

MARS-101 座席予約システム用 リアルタイムプログラム

Real Time Programing for MARS-101 Seat Reservation System

井 上 晴 雄* 佐 々 木 敬* 大 須 賀 節 雄**
Haruo Inoue Takashi Sasaki Setsuo Ōsuka
安 部 城 一*** 杉 田 一*** 上 野 三 朗***
Jōichi Abe Hajime Sugita Saburō Ueno

内 容 梗 概

端局より発信された呼をオンラインで処理するプログラムを以下リアルタイムプログラムと呼ぶことにする。本文ではリアルタイムプログラムをまず基本構想の生い立ちから述べ、次で呼の流れを主体としてプログラム構成を示し、最後にプログラム上の特色と考えられる点を列記している。

1. 緒 言

MARS-101 システムは MPC の統制の下に全システムが動作している。その MPC の座席予約システムとしての機能を規定し支配しているものがリアルタイムプログラムである。

処理機能を規定する点からプログラム開発には国鉄・日立両者の完全な協力を必要とし、ほぼ2年にわたる労作の結果さる39年2月一応の完成に達した。

2. リアルタイムプログラムの基本構想

リアルタイム処理システムにおいて最も重要なことは、より高い信頼性と処理能力を与えることである。また、ソフトウェアに要求されることは、できる限り使用メモリを少なくして、ハードウェアの機能を十分に生かし、信頼性、性能の面でシステムの能力を最大限に発揮させることである。しかし、信頼性を増すためには、時間とメモリを余分に必要とし、また、処理速度を早くしようとすれば、メモリをより多く必要とする。したがって、結果的にはこれらの間に妥協点をきめることになるが、一方を満足させるために他を犠牲にするという、二者択一的なものにせず、できる限り、これらのいずれをも満足するような方法を見いだすことが、リアルタイムシステムでは最も重要である。MARS-101 のリアルタイムプログラム、特にシステムの制御を行なうプログラムに関して基本的方針とでも言うべきものは、できるだけ自然の流れにさからわず、無理のないプログラムを作ることである。たとえば、あとに述べるように、このシステム内の情報の流れは、一種のランダムプロセスであり、常に平均的な状態から変動している。プログラムでは、それを無理に修正して平均の状態にするというようなことはせず、ある範囲内での変動を許し、それが自然に戻るような制御を行なう。変動が一定の限度を越えるような場合、あるいは、処理能力を低下させるような情報の分布がたまたま生じたような場合にはじめて積極的にそれを修正するような作用を与える。これによって、不必要な時間の損失を最小にすることができる。

MARS-101 のリアルタイムプログラムを作成するにあたって、まずあげられねばならないのは、Duplex System および Concurrent Processing と、それにより Multi Programming 方式が用いられることである。誤りを減らし、かつ事故の際の処理が中断されるのを防ぐために2台の装置が同時運転され、リアルタイム処理を行な

いながら、両装置間で情報の照合が行なわれる。そのために両装置間に同期の問題が生ずる。両システムをプログラムによって強制的に同期させることは損失を増すことになるから、必ずしも同期していなくても照会できるから、自然に両システムが同期するような方法が用いられている。

Concurrent Processing に関しては問題はやや複雑である。MARS-101 は、TRC, TC, FC ならびに SC が十分な Concurrency を有することによって、はじめて能率のよい処理を行なうことができ、システム処理能力を最大にすることになる。しかるに、リアルタイム処理の場合、これら各処理装置が相互に無関係に動作する訳にはゆかない。これら各装置は処理される情報の手順を分割して、それぞれ異なった部分を受け持っているのであるから、特定の装置(たとえば FC) が処理にはいるためには、そこで処理される情報が、その前段の装置(TC) で処理が済んだものでなくてはならない。しかも、これら各処理装置が Concurrency を有するためには、システム内に少なくともこれら各装置の数以上の情報がいってなくてはならない。実際には各装置で処理に要する時間はランダムであるため、この数倍のものがはいらねば Concurrency を増すことはできない。このため、システム内の各所で情報の待ち合わせが生ずる。この待ち合わせの数が多いほど Concurrency はよくなるが、反面、待ち合わせのためのメモリーが増大し、さらに個々の情報についてみると待ち時間が増えて通信回線や、送受信制御装置などが余計必要となる。したがって、待ち合わせの数にも Optimum なものをみいだすことが必要となる。あるいは、システム全体のバランスを保つように、そして各装置と SC との情報交換の方法に関連して、情報の流れかた、待ち行列の数と長さ、Working Storage の数および I/O Buffer の数などがきめられねばならない。また、プログラムはその流れを円滑に行なわせるように作られねばならない。このためにはプログラムの作成に先だってシステムの分析が必要である。また、Concurrent Processing にともない Multi-programming 方式が用いられているが、この特長として各装置からの割込みによって生ずる K 10~K 15 のプログラムは、相互に独立に作成することができ、プログラマは Multi-programming の一部であるということ意識せずに作ることが許されねばならず、また、そのプログラムが他のプログラムと干渉しない条件が満足されることが必要である。しかるに、前述のように各装置相互間の動作は関連し合っているから、その割込みにより生ずる各プログラムも相互に無関係ではない。そこで、このプログラム相互の関連し合う部分を抜き出して、別の独立したプログラムによりまとめて行なうことにより、上記の条件を満足させることができる。このプログラムは後述の

