

ミニコン用コンパイラ

NIT-FORTRAN について

丹羽 敏行・本田 行雄

電子工学教室

(1971年9月13日受理)

Design of NIT-FORTRAN Compiler for Mini-Computer

Toshiyuki NIWA and Yukio HONDA

Department of Electronics

(Received September 13, 1971)

There are several FORTRAN compilers for mini-computer to be used for various purposes. They are insufficient for scientific use or instructional use. Then the new FORTRAN is designed and practiced. This NIT-FORTRAN has many merits, especially, high speed execution, the grammar over the level JIS 3000 and easy handling. The sample program which solves 20 simultaneous equations is able to be carried out by the "Compile and Go" method. Twenty solutions have been given after 45 sec. cpu times.

1. ま え が き

最近、ミニコン（超小型電子計算機）が多方面で種々活用されており、特に研究室単位での科学技術計算用として、また高等学校などにおける情報処理教育用として、今後著しい普及が見込まれている。

このような背景から、ソフトウェアの面では、ミニコンといえどもアセンブラはもちろん BASIC, FORTRAN などのコンパイラも用意されるようになった。しかし、研究用、教育用として、まだ不十分な点が多い。

そこで、筆者らもミニコン用として、JIS 3000 レベルの FORTRAN コンパイラを作成し、NIT-FORTRAN と名付け、デバッグもほぼ完了したので、その概要を報告する。

2. ハードウェアの構成と仕様

コンパイラ開発の対象としたミニコンは、Fig. 1 のように、基本構成（記憶容量 4 Kw+入出力タイプライタ）に近いものとした。その仕様は次の通りである。

<CPU>

語 長	16ビット+パリティビット
サイクルタイム	1.5 μ s
記憶容量	4 Kw—32Kw
演算方式	並列2 進法
命 令 数	39 (基本) + 5 (オプション)
アドレス方式	1 1/2方式、ページ式
間接アドレス	指定可
修 飾	Bレジスタ修飾
割 込	4レベル

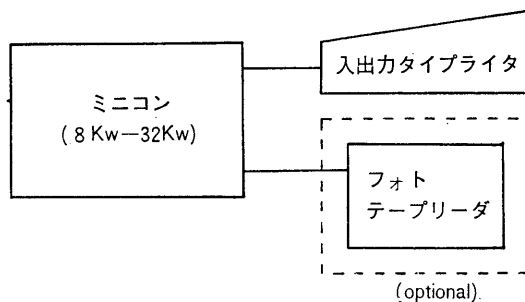


Fig. 1 Block diagram of Hardware

入出力制御	BUS方式
〈入出力装置〉	
タイプライタ	1000字/分
フォートテーブリーダー	500字/秒

3. コンパイラの仕様

本 FORTRAN の設計にあたり、研究用だけでなく、教育用としても充分使用できるよう配慮した結果、次のような種々の特長を持たせることができた。

(1) 最小機器構成

Fig. 1 にも示したように、基本構成 + 4 Kw の低価格のシステムから使用することができる。

(2) 翻訳と実行の手順

Compile and Go を原則とするが、ソース・プログラムが長くて、オブジェクト・プログラムが記憶装置にそのままでは入り切らない場合は、メッセージ印刷と同時に自動的にオブジェクト・プログラムをパンチ出力する。オブジェクト・プログラムは 2 進のままでも出力され、いつでも再入力し、実行することができる。

(3) オブジェクト・プログラムの形

翻訳、実行の方法には一般に Fig. 2 のように 2 通りの方法がある。

a) トランスレータ形式……これは、コンパイラがソース・プログラムを翻訳した結果得られるオブジェクト・プログラムが機械語になっていて、それを実行して目的の処理を行う方法である。一般に、この方法ではコンパイラの構成が複雑になるが、実行時間が短縮される。

b) インタープリタ形式……これは、ソース・プログラムをコード化された形のオブジェクト・プログラムに翻訳し、プログラムによってこれらのコードを解釈しながら実行する方法である。この方法では、コンパイラの構成が簡単になるが、実行時間が長くなる。

