配布先			
2902 → 2934A → 2934B 2979A → 2960 → 2925 2885B 2937A → B	[redacted] 科学技術計算用 LSI [redacted] の システム解析	IEL － 3465	1/25
		49 － 3 － 4	
		集積回路技術部　民生　課	
		承認 [redacted]　査閲 [redacted]　作成 小口	

このたび、[redacted] 科学技術計算用 MOS LSI [redacted] の [redacted] システム 及び 論理 の 徹底的な 解析を 行ないましたので 報告いたします。

〔1〕演算機能

(a) 置数8桁、演算結果8桁　(但し負数で整数部8桁はOVF)

(b) 使用演算キー

[+] [−] [×] [÷] [=] [AC] [CE] [π] [1/x] [√] [log] [ln] [e^x] [a^x] [sin] [cos] [tan] [度分秒/10進度数]

(c) 完全浮動小数点。　科学技術計算用卓電の場合、小数点は殆んど指数表示 (−99〜+99) されていたが [redacted] は普及型を志向した為に、小数点の移動範囲は、−7〜+7 と狭い。移動範囲を超えたものは常に OVF、又は UNF。

(d) [log] [ln] [e^x] [sin] [cos] [tan] のキーについては、(例) A [×] B [log] [=] のキー操作では "A log(B)" の演算結果は得られない。(ファンクションの記憶とデータの保存が上記キーでは不能となる為。) 下2桁切捨て。

(e) 三角関数のデータは10進度数法のものを入力する。この為に度分秒で書かれたデータを10進度数に変換するキーを持っている。

(例)　10度20分30秒 ＝ 10.341666

10 □ 20 □ 30 □ → 結果(10.341666)

(f) [a^x] キーの使用法

12.3^7 を求めたいとき　12.3 [a^x] 7　とすれば結果がでる。但し指数として小数部を持つデータを入れたとき演算不能となり OVF 表示をする。又指数は整数1桁に限られる。

(g) [e^x] キーにおいて、$x \geqq 10$ の場合はフローの簡略化の為 OVF表示をする。

〔2〕回路構成概略

(a) ROM ··· 512アドレス 出力16ビット、メモリ規模 8192ビット

(b) アダー ··· 10進と16進の加減算の可能な一般的なシリアル、ディレイド、アダー。

(c) レジスタ ··· 48ビット　6本

通常四則演算ではこのうち3本、√計算では4本、科学技術計算として6本全てのレジスタを使用する。

(d.) ステップ・カウンタ … [illegible] 又は [illegible] の如くに
次アドレスを直接指定するROM出力を持つ方式とは異なり、+1可能な、ステップ・カウンタ 8ビット を用いて、ROMアドレスを指定させる。 1ページ 256アドレスとして 2ページ分のアドレスをもっているので ページ指定のアドレスレジスタを1ビット持つ。

(e.) ジャッジ

(i) アダーで加減算を行なった結果、検出されたキャリー、ボローによって無条件に判定所をセットする。

(ii) ビットのキャリー、ボロー ではなく、ディジット単位のキャリー、ボローのみによって判定所をセットする。

以上2通りあり、ROM出力より"判定をせよ"という命令は全く出されない。 判定所がセットした事により次アドレス命令をNOOP扱いとし、実質的に1アドレススキップする。従がって、演算時間としては、スキップした場合も、しない場合も同一となる。

(f.) ジャンプ

(i) 無条件ジャンプ

(ii) 条件つきジャンプ

(iii) サブルーチン ジャンプ

大別すると以上3通り。詳しくは、(i)について改ページをするものとしないものとがある。又、(iii)についてアドレス・スタックレジスタをジャンプの際に更新するものと、更新しないものとがある。更新しないものについては、サブルーチン・エンド命令の次のアドレスはサブルーチン・ジャンプを行なった以前のアドレス・スタック・レジスタ内容によって指示されたものとなる。

(g.) スイープ・レジスタ … 4ビットのレジスタ、特にディジットタイムの異なるデータの移送に用いる為に設けてある。

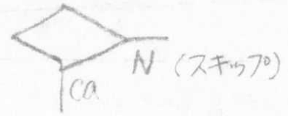
[3] ROM出力詳細

出力ビット数は16. B1～B16と名称をつける.

