U.D.C. 681.142 : 518.5

科学用パラメトロン計算機HIPAC 103

Parametron Computer for Scientific Use, HIPAC 103

萱島興三* 高橋延匡**
Kōzō Kayashima Nobumasa Takahashi
氏家一彬** 高須昭輔**
Kazuaki Ujiie Akisuke Takasu

内　容　梗　概

磁気テープ，ラインプリンタ，A-D，D-A変換器などを接続することのできるパラメトロン電子計算機 HIPAC 103 は，主として科学技術計算を目的として設計製作され，現在，日立製作所の科学用計算機の標準製品となっている。パラメトロン，トランジスタを主演算素子として使用した回路方式，高度の処理能力を有する計算機としての各種プログラム(HISIP 103，基本サブルーチン)，自動プログラム(HARP 103) などについて設計の目標とソフトウエアの開発の現状を述べている。

1. 緒　　言

HIPAC 103 は，当初関西電力株式会社において電力経済負荷配分装置(ＥＬＤ)を計画するにあたり，アナログとディジタルを組み合わせて能率よく計算を行なわせる方式を採用することとなったので，これに適合するディジタル形電子計算機として設計を開始した。また同時に単独に使用して高度の科学技術計算が行なえるようにという要求から，パラメトロンを主演算素子として使用し，磁気コア記憶装置を主記憶装置に，大容量の磁気ドラムを二次記憶装置に用いる方針をたてた。パラメトロンの回路設計については HIPAC 101 の経験をもとにしたが，演算速度と信頼度の向上について特に考慮した。システム設計については，磁気ドラムを経済的にかつ能率よく使用するために，ドラムに書き込まれたクロックパルスをもとにして全体のクロックを作り出すこととし，パラメトロンとドラム，あるいは磁気コアとの接続に独自の方式を案出した。また科学技術計算に不可欠といわれる浮動小数点演算については，できるかぎり経済的な方法で，けた落ちによって精度が失われることのないよう考慮した。インデックスレジスタに関する方式も，複雑な科学技術計算の処理に適するよう，個数は少ないけれども重複して修正のできる論理，累算器を使用しないで，記憶装置と情報の授受を行なう命令の用意など考慮されている。

ＡＤ変換器，ＤＡ変換器との接続については，計算機本体に大規模な切換回路を設けることは避け，外部にスキャナあるいはディストリビュータをおく方式とし，必要に応じて外部に適当な接続回路をおくことによって，任意の個数のアナログ量の授受が行なえるよう，スイッチの制御信号を設けた。また HIPAC 103 は，その演算装置の性能に見合うよう入出力装置として磁気テープ，高速度印刷機を接続できるよう設計されている。これらの機器は，入出力制御装置を介して本体との情報の授受を行ない，あるいは入出力制御装置単独でオフラインの動作をするようになっている。これにより，本体と直結あるいは非直結の状態でこれらの高速の入出力装置を働かせることができる。

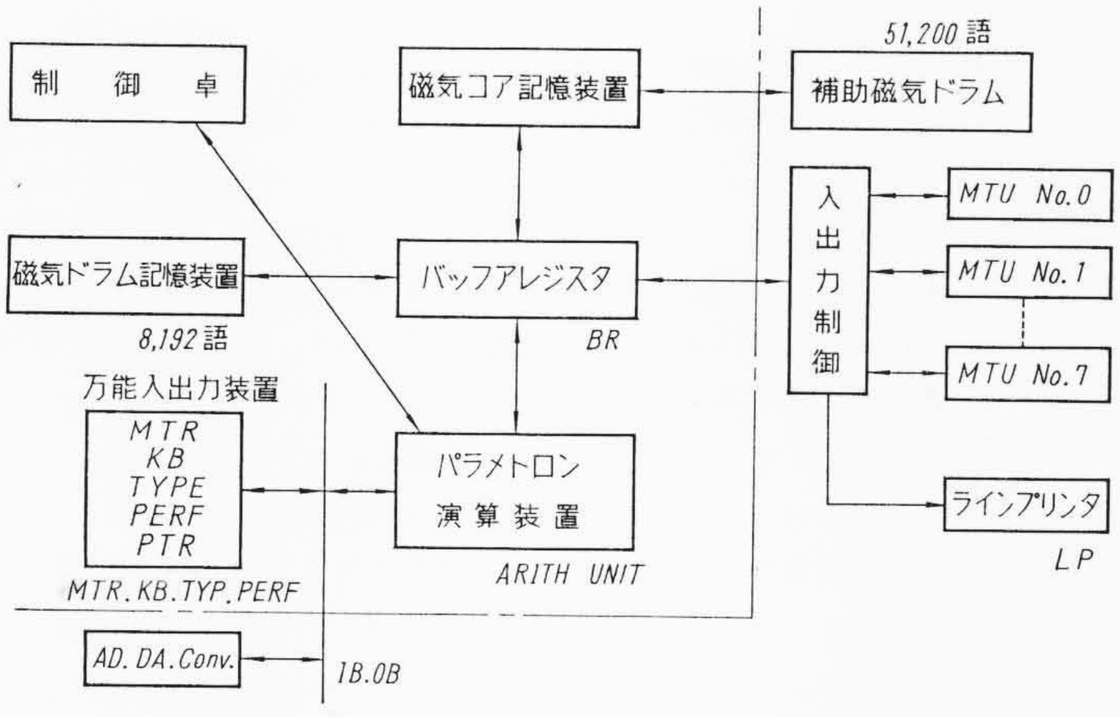
ディジタル計算機には，完備したプログラムの集積―ソフトウエアが，その能力を発揮するうえに不可欠のものであることは，近来常識になりつつある。この中には，プログラムを組み立てる構成要素として使用されるサブ・ルーチン群，計算機へのプログラムの読み込みを容易にする記号入力ルーチン，通常用いている言語-数式-を直接機械語に翻訳する自動プログラムなどがある。以下 **2** ～ **4** で計算機のハードウエアについて述べ，**5** ～ **7** でソフトウエアについて述べる。