一般に、ミニコンのように小容量の記憶装置しか持たない場合は、実行速度を犠牲にしてコンパイラを簡単にするのが普通で、市販のミニコン用 FORTRAN は全てインタプリタ形式をとっている。本 FORTRAN では、あえて前者のトランスレータ形式を採用し、実行速度の向上をはかった。

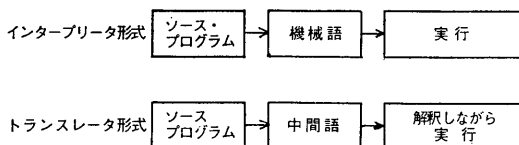


Fig. 2 Compiling and executing method

(4) 文法

本 FORTRAN のレベルは、補助入出力文を除いて JIS 3000 に準ずる。ただし、能率を落さない限りできるだけ拡張するようにした。また、あくまで上向きの互換性を考え、JIS 規格以外の記述をさけた。

JIS 3000 を拡張した部分は次のとおりである。

- a) 入出力変換に A 変換を許し、文字データも扱えるようにした。
- b) 文番号及び STOP 文と PAUSE 文に続く定数に 65535 までのものを許す。
- c) 継続行数を制限しない。
- d) DIMENSION 文は、添字付変数を使用する前であれば、どこに置いてよい。
- e) F あるいは E 変換で E あるいは F 型のデータを入力することができる。

(5) プログラムとデータの記述形式

プログラムは固定欄形式とし、データについては、データと FORMAT の作成を容易にするために、自由欄形式と固定欄形式の折衷式とした、すなわち、FORMAT 中の欄の巾 w および数値の型が入力データと対応づけられるが、改行コードあるいはスペースが w 桁内にあれば、その改行コードあるいはスペースをデータの切れ目とみなす。

(6) エラーメッセージ

教育用という観点から、エラーメッセージはコンパイル時 59、実行時 15 とできるだけ数を多くし親切なものとした。また、エラーメッセージ出力後も翻訳に支障のない限り、そのステートメントの残りのエラーもチェックし、軽度のエラーに対しては、適当な形にオブジェクト化する。

エラーメッセージは 3 つのレベルに分類され、軽度のものに対しては無視して実行することが可能、中度のものに対しては実行は不可能、重度の場合はコンパイル中止となる。また、実行時においては、それぞれのレベルに応じて、演算続行、演算一時停止、演算中止というようになる。

(7) 操作

翻訳から実行までの操作はできるだけ簡単にし、原則として、SCC の値をセットするのは一度だけですむようにした。また、翻訳時にスイッチ・レジスタを用いて、入力機器の選択と 1 ステートメントずつの処理が自由に指定できるようにし、デバッグの便をはかった。

4. 翻訳ルーチン

すでに述べたように、算術代入文及び無条件 GO TO 文などは直接機械語の形のオブジェクトになるが、その

計算形 GO TO	
文番号(番地)	
...	
変数の番地	
()内の数式のプログラム	
算術 I F	
文番号(番地)	
文番号(番地)	
文番号(番地)	
DO オープン	
DO クローズ	
パラメータ(m ₁)	
DO クローズ	
DO の変数	
パラメータ(m ₂)	
パラメータ(m ₂)	
繰り返し番地	
PAUSE	
10進数	
STOP	
10進数	
CALL	
実引数の番地	
...	
エンドコード	
RETURN	
繰り返し番地	

READ/WRITE
入出力機番
文番号(番地)
変数の番地
配列名
添字式
DOOコード
DOOの変数
パラメータ(m ₁)
DOOコード
DOOの変数
パラメータ(m ₁)
配列名
添字式
DOOのエンドコード
DOOの変数
パラメータ(m ₂)
パラメータ(m ₂)
繰り返し番地
DOOのエンドコード
DOOの変数
パラメータ(m ₂)
パラメータ(m ₂)
繰り返し番地
リストエンドコード

