

NIT-FORTRAN・2 を中心にした ミニコンのソフトウェア

丹羽 敏行・本田 行雄

電子工学教室
(1972年9月11日受理)

NIT-FORTRAN・2 Compiler and
Software System for Mini-Computer

Toshiyuki NIWA and Yukio HONDA

Department of Electronics
(Received Sept. 11, 1972)

There are many mini-computers for scientific or instructional use. Their software systems are the miniaturized ones of general computer systems, therefore they are inefficient and require troublesome operations. Then, NIT-system is designed and practiced. This software system is original and unique against the maker's.

NIT-FORTRAN・2 compiler has many merits, especially, high speed execution, the grammar over the level JIS 3000 and easy operation for the small memory (8kW~16kW) cpu.

New NIT-SAP assembler has many macros and NIT-FORTRAN・2 compiler is coded in this assembler. The assembled object program can be directly read into memory by simple IPL.

Key-in Loader is the loader which can read the hexadecimal-coded program from the keyboard and punch out the binary tape. This loader is also intended to be relocatable.

1. ま え が き

最近のミニコンの発展はめざましく、ここ2、3年の間に性能、価格とも大幅に改善され、より身近なものとなりつつある。とく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の伸びは著しい。

このような背景から、ソフトウェアの面では FORTRAN を中心にした各種のコンパイラ、アセンブラ、ローダ、ユーティリティなどのソフトウェア・システムが整備されつつある。しかし、研究用・教育用として、まだ不十分な点が多く、新しいハードウェアの発表時などはソフトウェアは皆無といってもよいぐらいである。筆者らの研究室にあるミニコンも70年3月の導入時にお

いては、不完全なアセンブラとローダが使用できるだけであった。

そこで、ソフトウェアに関してはメーカーとは完全に別な研究室独自の構想で出発し、最近 JIS 3000 レベルの FORTRAN コンパイラ NIT-FORTRAN を実用化し、¹⁾²⁾また今回改めて種々改良を加えた NIT-FORTRAN・2 を完成したりしてソフトウェア・システムもかなり充実したので、NIT-FORTRAN・2 と共にその概要を報告する。

2. ハードウェアの構成と仕様

cpu を中心にした筆者らの研究室のハードウェア構成を Fig. 1 に示す。

その仕様は次の通りである。この cpu の型名は

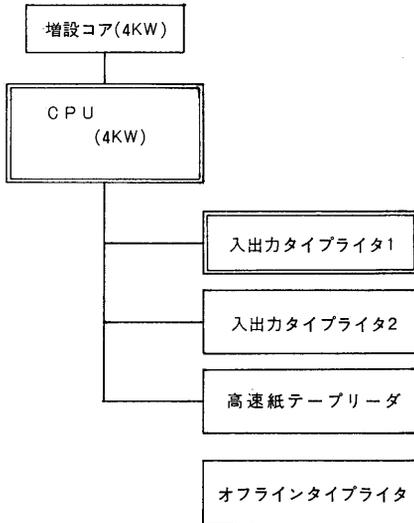


Fig. 1 Block diagram of NIT's hardware.

OKITAC-4300 A 型といい、その後発売されたS型、E型あるいは最近発表されたC型に比べて、オプションあるいは命令の数が少なかったり、演算速度が遅かったりしているが、基本的には同じである。

<cpu>	注：()内はC型
語長	16ビット+パリティビット
サイクルタイム	1.5 μ S (0.6 μ S)
記憶容量	4kW~16kW (4kW~65kW)
演算方式	並列2進, 固定小数点方式
命令数	39種 (44種)
アドレス方式	11/2方式, ページ方式
間接アドレス	指定可
インデックス・レジスタ	16ビット×1
割込み	4レベル
入出力制御	バス方式
乗除算機構	なし (標準装備)
IPL 機構	なし (オプション)

<入出力装置>	
入出力タイプライタ・1	1,000字/分 132字/行 活字 129種
入出力タイプライタ・2	500字/分 120字/行 活字 53種
高速紙テープリーダー	500字/秒

