

# 汎用大形電子計算機 HITAC 5020 のソフトウェアシステム

Software System of Large-Capacity Electronic Computer HITAC 5020

酒井博敬\*      原田昌孝\*\*      高橋延匡\*\*\*  
Hiroataka Sakai      Masataka Harada      Nobumasa Takahashi  
中田育男\*\*\*      三橋莊一郎\*\*\*  
Ikuo Nakata      Sōichirō Mitsuhashi

## 内 容 梗 概

汎用大形電子計算機 HITAC 5020 のソフトウェアはモニタ、HARP 5020、HISAP 5020 を主体とするモニタシステムである。本稿ではモニタシステムによる仕事の処理方法について解説し、おもなシステムプログラムの機能についても概要を述べた。

## 1. 緒 言

電子計算機のソフトウェアシステムの中心は FORTRAN, ALGOL, COBOL, オートコードなどの問題向き言語または記号言語の翻訳, 変換ルーチン, 分類あるいは報告作成ルーチン, シミュレーションシステム, その他もろもろのライブラリルーチンなどであるが, 計算機が大形高速化し, またデータ処理が複雑かつ大規模になり, プログラムあるいはプログラマ間の相互通信が必要になってくると, 人間と機械とプログラムの間の連絡をとり, 全体として計算機の稼働率を高めるための制御プログラム, すなわちモニタが必要となる。現在ではほとんどすべてのソフトウェアシステムがモニタを頂点として, あらゆるシステムプログラムをその制御下においたモニタシステム(またはオペレーティングシステム)の形をとっている。一口にモニタといっても, その制御方式は計算機の規模や構成, 仕事の内容, 作成者の考え方などによって異なるもので, モニタの機能の一般性, 融通性, 実用性という問題は他のシステムプログラムに比べるとかなり批判の多いところである。

HITAC 5020 のソフトウェアシステムはモニタおよび HARP 5020 (FORTRAN), HISAP 5020 (FCP 付きオートコード), 分類および報告作成ルーチンなどのプログラミング言語を主体として構成されるモニタシステムである。本稿ではモニタによる制御方式やおもなシステムプログラムについての概略を述べる。

