

## 第7章 組込み手続とライブラリ手続

HPF の仕様には、Fortran の組込み手続が含まれるほか、2 つの範疇に属する新しい組込み手続が追加される。システム問合せ組込み関数と計算組込み関数である。

これら新しい組込み関数に加え、HPF では HPF\_LIBRARY というライブラリモジュールも定義されており、HPF 仕様を完全に実装するベンダは必ずこれを提供しなければならない。

本章の HPF 組込み手続、ライブラリ手続に関する記述は、Fortran 規格の形式と規定に従っている。F95:13.1、F95:13.2、F95:13.3、F95:13.5.7、F95:13.8.1、F95:13.8.2、F95:13.9、及び F95:13.10 節の内容は、HPF の組込み手続やライブラリ手続、及びそれらに関する本章の解説に対しても有効である。

### 7.1 表記法

本章の例題において、T と F は、それぞれ論理値の真と偽を表すものとする。

### 7.2 システム問合せ組込み関数

マルチプロセッサ上の実装では、プロセッサの構成は実装ごとに異なる多次元プロセッサ配列となる。システム問合せ関数の返す値は、プロセッサ配列の大きさや形状といった、実際の計算機やプロセッサの構成に関するものである。NUMBER\_OF\_PROCESSORS は、プログラムで利用可能なプロセッサの総数、またはプロセッサ配列の指定された次元における、プログラムで利用可能なプロセッサの数を返す。また PROCESSORS\_SHAPE は、プロセッサ配列の形状を返す。

Fortran の制限式の定義は、HPF のシステム問合せ組込み関数を引用できるよう拡張される。特に、Fortran 規格の 7.1.6.2 項のリストの最後に、以下の項目が追加される。

(13) システム問合せ関数 NUMBER\_OF\_PROCESSORS または PROCESSORS\_SHAPE であって、あらゆる引数が制限式であるものの引用。

モジュールまたは主プログラムの有効域において、HPF 指示文中の制限式の中に現れる変数は、DO 型並び変数であるか、配列問合せ関数、ビット問合せ関数、文字問合せ関数、種別問合せ関数、数値問合せ関数のいずれかの引数としての引用でなければならない。

システム問合せ組込み関数の返す値は、1 つのプログラムの実行中は不変である。したがって、NUMBER\_OF\_PROCESSORS と PROCESSORS\_SHAPE は、制限式である値を持ち、Fortran の他の任意の制限式が使用できるところならばどこでも使用できる。特に、NUMBER\_OF\_PROCESSORS は、宣言式中で使用できる。

システム問合せ関数の値を、初期化式中で引用してはならない。それらは定数ではないことがあるからである。HPF プログラムは、翻訳時には構成の分からない計算機向けに翻訳されることもある。

システム問合せ関数が問い合わせるのは物理的な計算機に関してであり、いかなる PROCESSORS 指示文とも無関係であることに注意されたい。HPF プログラムがより大きな計算機の物理的な一部分で実行されている場合には、問合せが行われるのは実際にプログラムを実行している小さな部分に関してであり、その諸元がシステム問合せ関数により返される。

関数 PROCESSORS\_SHAPE の値として返すべき「自然な」形状を持たない計算機もある。たとえば、木構造のトポロジを持つ計算機などである。このような場合は、たとえば大きさが NUMBER\_OF\_PROCESSORS() の 1 次元配列など、ある程度の合理性と一貫性を持った計算機構成を与えるように実装されなければならない。言語処理系は、この構成と実際のハードウェア・プロセッサ識別方法との間の対応付けを行なう必要がある。

【利用者への助言】 SIZE(PROCESSORS\_SHAPE()) は、プロセッサ配列の次元数を返す。配列の宣言や HPF 指示文中で、システム問合せ関数を以下のように引用することができる。

```
INTEGER, DIMENSION(SIZE(PROCESSORS_SHAPE())) :: PSHAPE
!HPF$ TEMPLATE T(100, 3*NUMBER_OF_PROCESSORS())
```

【以上】

### 7.3 計算組込み関数

HPF には新たに計算組込み関数が 1 つ追加された。ILEN は、整数値を格納するのに必要なビット数を計算する。

### 7.4 ライブラリ手続

本章で述べるマッピング問合せサブルーチンと計算関数は、HPF ライブラリモジュール HPF\_LIBRARY の中に用意されている。これらの手続を引用する際は、各有効域において適宜 USE 文を指定しなければならない。これらは組込みではない。

#### 7.4.1 マッピング問合せサブルーチン

HPF はデータマッピング指示文を提供するが、それは本質的には言語処理系に対する助言にすぎない。マッピング問合せサブルーチンにより、プログラムは実行時に、配列が実際にどのようにマッピングされているかを知ることができる。正確なマッピングを知ることが、EXTRINSIC 副プログラムを呼び出すときに特に重要である。このような理由から、配列の計算機上での実際のマッピングを記述するマッピング問合せサブルーチンが HPF に含められた。手続の数を抑えるため、問合せ手続は省略可能な INTENT (OUT) 引数を持つサブルーチンとして構成されている。

## 7.4.2 ビット操作関数

HPF ライブラリには、3つの要素別処理ビット操作関数が含まれる。LEADZ は、整数表現の先頭にある0であるビットの数を計算する。POPCNT は、整数中の1であるビットの数を数える。POPPAR は、整数中の1であるビットの奇偶を計算する。

## 7.4.3 配列集計関数

HPF には Fortran の SUM、ANY などの組込み関数と同様の演算を行う配列集計関数が追加された。新しい集計関数は、IALL、IANY、IPARITY 及び PARITY であり、それぞれ交換則と結合則を満たす2進演算 IAND、IOR、IEOR、及び .NEQV. に対応する。

関数の仕様の記述で「XXX 集計」という用語が使われているが、XXX は上述の2進演算のうちの1つであるものとする。例示によってこれらを定義する。array の全要素のうち mask の対応する要素が真であるものの IAND 集計は、以下のプログラムで result に計算される整数型スカラとする。

```
result = IAND_IDENTITY_ELEMENT
DO i_1 = LBOUND(array,1), UBOUND(array,1)
  ...
  DO i_n = LBOUND(array,n), UBOUND(array,n)
    IF ( mask(i_1,i_2,...,i_n) ) &
      result = IAND( result, array(i_1,i_2,...,i_n) )
  END DO
  ...
END DO
```

ここで  $n$  は array の次元数とし、IAND\_IDENTITY\_ELEMENT は全ビットが1である整数とする(整数のビット列としての解釈は、F95:13.5.7 で与えられる)。ほかの3つの集計演算も同様に定義される。IOR と IEOR の単位元は0である。.NEQV. の単位元は.FALSE. である。

## 7.4.4 配列集計拡散関数

XXX\_SCATTER 関数は、配列集計関数を一般化したもので、要素の任意の部分集合を集計して結果の要素を構成することができる。このとき、結果の各要素に対応する配列の部分集合は、重複部分を持たない。HPF に 11 ある集計演算は、各々集計拡散関数の1つに対応しており、それとは別に COPY\_SCATTER によって、既存の値に対応する部分集合中の任意の1つの値で上書きすることもできる。元の配列のどの要素が結果のどの要素と対応するかは、本節で述べる。それらの値を集計する方法は、7.7 節の各関数の仕様中で述べる。

これらの関数は以下のような一般形を持つ。

```
XXX_SCATTER(ARRAY, BASE, INDX1, ..., INDXn, MASK)
```

ただし、あとで述べるような特殊な場合を除く。XXX の値として許されるのは、ALL、ANY、COPY、COUNT、IALL、IANY、IPARITY、MAXVAL、MINVAL、PARITY、PRODUCT、及び SUM である。配列 ARRAY、MASK、及びすべての INDX は、形状適合するものとする。配列 INDX は整

数型であり、引数 INDX の数は BASE の次元数と同じでなければならない。引数 MASK は論理型の引数で、省略可能とする。COUNT\_SCATTER を除き、ARRAY と BASE は同じ型の配列とする。COUNT\_SCATTER については、ARRAY が論理型、BASE が整数型とする (ALL\_SCATTER、ANY\_SCATTER、COUNT\_SCATTER、及び PARITY\_SCATTER の引数 ARRAY は、論理型でなければならない。これらの関数は、省略可能な引数 MASK を持たない。Fortran 規格書にならい、これらの関数に対する引数 ARRAY は、7.7 節の仕様の中では MASK と記述している)。どの場合も、結果の配列は BASE と同じ型、種別型パラメタ、及び形状を持つ。

配列 INDX は ARRAY と同じ形状なので、ARRAY のあらゆる要素  $a$  に対して、INDX にはそれぞれ対応する要素が存在する。 $n$  を BASE の次元数とするとき、各  $j = 1, 2, \dots, n$  に対して、 $s_j$  を ARRAY の要素  $a$  に対応する INDX $j$  の要素の値とする。これらの添字により、ARRAY の要素  $a$  に影響を受ける結果の要素が決定する。すなわち、添字  $s_j$  のそれぞれに対して、BASE の対応する添字は、 $b_j = s_j + \text{LBOUND}(\text{BASE}, j) - 1$  によって与えられる。

整数  $b_j, j = 1, \dots, n$  は、BASE の要素を選択する添字となる。すなわち  $\text{BASE}(b_1, b_2, \dots, b_n)$  が選択される。BASE と結果とは形状適合するから、BASE の各要素に対して、対応する結果の要素が存在する。

したがって配列 INDX は、ARRAY のすべての要素から、結果と BASE の選択された要素への写像を定める。見方を変えると、この写像は、BASE の各要素  $b$  と ARRAY の要素の集合  $S$  とを対応づけている。

$S$  が空のとき、BASE の要素  $b$  に対応する結果の要素は、 $b$  と同じ値を持つ。

$S$  が空でないとき、 $S$  の要素は要素  $b$  とともに集計されて、結果の要素を構成する。集計拡散関数の詳細な仕様に、値を集計する方法が述べられている。例えば SUM\_SCATTER において、 $S$  の要素が  $a_1, \dots, a_m$  のとき、BASE の要素  $b$  に対応する結果の要素は、 $\text{SUM}((/a_1, a_2, \dots, a_m, b/))$  を評価した結果となる。

