2.7 プログラム・データメモリ（ROM）

○プログラム・データメモリ（ROM）は16ビット出力構成になっており（1Word＝16bit），1chip型では64Kbit（4KWord）までのマスクプログラマブルのROMを内蔵できます。
○EVACHIP，ROM外付け型では1Mbit（64KWord）までのROMを外付けできます。
○プログラム・データメモリ（ROM）には音素データ，エンベロープデータ，ピッチデータ，時間長データ，連結波形数データ，ピークデータ，時間レートデータ，順序等の音声データテーブル，音声合成ソフト，順序解読プログラム及び他の制御変換ソフトがプログラムできます。
○プログラム・データメモリ（ROM）はプログラムカウンタ（PC）によりアドレスされます。
○サウンドシンセサイザー自身をテストする為に0番地から5番地までは次の固定コードを必らずROMに入れて下さい。従ってユーザーエリア

アドレス	コード	オペレーション
0番地	0000	NOP
1番地	0401	A←PA
2番地	0402	A←PB
3番地	FFFF	ダミーコード
4番地	0004	PB←A
5番地	0000	NOP

Pは6番地以降になります。

○プログラムデータメモリ（ROM）中に次の7ロケーションが特別に割り当てられています。

0番地----リセット入力（AC端子を"L"）を加えると，プログラム開始番地として0番地が割当てられます。

20H番地----トーン割込み要求信号で割込みがかかり，分周比レジスタNの内容が08H～0FHの時では，無条件に20H番地からのプログラムを実行します。

24H番地----トーン割込み要求信号で割込みがかかり，分周比レジスタNの内容が10H～1FHの時では，無条件に24H番地からのプログラムを実行します。

28H番地----トーン割込み要求信号で割込みがかかり，分周比レジスタNの内容が20H～3FHの時では，無条件に28H番地からのプログラムを実行します。

2CH番地----トーン割込み要求信号で割込みがかかり，分周比レジスタNの内容が40H～FFHの時では，無条件に2CH番地からのプログラムを実行します。

48H番地----ノイズ割込み要求信号で割込みがかかると，無条件に48H番地からのプログラムを実行します。

60H番地---外部割込み要求信号で割込みがかかると，無条件に60H番地からのプログラムを実行します．

80H番地---タイム割込み要求信号で割込みがかかると，無条件に80H番地からのプログラムを実行します．

○1chip 内蔵型のものはプログラム・データメモリ（ROM）は次のようなアドレス範囲になります

ROM内蔵型	アドレス範囲		
8K bit 内蔵	000H ～1FFFH	0.5K Word	全て0ページ内
16K bit 内蔵	000H ～3FFFH	1K Word	
32K bit 内蔵	000H ～7FFFH	2K Word	
48K bit 内蔵	000H ～BFFFH	3K Word	
64K bit 内蔵	000H ～FFFFH	4K Word	

○EVACHIP又はROM外付型のものはプログラム・データメモリ(外付けROM)は次のような アドレス範囲になります．

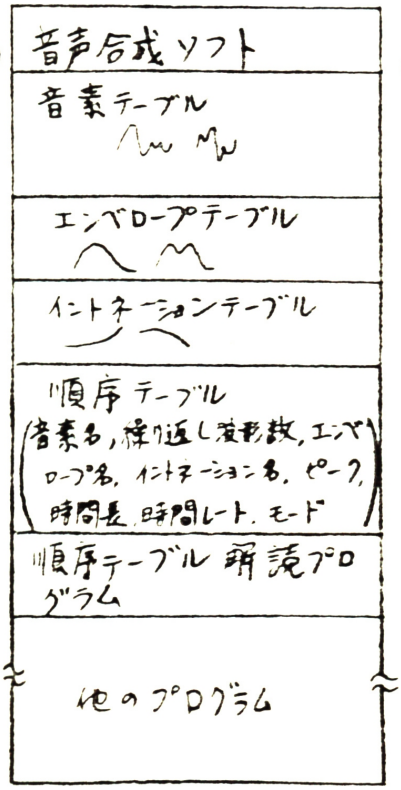
000000H～ FFFFFFH （4KWord × 16ページ）

○1ページは4096アドレス構成です．

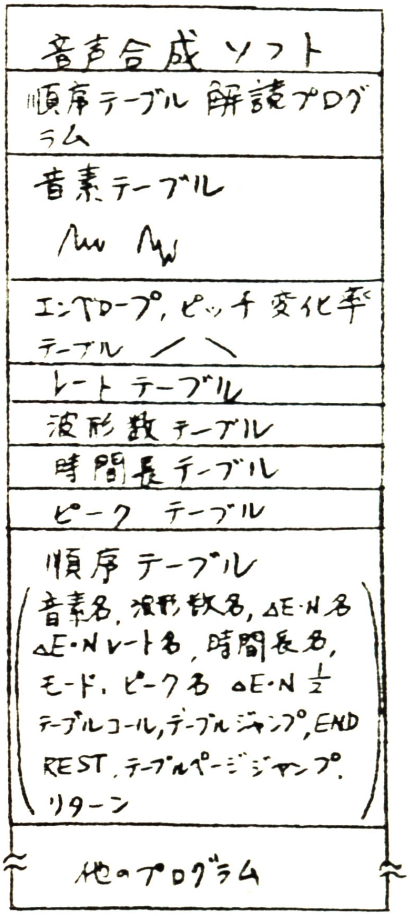
○プログラム・データメモリ（ROM）の全ての アドレス領域のテーブルデータをテーブル参照命令で呼び出すことができます．

○音声合成用ROM構成例

(例1)



(例2)



2.8 データ メモリ（RAM）

○データメモリとして64バイト（64×8ビット＝512ビット）のRAMを内蔵しており，

○データメモリの64バイトのうち 32バイトは直接ROMで指定することができ レジスタとして使用できます。

○コール命令及び割込み時の次の実行番地のスタックはスタックポインタ（SP）で指定されたアドレスに12ビットないし16ビットのプログラムカウンタのデータが1マシンサイクル中に同時に書き込まれます。
スタックポインタによる指定は 8ワード（16バイト）です。
○リターン命令，割込みリターン命令実行時にはスタックポインタで指定された12ないし16ビットのデータがプログラムカウンタに1マシンサイクル中に戻されます。

○テーブル参照命令実行の1マシンサイクル目では ROMで指定されたRAMアドレスの16ビットのデータがプログラムカウンタにラッチされます。

○ MVI命令，MOV命令，演算命令によるRAMからのリード及びRAMへのライトは8ビット単位で行なわれますが，上で示した命令の実行時には16ビット単位で データのリード・ライトがRAMに対して行なわれます。

○ROMによる直接指定時とスタックポインタ指定時以外はデータポインタ（H）によりRAM領域の全てがアドレスされます

○データメモリの内部は次のような構成になっています

(1) 0～31番地（0～1FH番地）

直接ROMにより指定でき ワーキングレジスタ0～1F（R0～R1F）として直接アドレッシングが可能です。
テーブル参照命令の場合にも 直接ROMによって指定され偶数アドレスのデータが下位バイト，奇数アドレスのデータが上位バイトになってROMへの16ビットの アドレス情報となり，1ビットライトシフトされて，その上位15ビットがプログラムカウンタ PCのPC0～PCFにラッチされてROMのアドレスとなり，下位1ビットがそのROMのアドレスの16ビットのデータのうち，下位バイトか上位バイトかを指定するビットになります。つまりテーブル参照アドレスをR0～R1Fに2バイト1組となり 最大16組ストアできます。

(2) 32～47番地（20H～2FH番地）

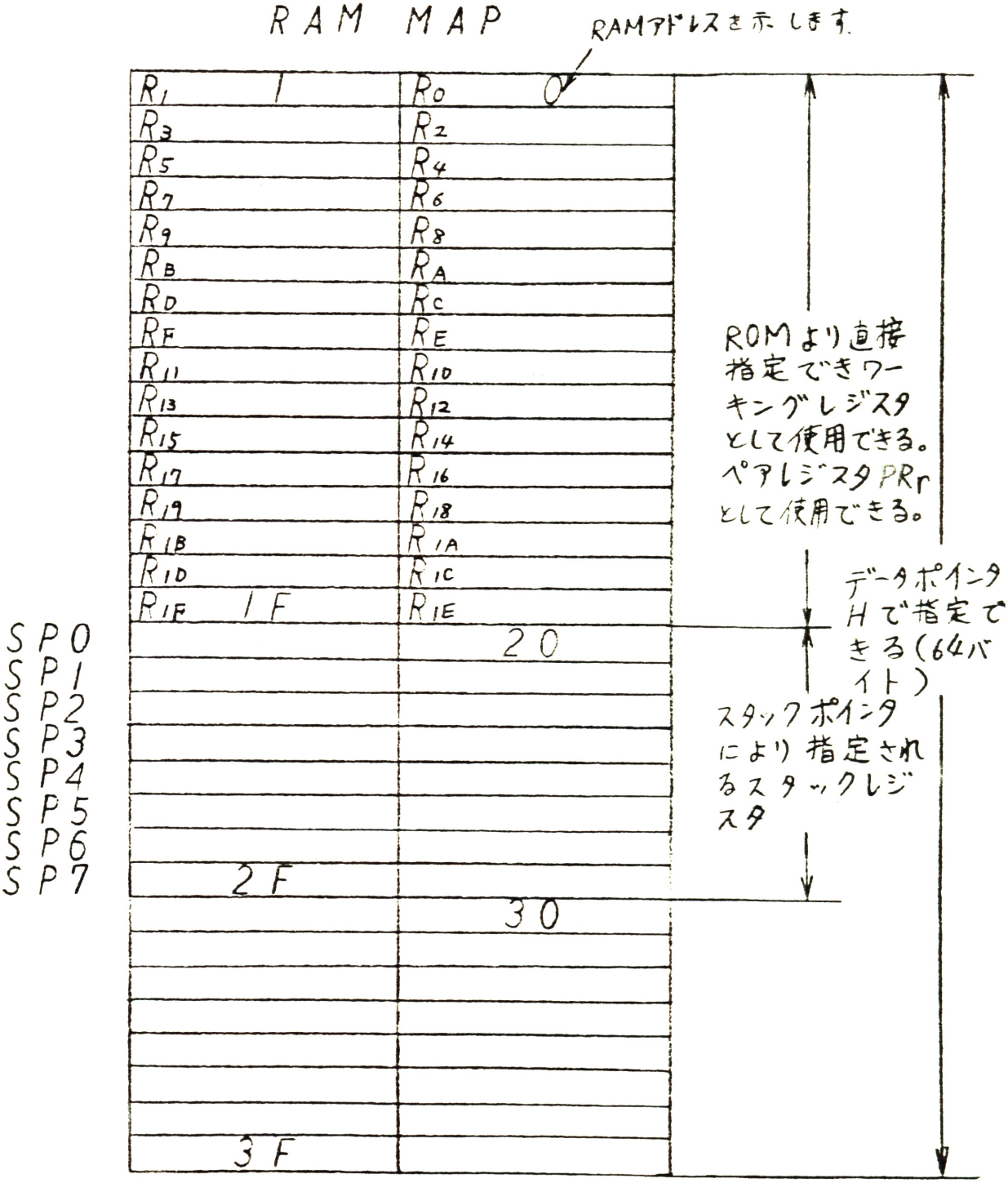
スタックポインタで指定される8レベルのスタックとして使用します
2バイト1組で1レベルのスタックレジスタになっていてスタックポインタ（SP）が0の場合はRAMアドレスは20H番地と21H番地がアドレスされます。

スタックに用いないレベルは 通常のRAMとして使用でき，データポインタ（H）によりアドレスされます．
ROMによる直接指定はありません．

(3) 48～63番地（30H～3FH 番地）

データポインタのみにより指定される通常のRAM領域です．

RAM MAP



○尚，外付けRAM領域はありませんので，64バイト以上のRAMを必要であれば 8155等をポートに接続して，ソフトでポートを介してアクセスして下さい。

2.9 スタック・レジスタ

○ データメモリの20H～2FH番地が8レベルのスタックレジスタとなり，2バイトを1組で1レベルになっています。
スタックレジスタはスタックポインタ(SP)によりアドレスされます。
割込みが生じたり，サブルーチンコール（CALL命令の実行）やテーブル参照コール命令を実行すると，割込みやコールからの戻り番地をスタックポインタで指定されたスタックレジスタに書き込み，スタックポインタはインクリメント（+1）されます。このようにしてスタックレジスタは0より7まで順次スタックされます。
割込みの場合は
プログラムカウンタそのままのデータがスタックされ，コールの場合はプログラムカウンタがインクリメント（+1）されてスタックされます。

○ ROM内蔵1chip型のものではスタックレジスタの上位4ビットには0がストアされ下位12ビットに $PC_{11\sim0}$ ないし $PC_{11\sim0}+1$ がスタックされます

SP	0 0 0 0	$PC_{11\sim8}$	$PC_{7\sim0}$
1			
2			
3			
4			
5			
6			
7			

○ ROM外付型又はEVACHIPのものではスタックレジスタにはプログラムカウンタ $PC_{15\sim0}$ ないし $PC_{15\sim0}+1$ がスタックされます

SP	$PC_{15\sim8}$	$PC_{7\sim0}$
1		
2		
3		
4		
5		
6		
7		

○ 割込みリターン命令またはリターン命令を実行するとスタックポインタはディクリメント（−1）され，スタックポインタで指定されたスタックレジスタのデータがプログラムカウンタに戻ります。

○ 割込みとサブルーチンコールの合計が9レベルになるとスタックポインタは7から0へオーバーフローしてしまうので，0レベルに書き込まれていたデータが消えてしまいます。
○ スタックレジスタに用いないレジスタは通常のRAMとして使用できます。

2.10 インストラクションデコーダー（I.D）

○インストラクションデコーダーはプログラム・データメモリ（ROM）からの16ビットのオペレーションコード（OPコード）をデコードして各機能ブロックに対しマイクロオーダーを発生し命令を実行させます.

