The University of Osaka
Institutional Knowledge Archive

Title	SX FORTRAN77概要(1) : 文法を中心として
Author(s)	大中, 幸三郎; 後藤, 米子
Citation	大阪大学大型計算機センターニュース. 1986, 61, p. 17-35
Version Type	VoR
URL	https://hdl.handle.net/11094/65691
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

資　料

SX FORTRAN77 概要（1）

―文法を中心として―

大阪大学大型計算機センター研究開発部
大中幸三郎・後藤米子

1. はじめに

当センターでは、FORTRAN77で書かれたプログラムをより高速に処理するために、HFPにかわってベクトル計算機SX-1（以下SXと略す）を新規に導入します[1,2]。SXはHFPと同系列のアーキテクチャの計算機であり、独立したシステムとしても使用できる計算機です。しかしながら、当センターではSXをACOSシステム1000（以下ACOS 1000と略す）のバックエンドプロセッサとして接続し、HFPの場合と同様に、利用者からはACOS 1000の一つのサブシステムと見えるように運用します[2,9]。HFPからの移行に際してはプログラムの変更などはほとんど不要ですが、ACOS 1000からの移行に際しては、SXとACOS 1000の運用およびアーキテクチャが異なるために、利用法の差異やプログラムの書換えなどの変更作業が必要となります[3,7~10]。

SXの利用形態は、大別してACOS 1000に準拠した利用法（以下簡易形という）とSX固有のジョブ制御言語などを用いる利用法（以下基本形という）の二種類に分けられますが[2,9]、通常は簡易形の機能で充分なものと思われます。本稿ではSXの簡易形におけるSX FORTRAN77（以下SX77と略す）[7~9]とACOS 1000のFORTRAN77のVモード（以下AF77Vと略す）[14,15]との主要な差異について、文法を中心として述べます。記述の順序は、FORTRAN77概説[11~13]およびHFP FORTRAN77概要[16,17]と同一ですから、これらの内容および速報の関連事項と比較すると、より理解しやすいと思います。なお、本稿ではベクトル化処理機能の詳細[4,8]および日本語情報処理機能に関する事項については省略しますから、了承して下さい。

2. システム構成と処理の流れ

2.1 システムの構成

FORTRAN77の利用者から見ると、ACOS 1000は1語（＝4バイト）が36ビットのワードマシンであり、処理形態もVモードとRモードの区別があった。一方、SXは1バイトが8ビットのバイトマシンであり、処理形態にVとRの区別は存在しない。また内部コードもAF77VではJISコード、SX77ではEBCDICカタカナコードであり、異なっている[9]。当センターのシステム構成はすでに述べられているが[2]、その概略を図1に示す。

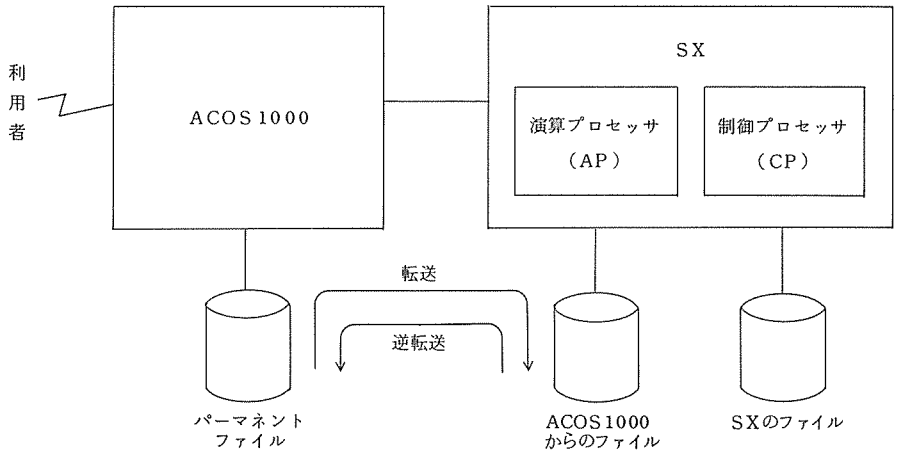


図1　システム構成の概略

図1に示すように、SXは演算プロセッサ(以下APと略す)と制御プロセッサ(以下CPと略す)からなり、機能の分散がはかられている。APはFORTRAN77で書かれたプログラムの超高速実行をバッチ処理で行う専用機であり、いわゆるベクトルプロセッサである。CPはシステムの制御を行う以外に、APで実行すべきプログラムの各種支援に使用し、通常の汎用機に近いプロセッサである。FORTRAN77の利用者から見たCPの用途としては、

プログラムのコンパイル、リンクおよびデバッグ

APで実行しているプログラム中の入出力文の処理

ファイルの作成、削除などのファイルメンテナンス

などがあげられる。CPでの処理には、TSS処理とバッチ処理の双方が使用できるが、ACOS 1000のIAPに相当する機能はない。

2.2　FORTRAN77コンパイラ

SXのアーキテクチャはHFPと同系列ではあるが、APの高速性を追求するために、APとCPの間には少し差異がある。したがってSX77においても、APで実行するための目的プログラムを作成するコンパイラと、CPで実行するための目的プログラムを作成するコンパイラの二種類が準備されている。これらの二つのコンパイラの間にはアーキテクチャを反映して少し差異があるが、後者の仕様はHFPのFORTRAN77[16~18]とほとんど同一である。