配列 INDX の要素の値は非負でなければならない、INDX $j$  の要素の値は  $\text{SIZE}(\text{BASE}, j)$  以下でなければならない。計算結果は、BASE の上下限の宣言の影響を受けず、BASE の形状だけに依存する。

スカラは任意の配列と形状適合するから、配列 INDX の位置に使用できる。このとき、選択される結果の要素は、1 つの超平面をなす。以下の例を参照されたい。

省略可能な最後の引数 MASK が指定されたとき、ARRAY の要素のうち、MASK が真であるような位置のものだけが演算に加わる。他のすべての ARRAY と配列 INDX の要素は無視され、結果のどの要素にも影響を与えない。

例えば、

$$\begin{array}{l} \text{配列 A が} \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \\ \text{配列 I1 が} \begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 1 \\ 3 & 2 & 1 \end{bmatrix} \end{array}, \begin{array}{l} \text{配列 B が} \begin{bmatrix} -1 & -2 & -3 \\ -4 & -5 & -6 \\ -7 & -8 & -9 \end{bmatrix} \\ \text{配列 I2 が} \begin{bmatrix} 1 & 2 & 3 \\ 1 & 1 & 2 \\ 1 & 1 & 1 \end{bmatrix} \end{array},$$

のとき、

1                   SUM\_SCATTER(A, B, I1, I2) は、  $\begin{bmatrix} 14 & 6 & 0 \\ 8 & -5 & -6 \\ 0 & -8 & -9 \end{bmatrix}$                    、  
2  
3  
4  
5                   SUM\_SCATTER(A, B, 2, I2) は、  $\begin{bmatrix} -1 & -2 & -3 \\ 30 & 3 & -3 \\ -7 & -8 & -9 \end{bmatrix}$                    、  
6  
7  
8                   SUM\_SCATTER(A, B, I1, 2) は、  $\begin{bmatrix} -1 & 24 & -3 \\ -4 & 7 & -6 \\ -7 & -1 & -9 \end{bmatrix}$                    、  
9  
10  
11  
12                   SUM\_SCATTER(A, B, 2, 2) は、  $\begin{bmatrix} -1 & -2 & -3 \\ -4 & 40 & -6 \\ -7 & -8 & -9 \end{bmatrix}$   
13  
14

15 になる。

16           配列 A が  $\begin{bmatrix} 10 & 20 & 30 & 40 & -10 \end{bmatrix}$  、 B が  $\begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix}$  、 IND が  $\begin{bmatrix} 3 & 2 & 2 & 1 & 1 \end{bmatrix}$   
17 のとき、SUM\_SCATTER(A, B, IND, MASK=(A .GT. 0)) は、  $\begin{bmatrix} 41 & 52 & 13 & 4 \end{bmatrix}$  となる。  
18  
19

#### 20 7.4.5 配列累計関数

21  
22 ベクトルの累計では、結果の各要素は、先行するベクトル要素の関数 (順累計のとき) である  
23 か、または後続するベクトル要素の関数 (逆累計のとき) である。これらの関数により、配列  
24 及び部分配列上の累計演算が提供される。関数は以下のような一般形を持つ。  
25

26       XXX\_PREFIX (ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)

27       XXX\_SUFFIX (ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)

28  
29 ただし、あとで述べるような特殊な場合を除く。XXX の値として許されるのは、ALL、ANY、  
30 COPY、COUNT、IALL、IANY、IPARITY、MAXVAL、MINVAL、PARITY、PRODUCT、及び SUM である。

31       順方向型 (prefix) と逆方向型 (suffix) の双方の手續に当てはまるとき、以下ではそれらを  
32 関数 YYYFIX と呼ぶ。

33       引数 DIM、MASK、SEGMENT、及び EXCLUSIVE は省略可能である。関数 COPY\_YYYFIX は、  
34 引数 MASK や EXCLUSIVE を持たない。また関数 ALL\_YYYFIX、ANY\_YYYFIX、COUNT\_YYYFIX、  
35 及び PARITY\_YYYFIX は、引数 MASK を持たない。これらの関数に対する引数 ARRAY は論理型  
36 でなければならず、7.7 節の仕様の中では MASK と記述している。  
37

38       引数 MASK 及び SEGMENT は論理型でなければならない。SEGMENT の形状は ARRAY と同じ  
39 でなければならない。MASK は ARRAY と形状適合しなければならない。EXCLUSIVE は論理型ス  
40 カラとする。DIM は整数型スカラであって、1 から ARRAY の次元数までの範囲の値であるも  
41 のする。  
42

43       結果の値. 結果は、ARRAY と同じ形状とする。また COUNT\_YYYFIX を除き、ARRAY と同  
44 じ型及び種別型パラメタを持つ (COUNT\_YYYFIX の結果は、基本整数型とする)。  
45

46       どの関数も、仕様に記載された関数ごとに決まったやり方で選ばれる ARRAY の要素の  
47 値によって、結果のすべての要素が決まる。省略可能な引数は、結果の各要素に対す  
48 る ARRAY の要素の選択方法に影響を与える。ここで選ばれた ARRAY の要素は、結果の

要素に寄与する、と呼ばれる。ある結果の要素に対して ARRAY のどの要素が寄与する  
のか、本節で完全に述べる。

ある結果の要素に対して ARRAY の要素が一つも選択されないとき、その結果の要素に  
は仕様に記載された関数ごとに決まった既定値が設定される。

ある結果の要素  $r$  に対して、ARRAY の対応する要素を  $a$  とする。以下の規則のいずれ  
かにより無効とされないすべての ARRAY の要素が  $r$  に寄与する。

1. 関数が XXX\_PREFIX であるとき、配列要素順序で  $a$  より後のどの ARRAY の要素も  
 $r$  には寄与しない。関数が XXX\_SUFFIX であるとき、配列要素順序で  $a$  より前のど  
の ARRAY の要素も  $r$  には寄与しない。
2. 引数 DIM が指定されたとき、第 DIM 次元の添字を除く、少なくとも 1 つの添字が  
対応する  $a$  の添字と異なるならば、ARRAY の要素  $z$  は  $r$  に寄与しない (DIM が省略  
されたとき、ARRAY、MASK、及び SEGMENT は、一時的に 1 次元配列と見なされる  
かのように、配列要素順序で処理される。DIM が指定されたときは、完全に独立  
な一群の累計演算が ARRAY の選ばれた次元に沿って実行される)。
3. 引数 MASK が指定されたとき、ARRAY の要素  $z$  は、MASK の対応する要素が真のと  
きだけ  $r$  に寄与する (MASK が偽の位置に対応する配列要素は、結果にはまったく  
寄与しない。しかし、それでも結果は、MASK が偽であるような位置も含め、すべ  
ての位置で定義される)。
4. 引数 SEGMENT が指定されたとき、ARRAY の要素  $z$  は、以下の性質をすべて満たす  
ARRAY の要素  $w$  ( $z$  自身のこともある) が存在するならば、寄与しない。
  - (a) 関数が XXX\_PREFIX のとき、 $w$  が配列要素順序で  $z$  より前にはないが、 $a$  より  
前にある。または関数が XXX\_SUFFIX のとき、 $w$  が配列要素順序で  $z$  より後  
にはないが、 $a$  より後にある。
  - (b) 引数 DIM が指定されたとき、第 DIM 次元の添字を除く  $w$  のすべての添字が、 $a$   
の対応する添字と同じである。
  - (c)  $w$  に対応する SEGMENT の要素は、 $a$  に対応する SEGMENT の要素と異なる値を持  
つ (言い換えれば、 $z$  が寄与するのは、 $z$  から  $a$  までの列に対応する SEGMENT の  
要素に同じ値が並んでいるときだけである)。
5. 引数 EXCLUSIVE が指定されていて、真であるとき、 $a$  自身は  $r$  に寄与しない。

これらの一般則から、以下のような重要な場合が導かれる。

場合 (i): ARRAY が 1 次元配列のとき、XXX\_PREFIX (ARRAY) の結果の要素  $i$  は、ARRAY  
の最初の  $i$  個の要素で決まる。また XXX\_SUFFIX (ARRAY) の結果の要素  
SIZE (ARRAY) -  $i$  + 1 は、ARRAY の最後の  $i$  個の要素で決まる。

場合 (ii): ARRAY が 2 次元以上の配列のとき、XXX\_PREFIX (ARRAY) の結果の各要素  
の値は、ARRAY の対応する要素  $a$  と、配列要素順序で  $a$  より前のすべての  
ARRAY の要素で決まる。XXX\_SUFFIX については、ARRAY の対応する要素  $a$   
と、配列要素順序で  $a$  より後の ARRAY の要素で決まる。

場合 (iii): XXX\_PREFIX (ARRAY, MASK=MASK) の結果の各要素の値は、ARRAY の対応す  
る要素  $a$  と、配列要素順序で  $a$  より前のすべての ARRAY の要素で決まるが、  
このとき MASK の対応する要素が真のものだけが結果に寄与する。この制

限のために寄与する配列要素が一つも選ばれないとき、結果の要素には関数ごとの既定値が設定される。

場合 (iv): `XXX_PREFIX(ARRAY, DIM=DIM)` の結果の各要素の値は、`ARRAY` の選ばれた要素、すなわち、`ARRAY` の対応する要素  $a$  と、第 `DIM` 次元で  $a$  より前のすべての `ARRAY` の要素で決まる。たとえば、`SUM_PREFIX(A(1:N, 1:N), DIM=2)` の結果の要素  $(i_1, i_2)$  は、`SUM(A(i_1, 1:i_2))` で計算できる。より一般的に、`SUM_PREFIX(ARRAY, DIM)` の場合、結果の要素  $i_1, i_2, \dots, i_{DIM}, \dots, i_n$  は、`SUM(ARRAY( i_1, i_2, \dots, i_{DIM}, \dots, i_n))` で計算できる (最後の式の、 $i_{DIM}$  の前のコロンに注目されたい)。

