世界が注目

磁気カードが使える　モデル65

マイクロ・カリキュレータ

YHPのモデル65は，世界で最初の超小形磁気カードリーダを備えた，プログラムのできるマイクロカリキュレータです。モデル65は使う場所を選びません。いつでもどこでも，手軽にお使いいただけます。

- 充実したソフトウェアパック（記録済み磁気カード付き）
- プログラムの書き込み，修正，記録が容易
- 51種類の関数および演算機能
- 超小形・軽量（310g）

横河・ヒューレット・パッカード株式会社
YOKOGAWA-HEWLETT-PACKARD, LTD.

2470→2485A→2518→4335
2446(2) 2486A→8
2508→4320→2509→4322

IEL - 3926

YHP マイクロ・カリキュレーター

"モデル65" 解析報告

昭和 50 年 3 月 10 日

集積回路(需)第2回路設計部
[redacted]
川口

pg(3) ce

YHP "モデル65" は磁気カード リーダ/ライター を内蔵し プログラム の書き込み 読み出し 可能な ハンディタイプ の卓上機 であって、同系列品 である、モデル35" "モデル45" について、昨年初めに 発表された。 このたび、販売(需)設計部 より 実機 と提供され、そのシステム、ソフトウェア の解析 を行なったので報告する。

〔1〕解析方法

今回の解析 では、ICケース のフタ をはずす事 をせず、非破壊 のまま、HP より発行された資料 及び 特許公開公報 を参考として 解析 を行なうに 留めた のち、プログラム機能部 及び それに関する 入出力制御命令 等に 不明な点 が 数多くあるが 全般的にみて、高度な 新味ある設計 を行なっている 事には 変わりなく、これ以上の解析 を行なっても、得るところは 余り 無いと考える。

ROM に格納されている プログラム・リスト を解読する 為 下図の様な構成で ROM コード の読み取り を行なった。

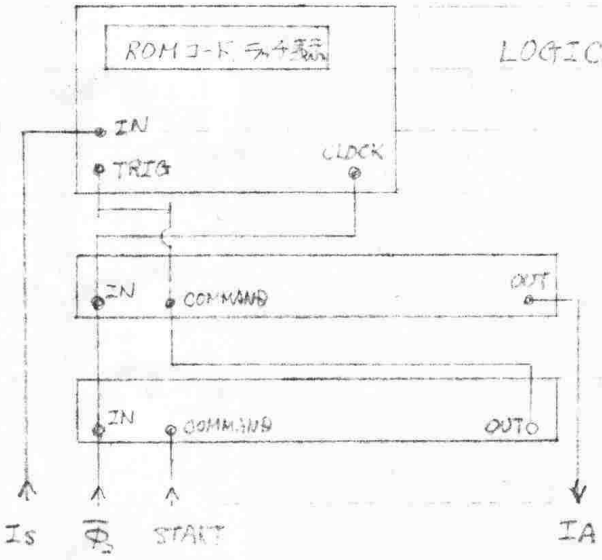


図1. ROMコード 読み取り

ワード開始 1ビット目に発生するシステム同期パルスである START 発生より 19ビット後に Instruction address Bus [IA] にシリアルなアドレス信号が出力される様に RPG. #1, #2 を調整。 さらに、ワード開始より 47ビット目より 10ビット期間 Instruction Bus [Is] に出力される ROM出力を、LOGIC ANALYZER によって、ラッチ信 表示させる。 ※出力信号のタイミングの詳細については、No.11. 参照。

アドレス	ROM コード 10 9 8 7 6 5 4 3 2 1	命令
06240	1 1 1 1 1 1 1 1	Cs+1 → Cs
1	1 1 1 1 1 1	Ⓐ − Ⓑ → Ⓐ
2	1 1 1 1	IF NO Ca GO TO 240
3	1 1 1 1 1 1	Ⓐ + Ⓑ → Ⓐ
4	1 1 1 1	SELECT ROM 07
07245	1 1 1 1 1 1 1	JP → 155
07155	1 1 1 1 1	Ⓑ ⟷ Ⓓ
.....		----------
07160	1 1	F2 ≠ 1
1	1 1 1 1 1 1	THEN GO TO 166
2	1 1 1	4 → C(P)
3	1 1 1 1 1 1	C+1 → C
4	1 1 1 1 1 1 1	IF NO Ca GO TO 171
5	1 1 1 1 1	6 → C(P)
6	1 1 1 1	P ≠ 1
7	1 1 1 1 1 1 1	THEN GO TO 165
07170	1 1 1 1 1 1	Ⓒ → RS
07171	1 1 1 1 1 1	Ⓒ → RS

図2. ROM 命令一例

フローチャート No.13 にある (pseudo) Division → 定数発生 の部分の ROM 番地の一部を 図2. に示す。

4] 卓電セット

(1) LSIの構成

CPU, データ・レジスタ, メモリ・レジスタ, プログラム・ストレージ・レジスタ および その制御回路, ダイナミック発生回路, キー入出力回路, 磁気カード リーダ/ライター インターフェース IOC, その他 負荷駆動 を含め 以上の回路を 混成集積化した 44ピン セラミック・パッケージ LSIを 中心として, 1KW×10ビット の ROM および 周辺制御回路を含む 16ピン セラミック DIP 3個, 原発振 800kHz の LC による バイポーラ OSC, それを 1/4 の 200kHz に分周し, $\overline{\Phi}$, Φ の 2相 クロック を発生させる回路を含む LED アノード・ドライバー 20ピン プラスチック DIP 1個, 同じく LED カソード・ドライバー 1個, 磁気ヘッド用 リード・センス・アンプ, ライト・アンプ をおさめた バイポーラ 16P プラスチック DIP 1個 を構成される。

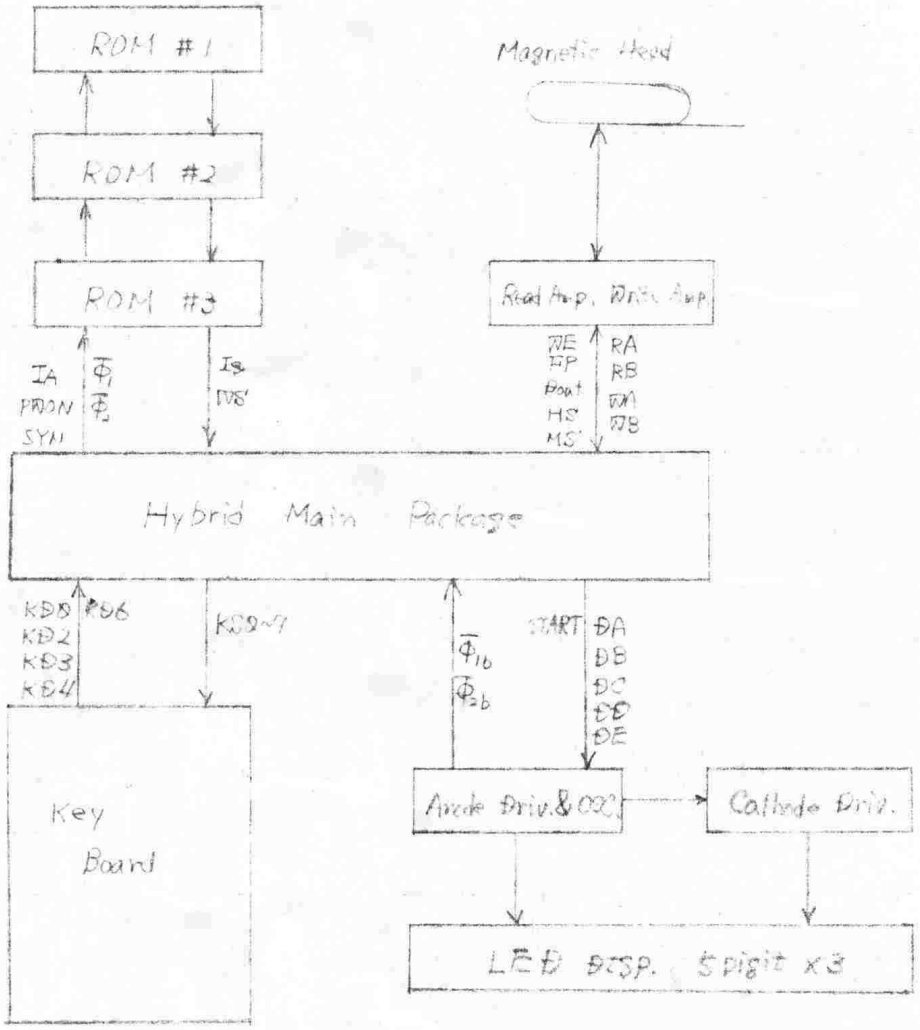


図3. LSI 構成図

(ii) プリント板、キーボード等.

CPU、ROM を取り付ける PCB は 4 層のものを使用している事、回路構成が、バイポーラ回路は全て薄膜混成化している様である事の為、この PCB 上に取り付けられている C.R.Tr.D 類は高電力消費 大容量の電源回路関係のみとなり、高密度実装を可能としている。 さらに、実装状態での試験を PCB ごと自動的に行なう為に 検査用端子が、あらかじめ、PCB 上に設置されている。

電源電圧 : 入力 +3.75V 出力 : +6V, +7.5V, −12V

クロック・レベル : バイポラ・レベル , MOS レベル

ROM チップの消費電力 $V_{GG} = -19.1$ V $I_{GG} = 0.16/3$ mA

$V_{DD} = -6.4$ V $I_{DD} = 2.2/3$ mA

ROM 1個あたりの消費電力 = 5.7 mW

キーボードは LED ドライバーを搭載した 1枚 PCB であり 触針電極用の金具を溶着して取り付けてある。

キーボードの配線及び LSI のピンコネクションを次ページに示す。

◦端子機能説明

LEARN ... 同期PROM/RUN の切換スイッチが RUN のとき "H" 同期PROM のとき "L"

SYN ... START と同様システム同期信号であって "H" の期間、Is (双方向性) に ROM出力が現われる。

PWON ... 電源自動クリアの信号が入力される。

WS ... レジスタ内データのうち処理したいタイミングのみ "H" となる。

BCD ... 演算レジスタ C と データ・ストレージ・レジスタ 及び プログラム・ストレージ・レジスタ とを結ぶ 双方向性 データ・バス。 PCB テスト用として出力されている。

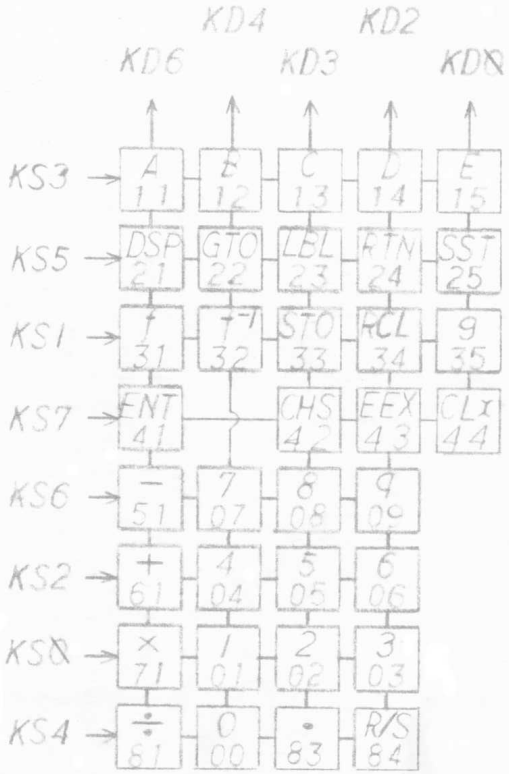
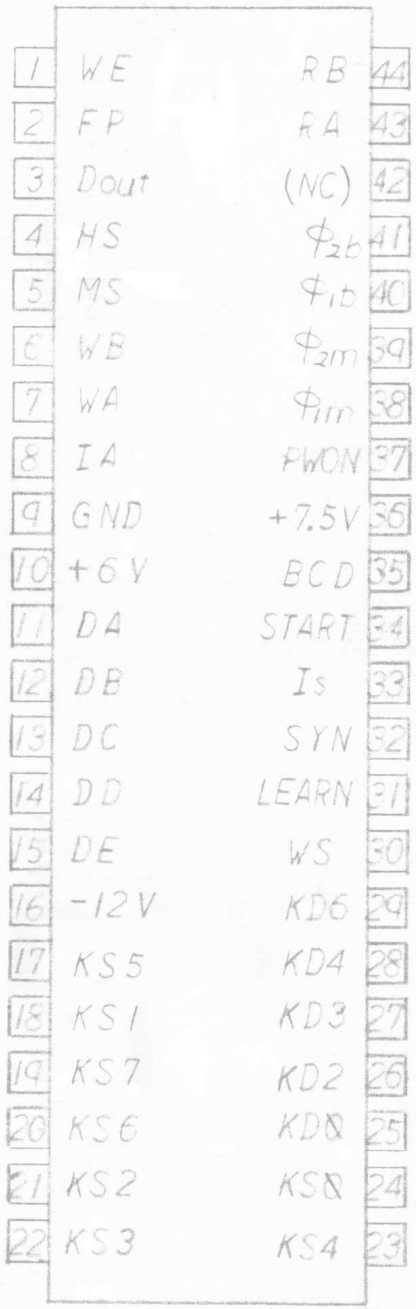


図4. キーボード

ROM x 3

MK6111P
MK6112P
MK6113P

図5 ピン接続図



[3] 演算仕様概略

- 逆ポーランド記法による四則演算
- √, Y^x, n!, 三角関数, 対数関数, 直角→極座標変換, 8進変換, [f^{-1}]キーによる逆関数又は逆変換
- 演算結果の指定 RADIAN, DEGREE, GRADIENT
- [f], [f^{-1}], [g], [STO], [RCL]の5種のプリフィクス・キーを持ち 1つのキーで最大6通りの演算機能を持つ。 (例) [5]キー … ① [5] … "5"の置数, ② [f]→[5] … COS(x), ③ [f^{-1}]→[5] … $COS^{-1}(x)$

  ④ [g]→[5] … Y^x, ⑤ [STO]→[5] … メモリ5にストア, ⑥ [RCL]→[5] … メモリ5をリコール
- 任意の関数をプログラムできる user definable key を持つ。

  (例) モデル65 には双曲線関数キーが無いが 例えば SINH(x) キーを [A]キー (user definable key) で代用したい場合には次の様にプログラムする。

  W/PRGM モードにて [LBL] [A] [f] [Ln] [↑] [g] [4] [−] [2] [÷] [RTN]

  ([f] [Ln] = (e^x), [g] [4] = (1/x))

W/PRGM モードにすると プログラム・ストレージ・レジスタ内容 1キー分が表示され [SST] (シングル・ステップ・キー) を押す事によって, 1ステップ毎にプログラム内容を読み取る事ができる。 さらに内容の DELETE (削除), INSERT (挿入) も可能である。

表示されるキーコードは キー配列の行・列に 1:1 に対応しており, (No.5 キーボード図 キー下部に示してある数値) プログラム確認が容易である。 このコードは 実際に プログラム・レジスタ内に格納されているコード (各々のキーのスタート・アドレスに相当する。) とは異なる為 コード変換を行なう為のリストが ROM に組まれている。

(例) [ENTER] 表示コード 41 レジスタ・コード 76

- その他 省略

(1) 逆ポーランド記法 (Polish Postfix Notation) について

通常の卓電では A+B·C を行なう為に, まず BXC を計算し, その結果に A を加算する。 従がって 数式通りの計算はできない。 ( ) 付の計算などはなおさらである。 これは 演算子 (オペレータ) が オペランドの中間に置かれる Infix 記法である為 デジタル・カルキュレータで 左から順に読んでいく際 オペレータの演算順位が明確でない為である。 順序制御機械である デジタル・カルキュレーターでは 左から順に読んで演算を行なえる, この

逆ポーランド記法は、数式が表現されている方がより実際的である。
この為 通常コンピュータ（HPの数式通りの計算式入力が可能なデスクトップ卓電を含め）では 数式を一旦 逆ポーランド記法に変換してから演算を実行する。

(例) $G = A - \frac{F}{\frac{C}{B}+DE}$ FORTRAN では G=A-F/(C/B+D*E)

スタック配列出力	オペレータ・スタック	入力配列チェック
A		−
A	−	F
AF	−	/
AF	−/	(
AF	−/(	C
AFC	−/(	/
AFC	−/(/	B
AFCB	−/(/	+
AFCB/	−/(+	D
AFCB/D	−/(+	*
AFCB/D	−/(+*	E
AFCB/DE	−/(+*	)
AFCB/DE*	−/(+	RTN
AFCB/DE*+	−/	〃
AFCB/DE*+/	−	〃
AFCB/DE*+/−		〃

↑ 逆ポーランド記法変換終了

図6. 逆ポーランド記法変換

逆ポーランド記法を直接的に用いた卓電はHP以外には無く一般的ではないが 科学技術計算プログラム計算用としては 数値一時記憶を頻繁に行なわせる必要がなく スムースに計算を行なわせる事ができる。