本稿では特に記述しないかぎり、SX77のAPとCP共通の仕様を示し、仕様が異なる場合には「SX77 -AP」、「SX77 -CP」と記述する。

2.3　ファイルとその処理の流れ

FORTRAN77のソースプログラムは、INCLUDE文のメンバーを除き(**15.1**参照)、ACOS 1000の磁気ディスク上に存在しなければならない。このソースプログラムはSXに自動的に転送

される。SXの磁気ディスク上のソースプログラムを用いることは、1986年秋ごろに可能となる予定である。

プログラムの実行時に使用する外部ファイルの処理は、磁気ディスク上になければならないが、その取扱いはパーマネントファイルとテンポラリファイルで異なっている(**11.2**参照)。パーマネントファイルはACOS 1000およびSXのいずれにあってもよいが、ACOS 1000 の磁気ディスク上の場合には、順編成書式付きのみが使用できる。このパーマネントファイルは、SXでのプログラムの実行開始以前にSXに自動的に転送される[6,9]。転送に際して内部コードの変換が行われ、SXの磁気ディスク上にファイルが作られる。そのファイルを用いてSX上のプログラムの実行を行い、SXでの実行終了時にACOS 1000から転送されてきたファイルは削除される。ただし、ファイルの内容が更新された可能性のあるものは、削除以前に転送時とは逆の変換を行ってSXからACOS 1000に逆転送される。テンポラリファイルは、ACOS 1000からの指令により、プログラムの実行開始以前にSXの磁気ディスク上に作成され、実行終了時に削除される。

3. 文字と文

3.1 FORTRAN文字集合と英字名

英　字	A～Z および $	
数　字	0～9	
特殊文字	空白　＝　＋　－　＊　／　(　)　,	
	.　&　'　:　;　"　<　>　下線	< 以下の3つが追加となる

SX 77では、コンパイラオプションにDOLLARを指定することにより、$を特殊文字として扱うことができる。AF 77Vのコンパイラオプション DLRと同一の機能である。

英字名は、英字で始まる1～8文字の英字、数字および下線の列である。AF 77 V では下線は使用できなかった。

3.2 文と行

SUBCHECK文(**7.4**参照)の場合を除いて、文と行の順序はSX 77とAF 77Vの間で差異はない。

ベクトル化指示行(**15.3**参照)と区別するために、注釈行およびデバック行に＊VDIRで始まるものを書いてはならない。

文を桁にとらわれずに自由形式で記述する場合には、AF 77Vとの互換性を考慮し、コンパイラオプションFREE2が用意されている。ただし、この場合においても記録の終りの取扱いは異なる。

4. データの型

4.1 算術データ

ACOS 1000が1バイト＝9ビットに対し、SXでは1バイト＝8ビットとなっているのに対応して、それぞれの型で表現できる範囲がせまくなっている。

4バイト整数型の場合には、-2^{35}～$2^{35}-1$から-2^{31}～$2^{31}-1$に、2バイト整数型の場合には、-2^{17}～$2^{17}-1$から-2^{15}～$2^{15}-1$と表現範囲が変更される。SX 77ではAF 77と異なり、特にコンパイラオプションを指定しなくても、2バイト整数型を使用することができる。

浮動小数点数の場合の表現範囲と表現桁数を10進換算で**表1**に示す。

表1 浮動小数点数の表現範囲と表現桁数

	表現範囲（絶対値）	表現桁数（10進）		
		単精度	倍精度	4倍精度
SX77	0, 10^{-78}～10^{75}	6.3	15.9	32.8
AF77V	0, 10^{-155}～10^{152}	7.2	18.0	36.1

内部表現の形式は、浮動小数点表現のみが異なっている。すなわち、AF 77Vでは、負は指数部、仮数部ともに2の補数表現となっていたが、SX 77では指数部はいわゆるげたばき表現、仮数部は絶対値表現となっている。

定数の桁数により、指数を示す文字DおよびQがなくても倍精度あるいは4倍精度として扱う機能は、SX 77ではAF 77Vと異なり、コンパイラオプションCONSTEXTに依存する。

4.2 論理型

4バイト論理型および1バイト論理型の内部表現は、値が真のときは全ビットが1，偽のときは全ビットが0である。AF 77Vでは、右端のビットのみが1のとき真、全ビットが0のとき偽であった。SX 77ではAF 77Vと異なり、特にコンパイラオプションを指定しなくても1バイト論理型を使用することができる。

4.3 文字型

文字型データの長さは、定数では1～256文字、定数以外では1～32,767文字である。AF77Vではいずれの場合にも511文字以下であった。また、内部表現はSX 77ではEBCDICカタカナコードであり、AF 77VのJISコードとは異なっている。ただし、AF 77Vにおいても次回のバージョンアップ時に、定数以外は1～32,767文字となる予定である。

4.4 2進型、8進型および16進型

2進型と8進型は1バイトのビット数の変更により、書くことができる桁数が短くなる。また、8進型を文字型に対応づけることはできない。16進型はAF 77Vになかった型であるが、その内

容については省略する。

4.5　2バイト整数型と1バイト論理型に対する領域拡張

2バイト整数型と1バイト論理型に対する領域拡張とは、これらの型として宣言されていても、4バイト整数型および4バイトの論理型とみなして処理を行う機能である。SX77−APでは特にコンパイラオプションを指定しないかぎり、実行時の効率を高めるために領域拡張を行っている。SX77−CPにはこのような機能はない。

