



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*

Abstract

This report represents the data transmission and control in computer complex, for real time simulation, which is constructed with five minicomputers.

One of them, No. 0 Computer, controls input-output bus circuit which broadcasts data and commands to each computer.

User can produce the task-programs without consideration of input-output instructions between the computers.

Linkage editor identifies, classifies the variables to be transmitted.

It also generates control informations by which No. 0 computer transmits variables and by which edits them for analog-digital, digital-analog converter.

Supervisor in No. 0 computer repeats communications at intervals of 10 ms with real time system monitor in the other computers.

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

1. ま え が き

並列処理に関する報告は、1個の命令制御装置の下に複数の演算装置を配列するもの(単一命令ストリーム複数データストリーム)と、独立に命令を実行できる演算装置を配列するもの(複数命令ストリーム複数データストリーム)に大別され、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)}への応用が提案されている。

これらは、レジスタのレベルで各演算装置が結合されている。

他方、1台の計算機では、演算速度・費用効果の面で難点を生ずるような場合、そのジョブを並列処理可能なタスクに分割し、各タスクを比較的安価な計算機で処理し、それらをチャンネル間で結合することにより、全体として性能の向上を目的とする Computer Complex (以後複合計算機と言う)が出現している⁷⁾。

これは、広域的に分散し、独立処理を原則とする計算機を通信回線で結合し、資源・情報の共有を目的とするコンピュータ・ネットワークとは異なるものである⁸⁾。

複合計算機は、その目的・処理内容・規模に応じ、要素となる計算機を一定の手法で増加させることにより、性能向上が期待でき、かつ必要に応じて計算機群を分割し、多目的同時使用ができる等、多くの利点が存在する。

複合計算機では、その目的・並列処理のアルゴリズム・各計算機間の同期・データ送受・応用プログラム作製手法を詳細に検討し、使い易く、効率の良い方法を取るべきである。

本報告は、航空宇宙技術研究所に、汎用飛行シミュレータ航空機動特性模擬装置として設置したミニコン(MELCOM-70)5台から成る複合計算機(FSK-II)で採用した計算機間のデータ転送と制御について述べる。

2. 実時間シミュレーションの概要

実時間シミュレーションとは、物理時間と計算機内

* Data Transmission and Control in Computer Complex (FSK-II) for Simulation by Koichi HARADA (Instrument and Control, National Aerospace Laboratory)

** 科学技術庁航空宇宙技術研究所計測部

の論理時間とが一致しているシミュレーションのことである。したがって、計算機は、初期値を与えられると、対応する物理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・現実金物（例、ジャイロ）の性能試験・パイロットの訓練のように環境を実時間で模擬する必要のある分野では、不可欠な手法である。

実時間との同期を取るため、計算機にリアル・タイム・クロック (RTC) を実装し、一定時間間隔で繰り返し演算を行う⁹⁾。

応答特性により、速く繰り返すものと、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する、(Fig. 1)