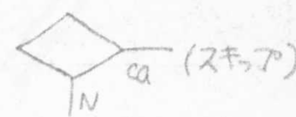
(a.) 48ビット1本のダイナミック・シフト・レジスタには数値データの他に、正負符号データ、小数点データ、演算制御データが各ディジット タイムで区分されて格納されている。どのデータをレジスタ内から取り出してきて処理するのかを限定する為に第1表にある様に、アダーへの読込みタイミング信号を作製する。この為に"B1""B2""B4"を専用に使い、"B3"及び他の出力を信号作製に絡ませる。

(b.) ジャッジ（条件つきジャンプ）の際、キャリー、ボローがでたらスキップジャンプをするのか、その逆なのかを決定する為に"B3"を用いる。"B3"は読み込みタイミング信号作製にも用いる。

(例) B3=0



B3=1



(c.) 加/減切換 B5=0又は B5=1の場合でも B11=0, B12=1ならば加算, 上記以外のとき減算

(d.) レジスタ交換 演算処理を6本のレジスタ全てに行なわせる為、レジスタ→アダー間にレジスタ選択ゲートを置いている。スイープ・レジスタの制御も含めて、"B6""B7""B13""B14""B15"の5ビットを専用している。

(e.) ジャンプ命令のとき B1～B8 の8ビットが次アドレスを指定する為に、ステップ・カウンタにロード・イミディエートされる。

10/16進変換 ±1等の命令については ROM出力の組合せによって作られているので一般な説明はできない。
命令一覧を次ページに示す。

B code 1	2	4	3′		ADDER RI TIME 1	2	3	4	5	6	7	8	9	10	11	12	ニーモニック
				0	■												Xs
			■	1													
		■		2		■											x
		■	■	3											■	■	Xcc
	■			4											■		X_{C1}
	■		■	5													
	■	■		6			■	■	■	■	■	■	■	■	■		X̸
	■	■	■	7													
■				8			■										X_L
■			■	9		■	■	■	■	■	■	■	■	■			X
■		■		A										■			Xm
■		■	■	B		■	■	■	■	■	■	■	■	■	■	■	Xc
■	■			C												■	X_{C2}
■	■		■	D													
■	■	■		E			■	■	■	■	■	■	■	■			X
■	■	■	■	F		■	■	■	■	■	■	■	■	■			xX

(注) B3=1 であっても、[X±Y→X]、[X±1→X]、$\boxed{X-Y}$、$\boxed{X \pm B_{5\text{-}8}}$ の命令実行の際には $B3'=0$ と等価。$B3=0 \rightarrow B3'=0$

又、$\left[\begin{array}{l} X \rightarrow LS \\ x+1 \rightarrow X \end{array}\right]$ の如くなる命令を 1ワード.サイクルタイムで同時処理する為、特に定められた上記以外のコードによって "X" タイミング信号を発生させる。

全てのジャンプ命令の場合 B1~B8 ROM出力は ロード イミディエート に用いられる為 常に "Xcc" タイミング信号を発生させる。

[redacted] データ セレクト タイミング 一覧表

第1表

B3	B5	B8	B9	B10	B11	B12	B16	オペレーション
		0	0	0	0	0	1	X→RS
		1	0	0	0	0	1	X→LS
			0	0	0	1	1	X→LS, $B_{5\sim8}$→X
X	0		0	0	1	0	0	X+Y→X ⑩
X	1		0	0	1	0	0	X−Y→X ⑩
X	0		0	0	1	0	1	X+Y→X
X	1		0	0	1	0	1	X−Y→X
			0	0	1	1	1	✓JPV 0ページへのジャンプ, $B_{1\sim8}$→ステップカウンタ
			0	1	0	0	0	$B_{5\sim8}$→X X内容と$B_{5\sim8}$とを重畳
X			0	1	0	1	0	X+$B_{5\sim8}$→X Judgeせず.
			0	1	0	1	1	$B_{5\sim8}$→X $B_{5\sim8}$を置数
			0	1	1	0	0	X⟷Y
			0	1	1	0	1	X→Y, →X
			0	1	1	1	0	SRE 0ページへジャンプ, スタックレジスター→ステップカウンタ
X	0		1	0	0	0	0	✓ X+1→X
X	1		1	0	0	0	0	✓ X−1→X
X	0		1	0	0	0	1	✓X→LS, X+1→X
X	1		1	0	0	0	1	✓X→RS, X−1→X
X	0		1	0	1	0	0	X+Y
X	1		1	0	1	0	0	✓ X−Y (使用せず)
X	X		1	0	1	0	1	✓ 0±Y (使用せず)
			1	0	1	1	0	✓ JP ページ内ジャンプ, $B_{1\sim8}$→ステップカウンタ
			1	0	1	1	1	JS ステップカウンター→スタックレジスタ 1ページへジャンプ, $B_{1\sim8}$→ステップカウンタ
X			1	1	0	1	0	X+$B_{5\sim8}$ 最上位桁あげCaによるジャッジ
X			1	1	0	1	1	X+$B_{5\sim8}$ 無条件Caによるジャッジ
			1	1	1	0	0	✓ Y→X, →Y
			1	1	1	0	1	Key→X