FORMAT
n
X
n
H
n
文字列
A
w
I
w
F
w
d
E
w
d
エンドコード

Fig. 3 Formats of object program

他のステートメントのオブジェクト形式を Fig.3に示す。

翻訳ルーチンを構成する主なプログラムは

- (1) ステートメント読み込みルーチン
- (2) ステートメント分類ルーチン
- (3) シラブル・リード・ルーチン
- (4) ステートメント処理ルーチン
- (5) オブジェクト格納ルーチン

である。

4-1. ステートメント読み込みルーチン

ソース・プログラムを1行読み込んで、所定のバッファに格納する。読み込んだステートメントが注釈行であれば、その上に次の1行を読み込み、継続行の処理をする。(Fig. 4)

バッファは Fig. 5 のような構成で、ソースの1字が1語に対応し、1行は72字までとし73字以降は読み飛ばす。また、72字以内に改行コードがあれば72字までスペースをつめる。

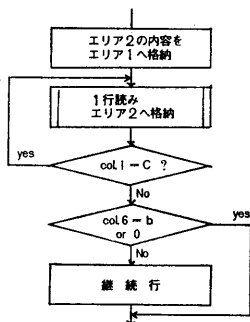


Fig. 4 Statement read routine

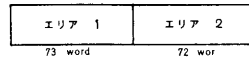


Fig. 5 Statment read area

4-2. ステートメント分類ルーチン

読み込まれたステートメントが、テーブルにある FORTRAN 語文字列 (GO TO, DO, FORMAT など) で始まっているかどうかをチェックし、一致すれば、その文字列に対応する各ステートメント処理ルーチンに分岐し、一致しない場合は算術代入文とみなして処理する。(Fig. 6)

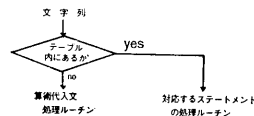


Fig. 6 Statement sort routine

4-3. シラブル・リード・ルーチン

シラブル・リード・ルーチンは変数処理ルーチンと定数処理ルーチンとで構成され、読み込まれた文字列が変数が定数かに応じてそれぞれの処理ルーチンへ分岐する。

変数処理ルーチンでは、変数が新しく定義されたものかどうかを判定し、新しい変数ならば、変数名か関数名あるいはサブルーチン名かを判定し、それぞれに番地を割り当てる。

定数処理ルーチンでは、読み込まれた文字列 (10進数) を2進の整定数あるいは実定数に変換し、割り当てられた番地に格納する。(Fig. 7)

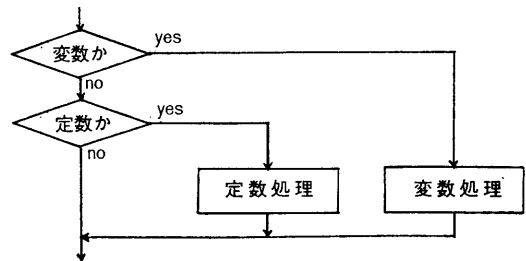


Fig. 7 Syllable read routine

4-4. ステートメント処理ルーチン

このルーチンは、代入文、制御文、入出力文、宣言文、および FORMAT 文を機械語命令に変換し、オブジェクト格納ルーチンを用いて、主記憶装置に格納するか、あるいはパンチ出力する。(Fig. 8~20)

4-5. オブジェクト格納ルーチン

翻訳されたオブジェクト・プログラムを1語ずつ主記憶装置に格納する。途中で1 pass のオブジェクト領域に入り切らなくなれば、それまでに格納したオブジェクト・プログラムを、再入力用 LOADER で読み込み可能な形のバイナリ・モードにしてパンチ出力する。以後、1ステートメント翻訳するごとにオブジェクト・プログラムをパンチ出力する。

パンチ出力するとき、次の点が問題になる。

- (1) オブジェクト・プログラム格納開始番地がずれる。
- (2) オブジェクト・プログラムの長さが、2 pass に切り替った時点では不定である。
- (3) テーブル類も一緒にパンチ出力しなければならない。