[4] 回路構成

演算制御命令を格納した ROM と データ・レジスタ として ダイナミック・シフト・レジスタ を使用した ROM-レジスタ 方式 である。

ⓐ ROM ... 3 KW × 10 bit

(i) ROM チップ内構成

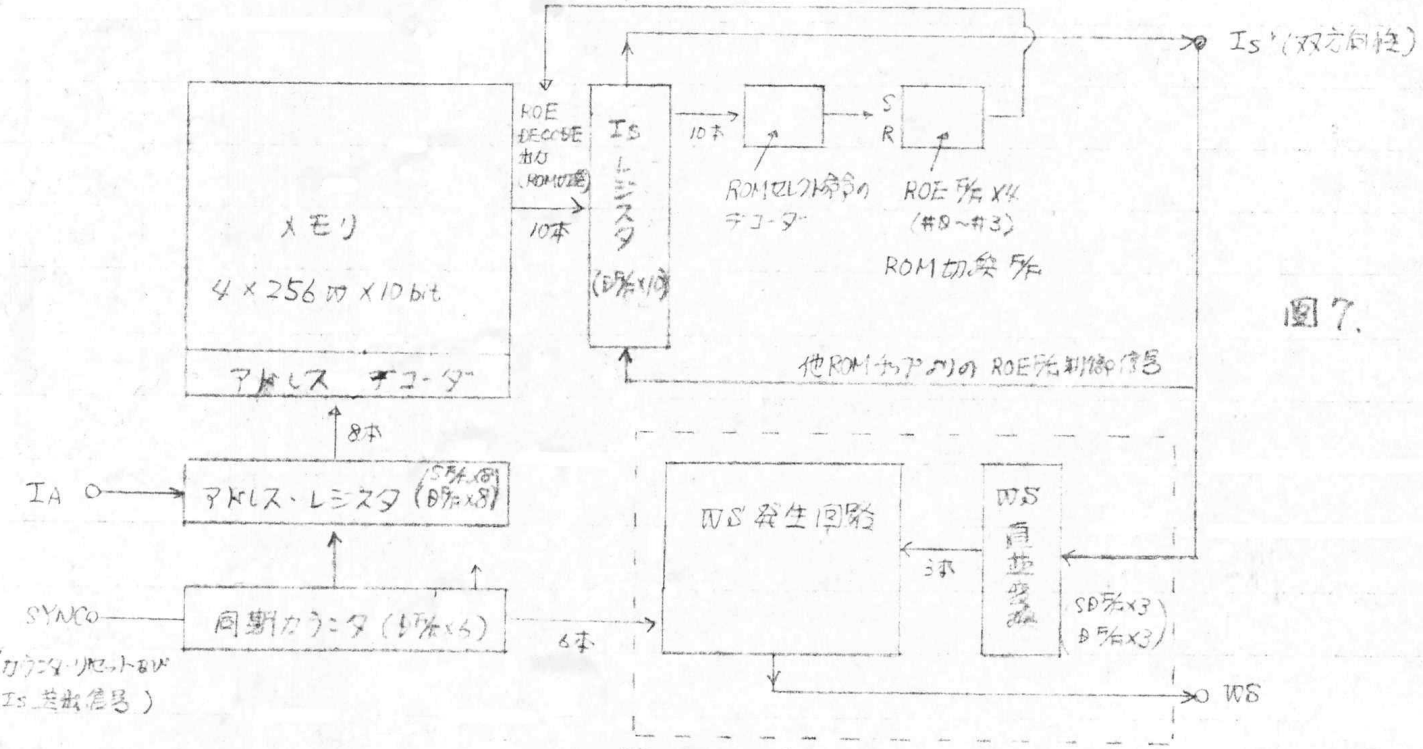


図7.

破線内の回路は必ずしもROMチップ内に入れる必要はならぬが必要は無いと思われる。

以上の様な構成の ROM チップが "モデル65" の場合 3チップ搭載されている。

ROM チップ制御, 入出力信号間の タイミング については No 11 参照。

(ii) 命令

No. 9. 10. 11 参照

(a) JS (Jump to Subroutine) R1=1, R2=0 のとき R3~R10 で示される サブルチン アドレス に ジャンプし, カレント・アドレス+1 を スタックする。判定ビット の状態を無視した ジャンプ命令 としても 用いられる。 又, キー・コード を プログラム・レジスタ に 格納する際, 変換を要する キー の場合 (No 16 後述)

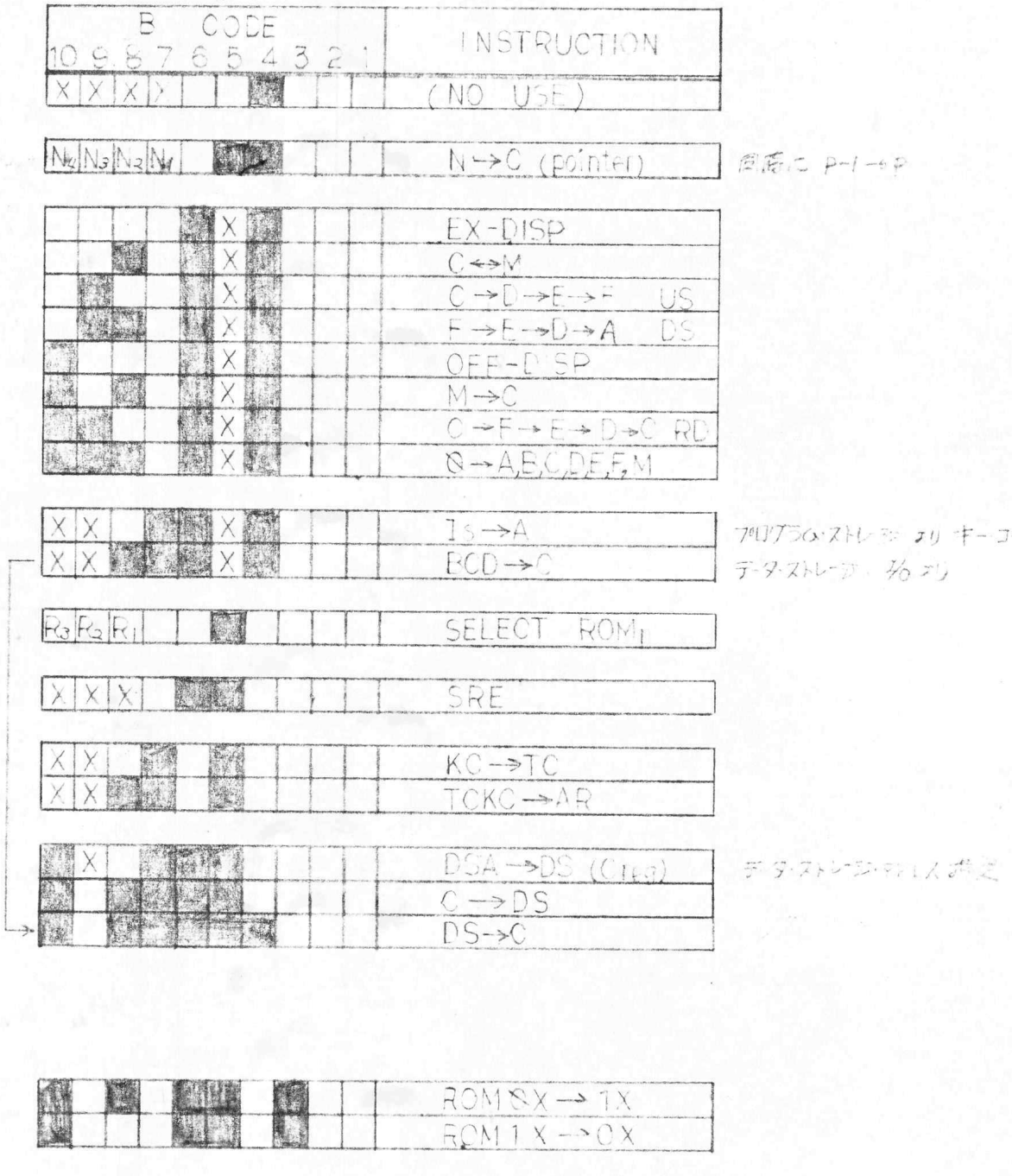
ROM code										Instruction
	12	11	10	9	8	7	6	5	4	
	A_9	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4		JS → A_8 A_7 A_6 A_5 ··· A_4
	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1		JP → A_8 A_7 A_6 A_5 ··· A_1

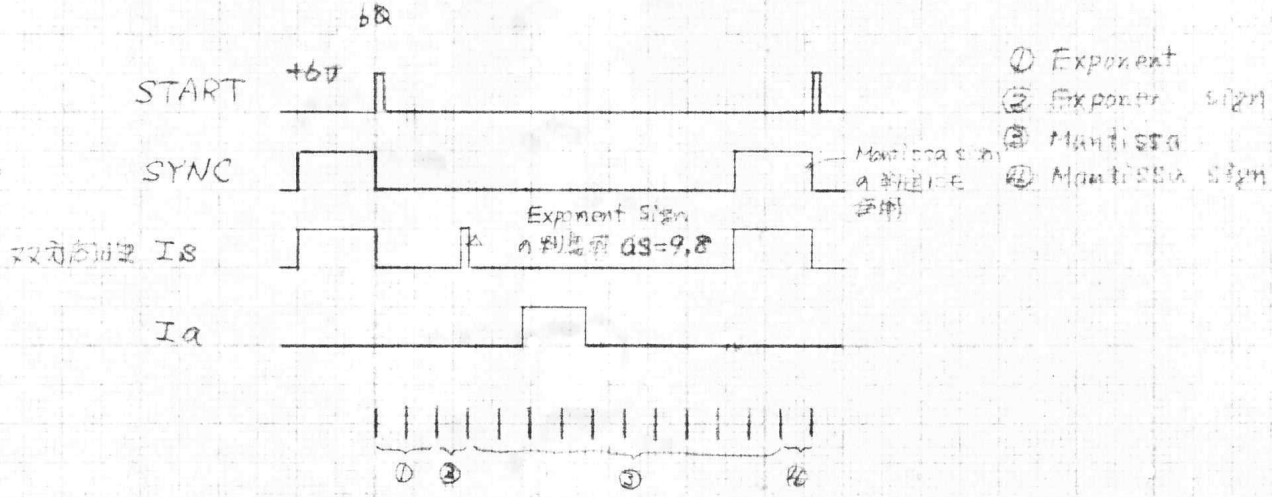
判定先 NO で 分岐アドレス

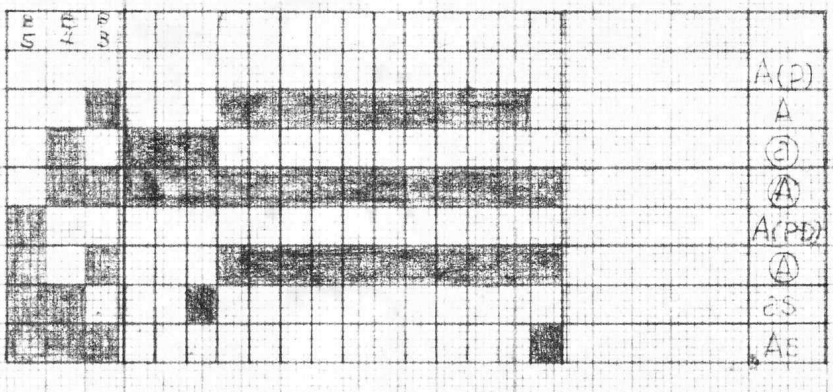
W_3	W_2	W_1	
			0 − B (B=0)
			0 → B
			A − C (A=C)
			C − 1 (C≠0)
			B → C
			0 − C → C
			0 → C
			0 − C − 1 → C
			A → LS
			A → B
			A − C → C
			C − 1 → C
			C → A
			0 − C (C=0)
			A + C → C
			C + 1 → C
			A − B (A=B)
			B ↔ C
			C → RS
			A − 1 (A≠0)
			B → RS
			C + C → C
			A → RS
			0 → A
			A − B → A
			A ↔ B
			A − C → A
			A − 1 → A
			A + B → A
			A ↔ C
			A + C → A
↓	↓	↓	A + 1 → A

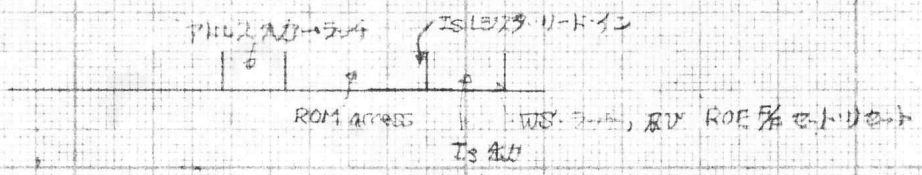
WS 発生 → No.11 参照

	N_4	N_3	N_2	N_1	
FLAG "N"					J → F_n
	↓	↓	↓	↓	F_n ≠ 1 (J)
	↓	↓	↓	↓	0 → F_n
	X	X	X	X	S → F_n
POINTER "N"	N_4	N_3	N_2	N_1	N → P
	X	X	X	X	P − 1 → P
	N_4	N_3	N_2	N_1	P ≠ N (J)
	X	X	X	X	P + 1 → P


B CODE
10 9 8 7 6 5 4 3 2 1
INSTRUCTION
X X X (NO USE)
N4 N3 N2 N1 N→C (pointer)
X EX-DISP
X C↔M
X C→D→E→F US
X F→E→D→A DS
X OFF-DISP
X M→C
X C→F→E→D→C RD
X Q→A,B,C,D,E,F,M
X X X IS→A
X X X ECD→C
R3 R2 R1 SELECT ROM
X X X SRE
X X KC→TC
X X TCKC→AR
X DSA→DS (CLR)
C→DS
DS→C
ROM 8X→1X
ROM 1X→0X


START
+6V
SYNC
Exponent Sign
の判定時 QS=9,8
Ia
① Exponent
② Exponent sign
③ Mantissa
④ Mantissa sign


A(P)
A
③
④
A(P1)
⑤
QS
AB


ROM access

JS→A8～A1 で示されたアドレス・コードが変換器のキー・コードとして用いられる。（フローチャート No.5 参照）

(b) JP --- 判定フラグ（セット条件は次の3条件である。①加減算器より上桁へのキャリー、ボローが出たとき ②フラグ（ステータス・ビット）の判定が Yes のとき ③ポインタ内容判定が Yes のとき）がセットされているときカレントアドレスをスキップ、セットされていないときは JP→A8～A1 で示されたアドレスへジャンプする。

(c) レジスタ間演算、交換、判定

R1=0・R2=1 のとき R3～5 で示される WS信号発生期間のみ当該レジスタよりデータを抽出して処理する。R6～10 をフル・デコードする最小符号化形式をとり 32種のマイクロ命令をつくりだしている為、マイクロ・オーダーの配線及びマイクロ・オーダーとデータ線とを組合わせて、演算を行なわせるデータを選択するゲートがかなり大規模なものであると予想される。

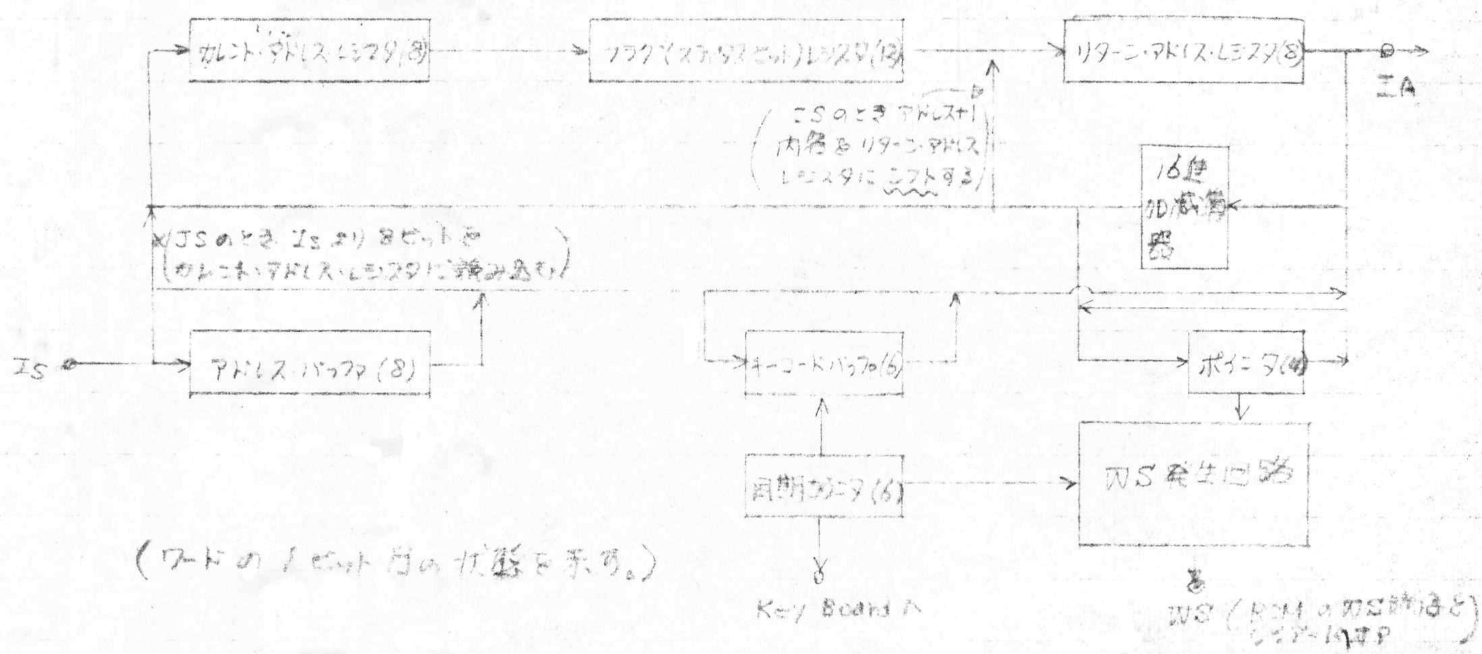
(d) フラグ セット、リセット、判定

F0～F11 までの12種用意されており演算制御の目的に使用される。

(e) ポインタ セット、+1、-1、判定

4ビットの記憶を持ち ±1 可能な 16進加減算器である。ポインタ内データの判定及びロードができる。上記(c)の命令で R3,4,5 が 000 又は 001 のときポインタ・データによって、WS の発生期間が決定される。さらに DO カウンタとして演算回数制御あるいはキー・コードの記憶等にも用いられる。

(a),(b),(d),(e) の命令を実現する回路構成を次ページに示す。



(ワードの1ビット目の状態を示す。)

図 8.