3. ソフトウェアの構成

Fig. 1 のハードウェア構成に対して準備されたソフトウェアを Fig. 2 と Fig. 3 に示す。ここで、メーカーは小型あるいは中型電子計算機のソフトウェア体系をそ

のまま持ち込んでいるが、筆者らのそれは操作性や能率を重視した結果独自の形となっている。2つの体系の特徴はすべてオブジェクト・プログラムの形から来ている。

すなわち、メーカーのオブジェクト・プログラムは変則16進数の形をとっているので、

- (1) オブジェクト・プログラムが長く、入出力に時間がかかる。
- (2) オブジェクト・プログラムをローディングするのに大きなローダを必要とする。
- (3) したがって、プログラムを格納する領域が大幅に制限される。
- (4) 操作が煩雑になる。

などの欠点がある。しかし、ローディング時に番地を割り付けるのでオブジェクト・プログラムをある程度リロケータブルな形にすることができ、ディスクなどの大容量な補助記憶装置があればオブジェクト・プログラムをテープの形でパンチ出力する必要がなくなり、上の(1),(4)は問題ならなくなる。

このようにメーカーのソフトウェアは主記憶の容量が大きく、大容量の補助記憶装置がついているシステムでないとひどく使い難いものとなる。

一方、筆者らのオブジェクト・プログラムは純2進の形をとっているため、上の(1)~(4)の反対の長所を持たせることができる。また、ハードウェアの方にリロケータブルなアドレス表現がないのでオブジェクト・プログ

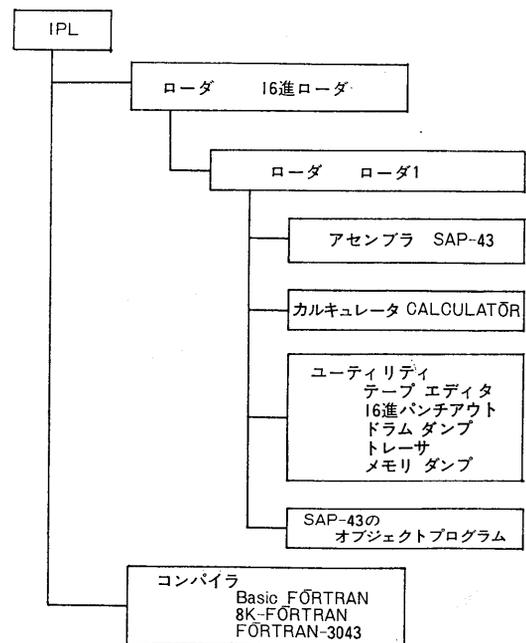


Fig. 2 Maker's software system.

ラムはリロケータブルにはなり得ないが、これは、FORTRAN コンパイラの作成などを例にとっても、アセンブル時の能率と操作性の良さで充分補われている。

3.1 メーカー提供のソフトウェア

メーカー提供のソフトウェアの主なものを Fig. 2 に示す。ここで、最も大きな特徴はオブジェクト・プログラムがリロケータブルな形になっていることであるが、頁を越えた間接指定などは不可能で完全なリロケータブルにはなり得ない。

3.2 名工大 OKITAC-4300 ソフトウェア

筆者らのソフトウェアの主なものを Fig. 3 に示すが、これらは全てメーカーのものとは異なり独自に開発したものである。その最も大きな特徴は操作性の良さとオブジェクト・プログラム入出力時の能率の良さであり、これはオブジェクト・プログラムが純2進の形をとっていることと、IPL がメーカーの16進ローダ程度の機能を有していることにある。

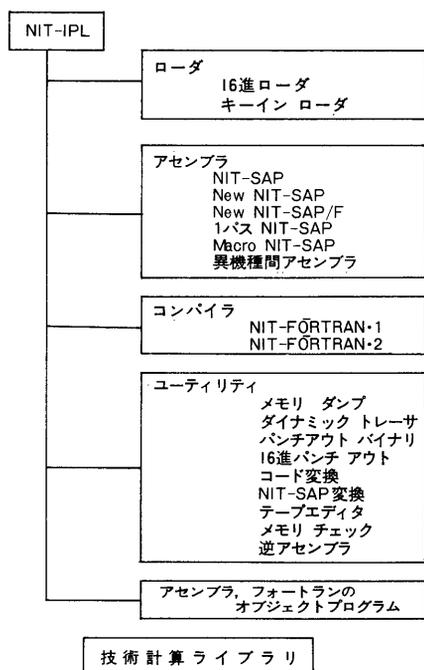


Fig. 3 NIT's software system.