システム入力用のブート・ストラップではバイナリ・テープの先頭に、格納開始番地、語数、実行開始番地の3語がパンチされていないといけないので、上の形のパンチ出力されたオブジェクト・プログラムに対しては別の形のプログラム (LOADER) を用意しなければならない。

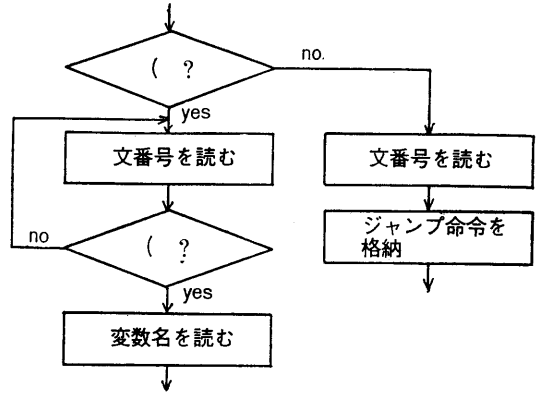


Fig. 9 Compile routine of GO TO statement and computed GO TO statement

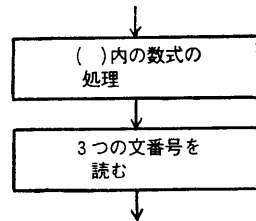


Fig. 10 Compile routine of arithmetic IF

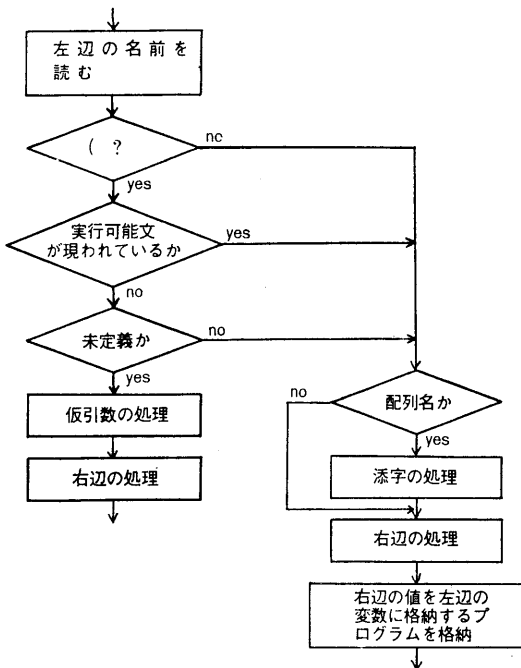


Fig. 8 Compile routine of arithmetic statement

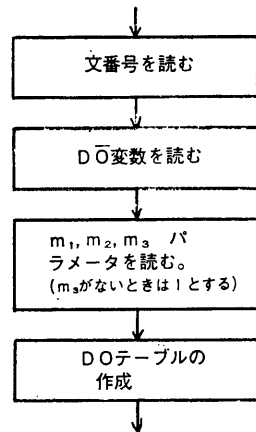


Fig. 11 Compile routine of DO statement

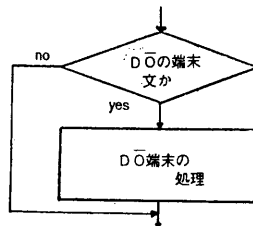


Fig. 12 Compile routine of CONTINUE statement

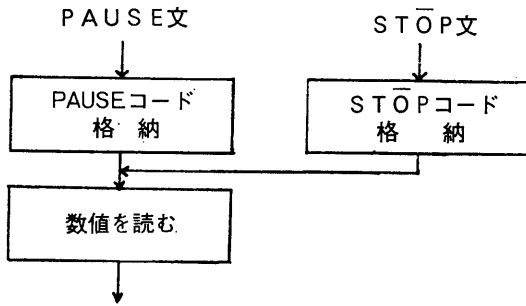


Fig. 13 Compile routine of PAUSE/STOP statement

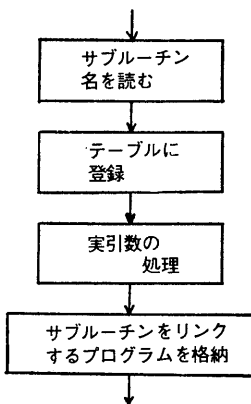


Fig. 14 Compile routine of CALL statement

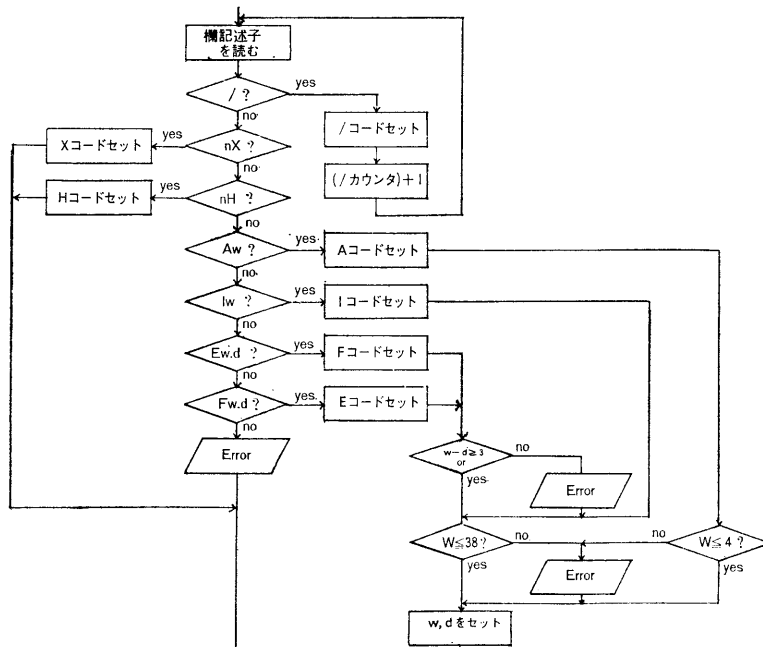


