

UDC 681.323:
62-5

航空宇宙技術研究所報告

TECHNICAL REPORT OF NATIONAL AEROSPACE LABORATORY

TR-511

シミュレーション用複合計算機 (FSK-II) の
データ転送と制御

原 田 公 一

1977 年 8 月

航空宇宙技術研究所
NATIONAL AEROSPACE LABORATORY

シミュレーション用複合計算機 (FSK-II) の データ転送と制御*

原 田 公 一**

Data Transmission and Control in Computer Complex (FSK-II) for General Purpose Flight Simulator

By Koichi HARADA

ABSTRACT

This paper contains a report on data transmission and control in a computer complex, for real time simulation, which is constructed with five minicomputers.

One computer, No. 0 Computer, controls the input-output bus circuit which broadcasts data and commands to the other computers. The user can produce task-programs without consideration to input-output instructions to the other computers.

The linkage editor identifies and classifies the variables to be transmitted. It also generates control information with which the No. 0 Computer transmits variables and edits them for the analog-digital, digital-analog converter. The supervisor, which is in No. 0 Computer, repeats communications at intervals of 10 ms with the real time system monitor in the other computers.

A method for broadcast-burst data transmission, generation of control information and synchronization of the computers are described.

1. はじめに

複雑なシステムを設計・製作もしくは改造する場合、その構成要素について十分に検討がなされても、組立てたシステムの性能・機能を完全に予測・理解することは、その設計者といえども困難な場合が多い。

システムが複雑・高価であればある程、不具合による経費・人手・時間の浪費や、事故等の取返しのつかない事態を避けることが求められ、さらに人間が組込まれるマン・マシン・システムの場合には、人間の感覚をも入れた検討が要求される。

航空機の開発の場合、テスト・パイロットをその機体と同じ動作をする装置に乘坐、現実と同じ環境を再現し、操縦性・機能等を検討する。

これをシミュレータ試験と言い、この装置を研究開発用または汎用フライト・シミュレータと言う。

航空機の運動は、力学系として数学モデルで表現できるので、パイロットに航空機に乗っている時と同じ感覚を持たせるためには、その数学モデルを計算機で模擬し、その信号で、油圧・電気サーボ機構を通じてコックピット (Cock pit)・計器・外部視界 (Visual Display) を動かす必要がある。

シミュレータに使用される計算機は厳密な意味での実時間処理が要求されるため、従来アナログ計算機が使用されていたが、航空機の大型化・高性能化に伴い、シミュレータ試験も、大規模・高精度に行う必要が高まり、

- (1) 演算精度が良い。
- (2) プログラムの蓄積ができ、大規模なシミュレーションに適する。
- (3) 信頼性が高い。

* 昭和52年4月8日 受付

** 計測部

(4) 保守維持が容易である。

(5) 非線型をも含めた複雑な演算ができる。

等の理由から、ディジタル計算機が使用されるようになってきた。

わが国では航空宇宙技術研究所計測部に昭和38年、アナログ計算機・コクピット・フライトテーブル・模擬視界装置からなる汎用飛行シミュレータが建設されたが、昭和45年頃から、計算機部の部品製造中止等のため、維持困難となり、また近年のシミュレータ試験への要求に対処するため、計算機部をディジタル計算機に更新する作業が昭和47～49年度にわたり行われた。

通常、ディジタル計算機による場合、大型航空機のシミュレーション・プログラムは20kW～80kWの規模となり、その周波数特性を5Hz以上とした場合、平均命令実行時間は500 ns以下でなければならない。

これは、ミニコンの4～5倍の演算速度であり、当時、このような要求を単一計算機で満たすためには高価な大型計算機を用いるのが通常であった。

他方、フライト・シミュレーションのプログラムは

- i) 運動力学
- ii) エンジン
- iii) 空気力学
- iv) 航 法
- v) 外部機器インタフェース
- vi) シミュレーションの監視

等のサブプログラムに分割でき、それらの間のデータ送受も全体で数十語程度であるとの判断から、記憶装置と演算回路が同期しているミニコン(MELCOM-70)5台をチャンネル間で結合し、各計算機で各々のサブプログラムを処理し、全体で一つのシミュレーションを行う計算機システムMGP-1050 が三菱ブレンドンより提案された。

複数の演算装置をレジスタレベルで結合し、流体力学の解法(偏微分方程式の解法)・パターン認識・マトリックス演算・確率過程の解法^{1)～6)}に適用する提案や、複数の計算機をチャンネル間で結合したComputer Complex⁷⁾⁸⁾(以後複合計算機と言う。)をプロセス制御等に応用した例は、以前にもある。

しかし複合計算機に汎用性を持たせ、応用プログラム作製手段を備えるためには、その結合回路・モニタの構成・計算機間のデータ転送・転送制御情報の生成法・応用プログラムの記述言語・プログラムの分割法等技術的にも未解決の点が多くあった。

昭和47年、更新作業が進行するにつれ、このMGP-1050 システムは、中央計算機の記憶装置を中継して、

データの交換を行うように構成されていたため、計算機間のデータ転送に多大な負荷を持っていることが判明した。

さらに、それは固定または、単能プログラム組込式のトレーナー用計算機として考えられていたため、汎用飛行シミュレータとして必要な、応用プログラムの作成法、特に分割作成されたサブプログラムを結合し、一つのシミュレーション・プログラムとして動作させる手法を欠いていた。

そこで、昭和48・49年度では、

- (1) データ転送の負荷を軽減させるため、
 - (a) 転送回路を改造し、1対多の転送を行うブロードキャスト(Broadcast)転送回路とする。
 - (b) 転送データを連続する領域に確保し、転送しやすくする。
 - (2) 応用プログラム作成法を確立するため、
 - (c) シミュレーション言語(RTSL)を開発する。
 - (d) 各計算機ごとに分割作成されたサブプログラムを結合し、一つのシミュレーション・プログラムを発生させるソフトウェア(LEP)を開発する。
 - (e) 各計算機に分割されたサブプログラムを矛盾なく動作させるため、計算機間のデータ送受法を確立する。
 - (f) 複合計算機の運用法、シミュレータ運転に必要な機能、および、そのためのハード・ソフトの構成等、不明な部分が多くあるため、必要不可欠な機能を明確にし、システムに組込む。
 - (3) 改造した計算機システム(以後、これをFSK-IIと呼ぶ。)の性能および応用プログラム作成法を確認するため、大型ジェット旅客機のシミュレーション・プログラム(Flight Simulation Programmed Package 略してFSPPと言う。)を開発する。
- ただし、シミュレーション言語の開発は、遅れることが予想されるため、FSPPはアセンブラ言語で記述する。

の作業を三菱ブレンドンに発注した。

本装置は、昭和50年6月に大型ジェット旅客機のシミュレーションを実現し、その機能・性能が確認されている。

複合計算機の場合、計算機間のデータ送受方法は、その性能・応用範囲・応用プログラム作成手法に大きな影響を与える。

本文では、FSK-IIの計算機間のデータ転送とその制御方法について述べる。

2 略 記 号

RTSL	Real Time System Language
FSPP	Flight Simulation Programmed Package
FSK-II	Flight Simulator for Kogiken Version II
RTC	Real Time Clock
DMA	Direct Memory Access
ICCL	Inter Computer Communication Logic
HSBC	High Speed Bus Coupler
PIO	Programmed Input/Output
LSBC	Low Speed Bus Coupler
CCW	Channel Control Word
SCP	System Control Program
RTSM	Real Time System Monitor
LEP	Linkage Editor Program
DI	Discrete Input
AI	Analog Input
AO	Analog Output
DO	Discrete Output

3. 実時間シミュレーション・システムについて

実時間シミュレーションとは、時々刻々と変化する実世界の現象と常に同じ状態にあるようなシミュレーションのことである。

したがって、現実の時刻と計算機内のプログラムが持っている時刻とが一致している必要があり、計算機は、

初期値を与えられると、対応する時刻までに状態量を演算し、かつ出力していなければならない。

実時間シミュレーションは、人間を制御ループ内に含むシステムの評価、実験機器（ジャイロ・オートパイロット等）の性能試験、パイロット等の訓練のように環境を実時間で模擬する必要がある分野では、不可欠な手法である。

従来、実時間シミュレーションには、演算速度・費用効果・使い易さの理由からアナログ計算機が使用されていたが、近年デジタル計算機が高速化し、信頼性が高いこと、ソフトウェアの保守が容易であること、非線型演算が容易であること、価格が低下したことからデジタル計算機が使用されるようになった。

デジタル計算機は、動作速度が一定の演算回路を時分割して使用するため、単位時間内の処理量が決まっている。

他方、使用する立場からは、多くの運動方程式を解く要求がある。

そこで、運動方程式の応答特性ごとに、速く繰返す必要のあるものと、比較的遅くてもよいものを区別し、後者は前者の空時間で演算し、処理量を多くする。⁹⁾

例えば、航空機の操舵反力の演算は、10msに一度の繰返しで解くのにに対し、航空機の運動のそれは40～50msで繰返す。

図1で10ms Iterationは前者で、40ms Iterationは後者の例を示す。

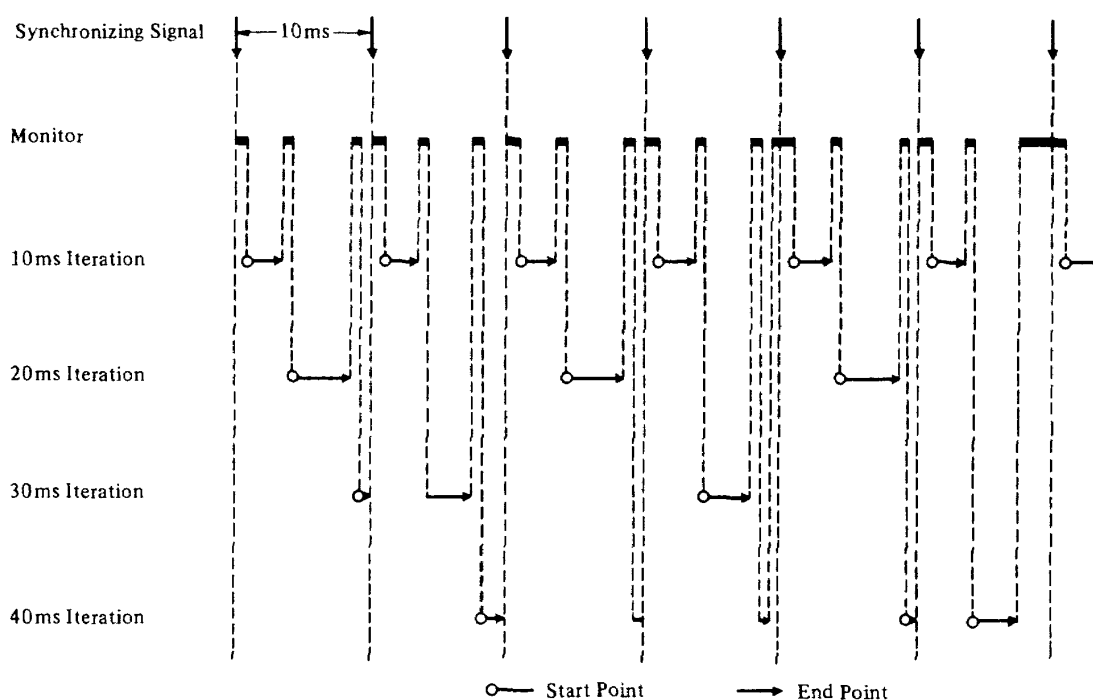


図1 計算機内のタスク スケジュール

これらの多くは、連立常微分方程式の解法である。

実時間との同期を取るためには、計算機にリアル・タイム・クロック (RTC ; Real Time Clock) を実装し、一定間隔で計算機に割込信号を送り、その信号に同期させて、運動方程式を模擬する。

実時間シミュレーション用計算機システムを構成する場合、次の事項を満たすことが必要である。

- (1) シミュレーションに必要な応答周波数を満たすこと。

応答周波数は、外部機器からの信号を入力した時から、計算機内の力学系を模擬した結果を出力するまでの時間によって決まる。

応答周波数が高い程、力学系が複雑な程単位時間内の処理量が増大し、高速な計算機を必要とする。

- (2) 演算精度・再現性が秀れていること。

演算精度は、演算レジスタのビット巾・積分公式・演算順序に関係する。

ビット巾は計算機により異なる。

許容誤差を 0.1 % とすると最底 10 ビットを必要とする。しかし、積分の場合、誤差が累積するため 30 ビット程度必要である。

積分公式は通常、修正オイラー法 (Modified Euler Method) を用い、一周期を 20 等分間隔 (応答周波数が 5 Hz の場合、10 ms 間隔) で演算する。

