

Title	FORTTRANコンパイラの機能拡張
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. 1978, 31, p. 61-70
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/65399">https://hdl.handle.net/11094/65399</a>
rights	
Note	

*Osaka University Knowledge Archive : OUKA*

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

Osaka University

# FORTRANコンパイラの機能拡張

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. ま え が き

10月16日からFORTRANコンパイラの機能が拡張され、16進指数部浮動小数点演算、4倍精度演算、SUBCHK等の機能が使用可能となりました。その概要についてはすでに速報N055(10月12日発行)でお知らせしましたが、本稿では、その詳細について解説いたします。

拡張機能を含むFORTRANコンパイラ(以下新コンパイラと略す)の最大の特徴は、浮動小数点演算を16進指数部浮動小数点形式(16進モード)と2進指数部浮動小数点形式(2進モード)の2種類が選択できることです。選択の範囲が広がることは利用者にとって好ましいことですが、反面、適切な選択を行う必要があります。適切な選択を行うには、その特徴を充分に把握するための情報が必要です。情報の元となるのは大部分がマニュアルですが、マニュアルそのものはけっして読み易いとはいえません。本稿が適切な選択のための情報として役立てば、幸いに存じます。

なお、主プログラム、副プログラム等のコンパイル単位間で16進モードと2進モードを混在させて使用することもできますが、使用法が複雑かつ制限が多く、センターの標準サービスとはいたしません。したがって、本稿では混在して用いる場合については省略します。

## 2. 演 算 モ ー ド

新コンパイラは、浮動小数点演算を16進指数部浮動小数点形式で行う16進モードと、2進指数部浮動小数点形式で行う2進モードを選択することができる。これらは\$FORTRAN文あるいはRUNコマンドに次のオプションを指定することにより選択する。

HEX : 浮動小数点演算を16進で行うオブジェクトプログラムを生成する。

BIN : 浮動小数点演算を2進で行うオブジェクトプログラムを生成する。

これらの2つのモードによって実数データを扱う機械語が変わるわけではなく、インジケータ・レジスタの32ビット目がオンであれば16進モード、オフであれば2進モードとなる。したがって、このビットが実行の途中で変わってしまうと、予期しない結果を得ることになる。HEX、BINいずれのオプションにおいても、オブジェクトプログラムは16進モードと2進モードを混在させて用いることは考慮していない。すなわち、両オプションのオブジェクトプログラムをリンクして実行させてはならない。2進モードと16進モードをコンパイル単位間で混在させて用いる場合のためにACCOMオプションがあるが、使用法が複雑かつ制限も多く、センターの標準サービスとはしない。

ACCOM : 浮動小数点演算を16進で行うが、2進モードで作られたオブジェクトプログ

ラムとリンクして実行可能なオブジェクトプログラムを生成する。なお、このオプションはHEXオプションと同時に指定したときに有効となる。

なお、他言語（GMAPを除く）とのリンク、旧コンパイラで生成されたオブジェクトプログラムとのリンク、アプリケーション（MATHLIB-6, TSS/LIB-6を除く）のOWNコーディングに用いる場合はBINあるいはACCOMオプションを使用しなければならない。

### 3. 浮動小数点数の表現範囲と内部表現

浮動小数点数の表現範囲に関する16進モードと2進モードの差異は次の3点である。（表1参照）

- ① 正規化のために16進モードの方が有効桁数が3ビット短い。
- ② 指数部の表現範囲は16進モードの方が大巾に広い。
- ③ 4倍精度演算は16進モードでのみ使用可能である。

表1 浮動小数点数の表現範囲

モード		16進モード	2進モード
単精度	有効桁数(10進)	7.2桁	8.1桁
	値の範囲(絶対値)	$0, 10^{-155} \sim 10^{152}$	$0, 10^{-38} \sim 10^{38}$
倍精度	有効桁数(10進)	18.0桁	18.9桁
	値の範囲(絶対値)	$0, 10^{-155} \sim 10^{152}$	$0, 10^{-38} \sim 10^{38}$
4倍精度	有効桁数(10進)	36.1桁	使用不可
	値の範囲(絶対値)	$0, 10^{-155} \sim 10^{152}$	