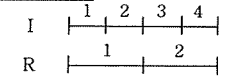
EQUIVALENCE文またはCOMMON文により、2バイト整数型あるいは1バイト論理型が他の型と結合されているときには、領域拡張によってその結合関係が変わる。これは他の型に対して領域拡張の考慮をしないためである。その例を次に示す。このようなプログラムに対しては、領域拡張を抑止するためにコンパイラオプションNOINLOG4を指定するか、ソースプログラムを変更しなければならない。

特定のプログラム単位を領域拡張した場合には、共通ブロックや引数などによりそのプログラム単位と領域を共有しているプログラム単位も、領域拡張を考慮しなければならない。また、入出力文は領域拡張後の型に従って動作するから、記録の長さ（**11.1**参照）などに十分注意する必要がある。

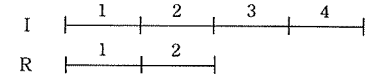
結合のための宣言文

```
INTEGER*2   I(4)
REAL        R(2)
EQUIVALENCE (I,R)
```

領域拡張前



領域拡張後



5.　配列および部分列

配列を参照するときの添字式の値は、その配列を宣言したときの対応する寸法の下限と上限の範囲内になければならない。

配列および文字部分列を参照するときの添字式と文字位置式は、整数型、実数型、倍精度実数型または4倍精度実数型の算術式である。

6.　式

6.1　関　係　式

関係演算子に右のものが追加される。それぞれ、従来から用いられている演算子と等価であり、表記法のみが異なっている。

追加される演算子	等価な演算子
<	.LT.
<=および=<	.LE.
>	.GT.
>=および=>	.GE.
<>および><	.NE.
=	.EQ.

文字式の値を比較する場合の大小順序は、EBCDICカタカナコードによって行われる。ただし、コンパイラオプショ

ンにCOLLATE＝JISを指定すると、AF77Vの場合と同様に、JISコードによって行われる。

6.2 論 理 式

論理演算子に .XOR. が追加される。その意味は排他的論理和であり、従来から用いられている .NEQV. と等価であるが、優先順位は .NEQV. より高く、 .OR. と同一である。

7. 宣 言 文

7.1 EQUIVALENCE文

EQUIVALENCE文によって、文字型と文字型以外の型を結合することができる。

AF77Vでは、倍精度実数型、4倍精度実数型、複素数型、倍精度複素数型および4倍精度複素数型のデータは、最初の語が偶数番地となるように記憶されていた。SX77においても共通ブロック以外の場合には、データの型に応じて、最初のバイトが下記の境界にとられる。ただし、番地のつけられている単位がACOS 1000では語、SXではバイトとなっていることに注意を要する。

2の倍数番地	2バイト整数型
4の倍数番地	4バイト整数型、実数型、複素数型、4バイト論理型
8の倍数番地	倍精度実数型、4倍精度実数型、倍精度複素数型、4倍精度複素数型
任意の番地	文字型、1バイト論理型

共通ブロック以外のデータに対して、EQUIVALENCE文による結合により上記の境界を満足しえない場合には、EQUIVALENCE文が優先される。ただし、ベクトル化処理機能は上記の境界を満足していることを仮定して実行しているから、十分な注意が必要である。

7.2 COMMON文

一つの共通ブロック内に文字型と文字型以外の型を混在させることができる。

共通ブロックの先頭のバイトは必ず8の倍数番地に割当てられる。コンパイラオプションがALCの場合には、実行効率を高めるために**7.1**に述べた境界となるように、ダミーのバイトが挿入される。ダミーの記憶領域の挿入される位置と長さは、SX77とAF77V では異なっている。この場合に、ダミーの記憶領域を挿入されたくなければ、共通ブロック中の順序を考慮しなければならない。その例をSX77の場合で示すと、8の倍数番地から始るもの、4の倍数番地から始るもの、2の倍数番地から始るもの、任意の番地から始るものの順に並べることである。

コンパイラオプションがALCの場合において、共通ブロック中のデータに対するEQUIVALENCE文により、**7.1**に述べた境界を満足しえなくなればエラーとなる。コンパイラオプションがNOALCの場合には、**7.1**に述べた境界を満足しなくてもよいが、ベクトル化処理機能は境界を満足していることを仮定して実行しているから、十分な注意が必要である。

7.3　型宣言文とIMPLICIT文

文字型以外の型宣言文とIMPLICIT文において指定できる長さ指定の差異を表2に示す。

表2　型宣言における長さ指定と宣言される型

型　宣　言	長さ指定（＊l）			宣言される型
	SX77	AF77V（コンパイラオプションに依存）		
		ALIGN（既定値）	NALIGN	
INTEGER＊l	2	対応機能なし	1，2	2バイト整数型
	4，省略	＊lの指定を無視	$3 \leqq l$	4バイト整数型
REAL＊l	4，省略	$1 \leqq l \leqq 7$，省略	$1 \leqq l \leqq 7$，省略	実数型
	8	$8 \leqq l \leqq 15$	$8 \leqq l \leqq 15$	倍精度実数型
	16	$16 \leqq l$	$16 \leqq l$	4倍精度実数型
COMPLEX＊l	8，省略	$1 \leqq l \leqq 15$，省略	$1 \leqq l \leqq 15$，省略	複素数型
	16	$16 \leqq l \leqq 31$	$16 \leqq l \leqq 31$	倍精度複素数型
	32	$32 \leqq l$	$32 \leqq l$	4倍精度複素数型
LOGICAL＊l	1	対応機能なし	1	1バイト論理型
	4，省略	＊lの指定を無視	$2 \leqq l$	4バイト論理型