* 日本国有鉄道

** 東京大学

*** 日立製作所神奈川工場

Supervisory Program の一部である。このように、K10～K15の各プログラムはシステム全体から見れば、局所的な処理、または制御を行なうことになる。

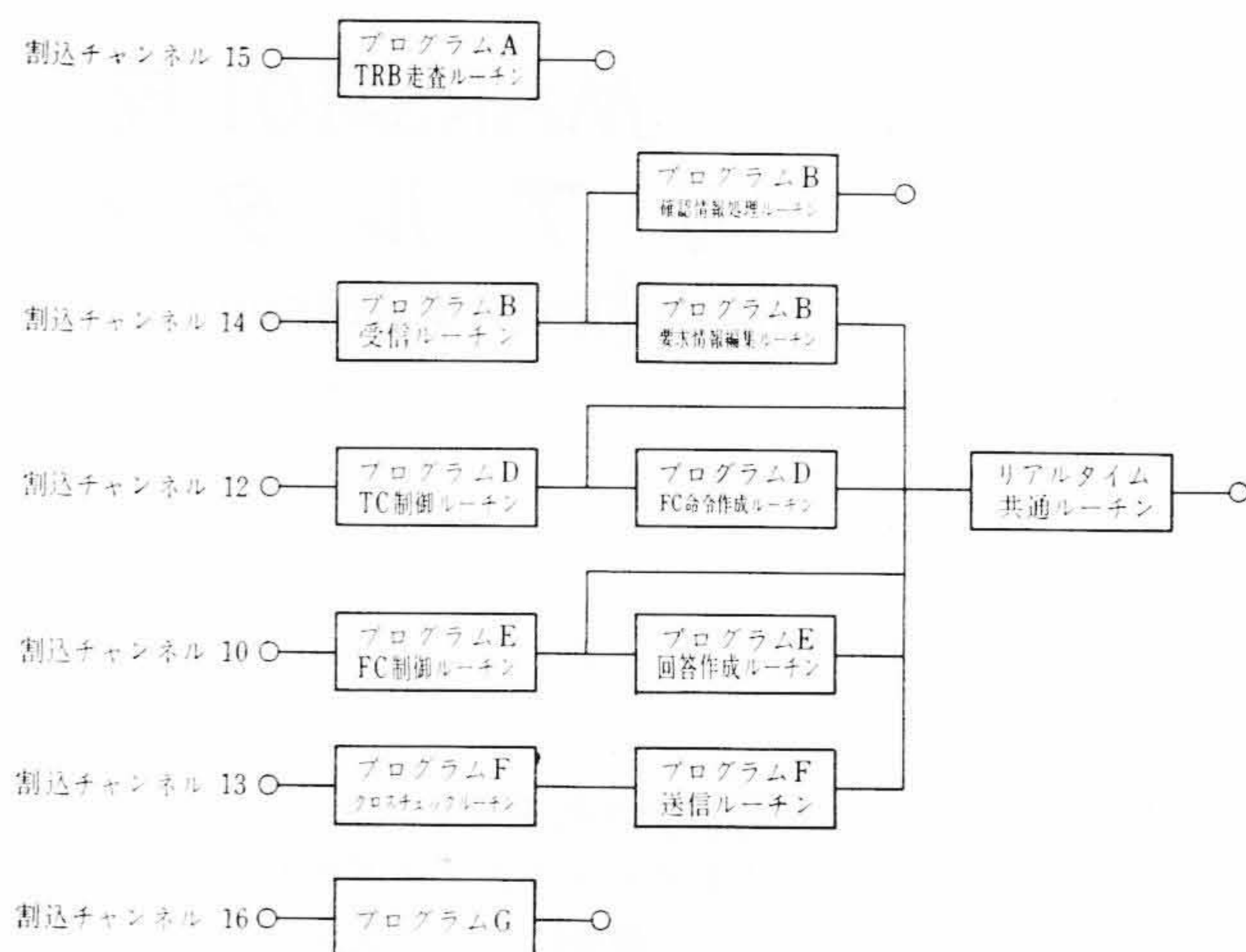
システムの情報の流れは、これら K10～K15 の各プログラムが成立するごとに変化するが、これら各プログラムの成立はランダムであるから、システム間の流れもランダムになる。このシステム内の流れは総括的に調整する必要がある。この調整を行なうために、常にシステムの状態を全体的に見て制御を行なう Supervisory Program が必要である。流れの制御を行なうためには各装置から割込みの制御をせねばならず、これが Supervisory Program の重要な役割であり、このほか前述の各プログラム間相互の関連をつけ、また必要に応じてシステム内の情報の順序づけを行なうための機能を備えている。このように Supervisory Program はシステム制御の面で非常な役割りを有することになるが、他方各装置と SC との Concurrency を増すために、この Supervisory Program の優先順位は他のプログラムより低位におかれる。Supervisory Program にこのような機能を与えることにより、K10～K15 の各プログラムは非常に限定された範囲の機能を有しておればよい。すなわち、各対応する装置から返されてきた情報の処理と、それら装置に新しく情報を与えること、エラーチェックとの対策および限定された制御機能である。さらにこれら各プログラムのうち共通なルーチン、または非常に似かよっていて、Index Modify などによりまったく同じルーチンにおきかえられるものは、できる限りまとめて Supervisory Program に含めるか、あるいはサブルーチンにすることが望ましい。このようにして、リアルタイムプログラムの基本形が定められるが、当然のことながら、これらすべてのプログラムはできるだけ短く作るように努力しなければならない。また以後個々のプログラム作製に際しては、将来制度変更や料金変更、ダイヤ変更などが生じてプログラムは変更しなくてすむように、あるいは局部的な変更で済むような方法を用いること、また収容列車数の増大とともに装置の増設があるような際にも、前のプログラムはそのまま使用できるような融通性を与えることが必要である。さらに割込みの処理やサブルーチンの使用の際の各インデックスの退避にはスタックなどの方法を用いて使用メモリを減らし、またシステム内で多量の情報の移動を最小限に押えて時間を減らすために、情報ごとにインデックスを付して大半の移動をインデックスで行なうなど、部分的に各種の方法が用いられている。