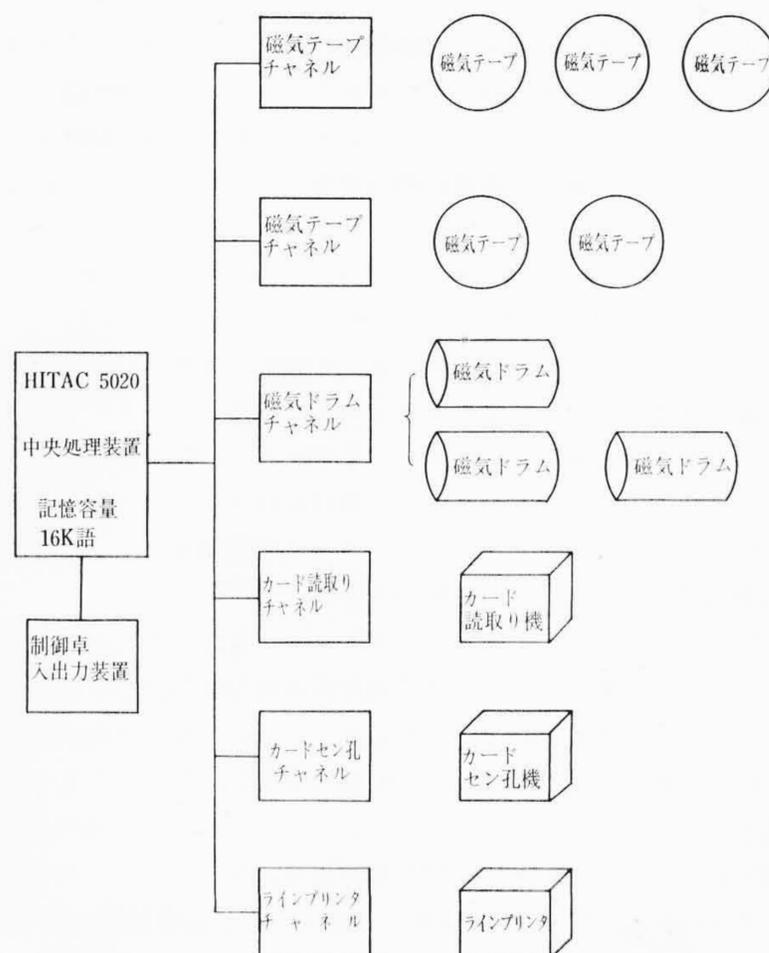
## 2. モ ニ タ

### 2.1 仕事の処理方法について

まずシステム全体に関する制御として, 計算機の連続運転や, 内部処理と周辺入出力の同時処理の制御がモニタの主要目的となる。ここで内部処理とは原始プログラムの翻訳, 変換, プログラムセグメントの連結, 目的プログラムの読込みと実行といった一連の作業のことで, 入出力媒体は磁気テープである。また周辺入出力とはカード→テープ, テープ→プリンタ/カードのような単純な作業である。周辺入出力(I/O)は内部処理(E)の作業と並行して時分的に行なわれるが, 小形計算機をもつ機器構成においては I/O をそっくり小形計算機に移行させることができる。われわれのシステムではむしろ前者の I-E-O の多重処理モードを標準的な処理方式としている。しかしこの場合も仕事の内容によっては多くの入出力装置や記憶装置を必要とすることがあり, このときは単独処理モードに切り換え

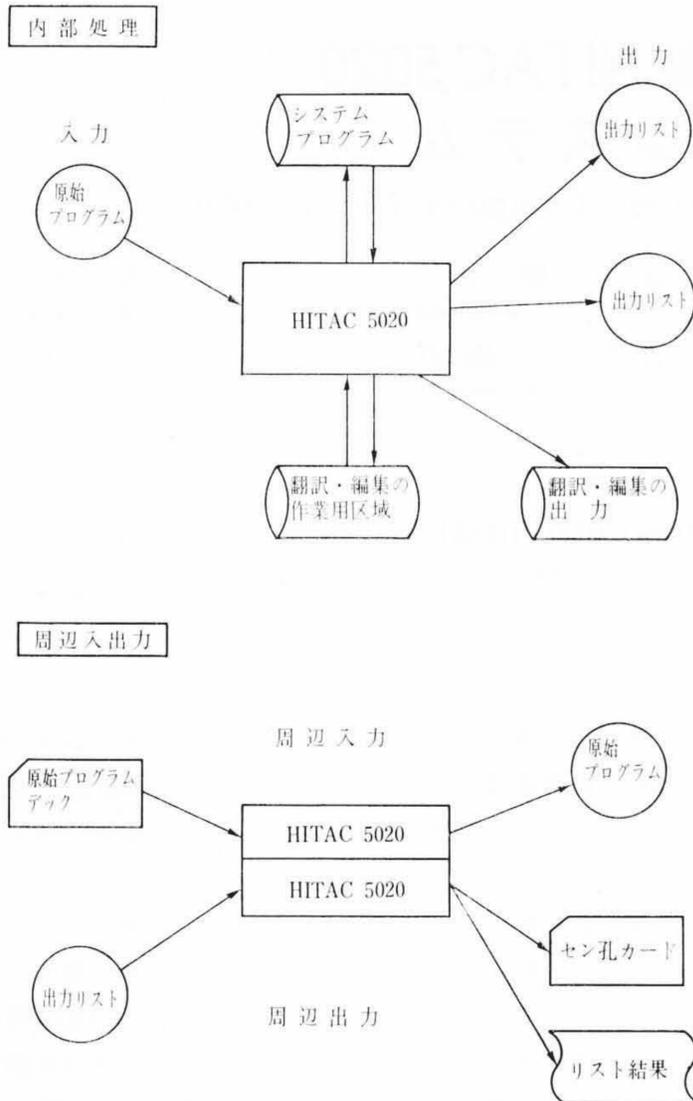
て E の処理だけに専念することもできるようにした。以上の処理方式のための標準機器構成として, 磁心記憶装置 16K 語, 磁気テープ 2 チャンネル 5 台, 大容量磁気ドラム 1 チャンネル 3 台, カード入出力装置各 1 チャンネル 1 台, ラインプリンタ 1 チャンネル 1 台, 制御卓入出力装置 1 台を使用する。(第 1 図参照)ここで磁気テープは入力ファイルとして 2 台, 出力ファイルとして 3 台を割り当て, 磁気ドラムはシステムプログラムのファイルとして 1 台, 内部処理の際のシステムおよび目的プログラムの作業用区域として 2 台を割り当てる。制御卓入出力装置は操作員との通信に用いられる。特に磁気ドラムをシステムファイルとして使用することによって, モニタルーチンをはじめ, すべてのシステムプログラムを必要なときだけ磁心記憶装置に呼び出すことが容易になり, 記憶装置がシステムプログラムによって必要以上に占有されることが避けられる。入出力ファイルとしての 2 台および 3 台の磁気テープは I-E-O が同時に実行できるように互いに代替テープとして働く(第 2 図参照)。