場合 (v): `ARRAY` が 1 次元配列のとき、`XXX_PREFIX(ARRAY, EXCLUSIVE=.TRUE.)` の結果の要素  $i$  は、`ARRAY` の最初の  $i - 1$  個の要素で決まる。

場合 (vi): 省略可能な引数は、任意の組合せで指定できる。

【利用者への助言】 偽から真、または真から偽へのすべての変わり目で、新しい区分が始まる。すなわち 1 つの区分は、同じ論理値が連続する最長部分列によって指定される。

```
(/T,T,T,F,T,F,F,F,T,F,F,T/)
----- - - - - - - - - - - - - - 7つの区分
```

#### 【以上】

#### 【仕様の根拠】

既存のライブラリには、各区分の開始点を指定することにより範囲を定めるものがある。また、各区分の終了点を指定することにより範囲を定めるライブラリもある。それぞれのやり方には、それぞれの利点がある。順方向型の代わりに逆方向型を実行する場合、この規定を変えるべきかどうか、という問題もある。HPF では、上述の対称的な表現を採用している。この表現の主な利点は以下の通りである。

- (A) 対称的であり、同じ区分指定子を (開始点か終了点かの) 解釈を変えずに順方向型に対しても逆方向型に対しても使用できる。
- (B) `PARITY_PREFIX` または `PARITY_SUFFIX` を引用することで、開始ビット表現や終了ビット表現を容易にこの形式に変換できる。このような方法は、言語処理系に認識させる標準的な表現法であると言えるだろう。

```
SUM_PREFIX(FOO, SEGMENT=PARITY_PREFIX(START_BITS))
SUM_PREFIX(FOO, SEGMENT=PARITY_SUFFIX(STOP_BITS))
SUM_SUFFIX(FOO, SEGMENT=PARITY_SUFFIX(START_BITS))
SUM_SUFFIX(FOO, SEGMENT=PARITY_PREFIX(STOP_BITS))
```

#### 【以上】

例示 . 以下の例は、`SUM_PREFIX` に対する省略可能な引数の、可能な組合せをすべて説明したものである。`SUM_YYYFIX` に対する既定値は 0 とする。

場合 (i): SUM\_PREFIX(/1,3,5,7/) は、 [ 1 4 9 16 ] となる。

場合 (ii): B が配列  $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$  のとき、SUM\_PREFIX(B) は配列  $\begin{bmatrix} 1 & 14 & 30 \\ 5 & 19 & 36 \\ 12 & 27 & 45 \end{bmatrix}$  となる。

場合 (iii): A が配列 [ 3 5 -2 -1 7 4 8 ] のとき、SUM\_PREFIX(A, MASK = A .LT. 6) は、 [ 3 8 6 5 5 9 9 ] となる。

場合 (iv): B が配列  $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

のとき、SUM\_PREFIX(B, DIM=1) は配列  $\begin{bmatrix} 1 & 2 & 3 \\ 5 & 7 & 9 \\ 12 & 15 & 18 \end{bmatrix}$

となり、SUM\_PREFIX(B, DIM=2) は配列  $\begin{bmatrix} 1 & 3 & 6 \\ 4 & 9 & 15 \\ 7 & 15 & 24 \end{bmatrix}$  となる。

場合 (v): SUM\_PREFIX(/1,3,5,7/), EXCLUSIVE=.TRUE.) は、 [ 0 1 4 9 ] となる。

場合 (vi): B が配列  $\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{bmatrix}$ 、M が配列  $\begin{bmatrix} T & T & T & T & T \\ F & F & T & T & T \\ T & F & T & F & F \end{bmatrix}$ 、S が配列  $\begin{bmatrix} T & T & F & F & F \\ F & T & T & F & F \\ T & T & T & T & T \end{bmatrix}$  のとき、

SUM\_PREFIX(B, DIM=2, MASK=M, SEGMENT=S, EXCLUSIVE=.TRUE.) は、  
 $\begin{bmatrix} 0 & 1 & 0 & 3 & 7 \\ 0 & 0 & 0 & 0 & 9 \\ 0 & 11 & 11 & 24 & 24 \end{bmatrix}$  となる。

SUM\_PREFIX(B, DIM=2, MASK=M, SEGMENT=S, EXCLUSIVE=.FALSE.) は、  
 $\begin{bmatrix} 1 & 3 & 3 & 7 & 12 \\ 0 & 0 & 8 & 9 & 19 \\ 11 & 11 & 24 & 24 & 24 \end{bmatrix}$  となる。

SUM\_PREFIX(B, DIM=2, MASK=M, EXCLUSIVE=.TRUE.) は、  
 $\begin{bmatrix} 0 & 1 & 3 & 6 & 10 \\ 0 & 0 & 0 & 8 & 17 \\ 0 & 11 & 11 & 24 & 24 \end{bmatrix}$  となる。

SUM\_PREFIX(B, DIM=2, MASK=M, EXCLUSIVE=.FALSE.) は、  
 $\begin{bmatrix} 1 & 3 & 6 & 10 & 15 \\ 0 & 0 & 8 & 17 & 27 \\ 11 & 11 & 24 & 24 & 24 \end{bmatrix}$  となる。



1 SUM\_PREFIX(B, DIM=2, SEGMENT=S, EXCLUSIVE=.TRUE.) は、 $\begin{bmatrix} 0 & 1 & 0 & 3 & 7 \\ 0 & 0 & 7 & 0 & 9 \\ 0 & 11 & 23 & 36 & 50 \end{bmatrix}$

2 となる。

3  
4  
5 SUM\_PREFIX(B, DIM=2, SEGMENT=S, EXCLUSIVE=.FALSE.) は、  
6  $\begin{bmatrix} 1 & 3 & 3 & 7 & 12 \\ 6 & 7 & 15 & 9 & 19 \\ 11 & 23 & 36 & 50 & 65 \end{bmatrix}$  となる。

7  
8  
9  
10 SUM\_PREFIX(B, DIM=2, EXCLUSIVE=.TRUE.) は、 $\begin{bmatrix} 0 & 1 & 3 & 6 & 10 \\ 0 & 6 & 13 & 21 & 30 \\ 0 & 11 & 23 & 36 & 50 \end{bmatrix}$  となる。

11  
12  
13 SUM\_PREFIX(B, DIM=2, EXCLUSIVE=.FALSE.) は、 $\begin{bmatrix} 1 & 3 & 6 & 10 & 15 \\ 6 & 13 & 21 & 30 & 40 \\ 11 & 23 & 36 & 50 & 65 \end{bmatrix}$  となる。

14  
15  
16  
17 SUM\_PREFIX(B, MASK=M, SEGMENT=S, EXCLUSIVE=.TRUE.) は、 $\begin{bmatrix} 0 & 11 & 0 & 0 & 0 \\ 0 & 13 & 0 & 4 & 5 \\ 0 & 13 & 8 & 0 & 0 \end{bmatrix}$

18 となる。

19  
20  
21 SUM\_PREFIX(B, MASK=M, SEGMENT=S, EXCLUSIVE=.FALSE.) は、  
22  $\begin{bmatrix} 1 & 13 & 3 & 4 & 5 \\ 0 & 13 & 8 & 13 & 15 \\ 11 & 13 & 21 & 0 & 0 \end{bmatrix}$  となる。

23  
24  
25  
26 SUM\_PREFIX(B, MASK=M, EXCLUSIVE=.TRUE.) は、 $\begin{bmatrix} 0 & 12 & 14 & 38 & 51 \\ 1 & 14 & 17 & 42 & 56 \\ 1 & 14 & 25 & 51 & 66 \end{bmatrix}$  となる。

27  
28  
29  
30 SUM\_PREFIX(B, MASK=M, EXCLUSIVE=.FALSE.) は、 $\begin{bmatrix} 1 & 14 & 17 & 42 & 56 \\ 1 & 14 & 25 & 51 & 66 \\ 12 & 14 & 38 & 51 & 66 \end{bmatrix}$  となる。

31  
32  
33 SUM\_PREFIX(B, SEGMENT=S, EXCLUSIVE=.TRUE.) は、 $\begin{bmatrix} 0 & 11 & 0 & 0 & 0 \\ 0 & 13 & 0 & 4 & 5 \\ 0 & 20 & 8 & 0 & 0 \end{bmatrix}$  となる。

34  
35  
36  
37 SUM\_PREFIX(B, SEGMENT=S, EXCLUSIVE=.FALSE.) は、 $\begin{bmatrix} 1 & 13 & 3 & 4 & 5 \\ 6 & 20 & 8 & 13 & 15 \\ 11 & 32 & 21 & 14 & 15 \end{bmatrix}$

38 となる。

39  
40  
41 SUM\_PREFIX(B, EXCLUSIVE=.TRUE.) は、 $\begin{bmatrix} 0 & 18 & 39 & 63 & 90 \\ 1 & 20 & 42 & 67 & 95 \\ 7 & 27 & 50 & 76 & 105 \end{bmatrix}$  となる。

42  
43  
44 SUM\_PREFIX(B, EXCLUSIVE=.FALSE.) は、 $\begin{bmatrix} 1 & 20 & 42 & 67 & 95 \\ 7 & 27 & 50 & 76 & 105 \\ 18 & 39 & 63 & 90 & 120 \end{bmatrix}$  となる。

45  
46  
47  
48

## 7.4.6 配列ソート関数

HPF には、多次元配列をソートするための手順が含まれている。関数 SORT\_UP と SORT\_DOWN は、ソートされた配列を返す。関数 GRADE\_UP と GRADE\_DOWN は、ソート置換列を返す。配列は、与えられた次元に沿ってソートすることも、全体配列を配列要素順序で一行に並んでいるとみなしてソートすることもできる。主キーと副キーによるソートが簡便にできるように、ソート置換列関数は安定ソートを用いる。