2.11 演算部

○演算部は次のブロックにより構成されています.

ALU
アキュムレータ（A）
被乗数レジスタ（Xレジスタ）
乗数レジスタ（Yレジスタ）
アキュムレータ退避レジスタ（A'レジスタ）

2.11.1 ALU

○ALUは8ビットの並列算術論理演算回路で，データバスより2つのテンポラリ・ラッチTEMP1とTEMP2を通してA入力とB入力にデータが入力され，演算結果はS出力からデータバスへ送り出されます.

○ALUで実行できる機能は次のようなものがあります.

キャリー判定無 加算
キャリー判定有 加算
ボロー判定無 減算
ボロー判定有 減算
ビットテスト無 論理積
ビットテスト有 論理積
論理和
排他的論理和

ローテート・ライト
ローテート・レフト
インクリメント
デクリメント

キャリー ジャッジ
キャリーバージャッジ
ボロー ジャッジ
ボローバージャッジ
イコール ジャッジ
イコールバージャッジ
ビットテスト

＊乗算用加算

＊モード切替による64進/32進加算とキャリー判定

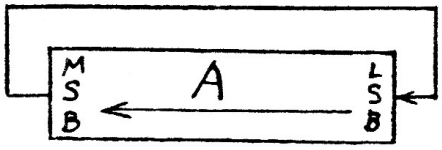
＊トーン・ノイズ混ぜ合わせ用加減算

＊印の機能は本システム独特のもので，音声合成に必要な機能です

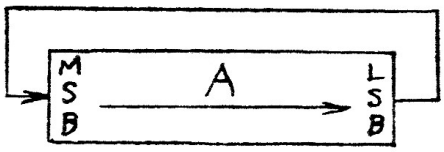
2.11.3 アキュムレータ（A）

○アキュムレータは8ビット構成のレジスタで，ALUへの入力，ALU演算結果のストア，PAポート，PBポート，RAMとのデータのやり取り，分周比レジスタ(Nレジスタ)，D/Aコンバータラッチ，被乗数レジスタ（Xレジスタ）へのデータ転送，イミディエイトデータのストア，テーブル参照命令でのテーブルデータのストアに使用します．

○アキュムレータだけが レフト ローテート，ライトローテートが可能となっています．



レフト ローテート



ライト ローテート

○アキュムレータのデータは割込みが生じると アキュムレータ退避レジスタ(A')にストアされ,割込みリターン命令で アキュムレータ退避レジスタ(A')より アキュムレータにデータが戻ります．

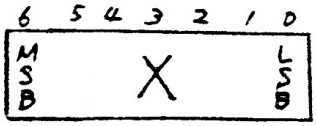
2.11.4 アキュムレータ退避レジスタ（A'）

○アキュムレータ退避レジスタ（A'）は 割込み発生時にアキュムレータのデータをストアし，割込みリターン命令で アキュムレータにデータを転送する8ビットのレジスタです．

2.11.5 被乗数レジスタ（X）

○被乗数レジスタ（X）は7ビット構成のレジスタで テーブル参照命令での正規化された音声素片データの絶対値のストア，7ビットの擬似ランダムジェネレータ（RG ）のデータのストア，アキュムレータのデータのストアに使用され，乗算命令における被乗数レジスタになっています．

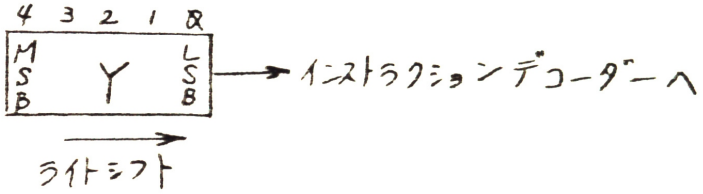
○乗算命令によりアキュムレータのデータと加算され 1ビットライトシフトされてアキュムレータにストアされます



2.11.6　乗数レジスタ（Y）

○乗数レジスタ（Y）は5ビット構成のレジスタで，テーブル参照命令でエンベロープデータのストア，データメモリ（RAM）からのデータのストアに使用し，乗算命令における乗数レジスタになっています

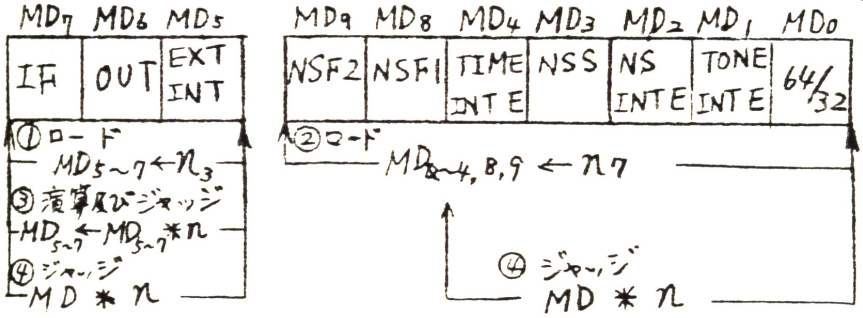
○乗算命令により　乗数レジスタのＬＳＢを判定して，アキュムレータをライトシフトするか，アキュムレータとＸレジスタとを加算させるかを決定し，Ｙレジスタも1ビットライトシフトします。



2.12　モードレジスタ（MODE）

○モードレジスタ（ＭＯＤＥ）は１０ビットのレジスタで，３ビットヌは7ビットのイミディエイトデータのロードと，上位3ビットの演算結果のストア，8ビットの判定ができます。またリセット入力ACが"0"になっているリセット状態ではＩＦは"1"，ＯＵＴは"0"に設定される。
MD₀～MD₅，MD₈，MD₉も"0"に設定されます。

モードレジスタ操作は次のように分かれます。



(注)エバチップで直接モニターできる範囲はMD₀～MD₇であり，MD₈とMD₉は見ることができません。

2.12.1　IF F/F，OUT F/F

IF及びOUTでPAポート，PBポートのインターフェースモードを下の表で示すように4種類のモードを決定します

IF	OUT	PAポート機能	PBポート機能	使用例
X	X	OUT	IN	PAよりスキャン信号出力 PBにキーマトリクス入力
X	1	OUT	OUT	PAよりセグメント出力 PBよりディジット出力 でディスプレイ
1	X	IN/OUT (DATA BUS)	PB7～3－IN，PB2～0－OUT (CS，WR，RD入力，READY出力)	他のCPU→SSコントロール
1	1	IN/OUT (DATA BUS)	OUT (RD，WR，ALE他出力)	SS→他の周辺デバイスコントロール

AC="L"でセットされる

2.12.2 EXT INT E F/F

外部割込み端子 $\overline{\text{EXT INT}}$ からの外部割込みをイネーブルにするフリップフロップです。"1"だとイネーブル "0"だとディスイネーブルになります。

2.12.3 TIME INT E F/F

TIME割込みをイネーブルにするフリップフロップで "1"だとイネーブル，"0"だとディスイネーブルになります

2.12.4 NOISE INT E F/F

NOISE割込みをイネーブルにするフリップフロップで，"1"だとイネーブル，"0"だとディスイネーブルになります。

2.12.5 TONE INT E F/F

TONE割込みをイネーブルにするフリップフロップで，"1"だとイネーブル，"0"だと ディスイネーブルになります

2.12.6 64/32 F/F

音声素片（トーン）の1波形当りの時間軸分割数を64分割モードまたは32分割モードにするかを切り替えるフリップフロップです。64/32 F/Fが"1"だと64分割モードになり64進演算を指定し，64/32 F/Fが"0"だと32分割モードになり32進演算を指定し，それぞれALUの6ビット目のキャリーか5ビット目のキャリーを判定するようになります。 特殊命令の時だけ選択されます。

64/32 F/F	64/32 進加算命令機能
0	32進加算（5ビット目から6ビット目へのキャリー禁止） 5ビット目のキャリージャッジ
1	64進加算（6ビット目から7ビット目へのキャリー禁止） 6ビット目のキャリージャッジ

2.12.7 NSS F/F

Noise 符号のモードを決定します。

NSS F/F	Noise 符号 (NS)
0	0 で 正(+)になる
1	1マシンサイクル毎に反転する1ビットのバイナリーウンタ (PNC2)の出力が出力され0(正)→1(負)→0(正)と変化

2.12.8 NSF1 F/F, NSF2 F/F

Noise割込み周波数を決定します。

~~NSF1~~ 2 F/F	~~NSF2~~ 1 F/F	NS INTE が"1"の時の NS INT 周波数
0	0	fosc/4096 (fosc=5.185MHz 時 1.266 KHz)
0	1	fosc/2048 (fosc=5.185MHz 時 2.532 KHz)
1	0	fosc/512 (fosc=5.185MHz 時 10.127 KHz)
1	1	fosc/256 (fosc=5.185MHz 時 20.254 KHz)