Fig. 17 Compile routine of FORMAT statement

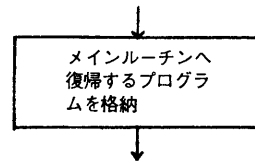


Fig. 15 Compile routine of RETURN statement

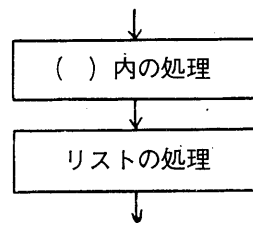


Fig. 16 Compile routine of READ/WRITE statement

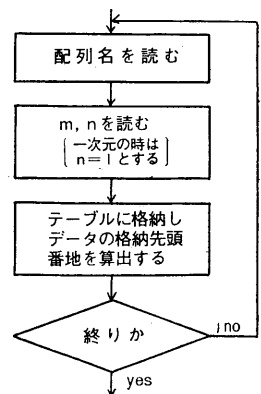


Fig. 18 Compile routine of DIMENSION statement

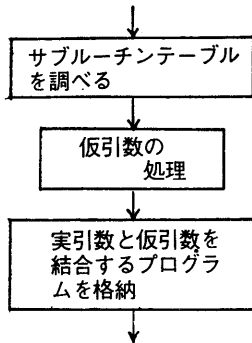


Fig. 19 Compile routine of SUBROUTINE statement

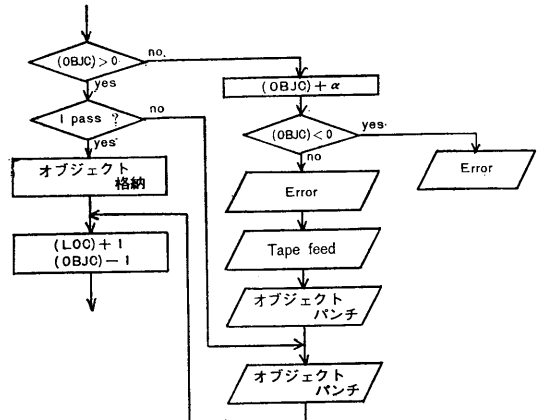


Fig. 21 Object store routine

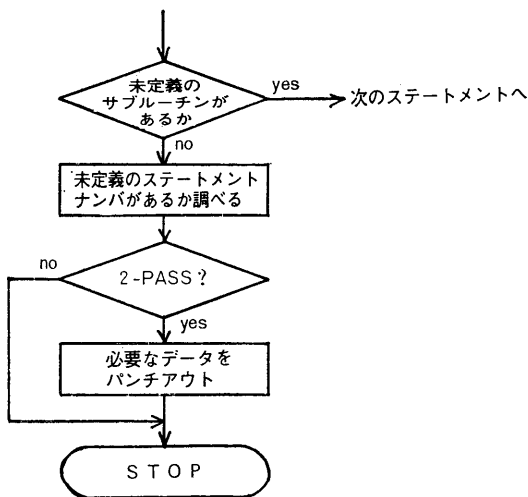


Fig. 20 Compile routine of END statement

5. 実行ルーチン

翻訳されたオブジェクト・プログラムを処理実行するルーチンは次のようなプログラムで構成されている。

- (1) 演算実行ルーチン
- (2) 計算型 GO TO 文実行ルーチン
- (3) 算術 IF 文実行ルーチン
- (4) DO 文実行ルーチン
- (5) DO close 実行ルーチン
- (6) PAUSE/STOP 文実行ルーチン
- (7) CALL 文実行ルーチン
- (8) RETURN 文実行ルーチン
- (9) READ/WRITE 文実行ルーチン