## 7.5 組込み手順とライブラリ手順の総称名

以下のすべての組込み手順とライブラリ手順において、表示された引数の名前は、実引数としてキーワード形式を使用するときに、キーワードとして用いなければならない名前である。多くの引数キーワードは、Fortran の場合と同様に、それらの使用法を示唆するような名前を持つ。F95:13.11 節を参照されたい。

### 7.5.1 システム 問合せ組込み関数

NUMBER\_OF\_PROCESSORS(DIM) 実行プロセッサの数  
省略可能な引数 DIM  
PROCESSORS\_SHAPE() 実行プロセッサ配列の形状

### 7.5.2 マッピング問合せサブルーチン

HPF\_ALIGNMENT(ALIGNEE, LB, UB, STRIDE, AXIS\_MAP, IDENTITY\_MAP, &  
NCOPIES)  
省略可能な引数 LB, UB, STRIDE, AXIS\_MAP, IDENTITY\_MAP, NCOPIES  
HPF\_DISTRIBUTION(DISTRIBUTE, AXIS\_TYPE, AXIS\_INFO, PROCESSORS\_RANK, &  
PROCESSORS\_SHAPE)  
省略可能な引数 AXIS\_TYPE, AXIS\_INFO, PROCESSORS\_RANK, PROCESSORS\_SHAPE  
HPF\_TEMPLATE(ALIGNEE, TEMPLATE\_RANK, LB, UB, AXIS\_TYPE, AXIS\_INFO, &  
NUMBER\_ALIGNED)  
省略可能な引数 TEMPLATE\_RANK, LB, UB, AXIS\_TYPE, AXIS\_INFO,  
NUMBER\_ALIGNED

### 7.5.3 ビット操作関数

ILEN(I) ビット長(組込み)  
LEADZ(I) 先頭の 0 ビットの数  
POPCNT(I) ビット 1 の数  
POPPAR(I) ビット 1 の奇偶

## 7.5.4 配列集計関数

1 IALL(ARRAY, DIM, MASK) ビット論理積集計  
2 省略可能な引数 DIM, MASK  
3  
4 IANY(ARRAY, DIM, MASK) ビット論理和集計  
5 省略可能な引数 DIM, MASK  
6  
7 IPARITY(ARRAY, DIM, MASK) ビット排他的論理和集計  
8 省略可能な引数 DIM, MASK  
9  
10 PARITY(MASK, DIM) 排他的論理和集計  
11 省略可能な引数 DIM

## 7.5.5 配列集計拡散関数

12  
13  
14 ALL\_SCATTER(MASK, BASE, INDX1 ..., INDXn)  
15 ANY\_SCATTER(MASK, BASE, INDX1, ..., INDXn)  
16 COPY\_SCATTER(ARRAY, BASE, INDX1, ..., INDXn, MASK)  
17 省略可能な引数 MASK  
18  
19 COUNT\_SCATTER(MASK, BASE, INDX1, ..., INDXn)  
20 IALL\_SCATTER(ARRAY, BASE, INDX1, ..., INDXn, MASK)  
21 省略可能な引数 MASK  
22  
23 IANY\_SCATTER(ARRAY, BASE, INDX1, ..., INDXn, MASK)  
24 省略可能な引数 MASK  
25  
26 IPARITY\_SCATTER(ARRAY, BASE, INDX1, ..., INDXn, MASK)  
27 省略可能な引数 MASK  
28  
29 MAXVAL\_SCATTER(ARRAY, BASE, INDX1, ..., INDXn, MASK)  
30 省略可能な引数 MASK  
31  
32 MINVAL\_SCATTER(ARRAY, BASE, INDX1, ..., INDXn, MASK)  
33 省略可能な引数 MASK  
34  
35 PARITY\_SCATTER(MASK, BASE, INDX1, ..., INDXn)  
36  
37 PRODUCT\_SCATTER(ARRAY, BASE, INDX1, ..., INDXn, MASK)  
38 省略可能な引数 MASK  
39  
40 SUM\_SCATTER(ARRAY, BASE, INDX1, ..., INDXn, MASK)  
41 省略可能な引数 MASK

## 7.5.6 配列累計関数

42 ALL\_PREFIX(MASK, DIM, SEGMENT, EXCLUSIVE)  
43 省略可能な引数 DIM, SEGMENT, EXCLUSIVE  
44  
45 ALL\_SUFFIX(MASK, DIM, SEGMENT, EXCLUSIVE)  
46 省略可能な引数 DIM, SEGMENT, EXCLUSIVE  
47  
48 ANY\_PREFIX(MASK, DIM, SEGMENT, EXCLUSIVE)  
49 省略可能な引数 DIM, SEGMENT, EXCLUSIVE  
50  
51 ANY\_SUFFIX(MASK, DIM, SEGMENT, EXCLUSIVE)  
52 省略可能な引数 DIM, SEGMENT, EXCLUSIVE

COPY_PREFIX (ARRAY, DIM, SEGMENT)	1
省略可能な引数 DIM, SEGMENT	2
COPY_SUFFIX (ARRAY, DIM, SEGMENT)	3
省略可能な引数 DIM, SEGMENT	4
COUNT_PREFIX (MASK, DIM, SEGMENT, EXCLUSIVE)	5
省略可能な引数 DIM, SEGMENT, EXCLUSIVE	6
COUNT_SUFFIX (MASK, DIM, SEGMENT, EXCLUSIVE)	7
省略可能な引数 DIM, SEGMENT, EXCLUSIVE	8
IALL_PREFIX (ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)	9
省略可能な引数 DIM, MASK, SEGMENT, EXCLUSIVE	10
IALL_SUFFIX (ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)	11
省略可能な引数 DIM, MASK, SEGMENT, EXCLUSIVE	12
IANY_PREFIX (ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)	13
省略可能な引数 DIM, MASK, SEGMENT, EXCLUSIVE	14
IANY_SUFFIX (ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)	15
省略可能な引数 DIM, MASK, SEGMENT, EXCLUSIVE	16
IPARITY_PREFIX (ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)	17
省略可能な引数 DIM, MASK, SEGMENT, EXCLUSIVE	18
IPARITY_SUFFIX (ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)	19
省略可能な引数 DIM, MASK, SEGMENT, EXCLUSIVE	20
MAXVAL_PREFIX (ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)	21
省略可能な引数 DIM, MASK, SEGMENT, EXCLUSIVE	22
MAXVAL_SUFFIX (ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)	23
省略可能な引数 DIM, MASK, SEGMENT, EXCLUSIVE	24
MINVAL_PREFIX (ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)	25
省略可能な引数 DIM, MASK, SEGMENT, EXCLUSIVE	26
MINVAL_SUFFIX (ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)	27
省略可能な引数 DIM, MASK, SEGMENT, EXCLUSIVE	28
PARITY_PREFIX (MASK, DIM, SEGMENT, EXCLUSIVE)	29
省略可能な引数 DIM, SEGMENT, EXCLUSIVE	30
PARITY_SUFFIX (MASK, DIM, SEGMENT, EXCLUSIVE)	31
省略可能な引数 DIM, SEGMENT, EXCLUSIVE	32
PRODUCT_PREFIX (ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)	33
省略可能な引数 DIM, MASK, SEGMENT, EXCLUSIVE	34
PRODUCT_SUFFIX (ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)	35
省略可能な引数 DIM, MASK, SEGMENT, EXCLUSIVE	36
SUM_PREFIX (ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)	37
省略可能な引数 DIM, MASK, SEGMENT, EXCLUSIVE	38
SUM_SUFFIX (ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)	39
省略可能な引数 DIM, MASK, SEGMENT, EXCLUSIVE	40
SUM_PREFIX (ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)	41
省略可能な引数 DIM, MASK, SEGMENT, EXCLUSIVE	42
SUM_SUFFIX (ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)	43
省略可能な引数 DIM, MASK, SEGMENT, EXCLUSIVE	44
	45
	46
	47
	48

## 7.5.7 配列ソート関数

GRADE_DOWN (ARRAY, DIM)	降順ソートによる添字の置換
省略可能な引数 DIM	
GRADE_UP (ARRAY, DIM)	昇順ソートによる添字の置換
省略可能な引数 DIM	
SORT_DOWN (ARRAY, DIM)	降順ソート
省略可能な引数 DIM	
SORT_UP (ARRAY, DIM)	昇順ソート
省略可能な引数 DIM	

## 7.6 組込み手続の仕様

### ILEN(I)

機能. 整数の 2 の補数表現のビット長より 1 小さい値を返す。

分類. 要素別処理関数

引数. I は整数型でなければならない。

結果の型 及び 型パラメタ. I と同じとする。

結果の値. I が非負のとき、ILEN(I) の値は  $\lceil \log_2(I + 1) \rceil$  とする。I が負のとき、ILEN(I) の値は  $\lceil \log_2(-I) \rceil$  とする。

例示. ILEN(4) = 3 である。また ILEN(-4) = 2 である。2\*\*ILEN(N-1) により、N は 2 のべき乗に切り上げられ (ただし N > 0 とする)、2\*\*(ILEN(N)-1) により、N は 2 のべき乗に切り捨てられる。LEADZ と比較されたい。

結果の値は、I の 2 の補数表現の長さより 1 小さい値である。4 の最も短い 2 の補数表現は、0100 である。先頭の 0 は、符号ビットとして必要である。3 ビットの 2 の補数表現 100 は -4 を表す。

### NUMBER\_OF\_PROCESSORS(DIM)

省略可能な引数. DIM

機能. プログラムで利用可能なプロセッサの総数、又はプロセッサ配列の指定された次元のプログラムで利用可能なプロセッサ数を返す。

分類. システム問合せ関数

引数.