次に内部処理においては, 入力ファイルにある仕事の列を順次一



第 1 図 HITAC 5020 の標準機器構成

\* 日立製作所神奈川工場  
\*\* 京浜日立エンジニアリング株式会社  
\*\*\* 日立製作所中央研究所



第2図 標準構成のときの周辺機器の割当

つずつ取り出して、原始プログラムの翻訳、変換から目的プログラムの実行までの処理を連続的に行なうわけであるが、一つの仕事を形作るプログラムはセグメント、デッキ、仕事の3段階の単位に分割され、システムによる処理過程で互いに他と独立に扱われる。ここにセグメントとは、あるまとまった命令群やデータ区域の最小単位であって、外部参照名をもつ。たとえばサブプログラム、ライブラリルーチン、チェインジョブの際の共通区域などはセグメントである。プログラムに対する記憶装置の割付けはセグメントを単位として行なわれる。次に同一のプログラミング言語で書かれたセグメントの集まりがデッキである。デッキはHARPやHISAPなどによるプログラムの翻訳、変換の際の単位となるもので、これには次のような種類がある。

- (1) HARP言語で書かれたセグメントからなるデッキ。
- (2) HISAP言語で書かれたセグメントからなるデッキ。
- (3) HARPまたはHISAPによって翻訳、変換された、再配置可能な2進形の間言語のセグメントからなるデッキ。

このデッキが集まって仕事というまとまった作業単位を構成する。一つの仕事の中でも、デッキが異なればプログラム言語も異なっていてよいわけであるが、異なるデッキの間でもセグメントの外部参照名は共通の名前として使用できる。

(3)にあげたように、原始プログラムをいきなり実行可能な目的プログラムに変換せず、ひとまず再配置可能な中間言語に落とすことはシステム処理の常用手段であるが、これには次の理由がある。

- (a) 原始プログラムは時と人を選ばず独立に作られたものが混在していてよいわけであるから、いったんすべてを共通の中間言語にしてから、セグメント間の相互参照、ライブラリセグメントやFCPルーチンの組み込み、入出力および記憶装置の割付け、チェインジョブのための編集に取りかからなければならない。
- (b) 使用ひん度の高いルーチン化したプログラムは中間言語の

形で磁気テープやカードに保存しておく。これは外部参照名をもった再配置可能な2進形のセグメントからなるデッキであるから、情報量としても少なく、目的プログラムへの変換は他の言語よりはるかに速い。

## 2.2 システムモニタ

モニタは機能上、周辺入出力操作、入出力装置の割当て、モニタールーチンの保護のようなシステム全体に関するシステムモニタと、内部処理における各仕事の連続処理を制御するジョブモニタの二つに分けられるが、ここで特にシステムモニタについて述べておく。

システムモニタは次のものから構成される。

(1) HICORE: 記憶装置の割付け状態表、システムファイルの分布表、入出力装置の機能表(UFT)と制御表(UCB)などからなる各システムプログラムとの情報交換の場で、磁心記憶装置内に常駐する。