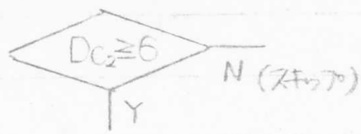
第2表　命令一覧 (1)

B6	B7	B13	B14	B15	オペレーション	
0	0	×	×	×	A → Y	
0	1	×	×	×	C → Y	第2表における Y 表記に相当する
1	0	×	×	×	E → Y	レジスタが左記の様に決定される
1	1	×	×	×	S → Y	
×	×	0	0	0	D → X	
×	×	0	0	1	E → X	
×	×	0	1	0	F → X	
×	×	0	1	1	S → X	第2表における X 表記に相当する
×	×	1	0	0	B → X	レジスタが左記の様に決定される
×	×	1	0	1	C → X	
×	×	1	1	0	A → X	
0	0	×	×	×	X → A	
×	×	1	1	0	Ad → A	左記 条件取れぬとき 優選
0	1	×	×	×	X → C	
×	×	1	0	1	Ad → C	
1	0	×	×	×	X → E	
×	×	0	0	1	Ad → E	
1	1	×	×	×	X → S	
×	×	0	1	1	Ad → S	〃
×	×	0	0	0	Ad → D	〃
×	×	0	1	0	Ad → F	〃
×	×	1	0	0	Ad → B	〃

第3表　命令一覧(2)

(f.) ジャッジの具体例

(i)

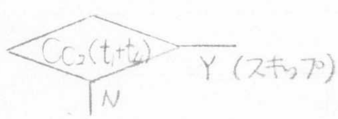


B1〜B16まで 1100101011010000 とすれば良い。

第1表より X_{c2} 、第2表より $X+B_{5-8}$ 、第3表より レジスタは D が選択され 桁あけキャリー についてのみのジャッジであるので $D_{c2}\geqq 6$ となる。さらに $B3=0$ であるので キャリー・ノー で スキップ する。

この種のジャッジは主に ファンクション・キーの種別をジャッジし それに対応して 演算処理をする為に用いている。

(ii)



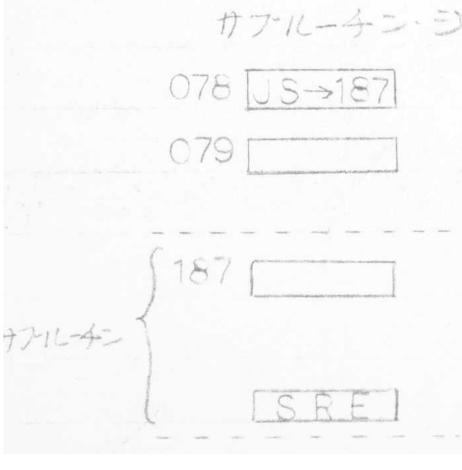
B1〜B16まで 1110001111011011 とする。

第1表より X_{c2} 、第2表より $X+B_{5-8}$ 、第3表より レジスタは C が選択され、X内容(C_{c2})が t_1, t_4 にデータがあれば キャリーが出て 判定F/Fをセットする。 $B3=1$ であるので キャリー・イエス で スキップする。

この種のジャッジは主に ファンクションF/Fの状態をジャッジする目的に用いられる。

(g.) アドレス・ステップ の具体例

サブルーチン・ジャンプ の場合についてのみ述べる。



79→スタック・レジスタ (スタック・レジスタは特別に構成されたものではなく、Fcc の8ビットを流用している。この為 サブルーチン・リンケージレベルは1レベルしか設定できない。)

1ページに改ページ。 87→ステップカウンタに ロード・イミディエート。

187番地以降のサブルーチン・プログラムを実行。　SRE命令で
79→ステップカウンタ、　8ページに改ページ。
879番地以降を実行。

又、SRE命令の前にJS、JPVがきた場合は、サブルーチンリンケージは一レベルのみであるところから、スタックレジスタ内容は更新されてしまう為、改ページを行なう単なるジャンプ命令として使用される事になる。

```
004  [JS→103 ]
005  [        ]
006  [JSS→103 ]
- - - - - - - - - -
         103  [        ]
サブルーチン {
              [ S R E  ]
- - - - - - - - - -
```

JSS命令では、スタックレジスタとして、Accを使わずSを用いる。その為SRE命令でステップ・カウンタに送られるデータは、JSS命令前の（左例では004番地）スタックレジスタ内容となり、アドレス・ステップの様子は次の様になる。