DIM (省略可能) 整数型スカラであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならない。ここで  $n$  は、プロセッサ配列の次元数とする。

結果の型、型パラメタ 及び 形状. 基本整数型スカラとする。

結果の値. 結果の値は、実装依存のハードウェアプロセッサ配列の第 DIM 次元の寸法とする。DIM が省略されたときは、実装依存のハードウェアプロセッサ配列の要素の総数とする。結果は常に 0 より大きいものとする。

例示. 8192 台のプロセッサが  $128 \times 64$  の矩形格子状に編成されたコンピュータでは、NUMBER\_OF\_PROCESSORS() の値は 8192 になる。また NUMBER\_OF\_PROCESSORS(DIM=1) の値は 128 になり、NUMBER\_OF\_PROCESSORS(DIM=2) の値は 64 になる。単一プロセッサのワークステーションでは、NUMBER\_OF\_PROCESSORS() の値は 1 になる。スカラのプロセッサ配列の次元数は 0 なので、引数 DIM を指定することはできない。

## PROCESSORS\_SHAPE()

機能. 実装依存のプロセッサ配列の形状を返す。

分類. システム問合せ関数

引数. なし

結果の型、型パラメタ 及び 形状. 基本整数型 1 次元配列で、実装依存のプロセッサ配列の次元数に等しい大きさとする。

結果の値. 結果の値は、実装依存のプロセッサ配列の形状とする。

例示. 2048 台のプロセッサがハイパーキューブ状に編成されたコンピュータでは、PROCESSORS\_SHAPE() の値は [2,2,2,2,2,2,2,2,2,2] になる。8192 台のプロセッサが  $128 \times 64$  の矩形格子状に編成されたコンピュータでは、PROCESSORS\_SHAPE() の値は [128,64] になる。単一プロセッサのワークステーションでは、PROCESSORS\_SHAPE() の値は [] になる (大きさ 0 の 1 次元配列)。

## 7.7 ライブラリ手続の仕様

### ALL\_PREFIX(MASK, DIM, SEGMENT, EXCLUSIVE)

省略可能な引数. DIM, SEGMENT, EXCLUSIVE

機能. MASK の DIM 番目の次元について、区分ごとに順方向の論理積累計を計算する。

分類. 変型関数

引数.

MASK 論理型でなければならず、スカラであってはならない。

DIM (省略可能) 整数型スカラであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならず。ここで  $n$  は、MASK の次元数とする。

1 SEGMENT (省略可能) 論理型でなければならず、MASK と同じ形状でなければならない。

2 EXCLUSIVE (省略可能) 論理型スカラでなければならない。

3  
4 結果の型、型パラメタ 及び 形状. MASK と同じとする。

5  
6 結果の値. 結果の要素  $r$  の値は、 $ALL((/ a_1, \dots, a_m /))$  とする。ここで、 $(a_1, \dots, a_m)$   
7 は、7.4.5 項に述べた規則により、 $r$  に寄与すべく選択された MASK の要素の集合であ  
8 る (空集合のこともある)。

9  
10 例示.  $ALL\_PREFIX( (/T,F,T,T,T/), SEGMENT= (/F,F,F,T,T/))$  の値は  
11  $\begin{bmatrix} T & F & F & T & T \end{bmatrix}$  となる。  
12

## 13 14 ALL\_SCATTER(MASK, BASE, INDX1, ..., INDXn)

15  
16 機能. MASK の要素を、指標配列 INDX1, ..., INDXn で指定される結果の位置に拡散す  
17 る。結果の要素は、BASE の対応する要素及びその位置に拡散される MASK の要素がす  
18 べて真であるとき、かつこのときに限り、真となる。

19  
20 分類. 変型関数

21  
22 引数.

23  
24 MASK 論理型でなければならず、スカラであってはならない。

25  
26 BASE MASK と同じ種別型パラメタの論理型でなければならず、スカラであっては  
27 ならない。

28  
29 INDX1, ..., INDXn 整数型でなければならず、MASK と形状適合しなければならない。  
30 引数 INDX の数は、BASE の次元数と等しくなければならない。

31  
32 結果の型、型パラメタ 及び 形状. BASE と同じとする。

33  
34 結果の値. BASE の要素  $b$  に対応する結果の要素の値は、 $ALL( (/a_1, a_2, \dots, a_m, b/))$   
35 とする。ここで  $(a_1, \dots, a_m)$  は 7.4.4 項に述べた方法で  $b$  と関連付けられる MASK の要  
36 素とする。

37  
38 例示.  $ALL\_SCATTER((/ T, T, T, F /), (/ T, T, T /), (/ 1, 1, 2, 2 /))$  は  
39  $\begin{bmatrix} T & F & T \end{bmatrix}$  となる。  
40

## 41 ALL\_SUFFIX(MASK, DIM, SEGMENT, EXCLUSIVE)

42  
43 省略可能な引数. DIM, SEGMENT, EXCLUSIVE

44  
45 機能. MASK の DIM 番目の次元について、区分ごとに逆方向の論理積累計を計算する。

46  
47 分類. 変型関数  
48

引数. 1

MASK 論理型でなければならず、スカラであってはならない。 2

DIM (省略可能) 整数型スカラであって、 $1 \leq DIM \leq n$  の範囲の値でなければならず。 3

ここで  $n$  は、MASK の次元数とする。 4

SEGMENT (省略可能) 論理型でなければならず、MASK と同じ形状でなければならず。 5

EXCLUSIVE (省略可能) 論理型スカラでなければならず。 6

結果の型、型パラメタ 及び 形状. MASK と同じとする。 7

結果の値. 結果の要素  $r$  の値は、 $ALL((/ a_1, \dots, a_m /))$  とする。ここで、 $(a_1, \dots, a_m)$  8

は、7.4.5 項に述べた規則により、 $r$  に寄与すべく選択された MASK の要素の集合である (空集合のこともある)。 9

例示. ALL\_SUFFIX( (/T,F,T,T,T/), SEGMENT= (/F,F,F,T,T/) ) の値は 10

$\begin{bmatrix} F & F & T & T & T \end{bmatrix}$  となる。 11

## ANY\_PREFIX(MASK, DIM, SEGMENT, EXCLUSIVE)

省略可能な引数. DIM, SEGMENT, EXCLUSIVE 12

機能. MASK の DIM 番目の次元について、区分ごとに順方向の論理和累計を計算する。 13

分類. 変型関数 14

引数. 15

MASK 論理型でなければならず、スカラであってはならない。 16

DIM (省略可能) 整数型スカラであって、 $1 \leq DIM \leq n$  の範囲の値でなければならず。 17

ここで  $n$  は、MASK の次元数とする。 18

SEGMENT (省略可能) 論理型でなければならず、MASK と同じ形状でなければならず。 19

EXCLUSIVE (省略可能) 論理型スカラでなければならず。 20

結果の型、型パラメタ 及び 形状. MASK と同じとする。 21

結果の値. 結果の要素  $r$  の値は、 $ANY((/ a_1, \dots, a_m /))$  とする。ここで、 $(a_1, \dots, a_m)$  22

は、7.4.5 項に述べた規則により、 $r$  に寄与すべく選択された MASK の要素の集合である (空集合のこともある)。 23

例示. ANY\_PREFIX( (/F,T,F,F,F/), SEGMENT= (/F,F,F,T,T/) ) の値は 24

$\begin{bmatrix} F & T & T & F & F \end{bmatrix}$  となる。 25

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48



## 1 ANY\_SCATTER(MASK, BASE, INDX1, ..., INDXn)

2 機能. MASK の要素を、指標配列 INDX1, ..., INDXn で指定される結果の位置に拡散す  
3 る。結果の要素は、BASE の対応する要素またはその位置に拡散される MASK の要素の  
4 中に一つでも真があるとき、かつこのときに限り、真となる。  
5

6 分類. 変型関数

7 引数.

8 MASK 論理型でなければならず、スカラであってはならない。

9  
10 BASE MASK と同じ種別型パラメタの論理型でなければならず、スカラであっては  
11 ならない。  
12

13 INDX1, ..., INDXn 整数型でなければならず、MASK と形状適合しなければならない。  
14 引数 INDX の数は、BASE の次元数と等しくなければならない。  
15

16 結果の型、型パラメタ 及び 形状. BASE と同じとする。  
17

18 結果の値. BASE の要素  $b$  に対応する結果の要素の値は、 $ANY( (/a_1, a_2, \dots, a_m, b/ )$   
19 とする。ここで  $(a_1, \dots, a_m)$  は 7.4.4 項に述べた方法で  $b$  と関連付けられる MASK の要  
20 素とする。  
21

22 例示.  $ANY\_SCATTER( (/ T, F, F, F / ), (/ F, F, T / ), (/ 1, 1, 2, 2 / ))$  は  
23  $\begin{bmatrix} T & F & T \end{bmatrix}$  となる。  
24

## 25 ANY\_SUFFIX(MASK, DIM, SEGMENT, EXCLUSIVE)

26 省略可能な引数. DIM, SEGMENT, EXCLUSIVE

27 機能. MASK の DIM 番目の次元について、区分ごとに逆方向の論理和累計を計算する。  
28

29 分類. 変型関数

30 引数.

31 MASK 論理型でなければならず、スカラであってはならない。  
32

33 DIM (省略可能) 整数型スカラであって、 $1 \leq DIM \leq n$  の範囲の値でなければならない。  
34 ここで  $n$  は、MASK の次元数とする。  
35

36 SEGMENT (省略可能) 論理型でなければならず、MASK と同じ形状でなければならない。  
37

38 EXCLUSIVE (省略可能) 論理型スカラでなければならない。  
39

40 結果の型、型パラメタ 及び 形状. MASK と同じとする。  
41

42 結果の値. 結果の要素  $r$  の値は、 $ANY( (/ a_1, \dots, a_m / ))$  とする。ここで、 $(a_1, \dots, a_m)$   
43 は、7.4.5 項に述べた規則により、 $r$  に寄与すべく選択された MASK の要素の集合であ  
44 る (空集合のこともある)。  
45

46 例示.  $ANY\_SUFFIX( (/ F, T, F, F, F / ), SEGMENT= (/ F, F, F, T, T / ))$  の値は  
47  $\begin{bmatrix} T & T & F & F & F \end{bmatrix}$  となる。  
48

## COPY\_PREFIX(Array, DIM, SEGMENT)

省略可能な引数. DIM, SEGMENT

機能. ARRAY の DIM 番目の次元について、区分ごとに順方向の複写累計を計算する。

分類. 変型関数

引数.