3. プログラムの構成

次にリアルタイムプログラムが関与する処理の流れを、最も典型的な呼の場合を例にとって述べる。呼は遠隔地の窓口で発生したもので、ある列車の座席を予約しようとする内容を持ち、処理の全過程を通じて装置は正常に動作したとする。

端局装置において、中央処理装置へ送るべき情報がすべてセットされ発信キーが倒されると、センタを指定する局選択符号が送出され、これによって電信交換機が動作して端局装置と中央処理装置の TRC との間の回線が構成される。TRC は直ちに SC に対して、これから受信する情報（要求情報）を格納すべきバッファレジスタ (TRB) の割当てを要求する。この要求は割込チャンネル 15 を介して行なわれ、もし割込が許される状態であればプログラム A (TRB 走査ルーチン) が開始する。

TRB は SC の主記憶装置の一部を使っており、4 語を 1 組として 32 組が準備されている。各 TRB が Available であるかどうかは特定のアドレスに表示されており、プログラム A はこの表示を走査して最初に見つけた Available な TRB の番号をレジスタに記入するとともにそのことを TRC に知らせる。バッファの割当てを受けた



注：プログラム A～G は各プログラムの仮称、プログラム C は現存しない。

第1図 リアルタイムプログラムの構成

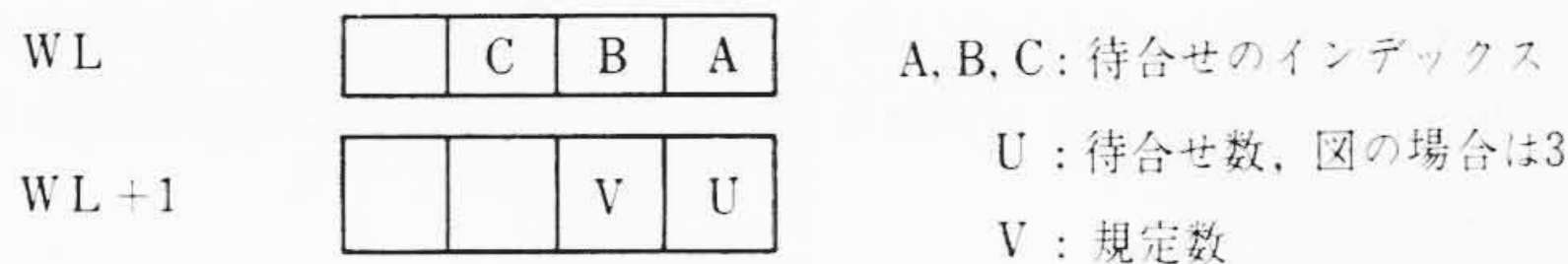
TRC は引き続き 100 ないし 130 ビットからなる要求情報の受信をはじめめる。

この受信が完了したとき、TRC から SC に割込要求信号 (チャンネル 14) が出、割込成立条件が満たされているとプログラム B が始まる。プログラム B は第1図のように三つの部分からなるが、要求情報の受信の場合はまず受信ルーチンに入り、TRC の受信動作が正常であったかどうかをアラームビットによってチェックする。それが正常ならばこの呼の処理に必要な作業用番地 (TS) 8 語を確保し、要求情報についてのパリティチェックを行なう。なお以後この呼は割り当てられた TS の先頭のアドレスをそのインデックスとして取り扱われる。

次に要求情報編集ルーチンは、TC に与えるべきデータ、たとえば列車名、乗降駅名などを要求情報の中から extract して、あるいは符号の変換を施して、決められたフォーマットに並べ替え TS に格納しておく。これでプログラム B での処理を終わり、リアルタイム共通ルーチンへ移って 3.1 で述べるように Waiting Line の制御を行なう。いま着目している呼はここで Waiting Line D に移され、その表示が特定のアドレスになされる。共通ルーチンはまた、リアルタイム処理の Master Control Program であるプログラム G が割込みチャンネル 16 で成立できるような準備をして終了する。