2. HIPAC 103 の構成

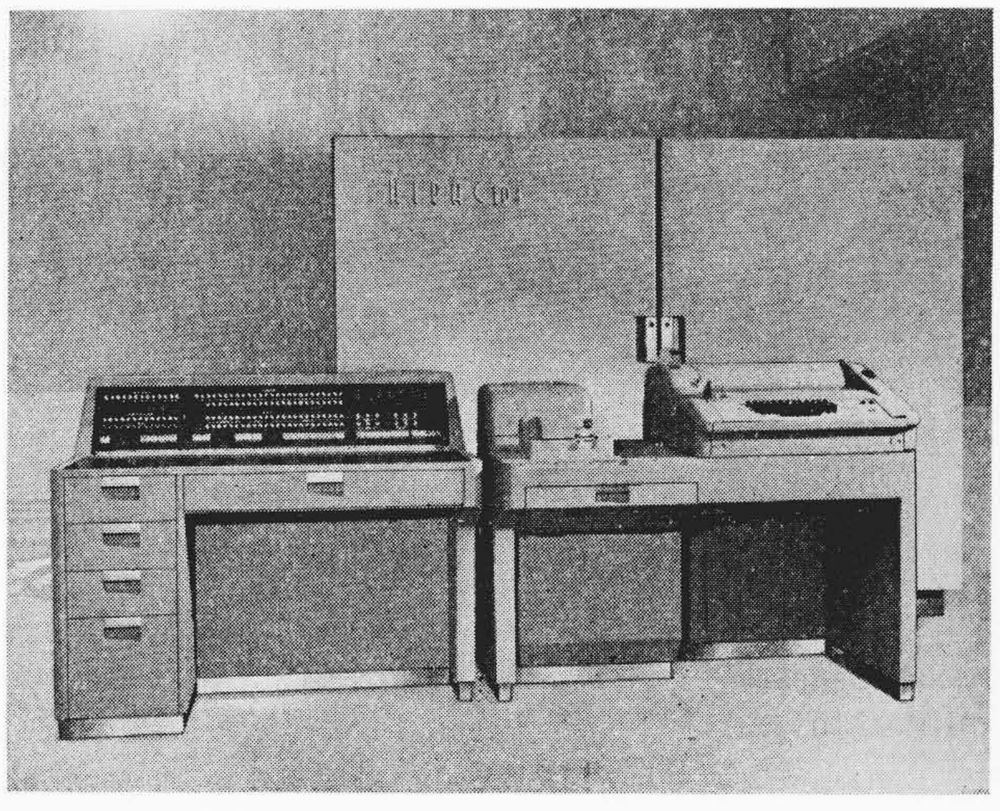
第1図に HIPAC 103 の構成ブロック図を示す。

図において，破線で囲った部分は計算機の主要部分で，計算機として動作させるに必要最小限度の性能を持っている。この部分の写真を**第2図**に示す。

計算機本体は，磁気コア記憶装置，バッファ・レジスタおよびパラメトロン演算装置より構成されており，きょう体の左側にパラメ



第1図　HIPAC 103 の構成ブロック図



第2図　HIPAC 103 の外観

* 日立製作所戸塚工場
** 日立製作所中央研究所

トロン回路が，右側に電源および磁気コア記憶装置，バッファ・レジスタが収められている。磁気ドラムは読み出し，書き込みの増幅器とともに別きょう体に収納されている。制御卓には，計算機の演算処理の動作を表示する各種レジスタ表示ランプ，演算を止めたりスタートさせたりする各種の制御用押ボタンスイッチなどが装備されている。1字ごとの信号の授受を行なう入出力装置としては，頁式プリンタ，さん孔機，機械式テープリーダ，光電式テープリーダ，キーボードなどがあり，これらは入出力卓にまとめて実装されている。AD変換器，DA変換器などは，外部よりパラメトロン演算装置に直接つながる。

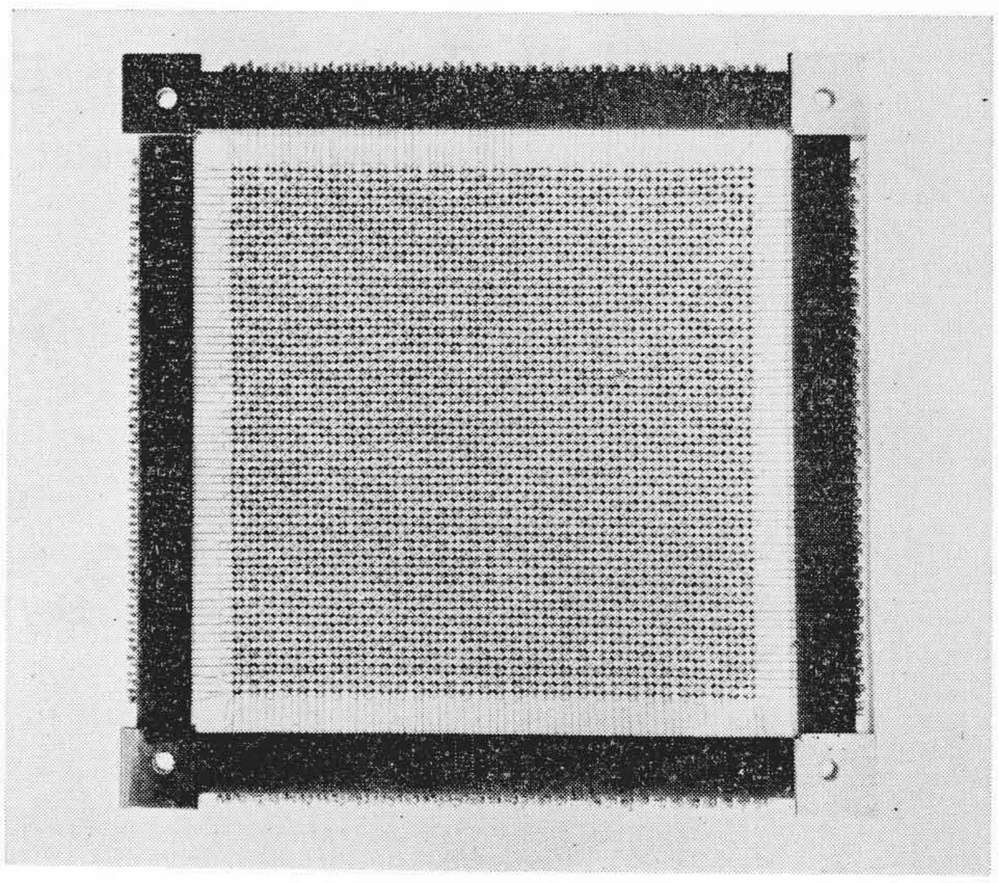
以上がベーシックと呼ばれる主要部分であるが，このほか，高速の入出力装置として，磁気テープ，ライン・プリンタなどを動作させるために HIPAC 103 ではバッファ用記憶装置を内蔵した入出力制御装置が用意されている。これには，磁気テープを8台まで，ラインプリンタを1台接続することができる。また，大容量のアクセス時間の短い記憶装置として，補助磁気ドラムを接続できるようになっている。