有効桁数の差異から、同一精度では2進モードの方が一般的に解の信頼性が高い。とくに、反復計算を行う場合には2進モードで実行すれば反復回数が少なくて収束することが予想される。反復の停止則がきびしい場合に3ビット(10進換算0.9桁)の差は大きく、極端な場合は2進モードで収束するにもかかわらず、16進モードでは収束しないこともある。一方、16進モードの利点は4倍精度演算と指数部の表現範囲の拡大である。NEAC2200-700から、ACOSにシステムが更新されて以来、オーバーフロー、アンダーフローになやまされているユーザーにとって、16進モードは有用であろう。

次に浮動小数点数の内部表現を示す。通常のプログラムでは必ずしも内部表現を知る必要はないが、B変換、O変換による入出力、ビット操作などの場合には、内部表現を知っておかねばな

らない。

実数型の内部表現

	0	7 8		35
e	m			
e <sub>7</sub> e <sub>0</sub>	m <sub>0</sub>			m <sub>27</sub>

指数部  $e = -e_7 2^7 + e_6 2^6 + \dots + e_0 2^0$

仮数部  $m = -m_0 2^0 + m_1 2^{-1} + \dots + m_{27} 2^{-27}$

値  $v = \begin{cases} m \times 16^e & (16 \text{ 進モード}) \\ m \times 2^e & (2 \text{ 進モード}) \end{cases}$

倍精度実数型の内部表現

	0	7 8		35 36		71
e	m					
e <sub>7</sub> e <sub>0</sub>	m <sub>0</sub>			m <sub>27</sub>	m <sub>28</sub>	m <sub>63</sub>

指数部  $e = -e_7 2^7 + e_6 2^6 + \dots + e_0 2^0$

仮数部  $m = -m_0 2^0 + m_1 2^{-1} + \dots + m_{63} 2^{-63}$

値  $v = \begin{cases} m \times 16^e & (16 \text{ 進モード}) \\ m \times 2^e & (2 \text{ 進モード}) \end{cases}$

4倍精度実数型の内部表現

	0	7 8		71 72	79 80	84		143
e <sub>u</sub>	m <sub>u</sub>		e <sub>l</sub>	0		m <sub>l</sub>		m <sub>l59</sub>
e <sub>7</sub> e <sub>0</sub>	m <sub>u0</sub>			m <sub>u63</sub>			m <sub>l0</sub>	m <sub>l59</sub>

上位指数部  $e_u = -e_7 2^7 + e_6 2^6 + \dots + e_0 2^0$

上位仮数部  $m_u = -m_{u0} 2^0 + m_{u1} 2^{-1} + \dots + m_{u63} 2^{-63}$

下位指数部  $e_l = \begin{cases} e_u - 15 & e_u - 15 \geq -128 \\ 0 & e_u - 15 < -128 \end{cases}$

下位仮数部  $m_l = m_{l0} 2^{-64} + m_{l1} 2^{-65} + \dots + m_{l59} 2^{-123}$

値  $v = (m_u + m_l) \times 16^{e_u} \quad (16 \text{ 進モード})$

#### 4. 宣 言 文

##### ○ 型 宣 言 文

16進モードの場合に4倍精度関係の宣言文として次のものを追加ならびに仕様変更を行う。

追 加      QUADRUPLE    PRECISION  
QUADRUPLE

COMPLEX QUADRUPLE PRECISION  
 COMPLEX QUADRUPLE

仕様変更

REAL * a	{	0 ≤ a < 8	実数型
		8 ≤ a < 16	倍精度実数型
		16 ≤ a	4倍精度実数型
COMPLEX * a	{	0 ≤ a < 16	複素数型
		16 ≤ a < 32	倍精度複素数型
		32 ≤ a	4倍精度複素数型

○ 整合寸法

16進モードの場合に整合寸法をCOMMON文によって定義する機能を追加する。

CHARACTER宣言において文字数を示す整数型変数名についても同様である。

<pre> 例  SUBROUTINE SUB1(X, Y)       COMMON I       DIMENSION X(I), Y(I)       )       RETURN       END </pre>	<pre> SUBROUTINE SUB2(A, B)   COMMON J   CHARACTER A*(J(10)), B   )   RETURN   END </pre>
----------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------

○ EXTERNAL文、ABNORMAL文

16進、2進のいずれのモードにおいても、EXTERNAL文又はABNORMAL文中に、組み関数名、基本外部関数名、基本外部サブルーチン名が現われると、その名前は組み関数名、基本外部関数名、基本外部サブルーチン名として扱われなくなる。すなわち、コンパイラは一般名などによる名前の変更<sup>(注)</sup>を行わず、未定義の手続きとして残す。この機能に関してABNORMAL文はNOPTZでも有効である。