プログラム G については 3.2 でも説明があるが、これは各 Waiting Line の状態を表示しているレジスタを一定の順序に従って走査し、最初に逢着した呼が属している Line の処理が進行するようにトラヒックの制御を行なうのである。各 Waiting Line は割込みチャンネルに対応しているので、割込みチャンネルのマスクフリップフロップを操作することによって金物で決まる優先順位を修正して、その時々トラヒックに応じたプログラムの進行を制御することが可能である。中央処理装置内での情報の分布状況から Waiting Line D の処理を行なうのが目下のところ最適であるとプログラム G が判断すると、プログラム D のマスクフリップフロップ K42 がセットされる。プログラム D が成立するためには、TC がある必要はあるが、これはフリップフロップ K12 で表示される。

プログラム D は第1図のように、TC 制御ルーチンと FC 命令作成ルーチンとからなる。TC 制御ルーチンはプログラム D が、新たに TC に与えるべきデータがあって割り込んだか、あるいは TC における処理が終了して割り込んだかを判断して分岐する。いま問題にしているのは前者の場合であって、さきに TS 中に作成しておいた TC の制御語 1 語を TC 用のバッファレジスタに移し TC を起動



第2図 WL (Waiting Line) の構成

する。起動はフリップフロップ K 12 をリセットすることでなされる。この後コントロールはリアルタイム共通ルーチンへ移り、Waiting Line がシフトされる。

TC の内部での処理は別項に述べられているので省略するが、一連の索表動作が終わったときの出力、すなわち座席ファイルの位置、料金種別、乗降車区間などは決められたバッファレジスタ 5 語にメモリシェアで書き込まれる。そして TC はフリップフロップ K12 をセットして割込みを要求するが、前もってプログラム G により準備されていれば再びプログラム D が始まる。今回は TC 制御ルーチンで TC における処理終了のほうへ分岐し、TC での処理中に生じたアラームを調べ、それが異常なれば要求が予約期間内にあるか、ダイヤ変更のある列車に関係するかなどの検定を行ない FC 命令作成ルーチンにはいる。このルーチンでは、TC で処理された結果を基にして FC に与える制御語を作成する。そのおもな内容は操作種別(座席データの読出し、サーチ、更新その他)、列車番号、座席ファイルの位置、乗降車区間、座席数などで、これらを決められたフォーマットに従い 2～4 語にまとめ TS に格納しておく。このうち共通ルーチンで Waiting Line のシフト、プログラム G で次に成立すべきプログラム E のための準備がされる。

