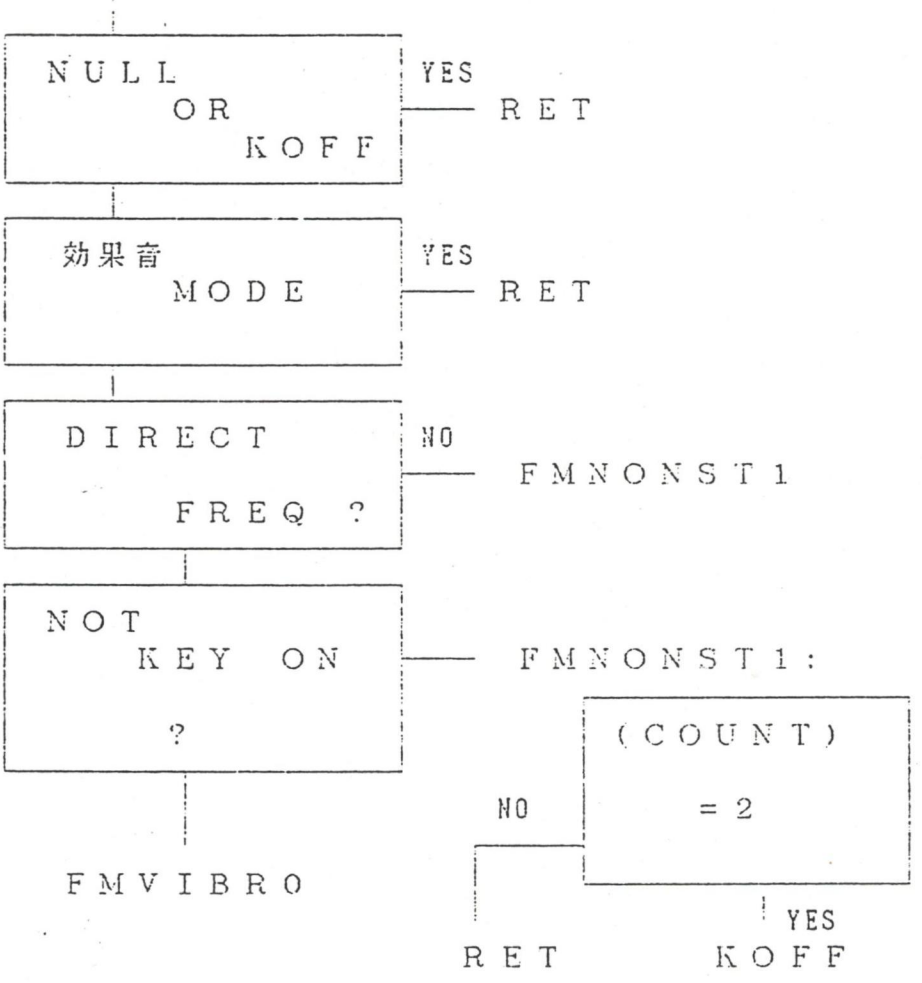

CLRDR2::
(ENDCT2)-(LCONT2) = 0?
YES
(LCONT2)<- 00H
(FREQA2)<- 00H
(FREQB2)<- 00H
(ONTWO) <- 00H
NO
RET
ｽﾛｯﾄ3、4 CHIP に書き込む FREQ を(FERQA2) (FREQB2)にSET
(IX+CHIAN) BIT 7 OFF
2回目以降
(FREQA2)<-(DELTA2)+(FREQA2)
(FREQB2)<-(DELTB2)+(FREQB2)
スロット 1 2も
上記と同様な処理を実行
(ONONE)ON
(ONONE)(ONTWO)ON
(ONTWO)ON
SLOT1,2 KEY ON
SLOT1,2,3,4 KEY ON
SLOT3,4 KEY ON