◎ 16進加減算器 は、① アドレス レジスタ +1 ② ポインタ ±1 ③ ポインタ と 比較データ との試し減算 の各用途に用いる。

◎ アドレス・バッファ・レジスタ は JP 命令のとき アドレス部を格納し、判定結果 NO のとき IA に出力する。Yes のときは 無視される。

◎ カレント・アドレス レジスタ、フラグ・レジスタ、リターン・アドレス・レジスタ は 28ビット の ダイナミック・シフト・レジスタ で構成され、1ワード に 2回転している。

(7) C レジスタ への置数

4ビットの任意の数値を ポインタ で示される C レジスタの任意桁に置数できる。この命令のとき、ポインタ は −1 される。

(8) SELECT ROM n

アドレス・レジスタ 8ビット ゆえ 1ページ 256 アドレス である。モデル[illegible] には ROM 00 ～ ROM 07、ROM 10 ～ ROM 13 の 12ページ分の ROM があり、このページの切換を行なう命令である。00～07 10～13 の間であれば自由に ROM の選択ができるが 0X → 1X, 1X → 0X の場合は ページアドレス下位 3ビット は変更できない。

(例) ROM 03 → ROM 13, ROM 10 → ROM 00

(i) 他の命令については MAIN 参照

⑧ データ・レジスタ, 加減算器

1ワード 14ディジット 56ビット の長さをもつダイナミック・シフトレジスタ。数値一時記憶をする ストーレージ・レジスタ 9本。演算, スタック として用いられる データ・レジスタ 7本。

(i) ストレージ・レジスタ

1ヘタ ずつの番号が割ふられ BSA→BC の命令によって Ds のデータが レジスタ選択 bit を操作する。 STO 1 のキー操作 を表示数値が レジスタ1 に読込まれ, RCL 2 の操作 で レジスタ2 内容が表示レジスタに 呼び戻される。

(ii) データ・レジスタ

A, B, C, D, E, F, M の7本あり レジスタ間の結びつきは下図の通り。演算中は A, B, C の3レジスタを用いて演算を実行する。 C, D, E, F で 4レベルの データ・スタック を構成する。 表示中には 表示をする為に

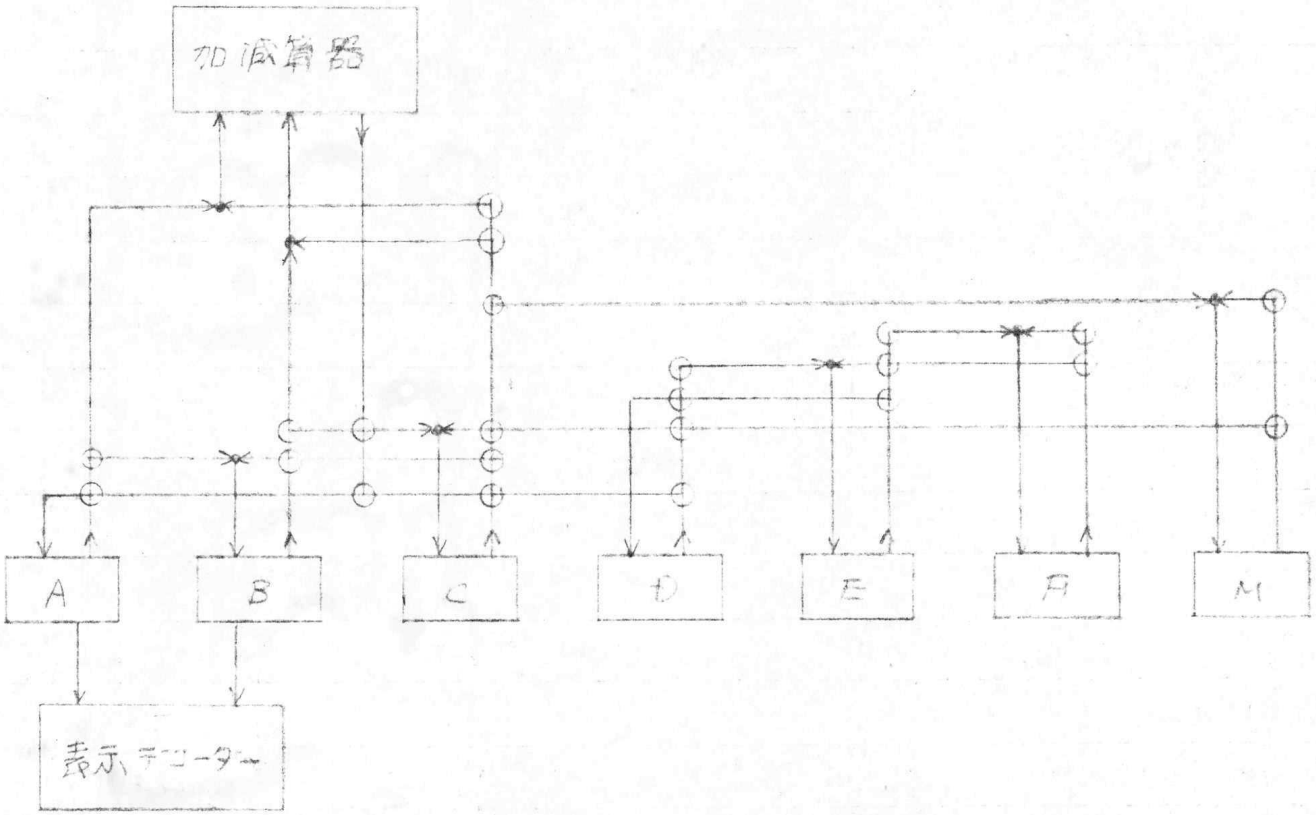


図7 レジスタ間結線

修正されたデータが Aレジスタに，小数点位置（表示1桁を小数点表示に使用する為）及び サプレス・コード を Bレジスタに，即時演算可能な形になっているデータを Cレジスタに格納している。

Mレジスタは，[F] [P] [0] [0] [1] の様な プリフィクス・キーが押されている場合には その記憶に用い，そうでない場合は 演算前に表示されていた数値を記憶している。

(iii) 加減算器

10進の演算のみ可能な 16進演算はできない)。さらに 1つのビットを独立に演算させる機能を持たないので，指数部，仮数部の符号桁として各々1桁ずつを専有している。

◎表示中のレジスタ内容例

-123.456
- A … 9123456000000
- B … 0002000999999
- C … 9123456000002

1.23456 x10^-78
- A … 0123456000078
- B … 9200000999000
- C … 0123456000022

符号桁は 0で正 9で負。(A,Cレジスタ)
マスクする桁には 9 小数点を表示させる桁には 2 を入れる。(Bレジスタ)
指数部が負の場合 符号桁を含め 2の補数 で表現する。

© プログラム・ストレージ・レジスタ

1ステップ 6ビットで、100ステップの記憶可能な ダイナミック・シフト・レジスタ。6ビットで、キーコードを表現するので、64種のコードを記憶できる。実際にキー・ボードに取付けられているキーは 35種であり 残りの29のコードを使用しないのは無駄が多い。さらに モデル65では プリフィクス・キーの使用頻度が高い事が予想され 常に プリフィクス・キー 1ストローク について、1ステップを消費してしまうと、実質的な プログラム・ステップ数が減少してしまう事

もあり，下図の様に ストローク のキー操作で 1ステップ の表現が
できる様に キーコード の変換を行なっている。

キー	コード	表示	キー	コード	表示	キー	コード	表示	キー	コード	表示
[g] [NOP]	00	35 01	[RCL] [7]	20	34 07	[RCL] [3]	40	34 02	[STO] [2]	60	33 02
[STO] [4]	01	33 04	[g] [x≠0]	21	35 07	[RCL] [1]	41	34 01	[STO] [1]	61	33 01
[LBL]	02	83	[6]	22	06	[R/S]	42	84	[9]	62	09
[2]	03	02	[5]	23	05	[.]	43	83	[8]	63	08
[1]	04	01	[4]	24	04	[R↓]	44	00	[7]	64	07
[GTO] [6]	05	33 06	[RCL] [6]	25	34 06	[STO] [7]	45	33 07	[STO] [8]	65	33 08
[x]	06	70	[+]	26	61	[-]	46	80	[-]	66	51
[g] [x≠y]	07	35 28	[RCL] [4]	27	34 04	x1			[g] [x≤y]	67	35 22
[9]	10	35	[E]	30	15	[SST]	50	25	[Clx]	70	44
[g] [R↑]	11	35 09	[g] [x=0]	31	35 23	[g] [x<0]	51	35 24	x1。		
[RCL]	12	34	[÷]	32	14	[R↓]	52	24	[E↑x]	72	43
[STO]	13	33	[C]	33	13	[LBL]	53	23	[CHS]	73	42
[f]	14	32	[E]	34	12	[GTO]	54	22	[g] [LSTx]	74	35 08
[g] [R↓]	15	35 08	[RCL] [5]	35	34 05	[STO] [5]	55	33 05	x1。		
[f]	16	31	[A]	36	11	[DSZ]	56	21	[ENTER]	76	41
[RCL] [8]	17	34 08	[RCL] [2]	37	34 03	[STO] [3]	57	33 03	キー無し	77	00 00

図 10

フローチャート No. 5 に キー・コード変換フロー の一部を示す。

プログラム・ストレージ・レジスタ には 制御コード が 3種 挿入されている。

① Marker : プログラム の 先頭 にあり，磁気テープ でいう BOT に相当し，識別を
行なう。

② Main Pointer : メイン・プログラム の 実行ステップ位置 を示す。

③ Second Pointer : メイン・プログラム 内にある [GTO] [0~9, A~E] によって [LBL]
[0~9, A~E] で示される 次2 プログラム に ジャンプ する事 ができる。

この第2プログラムの実行ステップ位置を示す。 LDJ によってこのポインタはメイン・ポインタ位置に復帰する。 従がって1巡するのサブルーチンリンクが可能となる。

以上の3種の制御コードが前ページ図10.*1.のいずれに相当するかは不明。 制御コードのサーチには フローチャートNo.18の 3412Z のサブルーチンを用いていると思われるが詳細は不明。

② キー入出力回路

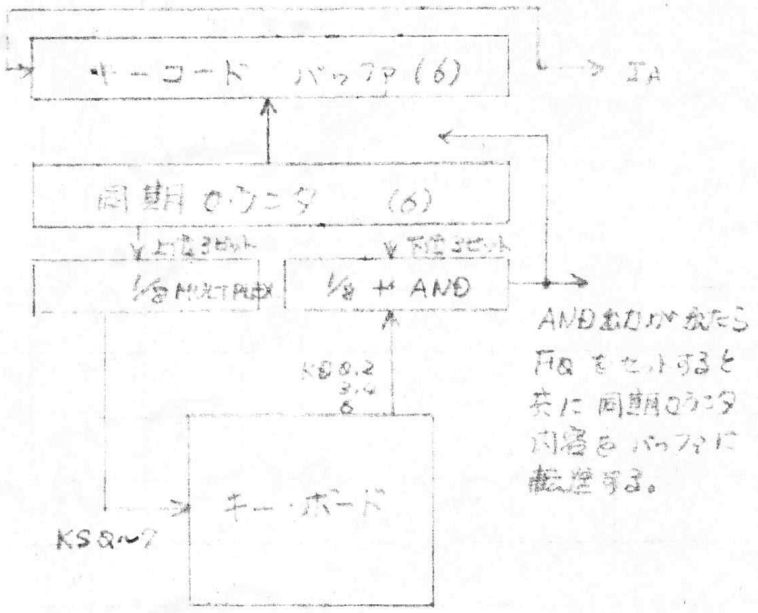


図11.

システム同期カウンタ6ビットの上位3ビットを 1/8 マルチプレクサーで デコードしたビット・タイム出力を キー信号として用いている為キー回路には スイッチング の良い素子を用いているものと思われる。

③ プログラム・ストレージ関連回路（基本となる演算回路とは別に、オプションとして付加されている回路） への ROM からの制御信号コードは R1~R5 = 0 で R6~R10 の値を変える事によって作られている。
モデル65では 22種 使用しているが詳細は不明

[5] 超越関数計算

カシオ製計算機である μPD179C は テイラーのべき級数展開式 第10項までの近似, μPD978C, 979C は チェビシェフ多項式近似の応用である ヘースティングスの近似計算式 によって 超越関数計算を行なっていたが "モデル65" は 従来より世間 よく知られている様に 疑似乗除算 (Pseudo Mult Div.) で構成される CORDIC (Co-ordinate Rotation Digital Computer) と称される計算法を用いている。

④ 三角関数の CORDIC

$$\begin{cases} x_{k+1} = x_k - \delta_k y_k & \text{——①} \\ y_{k+1} = y_k + \delta_k x_k & \text{——②} \end{cases}$$

の様に x,y 平面上の点に定数 δ_k による変換を加えていく事が基本的な考え方である。この様な変換を行なうと [illegible]・カルキュレータ での処理が 後述する様に スムース にいくという利点が生まれる。

①,② の点を 複素平面上の点であるとすると

$$x_{k+1} + i\,y_{k+1} = x_k - \delta_k y_k + i\,(y_k + \delta_k x_k)$$
$$= (x_k + i\,y_k)\cdot(1 + i\,\delta_k) \qquad \text{——③}$$

点 $x_k + i\,y_k$ の 絶対値を R_k, 偏角を A_k とし ③式を 極形式で [illegible] すると

$$R_{k+1}(\cos(A_{k+1}) + i\sin(A_{k+1}))$$
$$= R_k(\cos A_k + i\sin A_k)\cdot\sqrt{1+\delta_k^2}\,(\cos(\tan^{-1}\delta_k) + i\sin(\tan^{-1}\delta_k))$$
$$= R_k\sqrt{1+\delta_k^2}\,(\cos(A_k + \tan^{-1}\delta_k) + i\sin(A_k + \tan^{-1}\delta_k))$$

$$\begin{cases} R_{k+1} = R_k\cdot\sqrt{1+\delta_k^2} & ④ \\ A_{k+1} = A_k + \tan^{-1}\delta_k & ⑤ \end{cases}$$

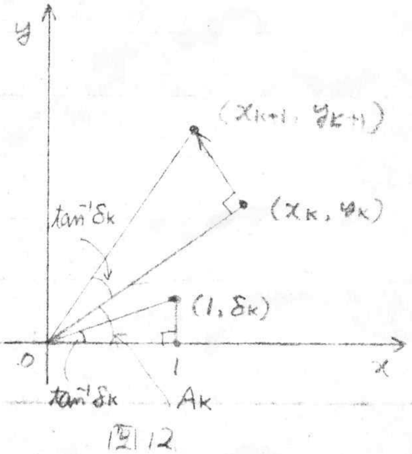


図12

ここで ③式による変換は 具体的に 平面上で どの様なものであるかを考える。左図において ④,⑤式より 2つの三角形は 相似である。従がって 原点と (x_k, y_k) を結ぶ線分に垂直に $\tan^{-1}\delta_k$ だけ 回転させたものとなる。

この変換を (x_0, y_0) から始め (x_n, y_n) に至ったとすると，

④，⑤ より

$$\begin{cases} R_n = R_0 \prod_{k=0}^{n-1} \sqrt{1+\delta_k^2} & \cdots\cdots ④' \\ A_n = A_0 + \sum_{k=0}^{n-1} \tan^{-1}\delta_k & \cdots\cdots ⑤' \end{cases}$$

x, y 座標は

$$\begin{aligned} x_n &= R_n \cos A_n \\ &= R_0 \left(\prod_{k=0}^{n-1} \sqrt{1+\delta_k^2}\right) \cos\left(A_0 + \sum_{k=0}^{n-1} \tan^{-1}\delta_k\right) \\ &= K(x_0\cos\alpha - y_0\sin\alpha) \quad \cdots\cdots ⑥ \end{aligned}$$