4. ロ - ダ

ローダは、プログラムを記憶装置に格納するためのプログラムであり、大別して、

IPL (Initial Program Loader)

16進ローダ

の2種類がある。

すべてのシステムプログラム (アセンブラ, フォート

ラン, 16進ローダ), ユーティリティプログラム, およびアセンブラ, フォートルランのオブジェクトプログラムはこの IPL により入力することができる。IPL が壊れている場合は cpu パネルのスイッチを操作してマニュアルで記憶装置に入力しなければならない。IPL のプログラムの長さは31語である。

4.1 16進ローダ

テープにせん孔された16進言語のプログラムを記憶装置に格納するプログラムである。

<仕様>

- 格納先頭番地を最初に書く。
- 印字されないコード (スペース, 改行, シフトコード) は読みとばす。
- /……/ はコメントとみなし, 読みとばす。
- 文字定数は:○○:と書く。
- プログラム単位の終りはZで示す。
- 実行開始番地は最後にZ○○○○Zと書く。

格納番地がシステムサブルーチン中か, 記憶装置の容量を超えた場合, 16進数の個数が4の倍数でないとき, 16進数以外の文字がある場合, および: :の中が3文字以上のときはエラーとなる。

4.2 キーインローダ

タイプライタのキーボードから直接入力できる16進ローダであり, 次のコントロールコードに従って処理される。

¥LOC XXXX/

XXXX 番地より以下のプログラムを格納する。

¥PNC XXXX, YYYY, ZZZZ/

XXXX番地からYYYY番地までのプログラムをバイナリモードでパンチアウトし, 実行開始番地をZZZZ番地とする。XXXX, YYYYを省略すると, 直前の ¥LOC から今までに格納したプログラムを出力する。

¥RUN XXXX/

XXXX番地よりプログラムの実行を開始する。

<仕様>

- 格納先頭番地がXXXX0でない場合は, その位置までスペースを印字して見やすくする。
- 1命令毎にスペースを1つ印字する。
- 格納番地がXXXXFになれば自動的に改行する。
- /……/ はコメントとみなす。
- 文字定数は:○○:と書く。
- 訂正したい場合はその個所まで後退キーを押し, 打ち直せばよい。

16進ローダのエラーの場合に加えて, 上記以外のコントロールコードがあるとエラーになる。

5. アセンブラ

アセンブラにはベーシックアセンブラ，マクロアセンブラ，異機種間アセンブラの3種類があり，ベーシックアセンブラには1パス方式のものと，2パス方式のNIT-SAP，New NIT-SAP，およびシステム記述用のNew NIT-SAP/Fがある。これらとメーカ提供のアセンブラ（SAP-43）の比較を，Table 1に示す。

仕様上の大きな相違点は，SAP-43のオブジェクトプ

ログラムはローダ言語でリロケータブルであるが，NIT-SAPのすべてのオブジェクトプログラムは，純2進絶対番地形式のプログラムでIPLから直接入力できる点である。

また，操作はFig. 4(a)のようにNIT-SAPの方がはるかに簡単である。すなわち，SAP-43を用いてアセンブルし，実行するには，

- (1) IPLによりSAP-43システムを入力する。
- (2) パス1を実行する。

Table 1 Comparison between New NIT-SAP and other assemblers.