プログラム E は FC 制御ルーチンと回答作成ルーチンとからなる。このプログラムの開始の条件はプログラム D の場合と類似しており、FC をこれから起動しようとする場合と FC が与えられた処理を完了した場合とがある。FC 制御ルーチンではまずこの分岐をし、前者の場合にはさきに作成した FC への制御語をバッファレジスタに移して FC を起動(フリップフロップ K 10 をリセット)する。FC はこの指令に基づいて座席のサーチ、ファイルの更新などを行なうが、この間 SC では waiting line のシフトやプログラム G によるトラヒックの制御が行なわれる。FC の処理結果は TC の場合と同様に SC のコア記憶装置に書き込まれ、動作の終了はプログラム E への割込要求信号として SC に知らされる。

ここでプログラム E が再びはじまり、FC 制御ルーチンで FC からのアラームのチェックをしたうえ回答作成ルーチンに移る。端局装置へ回答を送出するための符号の変換、パリティビットの付加、料金の計算、取扱番号の加算などがここで行なわれる。Waiting Line のシフトが行なわれ、かくて着目していた呼に対する処理は他系での結果との比較、磁気テープへの記録、端局装置への送信を残すのみで、これらはプログラム F で行なう。

プログラム F は割込チャンネル 13 を使用する。プログラムは、並列運転をしているとき両系の処理結果を比較照合するクロスチェックルーチンと要求情報や回答の磁気テープへの記録、TRC に対する送信の制御語の作成と起動、TS の回復などを行なう送信ルーチンとから成る。

回答情報が端局装置で受信され終わると、確認情報(DSG)が中央処理装置に向かって送出される。したがってプログラム F の次には、DSG を受信するプログラム B が成立できるように準備される必要がある。DSG はパリティビットを含んで 5 ビットで、これが受信され終わるとチャンネル 14 へ割り込む。プログラム B の受信ルーチンは DSG を調べて端局装置が異常なく回答を受信したことを確かめ、DSG 処理ルーチンでは回線を解放するための準備と DSG の磁気テープへの記録を行なう。

最後に回線の解放はプログラム F で行なわれるが、この場合にはプログラム G の制御を受けることなく、DSG 処理ルーチンでフリップフロップ K 13 を直接セットしている。

4. プログラム上の特色

4.1 リアルタイム処理を行なうこと

システムをリアルタイムで使用するため高能率化と高信頼性が要求される。本プログラムでは前者の目的のため待合せラインを、後者のため各種異常のチェックと自動再送もしくは再送要求機能を設けた。

(a) 待合せライン (以下 WL と略記)

入力された要求情報は Subprocessor である TRC, TC, FC のとおり順次処理されるがシステム全体として処理能率を上げるために、入力情報の分布状態を調べて Subprocessor の割り込みを制御することとし、各 Subprocessor が動作終了すると直ちに処理待ちの情報が与えられるように Subprocessor の一つに対応して 1 個の待合せラインが設置される。

また、一つの入力情報は処理途中ではさまざまな形に分解され、合成される。このため一つの入力情報には数語からまとまった作業番地(TS と略称)が対応し、この TS の先頭アドレスがインデックスとなって WL を流れるようになっている。

WL は割込レベル 14 (TRC 処理要求), 12 (TC 処理終了信号), 10 (FC 処理終了信号), 13 (TRC 命令要求) にそれぞれ 1 個ずつ設けられ 4 個までの待合せが許される。各 WL は 1 語ずつより成り、第 2 図のように、その次の番地に現在の待合せ数とその規定数を持っている。

各割込レベルに付随するルーチンは、WL にあるインデックスによって TS にある情報を処理する。処理が終了すれば、自己の WL と、次に待ち合わせすべき WL のアドレスをそれぞれ 2 個の B レジスタにセットし、共通ルーチンにはいる。共通ルーチンでは、この B レジスタにより手前の WL から次の WL にインデックスを移し、待合せ数の計算、および待合せ数と規定数の比較を行ない、規定数に達しているかどうかに従って FG (Master Control Program 参照) に割込み禁止、または許可パターンをそれぞれセットし終了する。