7.4　SUBCHECK文とSUBSTRCHK文

SX77ではデバックのためにSUBCHECK文とSUBSTRCHK文の二つの宣言文が追加されている。これらの二つの文はコンパイラオプションがDBGの場合に有効となる。機能としては、コンパイラオプションCHECKと類似であるが、部分的な指定ができることおよび最適化機能との整合性[3)]が異なっている。

SUBCHECK文は、配列要素の参照において添字の値が許容範囲にあるかどうかを検査するための文であり、次の形である。

SUBCHECK（n_1, n_2）a_1, a_2, ・・・　ただし、n_1とn_2は文番号、a_iは配列名を示す。

上記の文は、文番号n_1をもつ文から文番号n_2をもつ文までの間において、配列a_iが検査の対象となることを示している。n_1を省略するとプログラム単位の先頭、n_2を省略するとプログラム単位の最後を指定したものとみなされる。n_1とn_2を共に省略するときには、（n_1, n_2）全体を省略してもよい。a_iをすべて省略した場合には、そのプログラム単位中のすべての配列名を指定したものとみなされる。SUBCHECK文の使用上の注意事項を次に示す。

- ○　配列名を指定した場合には、その配列に対する配列宣言文より後になければならない。
- ○　同一の配列名が複数のSUBCHECK文に指定された場合は、最後のものが有効となる。
- ○　擬寸法配列に対しては、添字の値がゼロ以下かどうかだけを検査する。

SUBSTRCHK文は、文字部分列の参照において文字位置式の値が許容範囲にあるかどうかを検査するための文であり、次の形である。

SUBSTRCHK

SUBSTRCHK文はSUBCHECK文と異なり、プログラム単位中のすべての文字位置式が検査の対象となる。

8. DATA文

初期値の並びに16進定数を書くことができる。AF77Vには16進定数そのものがなかった。

拡張領域[3)]に割付けられたデータに対して初期値を与えることはできない。AF77Vには拡張領域という概念がなかった。

初期値の設定される要素が配列名でかつ名前の並びの最後に現れている場合には、必ずしも配列要素の数だけの初期値を表す定数または定数名が与えられなくてもよい。この場合は、対応する初期値をもつ配列要素のみに対して初期値が設定される。AF77Vにはこのような機能はなかった。

9. 代 入 文

SX77とAF77Vの間にはとくに差異はない。

10. 制 御 文

10.1 GO TO 文

計算形GO TO文において、算術式の型は、整数型、実数型、倍精度実数型または4倍精度実数型である。

文番号の並びを指定した割当て形GO TO文において、実際の分岐先がその並びの中に含まれていない場合の処理は、SX77とAF77Vで異なるが、その詳細については省略する。

10.2 IF 文

SX77-CPにおいて、算術IF文、論理IF文、ブロックIF文およびELSE IF文での分岐の条件を決定する式の演算中にオーバーフロー、アンダーフローまたはゼロ除算が発生すると、IF文による分岐の処理結果は保証されない。この処理方式はHFPと同一である。SX77-APにおける処理はAF77Vと同一であり、SX77-CPおよびHFPとは異なっている。

10.3 DO 文

DO文において、初期値、終値および増分を示す算術式の型は、整数型、実数型、倍精度実数型または4倍精度実数型である。

10.4 STOP文とPAUSE文

STOP n およびPAUSE n において、n が整数の値をとる場合には $0 \sim 2^{31}-1$、文字の値をとる場合には、STOP文では170文字以下、PAUSE文では176文字以下である。AF 77Vでは、$0 \sim 10^5-1$ および40文字以下であった。

11. 入出力文

11.1 FORTRAN記録の長さ

書式つき記録の長さの算出方法は、SX 77とAF 77Vで同一であり、文字の個数（バイト数）によって数える。

書式なし記録の長さの算出方法は、4バイトを1語とする語の個数によって数える。その算出式は、SX 77ではAF 77Vと異なり、次式となる。

$$\text{書式なし記録の長さ} = \mathrm{INT}((\sum_i n_i + k_l + 3)/4) + \sum_j \mathrm{INT}((k_j + 3)/4)$$

ただし、n_i, k_l, k_j はそれぞれ次のものを示す。

n_i　記録中の連続する文字型以外の要素の i 番目の集まりのバイト数。

k_l　記録の最後の要素が文字型の場合には、その要素を含む連続する文字型の要素のバイト数。記録の最後の要素が文字型以外の場合にはゼロ。

k_j　直後に文字型以外の要素が続いている連続する文字型の要素の j 番目の集まりのバイト数。

書式なし記録の長さは上記の通りであるが、実際にはSX 77の入出力ルーチンがファイルシステムのヘッダとは別に、ファイルの形式に応じて高々5バイトのヘッダを付加している[8)]。このヘッダの長さは書式なし記録の長さには入らないが、ファイルを作成するときには考慮が必要である。

11.2 ファイル編成

SX 77では、システムの構成、コード体系および運用上の制約などにより、外部ファイルとして

表3　SX 77で使用可能なファイル編成

		パーマネントファイル		テンポラリファイル
		ACOS 1000	SX	
順編成	書式つき	可	可	可
	書式なし	不可	可	可
直接編成注) 直編成	書式つき	不可	可	可
	書式なし	不可	可	可

注）ACOS 1000における直接編成に対応するものをSXでは直編成という。

は磁気ディスクファイルのみが使用出来る。パーマネントファイルはSXとACOS 1000のいずれにあってもよく、テンポラリファイルはSX上に作成される。使用できるファイル編成を表3に示す。