		SAP-43	1パスNIT-SAP	NIT-SAP	New NIT-SAP	Macro NIT-SAP
パス方式		2パス	1パス	2パス	2パス	2パス
オブジェクトプログラム		16進絶対番地	純2進絶対番地	純2進絶対番地	純2進絶対番地	純2進絶対番地
命令	機械語命令*	39	39	39	39	39
	制御命令	5	6	6	10	6
	疑似命令	12	—	—	42	—
	マクロ命令	—	—	—	14	33
定数	10語長 2語長 数 浮動小数点	そのまま書く Dを指定 そのまま書く	そのまま書く — —	そのまま書く — —	そのまま書く D “.....” そのまま書く	そのまま書く D : : そのまま書く
	16進数	$\wedge \pm h_1 h_2 h_3 h_4$	X : :	X : :	X “.....” H “.....”	X : :
	文字定数	: (2文字指定)	H : (2文字) :	H : (2文字) :	C “.....”	H : :
	アドレス定数	ラベル	ラベル	ラベル	ラベル ラベル ± n * ± n	ラベル
アドレス表現		ラベル リテラル	ラベル リテラル 16進数	ラベル リテラル 16進数	ラベル ラベル ± n * ± n 10進数 16進数	ラベル ラベル ± n 10進数 16進数
プログラムの記述		固定欄	固定欄	固定欄	固定欄	固定欄
ソースリスト		あり	—	—	指定可	—
エラーメッセージ		5	23	21	37(実行時10)	24

* 入出力命令を除く。

** SAP-43のアドレス定数はDEF命令のオペランドに書く。

他の定数はすべてDC命令のオペランドに書く。

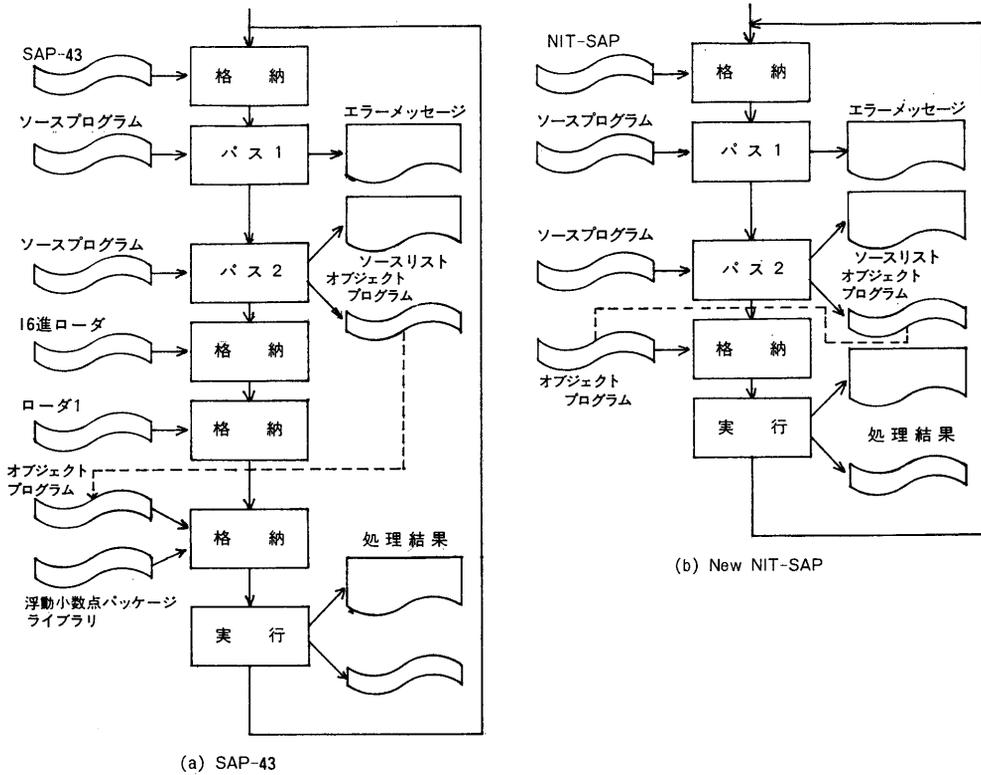


Fig. 4 Assembly operation of maker's assembler and NIT's.