例えば、操舵反力の演算は、10ms に1度の繰り返しで解くのにに対し、飛行力学のそれは40~50msで繰り返す。

これらの多くは、連立常微分方程式の解法である。

人間の操作量・現実金物によるアナログ信号を計算機に取り込むとき、A/D 変換器を通し、演算結果を

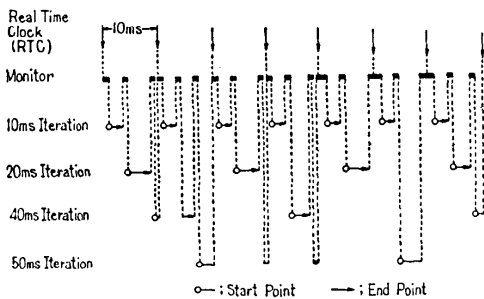


Fig. 1 A Sample of Task Schedule in a Computer

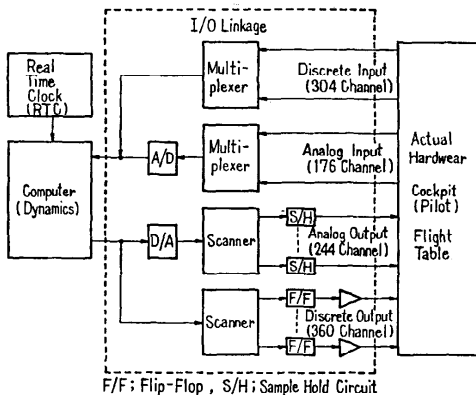


Fig. 2 I/O Linkage Configuration

アナログ信号で出力するとき、D/A 変換器を通す。

この他に、1/0 で入出力する信号 (Discrete 入出力信号と言う) がある。

この装置を I/O リンケージと言い、高速データ転送を要求されるため、DMA チャンネルに接続される。

アナログ入出力信号は、I/O リンケージで符号を含み11ビットで扱われ、計算機内では1語 (16ビット) が割当てられる。(Fig. 2)

3. ハードウェア構成

当所に設置した計算機の構成を Fig. 3 に示す。

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

ICCLはDMAチャンネル結合により、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 と、PIOチャンネル結合により、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 から構成される。

HSBCは、非同期の計算機間でバースト転送を行うため、計算機2台に1組8段のバッファを設け、最大826kw/秒の転送を行うことができる。転送レートとバッファ段数との関係をシミュレートした結果を Table 1 に示す。5段で充分であるが、8段バッファを作り易い MSI を使用した¹⁰⁾。

Fig. 4 (次頁参照)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を経由し、インプット・

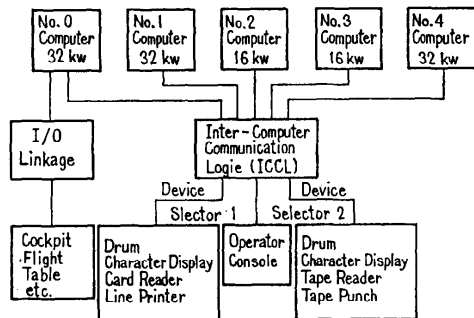


Fig. 3 System Configuration

Table 1 Data Transmission Rate

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

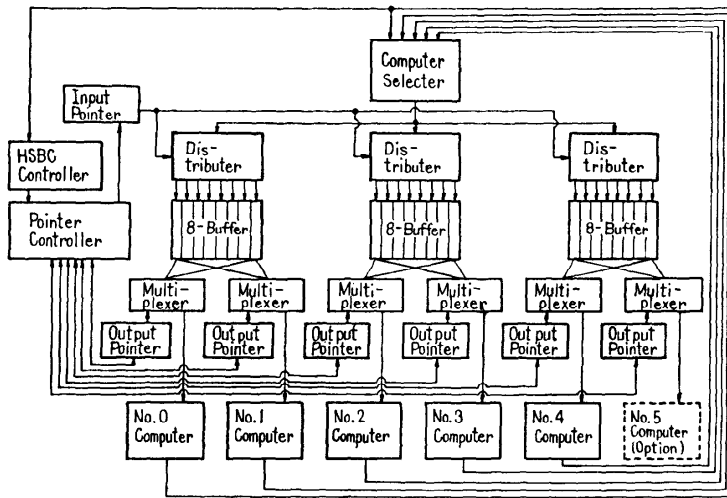


Fig. 4 HSBC Configuration

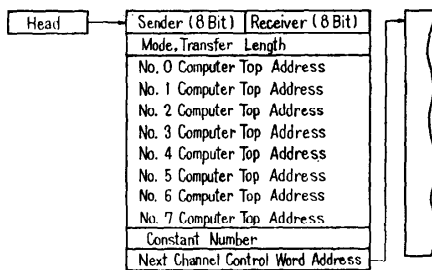


Fig. 5 CCW (Channel Control Word)

ポインタ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き込む。

これらのポインタは循環型になっており、バッファは、繰り返し使用される。

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 を有する。(Fig. 5) 以後この計算機を制御計算機と言う。