(注) 基本外部関数、基本外部サブルーチンは16進、2進のモードで各々異なったモジュールとなっている。これらはローダによって正しくリンクされるように、16進モードの場合には基本外部関数、基本外部サブルーチンの名前を変更し、FORTRANでユーザーが使用できない名前としている。

5. 定数と入出力関係

○ 定数

16進モードの場合の4倍精度指数部は文字E、Dの代わりに文字Qを使うこと以外は、単精度、倍精度の場合と同様である。

例	0.1 2 3 4 5 6 7 E 1	単精度
	0.1 2 3 4 5 6 7 D 1	倍精度
	0.1 2 3 4 5 6 7 Q 1	4倍精度

文字D、Qを用いなくても倍精度、4倍精度として扱われる場合の区分は、2進、16進モードの有効桁数の差異(表1参照)から次のようになる。すなわち、2進モードは従来通りであり、基本実定数又は整定数の部分の有効桁数が9桁を越えれば(正確には小数点を無視した10進数値が134217728以上)、倍精度実定数となる。一方、16進モードでは有効桁数が8桁を越え18桁までであれば(16777216以上、1152921504606846975以下)、倍精度実定数となり、18桁を越えれば(1152921504606846976以上)、4倍精度実定数となる。

○ 書式指定機能

16進モードの場合には、ASSIGN文によってFORMAT文の文番号をスイッチ変数に割当てることにより、そのスイッチ変数を書式指定として使用できる。

○ 入出力変換

16進モードの場合の4倍精度用入出力変換としては、Q変換又はG変換を用いる。数値欄記述子Qw.dは欄がw桁を占め、そのうち小数部がd桁から成ることを示す。d ≥ 37の場合にはd = 36とみなす。なお、4倍精度に対する暗黙書式はQw.d(入力)、Q44.36(出力)となる。

○ 桁移動子

16進、2進のモードにかかわらず、従来、E、D変換に桁移動子nPを作用させるとn ≥ 2の場合に出力形式がJIS規格と異なっていたので、JIS規格と同一仕様とした。

例 0.7654321を2PE15.5で出力する。

旧コンパイラ            76.54321E-02

新コンパイラ            76.5432E-02

なお、Q変換に対する桁移動子の効果はE、D変換の場合と同一である。

## 6. 組込み関数と基本外部関数

16進モードの場合に、4倍精度関係を主として以下の組込み関数、基本外部関数を追加する。各々の定義は単精度あるいは倍精度のものからの類推で判断できると思われるから、省略する。

組込み関数の追加

QABS, DINT, IQINT, QMAX1, QMIN1, QFLOAT, QSIGN, QDIM, SNGLQ,  
QREAL, QIMAG, DBLEQ, QEXT, QEXTD, QCMPLX, QCONJG.

基本外部関数の追加

QEXP, CQEXP, QEXP2, QEXP10, QLOG, CQLOG, QLOG10, QSIN, CQSIGN,  
QCOS, CQCOS, QTAN, QCOTAN(QCOT), QARSIN(QASIN), QARCOS(QACOS),  
QATAN, QATAN2, QSINH, QCOSH, QTANH, QSQRT, CQSQRT, QCBRT, CQABS,  
CDANG(DCANG), QINT, QMOD, IQE, QMT, QERF, QERFC.

略記計の追加

ARS IN(AS IN), DARS IN(DAS IN), ARCOS(ACOS), DARCOS(DACOS).

一般名の追加

AINT<sup>(註)</sup>, SNGL, REAL, AIMAG, DBLE, QEXT, CMLPX, CDNJG, CANG.