同様に $y_n = K(x_0\sin\alpha + y_0\cos\alpha)$ $\cdots\cdots ⑦$

ここで $K = \prod_{k=0}^{n-1} \sqrt{1+\delta_k^2}$ ，$\alpha = \sum_{k=0}^{n-1} \tan^{-1}\delta_k$

以上の変換を行なう際，始線の初期値 あるいは ΔX 未満を与えると，次に示す TAN(X)，TAN1(X) の計算ができる。

i) TAN(X)

変換の原点を実軸上の点 $(x_0, 0)$ に取ると，$A_0 = 0$

∴ ⑤' 式は $A_n = \sum_{k=0}^{n-1} \tan^{-1}\delta_k$

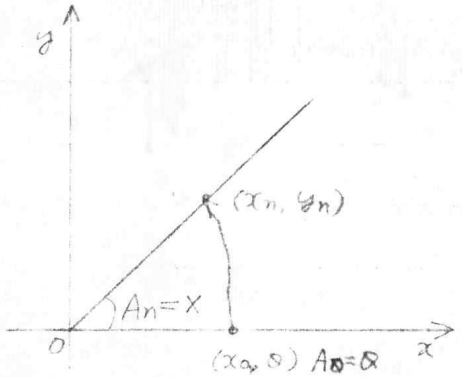
ここで変換後の偏角 A_n を与えられた値 X' とし，$X - \sum_{k=0}^{n-1} \tan^{-1}\delta_k \longrightarrow 0$ となる様にするには，上記の変換を行なった事に相当する。

右図より明らかなように ⑥，⑦ 式より

①，② 式の変換を行なった後の x_n, y_n を

$$\frac{y_n}{x_n} = \tan\left(\sum_{k=0}^{n-1} \tan^{-1}\delta_k\right)$$

$$= \tan(X) \quad \text{となる。}$$



フローチャート No.15 S6174 ～ S6218 の

$X - \sum_{k=0}^{n-1} \tan^{-1}\delta_k \longrightarrow 0$ を行なう疑似除算実行フロー，

S6211 ～ S6218 の ①，② 式の変換を行なうフローである。

このフローでは $y_k = -\delta_k' x_k'$ とし

①，② 式を

$$\begin{cases} x_{k+1} = x_k + \delta_k' y_k' & \cdots\cdots ①' \\ y_{k+1} = y_k' - \delta_k' x_k & \cdots\cdots ②' \end{cases}$$

と変形し，

$x_0 = 1$, $\delta_k = 10^{-k}$ の様に値を定め フロー を簡単にしている。
$\delta_k^2 \cdot x_n$ は [End→RS] を 2回 行う事により作り出している。

$$\sin x = \frac{(\delta_n/x_n)}{\sqrt{1+(\frac{\delta_n}{x_n})^2}}, \quad \cos x = \frac{(1/x_n)}{\sqrt{1+(\frac{1}{x_n})^2}} \quad \text{として 算出する。}$$

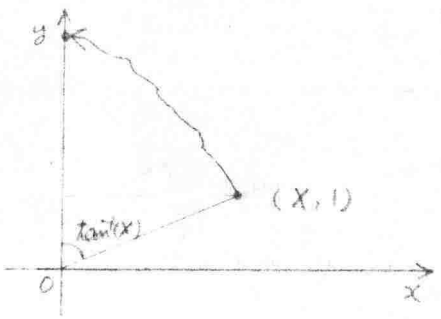
(ii) TAN^{-1}(X)

変換の原点を (X, 1) に取り ①', ②' 式 を $x_n \to \delta$ となる様に変換すると、右図より 明らかとなるが ⑥式より

$$\theta = K\left(X\cos\left(\sum_{k=0}^{n-1}\tan^{-1}\delta_k\right) - \sin\left(\sum_{k=0}^{n-1}\tan^{-1}\delta_k\right)\right)$$

$$X = \tan\left(\sum_{k=0}^{n-1}\tan^{-1}\delta_k\right)$$

$$\tan^{-1}X = \sum_{k=0}^{n-1}\tan^{-1}\delta_k \quad \text{となる。}$$

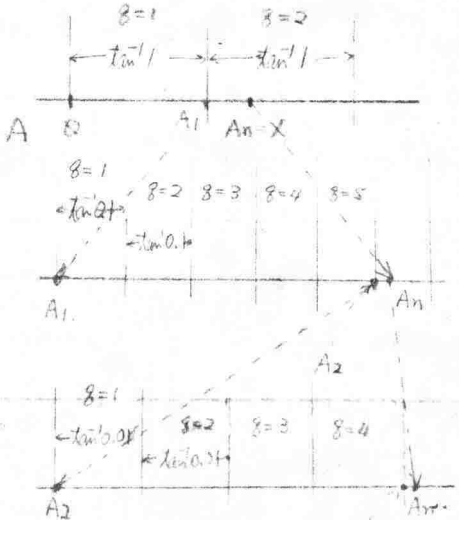


(演算に 次の3前に $\cos^{-1}X$, $\sin^{-1}X$ を $\tan^{-1}\frac{X}{\sqrt{1-X^2}}$ と変換する。

フローチャート No.14 05935～06121 で ①', ②' 式の変換を行なう。δ_k^2, x_k は As を 0, 2, 4, 6, 8 とし [Spl→RS] を 2回 行う事によって作る。
06123 以降の $\sum_{k=0}^{n-1}\tan^{-1}\delta_k$ の疑似乗算の実行フローにより 最後に $\cos^{-1}x = \frac{\pi}{2} - \sin^{-1}x$ として $\cos^{-1}X$ は求めている。

◎ 漸近方法の図解

TAN(X): 乗算後の収束適倍用 An は、与えられた値の X



1回目 $(\delta+1)\tan^{-1}1 > A_n \geq \delta\tan^{-1}1$ となる δ を求める。

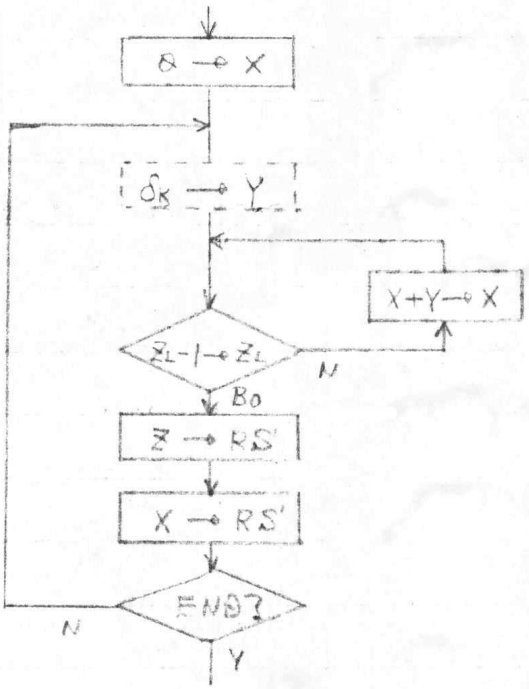
2回目 $\delta_k = \tan^{-1}0.1$ に変更して、同様に
$(\delta+1)\tan^{-1}0.1 > A_n \geq \delta\tan^{-1}0.1$ となる δ を求める。

以後 $\tan^{-1}0.01$, $\tan^{-1}0.001$, $\tan^{-1}0.0001$, ...

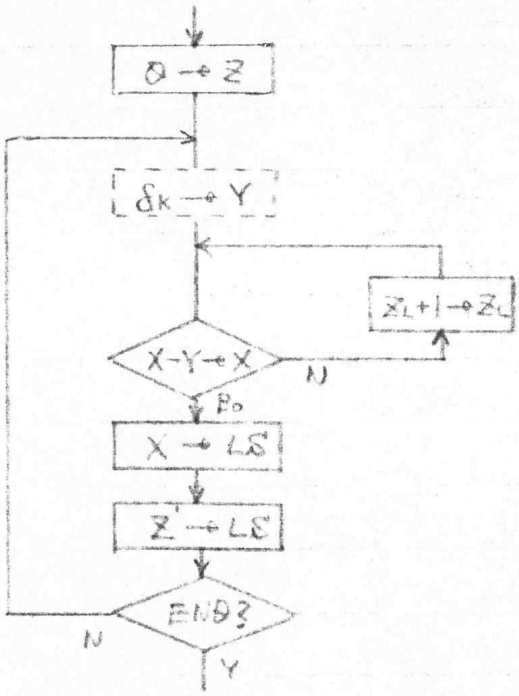
また δ_K を 5回変更して、各々の δ_K 時の減算可能回数 β を演算結果として、レジスタに記憶する。（第1段階の疑似除算） 図例では $\beta = 154$??

各々の δ_K 値について 8回 ①', ②' 式の変換を行なう。（第2段階の変形疑似乗算）

◎ 疑似乗除算について



疑似 乗算フロー



疑似 除算フロー

δK → Y の様にレジスタ内容を修正する命令が乗除算フロー内に作られたとき疑似乗除算フローと称している。 実際には くり返し加減算に類似し、疑似乗除算1回の演算は 一般乗除算1桁分に相当する為 上記 TAN(x) の疑似除算は 定数変更の手間を除けば 5桁の2数の除算1回の演算時間に匹敵する。 依って CORDIC を利用した超越関数計算は演算時間を大幅に短縮できる。 但し 他の計算法に比して数倍の ROM アドレス消費があるので、多通信及型の専用機としては最良の計算法とは必ずしも言えない。

(III) exp(x), Ln(x)

三角関数のCORDICの様に回転によって記憶する事はできない。

$x = \sum_{k=0}^{n} \theta_i \ln(\delta_k)$ — ① $\delta_k = 1+10^{-k}$ として $x - \sum_{k=0}^{n} \theta_i \ln(\delta_k) \longrightarrow 0$

となる各々 k についての θ をおめる。

①式を変形すると

$\exp(x) = \prod_{k=0}^{n} \delta_k^{\theta_i}$ — ② 式より おめられた θ より ②式を計算すれば exp(x) が計算できる。

又 $x = \prod_{k=0}^{n} \delta_k^{\theta_i}$ — ③ として $x - \prod_{k=0}^{n} \delta_k^{\theta_i} \longrightarrow 0$ となる

各々 k についての θ をおめる。

③式を変形すると

$\ln(x) = \sum_{k=0}^{n} \theta_i \cdot \ln(\delta_k)$ ③式より おめられた θ より ④式を計算すれば ln(x) が計算できる。

$\prod_{k=0}^{n} \delta_k^{\theta_i}$ の計算については

$$x_{k+1} = \prod_{k=0}^{k} \delta_k^{\theta_i} = \delta_k^{\theta_i} \cdot \prod_{k=0}^{k-1} \delta_k^{\theta_i} = \delta_k^{\theta_i} \cdot x_k$$

ここで $\delta_k = 1+10^{-k}$ に取ると $= x_k (1+10^{-k})^{\theta_i}$ となり

同一の k 値において $x_{k+1} = x_k + 10^{-k} \cdot x_k$ の乗算を θ 回

行なったものに相当する。

⑤exp(x) フロー

07102 ~ 07121 において $x < \ln 1.8$ とし ($\ln \delta_k = \ln 2$ の幅を許容を

向上する為) $x - \sum_{k=0}^{5} \theta_i \ln(\delta_k)$ を実行

07122 ~ 07223 で $\prod_{k=0}^{5} \delta_k^{\theta_i}$ を計算する。

⑥ Ln(x) フロー

07021 ~ 07031 で $x - \prod_{k=0}^{5} \delta_k^{\theta_i} \longrightarrow 0$ の θ をおめ

07032 以降で $\sum_{k=0}^{5} \theta_i \ln(\delta_k)$ 計算する。

◎ $\tan^{-1}$ において $\tan^{-1}10^{-k}$、ln において $\ln(1+10^{-k})$ を、漸近1回あたりの幅として選んでいるのは

$$\tan^{-1}(10^{-k}) > 10\,\tan^{-1}(10^{-(k+1)}) \quad , \quad \ln(1+10^{-k}) > 10\cdot\ln(1+10^{-(k+1)})$$

の関係が成立し 各K値における漸近の回数が 9回以内に抑えられるからである。 又、漸近の速度が 2分法（バイナリ・サーチ法）などよりも 高速である事によると思われる。

Kの値についても $\tan^{-1}$ では 0～4 ln では 0～5 であって N桁の演算結果を求めたいとき $N/2$ ないし $N/2+1$ 桁の値を取れば 誤差なく計算可能である。

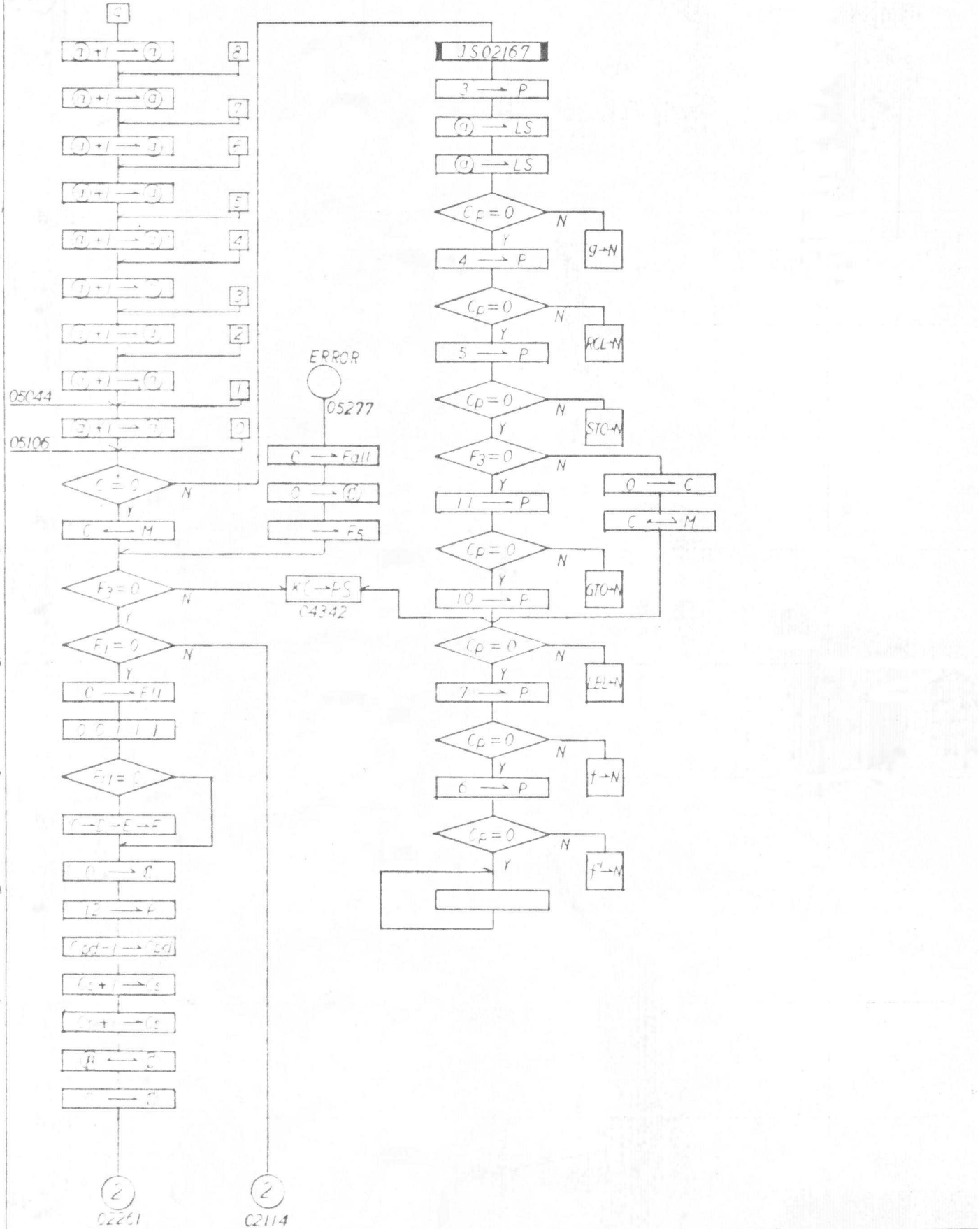
以上の記述に対して、二村先生 門田氏 発行の"科学電卓のアルゴリズム"，J.E. Meggitt "Pseudo Division and Pseudo Multiplication Processes" IBM J. APRIL 1962，一松信 "初等関数の数値計算" 教育出版，W.H. Specker "A class of algorithms for ln x …… and $\cot^{-1}x$" IEEE Trans. 1965，を参考にし、理解し易いように解説し（報告者自身理解する為）"モデル65" における実例を示した。