HIPAC 103 システムの主要性能をまとめると次のようになる。

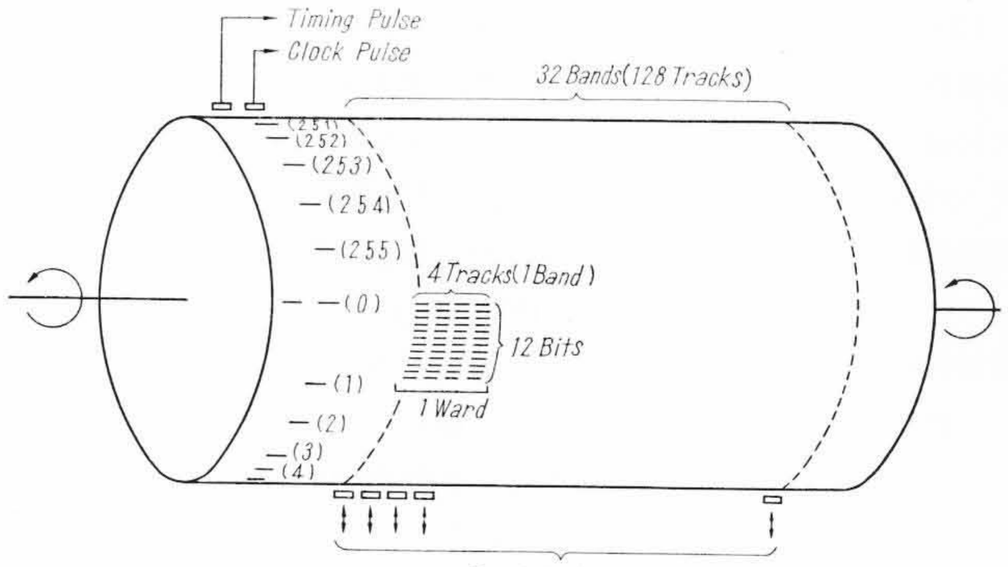
形　　式：	プログラム記憶式電子計算機
チェック方式：	入出力部におけるパリティチェック
回路方式：	パラメトロン，トランジスタ，ダイオードによる並列同期方式
主要演算素子：	パラメトロン　約 8,900 個
	トランジスタ　1,500 本
	ダイオード　2,500 本
	真空管　40 本
数値表現：	2進法，絶対値表示，固定小数点および浮動小数点
演算方式：	1½アドレス，1語2命令
命令の種類：	約110種類
主記憶装置：	コアマトリックス 1,024 語または 4,096 語，磁気ドラム 8,192 語
演算速度：	加減算 固定 650 μs　浮動 平均 1.3ms
	乗算 固定 2.3ms　浮動 平均 2.4ms
	除算 固定 8.1ms　浮動 平均 6.9ms
紙テープ入出力装置：	MTR 500字/分
	PTR 200字/秒
	さん孔タイプライタ 500字/分
	接続可能台数各1台
ラインプリンタ：	タイプホイール式
	数字，英字（大，小文字），記号計 94 種
1分間の行数：	300行（最大）
	接続可能台数 1台
磁気テープ装置：	テープの種類 幅½インチ，長さ 3,600 フィート
	記憶容量 700,000 語/リール
	クロック周波数 12 kc
	接続可能台数 8台
その他接続可能装置：	AD変換器 1チャンネル
	DA変換器 1チャンネル
	補助磁気ドラム 51,200 語

3. 記憶装置および演算装置

HIPAC 103 においては，主記憶装置として磁気コア記憶装置と，磁気ドラム記憶装置とを併用している。磁気コア記憶装置は読み出し書き込みに際して待時間が短いから，個々の演算に際し命令あるいはデータを貯蔵するのによく，磁気ドラム記憶装置は大量のデータ，あるいは一連のプログラムを一時貯蔵しておき，実際に演算を行なうときにブロックとしてコアに移してから処理するのが有利である。



第3図　磁気コア・プレーン

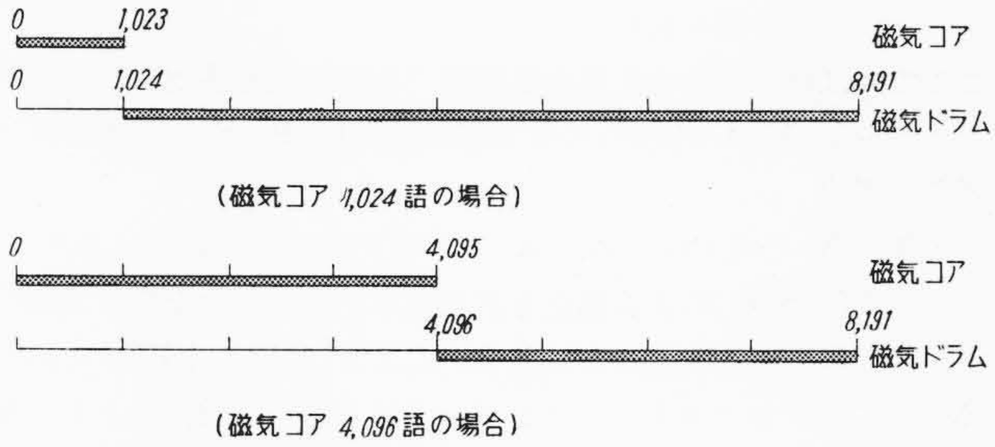


第4図　磁気ドラムの構造

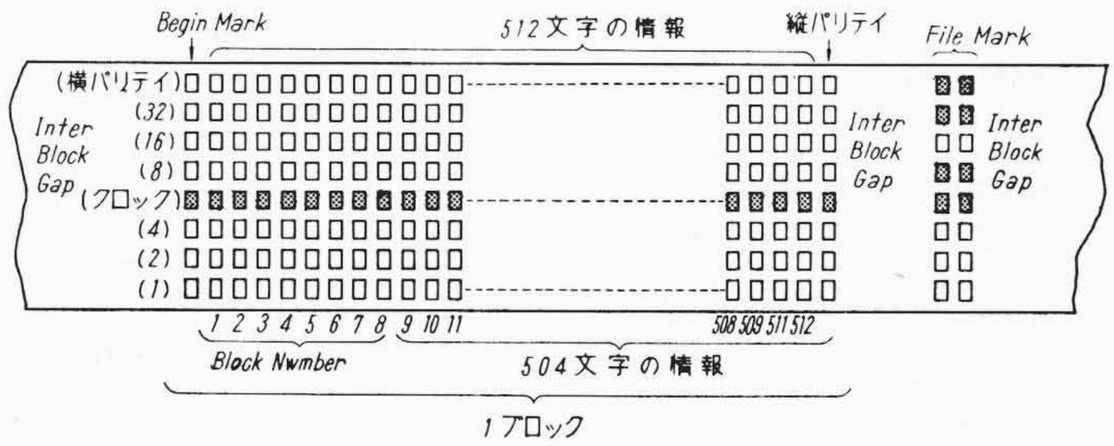
磁気コア記憶装置は，**第3図**に示すような，外径 0.05″ のコアを 64×64個，二次元に配置して編線を施したプレーンを積み重ねた構造をもち，コア1個に1ビットの情報を記憶させるものである。1語 48 ビットは 12 ビットずつ4回に分けて読み出し書き込みが行なわれる。1,024 語の場合は 12 枚，4,096 語の場合は 48 枚のプレーンを用いる。