- (3) バス 2 を実行する。
- (4) IPL により16進ローダを入力する。
- (5) 16進ローダによりローダ1を入力する。
- (6) ローダ1によりオブジェクトプログラムを格納する。
- (7) 必要があれば、浮動小数点演算パッケージ、ライブラリをローダ1により格納する。
- (8) プログラムを実行する。

という手順を踏まねばならず、しかも、各ステップで所定の番地に SCC をセットしなければならない。これに対し、(New) NIT-SAP を用いる場合は、Fig. 4 (b) のように、

- (1) IPL により (New) NIT-SAP システムを入力する。
- (2) バス 1 を実行する。
- (3) バス 2 を実行する。
- (4) IPL によりオブジェクトプログラムを格納する。
- (5) プログラムを実行する。

という手順で済み、SCC の値をセットするのは(1)の段階だけでよく、あとはスタートボタンを押すだけでよく、かなり使い易くなっている。また、エラーメッセージも豊富で親切である。

5.1 1パス NIT-SAP

ソースプログラムを1回通すだけでよいアセンブラである。したがってソースプログラムのすべての情報を記憶しておかねばならないので、ラベルテーブルはリスト構造になっている。オペレーションコードは機械語命令と1対1に対応しており、文字定数も1ステートメントに2文字(1語長)までであるなど、非常に基本的なアセンブラであるが、操作が簡単なので短いプログラムの時は便利である。

5.2 NIT-SAP

1パス NIT-SAP と同じ仕様の2パス方式のアセンブラである。

5.3 New NIT-SAP, New NIT-SAP/F

New NIT-SAP は NIT-SAP を拡張したもので、次の特徴をもつ。

- a 機械語命令の他に擬似命令を有する。
- b 1ステートメント中の文字定数の長さは58文字まで。
- c H変換(16進定数)を許す。
- d 制御命令(CAN, PAGE)が増えた。
- e ソースリストの指定ができる。
- f 複数個のプログラム単位をアSEMBルできる。

New NIT-SAP/F は、NIT-FORTRAN・2 コンパイラの記述用に New NIT-SAP を改造したものである。

5.4 Macro NIT-SAP

ハードウェア的に実行不可能なアクセス方法をした場合でも、これをマクロ命令に展開して実行できるようにした点が他のアセンブラと異なり、さらに、

- a 機械語命令の他に擬似命令を有する。
- b 2語長の論理演算、シフト演算命令がある。
- c 1ステートメント中の文字定数の長さに制限はなく、途中で改行コードがあってもよい。

という特徴をもつ。

5.5 異機種間アセンブラ

異なった機種のアセンブラ言語で書かれたプログラムを、その計算機で実行できる機械語命令に変換するアセンブラである。架空の計算機のシミュレートをしたり、より高いレベルの計算機を用いて能率よくアセンブルする時に用いる。現在あるものは、他のミニコンを想定したものである。

6. コンパイラ

最近のミニコンの普及はめざましく、研究用・教育用として多方面で活用され、FORTRAN コンパイラは不可欠のものとなっている。筆者らも、さきにミニコン用の FORTRAN コンパイラ NIT-FORTRAN を実用化し報告したが、今回改めて種々改良を加えたコンパイラを、NIT-FORTRAN・2 と名付け、デバッグもほぼ完了したのでその概要を述べる。

NIT-FORTRAN・2 の設計思想は NIT-FORTRAN のそれを引継いでおり、上向きの変換性を考慮しながら、教育用という観点から、MT 関係を除いて JIS 3000 を完全に満足するようにした。また、複素数演算、文字型データ、カナ文字なども取扱えるようにし、研究用としてもより使い易く、能率の良いものとなるようにした。本 FORTRAN の特徴を Table 2 に示すとともに以下に簡単に述べる。

(1) 機器構成

旧 FORTRAN が、主記憶 8kW、入出力タイプライタ、高速紙テープリーダー (オプション) というハードウェア構成しか考えていなかったのに対して、主記憶の容量および入出力装置の多様性に対して特別な操作を必要とせず簡単に使用できる。

(2) 翻訳と実行の手順

Compile & Go を原則とするが、ソースプログラムが大き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進のままで出力され、いつでも再入力し実行することができる。