11.3 接　続

SX 77では、プログラムの実行中に使用する外部ファイルの接続に強い制限がある。すなわち、パーマネントファイルに対しては、そのファイルがACOS 1000上に存在すれば事前の接続を行う必要があり、SX上に存在すれば事前の接続を行ってはならない。テンポラリファイルに対しては、SX 77-CPのTSS処理の場合を除き、事前の接続を行う必要がある。SX 77-CPのTSS処理では、OPEN文によるテンポラリファイルの作成、接続も可能である。

事前の接続の中で、システムで自動的に定められている関係を表4に示す。外部装置識別子5，6および7に対して、ファイルを割当てることも可能であるが、その場合に使用可能な入出力動作には表5の制限がある。ただし、SX 77 では外部装置識別子6と7は特別扱いの処理がなされており、通常は利用者には無意味なヘッダ等が出力される。したがって、これらをファイルに割当てることはさけるべきである。

接続を行う場合のファイル属性などの優先順位は以下の通りである[10]。

高位　ファイルの属性（ファイルラベルの情報）
↓　事前の接続による指定
↓　OPEN文による指定
低位　既定値（外部装置識別子に依存）

表4　既定値による事前の接続

	SX 77	AF 77V
カード読取装置	5	5，41
ラインプリンタ	6	6，42
カード穿孔装置	7	7，43
端末 入力	5	5，41
端末 出力	6	6，7，42，43

表5　外部装置識別子5～7に対して使用可能な入出力動作

	5	6	7
READ	可	不可	不可
WRITE	不可	可	可

11.4 READ文とWRITE文

SX 77ではAF 77Vと比較して、下記の点が異なる。

- ○ 外部装置識別子の値は1～99であり、整数型、実数型、倍精度実数型および4倍精度実数型の算術式である。ただし、AF 77Vにおいても次回のバージョンアップ時に、1～99となる予定である。
- ○ 記録指定子の右辺の型についても、前項と同一である。
- ○ 誤り条件が発生したときに、入出力状態指定子の右辺に返却される値が異なる。

11.5 OPEN文

SX 77ではAF 77Vと比較して、下記の点が異なる。

○　SX77−CPのTSS処理の場合を除いて、ファイルを作成することはできない。

○　直接探査で入出力する場合は、事前の接続を行っていてもOPEN文の実行が必要である。

○　事前の接続を引き継ぐためのOPEN文において、FILE指定子を省略した場合には、STATUS指定子も省略しなければならない。

○　外部装置識別子および入出力状態指定子については**11.4**と同様である。

○　FILE指定子の右辺の値には、SX上のファイル記述を書かねばならない。ACOS 1000上のファイルはプログラムの実行開始以前はSXに転送されるが、転送後にSX上に作られたファイル記述はINQUIRE文によってのみ知ることができる。

○　RECL指定子は、すでに存在しているファイルを順番探査で接続する場合は、意味をもたず無視される。

○　MAXREC指定子を省略すると1000記録分のファイルが作成される。

FILE指定子で指定するSX上のファイル記述は、SXのファイル管理[5,6]とは少し異なっており、簡単に示すと次のようになる。

完全修飾名　　利用者番号 . d_1 . d_2 . ・・・・ . d_m . ファイル名

部　分　名　　. d_1 . d_2 . ・・・・ . d_m . ファイル名

ただし、d_i はディレクトリ名である。

部分名はSX77−CPのTSS処理においてのみ使用でき、そのジョブを実行している利用者番号が付加されて完全修飾名となる。ACOS 1000のファイル記述と形式的に対応づければ、右に示す関係となる。

SX（FILE指定子）		ACOS 1000
利用者番号（6桁）	↔	利用者番号（6桁）
ディレクトリ名	↔	カタログ名
ファイル名	↔	ファイル名
.	↔	／
（対応機能なし）	↔	パスワード

11.6　CLOSE文

SX77ではAF77Vと比較して、下記の点が異なる。

○　SX77−CPのTSS処理の場合を除いて、ファイルを削除することはできない。

○　事前の接続を行っているテンポラリファイルに対して、STATUS指定子を省略またはSTATUS＝'KEEP'としたCLOSE文を実行すれば、事前の接続にどる。この場合には、その装置を異なるファイルに接続するOPEN文を実行した時点で、事前の接続を解除する。

○　外部装置識別子および入出力状態指定子については**11.4**と同様である。

11.7　INQUIRE文

SX77ではAF77Vと比較して、下記の点が異なる。

○　FILE指定子については**11.5**と同様である。

○　外部装置識別子および入出力状態指定子については**11.4**と同様である。

○　FORMATTED指定子およびUNFORMATTED指定子の意味は、そのファイルの許容形に含まれているか否かであり、探査法には依存しない。

○　ファイルによるINQUIRE文において、そのファイルが存在しなければNAMED 指定子に返却される値は不定となる。

○　NAME指定子に返却される値は、完全修飾名（**11.5**参照）または不定である。

○　事前の接続を行っている場合において、OPEN文あるいは実質上の入出力を一度も実行していない場合には、FORM指定子、ACCESS指定子およびBLANK指定子に返却される値は不定となる。

11.8　高速入出力機能

SX77では、大量のデータの入出力を高速に処理するために、非同期入出力機能、並行入出力機能および拡張記憶入出力機能が追加されている。しかしながら、拡張記憶入出力機能は当センターのハードウェア構成では使用できず、並行入出力機能も運用上の制約のために使用する利点はない。したがって、以下では非同期入出力機能についての概略を示す。