磁気ドラム記憶装置とは，回転する円筒の表面に磁性材料の薄膜を作り，この上の磁化の状態を近接して設けた磁気ヘッドにより変化させ，あるいは変化を検出して必要な情報の書き込み，読み出しを行なう機能を持つものである。**第4図**にその構造を模型的に示した。1語 48 ビットは，12 ビットずつ4トラックに分けて記憶される。ドラム一周には 3,072 個の記憶位置があり，12 ビットずつ区切って 256 語が4トラックで記憶されることになる。記憶位置を定めるには，あらかじめ特定のトラックに書かれているタイミングおよびクロック信号を用い，タイミングパルスの位置を基準にしてクロックパルスをカウンタで数え，0～255 のカウンタの指示を読み取っている。4トラックで構成される 256 語をバンドと呼んでいる。128 個のトラックは，4個ずつの 32 個のバンドに分かれ，256×32＝8,192 語の記憶ができることになる。

記憶装置のアドレスとして，0～8,191 番地を命令で指定できるがこのうち実装されている磁気コアがこの番地の初めの部分を占め，



第5図　磁気コアと磁気ドラムのアドレス



第6図　磁気テープ上の情報配列

残りは磁気ドラムになっている。磁気ドラムにも，0～8,192 番地のアドレスがついているが，磁気コアと重複する初めの部分は1語単位で取り扱うことができない。**第5図**は磁気コアと磁気ドラムのアドレスのつけ方を示す。

図において太線で示した部分は1語単位の命令で取り扱える内部記憶装置，細線で示した部分はブロックとしてでないと取り扱えない補助記憶装置である。ブロックとして取り扱う場合は，磁気コアとの間で情報の授受を行ない，最大の語数は 256 語である。

4. 入出力制御装置

HIPAC 103 では，磁気テープ，ラインプリンタを本体につないで動作させるために，入出力制御装置が用意されている。ここには64語(512字) の磁気コアを使用したバッファ記憶装置があり，比較的おそい入出力装置の動作を本体の演算処理と併行して行なわせ，全体として演算速度を向上させることを目的としている。入出力装置(磁気テープ，ラインプリンタ)と，バッファ記憶装置との間の情報授受の動作のときには，入出力制御装置は本体から指令を受けとると，以後は入出力制御装置だけで処理が行なわれ，本体は次の演算に移る。

磁気テープ上では，ブロックを単位として情報を取り扱う。磁気テープ上の情報配列を模型的に示すと**第6図**のようになる。

図に示すように，1ブロックは64語 (512字) の情報と，これに付随した信号から成っている。ブロックとブロックの間に Inter Block Gap と称して信号を全然書き込まない領域を設ける。これは磁気テープのスタート，ストップの速度変動時に誤った信号を読み取らないようにするためである。1ブロックの先頭の1文字は，Begin Mark と呼び，読み捨てられる。最後の1文字は，縦パリテイビットとして，誤りの検出に用いられる。紙テープのスプロッケットに相当するものとしてクロックがある。クロックの上下にそれぞれ3および4トラックの記憶位置があり，最上位のトラックは，紙テープ同様横パリテイビットとなっている。

ラインプリンタは，120 字分 94 種類の活字を植え付けた印字胴を一定速度で回転させ，固定して設けられたハンマーで選択的に活字をたたいて印字させる，いわゆるタイプホイール式のものを用いている。この印字コードは，頁式のものと合わせてあり，上，中，下段のシフトも制御装置の機能によって同様に処理される。ラインプリンタと頁式プリンタとの相違は，前者が1行分の情報を単位として処理する点にある。このため，入出力制御装置のコアバッファに1行を単位とする情報をたくわえ，頁式プリンタの場合と同様に，コントロールコードの使用により，フォーマットの制御を行なっている。

すなわち，“LF”，“CR”は印字に際し，バッファ記憶装置の中でコントロールコードとして取り扱われ，“LF”のコードが検出されるとそこまでの文字を1行分として印字したのち印字紙が改行される。紙送りの行数はプログラムテープによつて制御される。“LF”，“CR”と続いて現われた場合は，有効な情報の終わりを意味する。横方向のフォーマットを作るために 120 字分のパッチボードが用意されている。これには，DRIVER のハブと，MAGNET のハブとがあり，前者はコアバッファから読み出される情報1行分のフィールドに対応し，後者は 120 字の印字位置に対応する。コントロールコードを除くすべての信号は順次このフィールドに詰めて入れられ，パッチボードの配線によって指定された印字位置のマグネットを働かせる。縦方向のフォーマットの制御には，別に設けたプログラムテープを用いる。プログラムテープとしては12単位の紙テープを使用し，これをループ状にはり合わせてプログラムテープ機構にセットする。この上にさん孔された穴は，改行の動作に同期してブラッシュで読み取られ，マーク信号がくるとそこで改行の動作を止める。したがって指定されたチャンネルの穴から穴までのフィードホールの数に等しい行数だけ改行が実行される。

初めに述べたように，磁気テープ，ラインプリンタなどの動作はバッファ記憶装置を介して行なわれるので，本体と切り離した状態で，これらを独立に動作させることが可能である。特に磁気テープに印字用のフォーマットに従って書き込まれた情報をラインプリンタで印字することは容易に行なえる。このようなオフラインの操作や動作内容の監視などの目的に制御パネルが用意されている。

5. 記号入力ルーチン (HISIP 103A, B)

電子計算機を使って問題を処理する場合，最終的には計算機のもつ命令（機械語と呼ばれる）によって，その問題を記述することが必要であるが，プログラマが直接この機械の言語を使って，プログラムと取り組むということは，特殊な場合を除いて現在ほとんど行なわれていない。その理由は機械語がわれわれの通常使用しているものから縁の遠いものであって，（1）記憶に不便であること，また（2）記憶装置と演算装置との間で情報の授受を行なうに際し，記憶装置の位置を示す番地（4けたないし5けたの数字で表わされている）を逐一プログラマが記憶しておかなければならないことによる煩雑さのゆえである。これらの理由により，機械語によるプログラミングは，プログラマに過大の負担をかけるばかりでなく，それに要する時間も長い。さらにプログラムの中に，誤りの混じる可能性が大となり，その誤りを正すこと（この操作をデバッギングと呼ぶ）が，またプログラマの大きな仕事となる。この弊を除くため機械語の代わりにわれわれに記憶しやすい言語を設定し，プログラマはその設定された言語によってプログラムを組めば，あとは計算機が自動的にこれを機械の言語に翻訳して計算を遂行してくれるような自動プログラミングシステムが考えられている。