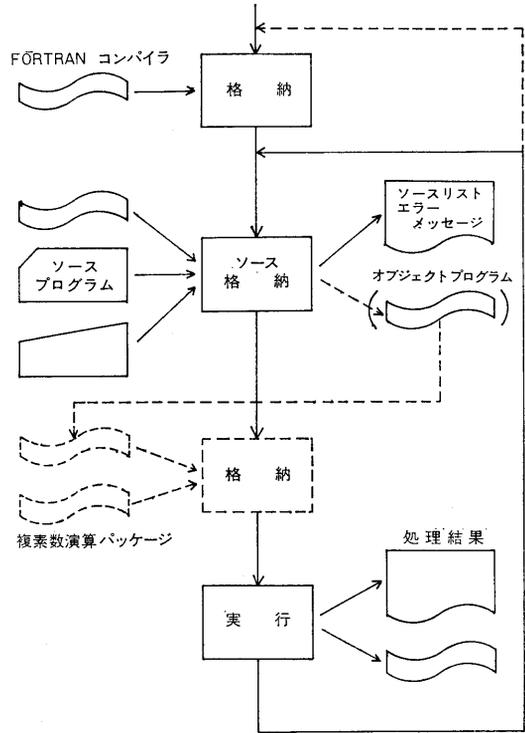


Fig. 5 "Compile and Go" operation of NIT-FORTRAN・2

コンパイル時		4K	8K	
IOCS,浮動小数点パッケージ 組込回数	コンパイラ	オブジェクト プログラム	各種テーブル	
実行時 (Compile & Go が可能な場合)				
IOCS,浮動小数点パッケージ 組込回数		オブジェクト プログラム	変数領域	
実行時 (2パスになった場合)				
IOCS,浮動小数点パッケージ 組込回数	複素数パッケージ	オブジェクトプログラム	変数領域	入出力バッファ 定数

Fig. 6 Memory allocation of NIT-FORTRAN・2

(3) オブジェクトプログラムの形

翻訳・実行の方法にはインタープリタ形式とトランスレータ形式がある。前者はコンパイラの構成が簡単になり記憶容量も比較的少なくてすむが、実行時間は長くなる。一方、後者はコンパイラの構成は複雑になるが、実行時間が短縮される。

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ためインタープリタ形式をとるのが普通であるが、本 FORTRAN ではあえて後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。

Table 2 Comparison between NIT-FORTRAN・2 and other compilers.

		JIS 3000	HIPAC 103 HARP 103	OKITAC-4300 NIT-FORTRAN	OKITAC-4300 NIT-FORTRAN 2
JIS 3000 から省いた 機能		—	引数のある SUBROUTINE FUNCTION COMMON EQUIVALENCE	FUNCTION COMMON EQUIVALENCE 補助入出力 書式なし READ, WRITE	補助入出力 書式なし READ, WRITE
JIS 5000 相当の機能	型宣言	—	—	—	INTEGER REAL COMPLEX
	行送り制御	—	—	—	JIS 5000 と同様印刷記 録の最初の文字による
	SUBROUTINE の 引数としての関数	—	—	—	文関数のみを許す
JIS 7000 相当の機能	文字データの入出力	—	—	A 変 換	A 変 換
	文字型定数	—	—	—	$nHh_1 \dots h_n$
	複素数 演算 データ 組込み関数 基本外部関数	—	—	—	JIS 7000 に準ずる
そ の 他	H変換, 文字データ 中のカナ文字	不 可	不 可	不 可	可
	文 番 号	1~9999	3~8191	1~65535	1~65535
	STOP n PAUSE n	0~7777 (8進)	—	0~65535 (10進)	0~65535 (10進)
	F,E 変換で E,F 型データの入力	不 可	不 可	可	可
	プログラム記述形式	固定欄	自 由 欄	固 定 欄	固 定 欄
	データの記述形式	固定欄	自 由 欄	固 定 欄	固 定 欄
	コンパイル時 ソース・リスト		不 可	不 可	可
	1 ステートメント ずつのコンパイル		可	可	可
	エラーチェックのみ		不 可	不 可	可
	入出力動作と演算の 併行処理		不 可	出力だけならば 可	可
	エラーメッセージ			COMPILE 93 RUN 14	COMPILE 58 RUN 14
機 器 構 成	最小構成		CORE 1kW, PTR DRUM 8kW, TTY	CORE 8 kW, TTY	CORE 8 kW, TTY
	最大構成		CORE 4kW, TTY DRUM 8kW, PTR LP, MT	CORE 8 kW, TTY PTR	CORE 16kW, TTY PTR, CR, LP, HTP
時 間	FORTRAN システム 入力		約 300sec	23 sec	25 sec
	コンパイル速度		1.5 行/sec	10 行/sec	10 行/sec
	実行速度 1 整数演算 2 実数演算 (20元連立方程式)		3.4 時間	1.4 時間 1.4 (46 sec)	1.0 時間 1.0 (33 sec)