(2) HISUP: 各システムプログラムの交通整理をする制御ルーチンで、必要なシステムルーチンを磁心記憶装置に読み込むほか、入出力装置の割り当てと保守、ジョブモニタへの引き継ぎなどを行なう。

(3) MCP: 多重プログラミングのためのチャンネルスケジューラ、各種割込み要因の監視ルーチンなどからなる主制御プログラムで、磁心記憶装置内に常駐する。(2.3参照)

(4) 制御命令の入力とメッセージの出力ルーチン: 操作員との通信を制御卓入出力装置を介して行なうためのルーチンで、磁心記憶装置内に常駐する。

(5) 誤りの制御ルーチン: 中央装置や入出力装置で誤りが発生したとき、プログラムに誤りがあったときなどの処理ルーチンである。

(6) 編集ルーチン: ソフトウェアシステムの保守を行なうためのルーチンで、システムモニタの機能やライブラリルーチンの追加をするために使用される。

(7) DUMPルーチン: 磁心記憶装置と磁気ドラムの内容を出力するためのルーチンである。

このなかで特にHISUPは各仕事の間でシステムモニタに制御が移ったときに磁心記憶装置に読み込まれ、必要な処理が終わるとふたたびシステムファイルにもどされる。またUFTはシステム入出力ファイルとUCBの対応を、UCBは実際の入出力装置の状態に関する情報を記録する表であって、HISUPによってのみ記入、変更、管理される。そして入出力命令における入出力装置の指定はすべてUFTとUCBを参照して間接的に行なわれる。

## 2.3 MCP (Master Control Program)

I-E-Oの3種のプログラムは、入出力動作の際生ずる中央装置の待ち時間を利用して、時分割的に実行されるわけであるが、MCPはこの三つのプログラムの間の制御の動きを監視し、制御があるプログラムから他のプログラムへ移るときに、レジスタ類や表示子類の退避、回復を行なう。また実行中のプログラムが他のプログラムやMCP自身を破壊することを防ぐための記憶装置保護、各プログラムの実行時間の計算、磁気テープの巻戻し時間の計算と巻戻し終了時の処理、入出力のチャンネルスケジューリング、またスイッチ、記憶装置誤り、あふれ、無操作(no operation)などによる割込みの処理などを行なう。チャンネルスケジューラが必要とするUFTやUCBの情報はHISUPによって作り出されたものである。一方MCPは各プログラムの状態を監視していて、たとえばプログラムの予想終了時間がフルカウントになったり、他のプログラムを破壊しようとしたとき、磁気テープの巻戻しが終了したりした場合、そのことを発見してHISUPに知らせ、メッセージの印字を行なわせる。

次にMCPはFCPとの関係が密接である。FCPの主目的は入出

力の際、バッファリングによって入出力によるプログラムの待ち時間をできるだけ少なくすることにある。したがって FCP では二つまたは三つの入出力区域を用意し、一つの入出力区域に対する入出力を実行中に他の入出力区域の処理を行なうという方法がとられるのが普通である。FCP 中のファイルスケジューラと呼ばれるルーチンは入出力区域の状態を常に監視しており、ある入出力区域の処理が終わるとただちに次の入出力区域の処理に移れるようにし、かつそのあいた入出力区域に対する入出力命令を実行させるために、MCP のチャンネルスケジューラに制御を渡す。この場合次の入出力区域に対する入出力が未終了であればプログラムは処理を続行することができない。このようなときは MCP は現在実行中であったプログラムを待たせて別のプログラムへ制御を渡す。一方入出力命令を実行しようとしたとき、必要なチャンネルが他のプログラムや他のファイルのために使用中であった場合は入出力命令を待たせなければならない。そこでチャンネルスケジューラはファイルスケジューラから要求された入出力命令を受け付け、必要ならばチャンネルの待ちテーブルに登録し、プログラムおよびファイルの優先順位から入出力命令の実行順位を決定する。また入出力動作が終了した場合はその入出力の誤りの有無を調べ、もし誤りがあれば何回か入出力命令を実行しなおし、さらに誤りの種類と回数の記録をとり、後の資料とする。入出力が正常に実行された場合は次の入出力命令を発するとともに、入出力区域の準備が完了したことを表示し、その区域を処理するプログラムが区域の準備完了を待って休止していた場合は、そのプログラムを再開させる手続きをとる。このように MCP と FCP は入出力チャンネルと入出力区域を介して密接に係るものである。