- (3) 外部機器が容易に接続できること。

人間の操作量・外部機器からのアナログ信号を計算機に取込む時 A/D 変換器を通し、演算結果をアナログ信号で出力する時には D/A 変換器を通す。

この他に、ON/OFF 信号を入力する必要がある。

この装置を I/O リンケージ (図 2 参照) と言い、高速データ転送を要するため、DMA チャンネル

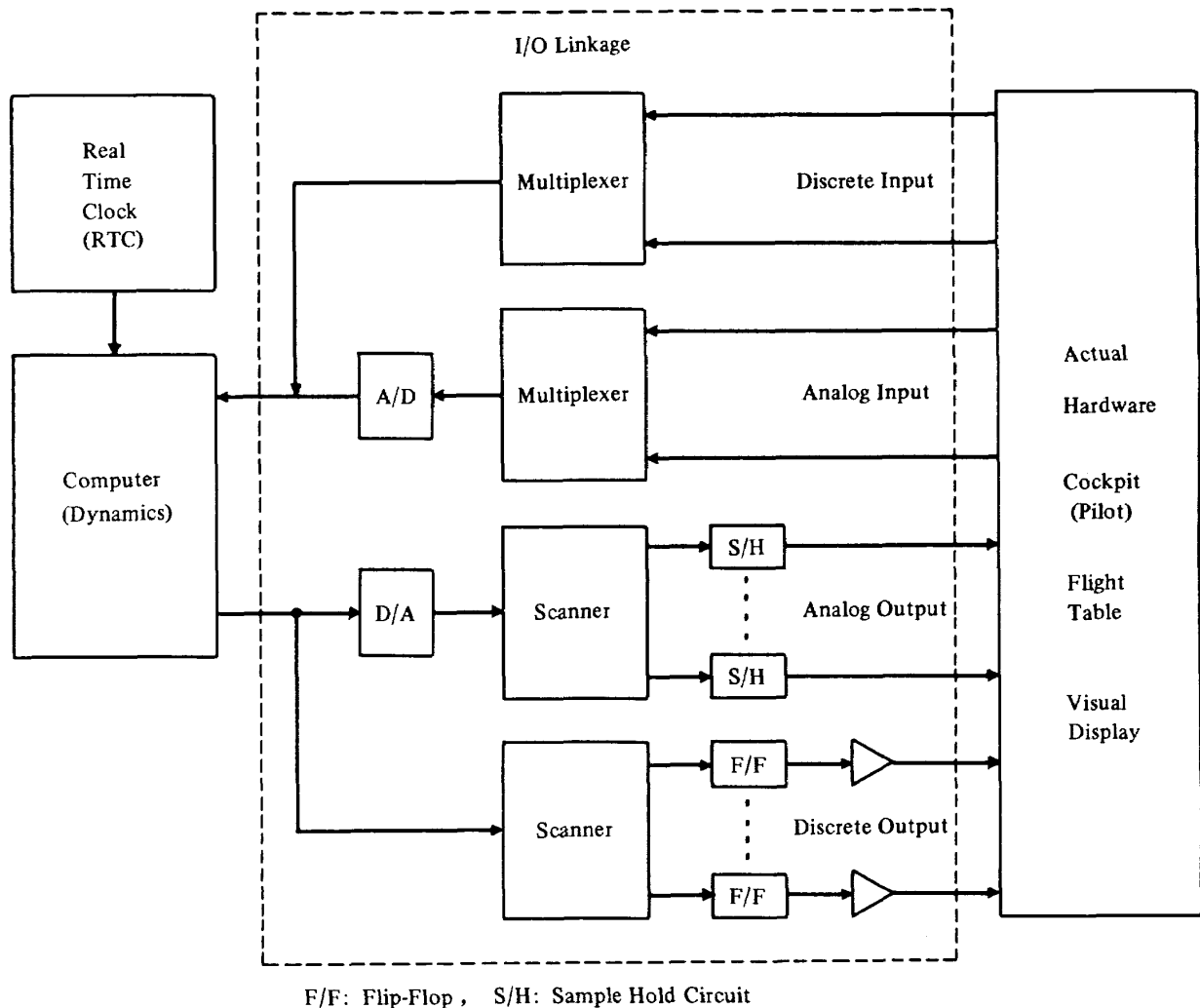


図 2 I/O リンケージの構成

(Direct Memory Access Channel)に接続する。

(4) プログラムの作成・修正・保存が容易であること。

汎用シミュレータの場合、応用プログラムの作成法は、その目的を達成する手段として重要である。

試行錯誤的にシミュレーションを行う場合、そのプログラムを手軽に修正できることが必要で、ソース・プログラムをカードで入力するだけでなく、インタラクティブに修正できることが望ましい。

デジタル計算機では、プログラムの保存・修正を、ファイル・メインテナンス等で容易に行え、大規模なシミュレーション・プログラムを作り易い。

(5) システムの構成要素に信頼性の高い部品を用いること。

安全性がまし、保守維持が容易になる。

また、部品として市販の計算機を用いると低価格でサービスプログラムが充実している等、構成上有効な材料をそのまま使用できる。

4. ハードウェア構成

当所に設置した計算機の構成を図3に示す。

ミニコン5台とそれらを結合するICCL(Inter Computer Communication Logic), 入出力機器を任意の計算機と結合するデバイス・セクタ(Device Selector), 計算機全体を運転監視するオペレータ・コンソール(Operator Console), 外部機器と接続するためのI/O リンケージより成る。

本装置に設置したI/O リンケージのチャンネル数を表1に示す。

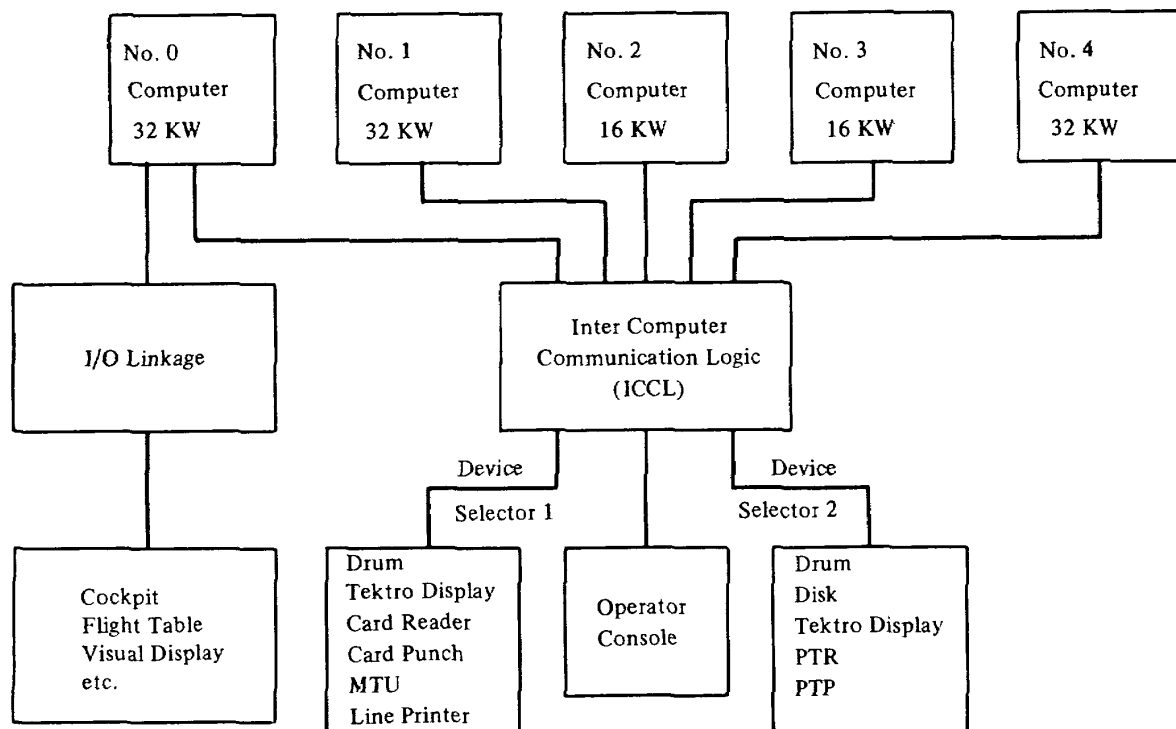


図3 計算機システムの構成

表1 I/O リンケージのチャンネル数

Mode	Channel	Comment
Analog Input (AI)	176	20 μ sec/Data ± 10 Volt
Analog Output (AO)	244	25 μ sec/Data ± 10 Volt
Discrete Input (DI)	304	0.88 μ sec/Data (16 Channel)
Discrete Output (DO)	360	1.2 μ sec/Data (16 Channel)

ICCLは、各計算機のDMA チャネルを互いに結合し、ブロードキャスト(Broadcast)・ブロック(Block)・バースト(Burst)転送を行うHSBC(High Speed Bus Coupler)とPIO(Programmed Input/Output)チャネルを互いに結合し、割込機構を使用して16ビットごとにブロードキャスト転送を行うLSBC(Low Speed Bus Coupler)から構成され、最大8台の計算機と接続できる。

4.1 計算機の結合方式について

複合計算機を構成する各計算機間の結合回路について検討する。

1対1のデータ転送方法の場合、各計算機で生成したデータを全計算機に転送する場合を考える。

計算機の台数をNとすると、

$$N \times (N - 1) \text{ 回}$$

の転送が必要となる。

転送語数をn語、1語転送するのに必要な時間をt、転送に伴う割込処理などのモニタのオーバーヘッドをhとすると、全転送時間Tは、

$$T = (n \times t + h) \times N \times (N - 1)$$

となる。

ここで、計算機間で転送するデータ数は全体で数十語であるとの判断から、各計算機ごとに転送するデータは10語程度と考えられ、

$$n = 10 \text{ 語}$$

とする。

また、1語を転送するに要する時間は、結合回路を使用する計算機によって、その最小値が決まってしまう。MELCOM-70の場合、最小

$$t = 1.21 \mu\text{s}/\text{語}$$

である。

モニタのオーバーヘッドは割込処理によるもので、最適化したとしても200 μs が限度と思われる

$$h = 200 \mu\text{s}$$

とする。

ここで、計算機数は5台であるから

$$N = 5$$

であり、

$$T = 4242 \mu\text{s} \div 4.2 \text{ ms}$$

となる。

他方、5Hzの応答周波数を満たすため、1サイクルを20等分して解く場合、その演算周期は10msとなり、約4割を計算機間のデータ転送に費すことになり、性能上問題が起きる。

そのため、次の対策を取った。

- (1) 転送回数を減ずるために、データ転送回路を1対多のBroadcast転送ができるように改造し、転送回数を計算機数に比例させる。
- (2) モニタのオーバーヘッドを軽減させるために、転送制御情報を記憶装置内に作っておき、これを自動的に読出し、すべての転送を行わせる。

4.2 HSBC

計算機間のデータ転送を高速かつ多量に行うため、この回路が使用される。

HSBCは、計算機2台に1組8段のバッファ(Buffer)を設け、非同期で動く計算機の記憶装置の間で最大826 kW/秒のバースト転送を行うことができる。

転送レートとバッファ段数の関係をFACOM-230-75で模擬した結果を表2に示し、そのプログラムと出力結果を付録1に示す。

表2 バッファの段数と転送レート

Buffer 段数	3	4	5	6	7
転送レート (kW/sec)	472	688	826	826	826

本計算機システムでは、5段バッファで充分であるが、8段バッファを作り易い集積回路¹⁰⁾を使用し、8段とした。

図4で1台の計算機から読出されたデータは、ディストリビュータ(Distributor)を経由し、インプット・ポインタ(Input Pointer)で指示されたバッファに格納される。

アウトプット・ポインタ(Output Pointer)はインプット・ポインタを追い、バッファのデータを1台以上の計算機に同時に書込む。

これらのポインタは循環型になっており、バッファは、繰返し使用される。

ハードウェアでは、すべての計算機が、このHSBCを制御できるよう、構成してある。

しかし、プログラムが暴走して、この回路を乱用しないよう、オペレータ・コンソールで、制御する計算機(これをControl Computerと言う)を指定することができる。

また、システム全体を制御するソフトウェアをNo.0計算機に組込み、HSBCを制御するよう階層を構成して使用するので、以後No.0計算機がHSBCを制御するものとして説明する。

No.0計算機は、記憶装置内にHSBCを制御し、12語で一命令を構成するCCW(Channel Control Word)を有する。(図5参照)