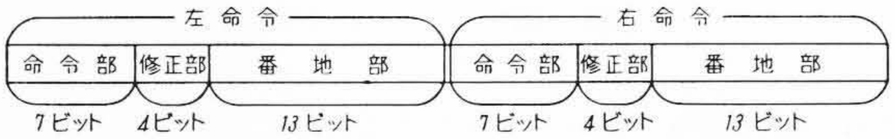
HIPAC103のために作成されたこの種のシステムとしてはHISIP 103A，HISIP 103B，HARP 103 がある。前二者はいわゆるアッセンブラとよばれるもので，一般に計算機の言語と1対1の対応がつく。これに対し後者はコンパイラと呼ばれる種類のもので言語としては非常に日常語に近いものを採用しており，機械語との1対1の対応はつかない。すなわちHISIPが機械向きな言語であるのに対し HARP は問題向き（あるいは人間向き）な言語であるというこ

とができよう。

HARP については**6**に譲ることにして本節では HISIP についての概説を与えることにする。

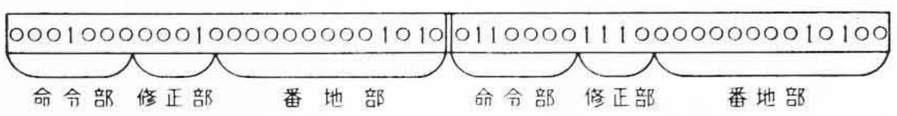
5.1 命　令　語

HIPAC 103 では一つの命令が2進法24けた（24ビット）より成り，1語に二つの命令がはいるペアードオーダ方式を採用している。すなわち命令レジスタには，次のような形で左命令と右命令の二つの命令がはいる。



修正部の4ビットは左より SCC (Sequencial Control Counter)，インデックス3，インデックス2，インデックス1を指定するビットでそのビットに1があれば命令を実行する際指定されたインデックスレジスタまたはSCCの内容を番地部に加えたものが実効番地となる。

たとえば



を命令語としてみると

左命令は：〔10＋(インデックス1の内容)〕番地にある数を Acc の内容に加えよ。

右命令は：Acc の内容を〔20＋(SCC の内容)＋(インデックス2の内容)＋(インデックス3の内容)〕番地に移せ。

という命令になる。

これを HISIP 語では

A/10, +1.　　T/20, +2+3+S.

と書けばよい。

一般に HISIP では命令を

$Op/X_1, X_2.$　または　$Op/X_1.$

の形に書けばよい。ここで Op は命令コードでアルファベット，数字の組み合わせで表わされる。字数は一定しておらず，1字ごとにある意味を持っていて記憶しやすいようになっているのが特長である（命令コード一覧表については紙数の関係で，ここでは省くが，“HIPAC 103 プログラミングマニュアル”を参照されたい）。いくつかの例をあげると

A：Add, Address, Accumulator など
B：Subtract
M：Multiply
D：Divide
I：Index
1, 2, 3：Index の区別，スイッチの区別
J：Jump
X：clear
S：Shift, Switch, Sign, Special, Search など
H：Character
O：Absolute

などであり，“1”命令コードと番地部，修正部を区別するもので必ず入れるものと約束する。X_1 は番地部を，X_2 は修正部を表わし，この間に“，”を，最後に一つの命令の終わりであることを示すため“．”をつける。修正部を要しないときは X_2 を省いた $Op/X_1.$ の形になる。

番地部 X_1，X_2 の表わし方は次のいずれかによる。

X_1：$\pm n$，記号番地，記号番地$\pm n$

X_2：＋1，＋2，＋3，＋S，－Sのいずれか，またはその組み合わせ

ここで n は0～8191の任意の整数値である。記号番地としてはアルファベット A～Ō のうちの1文字に数字0～9をつけたものだけが許される。

修正部の X_2 では＋1，＋2，＋3，＋Sはそれぞれインデックス1，2，3および SCC による修正を意味する。これらを重複して書けば重複して修正を行なうことになる。たとえば X_2＝＋1＋2 であれば X_1＋(インデックス1)＋(インデックス2) が有効番地として働き，X_2＝＋2＋3＋S であれば X_1＋(インデックス2)＋(インデックス3)＋(SCC)が有効番地として働く。

修正部の－Sは少し違った意味をもつ。たとえば

Op/X_1, －S.

と書いてあれば，読み込みの際

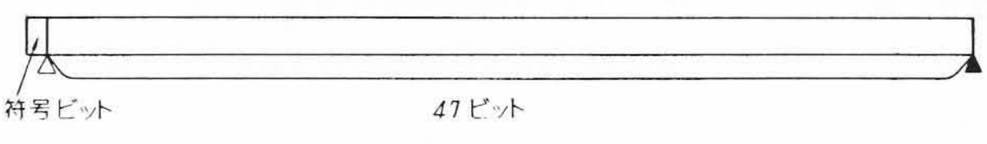
Op/X_1－(SCC), ＋S.

と直して格納する。したがって演算実行の際の有効番地は（実際に必要な番地と SCC の差)＋(SCC)＝(実際に必要な番地)となる。すなわち，この命令の番地部では絶対番地ではなく，この命令よりいくつあとの番地（あるいはいくつ前の番地）かを指定している。したがってすべての命令をこのような形に書いておけば，そのプログラム全体の格納番地をずらすこともでき便利である。

5.2 数　値　語

数値語には，固定小数点表示のものと浮動小数点表示のものとがある。

固定小数点表示の数値語は計算機の内部では2進法47けた(47ビット）および符号1けた（1ビット），計48けた (48ビット）で表わされる。



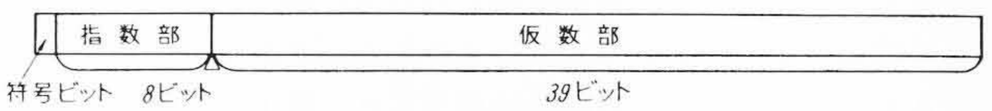
これは原則として△印のところに小数点を置いた小数とみなすが，ときには右端の▲印のところに小数点を置いた整数と考えることもある。固定小数点表示の数を HISIPを使って読み込ませるには

小数形：　±. n .（n のけた数は14けたまで)

整数形：　± n .　（n のけた数は14けたまで)

の形に書けばよい。

浮動小数点表示の数値語は計算機の内部では8ビットの指数部と39ビットの仮数部および符号ビット(1ビット)とで表わされる。



浮動小数点表示の数をHISIPを使って読み込ませるには次の形に書けばよい。

±. n ± p .（n は11けた以内，p は$-38 \leqq p \leqq +38$ の整数値)

これは小数±. n に $10^{\pm p}$ を乗じた数値を表わす。

5.3 制　御　指　令

HISIP ではプログラムの読み込みやデバッギングを容易にするため機械語に翻訳されない次のような制御指令が用意されている。

HISIP 103B.（または HISIP 103A.），ILA( )，記号番地)，記号番地( )，CANCEL( )，ENTER.，EXIT.，START( )，HALT.，WS( )，ŌCT( )，BCH( )，PUNCH( )，END.，CHECK.，CHECK 1.，STŌP CHECK.