(b) 情報のチェックと異常処理

リアルタイムシステムとして考え得る誤操作や誤動作からシステムの動作を保護するために以下のチェックと異常処理を行なっている。

端局装置から中央装置に送られる情報には、要求情報と中央装置から返送された回答に対する確認情報の 2 種類があり、これらの情報は金物、およびプログラムの両方からチェックされる。要求情報のうち、端局側の誤指定と考えられるものは再考コードが、端局装置および伝送上の誤りと考えられるものには再送コードがそれぞれ返される。また、確認情報の不良に対しては中央装置の回答を再度送信するが、それに対してさらに不良な確認情報がきても回答は送られず、送受信はいったん終了する。

(1) 要求情報に対するおもなチェック

(i) 割込レベル 14 (TRC 処理要求) で行なわれるもの
パリティ不良(再送)、あり得ない操作、日付、号車番号の指定(再送)、一等寝台の中段指定(再考)、人数と座席位置指定数との不一致(再考) *注 (再送)、(再考)は端局に対する回答以上のことが発見されると、TS には FC, TC 索引不要および、再送または再考のビットがそれぞれセットされ前述の待合せラインの上を流れる。

(ii) 割込レベル 12 (TC 終了信号) で行なわれるもの

列車名、駅名の誤指定(再考)、上下列車のまちがい(再考)、予約期間外指定(再考)

ここではFC索引不要、および再考のビットがTSにセットされる。

(iii) 割込みレベル10(FC終了信号)で行なわれるもの

指定号車と等級の不一致(再考)、存在しない号車、または等級(再考)

この時点で、(i)、(ii)、(iii)の結果によって端局に送信すべき回答情報が作成され、クロスチェックの待合せラインに流れる。

(2) 確認情報に対するチェック

確認情報には、受信完と再受要求の二種類があるが、処理上は受信完か、それ以外のものがチェックされ、これと送信回数が判定される。受信完、および二度送信されたものは処理終了となるが、それ以外のものはもう一度送信される。

4.2 多重プログラミング

与えられた速度の計算機を実時間用途に使用し効率(処理能力)を高めるためにはいわゆる割出しでなく割込みによる処理が必要と言える。MARS-101の統制処理装置は周辺装置としてTRC、TC、FC、MTC、HSPCなどを従えており、周辺の処理装置に同時処理をさせて処理流量の増大を図るために、プログラム面では割込みの上に割込みを許す多重割込み方式をとっている。

多重割込みを可能とするために開発したプログラムのうち主要なものを次に示す。

(1) ISR(割込み処理ルーチン)

割込みが発生すると今までのプログラムのあと始末と新しい処理の準備が必要となる。このためISRでは全レジスタとK36フリップフロップの退避と回復を行なっている。さらに、多重割込みに備えて、割込みプログラムごとの固有の退避、回復用メモリを廃止し、被割込みプログラムすべてに共通のSTACK構成を採用している。すなわち一連の割込みが発生すると、STACKの底から優先度の低いプログラムの使用したレジスタ群の値が積み重なり、STACKの中には、自動的に優先度のこう配が発生する結果、高位のプログラムが終了して、被割込みプログラムを再開する場合、STACKを上から回復してやれば、最優先順位と一致する利点が見いだされる。

このようにして、多重割込みに際してのレジスタの退避回復上の問題は解決され、現在7段(STACK容量からの制限)までの多重割込みを許し、容量以上割込みが重複すれば、一時的に割込み要求を蓄積しておき、現行処理が完了し、STACK容量に余裕ができたとき、始めて要求に応ずる方式を採っている。

(2) IOCC(入出力制御ルーチン)

多重プログラムを許すと、直ちに問題となるのは、異なった優先順位(以下レベルと略称)のプログラムが共通に使用する部分の制御管理である。たとえば低位のプログラムがチャンネル2により終了次第割込ませる予定で、印字を行っていたとする。このとき割込んだ高位のプログラムがチャンネル1によってPTRより情報を入力し、チャンネル1のほうがわずかに早く動作完了して、割込みフリップフロップK8(I/O終了表示)リセットを含む処理を正規に終了して、被割込みプログラムを回復したとする。しかし、低位のプログラムは、I/O終了による割込みを持っており、その終了表示は高位のプログラムにより抹消されているので、もはやこのプログラムは進行できなくなる。