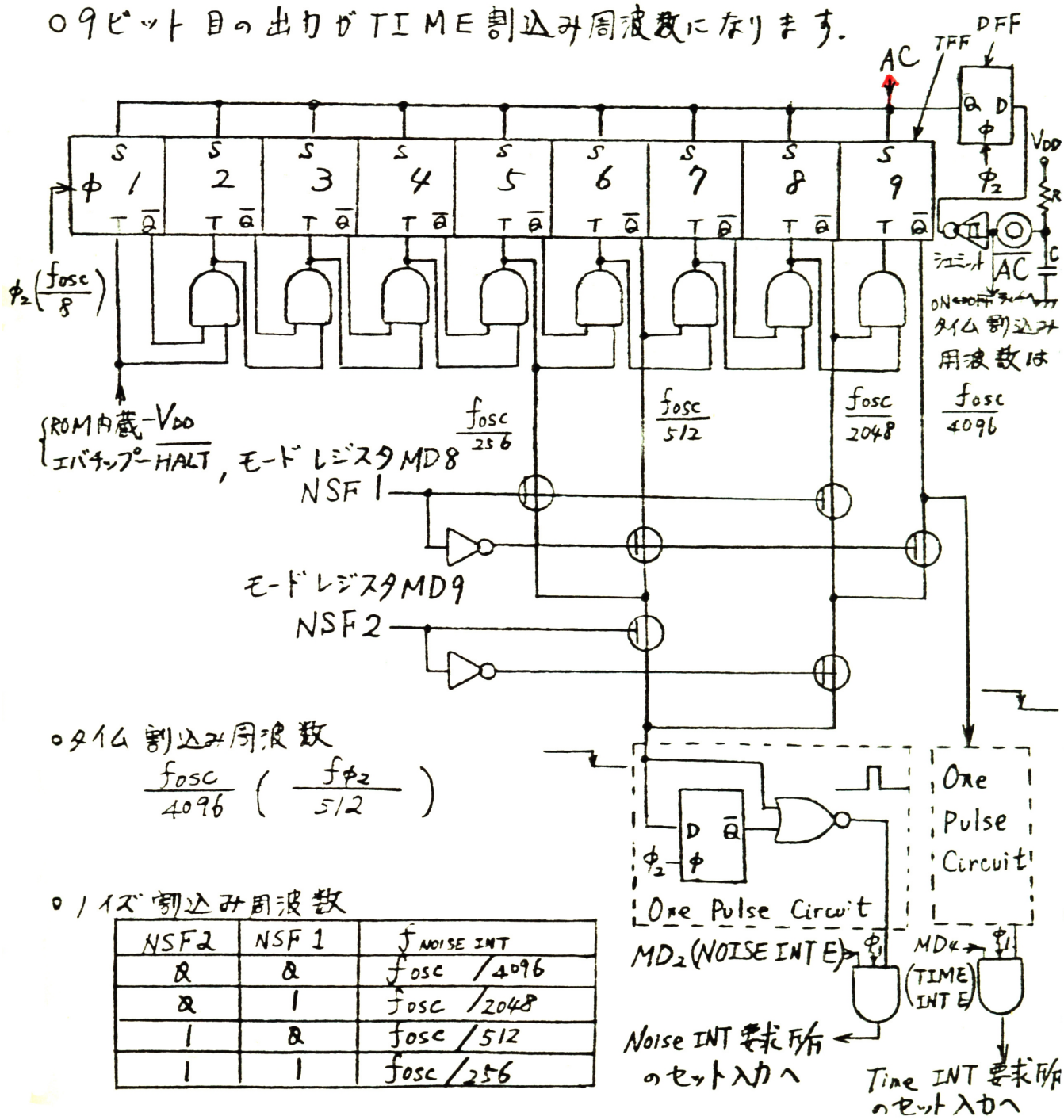
2.13.3 バイナリー カウンター (BC)

○ 1マシンサイクル毎に分周する9ビットの Binary Down Counter で $\overline{AC}$ 入力端子が "L" で 全て "1" にセットされ, $\overline{AC}$ 入力が "H" になった次のマシンサイクルよりダウンカウント(−1)されます。

○ エバチップでは HALT 状態で カウントはストップ状態になっています。

○ 5ビット目, 6ビット目, 8ビット目, 9ビット目の出力がそれぞれ モードレジスタ NSF2, NSF1 により選択されてノイズ割込み周波数になります。

○ 9ビット目の出力が TIME 割込み周波数になります。



○ タイム割込み周波数

$$\frac{fosc}{4096}\left(\frac{f\phi_2}{512}\right)$$

○ ノイズ割込み周波数

NSF2	NSF1	$f_{NOISE\ INT}$
0	0	$fosc/4096$
0	1	$fosc/2048$
1	0	$fosc/512$
1	1	$fosc/256$

2.14 擬似ノイズ発生回路

擬似ノイズ発生回路は次のブロックより構成されています.

- 擬似ランダムジェネレータ (RG)
- ノイズ符号カウンタ (NSC)

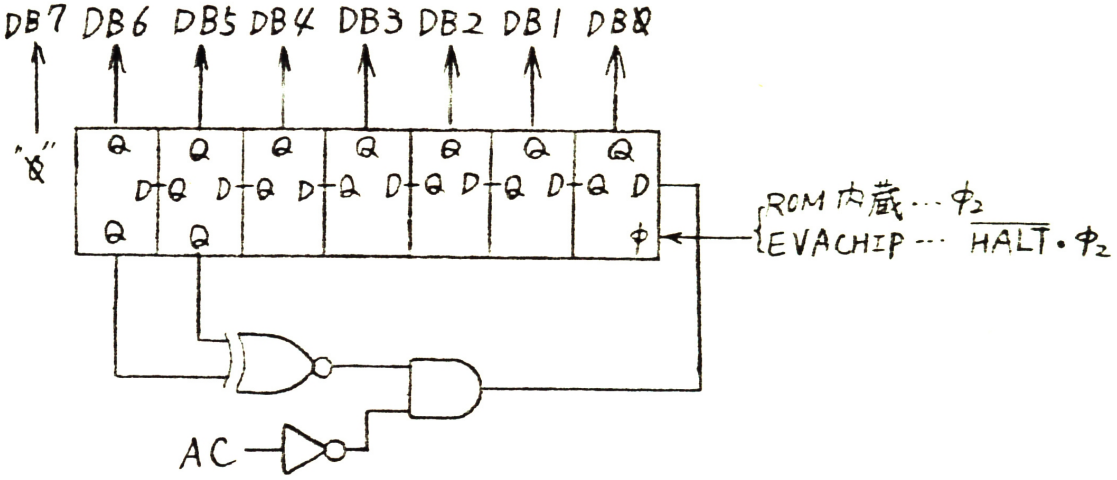
2.14.1 擬似ランダムジェネレータ (RG)

○擬似ランダムジェネレータ(RG)は7ビット構成で,擬似ノイズを作るポリノミナルカウンターです。

○$\overline{AC}$入力を7マシンサイクル以上"L"にして"H"にすると次のように1マシンサイクル毎にカウントします.

00→01→03→07→0F→1F→3F→7E→7D→7B
(リセット状態)
→77→ -------- →20→40→00→ 以後繰り返えし

○$\overline{AC}$入力を8マシンサイクル以内"L"で"H"になると完全にクリアされない状態からカウントします.



○擬似ランダムジェネレータ(RG)の出力は命令により被乗数レジスタ(Xレジスタ)に転送でき7ビットのノイズレベルとしてエンベロープデータ(Yレジスタ)と乗算できます。乗算された結果はアキュムレータにストアされます.

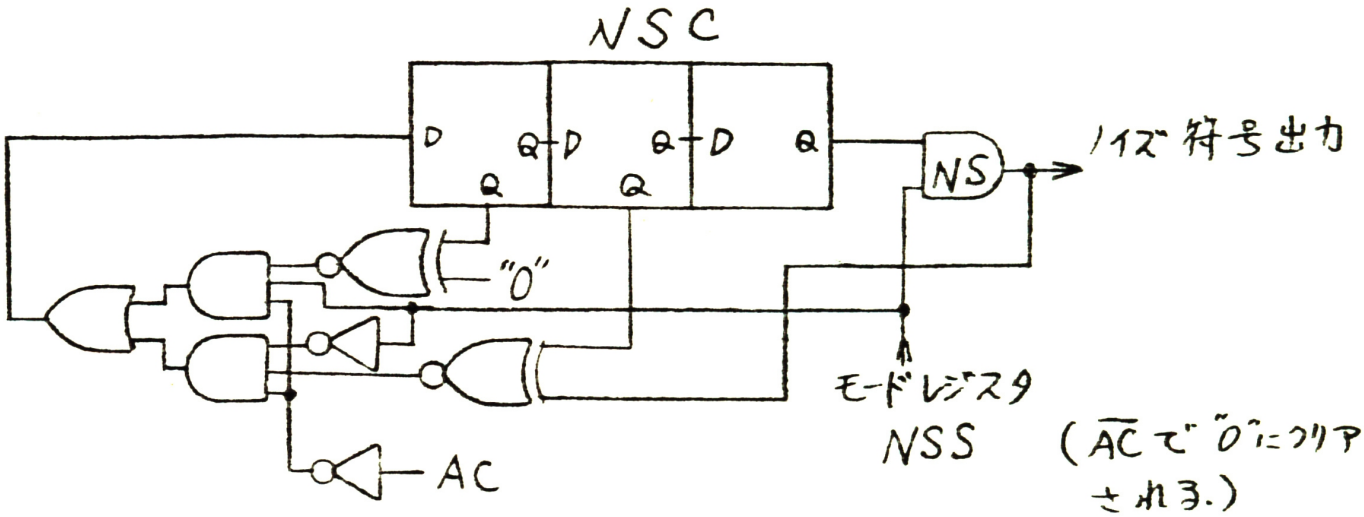
○擬似ランダムジェネレータ(RG)のデータをアキュムレータに転送したい時はYレジスタに01を入れ,乗算命令MUL2を実行すると1ビットライトシフトされてアキュムレータにストアされます.(これを実行する前にはアキュムレータをクリアしておきます)

○ランダム性を増すようにするためにはRAMに擬似ランダムジェネレータのデータを数回(3回位)加算または減算してアキュムレートしたデータを使用すると良いです。

2.14.2 ノイズ符号カウンタ (NSC)

○NSC は3ビットのポリノミナルカウンタで $\overline{AC}$ 入力が 5マシンサイクル以上 "L" でその後 "H" になると 1マシンサイクル毎にカウントをします。

○モードレジスタNSSによってノイズ符号の制御を行ない，NSSが "1" であればバイナリーで $\alpha \rightarrow 1 \rightarrow \alpha \rightarrow 1$ と変化しているNSCの出力がノイズ符号になり，NSSが "0" であればNSCで3ビットでカウントして出力されるのを禁止してノイズ符号出力として "0" (正) を出力します.



モードレジスタ NSS	ノイズ符号出力 NS
α	"α" (正)
1	$\alpha \rightarrow 1 \rightarrow \alpha$ と1マシンサイクル毎に繰り返す (正) (負) (正)

○ノイズ符号が有効になるのはトーンとノイズを混ぜ合わせをするための混合命令実行時だけとなっています。

2.15 トーン割込み信号発生回路

○トーン割込み信号発生回路には次のブロックがあります.

- 分周比レジスタ（Nレジスタ）
- プリセッタブル・ダウンバイナリー・カウンタ
- TONE割込周波数制御回路.

2.15.1 分周比レジスタ（N）

○分周比レジスタ（Nレジスタ）は8ビット構成のレジスタで，アキュムレータより データをセットすることができます.

○Nレジスタのデータにより トーン割込み周波数が決定されます
48ページに分周比Nとトーンサンプリング周波数，トーン周波数の関係を示します.

○プリセッタブル・ダウンバイナリー・カウンターのプリセット値をストアします。

2.15.2 プリセッタブル・ダウン・バイナリー・カウンタ(DOWN CO.)

○プリセッタブル・ダウン・バイナリー・カウンタは8ビット構成で
$\overline{AC}$入力が 3マシンサイクル以上 "L"になっていると00Hにプリアされ "H"になってから 2マシンサイクル以降に 1マシンサイクル毎にバイナリーでカウントダウンされます.

00H→FFH→FEH→--→01H→N→N-1→--→01H→N

分周比がプリセットされる。
リップルキャリーが出力される。

○ダウンカウンターが01HからS00Hになる時に分周比レジスタ（Nレジスタ）の内容Nがプリセットされ，リップルキャリーを出力します. リップルキャリー出力周波数は f_{OSC}/N（$f_{OSC}/8N$）になります.