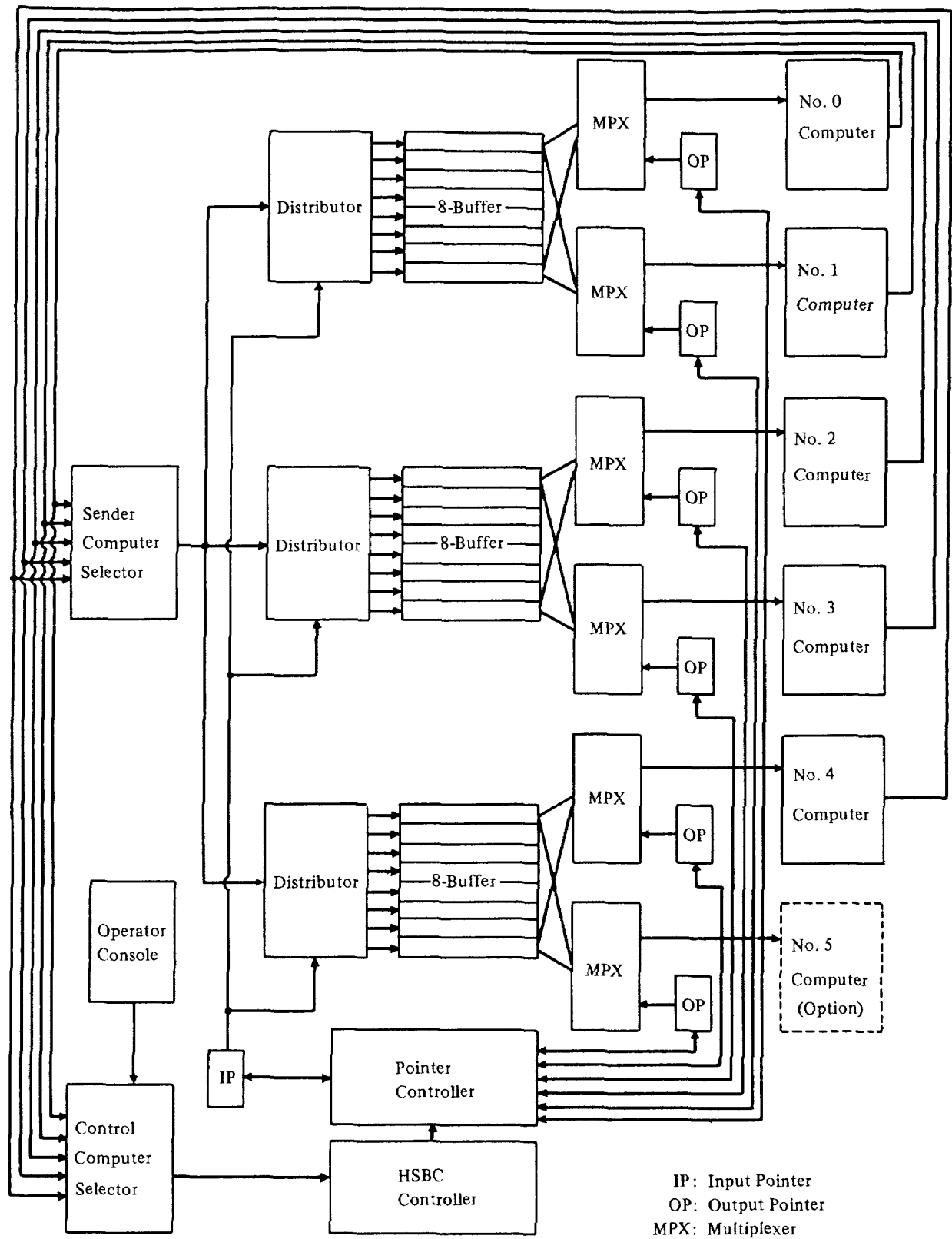


図4 HSBCの構成

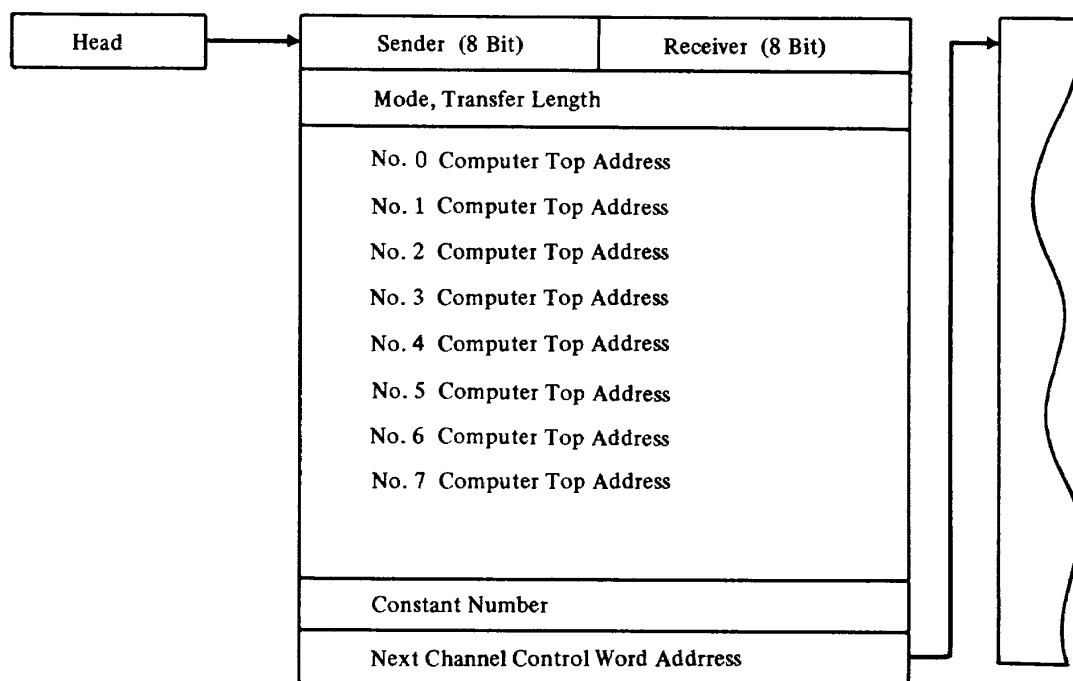


図5 CCWの構成

HSBC制御部はNo. 0 計算機からの起動信号でこのCCWをDMA チャネルを通して読出し、解読して計算機間のデータ転送を行う。

転送が終了するとNo. 0 計算機に割込信号が送られる。

誤動作が検知された時には、即時転送を中断し、エラー情報をNo. 0 計算機の特設アドレスに書き込み、割込信号を送る。

HSBCで検出する誤動作は、以下のものがある。

- (1) Rate Error : 1 語毎の転送間隔が、或る値をこえた時。
- (2) Memory Parity : 計算機が、データ送出時に、パリティ・エラーを生じた時。
- (3) ICCL Parity 1 : データを送出する計算機とHSBCの間でパリティ・エラーが発生した時。
- (4) ICCL Parity 2 : HSBC と、データを入力する計算機間でパリティ・エラーが生じた時。
- (5) DMA Inhibit : CCWで、インヒビットフラグが立っている計算機を入力計算機として指定した時。
- (6) Illegal End : 転送終了時に、送受の語数が一致しない時。
- (7) Illegal Command : CCW が不正な指定をした時。
- (8) Computer Combination : オペレータ・コンソールで、指定されたグループの外に転送指令があった時。

- (9) Illegal Control : オペレータ・コンソールで Control Computerとして許可されていない計算機がHSBCの制御権を要求した時。

4.3 LSBC

割込機構で、計算機間の同期・指令送受を行うために、この回路が使用される。

LSBCは、各計算機のPIOチャネルに16ビットのバッファと8×8ビットの送り先を記憶する回路からなる。信号を送ろうとする計算機は、相手を1台以上指定した後、データを出力する。

指定された計算機に割込信号が送られ、その全計算機がデータを読込むと、信号を送った計算機に転送完了の割込信号が帰ってくる。

5. ソフトウェア構成

システム全体は、実時間シミュレーションを制御する部分(システム・コントロール・プログラム(SCP)、リアル・タイム・システム・モニタ(RTSM))と応用プログラム作製に必要な言語プロセッサ、リンケージ・エディタ・プログラム(LEP)等から構成される。(図6参照)

5.1 SCP

これは、システム全体の管理を行うことを目的として、No. 0 計算機の前半16kWに常駐し、次の機能を有する。

- (1) 実時間シミュレーション時に必要な処理として、

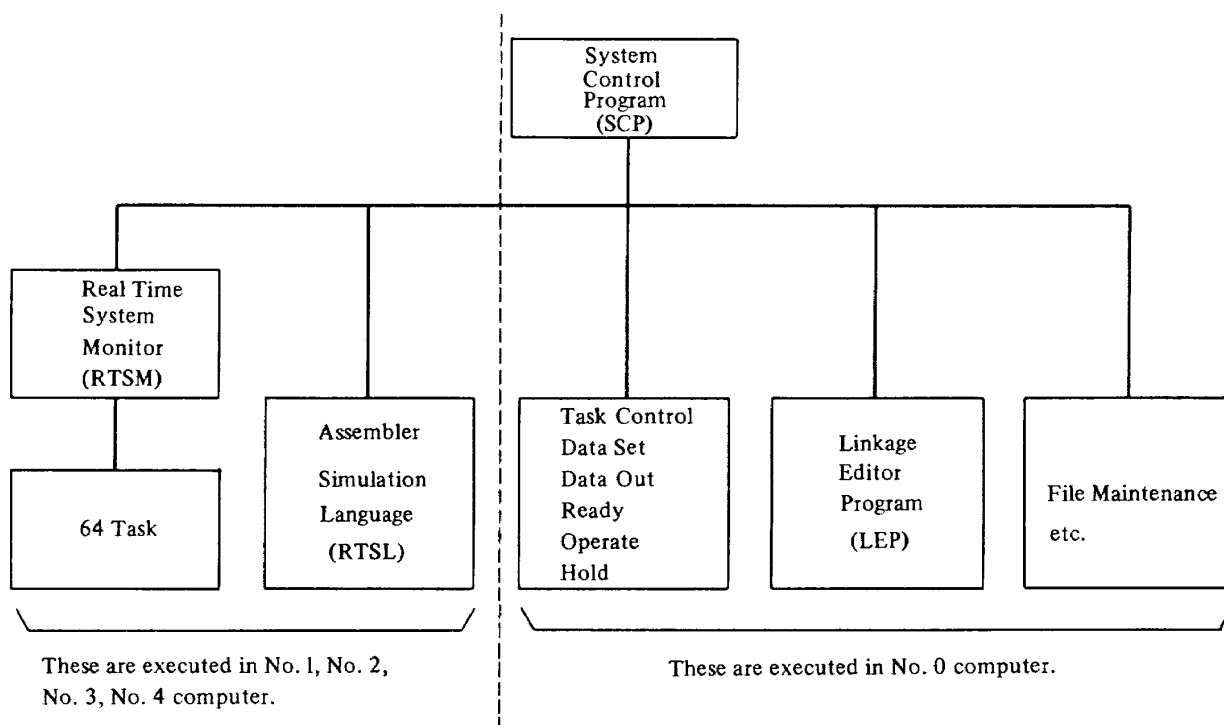


図6 ソフトウェアの構成

計算機間の同期・データ転送を制御し、他計算機と会話をし、I/O リンケージへ出力するデータを編集かつその制御を行う。

- (2) No. 0 計算機では、(1)の空時間を利用し、シミュレーションの管理等を行うため、多くのタスクが組込まれている。

このタスクで発生する内部割込み(マクロ命令)・入出力機器からの外部割込みを処理し、タスクの実行を制御する。

- (3) ジョブ制御指令にしたがい、言語プロセッサ、ファイル・メインテナンス等のジョブを生成消滅させる。

5.2 RTSM

これは、No. 0 計算機以外の計算機内に組込まれ、実時間シミュレーションの応用プログラムを一定時間々隔(10ms の倍数で指定できる。)で繰返す制御を行う。

SCP と RTSM は、図 7 の様な階層を構成し、表 3, 4 に示す会話を LSBC を通して行う。

5.3 LEP

これは、応用プログラムを作成する時、最も重要な役割を演ずるシステム・プログラムであるので、次章に詳述する。

6. LEP

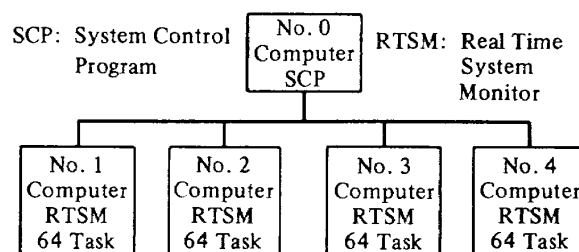


図7 ソフトウェアの階層

物理的に分散した要素よりなる対象のシミュレーション・プログラムは、各要素ごとのサブプログラムに分割され、さらにそれらは、必要に応じ、細分化される。

(図 8 参照)

その単位がタスクである。

飛行シミュレーションの場合、航空機の運動力学を約 30 個、エンジンのそれを約 10 個のタスクに分割作製する。

6.1 プログラム作成時の制限

プログラムをアセンブラ言語で記述する時、次の約束がある。

- (1) タスクへの入力変数は、他タスクで定義されていると考え、External と宣言する。(図 9 参照)
例) : EXT 入力変数 1, ……
- (2) タスクの出力に相当する変数を Entry と宣言し、かつ Pool 領域に定義し、そのワークスペースを

表 3 Supervisor からRTSMへのコマンド

機 能	コ ー ド	備 考
RTSM Ready	0 0 0 0	RTSMの初期値設定。
RTSM Active	0 0 0 1	RTSMのTask Schedule を実行する。
RTSM Pause	0 0 1 0	RTSMのTask Schedule を停止する。
Task Control	0 0 1 1	指定したTaskのMode を変更する。
Task Register	0 1 0 0	Task とその開始番地を指定してTask を登録する。
Data Request	0 1 0 1	Addressを指定して1語のデータを要求する。
Data Transmit	0 1 1 0	RTSMのData Requestに回答してデータを送る。
RTC Set	0 1 1 1	Supervisor から送られた設定値をリアルタイムクロックに設定する。
Iteration Update	1 0 0 0	指定したTask の実行周期を更新する。
Data Store	1 0 0 1	AddressとDataを指定し、格納する。
Schedule Restart	1 0 1 0	停止していたTask Scheduleを再開する。
Response	1 1 1 1	RTSMの要求がSupervisor で終了したことを知らせる。