(4) 文 法

本 FORTRAN のレベルは、補助入出力文と書式なし READ, WRITE 文を除いて完全に JIS 3000 を満足する。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 規格以外の記述を避けた。

JIS 3000 以上の機能で特筆すべきは、複素数演算と文字データおよびカナ文字の取扱いが可能なことである。

(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。

(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72, 実行時 23 と旧 FORTRAN よりもさらに数を多くし親切なものとした。

コンパイル時のエラーは4つのレベルに、実行時のエラーは3つのレベルに分類され、軽度のものに対しては演算の実行を続けることが可能である。

(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済むようにした。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およびソースプログラムのリストが自由に指定できるようにし、デバッグの便をはかった。また、ソースプログラムが大きくて2パスに切り換わる場合、エラーチェックのみを指定することもできる。

(8) そ の 他

数値データは整数・実数とも2語長32ビットで表わされるが、添字や DO 変数の値は1語16ビットの範囲で押え、実行速度の向上をはかった。

7. ユーティリティプログラム

ユーティリティプログラムはそれ自身で独立したプログラムであり、主にデバッグを行なう際に利用される。メモリダンプとダイナミックトレーサは、その性質上リロケータブルなプログラムとなっている。

7.1 ダイナミックトレーサ

あるプログラムを実行した時の、cpuの内部状態(レジスタなど)を1命令毎に印字するプログラムであり、トレースするプログラムが記憶装置上どこにあってもよいようにページ毎のリロケータブルになっている。

印字形式は、行数(10進数)、SCC、実行する命令、Cレジスタ、Aレジスタ、Bレジスタ、および予め指定した番地の内容を16進数で表わす。また、スイッチによ

り、

- a 全命令について印字する。
- b システムサブルーチンの中の命令だけを印字しない。
- c すべてのサブルーチンの中の命令を印字しない。

の3つの印字モードが指定できる。さらに、特定の番地から印字を開始したり、特定の番地から特定の番地の間の命令だけについて印字(スナップショット)したりすることも合わせて指定でき、デバッグの能率を上げることができる。

7.2 メモリダンプ

指定された番地の間の主記憶装置の内容を、その番地と共に16進数で印字するプログラムである。これも、ダンプする内容が記憶装置上どこにあってもよいようにBレジスタを用いた完全なリロケータブルプログラムとなっている。また印字された内容を見やすいように、印字用紙1枚に記憶装置1ページ分(256語)を印字する。

7.3 メモリチェック

記憶装置内にあるプログラムと、そのプログラムの純2進のテープとを比べて、異なっている番地の内容をその番地と共にプリントするプログラムである。たとえば、あるプログラムを実行する前と実行した後の記憶装置の内容の違いについて知りたい時などに使用する。

7.4 テープエディタ

テープにせん孔された長いプログラムの途中で修正するのは不便である。これを解消するために作られたのがテープエディタである。修正すべき個所の直前のラベルと、そのラベルから何行目から何行目までを修正するかを指示するテープを作り、挿入、削除、置き換えの修正を行なうことができる。

修正指示の例

```
置換... ABC 00, 0000000
      ABC DC X "FF"
```

```
削除... C 02, 000D 01, 00
```

```
挿入... 000D 01, 000E 05, 03
```

```
DA 2
```

```
C 01 DC -1
```

```
C 02 DC X "2A"
```

```
修正... / / / / /
終り
```

7.5 逆アセンブラ

純2進のテープをもとにして、それに相当するアセンブラ言語のプログラムを出力するプログラムである。逆アセンブラで問題となるのは、純2進の値を見ただけではそれが命令語か数値語かまたはその両方で命令を定数として利用しているのか判然としないことである。実際

Table 3 Scientific program library.