- (10) FORMAT 文実行ルーチン
- (11) オブジェクト再入力用 LOADER

Table 1 Arithmetic subroutines and Intrinsic functions

記号	内容	記号	内容
LD	2倍長のデータをロードする	IFIX	実数形→整数形
TD	2倍長のデータを格納する	FLOAT	整数形→実数形
AD	倍精度加算	IABS	x (整数形)
SD	倍精度減算	ABS	x (実数形)
MD	倍精度乗算	SIN	sin x
DD	倍精度除算	COS	cos x
FA	浮動小数点加算	ATAN	tan ⁻¹ x
FS	浮動小数点減算	TANH	tanh x
FM	浮動小数点乗算	EXP	e ^x
FD	浮動小数点除算	ALOG	log _e x
POW 1	べき乗 (整数の整数乗)	SQRT	√x
POW 2	べき乗 (実数の整数乗)	ISIGN	符号のつけかえ (整数形)
POW 3	べき乗 (実数の実数乗)	SIGN	符号のつけかえ (実数形)

5-1. 演算実行ルーチン

補助ルーチンを除いた各種の演算サブルーチンを、組み込み関数と共に Table 1 に示す。

5-2. 制御ステートメント実行ルーチン

制御ステートメントは無条件 GO TO を除いて、いずれも Fig. 3 のような2語以上で構成されているが、いずれも実行時の処理は簡単である。

5-3. 入出力ステートメント実行ルーチン

READ 文、WRITE 文の実行は、FORMAT の内容を参照しながら、Fig. 22 のような形で行われる。

FORMAT 文に制御の流れがきた時は、エンド・コードまで読み飛ばし、次の命令の実行に移る。

5-4. オブジェクト再入力用 LOADER

パンチ出力されたオブジェクト・プログラムをコンパイルの一部を壊して再入力する。(Fig. 23)