表 4 RTSMからSupervisor へのコマンド

機 能	コ ー ド	備 考
Task State	0 0 0 0	Task の制御Mode が変更されたことを原因と共に知らせる。
Task Control Block	0 0 0 1	RTSM内のTask Control Blockに変更が生じたことを知らせる。
Data Request	0 0 1 0	Supervisor にComputer, Address を指定して1語のデータを要求する。
Data Transmit	0 0 1 1	Supervisor のData Request に回答してデータを送る。
Task Caution	0 1 0 0	Task 動作時、異常状態が起きたことを知らせる。
RTSM Ready	0 1 0 1	他Computer内RTSMの初期値設定。
RTSM Active	0 1 1 0	他Computer内RTSMのTask Schedule を実行する。
RTSM Pause	0 1 1 1	他Computer内RTSMのTask Schedule を停止する。
Task Control	1 0 0 0	他Computer内TaskのMode を変更する。
Monitor Fail	1 0 0 1	RTSM が正常動作できないことを知らせる。
Schedule Wait	1 0 1 0	Task Schedule が中断したことを知らせる。
Response	1 1 1 1	Supervisor の要求がRTSMで終了したことを知らせる。

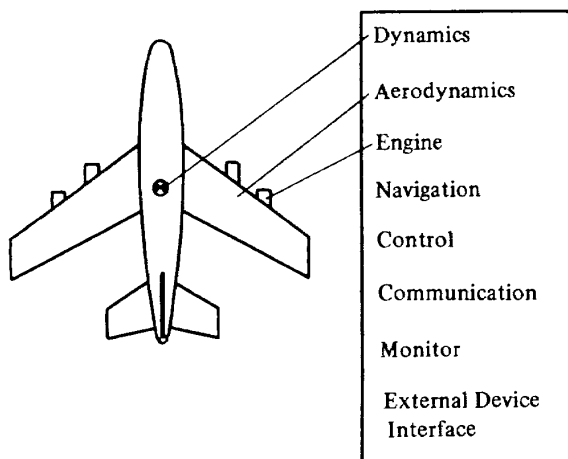


図 8 フライトシミュレーションプログラムの要素

：RES 命令で確保する。

例) ：ENT 出力変数 1, ………

 ：

 ：POOL

出力変数 1；：RES n

 ：

(3) タスクの入出力に相当する変数を間接アドレスで参照する。

(4) 先頭の実行文を、タスクの入口とする。

(5) リロケートブル・オブジェクト・プログラム(Relocatable Object Program)を生成するよう

 ：REL

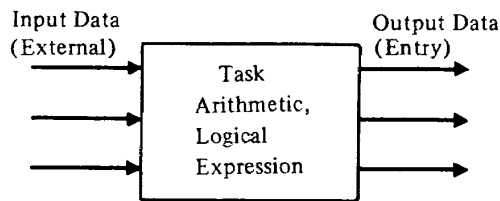


図9 タスクの入出力

命令を用いる。

(6) プログラムの出口には

C A L 0

命令を用いる。

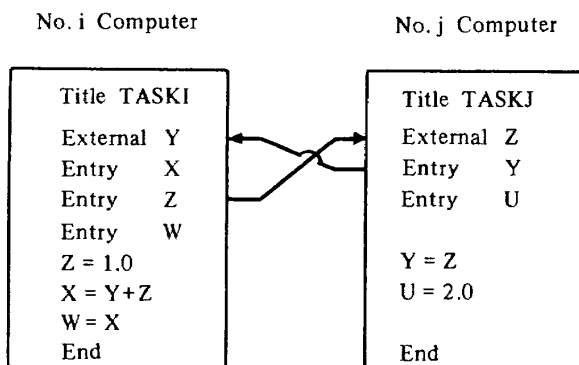
FSPP の例を付録 2 に示す。

6.2 転送データの決定

分割作製されたタスクは、LEPにより結合され、1つのジョブ(Job)となる。

この時、転送するデータが確定する。

図 10 にその原理を示す。



This example represents that Z should be transferred from No. i Computer to No. j Computer and that Y should be transferred from No. j Computer to No. i Computer.

図10 転送データの識別

No. i 計算機のタスク TASKI は、変数 X, Z, W を生成し、No. j 計算機のタスク TASKJ で生成される変数 Y を参照している。

したがって変数 Y を No. j 計算機から No. i 計算機に転送しなければならない。

変数 Z は、その逆である。

ここで変数間の同期について、応用プログラム作製者に任せ、システム側では考慮しない。

6.3 LEPの起動

LEP は、リロケートブル・オブジェクト・プログラム形式のタスクを結合し、ロード・モジュール(Load

```
(1) // EXEC LEP
    // CTRL *
(2) BI, 1.
(3) BO, 1.
(4) LO, 0.
(5) MD, 1.
(6) ET.
(7) JS, Job Name, Library Name (1), Library Name (2), .....
(8) CS, Computer Number, Iteration Time (1), ..., Iteration Time (4).
:
(9) TS, Task Name, Computer Number, Task Number, Task Control Flag.
:
(10) LS, Variable Name, Mode (AI/AO/DI/DO), Channel Number.
:
(11) ES.
(12) PN.
(13) PR.
(14) RD, Job Name.
(15) LD.
(16) EX.
```

図11 LEPのパラメータ

Module), 転送制御情報, 編集情報を生成する。

その手順を次に示す。(図 11 参照)

- (1) EXEC コマンドで SCP が、ドラムより LEP を No. 0 計算機の後半分に読出し、LEP を起動する。
- (2) BI コマンドで、リロケートブル・オブジェクト・プログラムの入力機器を指定する。
- (3) BO コマンドで、結合したロード・モジュールを出力する機器を指定する。
- (4) LO コマンドで、シンボル・リストを出力する機器を指定する。
- (5) MD コマンドで、ロード・モジュールが RTSM の下で動作することを示す。
- (6) ET コマンドで、以後、結合のパラメータであることを示す。
- (7) JS コマンドでロード・モジュールに付けるジョブ名、使用するライブラリ名を指定する。
- (8) CS コマンドで使用する計算機番号・繰返し時間を指定する。コンピュータ・テーブルをコア上に作る。(図 12 参照)
- (9) TS コマンドで計算機番号とタスク名との対応・タスク実行順序の指定・タスク制御フラグを指定する。
タスク・テーブルをコア上に作る。
- (10) LS コマンドで I/O リンケージのチャンネル番号とそこを通して入力する変数の対応を指定する。
リンケージ・テーブルをディスク上に作る。(図 13 参照)
- (11) ES コマンドでタスクのリンク(Link)・エディット(Edit)を開始する。
a) タスク・テーブルに登録されているタスクを外部機器よりディスクに入れる。

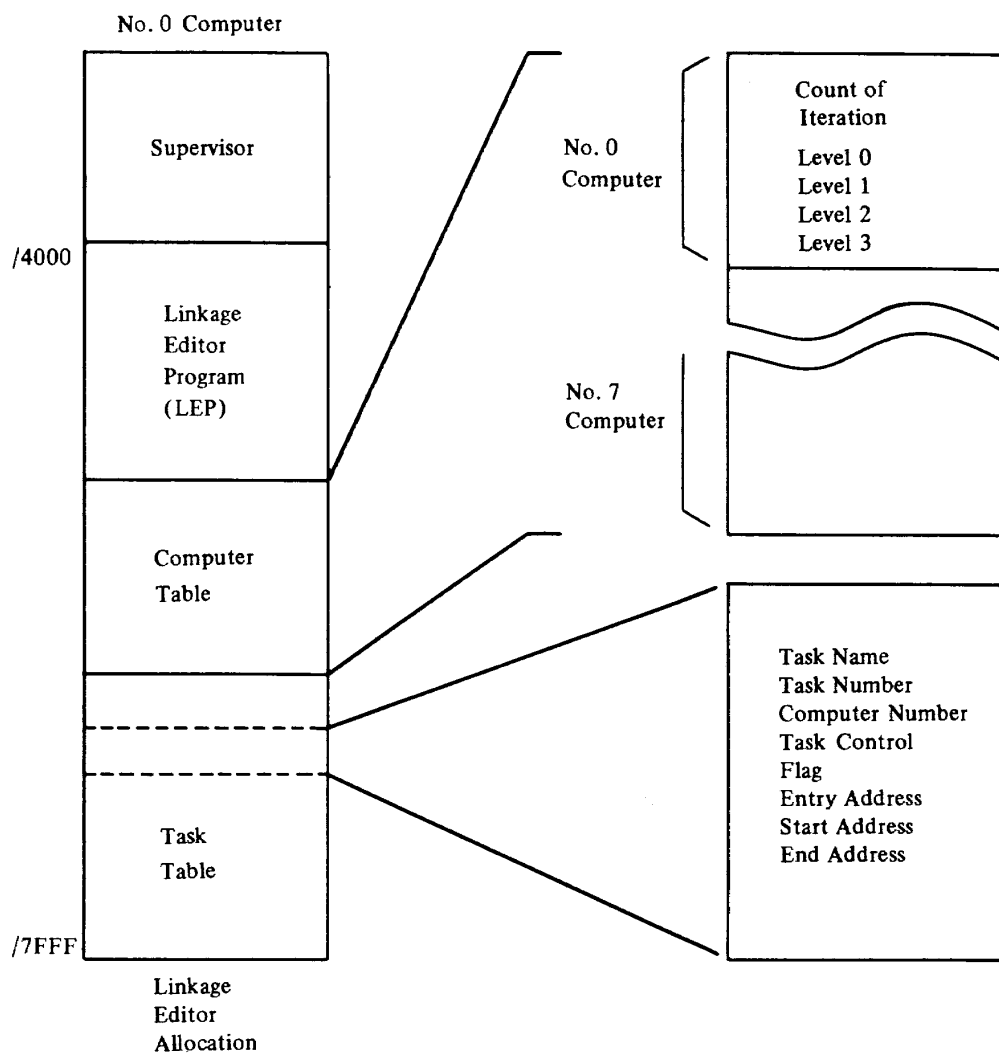


図 12 コア上のテーブル

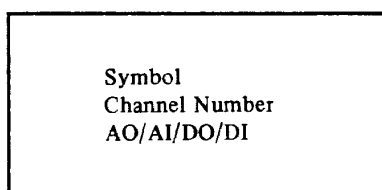


図 13 リンケージ・テーブル

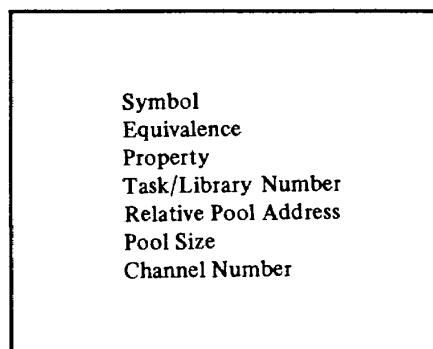


図 14 シンボル・テーブル

- b) J S コマンドに記されているライブラリをディスクに入れる。

ここで各計算機のコア・サイズに見合う領域をディスク上に確保する。

- c) 各計算機ごとに, External, Entry のシンボル・テーブルを作る。

同時に各シンボルの語数(単精度・倍精度等)を記録する。(図 14 参照)

- d) 各計算機の Entry テーブル中, Pool 領域内で

あるシンボル(変数はこの領域内に定義される。)を取出し, 他計算機の External シンボル・テーブルを参照し, 存在すれば, 転送を要する変数と判定する。

これをすべてのシンボルについて行う。

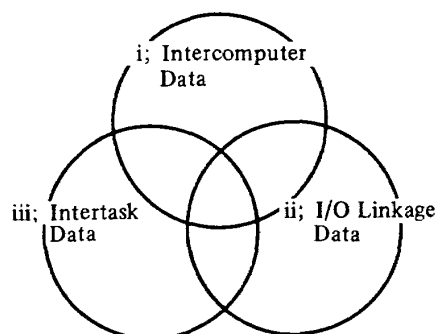


図 15 データの分類

ここで、タスクに入力する変数を下記の3種に
 区別できる。(図15参照)

i) 他計算機内にあるタスクから参照される変数。
 転送を要する。

(Intercomputer Data)

ii) I/O リンケージを経由する変数。
 転送を要する。

(I/O Linkage Data)

iii) 自己計算機内のタスクによって、参照される変
 数。

(Intertask Data)

転送を要すると判定された変数は、その変数を
 参照していない計算機にも、転送される。

その理由は、HSBCがブロードキャスト・ブロ
 ック転送を行うため、各計算機ごとに転送する変
 数を編集するよりも、そのまま転送してしまった

方が早いからである。

e) I/O リンケージを経由して入出力する変数名
 とチャンネル番号の対応表より、入力変数は、各計
 算機にそのまま転送され、出力変数はNo 0 計算機
 に転送される変数と考える。

f) 計算機間のデータ転送は、ブロック転送である
 ため、他計算機に転送するデータを各計算機ごと
 にアドレスが連続になるように配置し、変数のア
 ドレスを決定する。

g) 指定したタスクを結合し、各計算機のコア・イ
 メージを生成する。

この時転送データを参照するための指針を書換
 える。

その方法をアセンブラで示すと図16のようにな
 る。

図中Xは転送される変数とした例が示され、
 : RESはワークスペースを、Sは間接アドレス、
 LDはメモリの内容をレジスタに入れる命令およ
 びXsysは、LEPで、分離された転送データであ
 ることを示す。

h) 各計算機ごとにデータ転送制御情報を生成する。

これは、I/O リンケージからの入力変数をNo
 0 計算機から他計算機にブロードキャストする
 CCW、計算機間で演算に必要な変数を転送する
 CCW、I/O リンケージへ出力する変数をNo 0 計
 算機へ転送するCCWおよび、使用する計算機を
 示すフラグからなる。