非同期入出力機能は、1つのプログラム中における入出力処理と演算処理の並列実行を可能とし、CPとAPの機能分散方式を積極的に用いるものである。この機能は、順番探査書式なし入出力に対してのみ使用できるが、SX77-CPではその効果は少ない。非同期入出力機能を用いるには、非同期READ文、非同期WRITE文およびWAIT文を使用する。

READ　（*cl*）*iol*
WRITE（*cl*）*iol*　｝データの転送を開始させるための文

WAIT　（*cl*）　　データの転送の完了を待合せるための文

ただし、*cl*は制御情報並び、*iol*は非同期入出力並びを示す。

非同期入出力処理の流れの例を図**2**に示す。

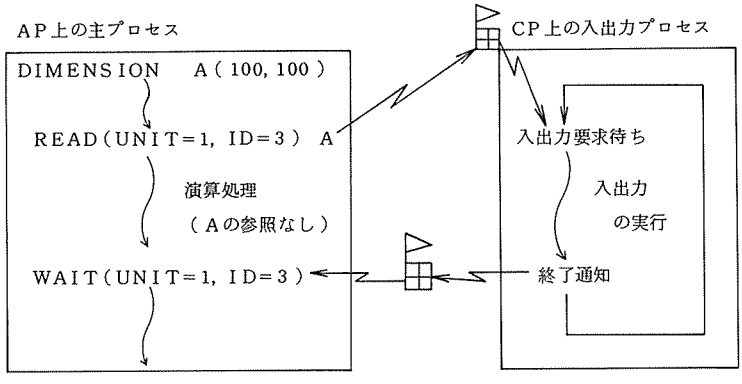


図2　非同期入出力処理の流れ

制御情報並びおよび非同期入出力並びの内容は、通常の入出力文の場合と異なっているが、本稿では省略する。また、使用上の注意事項についても省略する。これらについては、マニュアル[8)]を参照されたい。

12. 書式仕様

12.1 編集記述子

SX77では、編集記述子としてE*w*.*d*D*e*形およびZ*w*形が追加される。E*w*.*d*D*e* 形は指数部を示す文字がDとなることを除いて、E*w*.*d*E*e*形と同一である。Z*w*形は16進数字の編集に用いるが、その詳細については省略する。

E*w*.*d*形、D*w*.*d*形などの編集記述子を用いた入出力変換において、変換の対象となる桁数が内部表現の仮数部の桁数に比較して、不必要に長い場合には次のように処理される。ただし、*l*は単精度では9、倍精度では18、4倍精度では35とする。

入力の場合　意味のある数字列の中の左から*l*桁が用いられ、残余はゼロと見なされる。

出力の場合　*l*桁の後にゼロを補って出力する。

O*w*形およびB*w*形に対応できる入出力並び項目の型は、SX77とAF77Vでは少し異なるが、その詳細については省略する。

12.2 行送り文字

SX77で使用できる行送り文字は、0～9、空白および＋の12種類であり、&は使用できない。

12.3 並びによる書式

並びによる入力ではSX77とAF77Vの間に差異はない。並びによる出力では、適用される編集記述子が**表6**のように異なる。ただし、文字型以外では値を区切る空白は2つに編集される。

表6　並びによる出力において適用される編集記述子

出力並び項目の型	編集記述子	
	SX77	AF77V
整数型	I 11	I 15
実数型	1PG14.7 または1PE13.6 注1)	1PG15.8 または1PE15.7 注1)
倍精度実数型	1PG23.16または1PD22.15注1)	1PG25.18または1PD25.17注1)
4倍精度実数型	1PG40.33または1PQ39.32注1)	1PG43.36または1PQ43.35注1)
複素数型	(実部, 虚部)注2)	(実部, 虚部)注2)
倍精度複素数型	(実部, 虚部)注2)	(実部, 虚部)注2)
4倍精度複素数型	(実部, 虚部)注2)	(実部, 虚部)注2)
論理型	L 1	L 1
文字型	項目の値がそのまま出力される	項目の値がそのまま出力される

注1）G形が実質上F形と等価な場合にのみ，G形で出力される。

注2）実部と虚部の出力の形は，実数型，倍精度実数型および4倍精度実数型と同一。

13. 関数およびサブルーチン

13.1 組込み関数と外部関数

SX 77 において使用できる組込み関数は、AF 77 V と比較すると少し変更がある。その変更部分については **13.3** に示す。

外部関数の定義において、FUNCTION文中の型の指定を

INTEGER＊*l*, REAL＊*l*, COMPLEX＊*l*, LOGICAL＊*l*

で行う場合の *l* の値は、型宣言文と同じ仕様である（ **7.3** 参照 ）。

13.2 RETURN文と選択もどり

RETURN *e* 形のRETURN文において、*e* は整数型、実数型、倍精度実数型または 4 倍精度実数型の算術式が許される。AF 77 V では整数型の算術式のみであった。

コンパイラオプションにDOLLARを指定すると、RETURN *e* に対応する仮引数および選択もどり指定子として、＊のかわりに＄を使用することができる。AF 77 V のコンパイラオプションDLR と同一の機能であるが、コンパイラオプションの名前が異なる。

13.3 組込み関数の種類

SX 77 で使用できる組込み関数は、AF 77 V と比較して以下の追加、変更がなされている。

(1) SX 77 固有のもの

FACT, DFACT, QFACT	階乗（総称名なし）
DMAX0, QMAX0, DMIN0, QMIN0	最大値、最小値（総称名なし）
RANDOM	一様乱数（総称名なし）