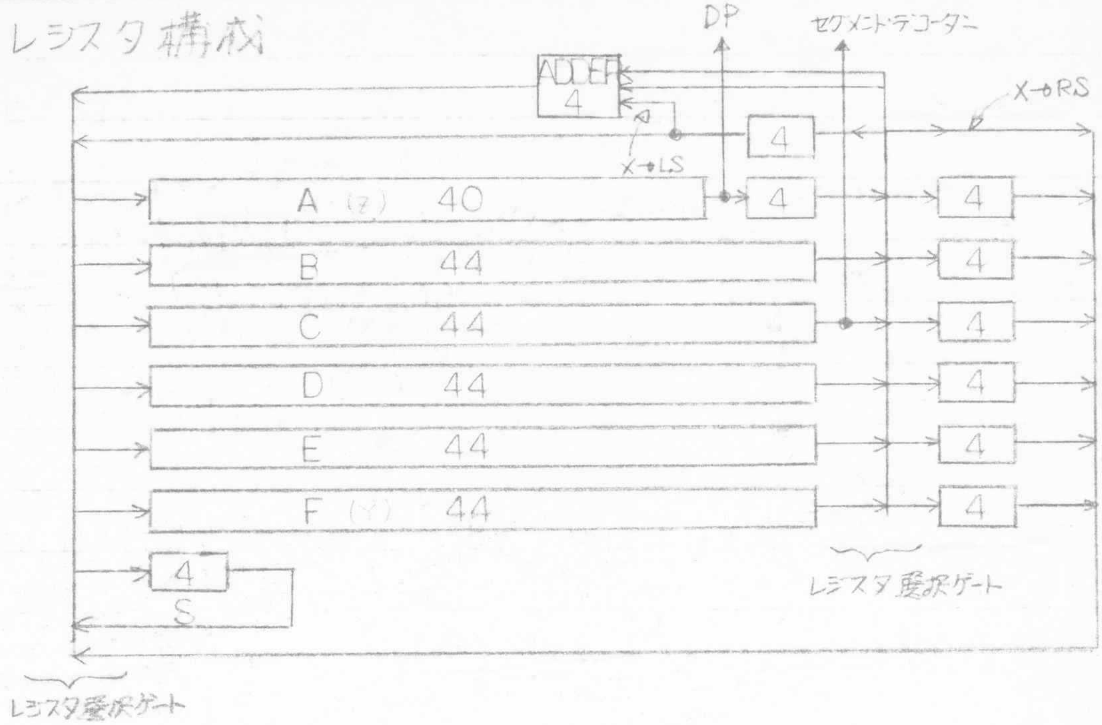
004→103……→005→006→103……→005→

（参）NECマイクロコンピュータにおけるサブルーチン・ジャンプ例

```
    LDI  nn
    XSA
    LAR  a
- - - - - - - - - -
nn  SDS  }
    ……   } サブルーチン
    LDS  }
- - - - - - - - - -
```

左の様に汎用性を持たせる為より細かい基本命令に区分されている事から、[redacted]では、JS→nnという1ワード命令で済ませているオペレーションを、マイクロコンピュータでは4ワード命令となり、且つ、SREという1ワード命令も、3ワードの命令が必要となっている。又、マイクロ・コンピュータについても言えるが、フォートラン・プログラムの様にサブルーチン・コール型でサブルーチンへのエントリー・ポイントの指定が自由にできないわけではないので、サブルーチンによるプログラム・ステップ減少は非常に効果が大きい。