[6] CPUの汎用性について

"モデル65" に使用されているCPUは、モデル35, 45 の様な他の表示式関数電卓、モデル46 プリンタ関数電卓、ユーパル80 ビジネス電卓、統計電卓に使用されているものと同一であり プログラムROMの変更によって種々の用途がある。 但し、プリンタ、磁気カード 等の I/O 制御の為には I/O 制御用の NEAC M4 にある様な PUN, STU, BTC, BTD 的な単一命令しか出せず I/O 制御を含めて ROM で全て制御する事が不能であるので プリンタ、磁気カード I/O 毎に専用の IOC が必要となり、これらを別チップで別個に作らねばならぬ繁雑さがある。 又、ROM-レジスタ方式を採っているので 演算処理 データ転送については 便利であるが 演算速度が問題となるが これはクロック周波数を高くする事によってカバーしている。

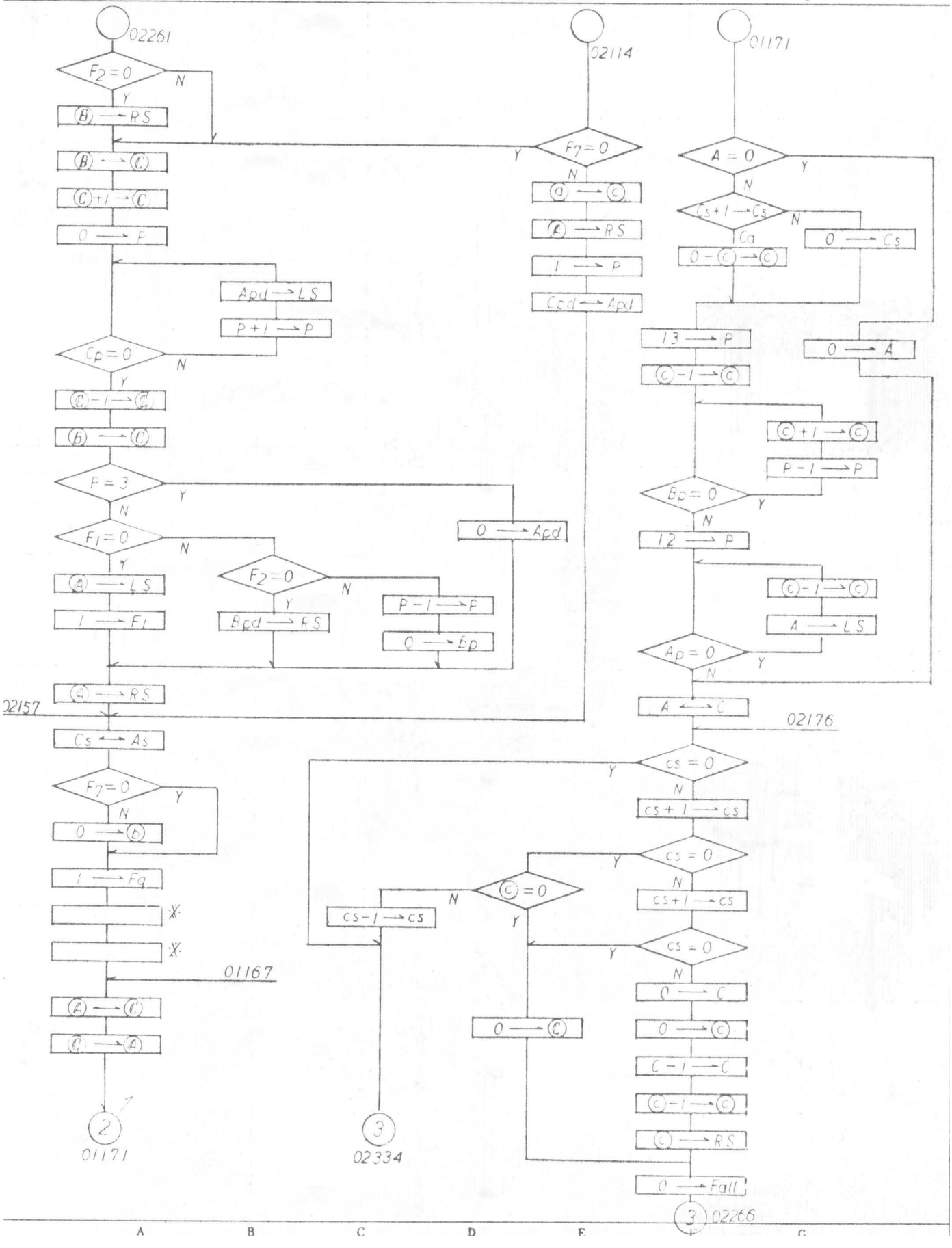
フローチャート

タイトル	年 月 日	版	承認	査閲	担当	年 月 日	版	承認	査閲	担当	登録番号
MODEL 65 No 1											参照番号
											作成者



フローチャート

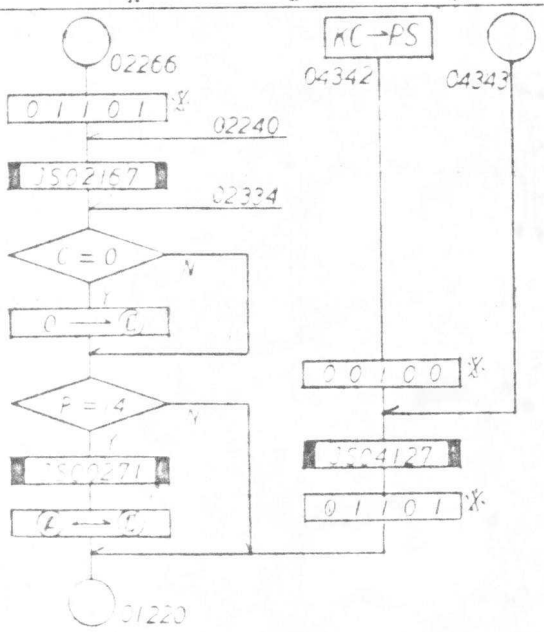
タイトル	年 月 日	版	承認	査閲	担当	年 月 日	版	承認	査閲	担当	登録番号
											参照番号
No. 2											作成者