(2) 使用できないもの

POW, DPOW, QPOW

(3) 仕様変更されたもの

SX 77 とAF 77 V の間で数の表現範囲、表現桁数、内部表現などに差異があるため（ **4.1** 参照 ）、組込み関数の定義域、値域およびビット操作については、当然ながら差異がある。その他に内部コードに関連して注意を要するものがあり、以下に示す。

- ○ 型変換を行うICHARおよびCHARは、EBCDICカタカナコードの大小順序に従う。ただし、コンパイラオプションにCOLLATE＝JISを指定すると、AF 77 V の場合と同様にJISコードの大小順序となる。
- ○ 文字列の大小比較を行うLGE, LGT, LLEおよびLLTは、コンパイラオプションに依存せず、AF 77 V の場合と同様にJISコードに従う。

(4) 内部処理の変更されたもの

一様乱数を発生するIRANDとFRANDは、ACOS 1000 とSXのCPではファームウェア

を用いている。しかしながら、APにはこのようなファームウェアの機能がないために、SX77−APではソフトウェアによって実現している。したがって、APはCPよりも高速[2)]であるにもかかわらず、IRANDとFRANDの実行速度はAPとCPでほぼ同一である。

13.4 組込みサブルーチンの種類

SX77で使用できる組込みサブルーチンは、AF77Vと比較して以下の追加、変更がなされている。ただし、CPU時間を測定するもの（CPTIME, PTIME, CLOCK）については(4)に示す。

(1) SX77固有のもの

ABTERM, DATE, ELTIME, ERRALT, ERRANY, ERROPT, ERRSAV, ERRSTR, MESPUT, REREAD, RSTSW, SETSW, TIME.

(2) 使用できないもの

ANYERR, ATTACH, BINHEX, CALLGT, CALLSS, CNSLIO, CORFL, CREATE, DETACH, FCLOSE, FILBSP, FILFSP, FMEDIA, FPARAM, FXALT, FXDVCK, FXEM, FXOPT, HEXBIN, LONGIO, RANSIZ, RBREAK, SETFIB, SETIME, TERMNO, TERMTM, USRCOD.

(3) 仕様変更されたもの

DATIM, ERRCNT, FNOISE, IRETSW, ISETSW, OVERFL, SLITE, SLITET, SSWTCH.

(4) CPTIME, PTIMEおよびCLOCK

CPU時間の測定には、AF77VではCPTIMEとPTIMEを用いた。その仕様は、バッチ処理ではプログラムの実行開始時から、TSS処理ではログオン時からの累積実行時間となっていた。SX77−CPにおいてもCPTIMEとPTIMEは使用できるが、バッチ処理とTSS処理のいずれにおいても、プログラムの実行開始時からの累積実行時間となっている。

SX77−APでは、その高速性のためにCPTIMEおよびPTIMEは分解能がひくすぎて使用できない。したがって、SX77−APとSX77−CPの双方で使用できるものとしてCLOCKが追加されている。その呼出し手順は、

CALL CLOCK（*d*）　　ただし、*d*は倍精度実数型の変数名または配列要素名

である。その仕様はバッチ処理とTSS処理で同一であり、プログラムの実行開始時からの累積実行時間が秒単位で引数*d*に返却される。

現時点ではAF77VにおいてCLOCKを使用することはできないが、次回のバージョンアップ時に追加される予定である。

13.5 ベクトル関数

SXのAPはベクトルプロセッサであり、ACOS 1000のIAPよりもハードウェアの機能が大幅に増強されている。これに対応して、SX 77−APのベクトル化処理機能もAF 77 Vよりも強化されているために、AF 77Vのベクトル関数に相当するものはSX 77には存在しない。

14. 主プログラムと初期値設定副プログラム

14.1 主プログラム

SX 77ではAF 77Vと比較して、下記の点が異なる。

- ○ 主プログラム名の既定値はFTMAINである。ただし、TSS処理においてソースプログラムがSX上の統合ライブラリ[5)]にあるときには、そのサブファイル名（先頭8文字）またはFTMAINのいずれかとなる。その選択の規則については省略する。
- ○ PROGRAM文がない場合に、コンパイラオプションMAINによって主プログラム名を与えることができる。
- ○ 主プログラム中のRETURN文はSTOP文とみなされる。

14.2 初期値設定副プログラム

SX 77ではAF 77Vと比較して、下記の点が異なる。

- ○ 初期値設定副プログラム名の既定値はFTBLKDである。
- ○ BLOCK DATA文に名前が省略されている場合に、コンパイラオプションBLKDATAによって初期値設定副プログラム名を与えることができる。

15. コンパイラ指示文

15.1 INCLUDE文

INCLUDE文はそのメンバーがSX上の統合ライブラリ[5)]にあるときのみ使用できるが、その詳細については省略する。

15.2 EJECT文

EJECT文は、コンパイル時のソースプログラムリストおよび編集リストの出力において、その文で改ページを指示するための文であり、次の形である。

EJECT

EJECT文に文番号をつけてはならず、また開始行のカラム7から72の間にのみ書くことができる。このような機能はAF 77Vにはなかった。

15.3 ベクトル化指示行

ベクトル化指示行は、ソースプログラムの解析によって得ることができない情報を利用者が与え

ることにより、SX77−APにおいてベクトル化の効果を促進させるためのものである。AF77Vには、このような機能はなかった。

ベクトル化指示行は、コンパイラオプションVECTOR，NOVECTORがプログラム単位の全体に適用されるのに対し、個々のDOループに指定でき、機能も強化されている。次にその形式を示す。