2.15.3 トーン 割込み周波数制御回路 (TONE)

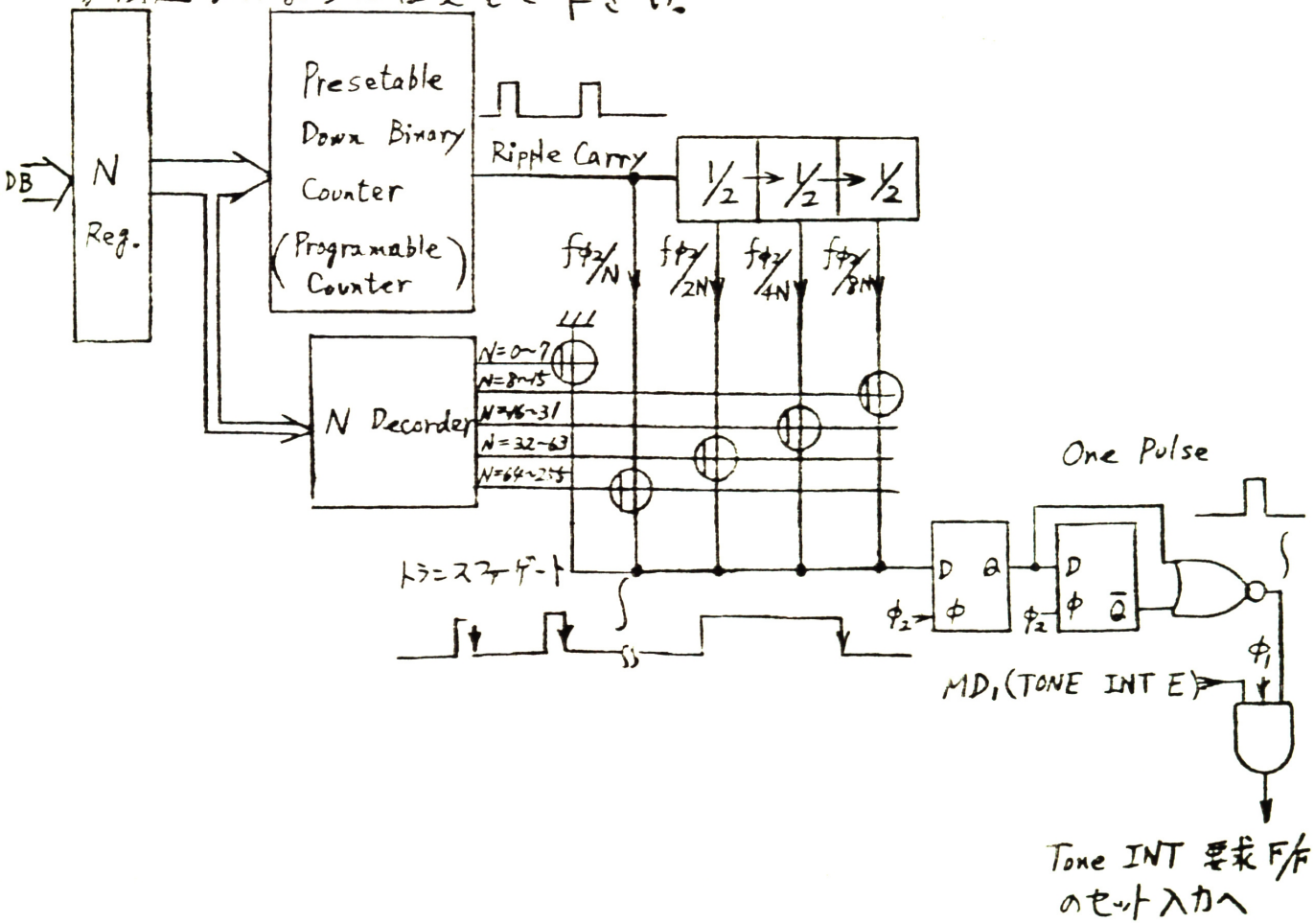
○ 分周比レジスタ N のデータが どの範囲にあるか 5つの範囲に分ける N デコーダがあり，N の範囲によって トーン割込み周波数を制御し，トーン割込み周波数が fosc/512 つまり $f_{\phi 2}/64$ になるよう制御します。

N	トーン割込み周波数
0～7 (0～7)	—
8～15 (08～FH)	fosc/64N ($f_{\phi 2}/8N$)
16～31 (10～1FH)	fosc/32N ($f_{\phi 2}/4N$)
32～63 (20～3FH)	fosc/16N ($f_{\phi 2}/2N$)
64～255 (40～FFH)	fosc/8N ($f_{\phi 2}/N$)

○ トーン割込み周波数が $f_{\phi 2}/64$ でありますので，64マシンサイクル中で 音声合成のための割込み処理が 全部 終了しなければなりません。 もし 64マシンサイクルで終了しない場合は トーン周波数が下がったり，タイム割込み周波数，ノイズ割込み周波数が低下し正確な時間カウントができません。

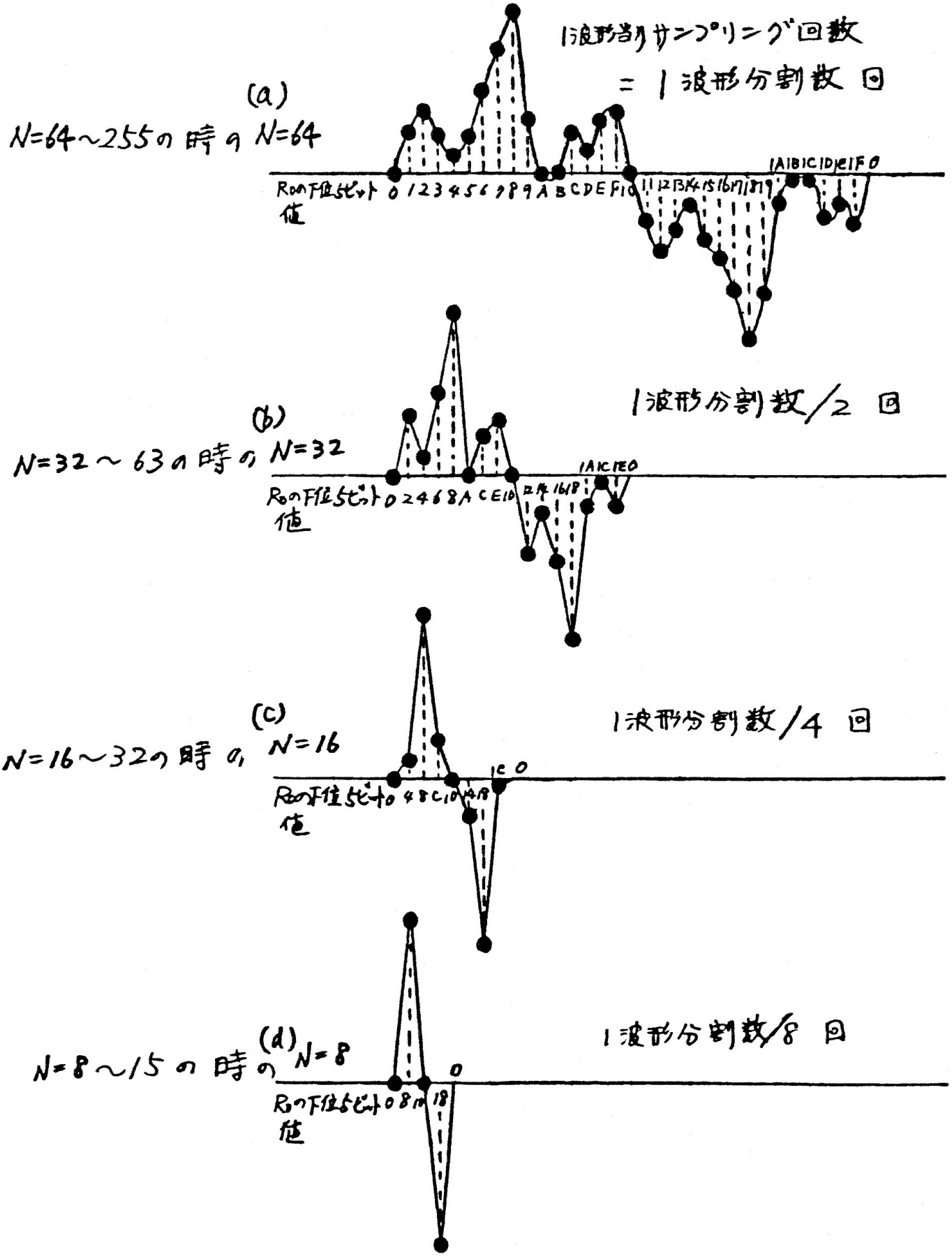
全割込み処理 マシンサイクル ≦ 64
トーン割込み処理マシンサイクル＋タイム割込み処理マシンサイクル＋ノイズ割込み処理マシンサイクル＋3 ≦ 64

○ 音声の場合には素片名変更，エンベロープ名変更等の時だけ64マシンサイクル以上消費する場合は割込み処理サイクルを 140 マシンサイクル位までにして音質が問題ないように注意して下さい。



2.17 1波形当りのサンプリング回数と発生周波数

1波形32分割モードの時のサンプリング出力波形



- OSC　5.185 MHz X'tal or CR 発振
- 内部動作スピード

$$f_\phi = 5.185\ \text{MHz} \div 8 = 648.1\ \text{KHz}$$

- 1 word = 1マシンサイクル
- 演算実行時間 = 1.54 μs
- INT合計マシンサイクル数 $= \dfrac{\text{内部動作周波数}}{\text{最大サンプリング周波数}} = \dfrac{648.1\text{KHz}}{10.13\ \text{KHz}} = 64$

1波形分割数	N	サンプリング周波数 Hz	1波形当りサンプリング回数	高調波	トーン周波数 Hz	最大高調波周波数 Hz
1波形32分割 (ROM 16 word)	8～15	$\frac{648.1}{N} \div 8$ = 10.13 ～ 5.4 KHz	4回	2倍	2.53 K～1.35K	5.06K～2.70K
	16～31	$\frac{648.1}{N} \div 4$ = 10.13 ～ 5.22KHz	8回	4倍	1.26K～653	5.06K～2.61K
	32～63	$\frac{648.1}{N} \div 2$ = 10.13 ～ 5.14KHz	16回	8倍	633 ～ 321	5.06K～2.57K
	64～255	$\frac{648.1}{N} \div 1$ = 10.13～2.54KHz	32回	16倍	316 ～ 79.4	5.06K～1.27K
1波形64分割 (ROM 32 word)	8～15	$\frac{648.1}{N} \div 8$ = 10.13～5.4KHz	8回	4倍	1.26K～653	5.06K～2.70K
	16～31	$\frac{648.1}{N} \div 4$ = 10.13～5.22KHz	16回	8倍	633 ～ 321	5.06K～2.61K
	32～63	$\frac{648.1}{N} \div 2$ = 10.13～5.14KHz	32回	16倍	316 ～ 79.4	5.06K～2.57K
	64～255	$\frac{648.1}{N} \div 1$ = 10.13～2.54KHz	64回	32倍	158 ～ 39.7	5.06K～1.27K

$08 \leqq N \leqq 255\ (FF_H)$, N=01 では カウンタストップ状態

(参考)

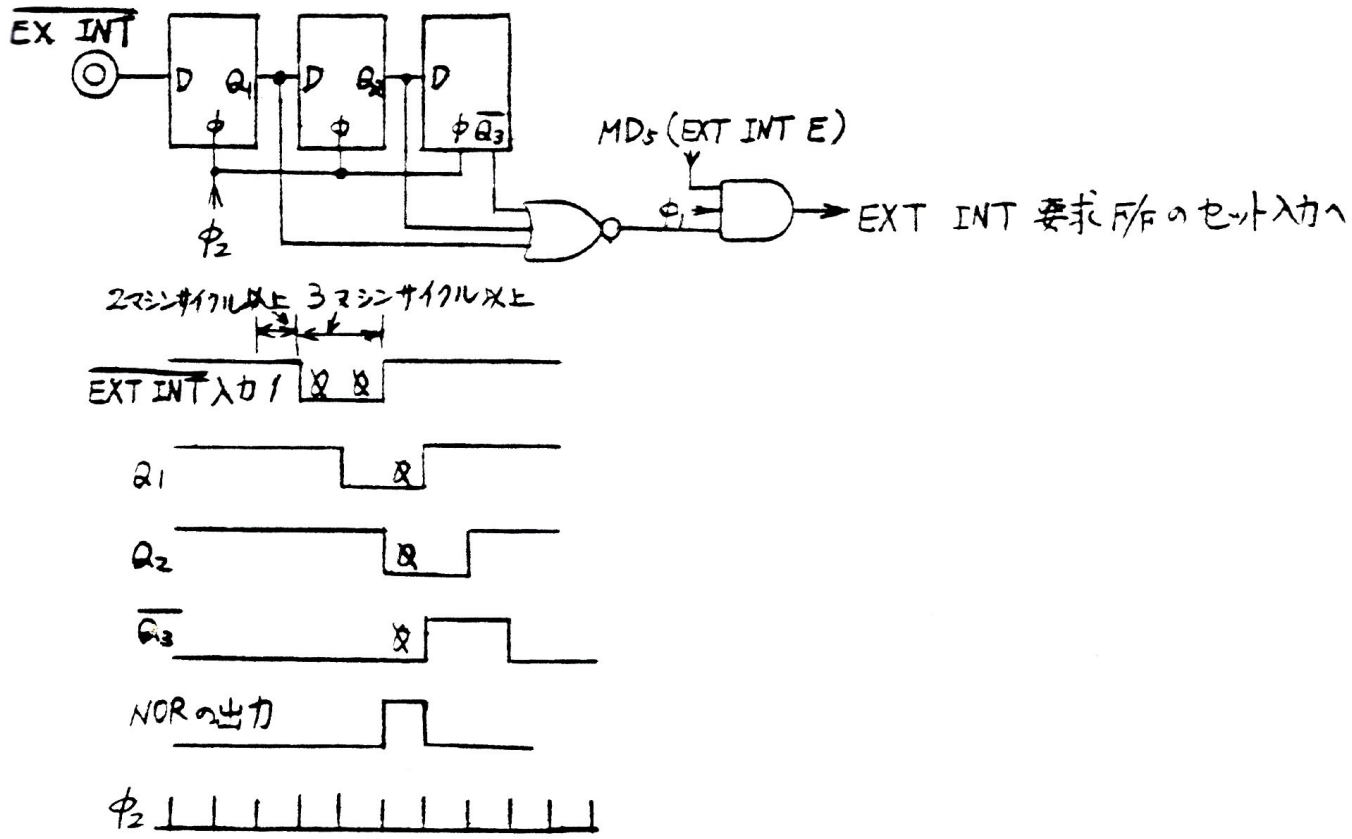
「イ」の音は30倍の高調波まで必要

バイオリン, クラリネット, ピアノ は 30倍の高調波まで必要 (ピアノ 500Hzで8倍, 1000Hzで4倍)