HSBC 制御部は制御計算機の起動により、この CCW を DMA チャン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。

転送が終了すると制御計算機に割込信号が送られ、誤動作が検知されたときには、即時転送を中断し、エラー情報を制御計算機の特

定アドレスに書き込み割込信号を送る。

LSBC は、各計算機の PIO チャンネルに 16 ビットのバッファと送り先を記憶する回路から成る。

信号を送ろうとする計算機は、相手を1台以上指定した後に、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み込むと、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。

4. 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 (スーパーヴァイザー、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コンパイラ、リンケージエディタ等から構成される。(Fig. 6)

ここでは、実時間シミュレーション時のデータ転送に関連するものについて述べる。

4.1 スーパーヴァイザー

これは制御計算機内に常駐し、他計算機内の RTSM に対して同期信号等を送り、計算機間のデータ転送、および I/O リンケージの制御を行う。

4.2 RTSM

これは、制御計算機以外の計算機内にあり、実時間シミュレーションの応用プログラムを一定時間間隔で

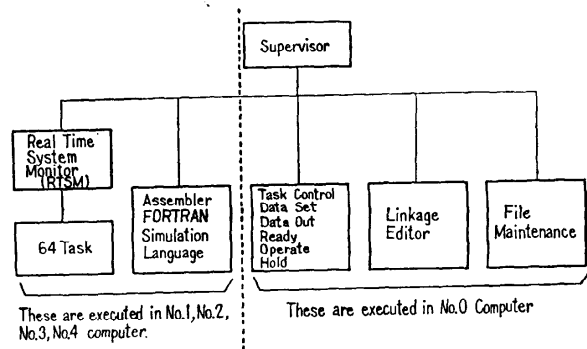


Fig. 6 Software Configuration

Table 2 Supervisor Commands to RTSM

機能	コード	備考
RTSM Ready	0000	RTSM の初期値設定.
RTSM Active	0001	RTSM の Task Schedule を実行する.
RTSM Pause	0010	RTSM の Task Schedule を停止する.
Task Control	0011	指定した Task の Mode を変更する.
Task Register	0100	Task とその開始番地を指定して Task を登録する.
Data Request	0101	Address を指定して 1 語のデータを要求する.
Data Transmit	0110	RTSM の Data Request に応答してデータを送る.
RTC Set	0111	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	1000	指定した Task の実行周期を更新する.
Data Store	1001	Address と Data を指定し, 格納する.
Schedule Restart	1010	停止していた Task Schedule を再開する.
Response	1111	RTSM の要求が Supervisor で終了したことを知らせる.

Table 3 RTSM Commands to Supervisor

機能	コード	備考
Task State	0000	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	0001	RTSM 内の Task Control Block に変更が生じたことを知らせる.
Data Request	0010	Supervisor に Computer, Address を指定して 1 語のデータを要求する.
Data Transmit	0011	Supervisor の Data Request に応答してデータを送る.
Task Caution	0100	Task 動作時, 異常状態が起きたことを知らせる.
RTSM Ready	0101	他 Computer 内 RTSM の初期値設定.
RTSM Active	0110	他 Computer 内 RTSM の Task Schedule を実行する.
RTSM Pause	0111	他 Computer 内 RTSM の Task Schedule を停止する.
Task Control Monitor Fail	1000	他 Computer 内 Task の Mode を変更する.
	1001	RTSM が正常動作できないことを知らせる.
Schedule Wait	1010	Task Schedule が中断したことを知らせる.
Response	1111	Supervisor の要求が RTSM で終了したことを知らせる.

繰り返す制御を行い, 4 レベルの繰り返し時間 (10 ms の倍数) を指定し, 各レベル 16 個のタスクを制御することができる.

スーパーバイザーと RTSM の会話は LSBC を通して行われ, その種類を Table 2, 3 に示す.

5. リンケージ・エディタ

物理的に分散した要素より成る対象のシミュレーション・プログラムは, 各要素ごとに分割作製され, さらにそれらは必要に応じ, 細分化される.