*VDIR　*ol*　　ただし、*ol*はオプションの並びを示す。

ベクトル化指示行の規則などを以下に示す。

- ○ コンパイラオプションがVDIRのときに有効であり、NOVDIRのときおよびSX77−CPでは注釈行とみなされる。
- ○ カラム1から5に*VDIRと書き、その後に1つ以上の空白を置いてオプションの並びを指定する。各オプション間の区切りにはコンマを用いる。
- ○ オプションのキーワードの途中および値の途中に空白を書いてはならない。
- ○ ベクトル化指示行を、開始行と継続行、および継続行と継続行の間に書いてはならない。
- ○ ベクトル化指示行の継続はできないが、複数のベクトル化指示行を連続して書いてもよい。
- ○ オプションはコンパイラへの指示を表し、コンパイラオプションVECTOR，NOVECTORの指定よりも優先される。
- ○ 相反するオプションが同時に指定された場合には、後で指定されたものが有効となる。
- ○ ベクトル化指示行を書く位置はオプションに依存する。位置および個々のオプションの内容については省略する[3)]。

16. おわりに

アーキテクチャの異なる計算機をバックエンドプロセッサとして接続[2)]するために、FORTRAN77の言語仕様および利用法は、ACOS1000での機能よりも制約を受けます。しかしながら、可能な範囲ではSX77とAF77Vは互換性を考慮されており、プログラムの修正が少くなるように配慮されています。大部分の利用者にとっては、プログラムの修正は不要、あるいは少しの修正によりSX上で動作させることができると思います。ただし、SX上で動作することと意味のある結果を得ることは別の問題であり、経験上、AF77Vからの移行において最も注意すべき点は、浮動小数点表現の仮数部の桁数が短くなっていることです。すなわち、収束判定常数の再検討、単精度演算の倍精度化などを考慮する必要があります。これらの考慮を行わなければ、反復回数の増加やエンドレスループとなることがあり、SXのAPのスカラー演算の能力さえも十分に引出すことはできません。ベクトル化処理機能を活用し、APのベクトル演算の能力を十分に引出すためのプログラミング技法については、別資料に記述がなされる予定です[4,8)]。

本稿では、SX77とAF77Vの文法上の差異を中心として述べました。SX77の利用法については、SX　FORTRAN77概要(2)[3]を御覧下さい。この原稿を書いている時点(1986年2月)では、SXのハードウェアおよびソフトウェアはでき上っているものの、ACOS1000のバックエンドプロセッサとして接続するためのソフトウェアは作成中です。したがって、今後、仕様や運用上の制約などに変更が行われる可能性があることを御了承下さい。種々の制約はあるものの、SXは高性能であり、他センターのベクトル計算機と比較してもベクトル演算では遜色なく、スカラー演算では優位に立っています。HFPの利用者がSXに移行するばかりでなく、ACOS1000からの移行をおすすめします。利用者がSXに移行するための資料として、本稿が役立つことを望みます。

謝　　辞

本稿をまとめるにあたり、資料提供などの御協力を賜った日本電気株式会社基本ソフトウエア開発本部片山博主任ならびに関西日本電気ソフトウェア株式会社三原敏敬氏、河原久美子氏をはじめとする諸氏に感謝します。

参　考　文　献

SX関連

1) 渡辺，近藤，端山，大中，藤井：スーパーコンピュータSX-1の概要(1)，大阪大学大型計算機センター・ニュース，Vol. 15，No.4 (1986).
2) 藤井：スーパーコンピュータSX-1の概要(2)，大阪大学大型計算機センター・ニュース，Vol. 15，No.4 (1986).
3) 後藤，大中：SX　FORTRAN77概要(2)，大阪大学大型計算機センター・ニュース，Vol.16，No.1 (1986).
4) 片山，河原，大中：FORTRAN77/SXにおける高速化技法，大阪大学大型計算機センター・ニュース，Vol. 16，No.1 (1986).
5) 馬野：スーパーコンピュータSX-1のタイム・シェアリング・システムATSS-AFの使い方（その1），大阪大学大型計算機センター・ニュース，Vol. 16，No.1 (1986).
6) 多喜：ACOS-1000とSX-1とのファイル転送について，大阪大学大型計算機センター・ニュース，Vol. 16，No.1 (1986).
7) GGB11-1　FORTRAN77言語説明書，日本電気(1985).
8) GGB12-2　FORTRAN77，77/SXプログラミング手引書，日本電気(1986).
9) GJF11-1　MSF-6利用説明書，日本電気(1986).
10) GDA11-2　SXCP解説書，日本電気(1986).

ACOS1000関連

11) 大中，後藤：FORTRAN77概説(1)，大阪大学大型計算機センター・ニュース，Vol. 11，No.3 (1981).
12) 大中，後藤：FORTRAN77概説(2)，大阪大学大型計算機センター・ニュース，Vol. 11，No.4 (1982).
13) 大中，後藤：FORTRAN77概説(3)，大阪大学大型計算機センター・ニュース，Vol. 12，No.1 (1982).

14）AGB01－4　FORTRAN77言語説明書，日本電気（1985）.

15）FGB27－1　FORTRAN77（V）プログラミング手引書，日本電気（1985）.

HFP関連

16）大中，後藤：HFP　FORTRAN77概要（1），大阪大学大型計算機センター・ニュース，Vol．14，No.3（1984）.

17）後藤，大中：HFP　FORTRAN77概要（2），大阪大学大型計算機センター・ニュース，Vol．14,,No.3（1984）.

18）FQA01－1　HFPサブシステム・利用手引書（標準形），日本電気（1984）.