(h.) レジスタ構成



X, Y, Z の 3レジスタに相当するものは Xreg→Creg, Y→Ereg, Zreg→Areg である。

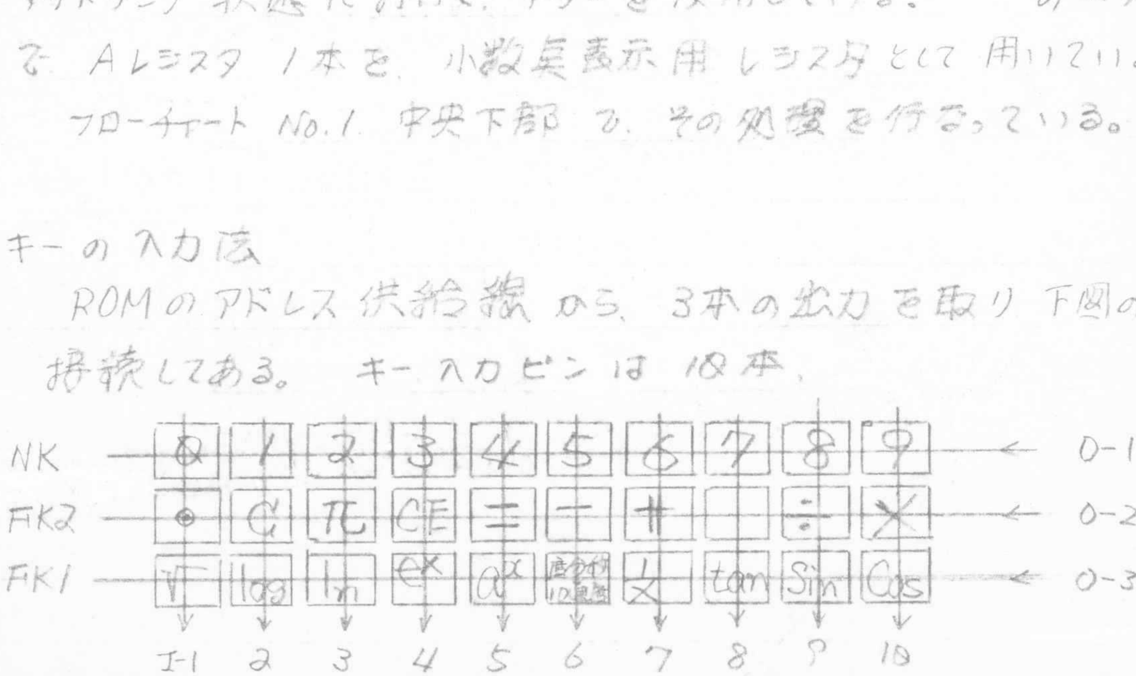
(i) 小数点表示法

一般卓電においては小数点の表示を行なう為に、アダーで小数点データを1桁毎にマイナス1し その時発生するボローを検出するのであるが [illegible] では ① レジスタの本数が多い。② キー・アイドリング状態において、アダーを使用している。 の2点の理由で Aレジスタ1本を、小数点表示用レジスタとして用いている。

フローチャート No.1 中央下部で、その処理を行なっている。

(j) キーの入力法

ROMのアドレス供給線から、3本の出力を取り下図の様に接続してある。キー入力ピンは10本。



キー・エンコーダー が内蔵されており、キーに応じて直列信号がつくられる。　ハードウェアとしては キー・エンコーダー 以外に大きなものは組まれていない。　メカニカル・キー の様に安価なキーの場合、キーを離したときのチャタリング（バウンシング、又はOFFチャタリング）が長時間（20msec程度と言う。）生ずる為特にデバウンス機能を持たないと キーを連続打したかの様な誤動作を起す。　[redacted] では ROM方式電卓設計思想が志向する"ハードを極力減少させ、且つ、プログラム・ソフトを高能率、高級化し、ROMメモリーサイズをも減少する"という理念を実行し、キーの ONチャタリング、OFFチャタリング防止 を ソフトで行ない、さらにチャタリング防止をも含めた、キー入力フロー での アドレス消費量はわずか 15アドレス に留め、チップ面積縮少に大きな効果を得ている。
　2重押し防止、ロールオーバー機能 はない。　[redacted] は 一貫した実用主義 で 卓電を設計しており アクセサリー的なにおいのするもの、実用上不要であると判断したものは、全て"切る"方針である事が うかがわれる。

(11) フローチャート

巻末に フローチャート を付ける。

○ …… 無条件ジャンプ
● …… 改ページ無条件ジャンプ　} 1アドレスを消費する。
▭ …… サブルーチン関連オペレーション

タイミング ニーモニック については 第1表 参照

(a) キー・スタート・アドレス について

$\boxed{0}$〜$\boxed{9}$ の NK については置数後 同一操作を行なわせるので同一アドレスよりスタートする。　FK2 も同一アドレスよりスタートし、キーコードのジャッジを 4回行なう事によって キーの振分けを行なう。FK1 は 同一アドレスよりスタートした後、アドレス修飾ルーチンに入り（142番地から始まるサブルーチン、No.4） 各々のキー・コードによって