その単位がタスクである.

飛行シミュレーションの場合, 飛行力学を約 30 個, エンジン約 10 個のタスクに分割作製する.

各タスクの入力を **External**, 出力を **Entry** と宣

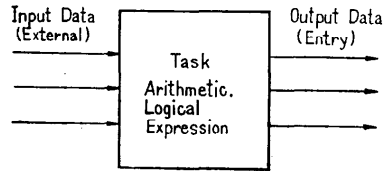


Fig. 7 Input/Output Data of a Task

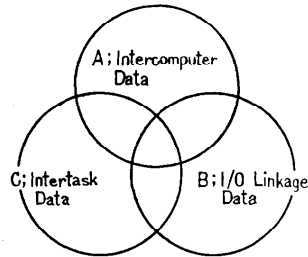


Fig. 8 Data Classification

言し, 識別する. (Fig. 7)

タスクに入出力する変数を 3 種に区別できる.

(Fig. 8)

(A) 他計算機内にあるタスクにより参照される変数. 転送を要する.

(B) I/O リンケージを経由する変数. 転送を要する.

(C) 自己計算機内のタスクによって参照される変数.

分割作製されたタスクは, リンケージ・エディタにより結合される.

リロケータブル・オブジェクト・プログラムのタスクを結合し, ロード・モジュール, 転送制御情報, 編集情報を得る手順は次の通りである. (Fig. 9)

- (1) 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.
- (2) CS コマンドで使用する計算機番号・繰り返し

- (1) JS, Job Name, Library Name (1), Library Name (2), ...
- (2) CS, Computer Number, Iteration Time (1) ..., Iteration Time (4).
- (3) TS, Task Name, Computer Number, Task Number, Task Control Flag.
- (4) LS, Variable Name, Mode (AI/AO/DI/DO), Channel Number.
- (5) ES.
- (6) LD.
- (7) EX.

Fig. 9 Parameter for Linkage Editor

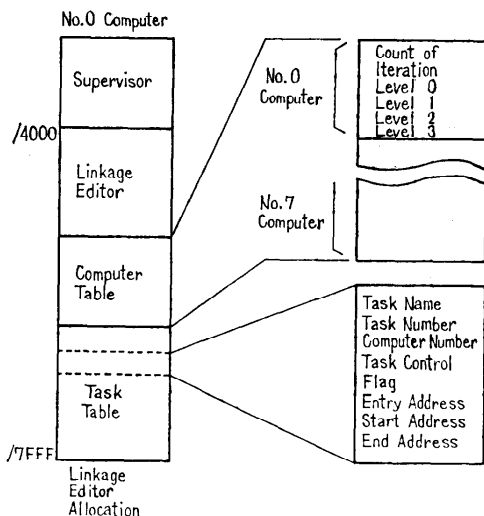


Fig. 10 Tables on Core

Channel Symbol
Channel Number
AO/AI/DO/DI

Fig. 11 Linkage Table

し時間を指定する。計算機テーブルをコア上に作る。(Fig. 10)

- (3) 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。

- (4) LS コマンドで I/O リンケージのチャンネル番号と入出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。(Fig. 11)

- (5) ES コマンドでタスクのリンク・エディットを開始する。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。

- (6) JS 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。

- (7) 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により、入力変数は、制御計算機で Entry 宣言されたものと見なし、出力変数は、制御計算機で External 宣言されているものとする。

- (8) 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属性

Symbol
Equivalence
Property
Task/Library Number
Relative Pool Address
Pool Size
Channel Number

Fig. 12 Symbol Table

(単精度, 倍精度, アナログ入出力, ディスクリット入出力等)を記録する (Fig. 12)

- (9) 各計算機の Entry シンボル・テーブル内で, Pool 領域内にあるもの (変数は, この領域内に定義される.) を取り出し, 他計算機の External シンボル・テーブルを参照する。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。

- (10) 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。

- (11) 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 Fig. 13 のようになる。図中, Xは転送する変数とした例が示され, : RES はワークスペースを, \$ 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であることを示す。