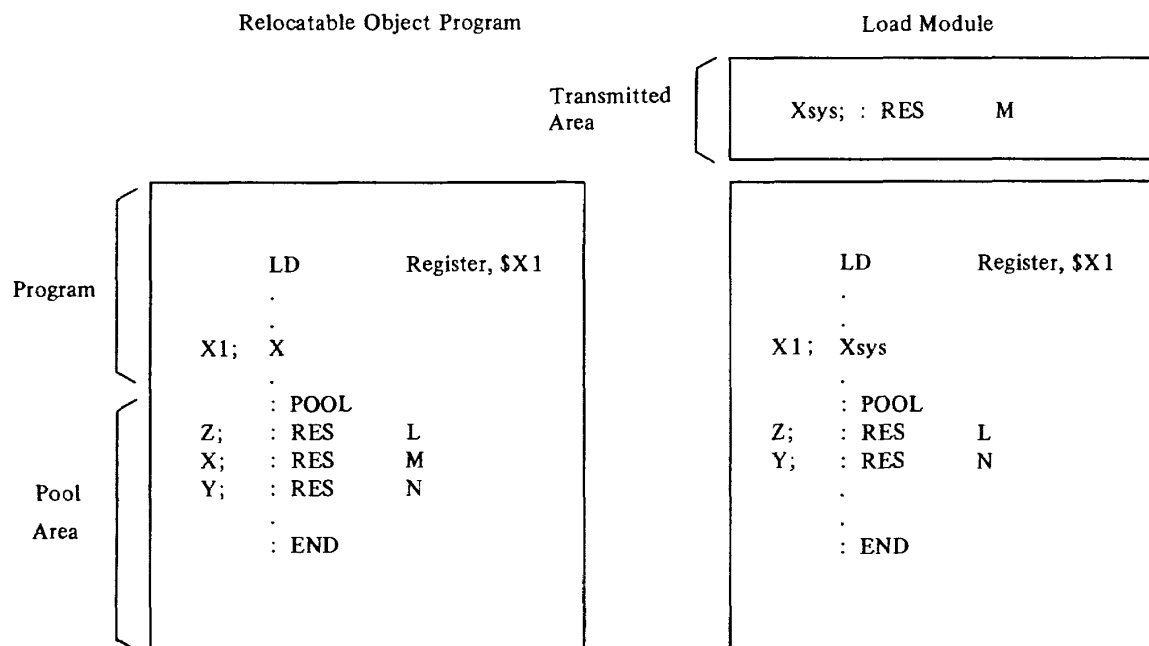


図 16 転送データの分離

- i) 計算機間のデータ転送はブロック転送であり、No.0 計算機に送られてきた変数は、I/O リンケージの出力チャネル順に並んでいない。

LEP は、出力チャネル順に並び替える編集情報を生成する。

- (12) PN コマンドで、ロード・モジュールを外部機器に出力し、ファイルとして保存することができる。
 (13) PR コマンドで、シンボル・リストを出力する。
 (14) RD コマンドで、外部機器よりロード・モジュールを読み、PN コマンド以前の状態を復元する。
 (15) LD コマンドで、ロード・モジュールを No.0 計算機を経由して、各計算機に組込む。

同じく RTSM をドラムから読出し、組込む。

同時にシンボル・テーブルをドラムに移し、シミュレーション時の変数を監視するためのテーブルとして、SCP の管理下に入れる。

各計算機の様子を図 17 に示す。

- (16) EX コマンドで、LEP の処理を終了する。

付録 3 に FSPP の結合リストを示す。

以後、プログラムが組込まれたことを、オペレータ・コンソールで確認した後、初期値を設定し運転に入る。

7. 実時間シミュレーション時のデータ転送

分割作製された応用プログラムは、LEP で結合かつ各計算機に組込まれ、以後、SCP と RTSM より実時間シミュレーションのための制御を受ける。

それは、計算機間の同期・データ転送を、応用プログラムの演算と矛盾なく調和させるためであり、その基本仕様は、次の通りである。

- i) 正常の確認を得るまで、転送されたデータを演算に使用しない。

したがって、データ転送中は、各計算機とも、演算を中断し、転送完了後、正常であることを確認して演算を再開する。

- ii) I/O リンケージから入力したデータをできるだけ早く、応用プログラムに転送し、その結果もできるだけ早く I/O リンケージに出力し、応答周波数を良くする。

- iii) 基本となる時間は、5Hz をさらに 20 等分した 10ms とする。

実時間シミュレーション時におけるデータ転送のタイム・チャートを図 18 に示す。

それは、次の通りである。

- (1) No.0 計算機の RTC から 10ms ごとに割り込みが掛かる。
 (2) No.0 計算機は LSBC を通して、他計算機に RTC を設定する信号を送る。
 受信した計算機は自身の RTC を設定する。
 (3) No.0 計算機は、I/O リンケージから、ディスクリット入力 (DI) アナログ入力 (IA) のデータを入力する。
 (4) RTC が各計算機に割り込むと、応用プログラムの演算を中断し、No.0 計算機に知らせる。
 No.0 計算機は、この信号を、ポーリングにより検知し、割り込みによるオーバーヘッドの増大を避けている。
 (5) No.0 計算機は、LEP により生成され、ロード時、SCP に渡されている転送制御情報を用いて HSBC

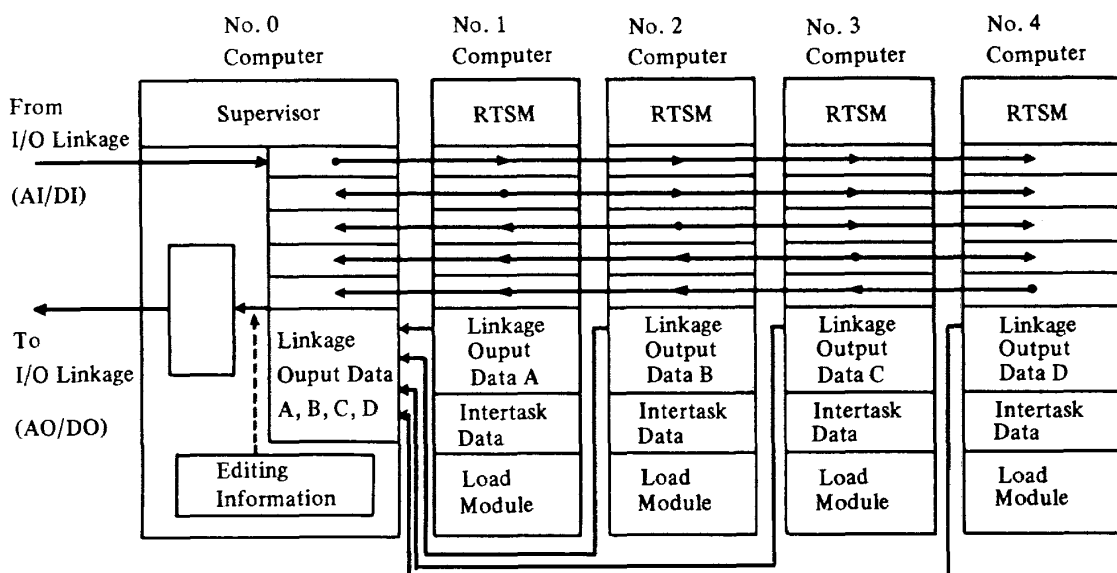


図 17 データの割振り

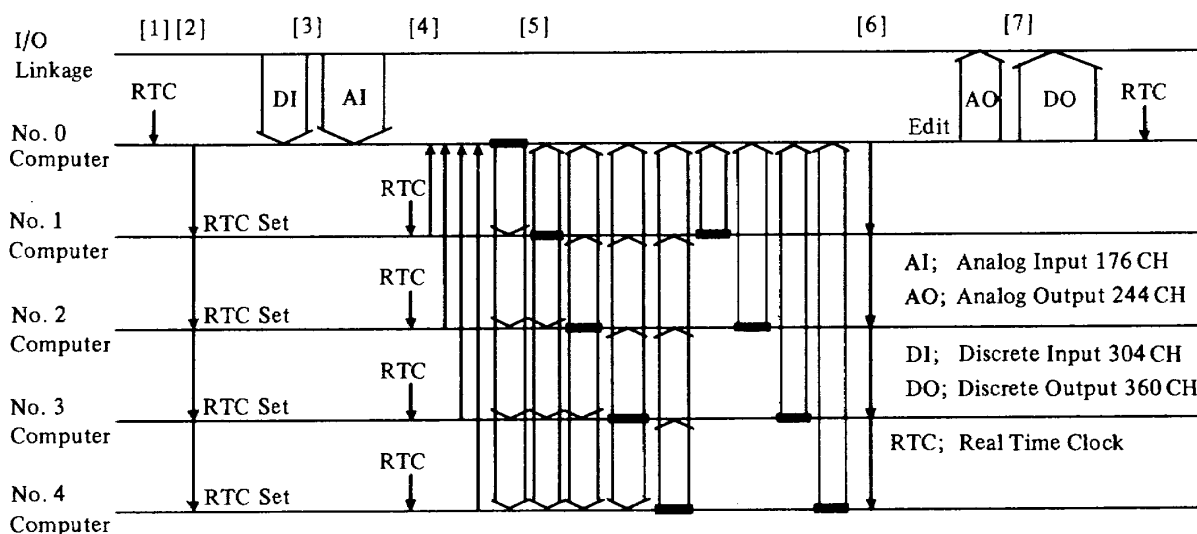


図 18 計算機間の同期

表 5 データ転送と Runge - Kutta 法の関係

	No. 1 Computer	---	No. 2 Computer	---	No. n Computer	
1	x_1^0 Broadcast	---	x_1^0 Broadcast	---	x_n^0 Broadcast	10ms
2	$k_1^1 = dx_0 \times f_1(x_0, x_1^0, \dots, x_n^0)$ $x_1^1 = x_1^0 + \frac{1}{2} k_1^1$	---	$k_1^1 = dx_0 \times f_1(x_0, x_1^0, \dots, x_n^0)$ $x_1^1 = x_1^0 + \frac{1}{2} k_1^1$	---	$k_n^1 = dx_0 \times f_n(x_0, x_1^0, \dots, x_n^0)$ $x_n^1 = x_n^0 + \frac{1}{2} k_n^1$	10ms
3	x_1^1 Broadcast	---	x_1^1 Broadcast	---	x_n^1 Broadcast	10ms
4	$k_1^2 = dx_0 \times f_1(x_0 + \frac{1}{2} dx_0, x_1^1, \dots, x_n^1)$ $x_1^2 = x_1^1 + \frac{1}{2} k_1^2$	---	$k_1^2 = dx_0 \times f_1(x_0 + \frac{1}{2} dx_0, x_1^1, \dots, x_n^1)$ $x_1^2 = x_1^1 + \frac{1}{2} k_1^2$	---	$k_n^2 = dx_0 \times f_n(x_0 + \frac{1}{2} dx_0, x_1^1, \dots, x_n^1)$ $x_n^2 = x_n^1 + \frac{1}{2} k_n^2$	10ms
5	x_1^2 Broadcast	---	x_1^2 Broadcast	---	x_n^2 Broadcast	10ms
6	$k_1^3 = dx_0 \times f_1(x_0 + \frac{1}{2} dx_0, x_1^2, \dots, x_n^2)$ $x_1^3 = x_1^2 + k_1^3$	---	$k_1^3 = dx_0 \times f_1(x_0 + \frac{1}{2} dx_0, x_1^2, \dots, x_n^2)$ $x_1^3 = x_1^2 + k_1^3$	---	$k_n^3 = dx_0 \times f_n(x_0 + \frac{1}{2} dx_0, x_1^2, \dots, x_n^2)$ $x_n^3 = x_n^2 + k_n^3$	10ms
7	x_1^3 Broadcast	---	x_1^3 Broadcast	---	x_n^3 Broadcast	10ms
8	$k_1^4 = dx_0 \times f_1(x_0 + dx_0, x_1^3, \dots, x_n^3)$ $x_1^4 = x_1^3 - \frac{1}{6} (k_1^1 + 2k_1^2 + 2k_1^3 + k_1^4)$	---	$k_1^4 = dx_0 \times f_1(x_0 + dx_0, x_1^3, \dots, x_n^3)$ $x_1^4 = x_1^3 - \frac{1}{6} (k_1^1 + 2k_1^2 + 2k_1^3 + k_1^4)$	---	$k_n^4 = dx_0 \times f_n(x_0 + dx_0, x_1^3, \dots, x_n^3)$ $x_n^4 = x_n^3 - \frac{1}{6} (k_n^1 + 2k_n^2 + 2k_n^3 + k_n^4)$	10ms

持たせ、連立常微分方程式、

- $$\begin{aligned} dx_1/dx_0 &= f_1(x_0, x_1, \dots, x_n), \\ &\vdots \\ dx_i/dx_0 &= f_i(x_0, x_1, \dots, x_n), \\ &\vdots \\ dx_n/dx_0 &= f_n(x_0, x_1, \dots, x_n), \end{aligned}$$

一定周期の繰返して全計算機に同じデータ・イメージを

を解く方法を表 5 に示す。

表中, 1, 3, 5, 7 で各計算機に生成した変数を他計算機に同時に送り, 各々の計算機に同一データ・イメージを持たせ, この値を用いて, 2, 4, 6, 8 で演算を行う。