上の制御指令の機能を簡単に述べると，

HISIP 103B. は記号番地の定義値をすべて解消する。

記号番地( )，記号番地)，は実際には A1(100)，A1) などと書いて記号番地に絶対番地を割り当てる。

ENTER., EXIT. は記号表をコアの裏番地へ移したりコアへ取り出したりする制御指令でこれを用いるとサブルーチンではメーンルーチンで使った記号番地を考えなくてもよい。

CANCEL( ) は ( ) の中の記号番地の定義値を解消する。

START ( ) は ( ) の中の番地より演算を開始させる制御指令である。

ILA( ) は ( ) の中の番地より順次プログラムを格納させる指令である。

HALT. は読み込みを停止する。

WS( ) は ( ) の中の個数だけ作業用番地を確保する。

ŌCT( ) は ( ) の中の 8 進数を読み込む。

BCH( ) は ( ) の中の文字または数字を 2 進数として読み込む。

PUNCH, END., CHECK., CHECK 1., STOP CHECK. はプログラミングにおいてデバッギングを容易にするために設けたもので PUNCH 指令はいったん機械語に編集された命令群をテープにさん孔させ，ふたたびこれを利用する際編集の手数を省くようにし，END. は記号番地の定義表と，未定義表の印字を，CHECK はプログラム進行中の中間結果の印字をする。

6. 自動プログラム (HARP 103)

5 で述べた記号入力ルーチンは，HIPAC 103 を能率良く働かすために作成されたシステムで，機械語の特色を生かしたプログラムを作成するためのものであり，プログラムの専門家向けのシステムということができる。しかし，一般の科学，工学の分野に出てくる問題を，その担当者が機械語やそれに準ずる言語を使って，プログラムを組むことはたいへんな負担となる。

HARP 103 は，われわれが普通使っている数式をほとんどそのままの形で書いてやれば，そのあとは HARP 103 が計算機を使って，それを解読し，機械語のプログラムを作り出す。すなわち，使用者が HARP 103 の言葉で書いたプログラムから機械語のプログラムを作り出すコーディングの仕事をプログラマの代わりに計算機に代行させるシステムプログラムである。

この段階のプログラミングの言語としては，IBM 社で開発した FORTRAN と世界共通語である ALGOL とがあるが，HARP 103 では FORTRAN 語を採用した。HARP 103 でプログラムを組むのは非常に簡単で，おそらく数時間の勉強で自由に計算機を使って仕事ができるようになると思う。

6.1 数式の取り扱い

四則演算とべき乗について

2 数 A, B の間の四則およびべき乗はそれぞれ次のように表わされる。

$X=A+B$　(加)

$X=A-B$　(減)

$X=A*B$　(乗)

$X=A/B$　(除)

$X=A\uparrow B$ または $A**B$　(べき)

(注)　乗算と除算については，×，÷の代わりにそれぞれ＊，／を用いる。またべき乗の表わし方は ALGOL では↑を，FORTRAN では＊＊を用いているので，両方許すことにした。

(注)　等号“＝”は等号の右辺の演算結果で左辺が置き換えられることを意味する。したがって

$I=I+1$

のような表現は許されるが，

$A+B=X$

というような表現は許されない。

第 1 表　Build in Function 一覧表

(1)	ABSF(X)	$\lvert X\rvert$ ただし X は実数形変数
(2)	XABSF(X)	$\lvert X\rvert$ の整数部分　ただし X は整数形変数
(3)	FLŌATF(X)	X を実数形表示に変換する
(4)	XFIXF(X)	X を整数形表示に変換する (X の整数部分をとる)
(5)	LŌGF(X)	$\log_{10} X$　ただし X は実数形変数
(6)	LŌGEF(X)	$\log_e X$　ただし X は整数形変数
(7)	EXPF(X)	10^X
(8)	EXPEF(X)	e^X
(9)	SINF(X)	$\sin X$
(10)	CŌSF(X)	$\cos X$
(11)	SQRTF(X)	$\sqrt{X}$

演算の順序は

(1)　関数

(2)　べき乗

(3)　乗算，除算

(4)　加算，減算

で演算は原則として左から右へ行なわれる。また括弧は通常約束されているように用いられる。たとえば

$$Z=A\uparrow B*C+D\uparrow E/(F-G)-H$$

は

$$A^B\times C+\frac{D^E}{(F-G)}-H$$

を計算して，その結果を Z とすることを意味する。

以上で数式の書き方はだいたい見当がついたと思われるので，例について説明する。

二次方程式

$$x^2+ax+b=0$$

の根はよく知られているように

$$x=\frac{-a\pm\sqrt{a^2-4b}}{2}$$

で求められる。この 2 根をそれぞれ $X1$, $X2$ と名付ければ，

$X1=(-A+(A\uparrow 2-4.0*B)\uparrow 0.5)/2.0$

$X2=(-A-(A\uparrow 2-4.0*B)\uparrow 0.5)/2.0$

と書けば 2 根は求まる。ここに記したように $\sqrt{X}$ の計算は $X\uparrow 0.5$ とすればよいのであるが，一般に，平方根を求める場合は非常に多くそのような計算はできるだけ短い時間で計算できるようにすることが望ましい。そのような目的のために，初等関数（三角関数や指数関数など）はその HARP 語で約束された記号を書けば関数を自動的に計算してくれるように用意してある。**第 1 表**にこれらの作りつけ関数 (Build in Function) を示す。