- (12) 各計算機ごとに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制御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制御計算機へ転送する CCW および使用する計算

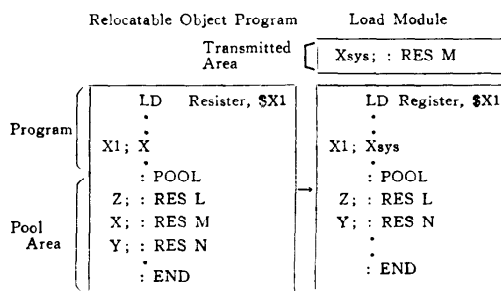


Fig. 13 Separation of Transmitted Data

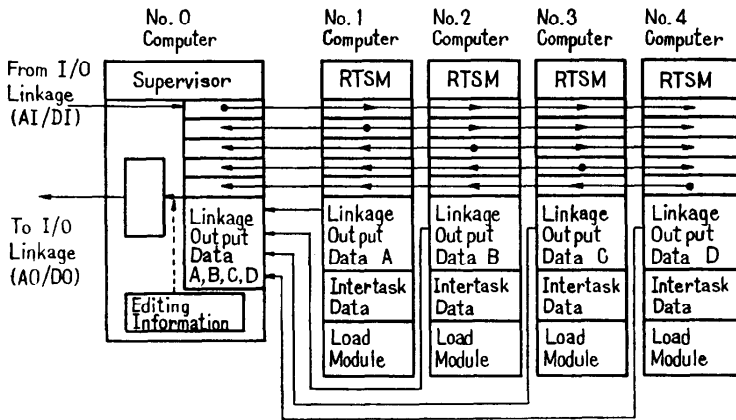


Fig. 14 Data Allocation

機を示すフラグから成る。

- (13) 計算機間のデータ転送はブロック転送であり、制御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ンネル順に並んでいない、リンケージ・エディタは、出力チャンネル順に並べ替える編集情報を生成する。
- (14) LD コマンドで、ロードモジュールを各計算機にロードする。その様子を Fig. 14 に示す。
- (15) EX コマンドで終了する。

6. 実時間シミュレーション時のデータ転送

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 Fig. 15 に示す。

それは、次の通りである。

- (1) 制御計算機の RTC から 10ms ごとに割り込みが掛かる。

- (2) 制御計算機は 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) 制御計算機は 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 (AI) のデータを入力する。

- (4) RTC が各計算機に割込むと、応用プログラムの演算を中断し、制御計算機に知らせる。

制御計算機は、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) 制御計算機は、リンケージ・エディタにより生成され、ロード時スーパーバイザーに渡されている転送制御情報を用いて HSBC を起動し、計算機間のデータ転送を行う。

- (6) データ転送終了後、各計算機に演算再開の信号を送る。

- (7) 制御計算機内に転送されたデータを I/O リンケージ出力チャンネル順に並べ替え、アナログ出力 (AO), ディスクリット出力 (DO) のデータを出力する。この時、ディスクリット出力は論理数であるから、16 語を 16 ビットにパックする。これで一周期が完了する。