このような異なるレベル間の動作混用が起これぬように、制御、保護するプログラムが必要であり、MARS-101では、各レベルのプログラムがI/Oを欲するときは、必ずIOCCに実行を依頼する形で統制を行なっている。

IOCCは各レベルからI/Oチャンネルへの依頼受付、I/O命令発行、終了回答チェックならびに依頼主に対する終了通告を行なっている。

(3) MTCP

IOCCと同様に、異なるレベルのプログラムがMTCを使用する点で多重プログラム対策の一環としてMTCPが作られた。ただし、IOCCの場合とは使命のうえで次の点で異なっている点を紹介したい。

(i) 統計の必要上AGT-MPC間の送受信情報をオフラインでMTに記録するプログラムである。

(ii) システム内の並列処理に、待合せ損失が発生しないように、M/T用バッファが一ぱいになった時点でMTCPのM/T要求STACKに用件を記入するだけで、終了を待たずに、次のプログラムに進行している。書込み完了以前に呼がはいってくれば、4件までをM/Tバッファ外に蓄積し、それ以上はいれば、リアルタイム処理にブレーキをかける制御を行なっている。

(iii) 書込み、読出しの誤りに対して行なう再試行機能に関しては一般用計算機の場合と同様であるが、さらにオンライン書き込み中のMTHが再試行不良となった場合、別MTHへの自動切替手段が用意されている。

(4) プログラムG(Master Control Program)

入力情報は、数種のSubprocessorで順々に処理されて行くのであるが、できるだけ停滞することなく円滑に行なうためハードウェアで固定されているSubprocessorの割込みを入力情報の数、各待合せ行列にある処理待ち情報の分布状態などによって制御し、システム全体の効率を上げることが考えられた。

このプログラムではリアルタイム処理で起こってくる割込みのうち、割込みレベル1①(FC終了信号)、1②(TC終了信号)、1③(TRC命令要求信号)、1④(TRC処理要求信号)の各割込みをそれぞれのマスクフリップをセトリセットすることにより制御している。制御は以下の方法で行なわれる。

(1) 各レベルに直結する制御プログラムは自己の待合せ行列の状態を特定レジスタ(コアメモリ、AGと略称)にセットし、次に行くべき待合せ行列の状態をみて、AGに対するマスクパターンを特定レジスタ(コアメモリ、FGと略称)にセットする。

(2) 各レベルのプログラムは終了時には自己のマスクフリップフロップをリセットし、かつフリップフロップ16をセットし割込み制御プログラムに起動をかける。

(3) もしいづれかの障害、再試行不良などがあればFGのマスクパターンをリセットし再びその割込みが起これないようにする。

また、レベル14では入力情報の個数が規定数以上になるとその旨を特定番地にセットする。

割込制御ルーチンは(2)によって起動される。まずAGとFGとの論理積がとられ次に割当てべき仕事が見つかる。もしあればその割込レベルに対応するフリップフロップがセットされ、なにもないときはフリップフロップ16をリセットし、いったん終了する。仕事の搜索は割当てべき仕事が無くなるまで何回でも続けられる。また、レベル14の割当ては上記(3)を考慮に入れて行なわれる。

4.3 二重系制御

本システムは、同じ機能をもつ2台のシステムから構成され、システム相互は関連を保ち、情報交換を行ないながら動作する。

システムの動作状態には、オンラインでは片方が主系、他方が従系となって動作する並列動作と、他系に関係なく動作をする単独動作が可能である。

また、システム間の情報交換には、他システムのアドレスの内容

を読み取るメモリモードと、他システムに積極的に割込みを起こす K 6 の割込みとが設定されている。これらは、二重系の大きな特長である相互照合チェックに用いられている。

以下二重系制御の目的で用意されたプログラムを述べる。

(a) 動作状態制御プログラム

2 台の MPC の動作状態を制御するプログラムで、割込レベルは 2 である。動作状態には、オンライン並列動作、オンライン単独動作、オフラインのそれぞれの状態があり、これらは制御卓より指定される。

MPC には、動作状態を示すフリップフロップ 66 (オンライン主動作中)、67 (オンライン従動作中) があり、さらに制御卓からの指令を示すフリップフロップ 23 (オンライン主動作指令)、24 (オンライン従動作指令) がある。

プログラムは次の処理を行なう (以下、フリップフロップは K と略称)。

(1) オンライン並列動作指令 (K 23, K 24 とともにセット) されたとき、ひとまず K 66・K 67 とし終了する。この場合、従動作を指令された系は、主系の動作状態により K 23 となり、再び割込みが起こるからこのとき K 66・K 67 とし処理を終わる。