ARRAY どの型でもよい。スカラであってはならない。

DIM (省略可能) 整数型スカラであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならない。  
ここで  $n$  は、ARRAY の次元数とする。

SEGMENT (省略可能) 論理型でなければならない、ARRAY と同じ形状でなければならない。

結果の型、型パラメタ 及び 形状. ARRAY と同じとする。

結果の値. 結果の要素  $r$  の値は、 $a_1$  とする。ここで、 $(a_1, \dots, a_m)$  は、7.4.5 項に述べた規則により、 $r$  に寄与すべく選択された ARRAY の要素の、配列要素順序による集合である。この集合は決して空集合にはならない。

例示. `COPY_PREFIX( (/1,2,3,4,5/), SEGMENT= (/F,F,F,T,T/) )` の値は  
$$\begin{bmatrix} 1 & 1 & 1 & 4 & 4 \end{bmatrix}$$
 となる。

## COPY\_SCATTER(Array, BASE, INDX1, ..., INDXn, MASK)

省略可能な引数. MASK

機能. MASK で選択される ARRAY の要素を、指標配列  $\text{INDX1}, \dots, \text{INDXn}$  で指定される結果の位置に拡散する。結果の各要素は、その位置に拡散される ARRAY の要素のうちの一つに等しいか、またはそのような要素がないときは、BASE の対応する要素に等しい。

分類. 変型関数

引数.

ARRAY どの型でもよい。スカラであってはならない。

BASE ARRAY と同じ型及び同じ種別型パラメタでなければならない。

$\text{INDX1}, \dots, \text{INDXn}$  整数型でなければならない、ARRAY と形状適合しなければならない。  
引数  $\text{INDX}$  の数は、BASE の次元数と等しくなければならない。

MASK (省略可能) 論理型でなければならない、ARRAY と形状適合しなければならない。

結果の型、型パラメタ 及び 形状. BASE と同じとする。

1 結果の値. 7.4.4 項に述べた方法で BASE の要素  $b$  と関連付けられる ARRAY の要素の集  
2 合を  $S$  とする。

3  $S$  が空のとき、BASE の要素  $b$  に対応する結果の要素の値は、 $b$  と同じとする。

4  $S$  が空でないとき、BASE の要素  $b$  に対応する結果の要素の値は、 $S$  から 1 つの要素を  
5 選択した結果とする。HPF はこの選択方法を規定せず、実装依存とする。

7  
8 例示. COPY\_SCATTER((/ 1, 2, 3, 4 /), (/ 7, 8, 9 /), (/ 1, 1, 2, 2 /))  
9 は  $[x, y, 9]$  となる。ここで  $x$  は集合  $\{1, 2\}$  の要素の 1 つであり、 $y$  は  $\{3, 4\}$  の要素  
10 の 1 つである。

## 12 COPY\_SUFFIX(ARRAY, DIM, SEGMENT)

13  
14 省略可能な引数. DIM, SEGMENT

15  
16 機能. ARRAY の DIM 番目の次元について、区分ごとに逆方向の複写累計を計算する。

17  
18 分類. 変型関数

19  
20 引数.

21  
22 ARRAY どの型でもよい。スカラであってはならない。

23  
24 DIM (省略可能) 整数型スカラであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならない。  
25 ここで  $n$  は、ARRAY の次元数とする。

26  
27 SEGMENT (省略可能) 論理型でなければならない、ARRAY と同じ形状でなければならない。

28  
29 結果の型、型パラメタ 及び 形状. ARRAY と同じとする。

30  
31 結果の値. 結果の要素  $r$  の値は、 $a_m$  とする。ここで、 $(a_1, \dots, a_m)$  は、7.4.5 項に述べ  
32 た規則により、 $r$  に寄与すべく選択された ARRAY の要素の、配列要素順序による集合  
33 である。この集合は決して空集合にはならない。

34  
35 例示. COPY\_SUFFIX( (/1,2,3,4,5/), SEGMENT= (/F,F,F,T,T/) ) の値は  
36  $\begin{bmatrix} 3 & 3 & 3 & 5 & 5 \end{bmatrix}$  となる。

## 38 COUNT\_PREFIX(MASK, DIM, SEGMENT, EXCLUSIVE)

39  
40 省略可能な引数. DIM, SEGMENT, EXCLUSIVE

41  
42 機能. MASK の DIM 番目の次元について、区分ごとに順方向の COUNT 累計を計算する。

43  
44 分類. 変型関数

45  
46 引数.

47  
48 MASK 論理型でなければならない、スカラであってはならない。

DIM (省略可能) 整数型スカラであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならない。  
ここで  $n$  は、MASK の次元数とする。

SEGMENT (省略可能) 論理型でなければならない、MASK と同じ形状でなければならない。

EXCLUSIVE (省略可能) 論理型スカラでなければならない。

結果の型、型パラメタ 及び 形状. 基本整数型であって、MASK と同じ形状の配列とする。

結果の値. 結果の要素  $r$  の値は、 $\text{COUNT}((/ a_1, \dots, a_m /))$  とする。ここで、 $(a_1, \dots, a_m)$  は、7.4.5 項に述べた規則により、 $r$  に寄与すべく選択された MASK の要素の集合である (空集合のこともある)。

例示.  $\text{COUNT\_PREFIX}((/ F, T, T, T, T /), \text{SEGMENT} = (/ F, F, F, T, T /))$  の値は  
 $\begin{bmatrix} 0 & 1 & 2 & 1 & 2 \end{bmatrix}$  となる。

### COUNT\_SCATTER(MASK, BASE, INDX1, ..., INDXn)

機能. MASK の要素を、指標配列 INDX1, ..., INDXn で指定される結果の位置に拡散する。結果の各要素は、BASE の対応する要素と、その位置に拡散される MASK 中の真である要素の個数の和となる。

分類. 変型関数

引数.

MASK 論理型でなければならない、スカラであってはならない。

BASE 整数型でなければならない、スカラであってはならない。

INDX1, ..., INDXn 整数型でなければならない、MASK と形状適合しなければならない。  
引数 INDX の数は、BASE の次元数と等しくなければならない。

結果の型、型パラメタ 及び 形状. BASE と同じとする。

結果の値. BASE の要素  $b$  に対応する結果の要素の値は、 $b + \text{COUNT}((/ a_1, a_2, \dots, a_m /))$  とする。ここで  $(a_1, \dots, a_m)$  は 7.4.4 項に述べた方法で  $b$  と関連付けられる ARRAY の要素とする。

例示.  $\text{COUNT\_SCATTER}((/ T, T, T, F /), (/ 1, -1, 0 /), (/ 1, 1, 2, 2 /))$   
は  $\begin{bmatrix} 3 & 0 & 0 \end{bmatrix}$  となる。

### COUNT\_SUFFIX(MASK, DIM, SEGMENT, EXCLUSIVE)

省略可能な引数. DIM, SEGMENT, EXCLUSIVE

機能. MASK の DIM 番目の次元について、区分ごとに逆方向の COUNT 累計を計算する。

1 分類. 変型関数

2  
3 引数.

4 MASK 論理型でなければならず、スカラであってはならない。

5  
6 DIM (省略可能) 整数型スカラであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならず。  
7 ここで  $n$  は、MASK の次元数とする。

8  
9 SEGMENT (省略可能) 論理型でなければならず、MASK と同じ形状でなければならず。

10 EXCLUSIVE (省略可能) 論理型スカラでなければならず。

11  
12 結果の型、型パラメタ 及び 形状. 基本整数型であって、MASK と同じ形状の配列と  
13 する。

14  
15 結果の値. 結果の要素  $r$  の値は、 $\text{COUNT}((/ a_1, \dots, a_m /))$  とする。ここで、  
16  $(a_1, \dots, a_m)$  は、7.4.5 項に述べた規則により、 $r$  に寄与すべく選択された MASK の要  
17 素の集合である (空集合のこともある)。

18  
19 例示.  $\text{COUNT\_SUFFIX}((/T,F,T,T,T/), \text{SEGMENT}=(/F,F,F,T,T/))$  の値は  
20  $\begin{bmatrix} 2 & 1 & 1 & 2 & 1 \end{bmatrix}$  となる。  
21

## 22 23 GRADE\_DOWN(ARRAY,DIM)

24 省略可能な引数. DIM

25  
26 機能. 要素の値の降順ソートを表現する、基底を 1 とした配列添字の置換列を生成する。

27  
28 分類. 変型関数

29  
30 引数.

31  
32 ARRAY 整数型、実数型又は文字型でなければならず、スカラであってはならない。

33  
34 DIM (省略可能) 整数型スカラであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならず。  
35 ここで  $n$  は、ARRAY の次元数とする。対応する実引数は、省略可能な仮引  
36 数であってはならない。

37  
38 結果の型、型パラメタ 及び 形状. 基本整数型とする。DIM を指定したとき、ARRAY と  
39 同じ形状とする。DIM を省略したとき、 $(/ \text{SIZE}(\text{SHAPE}(\text{ARRAY})), \text{SIZE}(\text{ARRAY}) /)$   
40 という形状とする。  
41

42 結果の値.