### 3. HISAP 5020

HISAP 5020 はファイル制御システム FCP (File Control Processor) を備えたオートコーダで、そのおもな処理能力として次の点があげられる。

- (1) 記号言語は機械語に直接対応するプログラムを容易に組めるものでなければならない。そのためには複雑な記号演算やデータの表現を許すだけでなく、一度定義した機械語のパターンを簡単に使用できるプログラマのマクロの処理機能をもつ。
- (2) 異なるセグメントとの連結を行なうために、セグメントへの分割、セグメントの間の相互参照を容易にするための疑似命令をもつ。
- (3) 事務計算におけるデータ処理用言語として FCP を備えたシステムとする。多くの場合データは磁気テープファイルとして入出力されるが、磁気テープファイルの形式は単に目的プログラムのファイルだけではなく、周辺入出力としてシステムで使用するファイルも含めて、小形計算機との互換性を考慮してすべてを統一してある。
- (4) FCP のうち、ファイルスケジューラは単に HISAP のためにだけ用意するのではなく、周辺入出力でも使用できるようにシステムモニタと直接連結できるようにしてある。
- (5) プログラム手直しのために十分な疑似命令をもつ。

上のような機能はすべて疑似命令という形で実現されるわけであるが、これは一般的な疑似命令、FCP に関する疑似命令、プログラムの手直しに関する疑似命令の 3 種類に分けられる。特に FCP はファイルの入出力に関するバッファリング、誤りの処理などを統一して扱う制御システムであってファイルスケジューラが主体となり、2.3 で述べたように MCP と連絡して働く。HISAP におけるファイルの処理は次の 3 種類の疑似命令を組み合わせて使うことにより行なわれる。

(a) ファイル記述とデータ記述： ファイル記述はファイル名、入出力装置との対応 (UFT との対応)、ブロッキングの形式、入力か出力かの別、磁心記憶装置内の入出力区域との対応、バッファリングの方法、前書きおよび後書きラベルの情報、誤りの処理の指定などファイルに関する情報の記述である。ファイル記述からはファイル制御表 FCT が作り出される。またデータ記述というのはファイルの構成要素であり、また内部処理の際の単位であるデータ記録について、データの形式や長さを指定するとともに、記憶装置内に入出力区域または作業用区域をとるための記述である。

(b) 入出力マクロ： これはファイルの読み書きを行なうマクロであって、パラメータとしてはファイル名だけ (ファイル→入出力区域のデータの移動)、またはファイル名と作業用区域名 (ファイル→入出力区域→作業用区域のデータの移動) を指定する。このマクロはファイルスケジューラの呼び出し系列に変換され、実際の入出力制御はファイルスケジューラで行なわれる (2.3 参照)。

(c) 演算マクロ： このマクロは入出力区域内または作業用区域内のデータ記録に対して、算術演算、論理演算、移動、編集などを行なうものであって、データ記述の指定にしたがって必要な命令群に変換される。

このようにファイルとデータに関する記述は入出力マクロや演算マクロから分離して与えられるため、入出力マクロや演算マクロではパラメータの書き方が非常に簡単になり、ファイルやデータを自由に処理できるようになる。

このほか FCP はファイルの処理中に発生する災害に備えて、磁気テープリールの処理が終わるたびに、その時点での記憶装置のすべての内容や、各ファイルのブロックカウントなどの記録を磁気テープにとっておく「検査点記録」に関する疑似命令と、記録をとった時点の状態から仕事を再開する「再運転」の機能をもっている。ただし再運転はモニタの制御によって行なわれる。