HARP 103 で取り扱う数値には，一般の数を扱う実数形表示のものと整数のみを扱う整数形表示のものと二とおりある。このうち前者は計算機の中では指数部と仮数部に分けて表わす浮動小数点表示の数に対応し，後者は固定小数点表示の数に対応する。この両者を区別するために，実数形表示の定数は必ず小数点をつけて表わし，他方整数定数は小数点をつけないで表わす。また実数形変数と整数形変数の区別をするため，後者の変数名は先頭の文字が I，J，K，L, M, N のうちのどれかであるとし，それ以外は実数形変数とみなす。また，原則として一つの式の中に実数形表示の数と整数形表示の数が混じっていてはならない。ただし実数形の数の整数乗と，添字（整数形）付実数形変数は許される。したがって 2 次方程式の根の計算式は A, B, $X1$, $X2$ などは皆，実数形変数であるから $A\uparrow 2$ を除き他の定数は皆 4.0 とか 2.0 のように実数形定数の形に書かなければならない。

6.2 制御文の取り扱い

とび越し指令

さきに二次方程式の根を求める例を示したが，a^2-4b が負かど

うか判定して，負ならば Stop し，負でなければ根の計算をしたのち Stop するというようなことも必要となる。このような条件によるとび越しのためのステートメントは種々あるが，ここでは IF ステートメントと GŌ TŌ ステートメントの説明をしておく。

IF　(A)　n_1,　n_2,　n_3

ここでAは数式で上の意味は

もし　Aが負ならばステートメントナンバ n_1 のステートメントへ

　　Aが零ならばステートメントナンバ n_2 のステートメントへ

　　Aが正ならばステートメントナンバ n_3 のステートメントへ

とべという指令である。

GŌ　TŌ　n

この意味は，無条件にステートメントナンバnのステートメントへとべという指令である。

そこで上の二次方程式の根を求める問題では

ステートメント No.	HARP 103 ステートメント
	IF (A ↑ 2 −4.0 * B) 10, 30, 20
20	X1=(−A+SQRTF (A ↑ 2 −4.0 * B))/2.0
	X2=(−A−SQRTF(A ↑ 2 −4.0 * B))/2.0
10	STŌP
30	X1=−0.5 * A
	X2=X1
	GŌ TŌ 10

のようになる。

（注）　上の問題で $A \uparrow 2 - 4.0 * B$ が零になる場合はほとんど起こらないと仮定すれば IF(A ↑ 2 −4.0 * B) 10, 20, 20としたほうがよい。そのときはステートメントナンバ30以降のステートメントは省略できる。

6.3 反　復　指　令

電子計算機が偉力を発揮する例として計算を反復する場合があげられる。いま二つのベクトル $(A_1, A_2, \cdots\cdots, A_n)$, $(B_1, B_2, \cdots\cdots, B_n)$ の内積 $\sum_{i=1}^{n} A_i B_i$ を計算する場合を考えよう。 $n=5$ とすれば

$$P = A(1) * B(1) + A(2) * B(2) + A(3) * B(3) + A(4) * B(4) + A(5) * B(5)$$

と書けばよい。ここで$A(i)$はAiを意味する。なお添字の一般的説明を加えておく。上のように添字は変数の次に括弧の中に入れて表わされる。添字の形としては

$$\alpha * I \pm \beta$$

の形のものが許される。ここで α, β は正の整数形定数，Iは正の整数形変数である。また添字全体として，すなわち $\alpha * I \pm \beta$ が正でなければならない。

添字の数については二重まで許される。すなわち

$$X(\alpha * I \pm \beta,\quad \gamma * J \pm \delta)$$

問題をもとにもどして，内積Pを求めるような問題では次の反復指令を用いればよい。

DŌ　n　$i=m_1$,　m_2,　m_3

ここでnはステートメントナンバ，iは添字のつかない整数形変数で m_1, m_2, m_3 は符号のつかない整数形定数か，添字のつかない整数形変数である。また m_3 が書いてないときは1とみなされる。

このステートメントの意味は，この次のステートメントからステートメントナンバnまでのステートメントを，まず$i=m_1$として実行し，次に$i=m_1+m_3$として実行してiを m_3 ずつ増して m_2 をこえる直前まで繰り返し実行したのち，nの次のステートメントに移る。

DŌ ステートメントを用いると

$$P = \sum_{i=1}^{5} Ai\, Bi$$

は

```
      .....
      P=0.0
      DŌ 10 I=1, 5
10    P=P+A(I)*B(I)
      .....
```

とすればよい。

6.4 入出力ステートメントおよびそのほかのステートメント

実際のプログラミングで，面倒なものにデータの入出力がある。特に出力はデータをきれいに並べて印字させることとか，データの項目名を印字させることとかにいろいろ手間がかかる。HARP 103では，FORMAT ステートメントを設け，それらのことが簡単にできるようになっている。

データをたくわえる番地を確保するステートメントとしては，DIMENSION と ARRAY ステートメントとがある。前者はコアメモリに，後者はドラムメモリにたくわえるときに使われる。そのほかのステートメントについては説明を省略する。

6.5 プログラムのデバッギング

HARPを使ってプログラムを組むことのいちばんの利点はだれでも簡単にプログラムが組めることである。また同時に機械語のプログラミングに比べてコーディングの誤りが非常に少ない点である。しかし，少ないといっても誤りを犯すことがあるであろうからHARPシステムでは，コーディングによる誤りの検出と数式自身の論理的な誤りを簡単に検出できるようなくふうがされている。

また，数式自身に論理的な誤りがあるかどうか計算結果のチェックが要求される。プログラムを実行したときに，その結果がまちがっていると，どの計算式がまちがっているかを調べなければならない。それを助けるために，中間結果を印字させたい場合がしばしばある。そのため，TYPE ステートメントを各所にそう入しておいてそれで中間結果を印字させて誤りを調べ，誤りを正したのちは，このステートメントを無視するようにしたい。このようなことを可能にするために HARP 103 では制御卓のスイッチの操作によって，同じテープで異なった処理ができるようになっている。

7. HIPAC 103 ライブラリについて

計算を短時間にかつ合理的に行なうために HIPAC 103 のソフトウエアとしてライブラリルーチンを作製中であるが，37年4月10日現在約70種のルーチンがすでに登録され使用可能の状態となっている。ライブラリの題目は次のとおりである。

（1）　四則演算ルーチン

　（a）　浮動小数点表示による複素数の四則演算

　（b）　固定小数点表示による倍精度演算ルーチン（四則，入出力）

　（c）　固定小数点表示の倍精度数値読込ルーチン

　（d）　浮動小数点表示から固定小数点表示への変換サブルーチン

（2）　プログラムチェック・ルーチン

　（a）　検屍ルーチン

　（b）　ブランチトレーサ

　（c）　整数形，小数形固定小数点表示数値プリント

（d） 浮動小数点表示数値プリント
（e） HISIP 103 命令プリント
（f） ダイナミックチェックプログラム (Dynamic Check Program)
（g） ストアアンドジャンプトレーサ (Store and Jump Tracer)