フローチャート

タイトル	年 月 日	版	承認	査閲	担当	年 月 日	版	承認	査閲	担当	登録番号
No 3											参照番号
											作成者

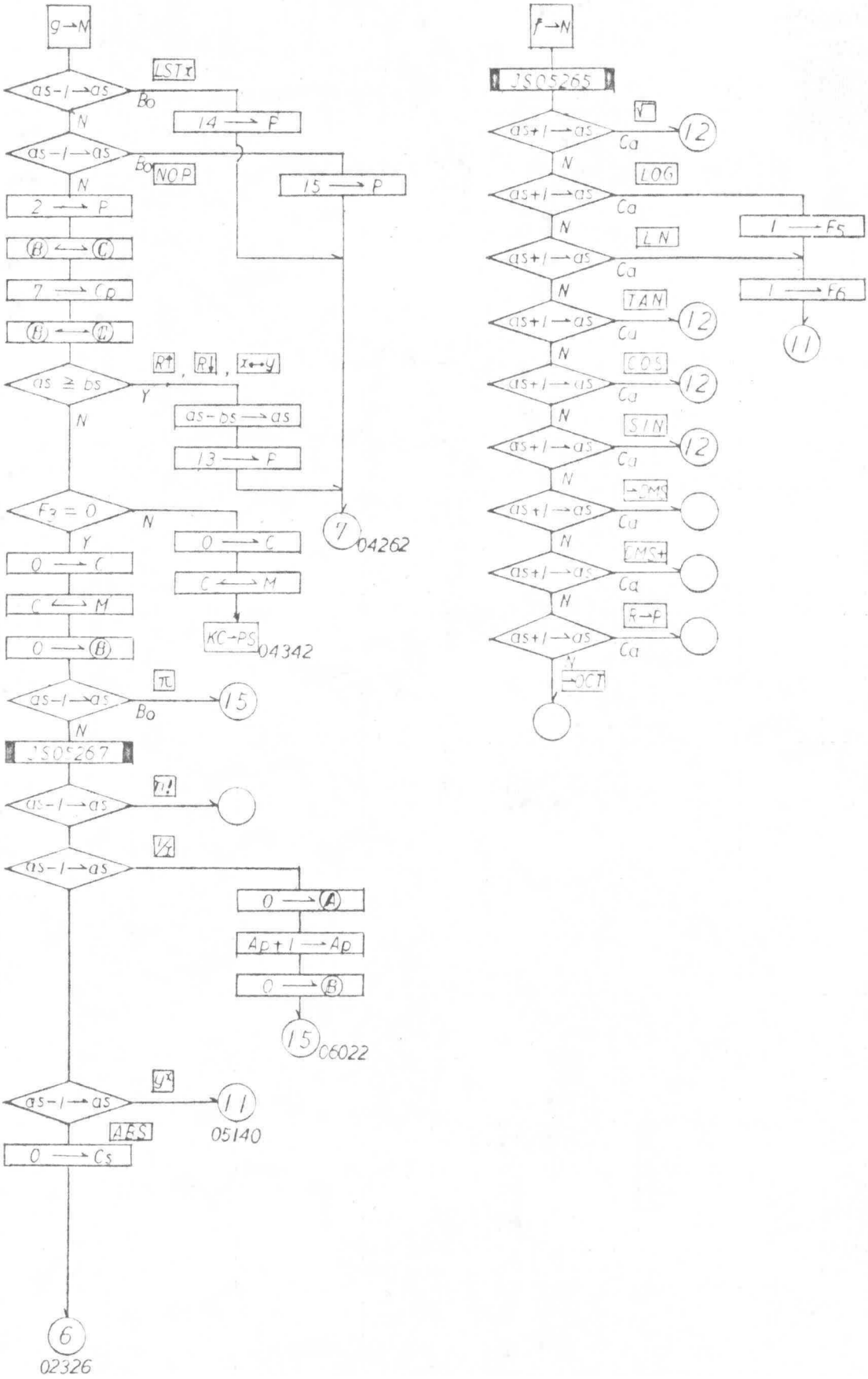
A B C D E F G



フローチャート

タイトル	年 月 日	版	承認	査閲	担当	年 月 日	版	承認	査閲	担当	登録番号
No 4											参照番号
											作成者

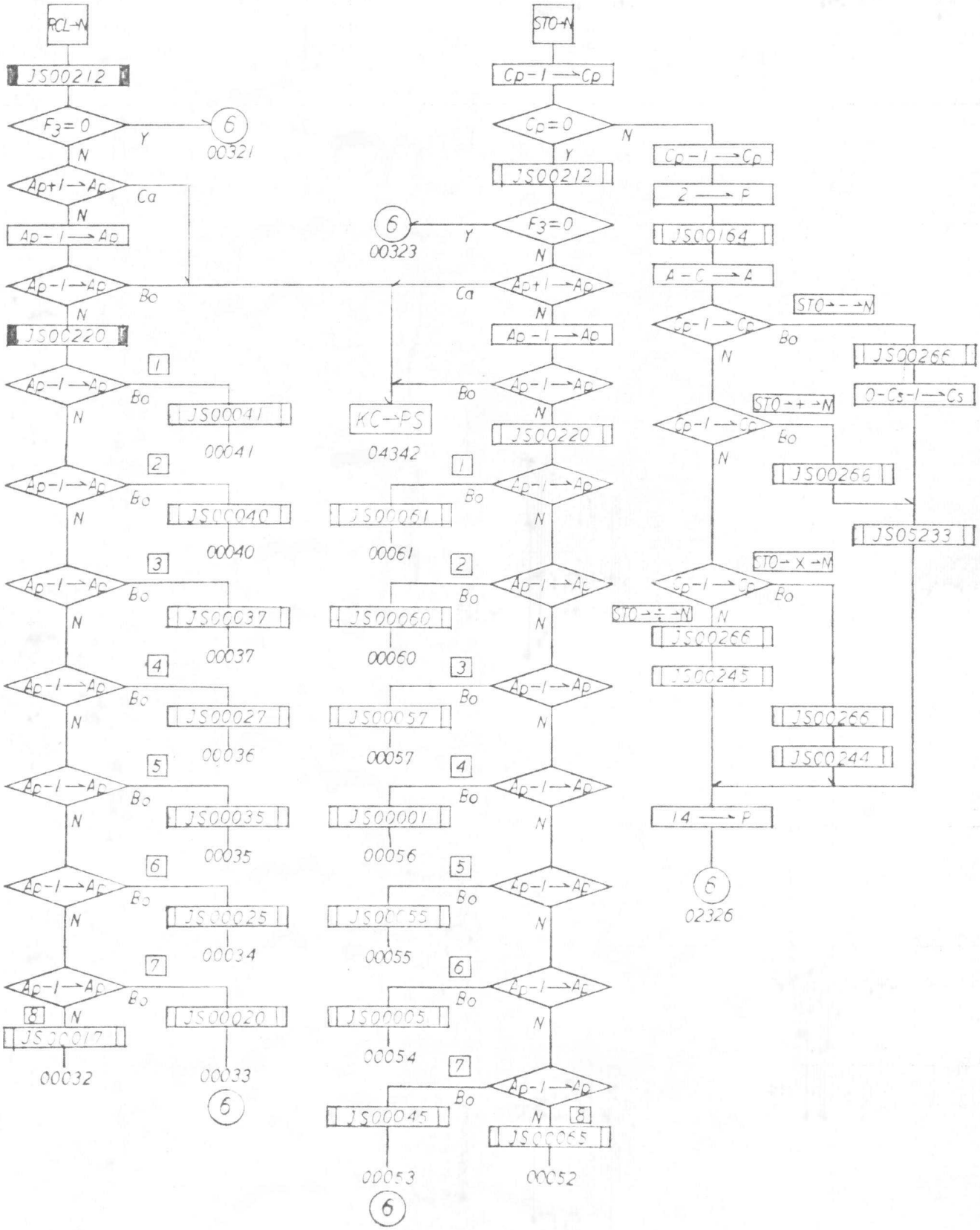
A B C D E F G



フローチャート

タイトル	年 月 日	改	承認	査閲	担当	年 月 日	改	承認	査閲	担当	登録番号
No 5											参照番号
											作成者

A B C D E F G



フローチャート

タイトル

年 月 日	版	承認	査閲	担当	年 月 日	版	承認	査閲	担当

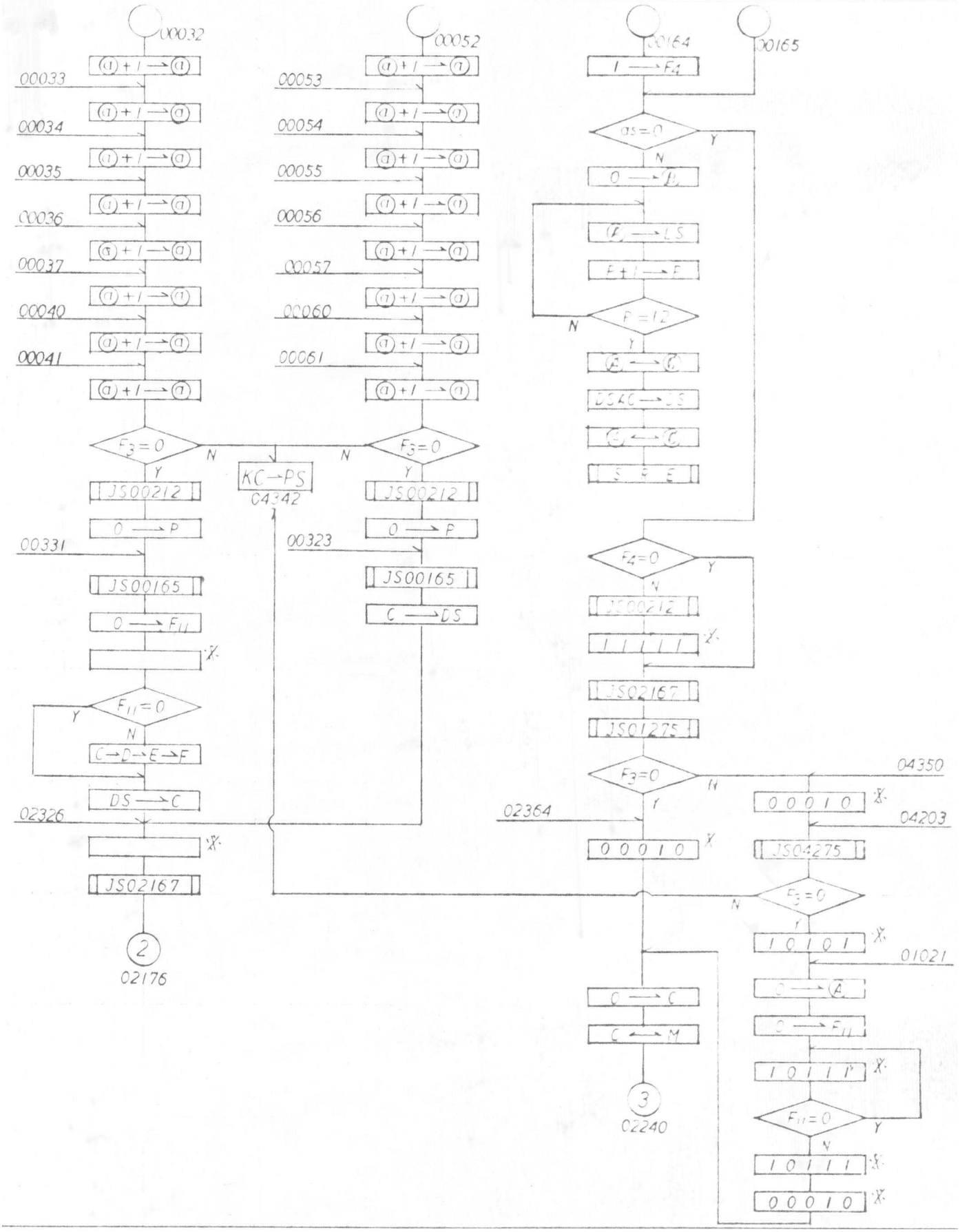
登録番号

事務番号

作成者

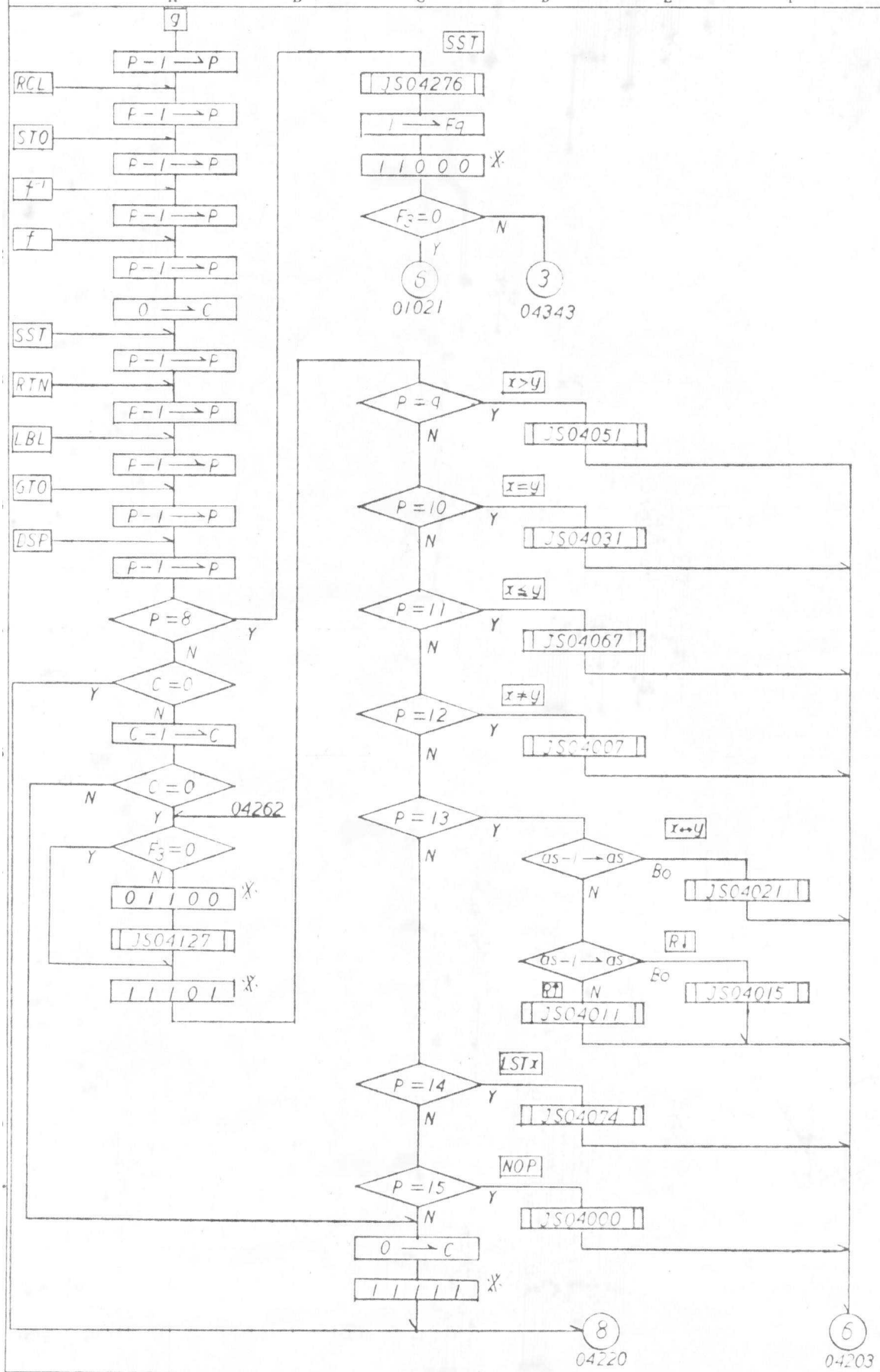
No 6

A B C D E F G



フローチャート

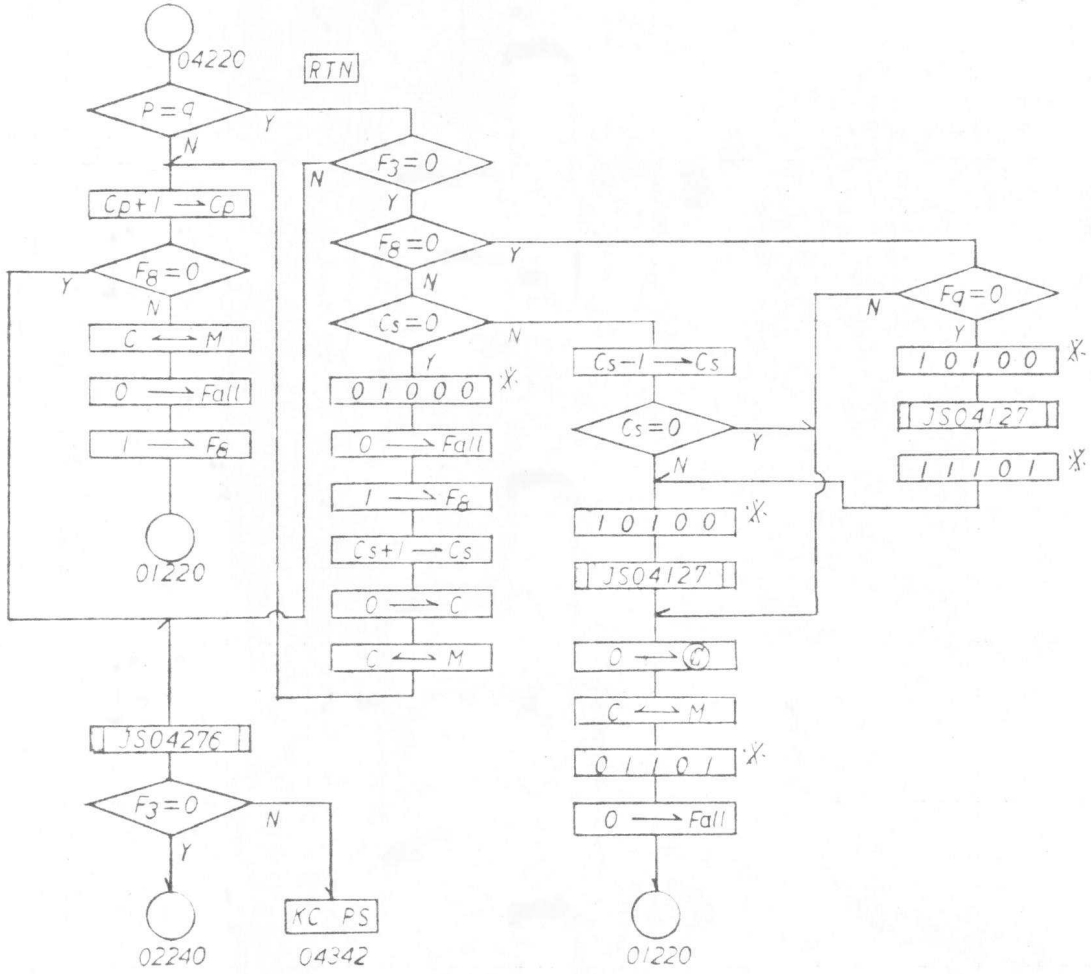
タイトル	年 月 日	版	承認	査閲	担当	年 月 日	版	承認	査閲	担当	登録番号 / 参照番号 / 作成者
No 7											



フローチャート

タイトル	年 月 日	版	承認	査閲	担当	年 月 日	版	承認	査閲	担当	登録番号
No 8											参照番号
											作成者

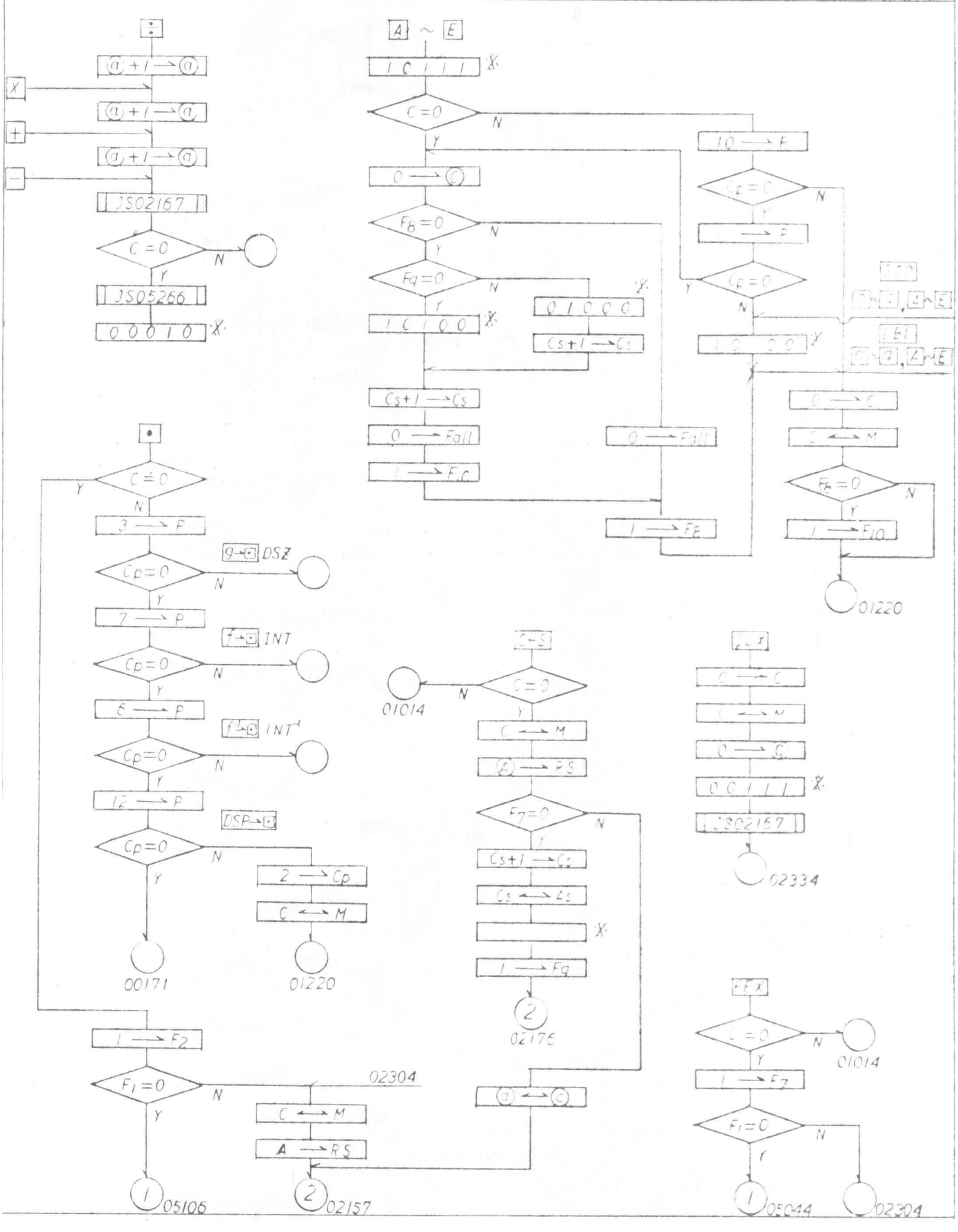
A B C D E F G



フローチャート

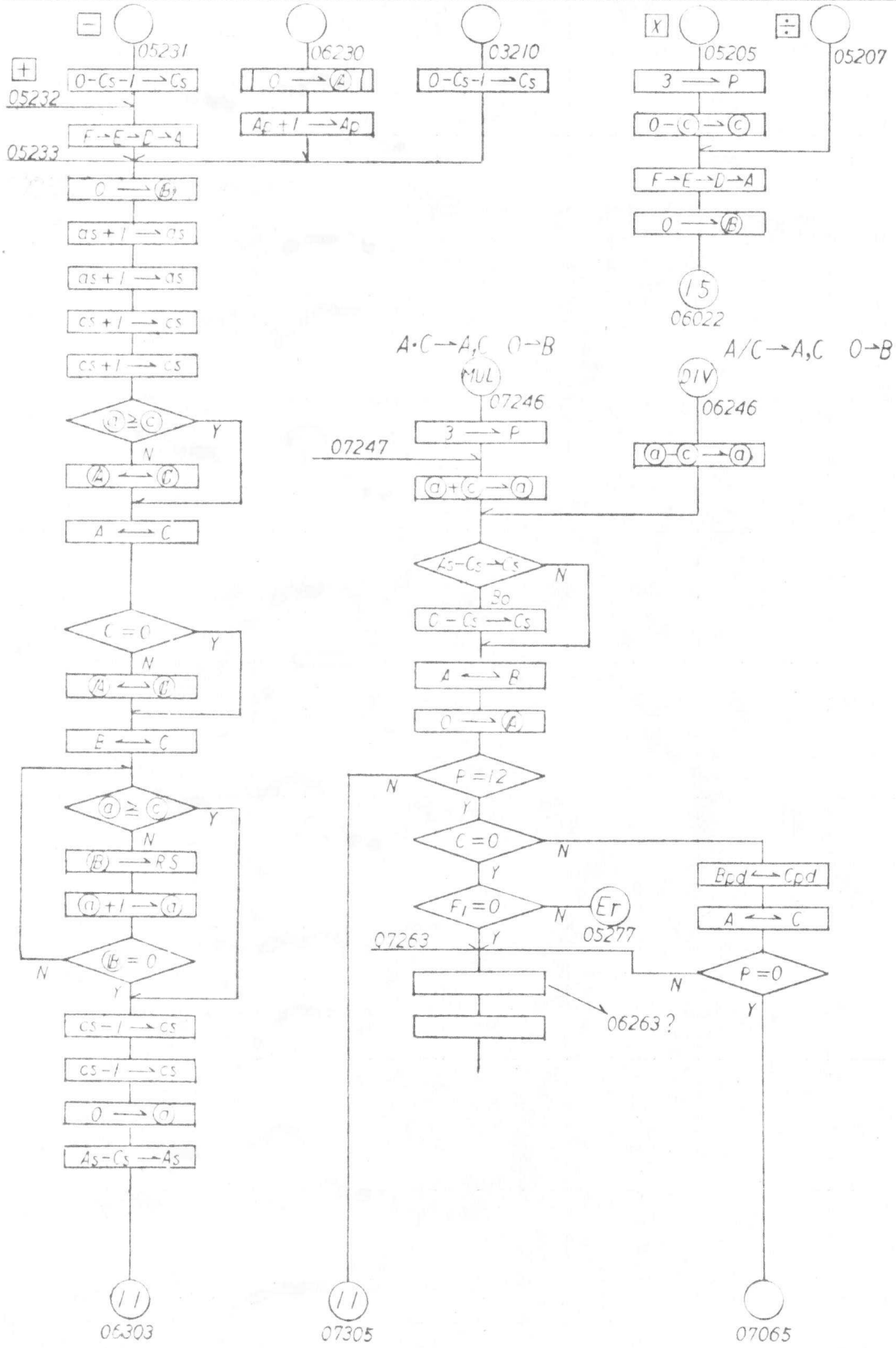
タイトル	年 月 日	版	承認	査閲	担当	年 月 日	版	承認	査閲	担当	登録番号
No 9											参照番号
											作成者