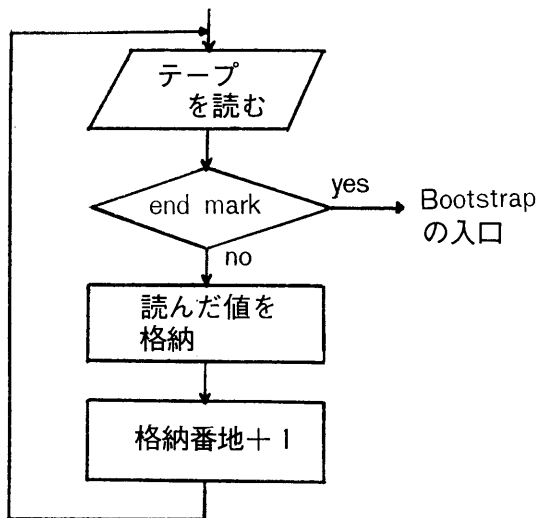


Fig. 23 Loader

6. 文 法

本 FORTRAN で使用できるステートメントは次のとおりである。

- (1) 算術代入文
- (2) 無条件 GO TO 文
- (3) 計算型 GO TO 文
- (4) 算術 IF 文
- (5) DO 文
- (6) CONTINUE 文
- (7) STOP 文
- (8) PAUSE 文
- (9) CALL 文
- (10) RETURN 文
- (11) READ 文
- (12) WRITE 文
- (13) FORMAT 文
- (14) DIMENSION 文
- (15) SUBROUTINE 文
- (16) 文関数定義文
- (17) END 文

主な制限事項は次のとおりである。

- (1) 整数の絶対値は $2^{31}-1$ を越えてはならない。
- (2) 実定数の絶対値は約 10^{+38} を越えてはならない。
- (3) 実定数は小数部、整数部とも9桁までとする。
- (4) 変数名は4文字までとする。
- (5) 文番号は 65535 以内の符号のつかない整数

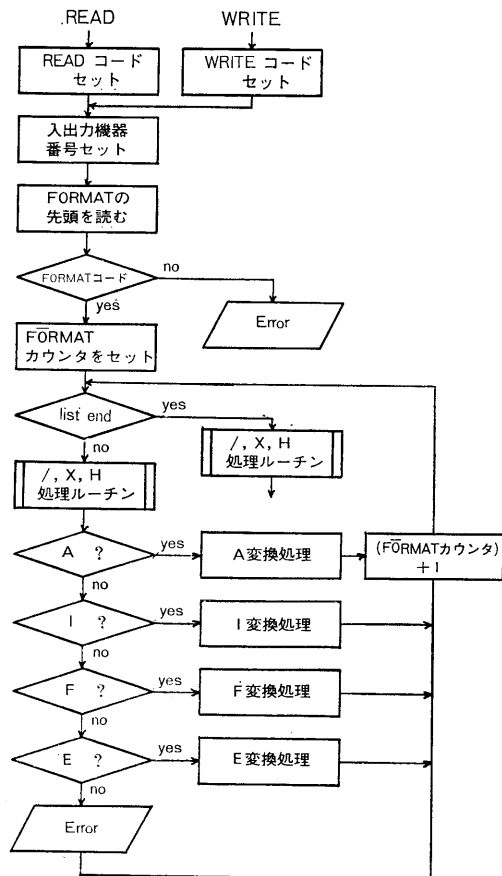


Fig. 22 READ/WRITE executer

でなければならない。

- (6) (定数の数) + (変数の数) ≤ 256
- (7) 文番号は 96 個までとする。
- (8) DO の深さは 10 重 (入出力リスト中の DO は含まない) までとする。
- (9) 配列は 2 次元までとする。
- (10) 配列として使用する変数名は最大 15 個とする。
- (11) FORMAT 中の変換記号は X, H, A, I, F, E のみとする。

7. む す び

本 FORTRAN が一応完成してから 8 ヶ月ほどの間に、種々の大虫、小虫が見つけれられ、その都度コンパイラの改造を行って、現在はデバッグもほぼ完了したと思われる。