フルート オーボエは 15倍の高調波まで必要

2.16 外部割込み信号発生回路

$\overline{\text{EX INT}}$ 端子は外部割込み要求入力端子で，"1"が少なくとも2マシンサイクル連続した後で"0"が少なくとも3マシンサイクル連続していれば外部割込み要求F/Fへワンパルス信号を発生し，もしモードレジスタのMD5（EXT INT E）が"1"になっておれば外部割込み要求フリップフロップをセットします。



2.17 割込み制御回路

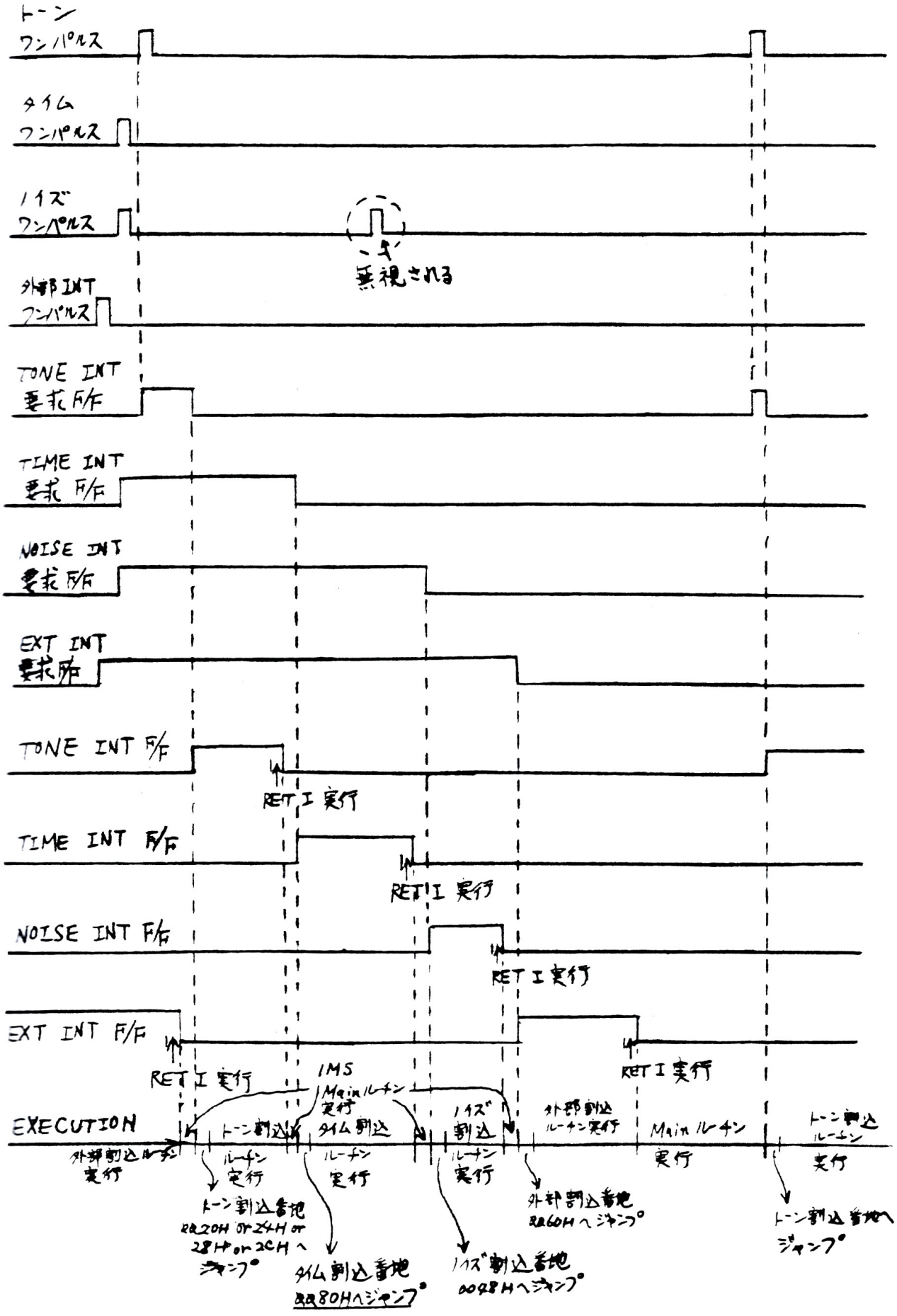
2.17.1 割込み優先順位

各割込み要求フリップフロップが2つ以上同時にセットされていて，割込み中でなくなっている状態の時に，次の割込み優先順位で割込みがかかり，その割込みがかかった割込要求フリップフロップがリセットされます。また単独で割込み要求がある場合には割込み中でなくなっている状態の時に割込みがかかります。

割込み優先順位

TONE INT > TIME INT > NOISE INT > EXT INT

2.17.2 割込み動作タイミング



2.17.3 割込み動作

各割込み要求フリップフロップと各割込みフリップフロップ(TONE INT F/F, TIME INT F/F, NOISE INT F/F, EXT INT F/F)はリセット入力 $\overline{AC}$ 端子が"0"でリセットされます。
また各割込み要求フリップフロップはモードレジスタMDの各割込みイネーブルフリップフロップが"1"になっている時だけ各割込み発生回路からのワンパルス信号によってセットされます。

割込みレベルは1レベルしかありませんので既に割込み中になっている時はそのルーチンの最後でのRET I命令により、割込みルーチンからメインルーチンに復帰しないと待機中の次の割込みには入りません。

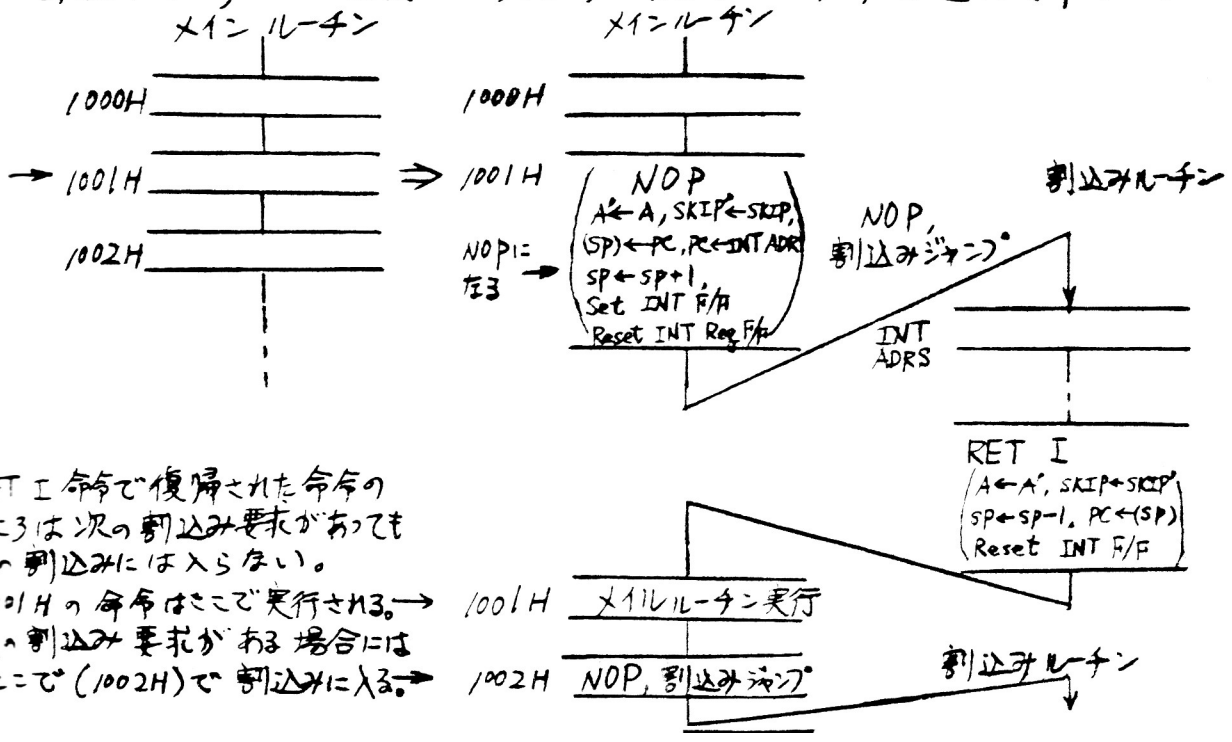
割込み動作は次のようになります。

NOP, A'←A, SKIP'←SKIP,

(SP)←PC, PC←INT ADRS, SP←SP+1

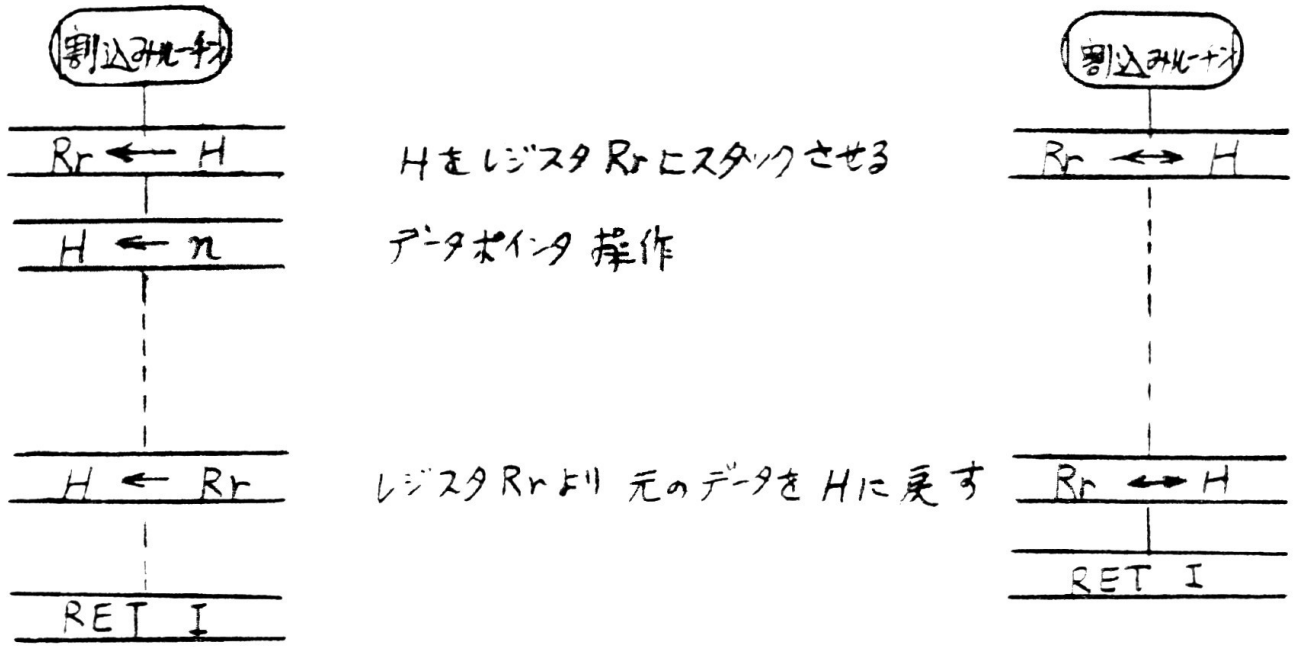
Set INT F/F, Reset INT Request F/F

次に実行すべき命令をNOPにして、アキュムレータの内容をアキュムレータ退避レジスタA'にストアし、スキップフラグフリップフロップの内容をスキップフラグ退避フリップフロップSKIP'にストアし、プログラムカウンタの内容をスタックポインタで指定したデータメモリにストアし、各割込みに従った割込み番地INT ADRSをプログラムカウンタにロードし、スタックポインタをインクリメントします。(データポインタの退避と2マシンサイクル命令記憶F/Fの退避はありません)
従って実行すべき命令のところで割込みがかかると、その命令は実行されずNOPとなり、割込みルーチンが終わってから割込みによってNOPにされた命令から実行されます。
割込みルーチンの最後には必ずRET I命令を置いて下さい。