A B C D E F G



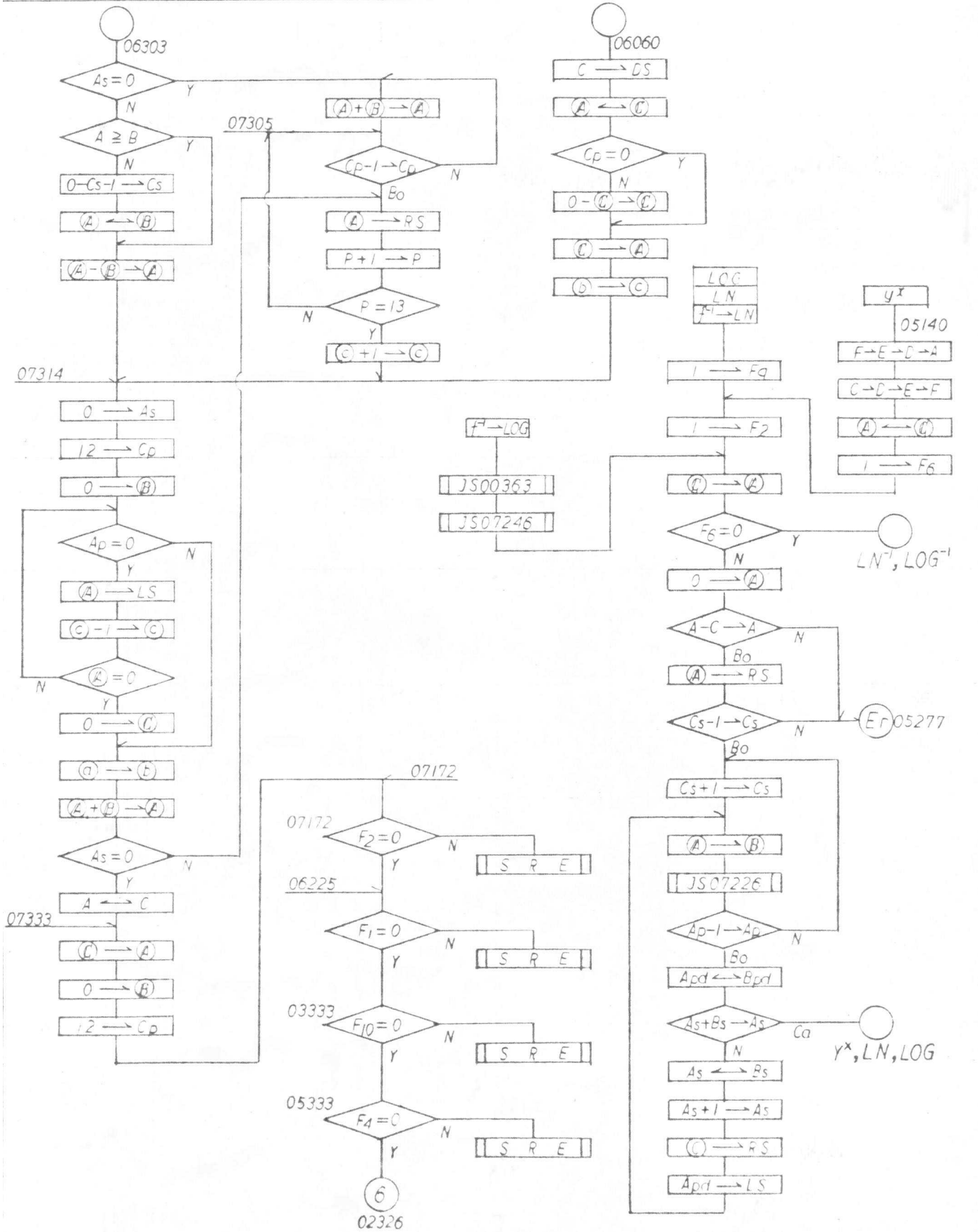
フローチャート

タイトル	年 月 日	版	承認	査閲	担当	年 月 日	版	承認	査閲	担当	登録番号
No 10											参照番号
											作成者



フローチャート

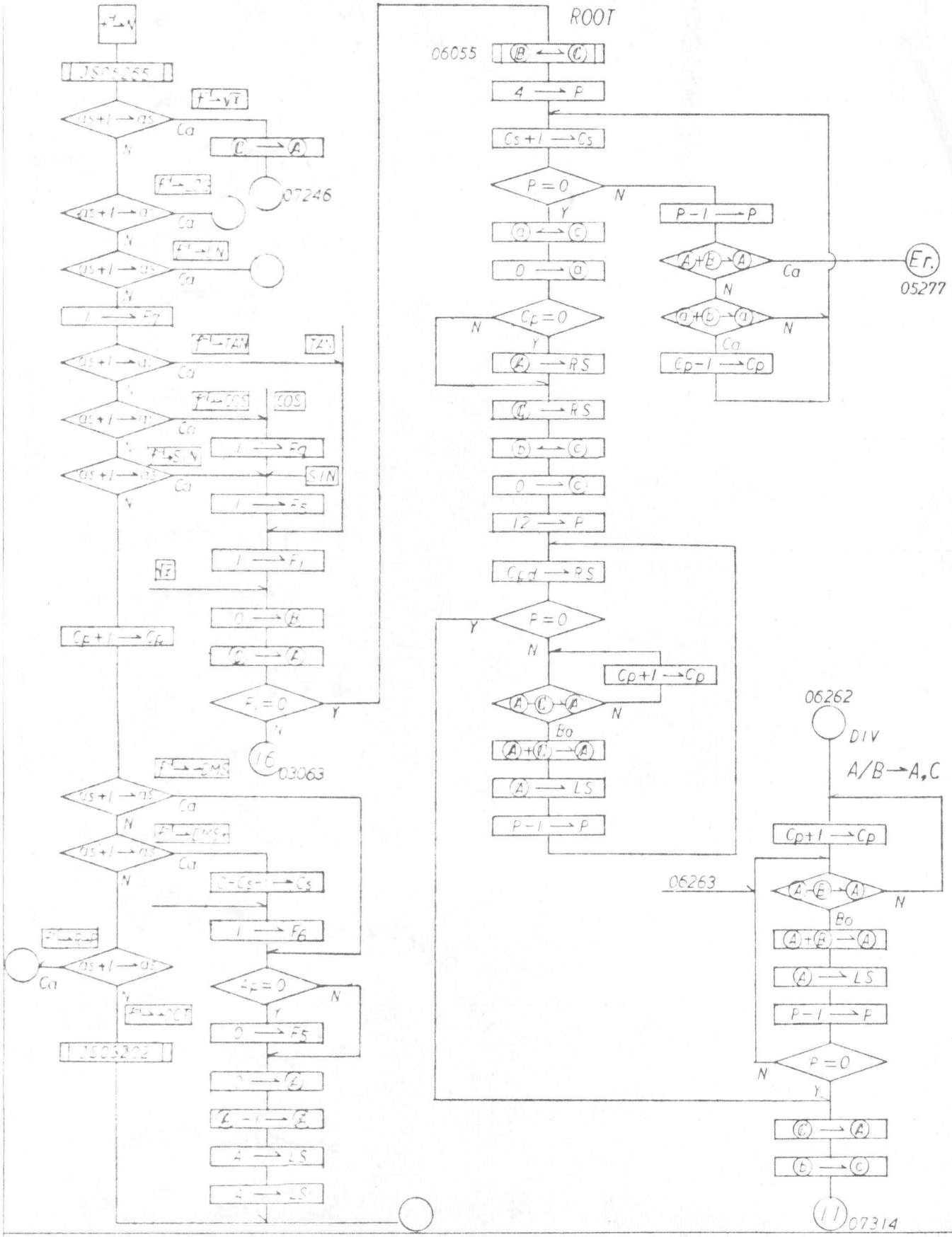
タイトル	年 月 日	版	承認	査閲	担当	年 月 日	版	承認	査閲	担当	登録番号 / 参照番号 / 作成者
No 11											



フローチャート

タイトル	年 月 日	版	承認	査閲	担当	年 月 日	版	承認	査閲	担当	登録番号
No 12											参照番号
											作成者

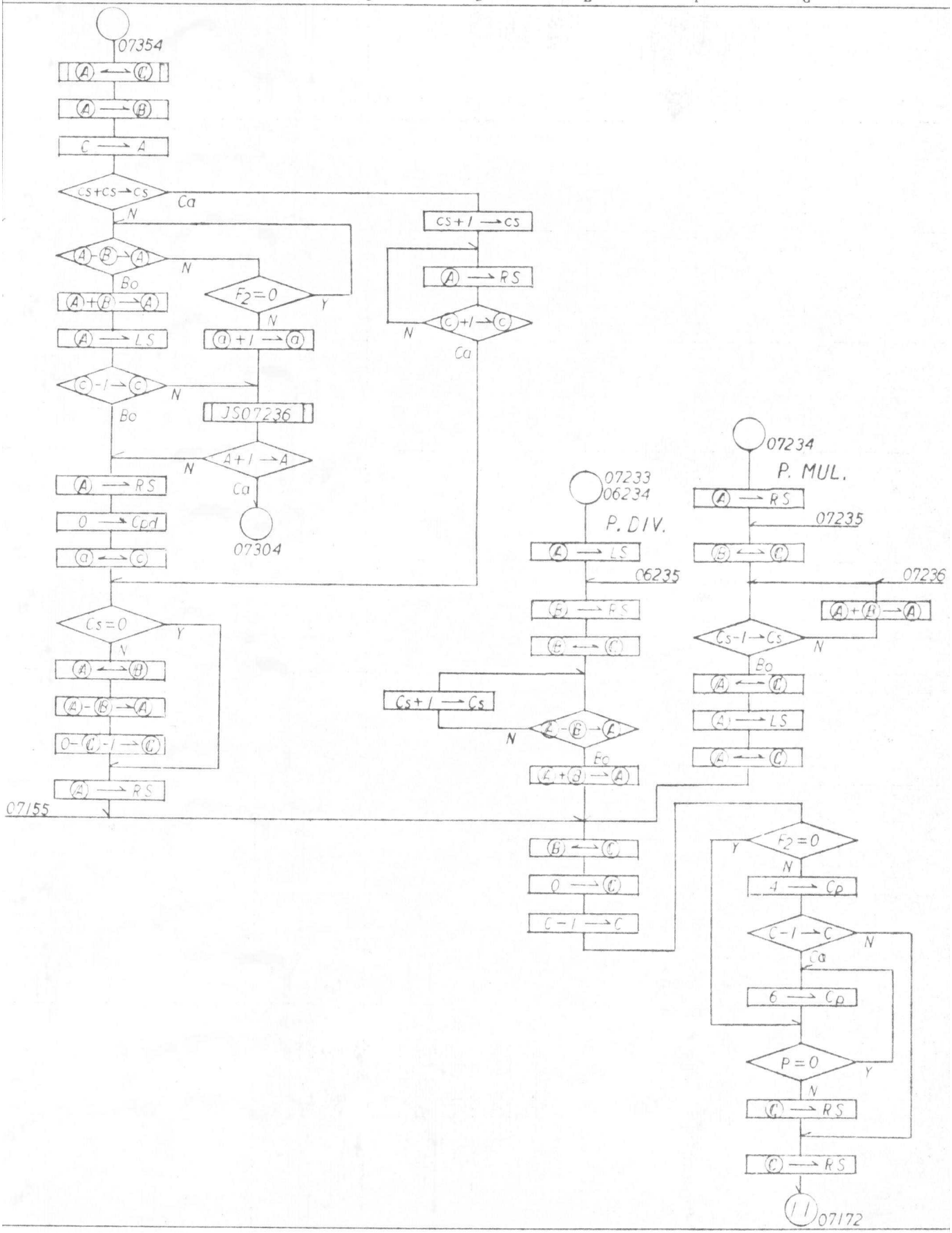
A B C D E F G



フローチャート

タイトル	年 月 日	版	承認	査閲	担当	年 月 日	版	承認	査閲	担当	登録番号
No 13											参照番号
											作成者

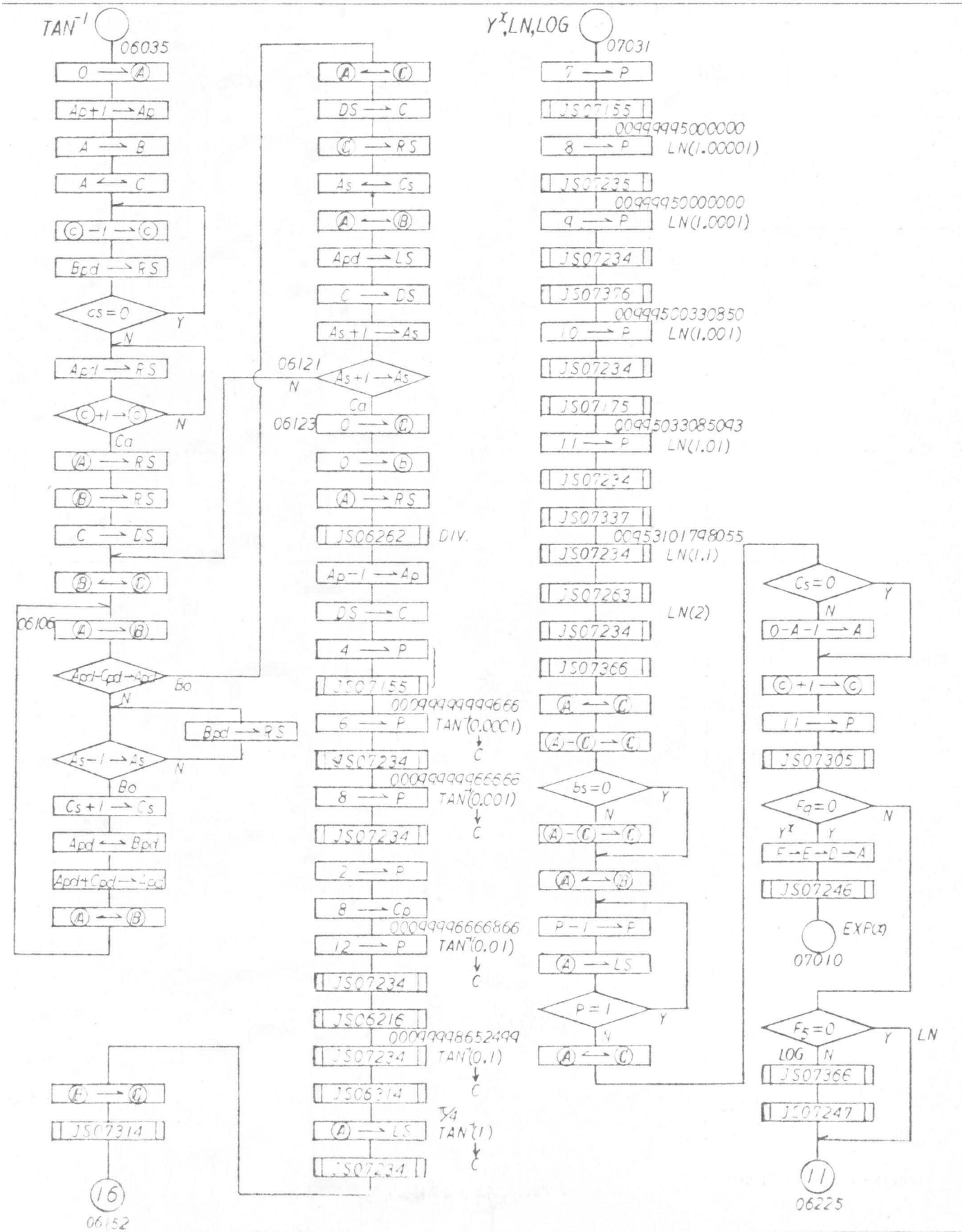
A B C D E F G



フローチャート

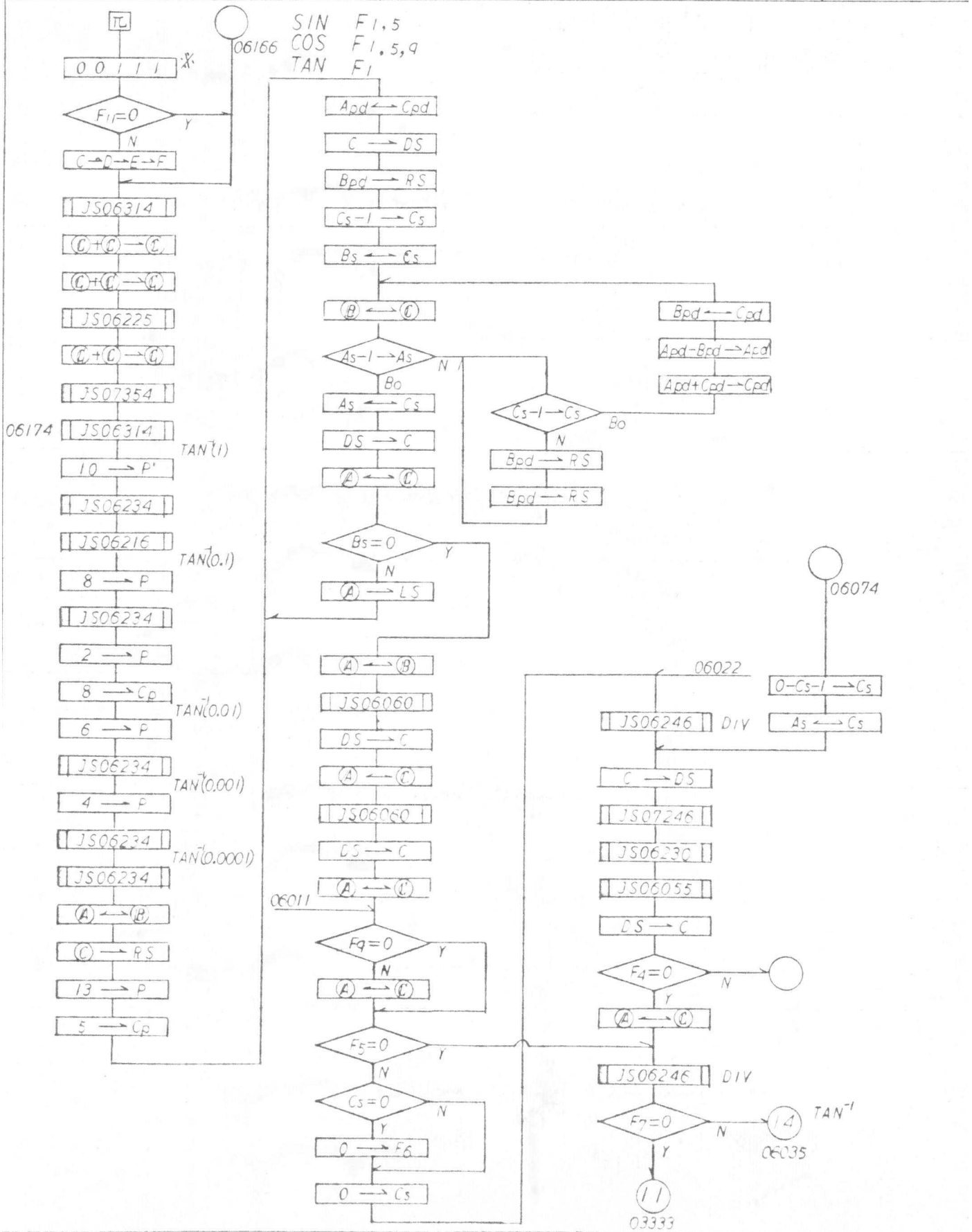
タイトル	年 月 日	版	承認	企間	担当	年 月 日	版	承認	企間	担当	登録番号
											参照番号
No. 14.											作成者