一定周期の繰り返しで全計算機に同じデータを持た

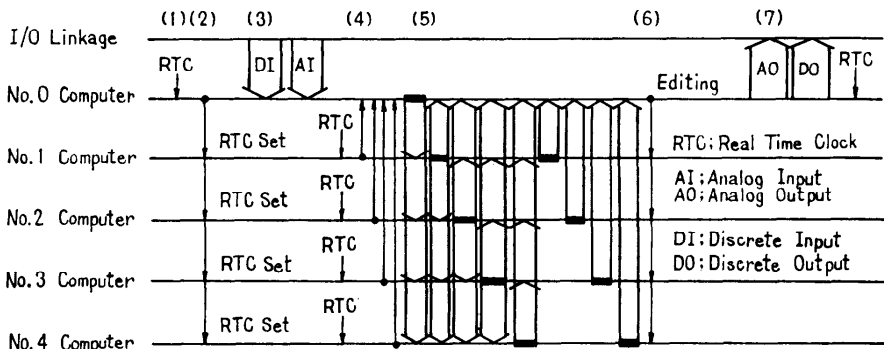


Fig. 15 Synchronizing Time Chart

Table 4 Data Transmission in Runge-Kutta Method

	No. i Computer	No. j Computer
1	x_i^0 Broadcast	x_j^0 Broadcast
2	$k_i^1 = \Delta x_0 \times f_i(x_0^0, \dots, x_n^0)$ $x_i^1 = x_i^0 + (1/2)k_i^1$	$k_j^1 = \Delta x_0 \times f_j(x_0^0, \dots, x_n^0)$ $x_j^1 = x_j^0 + (1/2)k_j^1$
3	x_i^1 Broadcast	x_j^1 Broadcast
4	$k_i^2 = \Delta x_0 \times f_i(x_0^1, \dots, x_n^1)$ $x_i^2 = x_i^1 + (1/2)k_i^2$	$k_j^2 = \Delta x_0 \times f_j(x_0^1, \dots, x_n^1)$ $x_j^2 = x_j^1 + (1/2)k_j^2$
5	x_i^2 Broadcast	x_j^2 Broadcast
6	$k_i^3 = \Delta x_0 \times f_i(x_0^2, \dots, x_n^2)$ $x_i^3 = x_i^2 + k_i^3$	$k_j^3 = \Delta x_0 \times f_j(x_0^2, \dots, x_n^2)$ $x_j^3 = x_j^2 + k_j^3$
7	x_i^3 Broadcast	x_j^3 Broadcast
8	$k_i^4 = \Delta x_0 \times f_i(x_0^3, \dots, x_n^3)$ $x_i^4 = x_i^3 + (1/6)(k_i^1 + k_i^2 + k_i^3 + k_i^4)$	$k_j^4 = \Delta x_0 \times f_j(x_0^3, \dots, x_n^3)$ $x_j^4 = x_j^3 + (1/6)(k_j^1 + k_j^2 + k_j^3 + k_j^4)$

せ、連立常微分方程式

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

を解く方法を Table 4 に示す。

表中 1, 3, 5, 7 で各変数を転送し, 2, 4, 6, 8 で演算を行い, 4 周期で Runge-Kutta 法の 1 ステップが完了する。

ここで, 各演算は, どの計算機で実行されてもよく, Δx_0 は独立変数のきざみ幅を示す。

7. むすび

本装置は既に完成し, 次のことが確かめられている。

- (1) 計算機間で対一の転送をすると, その台数のほぼ二乗に比例して, 転送負荷が増大する。
ハードウェアでブロードキャスト転送を行い, 計算機数にほぼ比例した負荷に押えることができた。
- (2) リンケージ・エディタのもとに, 転送データを連続する領域に確保し, 転送効率を向上させることができた。
- (3) 全計算機に同一データを周期的に持たせると, シミュレーションプログラム作製時にタスクの入出力実数を宣言するだけで, 計算機間の

入出力命令を記述する必要がない。

さらに, 通常のシミュレーションで, I/O リンケージを経由するデータは 50 語以下, 演算のために転送されるデータは 20 語程度, 転送に要する時間は割込み処理を入れて数百 μ s である。

謝辞 協力援助頂いた堀川室長, 名越技官, 携わる機会を与えてくださった故樋口元計測部長, またブロードキャスト, データ領域確保の提案に積極的に応じてくださった三菱プレジジョンの諸氏に感謝する。

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: Modal Trajectory Estimation and Parallel Computers, Proc. 2nd Symposium on Nonlinear Estimation Theory, pp. 188~198 (Sept., 1971)
- 6) E. Tse: Parallel Computaton 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, p. 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)
- 10) R. C. Ghest: The Am 9338 Multiple Port Memory, Advanced Micro Devices, pp. 379~382

(昭和 51 年 3 月 31 日受付)

(昭和 51 年 11 月 17 日再受付)