(註) 一般名AINTは引数の型によりAINT, DINT, QINTに変換されるが、AINT, DINTは組込み関数、QINTは基本外部関数である。

## 7. ENTRY文

2進モードの場合は従来と同一仕様であるが、16進モードでは機能が拡張される。その変更部分のみを示す。

2進モード(従来と同一)

関数副プログラムでは、引用しているプログラムに関数値をもどす変数として関数名しか使用してはならない。また関数名の型と入口名の型は同一でなければならない。

16進モード

関数副プログラムでは、入口名は関数値を定義する変数名として現われてもよい。このような変数名はFUNCTION文に現われた英文字名とEQUIVALENCE結合が行われる。この副プログラム内にあるRETURN文のいずれかが実行されるとき、これらの変数のうち最後に定義された変数の値が引用された関数の値となる。この変数の型が引用された入口名の型と異なる場合、関数の値は不定となる。

## 8. デバッグ機能

2進モードの場合は従来と同一仕様であるが、16進モードでは次の2点に変更される。

添字の値の検査機能(SUBCHK機能)の追加

ローダのデバッグ機能は使用できない。

\$FORTRAN文又はRUNコマンドでSUBCHKオプションを指定すると、プログラム中のすべての配列に対し、その引用時に添字の値が検査される。添字がすべて定数で与えられており、その配列が整合寸法を持たなければ検査はコンパイル中に行われる。それ以外では実行時に行われる。実行時に、添字の値が宣言された範囲を越えると、添字の値、配列名と配列の大きさが装置番号6に出力され、実行はそのまま続行される。SUBCHKオプションがOPTZオプションと同時に指定されると、SUBCHKオプションはないものとして扱われる。このSUBCHK機能は特定の配列名のみ作用させることができないこと、実行中にオン/オフできないことなどの点でNEAC2200-700時代よりも機能が後退していることを御了承下さい。

コンパイラの新、旧には無関係であるが、コンパイラとFDS(FORTRAN DEBUGGING SYSTEM)のコマンドプロセッサFDEBUGとの演算順序の差異に注意を要する。その注意事項と

は、べき乗演算で底が負符号付きの定数の場合である。たとえば、 $-2 ** I$ と記述すれば、コンパイラは $-(2 ** I)$ と扱い、FDEBUGは $(-2) ** I$ と扱う。

## 9. ライブラリーの使用法

16進、2進のモードが選択可能となることにともない、センター・ライブラリー、MATHLIB-6、TSS/LIB-6の格納されているファイル名が変更になるが、サブ・プログラム名、パッケージ名、引数等は従来と同一である。

### ○ センター・ライブラリーとMATHLIB-6

非標準のACCOMオプションを含めると、センター・ライブラリーとMATHLIB-6は各々HEX版、BIN版、ACCOM版が存在することになる。使用法の点では、バッチ処理の場合のファイル・コード、TSS処理の場合のULIBの指定が表2のように変更となる。なお、参考のために( )内に従来のもを示す。

表2 ファイル・コードとULIB

		センター・ライブラリー	MATHLIB-6
ファイル・ コード	HEX	CH	MH
	BIN	CB(LC)	MB(LM)
	ACCOM	CA	MA
ULIB	HEX	LIB/CLIBH	LIB/MLIBH
	BIN	LIB/CLIBB (LIB/CLIB)	LIB/MLIBB (LIB/MLIB)
	ACCOM	LIB/CLIBA	LIB/MLIBA

### ○ TSS/LIB-6

TSS/LIB-6はHEX版、BIN版のみでACCOM版は用意されない。

次にその使用法の例を示す。ただし、C\*形式の例においては、関数副プログラムがカレント・ファイル上に存在するものとする。

HEX版の使用例

H\*形式      RUN\_TLIBH/パッケージ名, R



C\*形式 RUN\_TLIBH / パッケージ名, R ; \* = (HEX)

BIN版の使用例

H\*形式 RUN\_TLIBB / パッケージ名, R

C\*形式 RUN\_TLIBB / パッケージ名, R ; \* = (BIN)

従来の使用例

H\*形式 RUN\_LIB / パッケージ名, R

C\*形式 RUN\_LIB / パッケージ名, R ; \*

## 10. JCLとRUNコマンド

HEX、BINオプションの場合の標準的なJCLとRUNコマンドの例を示す。なお、このオプションを指定しない場合の処理はACOS77-800ではBINとなるが、ACOS77-900ではHEXとする。

JCL

1	8	16	1	8	16
\$	SNUMB	受付番号	\$	SNUMB	受付番号
\$	JOB	課題番号など	\$	JOB	課題番号など
\$	LIBRARY	CH, MH 注1)	\$	LIBRARY	CB, MB 注1)
\$	FORTTRAN	HEX	\$	FORTTRAN	BIN
\$	LIMITS	, nnK 注2)	\$	LIMITS	, nnK 注2)
	}	ソースプログラム		}	ソースプログラム
\$			GO		
	}	データ		}	データ
\$			ENDJOB		