### 4. HARP 5020

HARP 5020 言語としては IBM 社の FORTRAN IV 言語をそのまま採用しているが、さらに HITAC 5020 の機能を生かすために 2 重精度演算、複素数演算 (2 重精度演算を含む)、論理演算、文字の取り扱い、磁気ドラムの入出力などのステートメントのほかに、プログラムの手直しを容易にするためにいくつかのステートメントを追加してある。

HITAC 5020 の 2 番地から 15 番地までは A レジスタと呼ばれ、累算器の機能をもっている。また 2 番地から 7 番地までは B レジスタとも呼ばれ、指標レジスタの機能をもっている。これら多くのレジスタを利用して、目的プログラムを短く能率のよいものにすることができる。HARP 5020 では 8 番地から 15 番地までの 8 語の A レジスタを、普通の整数演算では 4 個の累算器として使い、2 重精度の整数演算では 2 個の累算器として使い、普通の実数および 2 重精度の実数演算では 4 個の累算器として使い、さらに普通の複素数および 2 重精度の複素数演算では 1 組の複素数累算器として使うことにより演算速度を速くしている。また DO ループの制御を能率よくすることが重要であるが、特に DO ループの中で配列要素 (array element) を効果的に処理するために、6 個の B レジスタ、MNI 命令 (modify next instruction, 次の命令のアドレス変更を行なうための命令)、間接アドレス機能などを用いている。

HARP 言語でプログラムを書けば、もっと機械語に近い言語で書く場合よりプログラムの手直しははるかにらくになるが、しかしそれでも現在多くの時間がプログラムの手直しのために使われている。そこで手直しが能率よくできるようにするために、HARP 5020

には次のようなプログラムの手直し用ステートメントを設けた。

(a) DEBUG VALUES  $n_1, n_2$  IF( $e$ )  $a_1, a_2, \dots$

(b) DEBUG FLOW  $n_1, n_2$

(c) DEBUG DUMP  $n$  IF ( $e$ )  $a_1, a_2, \dots$

(d) DEBUG TERMINATE  $n$  IF ( $e$ )

(a)はステートメント  $n_1$  と  $n_2$  の間で、 $a_1, a_2, \dots$ を左辺にもつ算術ステートメントがあればそのステートメントを実行したときに論理式  $e$  の値が真 (true) であるときだけ  $a_1, a_2, \dots$  の値を印字する。(b)はステートメント  $n_1$  と  $n_2$  の間に GO TO ステートメントや IF ステートメントがあった場合、それらのステートメントを実行するとき、どこへ制御が移るかを印字する。(c)はステートメントを実行したとき  $e$  の値が真であれば  $a_1, a_2, \dots$ の値を印字する。(d)はステートメント  $n$  を実行したとき  $e$  の値が真であればプログラムの実行を終了させる。

### 5. 結 言

HITAC 5020 のソフトウェアシステムはその文法構造の完成だけで1年半を要した。これはわれわれにとって本格的な大形計算機の

ソフトウェアの開発がはじめてであることにもよるが、モニタシステムそのものが本来、モニタによる制御方式、多重プログラミング、入出力制御システム、個々のシステムプログラムとの連絡など、詳細な検討を要する数多くの問題をかかえているからであろう。これまでに述べたものはわれわれが第1次開発計画として作成しているソフトウェアのおもなものであるが、このシステムはこれで閉じているのではなく、将来の第2次以降の開発を予定して拡張可能な形になっている。

終わりに、このシステムの開発に当たってご指導をいただいた、中央研究所第82研究室、第83研究室、日立工場事務管理部および神奈川工場方式課の関係者のかたがたに心から感謝する次第である。

### 参 考 文 献

- (1) HITAC 5020 命令語説明書
- (2) HITAC 5020 ソフトウェアシステム概説書
- (3) HITAC 5020 モニタ説明書
- (4) HARP 5020 説明書
- (5) HISAP 5020 説明書