(IX+CHIAN)
BIT 7 ON
NO
RET
KEY ON
(IX+CHIAN)
BIT 7 OFF

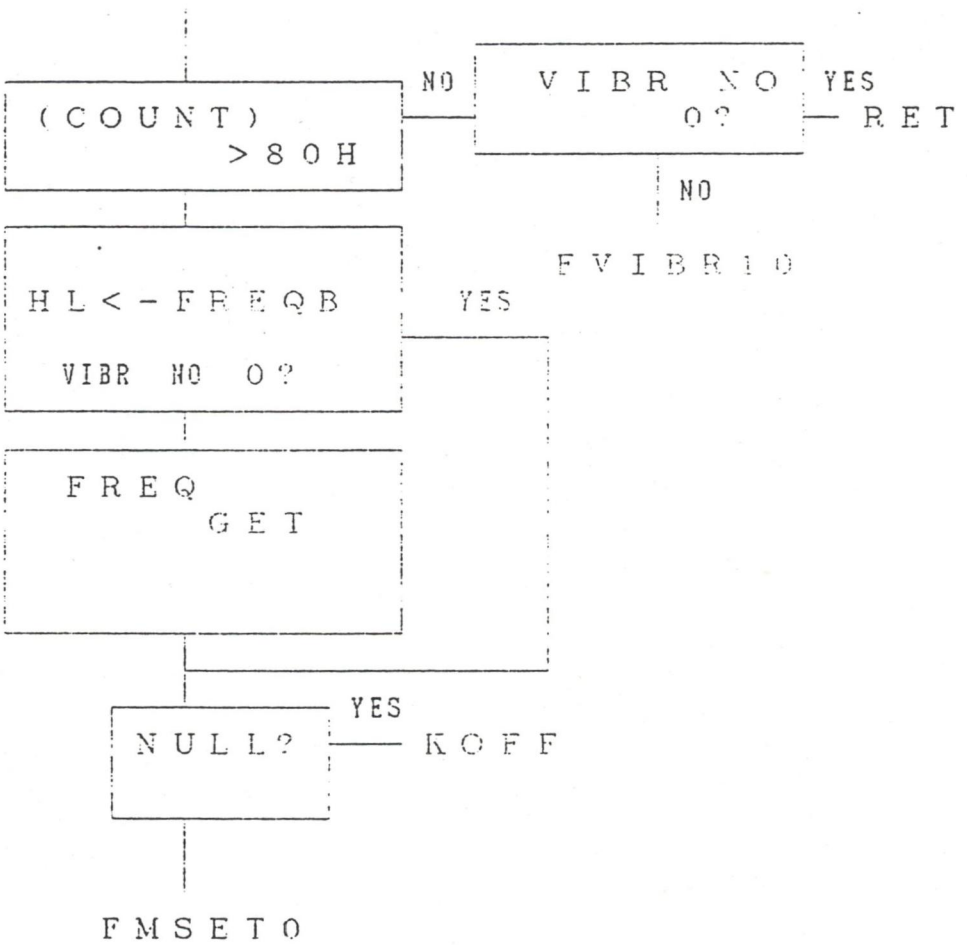
FEXCUT:

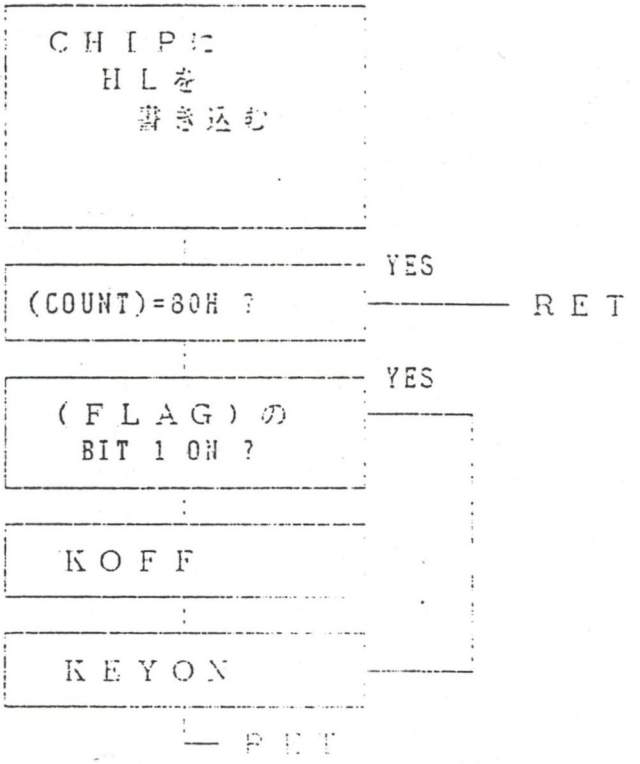

(ENDCT)
-(LCONT)
=0?
YES
FNEXTD
NO
A <- (ENDCT)-(LCONT)
A <- 080H
(COUNT) <- A
NEW DATA?
A=080H
YES
ABL_CHE
NO
ポルタメント?
NO
ABL_CHE
ポルタメント処理
別記 *
ABL_CHE
ABL_CHE::
DISABLE
?
YES
RET
NO
FMVIBR

FMVIBR:


NULL OR KOFF
YES
RET
効果音 MODE
YES
RET
FMNONST:
DIRECT FREQ ?
NO
FMNONST1
NOT KEY ON ?
FMNONST1:
(COUNT) = 2
NO
YES
FMVIBRO
RET
KOFF

FMVIBRO:


(COUNT)
>80H
NO
VIBR NO
0?
YES
RET
NO
FVIBR10
HL<-FREQB
VIBR NO 0?
YES
FREQ
GET
FMSET:
NULL?
YES
KOFF
FMSET0


CHIPに
HLを
書き込む
(COUNT)=80H ?
YES
RET
(FLAG)の
BIT 1 ON ?
YES
KOFF
KEYON
RET