43  
44 場合 (i):  $S = \text{GRADE\_DOWN}(\text{ARRAY})$   
45  $+ \text{SPREAD}(\text{LBOUND}(\text{ARRAY}), \text{DIM}=2, \text{NCOPIES}=\text{SIZE}(\text{ARRAY}))-1$   
46 の結果は、大きさ  $\text{SIZE}(\text{ARRAY})$  の 1 次元配列 B を  
47  $\text{FORALL} (K=1:\text{SIZE}(B)) B(K)=\text{ARRAY}(S(1,K), S(2,K), \dots, S(N,K))$   
48

によって計算したとき、Bが降順にソートされるような値とする。ここで、NはSIZE(SHAPE(ARRAY))とする。さらに、Sのすべての列は異なるものとする。すなわち、 $j \neq m$ ならば $ALL(S(:,j) .EQ. S(:,m))$ は偽となる。また、ソートは安定である。すなわち $j \leq m$ かつ $B(j) = B(m)$ ならば、ARRAYの配列要素順序で $ARRAY(S(1,j), S(2,j), \dots, S(n,j))$ が $ARRAY(S(1,m), S(2,m), \dots, S(n,m))$ に先行する。文字型配列の大小順序は、Fortranの組込み関数で使われるもの、すなわちASCII大小順序とする。

場合 (ii):  $R = \text{GRADE\_DOWN}(\text{ARRAY}, \text{DIM}=\text{K}) + \text{LBOUND}(\text{ARRAY}, \text{DIM}=\text{K}) - 1$

の結果は、配列Bを

$$B(i_1, i_2, \dots, i_k, \dots, i_n) =$$

$$\text{ARRAY}(i_1, i_2, \dots, R(i_1, i_2, \dots, i_k, \dots, i_n), \dots, i_n)$$

によって計算したとき、すべての $i_1, i_2, \dots, (i_k$ を除く),  $\dots, i_n$ についてベクトル $B(i_1, i_2, \dots, :, \dots, i_n)$ が降順にソートされるような値とする。さらに、 $R(i_1, i_2, \dots, :, \dots, i_n)$ は1から $\text{SIZE}(\text{ARRAY}, \text{DIM}=\text{K})$ までのすべての整数を並べ替えたものとする。また、ソートは安定である。すなわち $j \leq m$ かつ $B(i_1, i_2, \dots, j, \dots, i_n) = B(i_1, i_2, \dots, m, \dots, i_n)$ ならば、 $R(i_1, i_2, \dots, j, \dots, i_n) \leq R(i_1, i_2, \dots, m, \dots, i_n)$ となる。文字型配列の大小順序は、Fortranの組込み関数で使われるもの、すなわちASCII大小順序とする。

例示.

場合 (i):  $\text{GRADE\_DOWN}((/30, 20, 30, 40, -10/))$ は、形状が $\begin{bmatrix} 1 & 5 \end{bmatrix}$ の2次元配列で、値は $\begin{bmatrix} 4 & 1 & 3 & 2 & 5 \end{bmatrix}$ になる(結果を1次元配列にするためには、省略可能な引数 $\text{DIM} = 1$ を指定しなければならない)。

$$\text{配列 A が } \begin{bmatrix} 1 & 9 & 2 \\ 4 & 5 & 2 \\ 1 & 2 & 4 \end{bmatrix}$$

であるとき、 $\text{GRADE\_DOWN}(A)$ の値は、

$$\begin{bmatrix} 1 & 2 & 2 & 3 & 3 & 1 & 2 & 1 & 3 \\ 2 & 2 & 1 & 3 & 2 & 3 & 3 & 1 & 1 \end{bmatrix}$$

になる。

場合 (ii): 配列 A が  $\begin{bmatrix} 1 & 9 & 2 \\ 4 & 5 & 2 \\ 1 & 2 & 4 \end{bmatrix}$

であるとき、 $\text{GRADE\_DOWN}(A, \text{DIM} = 1)$ の値は  $\begin{bmatrix} 2 & 1 & 3 \\ 1 & 2 & 1 \\ 3 & 3 & 2 \end{bmatrix}$

になり、

$\text{GRADE\_DOWN}(A, \text{DIM} = 2)$ の値は  $\begin{bmatrix} 2 & 3 & 1 \\ 2 & 1 & 3 \\ 3 & 2 & 1 \end{bmatrix}$

になる。

# GRADE\_UP (ARRAY, DIM)

省略可能な引数. DIM

機能. 要素の値の昇順ソートを表現する、基底を 1 とした配列添字の置換列を生成する。

分類. 変型関数

引数.

ARRAY 整数型、実数型又は文字型でなければならず、スカラーであってはならない。

DIM (省略可能) 整数型スカラーであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならず。  
ここで  $n$  は、ARRAY の次元数とする。対応する実引数は、省略可能な仮引数であってはならない。

結果の型、型パラメタ 及び 形状. 基本整数型とする。DIM を指定したとき、ARRAY と同じ形状とする。DIM を省略したとき、( / SIZE(SHAPE(ARRAY)), SIZE(ARRAY) / ) という形状とする。

結果の値.

場合 (i):  $S = \text{GRADE\_UP}(\text{ARRAY})$

+ SPREAD (LBOUND (ARRAY), DIM=2, NCOPIES=SIZE(ARRAY))-1

の結果は、大きさ SIZE(ARRAY) の 1 次元配列 B を

FORALL (K=1:SIZE(B)) B(K)=ARRAY(S(1,K),S(2,K),...,S(N,K))

によって計算したとき、B が昇順にソートされるような値とする。ここで、N は SIZE(SHAPE(ARRAY)) とする。さらに、S のすべての列は異なるものとする。すなわち、 $j \neq m$  ならば  $\text{ALL}(S(:,j) \text{ .EQ. } S(:,m))$  は偽となる。また、ソートは安定である。すなわち  $j \leq m$  かつ  $B(j) = B(m)$  ならば、ARRAY の配列要素順序で  $\text{ARRAY}(S(1,j), S(2,j), \dots, S(n,j))$  が  $\text{ARRAY}(S(1,m), S(2,m), \dots, S(n,m))$  に先行する。文字型配列の大小順序は、Fortran の組込み関数で使われるもの、すなわち ASCII 大小順序とする。

場合 (ii):  $R = \text{GRADE\_UP}(\text{ARRAY}, \text{DIM}=K) + \text{LBOUND}(\text{ARRAY}, \text{DIM}=K) - 1$

の結果は、配列 B を

$B(i_1, i_2, \dots, i_k, \dots, i_n) =$

$\text{ARRAY}(i_1, i_2, \dots, R(i_1, i_2, \dots, i_k, \dots, i_n), \dots, i_n)$

によって計算したとき、すべての  $i_1, i_2, \dots, (i_k \text{ を除く}), \dots, i_n$  についてベクトル  $B(i_1, i_2, \dots, :, \dots, i_n)$  が昇順にソートされるような値とする。さらに、 $R(i_1, i_2, \dots, :, \dots, i_n)$  は 1 から SIZE(ARRAY, DIM=K) までのすべての整数を並べ替えたものとする。また、ソートは安定である。すなわち  $j \leq m$  かつ  $B(i_1, i_2, \dots, j, \dots, i_n) = B(i_1, i_2, \dots, m, \dots, i_n)$  ならば、 $R(i_1, i_2, \dots, j, \dots, i_n) \leq R(i_1, i_2, \dots, m, \dots, i_n)$  となる。文字型配列の大小順序は、Fortran の組込み関数で使われるもの、すなわち ASCII 大小順序とする。

例示.

場合 (i): `GRADE_UP( (/30, 20, 30, 40, -10/) )` は、形状が  $\begin{bmatrix} 1 & 5 \end{bmatrix}$  の 2次元配列で、値は  $\begin{bmatrix} 5 & 2 & 1 & 3 & 4 \end{bmatrix}$  になる (結果を 1次元配列にするためには、省略可能な引数 `DIM = 1` を指定しなければならない)。

配列 A が  $\begin{bmatrix} 1 & 9 & 2 \\ 4 & 5 & 2 \\ 1 & 2 & 4 \end{bmatrix}$  であるとき、`GRADE_UP(A)` の値は、 $\begin{bmatrix} 1 & 3 & 3 & 1 & 2 & 2 & 3 & 2 & 1 \\ 1 & 1 & 2 & 3 & 3 & 1 & 3 & 2 & 2 \end{bmatrix}$  になる。

場合 (ii): 配列 A が  $\begin{bmatrix} 1 & 9 & 2 \\ 4 & 5 & 2 \\ 1 & 2 & 4 \end{bmatrix}$

であるとき、`GRADE_UP(A, DIM = 1)` の値は  $\begin{bmatrix} 1 & 3 & 1 \\ 3 & 2 & 2 \\ 2 & 1 & 3 \end{bmatrix}$  になり、

`GRADE_UP(A, DIM = 2)` の値は  $\begin{bmatrix} 1 & 3 & 2 \\ 3 & 1 & 2 \\ 1 & 2 & 3 \end{bmatrix}$  になる。

## HPF\_ALIGNMENT(ALIGNEE, LB, UB, STRIDE, AXIS\_MAP, IDENTITY\_MAP, NCOPIES)

省略可能な引数. `LB, UB, STRIDE, AXIS_MAP, IDENTITY_MAP, NCOPIES`

機能. 変数と、それが最終的に整列する *align-target* (配列またはテンプレート) との対応に関する情報を返す。

分類. マッピング問合せサブルーチン

引数.

**ALIGNEE** どの型でもよく、スカラーでも配列でもよい。大きさ引継ぎ配列であってはならない。結合変数グループのメンバであるとき、そのグループの結合被覆でなければならない (「結合変数グループ」及び「結合被覆」の定義については 3.8 項を参照されたい)<sup>1</sup>。空状態のポインタ又は割り付けられていない割付け配列であってはならない。INTENT (IN) 引数とする。

`ALIGNEE` がポインタであるときは、指示先の整列に関する情報が返されるものとする。指示先は大きさ引継ぎの仮引数又は大きさ引継ぎの仮引数の部分配列であってはならない。

<sup>1</sup> 訳注: 原文の誤り。HPF 2.0 では「結合被覆」の定義は抹消されている。`ALIGNEE` は結合変数グループのメンバであってはならないと解釈すべきである。



1 LB (省略可能) 基本整数型の 1 次元配列でなければならない。少なくとも ALIGNEE の  
2 次元数に等しい大きさを持たなければならない。INTENT (OUT) 引数とす  
3 る。ALIGNEE の  $i$  番目の次元軸の最初の要素が、対応する *align-target* の次  
4 元軸の LB( $i$ ) 番目の要素に最終的に整列している。ALIGNEE の  $i$  番目の次  
5 元軸が縮退しているとき、LB( $i$ ) に設定される値は実装依存とする。