ここで, 各々の演算は, どの計算機で実行されてもよく, プログラム作製時, 計算機間の入出力を指定する必要がない。

8. 転送データの数量

6.2 で述べた様に, 転送データは, LEP で結合した時に確定するものであるから各タスクを各計算機に割振る方法により, その数量は左右される。

FSPF を例として取り上げる。

そのプログラムは,

- (1) 運動力学
- (2) エンジン
- (3) 外部機器インタフェース

のサブプログラムに分割でき,

- (1) と (2) の間で約 20 語
- (1) と (3) の間で約 70 語
- (2) と (3) の間で約 20 語
- (3) と I/O リンケージの間で約 40 語

のデータ送受がある。(図 19 参照)

しかし, 計算機にタスクを割振る時には, このサブプログラムにこだわる必要はなく, 付録 3 の場合の計算機間のデータ送受を図 20 に, そのデータ名を表 6, 7, 8 に示す。

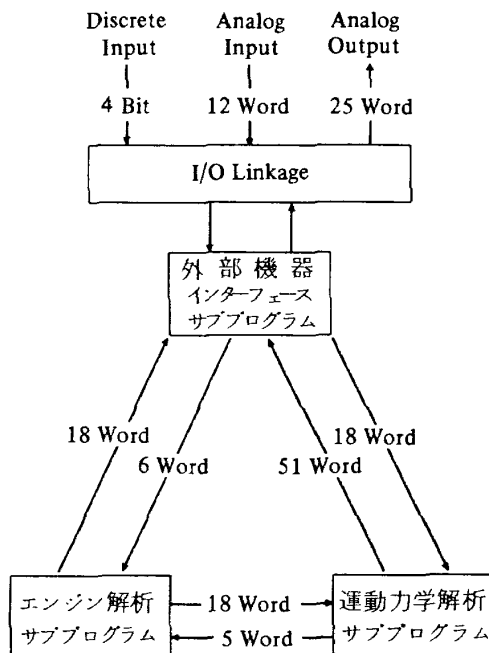


図 19 サブプログラム間のデータ送受 (FSPF の場合)

計算機間で送受するデータは全体で約 60 語, 計算機と I/O リンケージの間で送受するデータは約 40 語である。

9. むすび

本システムは, 既に完成し, 昭和 50 年 6 月に, 大型ジェット機のシミュレーションを, 計算機 3 台と FSPF を用いて実現した。

その結果, 次のことが確かめられた。

- (1) ブロードキャスト (Broadcast) データ転送を行い, データ転送に伴う負荷を, 軽減した。
- (2) 転送データを連続する領域に集めたため, 転送効率を向上させることができた。
- (3) 飛行シミュレーションの場合, サブプログラム間で送受するデータ数は 120 語程度であり, 計算機間で送受するデータ数は 60 語程度である。
- (4) 一定周期の繰返しで全計算機に同一データを持たせたため, 応用プログラム作製時に, タスクの入出力変数を宣言するだけで, 計算機間のデータ送受について配慮する必要がない。

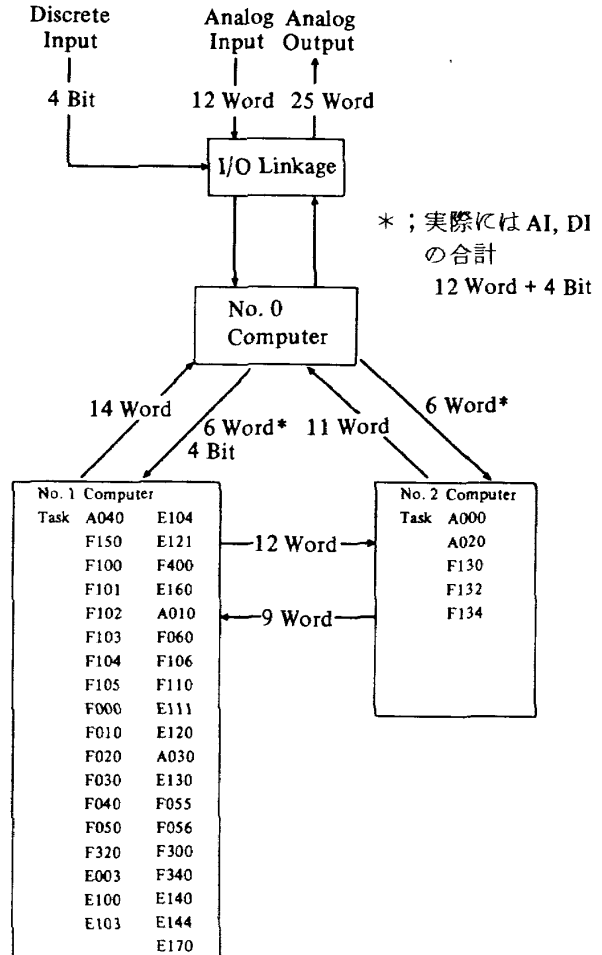


図 20 計算機間のデータ送受 (FSPF の場合)

(5) 複合計算機の応用プログラム作製法に矛盾がない。た。

しかし、データ転送では、 3×10^7 bit の転送に1回の割合でエラーが起り、さらに、分割作製したタスクを結合するのに多大な時間を必要とし(FSPPの場合、42タスク、語数約20 K語、変数約600個を結合するのに約40分を要した。)シミュレーションを試行錯誤で行うことは不可能である等、多くの改良を望まれる点が発見され

終りに、本研究に携わる機会を与えてくださった故樋口元計測部長、協力援助いただいた当所計測部自動制御第一研究室の堀川室長・渡辺技官・名越技官、また製作にあたられた、三菱プレジジョン㈱の林岑男氏、梶原景範氏をはじめとする担当方々の諸氏に感謝する。

表6 No. 0計算機に転送されるデータ

	変数	属性	備考	FSPPでのシンボル
No. 1 Computer から No. 0 Computer に転送されるデータ	hPAO	語	Pressure Altitude	HPAO
	ViAO	"	Indicated Airspeed	VIAO
	RCAO	"	Indicated Rate of Climb	RCAO
	λBAO	"	Indicated Ball Angle	RBAO
	rAAO	"	Indicated Yaw Rate	RAAO
	θAO	"	Indicated Pitch Angle	THAO
	φAO	"	Indicated Roll Angle	PAIAO
	ψAO	"	Indicated Yaw Angle	PSIAO
	EPRRAO	"	EPR Right	EPRAO
	EPRLAO	"	EPR Left	EPLAO
	N ₂ RAO	"	Percent N ₂ Right	N ₂ RAO
	N ₂ LAO	"	Percent N ₂ Left	N ₂ LAO
	EGTRAO	"	EGT Right	EGRAO
	EGTLAO	"	EGT Left	EGLAO
No. 2 Computer から No. 0 Computer に転送されるデータ	θ _M	語	Motion Pitch Angle	THETM
	φ _M	"	Motion Roll Angle	PAIM
	FSAO	"	Stick Force	FSAO
	F _W AO	"	Wheel Force	F _W AO
	F _P AO	"	Pedal Force	F _P AO
	X _{VD2}	"	Visual Display T.V. Drive	XVD2
	Y _{VD2}	"		YVD2
	Z _{VD2}	"		ZVD2
	θ _{VD2}	"		THEVD2
	φ _{VD2}	"		PAIVD2
	ψ _{VD2}	"		PSIVD2

表 7 No. 1 計算機に転送されるデータ

	変 数	属性	備 考	FSPPでの シンボル
No. 0 Computer から No. 1 Computer に転送されるデ ータ	ICCP	ビット	Cockpit Desk IC ON	DIWORD
	STRTCP	"	Cockpit Desk Start ON	
	STOPCP	"	Cockpit Desk Stop ON	
	ZEROC	"	Cockpit Desk Zero	
	δT_1	語	Left Engine Throttle	DT1
	δT_2	"	Right Engine Throttle	DT2
	FLP	"	Flap Position Input	FLP
	SFRLAI	"	Stabilizer Deflection	SFRLAI
	δt_{TAI}	"	Aileron Tab	DTTAAI
	δt_{TRA}	"	Rudder Tab	DTTRA
No. 2 Computer から No. 1 Computer に転送されるデ ータ	$\delta SPIR$	語	Inbord Right Spoiler Deflection	DSPIR
	$\delta SPOR$	"	Outbord Right Spoiler Deflection	DSPOR
	$\delta SPIL$	"	Inbord Left Spoiler Deflection	DSPIL
	$\delta SPOL$	"	Outbord Left Spoiler Deflection	DSPOL
	δe	"	Elevator Deflection	DE
	δet	"	Elevator Tab Deflection	DET
	δr	"	Rudder Deflection	DR
	δai	"	Inbord Aileron Deflection	DAI
	δao	"	Outbord Aileron Deflection	DAO

参 考 文 献

- 1) J. Gregory, R. McReynolds; The SOLOMON Computer, IEEE Trans. on EC, 12, 6, pp. 774 ~ 781 (Dec. 1963).
- 2) J. Holland; A Universal Computer Capable of Executing an Arbitrary Number of Subprograms Simultaneously, Proc. EJCC 16, pp. 108 ~ 113 (Dec. 1959).
- 3) G. H. Barnes, R. M. Brown, M. Kato, D. J. Kuck, D.L. Slotnick, R.A. Stokes; The ILLIAC IV Computer, IEEE Trans. on C, 17, 8, pp. 746 ~ 757 (Aug. 1968).
- 4) D. J. Kuck; ILLIAC IV Software and Application Programming, IEEE Trans. on C, 17, 8, pp. 758 ~ 770 (Aug. 1968).
- 5) R. E. Larson, E. Tse; Model Trajectory Estimation and Parallel Computers, Proc. 2nd Symposium on Nonlinear Estimation Theory, pp. 188 ~ 198 (Sept. 1971).
- 6) E. Tse; Parallel Computation of the Conditional Mean State Estimate for Nonlinear Systems, Proc. 2nd Symposium on Nonlinear Estimation Theory, pp. 385 ~ 394 (Sept. 1971).
- 7) 田丸他：マルチ・ミニコンピュータ・システム，情報処理第14回大会講演予稿集，pp. 349(1973. 12. 10)
- 8) 実用普及期を迎える分散型コンピュータネットワーク，日経エレクトロニクス，pp. 50 - 81 (1975. 2. 24)
- 9) 林，久良知：フライトシミュレータ，計測と制御，第12巻，第10号，pp. 785 - 796 (昭和48年10月)

表 8 No. 2 計算機に転送されるデータ

	変 数	属性	備 考	FSPPでの シンボル
No. 0 Computer から No. 2 Computer に転送されるデ ータ	δSAI	語	Stick Deflection	DSAI
	FSCAI	"	Pilot Applied Stick Force	FSCAI
	δWAI	"	Wheel Deflection	DWAI
	FWCAI	"	Pilot Applied Wheel Force	FWCAI
	δPAI	"	Pedal Deflection	DPAI
	FPCAII	"	Pilot Applied Pedal Force	FPCAII
No. 1 Computer から No. 2 Computer に転送されるデ ータ	q	語	Dynamic Pressure	Q
	δFWN	"	Normalized Delta Flap	DFWN
	$\delta FWIN$	"	Normalized Delta Flap Inbord	DFWIN
	δtta	"	Aileron Tab Deflection	DTTA
	δttr	"	Rudder Tab Deflection	DTTR
	β	"	BETA	BETA
	X	"	Position Erth Axis	X
	Y	"		Y
	Z	"		Z
	θ	"	Pitch Angle	THETA
	ϕ	"	Roll Angle	PAI
	ψ	"	Yaw Angle	PSI

- 10) R. C. Ghest; The Am9338 Multiple Port Memo-
ry, Advanced Micro Devices, pp. 379 ~ 382.