(2) オンライン単独動作指令 (K 23 セット) されたときは、K 66・K 67 とし終了する。

(3) オフライン指令 (K 23・K 24) のとき、K 66・K 67 とする。

(4) オンライン動作中、それ以上、処理続行不可能であるような異常時態が発生すると、それを発見したプログラムは特定番地にその旨をセットビットし、K 66、または K 67 とする。このとき、本ルーチンはその特定番地を判断し、制御卓へ警報表示を行ない、いったん停止 (Halt Stop) して操作者からの指令を待つ。

(b) K 6 プログラム

他の MPC からの信号に応じて動作するプログラムである。このシステムの運営上、他系について制御を必要とする場合がかなりあり、そのため次のような操作種別をコントロールワードにより与えることにしてある。

(1) 他系よりプログラムを読み取って、その先頭から K 6 または K 18 のレベルで実行する。

(2) 他系に情報を移す。

(3) 2 系統間のクロスチェックを行なう。

このプログラムは現在系統間の FC または TC の情報転送に、またクロスチェック不一致の際、従系に不一致状況印刷をさせる用途に使われている。

(c) 相互照合チェック (クロスチェック)

本システムはさきに述べたように、大きな特長の一つとして、回答情報の信頼性を上げるために、一つの入力情報に対し、二つの系が独立に処理を行ない回答を作成する。得られた二つの回答はクロスチェックされ、合っていればその回答情報は AGT

に送信される。

以下クロスチェックの方法を述べるのであるが、その方法には二つの系が非同期であることと、マルチプログラミングを採用しているという二つの大きな制約がある。独立に処理されて来た呼は、クロスチェックを行なうために F-ラインに行列を作る。いま、ある一つの呼に着目すると、非同期であるためにどちらかの系が進んでいる。そこで F-ラインの先頭に呼がさきに到着した系、すなわち進んでいる系が、自系の F-ラインの先頭の呼と、それに対応する他系の呼は F-ライン (F-ラインは四つの呼まで収容できる) のどこにいるかを探し出し、クロスチェックを行なう。他系の対応する呼を探すにはそれぞれの系の F-ラインをとった呼の差 (CCR という記号で表す) をカウントしておけば、対応する呼は他系の F-ラインのどこにいるかが容易にわかる。進んでいる系がクロスチェックを行ない、遅れている系は F-ラインの先頭に来て、他系からのクロスチェックの終わるのを待って、自系はクロスチェックを省略する。クロスチェックが正しい結果で終わった場合は、その回答情報は TRC に送信を命令するルーチンに結合される。悪い結果で終わった場合、計算機は両系ともオフラインとし、相当の処理を行なわねばならない。しかし、その生じた時刻が最繁時である場合、でき得るなら避けたいので、二回まで保留の回答を AGT に返し、三回目が生じたときにのみ、オフラインのルーチンに結合する。

また、回答情報のクロスチェックを行なう前に、T. C および F. C の処理が良い結果で終わっているかどうか、自他両系についてチェックする。もしエラーで終わっている場合、それが主系または主従両系で生じている場合計算機はオフラインとなり、従来のみの場合は、自動的に従系を切りはなし、主系が単独主となって運転を継続する。

5. 結 言

開局以来リアルタイムプログラムは重大な BUG もなく、大過なく役割を果たしつつあり、このような新形式のプログラムの動作実績が作られていくことは誠に喜ばしい。

終わりに長期にわたる作成期間中、執筆者ら以外に石原嘉夫氏、渡辺寿夫氏、服部治雄氏、沢田正方氏、林義郎氏、酒井芳治氏、栄辰平氏、小川茂氏、善如寺正雄氏、吉村善兼氏、渡辺和夫氏、伊東俊一氏、大塚晋氏、村島博氏、米谷勇氏、坂巻昭一郎氏 (以上が日本国有鉄道)、また東京大学の穂坂衛氏、および安藤文雄氏、高橋茂氏、清水三重二氏、谷恭彦氏、前田治氏、前田寿一氏、西門左千夫氏、浅井政幸氏、鴨川和正氏、小山澄氏、久木田須美笑氏 (以上が日立製作所) の方々に計画、指導、作成およびデバッグに参加いただいた。これらの方々の協力により MARS-101 のリアルタイムプログラムが完成したことを明らかにし、その功績に対し感謝の意を表わす次第である。