1001Hで割り込みが発生したときの例

データポインタHの割込み退避レジスタは持っていませんので，もしも割込み中にデータポインタを操作する時は割込みルーチンの中でデータポインタの退避を行なって，データポインタの操作命令を実行し，そしてレジスタRrより元のデータをHに戻して下さい。



2.17.4 割込みアドレスと割込み周波数

割込みの種類	割込み番地	割込み周波数
トーン	0020H (N=08～0FH)	$f_{osc}/64N = f_{\phi 2}/8N$
	0024H (N=10～1FH)	$f_{osc}/32N = f_{\phi 2}/4N$
	0028H (N=20～3FH)	$f_{osc}/16N = f_{\phi 2}/2N$
	002CH (N=40～FFH)	$f_{osc}/8N = f_{\phi 2}/N$
	— (N=00～07H)	—
ノイズ	0048H	$f_{osc}/4096 = f_{\phi 2}/512$ (NSF1=0, NSF2=0)
		$f_{osc}/2048 = f_{\phi 2}/256$ (NSF1=1, NSF2=0)
		$f_{osc}/512 = f_{\phi 2}/64$ (NSF1=0, NSF2=1)
		$f_{osc}/256 = f_{\phi 2}/32$ (NSF1=1, NSF2=1)
タイム	0080H	$f_{osc}/4096 = f_{osc}/512$
外部	0060H	—

2.19 D/Aコンバータ

○ 符号＋8ビットの電流型D/Aコンバータを内蔵していますので，スピーカーをダイレクトドライブをする事ができます。　特に大きい音声出力を出さなければならないものは，　AMPの外付けをして下さい。
○ CH1端子にボリュームを付ける事により音の大きさが可変できます。
○ スピーカーのインピーダンスは　48Ω～64Ωのものを使用して下さい。スピーカーの両端に折り返しノイズをカットするために0.1～10μF位のコンデンサーを必らず接続し，ローパスフィルターとして下さい。
○ OUT DA 命令により，アキュムレータの内容をD/Aコンバータのレベルラッチに転送し，サウンドサインSSの内容をD/Aコンバータの符号ラッチに転送します。またトーン符号TSの内容とサウンドサインSSの内容の排他的論理和をとった出力をレベル反転ラッチに転送します。符号ラッチの出力はD/Aコンバータの符号入力に入力される。
○ レベル反転ラッチの内容が"0"（つまりSSがTSと同符号であって，MIX Rr 命令にてキャリーがでなかった時）の時には　8ビットのレベルラッチの内容がそのままD/Aコンバータの入力になります。
レベル反転ラッチの内容が"1"，つまりSSがTSに対し異符号になっている時（混合命令 MIX Rr 命令にて減算した結果ボローが発生した時）に8ビットのレベルラッチの反転（"1"→"0"，"1"→"0"）データ，　アキュムレータの反転データ $\overline{A}$ がD/Aコンバータに入力されます。

```
MIX  Rr   ( A ← Rr ± A , SS ← TS ⊕ (Bo or Ca)
            if  TS=NS , ADD,  if TS≠NS , SUB )
  :
  :
OUT DA    ( if SS=TS , DA LEVEL ← A  , DA Sign ← SS )
          ( if SS≠TS , DA LEVEL ← Ā  , DA Sign ← SS )
```

TS, NS, SS, DA Sign は"0"で正，"1"で負を表わします。

○ AMPは差動AMP（例 μPC1212C または μPC1350C）型のものを使用してもらいますが，その時も　SOUND1とSOUND2の両端に抵抗（約数10Ω）を接続して　差動出力をとりだし，その両端にコンデンサーにて　フィルター効果をもたせて下さい。
　　コンデンサーの値は 0.1μF～10μF位のよい値を選択して下さい。コンデンサーによって高い折り返しノイズがカットされ音色が良くなります。

○ どうしても単極性のAMPを使用したい時は，D/Aコンバータ出力は符号なしで　80Hを中心に00～FFHまでの値で出力する必要があり，次に示すソフト処理例によって行なって下さい。

単極性出力ソフト例

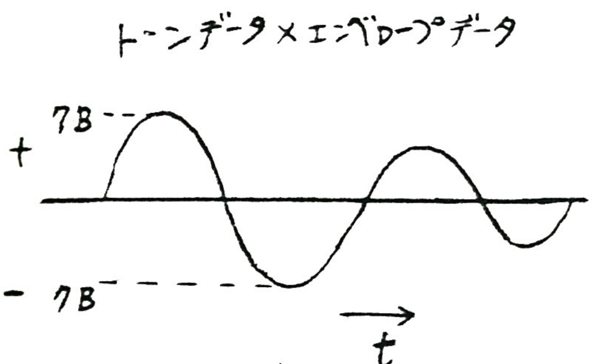
```
TBLØ    A, (R0)      (TSがAのMSBに。)
RAL                  (TSがAのLSBに。)
ANDI    A, Ø1
MOV     RF, A        (TSがRFのLSBに。)
 ┌ - - - - - - - - - - ┐
 TBLØ   X, R0          }  トーンデータとエンベロープデータ
                          の乗算結果→A
 MUL 2
 └ - - - - - - - - - - ┘
```

```
  1 1 1 1 1 1 1 (7FH)
  X   1 1 1 1 1 (1FH)
 ----------------
  1 1 1 1 0 1 1 (7BH)
```

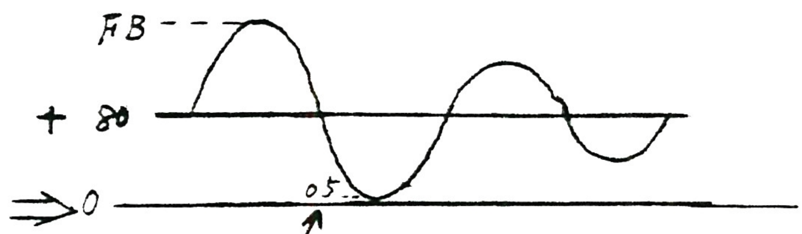
+7Bの時　NS=Øとなるよう、NSS=Øにモード設定してあるとする。

```
MIX      RF          A← RF±A   (TSが正(0)、A←Ø+Aとなり (RF=00)
                                TSが負(1) A←01−Aとなる。(RF=01))
TSBINC   RF, 1       (RF−01 if Ca SKIP)
トーンTS正の時
┌ JMP
│ EORI   A, FF       (A←A⊻FF でAを反転)
└→ADI    A, 80       (A←A+80)
OUT      DA
```

トーンデータ×エンベロープデータ



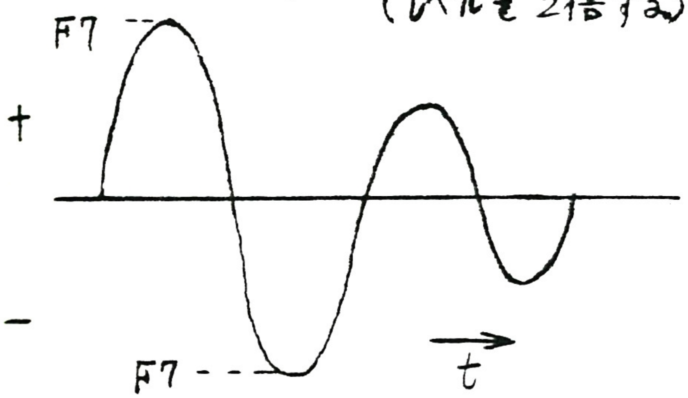
⇒



04のところが05になる。04にするには
(2の補数をとる必要がありもう少しステップとレジスタが必要。)

D/Aの単極性出力

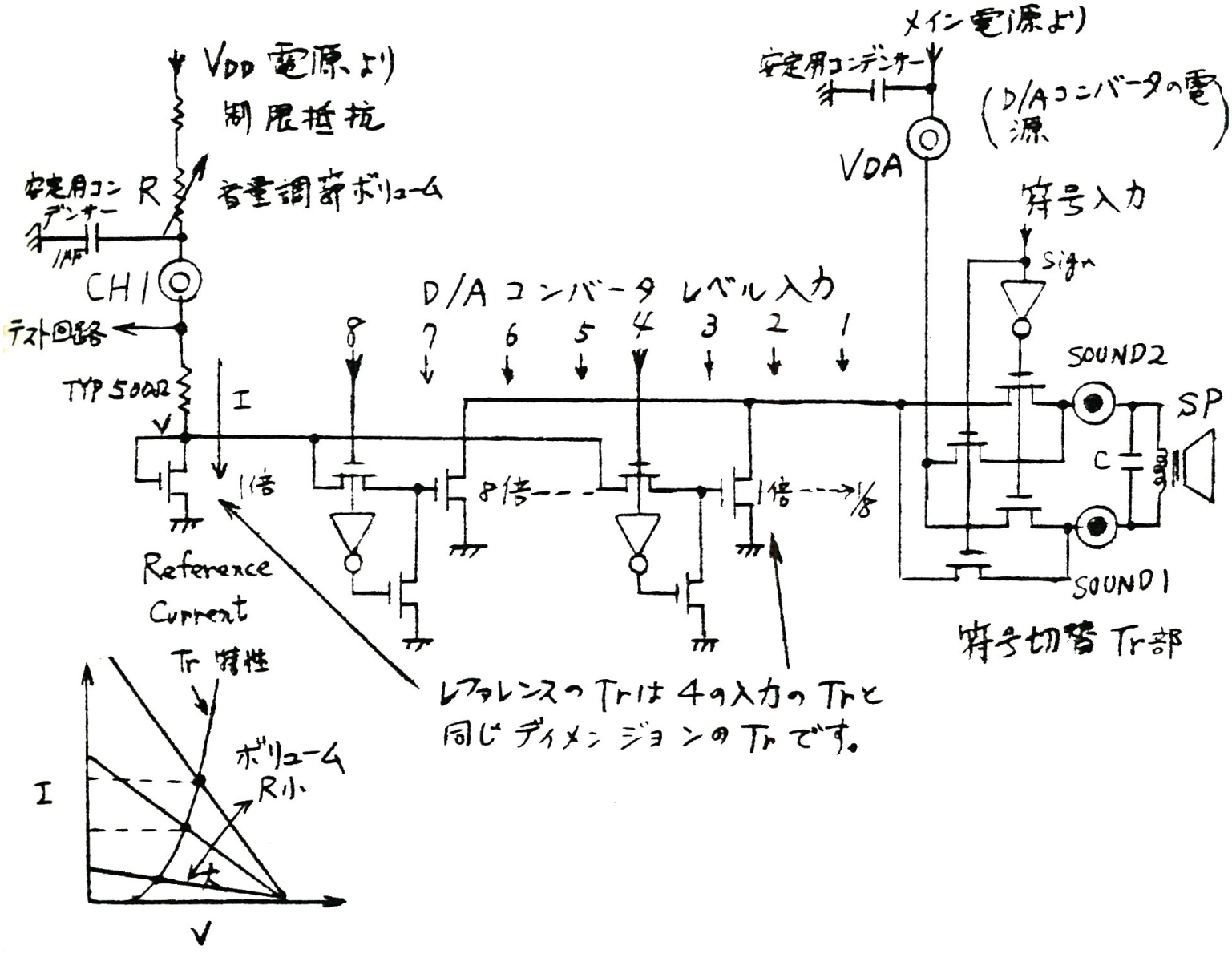
⇓ Aを左回転(RAL)
(レベルを2倍する)