付 録 1

HSBCのバッファ段数とデータ転送 のシミュレーション・プログラム

模擬のモデルとプログラムの説明

使用した計算機 (MEL COM-70) のメモリ・サイクルは、DMA チャネルを経由する時、

読出しサイクル 1210 ns

書込みサイクル 880 ns

である。

模擬に用いたモデルは、1 台の計算機から、メモリ・サイクルが 110 ns ずつずれた 11 台の計算機にデータを 1 対多で転送する場合を想定し、

- (1) 各計算機ごとに、同じ段数のバッファを設け、各々 3 段～8 段の場合について調べる。
- (2) 計算機から読出されたデータは、各バッファの同一の段に設定される。
- (3) データが、バッファに設定されてから 440 ns 後に、データ書込み要求信号を入力側の計算機に送る。
- (4) 各バッファの同一の段が、すべて空になった場合、その 440 ns 後に、データ読出し要求信号を出力側の計算機に送る。

の条件を満たすものとしている。

ここで、440 ns という値は、ゲートの遅れ、信号の伝達遅れ等の合計であり、かなり大きめに設定してある。

読出し／書込み要求信号は、各々の計算機のメモリ・サイクルがずれているため、ただちに受け付けられる場合から、次のメモリ・サイクルまで待たされる場合もある。

その模擬は、55 ns きざみで進められ、データ転送の立上がりによる誤差を避けるため、初めの 200 回分

$$55 \text{ ns} \times 200 = 11 \mu\text{s}$$

のデータを使わずに 200000 回分

$$55 \text{ ns} \times 200000 = 11 \text{ ms}$$

の間に転送したデータ数を計算する。

出力側の計算機では、メモリ・サイクルを 55 ns ごとに 22 のフェイズに分け、次の動作を行う。

- (1) 0～55 ns の間 (INS 92～96)

1 つ前のサイクルで、読出し要求信号が送られて来ているかを PREREQ(1) に記憶し、もし、送られて来ている (PREREQ(1)=1) なら、読出しサイクルとなるため、メモリ・サイクルを 1210 ns と設定する。

- (2) 55～110 ns (INS 97～109)

各バッファの同一段すべてが空になってから、440 ns 以上を経たものがある時には読出し要求信号を出す (REQ(1)=1)。

- (3) 660～715 ns の間 (INS 110～119)

1 つ前のサイクルで、読出し要求信号が送られて来ているなら (PREREQ(1)=1)、転送するデー

タ (DATA1) に番号をつけ、データをバッファに設定する。

その後、バッファの空を調べ、空がなければ読出し要求信号を出さないように (INHIB) 設定する。

入力側の計算機では、次の動作を行う。

- (1) 0～55 ns の間 (INS 127～134)

1 つ前のサイクルで書込み要求信号が送られて来ているかを PREREQ(2)～PREREQ(12) に記憶する。

もし、書込み要求信号があれば、書込みサイクルになるため、メモリ・サイクルを 880 ns に設定する。

書込み要求信号が無ければ、1210 ns とする。

これは、入力側のメモリ・サイクルが長い程、バッファ内のデータが待ちに入る時間が長くなり、バッファ段数が多く必要となるであろうとの予想によるものである。

- (2) 55～110 ns の間 (INS 124～126) バッファにデータが設定されてから 440 ns 以上経ていれば、書込み要求信号を送る。

- (3) 440～495 ns の間 (INS 135～142)

1 つ前のサイクルで、書込み要求信号が送られて来ていれば (PREREQ(i)=1 ただし i=2～12) バッファ内のデータを送り出し、バッファに余裕ができれば INHIB を払う。

両者共に、それ以外の時には、一度計算すると共に、フェイズを進め、メモリ・サイクルとフェイズが一致すれば、フェイズを 0 にもどす。

ISN	ST-NO	SOURCE PROGRAM
1	C	TRANSMISSION SIMULATION
2		INTEGER REQ(12), PREREQ(12), WRTIND(12), REDIND(12)
3		INTEGER PHASE(12), CYCTIM(12), BUFDAT(8,12), BUFTIM(8,12), COUNT
4		INTEGER DATA1
5		INTEGER INHIB(12), BUF
6		INTEGER IX, IX1, IX0, IJ
7		BUF = 3
8	1	DO 2 I = 1, 12, 1
9		PREREQ(I) = 0
10		REQ(I) = 0
11		WRTIND(I) = 1
12		REDIND(I) = 1
13		INHIB(I) = 0
14	2	CONTINUE
15		DO 3 I = 1, 8, 1
16		DO 4 J = 1, 12, 1
17		BUFDAT(I, J) = 0
18		BUFTIM(I, J) = -440
19	4	CONTINUE
20	3	CONTINUE
21		IX0 = 0
22		IX1 = 1
23		PHASE(1) = 0
24		PHASE(2) = 0
25		PHASE(3) = 110
26		PHASE(4) = 220
27		PHASE(5) = 330
28		PHASE(6) = 440
29		PHASE(7) = 550
30		PHASE(8) = 660
31		PHASE(9) = 770
32		PHASE(10) = 880
33		PHASE(11) = 990
34		PHASE(12) = 1100
35		CYCTIM(1) = 1210
36		CYCTIM(2) = 1210
37		CYCTIM(3) = 1210
38		CYCTIM(4) = 1210
39		CYCTIM(5) = 1210
40		CYCTIM(6) = 1210
41		CYCTIM(7) = 1210
42		CYCTIM(8) = 1210
43		CYCTIM(9) = 1210
44		CYCTIM(10) = 1210
45		CYCTIM(11) = 1210
46		CYCTIM(12) = 1210
		COUNT = 0

ISN	ST-NO	SOURCE PROGRAM	(FTMAIN)
47		DATA1 = 0	
48	C SENDER COMPUTER		
49	10	I = 1	
50		IFC PHASE(I) .EQ. 0) GO TO 100	
51		IFC PHASE(I) .EQ. 55) GO TO 200	
52		IFC PHASE(I) .EQ. 660) GO TO 300	
53	C RECEIVER COMPUTER		
54	400	I = 2	
55	C	SPACE	
56	500	IFC PHASE(I) .EQ. 0) GO TO 600	
57		IFC PHASE(I) .EQ. 55) GO TO 700	
58		IFC PHASE(I) .EQ. 440) GO TO 800	
59	410	I = I + 1	
60		IFC I .NE. 13) GO TO 500	
61		I = 1	
62	455	PHASE(I) = PHASE(I) + 55	
63		IFC PHASE(I) .EQ. 880) GO TO 420	
64		IFC PHASE(I) .EQ. 1210) PHASE(I) = 0	
65	430	J = 1	
66	435	IFC BUFTIM(J, I) .GE. 0) GO TO 470	
67		BUFTIM(J, I) = BUFTIM(J, I) - 55	
68	490	J = J + 1	
69		IFC J .NE. 9) GO TO 435	
70	450	I = I + 1	
71		IFC I .NE. 13) GO TO 455	
72		IFC COUNT .EQ. 200) II = DATA1	
73		COUNT = COUNT + 1	
74		IFC COUNT .LE. 200200) GO TO 10	
75		WRITE (6, 1150)	
76		DO 950 J = 1, 12, 1	
77		WRITE (6, 1000) (BUFDAT(I, J), I = 1, 8)	
78	950	CONTINUE	
79		IJ = (DATA1 - II) * 1000 / 11	
80		WRITE (6, 1050) BUF	
81		WRITE (6, 1050) II	
82		WRITE (6, 1050) DATA1	
83		WRITE (6, 1050) IJ	
84		WRITE (6, 1100) (REDIND(I), I = 1, 6)	
85		WRITE (6, 1100) (REDIND(I), I = 7, 12)	
86		WRITE (6, 1100) (WRTIND(I), I = 1, 6)	
87		WRITE (6, 1100) (WRTIND(I), I = 7, 12)	
88		BUF = BUF + 1	
89		IFC BUF .NE. 9) GO TO 1	
90		STOP, TRANSMISSION SIMULATION,	
91	1000	FORMAT (2X, 8(I1, 2X))	
	1050	FORMAT (2X, I11)	
	1100	FORMAT (2X, 6(I11, 2X))	
	1150	FORMAT (1H)	

ISN	ST-NO	SOURCE PROGRAM	(PTMAIN)
92	C 100	SPACE	
93		PREREQ(I) = REQ(I)	
94		REQ(I) = 0	
95	150	IF(PREREQ(I) .EQ. 1) GO TO 150	
96		CYCTIM(I) = 1210	
		GO TO 400	
97	C 200	SPACE	
98		K = 0	
99		DO 250 J = 2, 12, 1	
100		IF(BUFTIM(WRTIND(J) , J) .GT. -440) GO TO 250	
101		K = K + 1	
102	250	CONTINUE	
103		IF(K .EQ. 11) GO TO 260	
104		GO TO 400	
105	260	IN = 0	
106		DO 270 J = 2, 12, 1	
107		IN = INHIB(J) + IN	
108	270	CONTINUE	
109		IF(IN .EQ. 0) REQ(I) = 1	
		GO TO 400	
110	C 300	SPACE	
111		IF(PREREQ(I) .NE. 1) GO TO 400	
112		DATA1 = DATA1 + 1	
113		DO 350 J = 2, 12, 1	
114		BUFDAT(WRTIND(J) , J) = DATA1	
115		BUFTIM(WRTIND(J) , J) = 1	
116		WRTIND(J) = MOD(WRTIND(J) , BUF) + 1	
117		N = MOD(WRTIND(J) , BUF) + 1	
118	350	IF(N .EQ. REDIND(J)) INHIB(J) = 1	
119		CONTINUE	
		GO TO 400	
120	C 420	SPACE	
121		IF(CYCTIM(I) .EQ. 880) PHASE(I) = 0	
122		GO TO 430	
123	470	BUFTIM(J , I) = BUFTIM(J , I) + 55	
		GO TO 490	
124	C 700	SPACE	
125		IF(BUFTIM(REDIND(I) , I) .LT. 440) GO TO 410	
126		REQ(I) = 1	
		GO TO 410	
127	C 600	SPACE	
128		PREREQ(I) = REQ(I)	
129		REQ(I) = 0	
130	670	IF(PREREQ(I) .EQ. 1) GO TO 640	
131		CYCTIM(I) = 1210	
132	640	GO TO 410	
133	850	IX = 880	
		CYCTIM(I) = IX	

```
134      C
135      GO TO 410
136      SPACE
137      IF( PREHE( I ) .NE. 1 ) GO TO 410
138      BUFDAT( REDIND( I ), I ) = 0
139      BUFTIM( REDIND( I ), I ) = -1
140      REDIND( I ) = MOD( REDIND( I ), BUF ) + 1
141      N = MOD( WRTIND( I ), BUF ) + 1
142      N = MOD( N , BUF )
143      IF ( N .EQ. REDIND( I ) ) INHIB( I ) = 0
144      GO TO 410
145      SPACE
146      END
```

[illegible]

0 0 0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0	4 4 5 5	0 0 0 0 0 0 0 0 0 0 0 0 0 0	3 3 4 4	0 9099 0 9099 9099 9099 9099 9099 9099 9099 9099
0 0 0 0	3 4 5 5	0 0 0 0 0 0 0 0 0 0 0 0 0 0	2 3 4 4	0 0 0 0 9099 9099 9099 9099 9099 9099 9099 9099 9099
9099 9099 9099 9099	3 4 5 5	0 0 0 0 0 0 0 0 0 0 0 0 0 0	2 3 4 4	0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0	5 4 5 5	0 9099 0 9099 9099 9099 9099 9099 9099 9099 9099	4 3 4 4	0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0	4 4 5 5	0 0 0 9098 9098 0 0 0 0 0 0 0 0 0 0 0 0	3 3 4 4	0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 5 9 9099 826454 1 4 1 5	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6 9 9099 826454 1 3 1 4	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7 9 9099 826454		

STOP STATEMENT EXECUTED.				TRANSMISSION SIMULATION			
1	6	7	0	0	0	0	0
6	6	0	0	0	0	0	0
7	7	0	0	0	0	0	0
7	0	9099	0	0	0	0	0
		0	0	0	0	0	0
	9098	9099	9099	0	0	0	0
	9098	9099	9099	0	0	0	0
	0	9099	9099	0	0	0	0
	0	9099	9099	0	0	0	0
	0	9099	9099	0	0	0	0
	0	9099	9099	0	0	0	0
	0	9099	9099	0	0	0	0
	0	9099	9099	0	0	0	0
8	8	9099	826454	2	3	4	4
8	8	9099	826454	2	3	4	4
1	3	1	3	3	3	4	4
4	4	4	4	3	3	4	4

付 録 2

タスク・プログラムの一例


```

;;      SUB TASK F000
      ;TTL F000;
      ;ENT F000;
      ;ENT F000,FXA,FYA,FZA,MXA,MYA,MZA
      ;EXT Q,S,B,C,CD,CY,CL,C1,CM,CN
      ;EXT COSA,SINA,LXCG,LYCG
      ;EXT FXJ,FYJ,FZJ,FXG,FYG,FZG
      ;EXT MXJ,MYJ,MZJ,MXG,MYG,MZG
      ;EXT DIVIDE,DADD
      ;POOL
FXA:   ;RES 2
FYA:   ;RES 2
FZA:   ;RES 2
MXA:   ;RES 2
MYA:   ;RES 2
MZA:   ;RES 2
      ;REL
F000:  ;BSS F000
      CAL 0
F000:  ;::
      ;; Q*S @-6
      LD R1,*AQ           ;; Q @3
      LD R2,*AS           ;; S @1
      MPY R2              ;; Q*S @4 DOUBLE
      LD R2,;1B10
      BSS *ADIVIDE
      ST R1,QS           ;; Q*S @-6 SINGLE
      ;; Q*S*B @-14
      LD R1,*AB           ;; B @7
      LD R2,QS           ;; Q*S @-6
      MPY R2              ;; Q*S*B @7-6+1-16=-14
      BSS LSHIFT1
      ST R1,QSB
      ;; Q*S*C @
      LD R1,*AC           ;; C @8
      LD R2,QS           ;; Q*S @-6
      MPY R2              ;; Q*S*C @8-6+1-16=-13
      BSS LSHIFT1
      ST R1,QSC
      ;; FXS*COSA @10
      LD R1,QS           ;; Q*S @-6
      CPL R1             ;; -Q*S @-6
      INC R1             ;;
      MOV R3,R1          ;; (R3)=-Q*S @-6
      LD R2,*ACOSA       ;; COSA @15
      MPY R2             ;; -Q*S*COSA @-6+15+1-16=-6
      BSS LSHIFT1
      ST R1,QSCOSA
      LD R2,*ACD         ;; CD @13
      MPY R2             ;; FXS(-Q*S*CD)*COSA @-6+13+3=10
      BSS DLSHIFT3
      ST R0,FXSCA
      ST R1,FXSCA+1
      ;; FXS*SINA @10
      MOV R1,R3          ;; -Q*S @-6
      LD R2,*ACD         ;; CD @13
      MPY R2             ;; FXS=-Q*S*CD @-6+13+1-16=-8
      BSS LSHIFT1
      LD R2,*ASINA       ;; SINA @15
      MPY R2             ;; FXS*SINA @-8+15+3=10
      BSS DLSHIFT3
      ST R0,FXSSA
      ST R1,FXSSA+1
      ;; FZS*COSA @10
      LD R1,QSCOSA       ;; -Q*S*COSA @-6