## 新 案 の 紹 介



登録新案第728545号

角 七 男

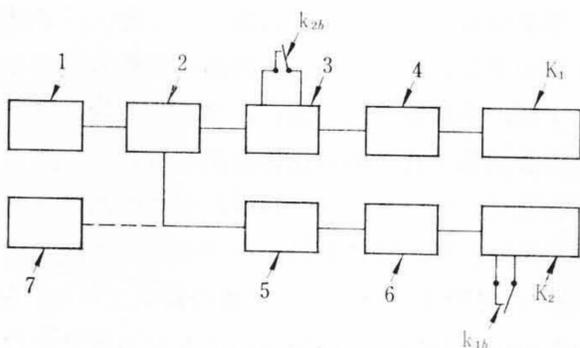
### 列 車 近 接 自 動 警 報 装 置

無人踏切等において線路を伝播する列車の進行音波を利用して列車が近接した時に警報を発し、列車が通過した直後に警報を自動的に停止するもので、検出器1は列車の進行音が線路を伝播して来る音波イを電気的信号に変換する。この信号を増幅器2により増幅し、波振幅制限回路3にて適当な周波数だけを通す波器および継電器動作に必要な振幅制限回路を経て、検波器4で継電器制御の直流出力を得る。この直流出力により継電器  $K_1$  が波形口の点8で動作し、点10で動作を停止するように増幅器2の利得の動作点を選んで置き、次に増幅器2の出力の一部を波ゲート増幅器5に加え、列車の進行通過時だけ波形ハに示す電気信号が点9で現われるようにし、この信号を検波器6に加えて検波し、有極継電器  $K_2$  を動作させるため適当な制御電圧を供給する。次に継電器動作は列車の近接によって継電器  $K_1$  の接点  $K_{1a}$  が閉じると継電器  $K_3$  が動作し、接点  $K_{3a}, K_{3b}$  が閉じ警報器13, 14が動作する。この場合に第1図の有極継電器  $K_2$  は波ゲート増幅器5に信号が来ない時は、常に接点  $K_{2a}$  が閉じる方向に回路が選んである。次に列車がこの装置の近くに来て第2図口の波形の点9になったとき有極継電器  $K_2$

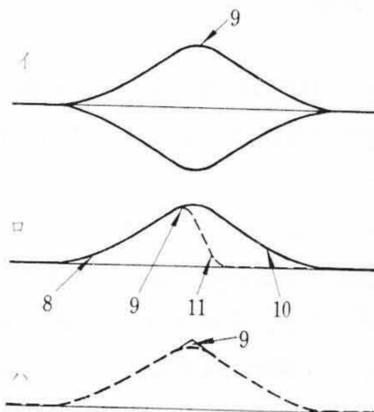
が動作し、その接点  $K_{2a}$  が開き同時に接点  $K_{2b}$  が閉じて継電器  $K_3$  が不動作となり、その接点  $K_{3a}, K_{3b}$  が開いて警報器13, 14の回路が切断され警報が停止する。しかし第2図の波形図口に示す通り接点  $K_{1a}$  は未だ閉じており、列車が通過しさらに点10の距離まで離れると接点  $K_{1a}$  が開き同時に第1図の有極継電器  $K_2$  において継電器  $K_1$  の接点  $K_{1b}$  で有極継電器  $K_2$  を反転させ接点  $K_{2a}$  を閉じる。ここで自動警報の一操作が完了し、初めの待受状態となるが、第2図に示すように点9で警報停止後、復元するまでの時間点9から10までは、列車速度を一定とすれば、警報動作点8から9までの時間とほぼ一致する。列車運転ダイヤの輻輳する場所では警報動作の時間を短かくして直ちに待受状態に戻さねば、警報不動作となる場合があり、これを防ぐには有極継電器  $K_2$  の接点  $K_{2a}$  の反対側接点  $K_{2b}$  を利用し、波振幅制限回路3における利得を急激に低下させることにより点11にて初めの待受状態に復元することが出来る。また警報停止用制御電圧は光学的電磁的音響的に列車から間接的に検出器7により検出して有極継電器  $K_2$  を動作することも出来る。

(後藤)

第1図



第2図



第3図