D/Aの両極性出力

○ D/Aコンバータの出力電流はテスト端子兼用ピンのCH1に流れる電流のTYP. 32倍の電流がD/Aコンバータの入力データがALL"1"の時に流れます。(但し，電流が最大値に近いところでは32倍より小さくなります。)

○ Power OFFモード(V_{DD}=0V)の時はD/Aコンバータのデータは ALL"0"になりますので，D/Aコンバータ出力はフローティングになります。



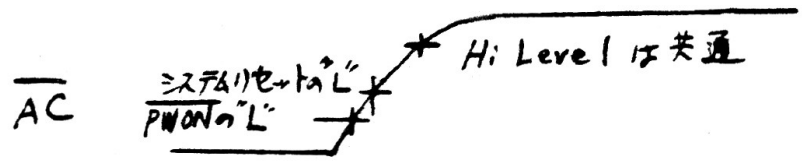
2.20. パワーON・OFF制御回路（ON・OFF）

○ パワーON・OFF制御回路は次のような動作をします。
PWON端子はオープンドレイン出力です。

$\overline{AC}$ 端子 "L"	→	$\overline{PWON}$	ON
$\overline{AC}$ 端子 "H"	→	$\overline{PWON}$	保持
OFF命令	→	$\overline{PWON}$	OFF

○ 回路はその他の命令のOFF命令のところで記述しております。

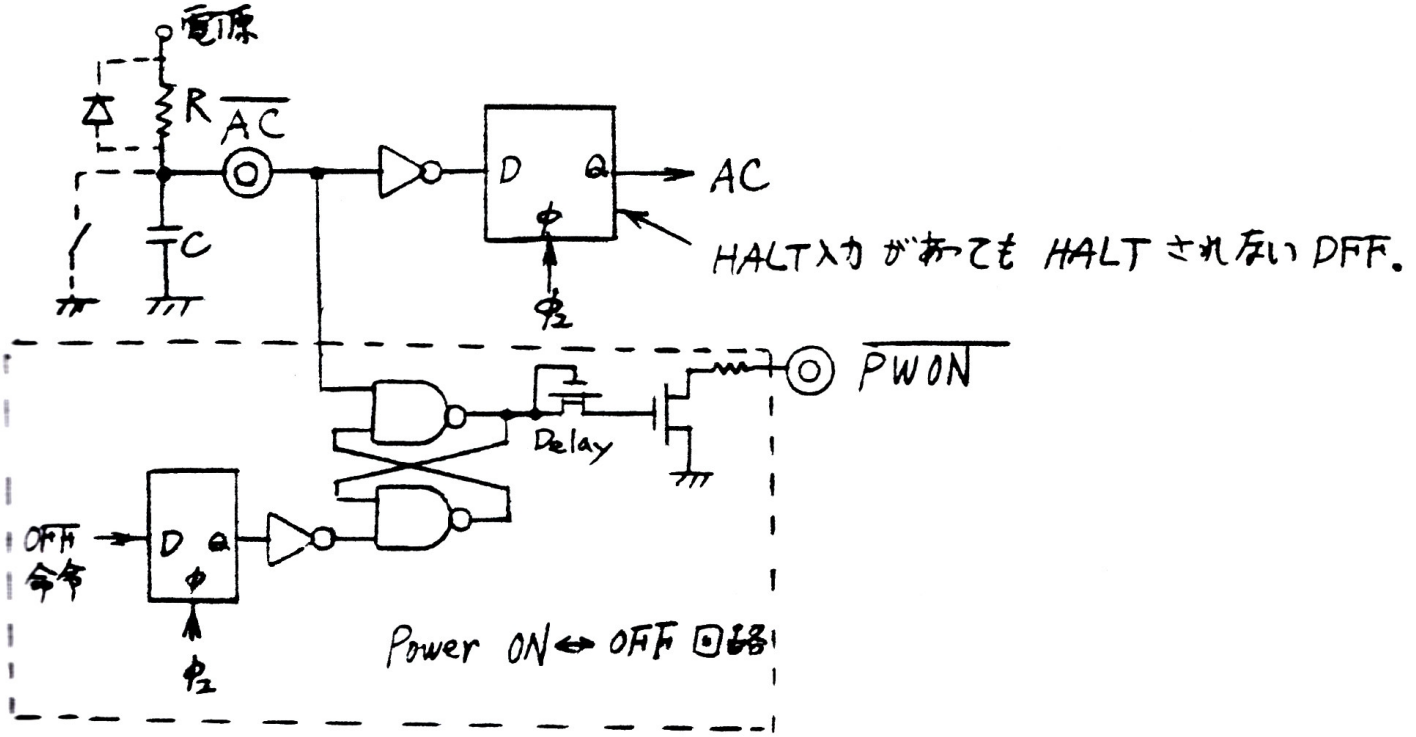
(注) $\overline{PWON}$ をONする $\overline{AC}$ の Low Level は システムリセットの Low Level より低い電圧です。



2.21 リセット入力

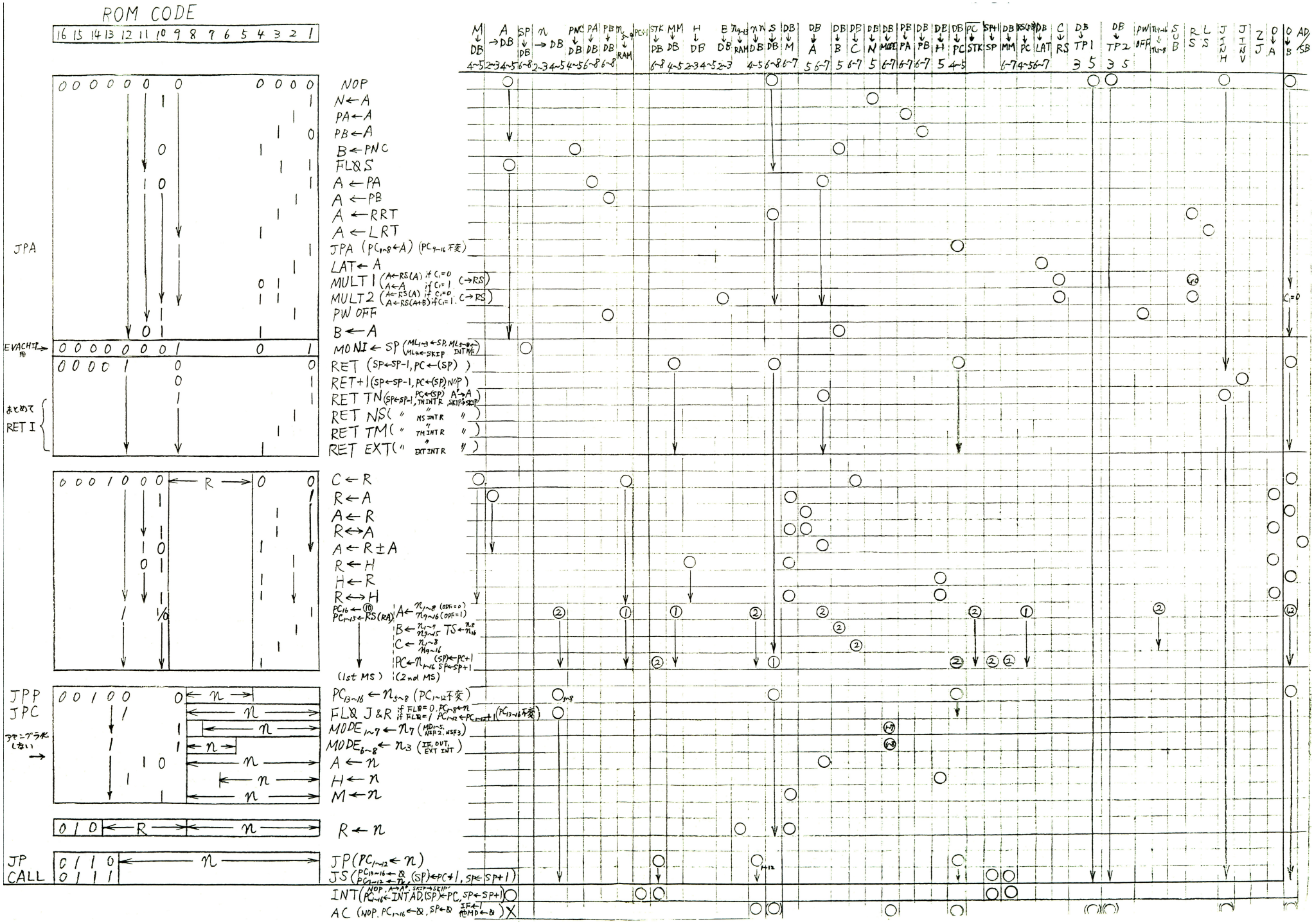
○ リセット入力端子 $\overline{AC}$ に対して "L" を最低9マシンサイクル分入力すると次の回路がリセットされます。(但し, エバチップでは HALT 端子は "L" になっていなければなりません。)

PC ← &&&&H
SP ← &H
SKIP ← &
各INT F/F ← &
各INT要求F/F ← &
MD_7(IF) ← 1
MD_0~MD_6, $MD_{8\sim9}$ ← &
RG ← &&
NSC ← &H
DOWN CO ← &&H
Binary CO ← 1FFH
$\overline{PWON}$ ← ON
NOP 命令を出す.



○ $\overline{AC}$ 端子が "H" になって 1マシンサイクル後より プログラムは &&&&H 番地より スタートします。

○ EVACHIP では HALT 中に $\overline{AC}$ 端子を "L" にしても 1部の回路しかリセットがかかりませんので, HALT 中は $\overline{AC}$ 端子を "H" に保って下さい。


ROM CODE
16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
NOP
N←A
PA←A
PB←A
B←PNC
FLQS
A←PA
A←PB
A←RRT
A←LRT
JPA
JPA (PC1~8←A) (PC9~16不変)
LAT←A
MULT1 (A←RS(A) if C1=0 / A←A if C1=1 C→RS)
MULT2 (A←RS(A) if C1=0 / A←RS(A+B) if C1=1 C→RS)
PW OFF
B←A
EVACHIP用
MONI←SP (ML1~3←SP, ML5~8←INT R/F / ML4←SKIP)
RET (SP←SP-1, PC←(SP))
RET+1 (SP←SP-1, PC←(SP), NOP)
まとめて RET I
RET TN (SP←SP-1, PC←(SP), TN INT R, A'→A, SKIP→SKIP)
RET NS (" NS INT R ")
RET TM (" TM INT R ")
RET EXT (" EXT INT R ")
C←R
R←A
A←R
R↔A
A←R±A
R←H
H←R
R↔H
PC16←⑩ PC1~15←RS(RA)
A←n1~8 (ODF=0) n9~16 (ODF=1)
B←n1~7 n9~15 TS←n8 n16
C←n1~8 n9~16
PC←n1~16 (SP)←PC+1 SP←SP+1
(1st MS) (2nd MS)
JPP
JPC
アセンブラ化しない
PC13~16←n5~8 (PC1~12不変)
FLQ J&R if FLQ=0, PC1~8←n / if FLQ=1 PC1~12←PC1~12+1 (PC13~16不変)
MODE1~7←n7 (MD1~5, NSF2, NSF3)
MODE6~8←n3 (IF, OUT, EXT INT)
A←n
H←n
M←n
R←n
JP
CALL
JP (PC1~12←n)
JS (PC13~16←Q, PC1~12←n, (SP)←PC+1, SP←SP+1)
INT (NOP, A→A', SKIP→SKIP', PC1~16←INT AD, (SP)←PC, SP←SP+1)
AC (NOP, PC1~16←Q, SP←Q, IF←1 他MD←Q)
M→DB 4~5
A→DB 2~3 4~5
SP→DB 6~8
n→DB 2~3 4~5
PNC→DB 4~5
PA→DB 6~8
PB→DB 6~8
n5~9→RAM
PC+1
STK→DB 6~8
MM→DB 4~5
H→DB 2~3 4~5
B→DB 2~3
n9~13→RAM
nN→DB 4~5
S→DB 6~8
DB→M 6~7
DB→A 5 6~7
DB→B 5
DB→C 6~7
DB→N 5
DB→MODE 6~7
DB→PA 6~7
DB→PB 6~7
DB→H 5
DB→PC 4~5
PC→STK
SP+1→SP
DB→MM 6~7
RS(DB)→PC 4~5
DB→LAT 6~7
C→RS
DB→TP1 3 5
DB→TP2 3 5
PW OFF
n9~16→n1~8
SUB
RS
LS
JINH
JINV
ZJ
O→A
O→B
AD/SB
C1=0