```

```

LD      R2,*ACL          ;;
MPY     R2                ;; FZS(=-Q*S*CL)*COSA @-6+13+3=10
BSS     DLSHIFT3
ST      R0,FZSCA
ST      R1,FZSCA+1
;; -FZS*SINA @10
LD      R1,@S
LD      R2,*ACL          ;; Q*S @-6
MPY     R2                ;; CL @13
BSS     LSHIFT1          ;; FZS=-Q*S*CL @-6+13+1-16=-8
LD      R2,*ASINA        ;; SINA @15
MPY     R2                ;; -FZS*SINA @-8+15+3=10
BSS     DLSHIFT3
ST      R0,FZSSA
ST      R1,FZSSA+1
;; MXS*COSA @4
LD      R1,@SB
LD      R2,*ACOSA        ;; Q*S*B @-14
MPY     R2                ;; Q*S*B*COSA @-14+15+1-16=-14
BSS     LSHIFT1
ST      R1,@SBCOSA
LD      R2,*AC1
MPY     R2                ;; MXS(=Q*S*B*C1)*COSA @-14+15+3=4
BSS     DLSHIFT3
ST      R0,MXSCA
ST      R1,MXSCA+1
;; MXS*SINA @4
LD      R1,@SB
LD      R2,*AC1          ;; Q*S*B @-14
MPY     R2                ;; MXS=Q*S*B*C1 @-14+15+1-16=-14
BSS     LSHIFT1
LD      R2,*ASINA        ;; MXS*SINA @-14+15+3=4
MPY     R2
BSS     DLSHIFT3
ST      R0,MXSSA
ST      R1,MXSSA+1
;; MZS*COSA @4
LD      R1,@SBCOSA
LD      R2,*ACN          ;; Q*S*B*COSA @-14
MPY     R2                ;; CN @15
BSS     DLSHIFT3        ;; MZS(=Q*S*B*CN)*COSA @-14+15+3=4
ST      R0,MZSCA
ST      R1,MZSCA+1
;; -MZS*SINA
LD      R2,@SB          ;; -Q*S*B @-14
CPL     R2
INC     R2
LD      R1,*ACN          ;; CN @15
MPY     R2                ;; -MZS=-Q*S*B*CN @-14+15+1-16=-14
BSS     LSHIFT1
LD      R2,*ASINA        ;;
MPY     R2                ;; -MZS*SINA @-14+15+3=4
BSS     DLSHIFT3
ST      R0,MZSSA
ST      R1,MZSSA+1
BRN     S0
LSHIFT1; ;;
SLL     R0,1
SRL     R1,8
SRL     R1,7
ADR     R1,R0
BRS     LSHIFT1
;;
DLSHIFT3; ;;
MOV     R2,R1
SLL     R0,3

```

```

SRL R2,8
SRL R2,5
ADR R0,R2
SLL R1,3
BRS DLSHIFT3

;;
RSHIFT3: ;;
MOV R1,R0
SLL R1,8
SLL R1,5
SRA R0,3
BRS RSHIFT3

;;
ADIVIDE: DIVIDE
;1B10: 1B10
AQ: Q ;; Q @3
AS: S ;; S @1
AB: B ;; B @7
AC: C ;; C @8
ACOSA: COSA ;; COSA @15
ASINA: SINA ;; SINA @15
ALYCG: LYCG
ALXCG: LXCG
ACD: CD
ACY: CY
ACL: CL
AC1: C1
ACM: CM
ACN: CN
QS: ;RES 1 ;; Q*S @-6
QSB: ;RES 1 ;; Q*S*B @-14
QSC: ;RES 1 ;; Q*S*C @-13
QSCOSA: ;RES 1 ;; -Q*S*COSA @-6
QSB COSA: ;RES 1 ;; Q*S*COSA @-14
ADADD: DADD
AFXSCA: FXSCA
FXSCA: ;RES 2 ;; FXS*COSA @10 DOUBLE
AFXSSA: FXSSA
FXSSA: ;RES 2 ;; FXS*SINA @10 DOUBLE
AFZSCA: FZSCA
FZSCA: ;RES 2 ;; FZS*COSA @10 DOUBLE
AFZSSA: FZSSA
FZSSA: ;RES 2 ;; -FZS*SINA @10 DOUBLE
AMXSCA: MXSCA
MXSCA: ;RES 2 ;; MXS*COSA @4 DOUBLE
AMXSSA: MXSSA
MXSSA: ;RES 2 ;; MXS*SINA @4 DOUBLE
AMZSCA: MZSCA
MZSCA: ;RES 2 ;; MZS*COSA @4 DOUBLE
AMZSSA: MZSSA
MZSSA: ;RES 2 ;; -MZS*SINA @4 DOUBLE
ATEMP: TEMP
TEMP: ;RES 2
;; FXA=(FXS*COSA-FZS*SINA)+FXJ+FXG @10
SD:
LD R0,*AFXJ ;; FXJ @-3
BSS RSHIFT3 ;; FXJ @-3-3+16=10
AD R0,*AFXG
LD R2,*AFXSCA
BSS *ADADD
LD R2,*AFXSSA
BSS *ADADD
LD R2,*FXA:
ST R0,0,R2
ST R1,1,R2
;; FYA=Q*S*CY+FYJ+FYG

```

```

LD      R0,*AFYJ
BSS     RSHIFT3
AD      R0,*AFYG
ST      R0,TEMP
ST      R1,TEMP+1          ;; FYJ+FYG @ 10
LD      R1,@S              ;; Q*S @-6
LD      R2,*ACY            ;; CY @15
SLL     R2,1               ;; Q*S*CY @15+1-6=10
MPY     R2
ST      R0,FYAS            ;; FYA @-6
LD      R2,ATEMP
BSS     *ADADD
LD      R2,FYA;
ST      R0,0,R2
ST      R1,1,R2
;; FZA=(FXS*SINA+FZS*COA)+FZJ+FZG
LD      R0,*AFZJ
BSS     RSHIFT3
AD      R0,*AFZG
LD      R2,AFXSSA
BSS     *ADADD
LD      R2,AFZSCA
BSS     *ADADD
LD      R2,FZA;
ST      R0,0,R2
ST      R1,1,R2
;; MXA=(MXS*COA-MZS*SINA)-LYCG*FZA'+MXJ+MXG
LD      R1,FXSSA
AD      R1,FZSCA           ;; FZA' @-6
ST      R1,FZAS
LD      R2,*ALYCG         ;; LYCG @11
SRA     R2,1
CPL     R2
INC     R2
MPY     R2                 ;; -LYCG*FZA' @-6+11-1=4
LD      R2,*AMXJ           ;; MXJ @-11
SRA     R2,1
ADR     R0,R2
LD      R2,*AMXG           ;; MXG @-11
SRA     R2,1
ADR     R0,R2
LD      R2,AMXSCA
BSS     *ADADD
LD      R2,AMZSSA
BSS     *ADADD
LD      R2,MXA;
ST      R0,0,R2
ST      R1,1,R2
;; MYA=Q*S*C*CM-LXCG*FZA'+MYJ+MYG
LD      R1,@SC             ;; Q*S*C @-13
LD      R2,*ACM            ;; CM @ 13
SLL     R2,1
MPY     R2                 ;; MYS=Q*S*C*CM @-13+13+1+3=4
BSS     *ADLSH13
ST      R0,TEMP
ST      R1,TEMP+1
LD      R1,*ALXCG         ;; LXCG @11
SRA     R1,1
CPL     R1
INC     R1
LD      R2,FZAS            ;; FZA' @-6
MPY     R2                 ;; -LXCG*FZA' @4
LD      R2,*AMYJ           ;; MYJ @-11
SRA     R2,1
ADR     R0,R2
LD      R2,*AMYG           ;; MYG @-11

```

```

SRA    R2,1
ADR    R0,R2
LD     R2,ATEMP
BSS    *ADADD
LD     R2,MYA;
ST     R0,0,R2
ST     R1,1,R2
;; MZA=(MXS*SINA+MZS*COSA)-LXCG*FYA'+MZJ+MZG
LD     R1,*ALXCG          ;; LXCG @11
LD     R2,FYAS            ;; FYA' @-6
SRA    R1,1
CPL    R1
INC    R1
MPY    R2                ;; -LXCG*FYA' @4
LD     R2,*AMZJ          ;; MZJ @-11
SRA    R2,1
ADR    R0,R2
LD     R2,*AMZG          ;; MZG @-11
SRA    R2,1
ADR    R0,R2
LD     R2,AMXSSA
BSS    *ADADD
LD     R2,AMZSCA
BSS    *ADADD
LD     R2,MZA;
ST     R0,0,R2
ST     R1,1,R2
BRS    *AF000
AF000: F000
ADLSHI3: DLSHIFT3
FYAS:   :RES 1          ;; FYS @-6
FZAS:   :RES 1          ;; FXS*SINA+FZS*COSA @-6
AFXJ:   FXJ
AFXG:   FXG
AFYJ:   FYJ
AFYG:   FYG
AFZJ:   FZJ
AFZG:   FZG
AMXJ:   MXJ
AMXG:   MXG
AMYJ:   MYJ
AMYG:   MYG
AMZJ:   MZJ
AMZG:   MZG
FXA:;   FXA
FYA:;   FYA
FZA:;   FZA
MXA:;   MXA
MYA:;   MYA
MZA:;   MZA
        :END

```

付 録 3

LEPのパラメータ例

```

//EXEC LEP
//CTRL *
BI,1.
BO,1.
LO,0.
MD,1.
ET.
JS,FSPP,FSPP:LB,CACDMODL.
CS,2,2,2,2,1.
CS,1,84,42,21,8.
TS,A000:,2,3F,1.
TS,F130:,2,3E,1.
TS,F132:,2,3D,1.
TS,F134:,2,3C,1.
TS,A020:,2,3B,1.
TS,A040:,1,3F,1.
TS,F150:,1,3E,1.
TS,F100:,1,3D,1.
TS,F101:,1,3C,1.
TS,F102:,1,3B,1.
TS,F103:,1,3A,1.
TS,F104:,1,39,1.
TS,F105:,1,38,1.
TS,F000:,1,37,1.
TS,F010:,1,36,1.
TS,F020:,1,35,1.
TS,F030:,1,34,1.
TS,F040:,1,33,1.
TS,F050:,1,32,1.
TS,F320:,1,2F,1.
TS,E003:,1,2E,1.
TS,E100:,1,2D,1.
TS,E103:,1,2C,1.
TS,E104:,1,2B,1.
TS,E121:,1,2A,1.
TS,F400:,1,29,1.
TS,E160:,1,28,1.
TS,A010:,1,27,1.
TS,F060:,1,1F,1.
TS,F106:,1,1E,1.
TS,F110:,1,1D,1.
TS,E111:,1,1C,1.
TS,E120:,1,1B,1.
TS,A030:,1,1A,1.
TS,E130:,1,19,1.
TS,F055:,1,0F,0.
TS,F056:,1,0E,1.
TS,F300:,1,0D,1.
TS,F340:,1,0C,1.
TS,E140:,1,0B,1.
TS,E144:,1,0A,1.
TS,E170:,1,09,1.
LS,DIWORD,DI,10F.
LS,DWAI, AI,80.
LS,DSAI, AI,81.
LS,DPAI, AI,82.
LS,FWCAI, AI,86.
LS,FSCAI, AI,87.
LS,FPCAI, AI,88.
LS,DTTAAI, AI,89.
LS,SFRLAI, AI,8A.
LS,DTTRAI, AI,8B.
LS,FLP, AI,8D.
LS,DT1, AI,8E.
LS,DT2, AI,8F.
LS,PAIM, AO,80.
LS,THETM, AO,81.
LS,VIAO, AO,83.
LS,HPAO, AO,85.
LS,PAIAO, AO,86.
LS,THAO, AO,87.
LS,PSIAO, AO,88.
LS,RBAO, AO,89.
LS,RAAO, AO,8A.
LS,RCAO, AO,8B.
LS,FWAO, AO,8E.
LS,FSAO, AO,90.
LS,FPAO, AO,91.
LS,PAIVD2, AO,95.
LS,THEVD2, AO,96.
LS,PSIVD2, AO,97.
LS,XVD2, AO,98.
LS,YVD2, AO,99.
LS,ZVD2, AO,9A.
LS,EPRAO, AO,F0.
LS,EPLAO, AO,F1.
LS,N2RAO, AO,F2.
LS,N2LAO, AO,F3.
LS,EGRAO, AO,F4.
LS,EGLAO, AO,F5.
ES.
PN.
PR.
RD,FSPP.
LD.
EX.

```

航空宇宙技術研究所報告511号

昭和52年8月発行

発行所 航空宇宙技術研究所
東京都調布市深大寺町1880
電話武蔵野三鷹(0422)47-5911(大代表)〒182
印刷所 株式会社 共 進
東京都杉並区久我山4-1-7(羽田ビル)