いままでの使用結果から次の諸点が明らかとなり、実用的には、まず満足のできるものであろう。

- (1) コンパイル速度
コンパイル速度は PTR の性能に負うところが多く、HIPAC103 のほぼ 3 倍である。
- (2) 実行速度 (乗, 除算オプションなし)
整形数演算の場合 HIPAC103 の 2 倍程度
実数形演算の場合 HIPAC103 と同程度
Fig. 24 のサンプル・プログラム (20 元連立方程式) の場合、演算時間 (入出力時間を含まない) は 45 秒である。

Fig. 24 Sample program

```
C SIMUL TANEIOUS LINEAR EQUATIONS
C BY GAUSS-JORDAN METHOD
C T. NIWA SEP. 6, '71
C
  DIMENSION A (20, 21)
  WRITE (6, 1000)
  READ (5, 1010) N
  N1=N+1
  READ (5, 1020) ((A (I, J), J=1, N1)
                  I=1, N)
  WRITE (6, 1030) N
  DO 500 I=1, N
500 WRITE (6, 1040) (A (I, J), J=1, N1)
  DO 100 I=1, N
  AA=0.
  DO 600 K=I, N
  AAB=ABS (A (K, I))
  IF (AAB--AA) 600, 600, 2
```

```
  2 AA=AAB
  L=K
600 CONTINUE
  AA=A (L, I)
  IF (AA) 3, 5, 3
  5 WRITE (6, 1060)
  STOP 9999
  3 DO 700 K=I, N1
  ALK=A (L, K)
  A (L, K)=A (I, K)
700 A (I, K)=ALK
  4 DO 200 J=1, N1
200 A (I, J)=A (I, J)/AA
  DO 300 K=1, N
  IF (I-K) 1, 300, 1
  1 AKI=A (K, I)
  DO 400 J=I, N1
400 A (K, J)=A (K, J)-AKI*A (I, J)
300 CONTINUE
100 CONTINUE
  WRITE (6, 1050) (I, A (I, N1), I=1, N)
1000 FORMAT (57H*** SOLUTION OF SIMUL
TANEIOUS EQUATIONS BY ELIMI
NATION **
  ¥*/)
1010 FORMAT (I4)
1020 FORMAT (F20.5)
1030 FORMAT(24HNUMBERS OF EQUATIONS
  N=, I3//14HCOEFFICIENTS ;/)
1040 FORMAT (E15.7, 4E16.7)
1050 FORMAT (/5X, 2HX (, 12, 3H )=, E14.7)
1060 FORMAT (22HNO UNIQUE SOLUTION
  !!!)
  END
```

- (3) システム入力に要する時間

PTR の性能とコンパイラのコンパクトさが大きく効いて、HIPAC103 の 1/10 以下にあたる 23 秒で、FORTRAN システム・テープを読み込むことができる。

- (4) コンパイル量

1 pass で実行することのできるソース・プログラムの大きさは最大 200 ステートメントであったが、これは IF と GO TO が多かったからで、普通は 100 ステートメントぐらいまで可能である。

今後の問題点としては、オブジェクト領域を小さくし

ないで、できるだけ文法の制限をゆるめ、拡張が容易なところは含める。すなわち、

- (1) COMMON 文, EQUIVALENCE 文を入れる。
- (2) FUNCTION 文を入れる。
- (3) 変数名を5～6字までにする。
- (4) 型宣言文を入れる。
- (5) 改行制御を可能にする。

などである。またハードウェアの拡張に伴う、FORTRAN システムの拡張を容易にすることも重要な課題である。

終りに、本FORTRANの開発に当り、種々ご便宜をはかっていただいた沖電気工業株式会社関係各位、ならびにプログラミングを手伝っていただいた丸永哲男氏と雁木秀一氏、また日頃適切なご助言をいただく名古屋大学理学部近藤一郎助教授の方々に深く感謝いたします。

参 考 文 献

- 1) 宇野, 和久田, 山崎, 栗田; 沖電気時報, 88, p. 1 (1971)
- 2) 岸田孝一; システム・プログラム入門, 日本出版協会
- 3) J I S ハンドブック (情報処理), 日本規格協会