定められたスタート番地から、再スタートさせている。

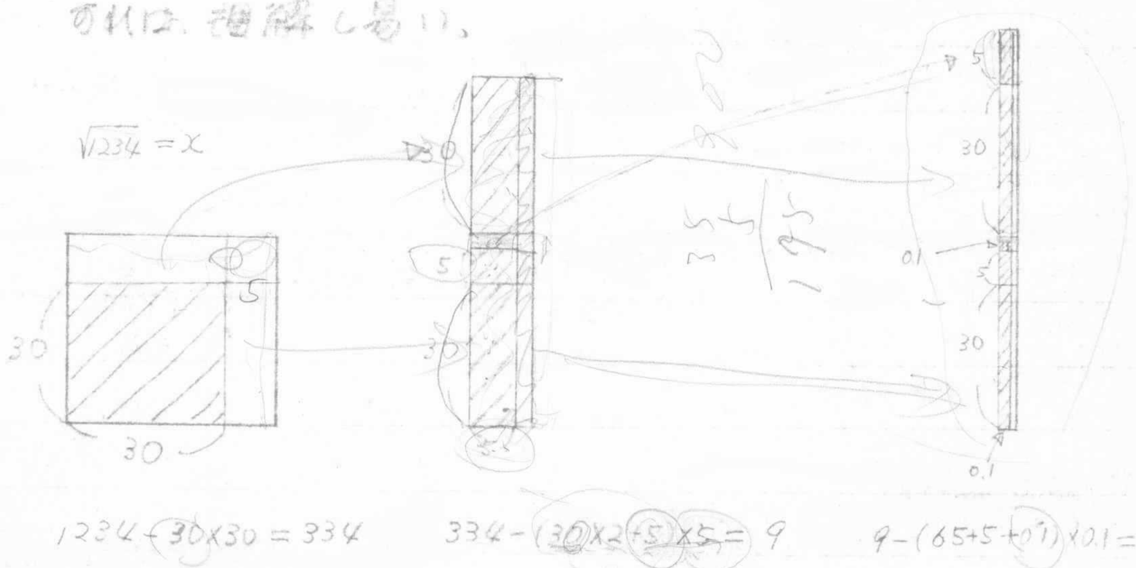
(b) 四則演算フロー

演算フローにおけるアドレス消費は、ステップカウンタ方式によって生ずる不要なジャンプ命令を除くと、実質50数アドレスである。これは［G+RS C+1+C］を1アドレスで実行する等基本命令が豊富で幅がある事、8桁演算の為、小数点処理が楽な事、などによるものである。[redacted]では、フローの簡略化の為に、6桁ダブルレングスという特質を生かし、巧妙なフローを組んでいたが、この[redacted]では、ここで記述する程、特徴のあるフローではないように思われる。

サブルーチン・ジャンプの際のエントリー・ポイントが任意に選択できる為、四則のサブルーチンとして、科学技術計算において大いに活用される。又常にセカンド・ファクターが定数となる様に演算開始時のCレジスタ数値は保存されている。

(c) ルート演算フロー

ルートの計算の方法については、下図の様な模式図を参考にすれば理解し易い。



$1234 - 30 \times 30 = 334$　　$334 - (30 \times 2 + 5) \times 5 = 9$　　$9 - (65 + 5 + 0.1) \times 0.1 = 1.99$

上図の横に倒れていた余りの分を縦積みする為、2倍をする（30×2，5×2）操作が必要となってくる。

右の筆算例で 実際に人間の頭では
$12-1$, $12-2^2$, $12-3^2$, $12-4^2$
の計算をさせ 自然数の自乗の数の
なかで 12以下であって 12に一番近い
ものは "3" であるとの思考を行なう。

ディジタル計算機では 次の様な
演算のくりかえしによって 同等の内容を
実行する。

```
      3  5. 1
 3  ) 1234.
 3     9
 65    334
  5    325
 701     900
   1     701
 702     199
```

（筆算の例）

$\sum_{k=1}^{n}(2k-1) = n^2$ の等式に従がって

$12-1=11$, $11-3=8$, $8-5=3$, $3-7=-4$ （ボロー発生）

1回目　2回目　3回目　4回目

この為に 減数となる奇数を 任意の桁に発生させてやらねばならぬ。又、前ページにあるように、2倍をする操作が必要となる。純2進の演算器の場合には 2倍の操作は単に データを 1ビット 左シフトさせるだけで良いのだが 一般卓電においては、置数及び表示の際の 10進−2進，2進−10進の変換操作の簡略化、直接化を行なう為、2進化10進法（BCD）によって 数値を表現している。この為 BCD においては 同じ数を 加算させる事によって実現する事になるが 実際には ルート演算については この様にはせず、1回余計に引いた時点（即ちボローが発生したとき）での 減数の値、上例で言えば "7" より 1 を減算する事によって可能ならしめる。

しかしながら この様な形で 演算を進めていった場合、（何回減算をしたかを記憶しているレジスタ、即ち演算結果を貯わえるレジスタ、を持っている場合を除いて） 演算結果は 実際の根の2倍の値となる。 純2進法ならば 1ビット 右シフトすれば 即座に ½ できるのであるが BCD の場合 5回加算して 1桁分 右シフトする操作をせねばならぬ。

以上の様に やや煩雑な操作が必要であるが
では 巧妙なフローを作って 簡略化を実現している。

演算結果を5倍する必要性のあるところから、
①被演算数を当初から5倍する。
②奇数の減算に際しても5倍したものを使用し、（5、15、25、35……）2桁目に1を加算すれば良いというハードウェア上の簡略化、及び2桁目の値が減算の回数を表わし、演算結果として、そのまま使用できる、という二重性を持たせる。
という2点の特徴をもったフローを使用している。