16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

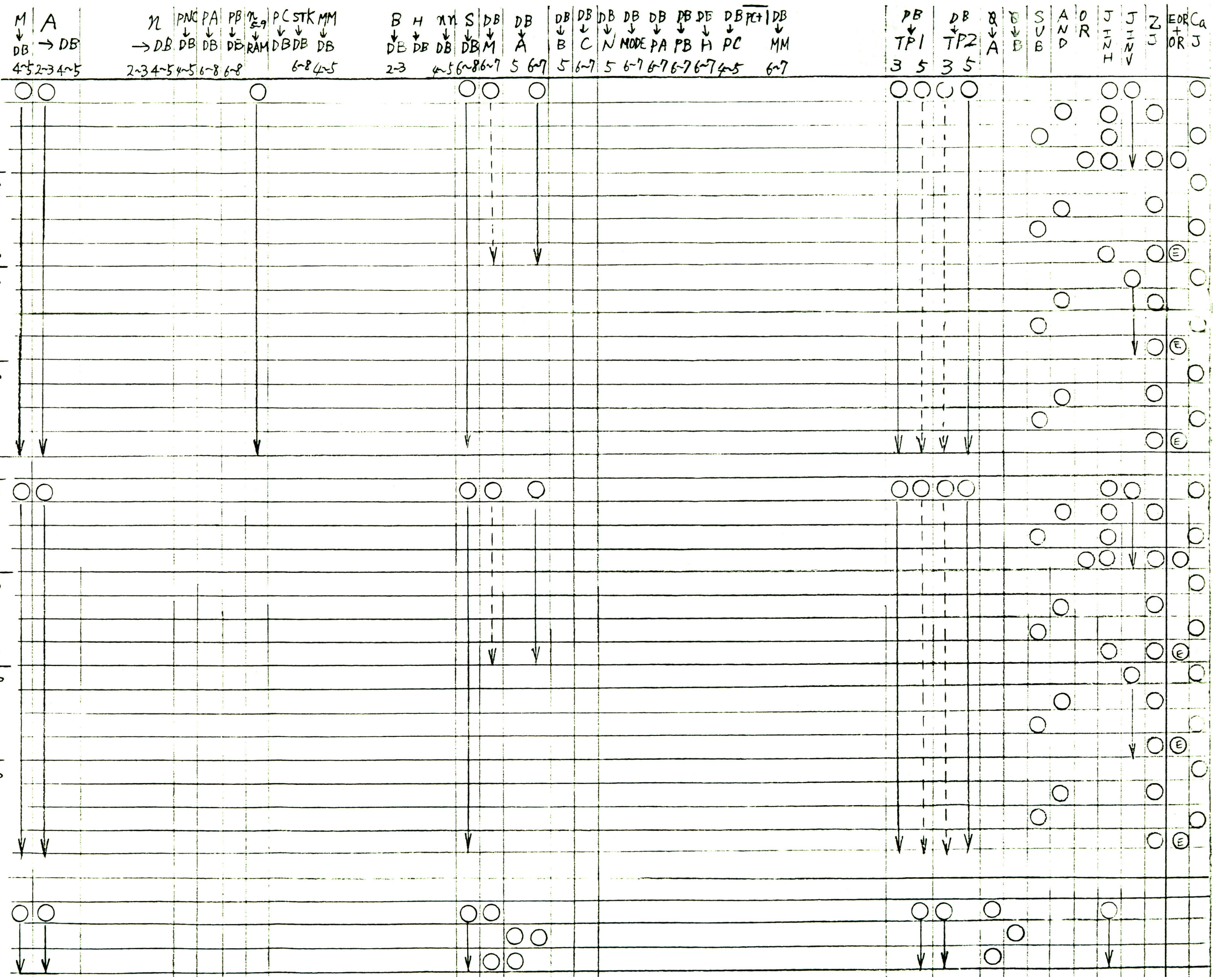
Dest — 0→A, --- 1→R

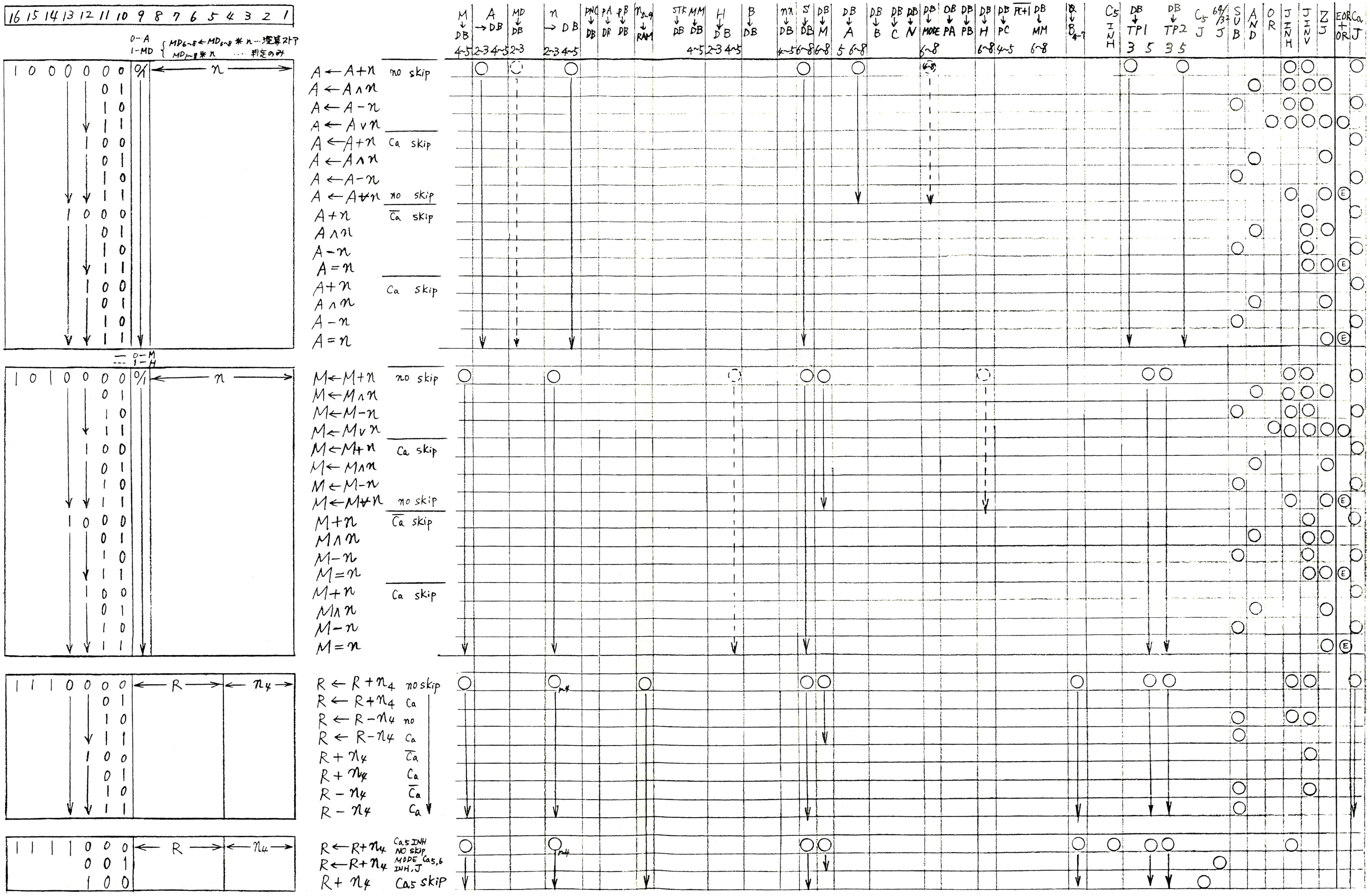
Bits 16–10	9–?	Dest		Operation	Skip
1 1 0 0 0 0 0	← R →	0/1	0	$A \leftarrow A+R$	no skip
0 1				$A \leftarrow A \wedge R$	
1 0				$A \leftarrow A-R$	
1 1				$A \leftarrow A \vee R$	
1 0 0				$A \leftarrow A+R$	Ca skip
0 1				$A \leftarrow A \wedge R$	
1 0				$A \leftarrow A-R$	Ca
1 1				$A \leftarrow A \veebar R$	no skip
1 0 0 0				$A+R$	$\overline{Ca}$ skip
0 1				$A \wedge R$	
1 0				$A-R$	
1 1				$A=R$	
1 0 0				$A+R$	Ca skip
0 1				$A \wedge R$	
1 0				$A-R$	
1 1				$A=R$	

Dest — 0→A, --- 1→M

Bits 16–10		Dest		Operation	Skip
1 1 0 0 0 0 0		0/1	1	$A \leftarrow A+M$	no skip
0 1				$A \leftarrow A \wedge M$	
1 0				$A \leftarrow A-M$	
1 1				$A \leftarrow A \vee M$	
1 0 0				$A \leftarrow A+M$	Ca skip
0 1				$A \leftarrow A \wedge M$	
1 0				$A \leftarrow A-M$	
1 1				$A \leftarrow A \veebar M$	no skip
1 0 0 0				$A+M$	$\overline{Ca}$ skip
0 1				$A \wedge M$	
1 0				$A-M$	
1 1				$A=M$	
1 0 0				$A+M$	Ca skip
0 1				$A \wedge M$	
1 0				$A-M$	
1 1				$A=M$	

Bits 16–10				Operation
0 0 0 1 0 1 1	1	0	1	$M \leftarrow A$
	0	1	1	$A \leftarrow M$
	1	1	1	$M \leftrightarrow A$




16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
0–A
1–MD
MD6~8 ← MD6~8 ✱ n … 演算ストア
MD1~8 ✱ n … 判定のみ
1 0 0 0 0 0 0 0/1 ← n →
A ← A+n no skip
A ← A∧n
A ← A−n
A ← A∨n
A ← A+n Ca skip
A ← A∧n
A ← A−n
A ← A⊕n no skip
A+n $\overline{Ca}$ skip
A∧n
A−n
A=n
A+n Ca skip
A∧n
A−n
A=n
0–M
1–H
1 0 1 0 0 0 0 0/1 ← n →
M ← M+n no skip
M ← M∧n
M ← M−n
M ← M∨n
M ← M+n Ca skip
M ← M∧n
M ← M−n
M ← M⊕n no skip
M+n $\overline{Ca}$ skip
M∧n
M−n
M=n
M+n Ca skip
M∧n
M−n
M=n
1 1 1 0 0 0 0 ← R → ← n4 →
R ← R+n4 no skip
R ← R+n4 Ca
R ← R−n4 no
R ← R−n4 Ca
R+n4 $\overline{Ca}$
R+n4 Ca
R−n4 $\overline{Ca}$
R−n4 Ca
1 1 1 1 0 0 0 ← R → ← n4 →
R ← R+n4 Ca5 INH NO SKIP
R ← R+n4 MODE Ca5,6 INH,J
R+n4 Ca5 skip
M→DB 4~5
A→DB 2~3 4~5
MD→DB 2~3
n→DB 2~3 4~5
PNC→DB
PA→DB
PB→DB
n3~4→RAM
STK→DB
MM→DB 4~5
H→DB 2~3
B→DB 4~5
nn→DB 4~5
S→DB 6~8
DB→M 6~8
DB→A 5 6~8
DB→B
DB→C
DB→N
DB→MODE 6~8
DB→PA
DB→PB
DB→H 6~8
DB→PC 4~5
PC+1
DB→MM 6~8
R→B 4~7
C5 INH
DB→TP1 3 5
DB→TP2 3 5
C5 J
64/32 J
SUB
AND
OR
J INH
J INV
Z J
EOR+OR
Ca J