名 称	記 述 言 語	精 度 有効数字	備 考
マルチカーブプロッタ	NIT-FORTRAN		9本まで
グラフィック	NIT-FORTRAN		$y=f(x)$, y の値をプロット
第1種ベッセル関数	NIT-FORTRAN	10^{-6}	$J_n(x)$ $ n \leq 30000$ $ x \leq 30000.0$
第2種ベッセル関数	NIT-FORTRAN	10^{-6}	$Y_n(x)$ $ n \leq 30000$ $ x \leq 30000.0$
第1種球ベッセル関数	NIT-FORTRAN	10^{-5}	$j_n(x)$ $0 \leq n \leq 30000$ $0 \leq x \leq 30000.0$
ガンマ関数	NIT-FORTRAN	6桁	$\Gamma(x)$, $ x \leq 10^{99.00}$
連立一次方程式	New NIT-SAP	6桁	実係数または複素係数, 30元まで
代数方程式	NIT-FORTRAN	5桁	ヒッチコック・ベアスター法
数値積分	NIT-FORTRAN	5桁	シンプソン則
正規分布の逆関数	NIT-FORTRAN	10^{-6}	
誤差関数	NIT-FORTRAN	10^{-6}	

にはその番地が、他の命令で参照されていれば定数とみなして逆アセンブルする。

操作は3パス方式であり、3パス目に指定によってアセンブルリストだけ、ソーステープだけ、あるいはその両方を得ることができる。

使用例としては、オブジェクトテープだけ残ってソーステープを紛失した場合、もとのアセンブラと異なった仕様のアセンブラ言語のソースプログラムが欲しい場合（もちろんこの場合は、その仕様に合った逆アセンブラを作っておく必要がある。）があり、とくに後者の方が利用価値が高い。現在 New NIT-SAP 用と、SAP-43用の2種類ある。

7.6 コード変換

オフライン・タイプライタは HIPAC 103 用のコードであり、これでパンチしたテープをフォートラン、アセンブラで使えるように ISO コードに変換するプログラムである。

7.7 NIT-SAP 変換

NIT-SAP と New NIT-SAP では仕様が一部異なり、NIT-SAP で : で記述した部分が New NIT-SAP では"となったので、:を"に変換するプログラムである。

7.8 パンチアウトバイナリ

記憶装置内のプログラムを、バイナリモードでパンチアウトし、IPL で直接入力できる形にするプログラムである。これは、16進言語で書いたプログラムを純2進にしたい場合や、アセンブラのオブジェクトプログラムを記憶装置上でマニュアルで修正して保存したい場合に用いる。

8. ライブラリ

ライブラリとして、Table 3 に示すプログラムが用意されている。

9. む す び

以上、ミニコンを導入してから2年ほどの間に開発してきたソフトウェアの主なものを体系化してみたが、現在のハードウェア構成ではほぼこれが限界であろうと思われる。

しかし、問題によっては FORTRAN とアセンブリ言語あるいは機械語とのリンクが必要となってくるので、そのためには、リロケータブルな機能を有するロード、アセンブラの開発が今後の課題となるであろう。また、教育用として計算機実習を行なうとすれば、ドラム、ディスクなどの補助記憶装置やラインプリンタ、カードリーダーなどの入出力装置の充実が必要である。

終りに、ハード、ソフト共に種々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卒研生諸君、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。

文 献

- 1) 丹羽・本田；昭46 電学会東海支連大論文集
- 2) 丹羽・本田；名工大学報 23 (1971) P.349
- 3) 丹羽・本田；昭47 電学会東海支連大論文集
- 4) 日本規格協会；JIS ハンドブック (情報処理)
- 5) 共立出版；bit 臨時増刊 1972—12