この[redacted]のルート計算法は[redacted]プラス ROMの構成によって卓電を製造していた時点において既に使用されていたが、その当時は、1を加算する桁を算定する為に、1を減算する4ビットのカウンターを内蔵していた。ところが"ハードを極力減らす"という、ROM方式設計思想から、[redacted]（8桁ダブルレジスタ 1メモリ、このLSIのフローチャートについてはIEL-3300"[redacted]のシステム解析"の付録として添付してあるので参照）以降、この[redacted]においても、4ビットのカウンターは内蔵しておらず、レジスタ操作によって全てソフトで行なっている。この方式については、NEC においても[redacted]発表以前に既に、プログラム開発済であって、実際に[redacted]（6桁表示12桁卓電用LSI）に採用されている。但し、演算レジスタを3本しか持っていない一般卓電においては、レジスタ操作のみでルート演算を行なわせると $A \times \sqrt{B}$ の様な演算が不能となる。（(イ)演算機能[d]で述べた指数関数、三角関数キーの仕様理由と同じ）この事からも、[redacted]の"不要なものは切る"という思想が感じられる。但し"不要なもの"と判断する基準は電卓各社、各設計者によってまちまちであり、明確な条文として定まっているものではない事は確かである。

(d.) $\boxed{e^x}$ 演算フロー

超越関数計算については、近似計算法を用いる方法もあるが、 ではベキ級数展開式を用いて演算を実行している。

(参) $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots\cdots$ $(-\infty < x < +\infty)$

$\ln x = 2\left(a + \frac{a^3}{3} + \frac{a^5}{5} + \cdots\cdots\right)$ $a = \frac{x-1}{x+1}$ $(0 < x < +\infty)$

$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots\cdots$ $(-\infty < x < +\infty)$

$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots\cdots$ $(-\infty < x < +\infty)$

$\tan x = x + \frac{x^3}{3} + \frac{2}{15}x^5 + \frac{17}{315}x^7 + \cdots\cdots$ $\left(-\frac{\pi}{2} < x < \frac{\pi}{2}\right)$

$\boxed{e^x}$ については　$\exp(I+d)$ のとき（I：整数、d：小数）

$10 \leqq I$ のときは OUT とし、$1 \leqq I \leqq 9$ のように整数部を持つ場合は、その値を保持し、小数部のみ展開式に従がって演算を行なう。これは、整数部のあるデータを展開式に代入していくと、8桁以上の演算結果がでるおそれがある為でもあり、又、その必然性がないからでもある。展開式第10項までの計算を行なう。まとめると、

$$\exp(I+d) = \exp(I) \times \exp(d)$$
$$= (2.718282)^I \times \left(1 + d + \frac{d^2}{2!} + \frac{d^2}{3!} + \cdots\cdots \frac{d^9}{9!}\right)$$

として演算結果を求めている。

$I+d < 0$ の場合は逆数を求めるルーチンを通す。

(e.) $\boxed{\sin}\boxed{\cos}\boxed{\tan}$ 演算フロー

$\sin x$ を展開式第5項まで求め、その値を基に、他の関数は代数式によって計算する。

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!}$$

$\cos x = \sin(x - 90°)$　　$(\cos(x-90°) = -\sin x)$

$\tan x = \frac{\sin x}{\sqrt{|\sin^2 x - 1|}}$ $(0° < x < 10°)$　　$\tan x = \frac{\sqrt{|\cos^2 x - 1|}}{\cos x}$ $(10° \leqq x \leqq 90°)$

角度は 10進度数法で 与えるが まず，その角度を $0° \leq x \leq 90°$ の第1象限の角度に限定する。一般には，$0° \leq x \leq 360°$ に変換する操作をまず行なうが，フローの簡略化の為に，$90°$ を区切り変換していく機能しか持っていない。この為 角度が非常に大きな値（この様な場合は無いと考えて良い。）となると 演算時間が非常に長くなる。これを避ける為に，与え得る角度は，$-1440° \leq x \leq 1440°$ に限り，範囲外の値については，OVF 表示をする。第1象限に変換したのち $\frac{x}{(\frac{180}{\pi})} = \frac{x}{57.29578}$ によって 10進度数 → 弧度法変換をし，展開式に従って 演算結果を求めている。

(f) $\boxed{\ln}$ $\boxed{\log}$ 演算フロー

演算精度をあげる為に， $\ln(X)$ の計算では $X = x^{32}$ とおき

$$\ln(X) = \ln(x^{32})$$
$$= 32 \ln(x)$$

とし，展開式を計算し，

$$\ln(X) = \overset{(64)}{32 \times 2} \left(a + \frac{a^3}{3} + \frac{a^5}{5} + \frac{a^7}{7} + \frac{a^9}{9} \right)$$