注1) 使用するライブラリーのファイル・コードを書くこと。

注2) 新コンパイラは最低33KW必要であり、旧コンパイラより約5KW大きくなっている。

RUNコマンド(カレント・ファイル上のプログラムを実行する例)

HEXオプション

RUN\_=(HEX)

HEXオプション(センター・ライブラリー, MATHLIB-6使用)

RUN\_=(HEX, ULIB)LIB/CLIBH, R; LIB/MLIBH, R

BIN オプション

RUN □ = ( BIN )

BIN オプション ( センター・ライブラリー, MATHLIB-6 使用 )

RUN □ = ( BIN, ULIB ) LIB / CLIBB, R ; LIB / MLIBB, R

16 進モードと 2 進モードを混在させて用いる場合の JCL, RUN コマンドについては省略する。

## 11. 制 限 事 項

制限事項とは近い将来に解除される予定のもので 10 月 31 日現在、新コンパイラには以下に示す 7 つがある。なお、③～⑥の事項は旧コンパイラからひきついだものである。

- ① 2 進モードかつ JIS オプションを指定してローダのデバッグ機能を使用すると、出力リストの印字が乱れる。

回避策 なし。ただし BCD オプションで実行すれば正常。

- ② 16 進モードでは、FDS の測定機能 ( FTIMER ) は使用できない。

回避策 なし。

- ③ DO 文に対し、文法違反をするとコンパイル時にアボートすることがある。

回避策 DO 文を正しく書き直す。

- ④ 論理式中に論理演算子 . AND . と . OR . を含み、かつ同じ添字式をもつ配列要素の引用が 2 個以上ある場合に、添字式の値が正しく評価されないことがある。

IF ( L . AND . AR(I) . GT . 10 . . OR . AR(I) . LT . 2 . ) GO TO 10

①

②

③

上記の論理 IF 文において①が偽であれば②の評価は行わずに③の評価を行う。③の評価の際、配列要素 AR(I) の添字計算は②で計算されていることになっているため、その値を利用したオブジェクトコードを生成する。このオブジェクトコードは①が真であれば正しいが、偽の場合には誤まった添字計算を行うことになる。

回避策 同じ添字式をもつ配列要素を変数に代入し、その変数を論理式中で引用する。上記の例であれば、次のように書き直せばよい。

X = AR ( I )

IF ( L . AND . X . GT . 10 . . OR . X . LT . 2 . ) GO TO 10

- ⑤ FDS 機能を使用する場合、STOP 文に対してデバッグコマンドを設定すると、プログラム終了時に FDUMP ( バッチ処理 ) 又は FDEBUG ( TSS 処理 ) が誤って引用される。

回避策 なし。ただし実行は正常に行われる。

- ⑥ 文関数定義文で、右辺に左辺で定義した文関数名を書くと、コンパイル時にアボートする。

回避策 文法の誤りを正して実行する。

- ⑦ 2 進モードにおける F 変換の出力で、整数部分の有効桁数が 9 桁以上のとき出力形式が乱れる。

回避策 E変換を使用する。

## 12. む す び

FORTRANコンパイラに新しく追加された機能について述べましたが、新、旧コンパイラの差異のみならず、新コンパイラでも16進モードと2進モードではもはや同一のコンパイラとは言えないほど異なっています。本稿で述べた差異の中で、下記の4点については次回のバージョンアップ(日本電気のリリース予定 54年3月末)のときに2進モードでも使用可能となる見込みです。

- ① 整合寸法のCOMMON文による定義
- ② スイッチ変数による書式指定機能
- ③ ENTRY文の機能拡張
- ④ SUBCHK機能

本稿では省略しました16進と2進の両モードを混在させて用いる場合、ならびに疑問点についてはプログラム相談室にお問合せ下さい。なお、新コンパイラのマニュアルはプログラム相談室とデバッグ室に備えています。

最後に、本稿をまとめるに際し、有益な御助言を賜った日本電気株式会社奥居稔、端山信幸の両氏に感謝いたします。

### 参考マニュアル

FGB02-4 ACOS-6 プログラム管理 FORTRAN 文法説明書。

FGB03-4 ACOS-6 プログラム管理 FORTRAN プログラミング説明書。

FGB04-4 ACOS-6 プログラム管理 FORTRAN サブルーチンライブラリ説明書。

(S53. 10. 17受理)