A B C D E F G



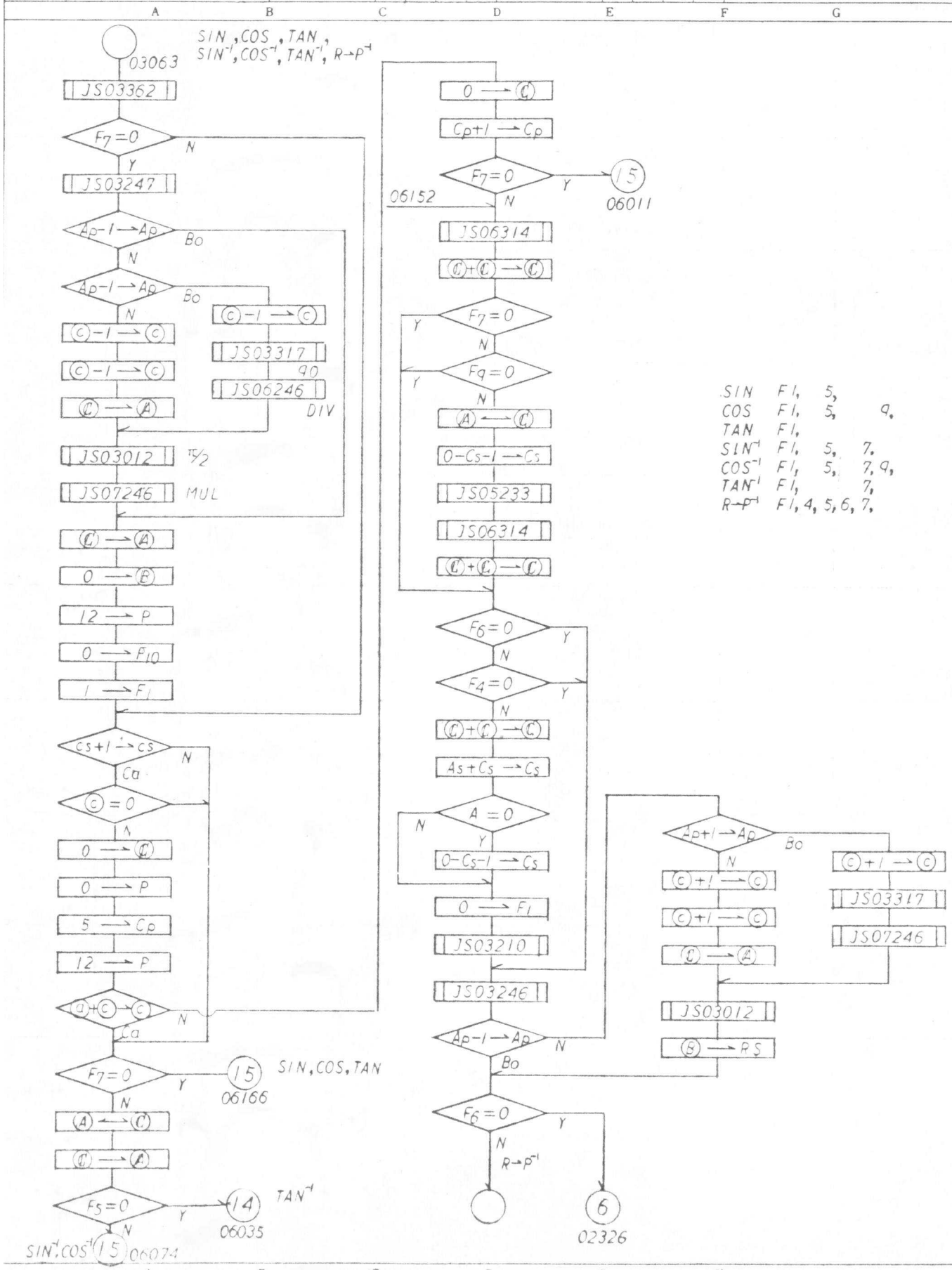
フローチャート

タイトル	年 月 日	版	承認	合間	担当	年 月 日	版	承認	合間	担当	登録番号
No 15											参照番号
											作成者



フローチャート

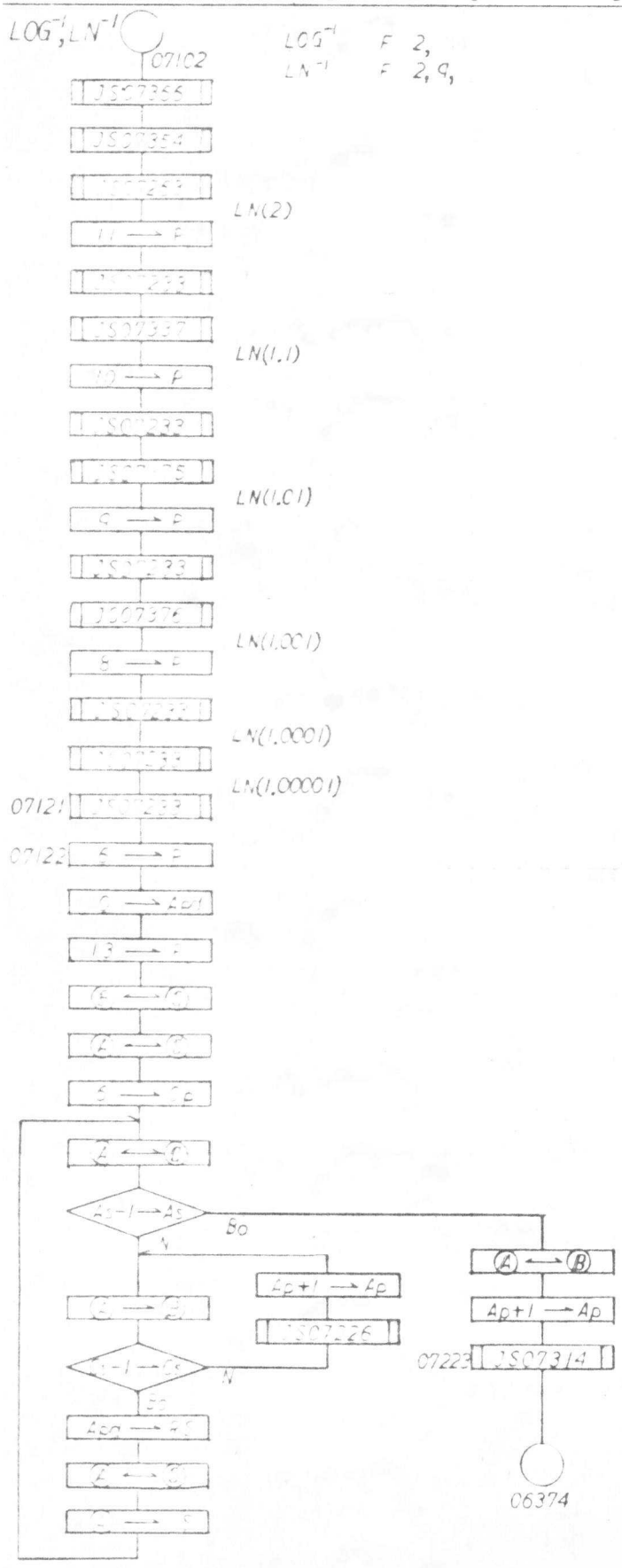
タイトル	年 月 日	版	承認	査閲	担当	年 月 日	版	承認	査閲	担当	登録番号
No 16											参照番号
											作成者



フローチャート

タイトル	年 月 日	版	承認	査閲	担当	年 月 日	版	承認	査閲	担当	登録番号
No 17											参照番号
											作成者

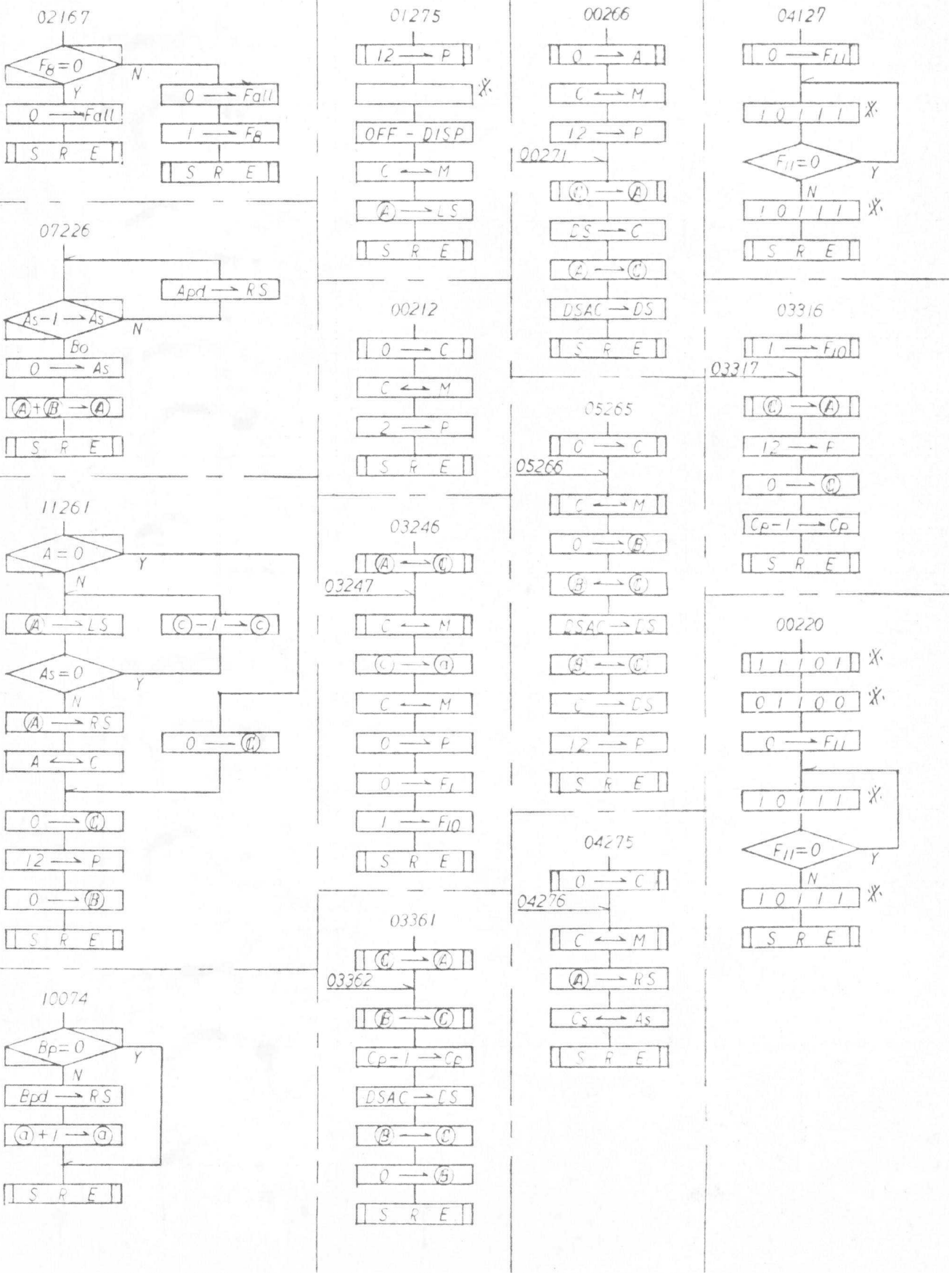
A B C D E F G



フローチャート

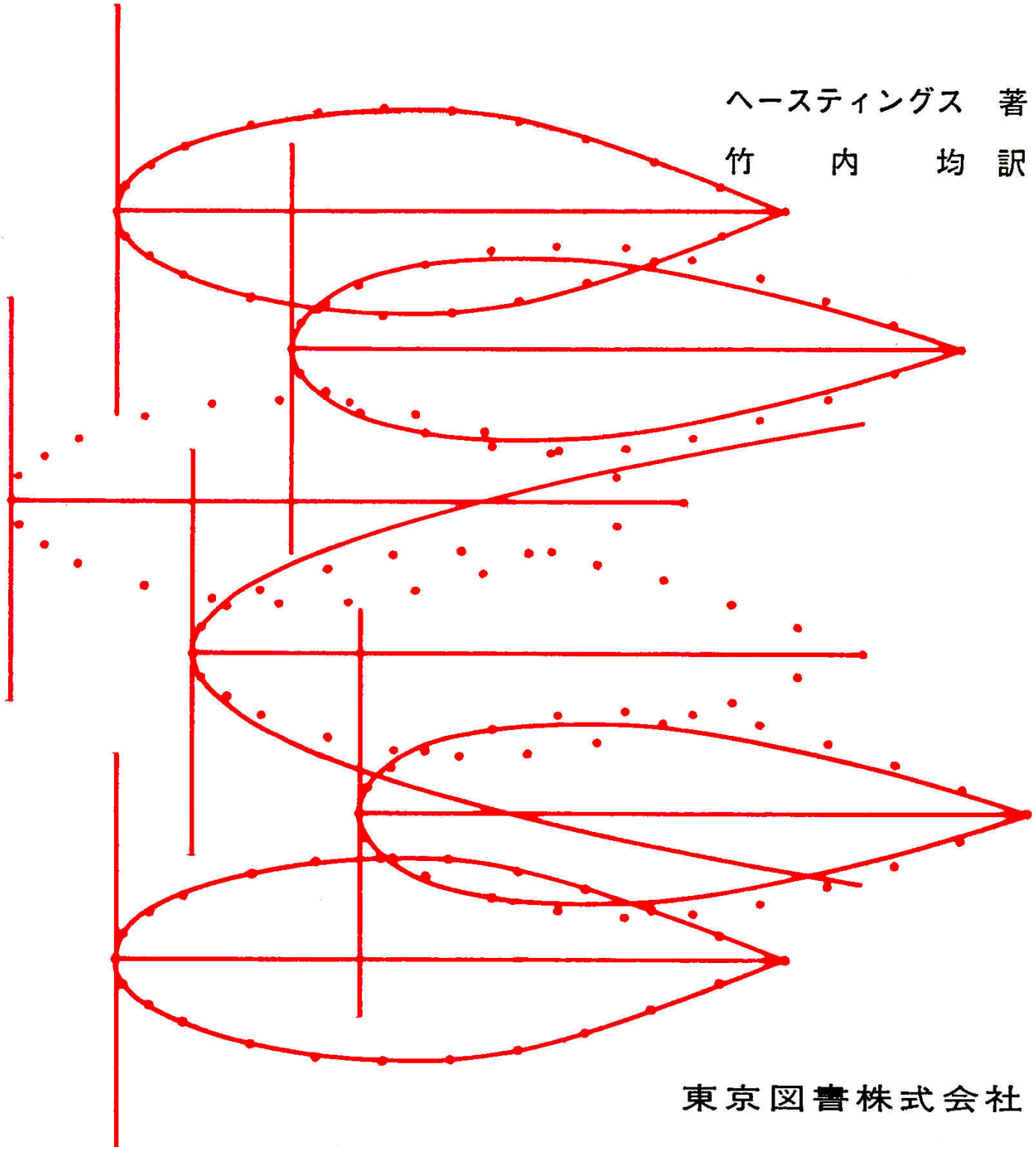
タイトル	年 月 日	版	承認	倉間	担当	年 月 日	版	承認	倉間	担当	登録番号
No 18											参照番号
											作成者

A B C D E F G



電子計算機のための 近似計算法

ヘースティングス 著
竹内均 訳



東京図書株式会社