として，又，常用対数は

$$\log(X) = \log(e) \times 64 (\cdots\cdots)$$
$$= 27.79485 (\cdots\cdots)$$

として求める。

(g) 演算精度について，

超越関数計算においては 加減乗除のくり返しによって 結果を求める為に できうる限り その演算途中で オーバー・フロー や アンダー・フロー（指数小数点方式では，さほど配慮する必要はない。）を させぬ様に 数値 あるいは 計算法を選定する事，又，演算結果の精度を良くする工夫が必要になる。[redacted] のように 浮動小数点方式の場合，加減乗除 全ての演算を1回行なう毎に 精度が悪くなる。特に乗除計算の際，OVF，UNF させぬ様にしたとしても，小数演算結果のみの場合 表示桁外の数値は，切り捨てられてしまう。又，加減算のときは，指数方式の場合も同様であるが 小数点位置合わせの際に 桁落としが行なわれる。 精度の点では 指数小数点方式のほうが数段 まさっていると言える。（[redacted] は普及型科学技術計算用卓電を志向した品種である。）この為 [redacted] は 演算結果の

下2桁を切り捨てて表示している。

演算精度をあげる為に、[illegible] では、次の様な処置をしている。

(i) $\boxed{e^x}$ 展開式に直接 x の値を代入して求めても良いが整数部と小数部に分け、小数部のみを展開式で求める。

(ii) 三角関数 ···· x のベキ乗解が多いので $0 \leqq x \leqq \frac{\pi}{2}$ に変換しオーバー・フローせぬ様にし、特に $\boxed{\tan}$ 計算では、$0 \leqq x \leqq 10°$ と $10° \leqq x \leqq 90°$ とに区分し、前者においては $\tan x \fallingdotseq \sin x$ となるところから分母の $\sin x$ を優先させ、精度良く計算し、分子の $\cos x$ は $\sin x$ から導出しても、1 の近傍であるところから、精度があまり落ちないようにしている。又 後者では分母が 0 に近くなり演算回数の増加によって精度が極度に落ちる事を考慮し $\cos x$ を直接求めて、計算をしている。

(iii) 対数計算 ···· 展開式の第5項までの計算によって精度良く計算する為には $a = \frac{x-1}{x+1}$ の値を"0"から離れた値にする。即ち x を 1 に近似させてやる必要がある。この為に、まず x の32乗根を求めて x に代入している。

(参) 精度低下につながる演算例 (表示4桁、浮動小数点、指数小数点)

$1111. + 0.999 = 1111.999$

$\fallingdotseq 1112.$ —— 小数点表示方式の如何に拘らず "1111." となる。(5/4 機能なしとする)

$1111. \times 0.001999 =$

浮動小数点の場合 $1111. \times 0.001 = 1.111$ として誤差が大きくでるが、指数方式では、$1.111 \times 10^3 \times 1.999 \times 10^{-3} = 2.220 \times 10^{-3}$ となって有効桁数未満を切り捨てた形で正確に答が得られる。

(高級な科学技術用卓電は、1桁余分に演算結果を出し、最下桁4捨5入という機能を持っているのかもしれない。)

(h.) 展開式演算ルーチン

実際に演算に用いている展開式は次の三式である。

$$e^x = \left(x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \frac{x^6}{6!} + \frac{x^7}{7!} + \frac{x^8}{8!} + \frac{x^9}{9!}\right) + 1$$

$$\ln x = \left(a + \frac{a^3}{3} + \frac{a^5}{5} + \frac{a^7}{7} + \frac{a^9}{9}\right) \times 64$$

$$\sin x = \left(x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!}\right)$$

この三式の違いは、①階乗をつくるか、②奇数のみに演算を限定するか ③正負符号を1回の演算の度反転させるか の3点に集約され、ジャッジによって、この振分けができれば、上式の括弧内の演算は同一ルーチンで実行可能となる。[redacted]では、32アドレスでこのルーチンを作っている。

[5] まとめ

[redacted]は、[redacted]が開発した、8桁浮動小数点方式科学技術計算卓電用LSIであるが、キーについては、使用頻度の高いものだけに限定し、演算、表示の簡略化の為に演算精度を落とし、一般、通常にはこの程度で充分であると思われるが、演算結果は、6桁までしか得られない等、普及型卓電を志向している。

先年[redacted]が[redacted]を発売して、安物家庭用卓電の一番乗りを宣言して大反響を巻き起こし、他の電卓メーカーもこれに追随しようと必至だが自社技術では設計できず、NECなど、LSIメーカーの標準品を使って、この場をしのごうとするメーカーばかりで、ここ当分、[redacted]の電卓業界におけるトップの座はゆるぎないものとなろう。さらに[redacted]の開発によって、科学技術卓電分野にも、実用本意の安物攻勢を仕掛けてきたわけで、[redacted]の高い電卓設計技術と、合理精神とによって[redacted]がますます電卓業界に吹き荒れる事は、まず間違いない。

Seven pages (page 19 to 25) of flowchart were omitted.