（3） 微分方程式用ルーチン
（a） k 元連立 Runge-Kutta-Gill 法
（b） 数値積分 (Gau β)

（4） 関数ルーチン

（a） $\sin x$	（b） $\cos x$	（c） $\tan x$
（d） $\sin^{-1}x$	（e） $\cos^{-1}x$	（f） $\tan^{-1}x$
（g） $\sinh x$	（h） $\cosh x$(1), (2)	（i） $\tanh x$
（j） $\log_{10} x$	（k） $\log_e x$	（l） $\log_2 x$
（m） $\sqrt{x}$	（n） $\sqrt[3]{x}$	（o） $\sqrt[n]{x}$
（p） 10^x	（q） e^x(1), (2)	（r） 2^x
（s） a^x	（t） $J_0(x)$	（u） $J_1(x)$

（5） 入力ルーチン
（a） 10進入力ルーチン (K-code)（1），（2）
（b） 初期入力ルーチン
（c） HISIP 103A
（d） HISIP 103B

（6） 内そうおよびカーブフィッティング用ルーチン
（a） 最小二乗法による関数近似
（b） ラグランジアン内そう

（7） マトリクス計算ルーチン
（a） 連立一次方程式，逆行列
（b） 行列の固有値，固有ベクトル
（c） 行列の加減乗算，転置
（d） リニアプログラミング

（8） 出力ルーチン
（a） リードタイプルーチン
（b） 8進印字サブルーチン
（c） 印字サブルーチン
（d） ラインプリンタ用印字サブルーチン
（e） ラインプリンタ用編集サブルーチン

（9） 記憶装置制御ルーチン
（a） 磁気テープ書込みサブルーチン
（b） 磁気テープ読出しサブルーチン
（c） HISIP 103A の移動
（d） メモリ読み出し

(10) 代数方程式用ルーチン
（a） 高次代数方程式の根
（b） 二次方程式の根を求めるサブルーチン
（c） 多項式の計算

(11) 統計用ルーチン
（a） 多次元分布の共分散
（b） モーメント
（c） 回帰分析
（d） 実験計画，(二元配置)，(三元配置)

上記ライブラリは HISIP 語で書かれているが，HARP 103 で書かれたライブラリも現在作製中である。

8. 結　　言

HIPAC 103 の製品化にあたって直面した設計上の諸問題およびディジタル計算機に固有のソフトウエア開発の現状について述べた。前者は日立製作所戸塚工場第3設計課において，後者は日立製作所中央研究所83研究室において遂行されたものである。

本文を草するにあたり，ソフトウエア開発に種々ご指導賜わった東京大学森口教授はじめMH委員会の委員各位および設計製作にあたってご激励を賜わった日立製作所戸塚工場高田副工場長，中谷部長，安藤課長に衷心より感謝の意を表するものである。

参　考　文　献

（1） 日立製作所： HIPAC 103 プログラミングマニュアル

特　許　と　新　案

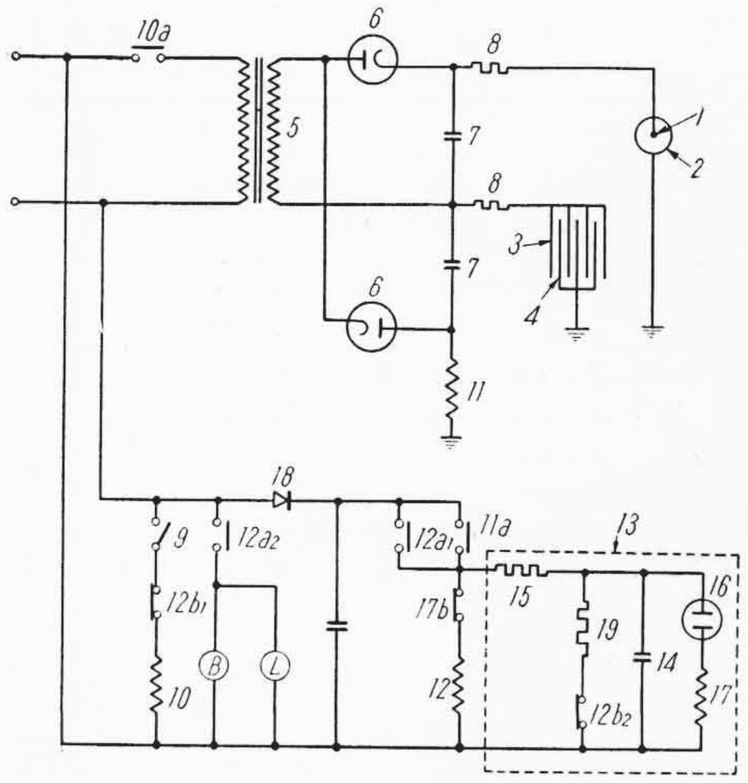
特 許 第 284697 号

海和弥太郎・安藤文蔵

電 気 集 じ ん 器 の 保 護 装 置

室内空気清浄用の電気集じん器では，せんい類のような大きなごみにより電極間が短絡されることがしばしばある。この場合普通は電流リレーが動作して電源を切るようにしているが，電源の再投入を手動で行なわねばならないので保守が面倒である。またリーケージトランスなどを用いて短絡電流を制限するようにしたものでは電極間にはさまったごみが焼き切れずに残り，電圧降下により集じん不能におちいる場合がある。この発明は電極がごみにより短絡された場合，間けつ的に大電流を流してごみを焼き切り，短絡が除かれると自動的に正常運転にもどるようにしたものである。図示の回路において正常時には起動スイッチ9の投入により電磁開閉器10の接点10aが閉じ高圧トランス5，整流管6，コンデンサ7，保護抵抗8を経て電離電極1,2と集じん電極3,4に高電圧が印加されるが，電極が短絡されると電流リレー11の接点11aが閉じるため，リレー12が動作して接点 $12b_1$ を開き電磁開閉器10を消勢して接点10aを開放させる。一方接点 $12b_2$ の開放によりタイマー13のコンデンサ14が充電され，一定時間後二極管16が放電，リレー17が動作して接点17bを開き，これにより接点10aが再投入される。このとき短絡状態が続いておれば再びリレー11，12が動作して接点10aを開放する。そして一定時間後タイマー13によってまた接点10aが投入される。このような動作をくり返して回路に間けつ的に大電流を流し，ごみが焼き切れてしまうとリレーの動作がやみ自動的に正常運転にもどるから，保守が容易であり，集じん不能の状態で放置されることもない。またタイマーの時限を適当に定めればたとえ永続的な短絡事故が生じた場合でも電源機器を保護することができる。この場合短絡事故の発生はブザーB，ランプLによって報知される。

（坂　本）



第 1 図