6  
7 UB (省略可能) 基本整数型の 1 次元配列でなければならない。少なくとも ALIGNEE の  
8 次元数に等しい大きさを持たなければならない。INTENT (OUT) 引数とす  
9 る。ALIGNEE の  $i$  番目の次元軸の最後の要素が、対応する *align-target* の次  
10 元軸の UB( $i$ ) 番目の要素に最終的に整列している。ALIGNEE の  $i$  番目の次  
11 元軸が縮退しているとき、UB( $i$ ) に設定される値は実装依存とする。

12  
13 STRIDE (省略可能) 基本整数型の 1 次元配列でなければならない。少なくとも ALIGNEE  
14 の次元数に等しい大きさを持たなければならない。INTENT (OUT) 引数と  
15 する。STRIDE の  $i$  番目の要素には、ALIGNEE の  $i$  番目の次元軸に沿って各  
16 要素が整列するときの刻み幅が設定される。ALIGNEE の  $i$  番目の次元軸が  
17 縮退しているとき、STRIDE( $i$ ) は 0 に設定される。

18  
19 AXIS\_MAP (省略可能) 基本整数型の 1 次元配列でなければならない。少なくとも  
20 ALIGNEE の次元数に等しい大きさを持たなければならない。INTENT (OUT)  
21 引数とする。AXIS\_MAP の  $i$  番目の要素には、ALIGNEE の  $i$  番目の次元軸に  
22 対応する *align-target* の次元軸が設定される。ALIGNEE の  $i$  番目の次元軸が  
23 縮退しているとき、AXIS\_MAP( $i$ ) は 0 に設定される。

24  
25 IDENTITY\_MAP (省略可能) 基本論理型スカラーでなければならない。INTENT (OUT) 引  
26 数とする。ALIGNEE の最終 *align-target* が ALIGNEE と同じ形状を持ち、次  
27 元軸が恒等置換を使ってマップされ、刻み幅がすべて正である (したがって  
28 形状の制限から 1 に等しい) とき、真が設定される。それ以外するとき、偽  
29 が設定される。変数が、ALIGN 指示文や REALIGN 指示文の *alignee* に現れ  
30 ず、INHERIT 属性も持たないとき、IDENTITY\_MAP は真に設定される。た  
31 だし、その他の状況でも真となることがある。

32  
33 NCOPIES (省略可能) 基本整数型スカラーでなければならない。INTENT (OUT) 引数とす  
34 る。*align-target* に最終的に整列している ALIGNEE の複製の数が設定され  
35 る。複製がないときは 1 が設定される。

36  
37 ALIGNEE がスカラーのとき、LB、UB、STRIDE、AXIS\_MAP には何も設定されない。

38  
39 例示. 以下の宣言が与えられ、

40  
41 REAL PI = 3.1415927  
42 DIMENSION A(10,10),B(20,30),C(20,40,10),D(40)  
43 !HPF\$ TEMPLATE T(40,20)  
44 !HPF\$ ALIGN A(I,:) WITH T(1+3\*I,2:20:2)  
45 !HPF\$ ALIGN C(I,\*,J) WITH T(J,21-I)  
46 !HPF\$ ALIGN D(I) WITH T(I,4)  
47 !HPF\$ PROCESSORS PROCS(4,2), SCALARPROC  
48

```

!HPF$ DISTRIBUTE T(BLOCK,BLOCK) ONTO PROCS
!HPF$ DISTRIBUTE B(CYCLIC,BLOCK) ONTO PROCS
!HPF$ DISTRIBUTE ONTO SCALARPROC :: PI

```

実際のマッピングが宣言通りに行なわれたとすると、HPF\_ALIGNMENTの呼出しの結果は以下のようになる。

	A	B	C	D
LB	[4, 2]	[1, 1]	[20, N/A, 1]	[1]
UB	[31, 20]	[20, 30]	[ 1, N/A, 10]	[40]
STRIDE	[3, 2]	[1, 1]	[-1, 0, 1]	[1]
AXIS_MAP	[1, 2]	[1, 2]	[2, 0, 1]	[1]
IDENTITY_MAP	偽	真	偽	偽
NCOPIES	1	1	1	1

ここで、「N/A」は実装依存の結果を表わす。NCOPIESの利用法を次の例で見る。

```

LOGICAL BOZO(20,20),RONALD_MCDONALD(20)
!HPF$ TEMPLATE EMMETT_KELLY(100,100)
!HPF$ ALIGN RONALD_MCDONALD(I) WITH BOZO(I,*)
!HPF$ ALIGN BOZO(J,K) WITH EMMETT_KELLY(J,5*K)

```

このとき、CALL HPF\_ALIGNMENT(RONALD\_MCDONALD, NCOPIES = NC) は NC に 20 を設定する。また次の例では、

```

LOGICAL BOZO(20,20),RONALD_MCDONALD(20)
!HPF$ TEMPLATE WILLIE_WHISTLE(100)
!HPF$ ALIGN RONALD_MCDONALD(I) WITH BOZO(I,*)
!HPF$ ALIGN BOZO(J,*) WITH WILLIE_WHISTLE(5*J)

```

CALL HPF\_ALIGNMENT(RONALD\_MCDONALD, NCOPIES = NC) は NC に 1 を設定する。

## HPF\_DISTRIBUTION(DISTRIBUTE, AXIS\_TYPE, AXIS\_INFO, PROCESSORS\_RANK, PROCESSORS\_SHAPE)

省略可能な引数. AXIS\_TYPE, AXIS\_INFO, PROCESSORS\_RANK, PROCESSORS\_SHAPE

機能. 変数の最終 *align-target* の分散に関する情報を返す。

分類. マッピング問合せサブルーチン

引数.

DISTRIBUTE どの型でもよく、スカラでも配列でもよい。大きさ引継ぎ配列であってはならない。結合変数グループのメンバであるとき、そのグループの結合被覆でなければならない(「結合変数グループ」及び「結合被覆」の定義

1 については 3.8 項を参照されたい)<sup>2</sup>。空状態のポインタ又は割り付けられて  
2 いない割付け配列であってはならない。INTENT (IN) 引数とする。

3 DISTRIBUTEE がポインタであるときは、指示先の分散に関する情報が返さ  
4 れるものとする。指示先は大きさ引継ぎの仮引数又は大きさ引継ぎの仮引  
5 数の部分配列であってはならない。  
6

7 AXIS\_TYPE (省略可能) 基本文字型の 1 次元配列でなければならない。長さは任意で  
8 あるが、完全な値を得るためには少なくとも長さが 9 必要である。各要  
9 素には以下の値が文字組込み代入文と同じ方法で設定される。少なくとも  
10 DISTRIBUTEE が最終的に整列している *align-target* の次元数と等しい大き  
11 さを持たなければならない (これは HPF\_TEMPLATE が TEMPLATE\_RANK に返  
12 す値である)。INTENT (OUT) 引数とする。AXIS\_TYPE の *i* 番目の要素には、  
13 *align-target* の *i* 番目の次元軸の分散に関する情報が設定される。HPF で  
14 は以下の値が定義されている (実装によってはその他の値が定義されてい  
15 ることもある)。  
16

17 'BLOCK' この次元軸は BLOCK 分散されている。AXIS\_INFO の対応する要  
18 素にはブロックの大きさが設定される。

19 'COLLAPSED' この次元軸は縮退している ('\*' 指定により分散されてい  
20 る)。AXIS\_INFO の対応する要素の値は実装依存とする。

21 'CYCLIC' この次元軸は CYCLIC 分散されている。AXIS\_INFO の対応する  
22 要素にはブロックの大きさが設定される。  
23

24  
25 AXIS\_INFO (省略可能) 基本整数型の 1 次元配列でなければならない。少なくとも  
26 DISTRIBUTEE が最終的に整列している *align-target* の次元数と等しい大き  
27 さを持たなければならない (これは HPF\_TEMPLATE が TEMPLATE\_RANK に返  
28 す値である)。INTENT (OUT) 引数とする。AXIS\_INFO の *i* 番目の要素には、  
29 DISTRIBUTEE の最終 *align-target* の *i* 番目の次元軸の、ブロック分散又は  
30 サイクリック分散でのブロックの大きさが設定される。次元軸が縮退して  
31 いるとき、設定される値は実装に依存する。  
32

33 PROCESSORS\_RANK (省略可能) 基本整数型スカラーでなければならない。DISTRIBUTEE  
34 が分散されるプロセッサ構成の次元数が設定される。INTENT (OUT) 引数  
35 とする。  
36

37 PROCESSORS\_SHAPE (省略可能) 基本整数型の 1 次元配列でなければならない。少なくと  
38 も PROCESSORS\_RANK の返す値 *m* に等しい大きさを持たなければならない。  
39 INTENT (OUT) 引数とする。最初の *m* 個の要素に、DISTRIBUTEE がマップさ  
40 れるプロセッサ構成の形状が設定される (HPF\_DISTRIBUTION は 2 度呼び出  
41 す必要があるかもしれない。すなわち最初の呼び出しで PROCESSORS\_RANK  
42 の値を得、それにより PROCESSORS\_SHAPE を割り付けるのである)。  
43  
44

45 例示. HPF\_ALIGNMENT の例に示す宣言が与えられ、実際のマッピングが宣言通りに行  
46 なわれたとすると、HPF\_DISTRIBUTE の呼出しの結果は以下ようになる。

47 <sup>2</sup>訳注: 原文の誤り。HPF 2.0 では「結合被覆」の定義は抹消されている。DISTRIBUTEE は結合変数グループ  
48 のメンバであってはならないと解釈すべきである。

	A	B	PI
AXIS_TYPE	['BLOCK', 'BLOCK']	['CYCLIC', 'BLOCK']	[ ]
AXIS_INFO	[10, 10]	[1, 15]	[ ]
PROCESSORS_SHAPE	[4, 2]	[2, 2]	[ ]
PROCESSORS_RANK	2	2	0

## HPF\_TEMPLATE(ALIGNEE, TEMPLATE\_RANK, LB, UB, AXIS\_TYPE, AXIS\_INFO, NUMBER\_ALIGNED)

省略可能な引数. LB, UB, AXIS\_TYPE, AXIS\_INFO, NUMBER\_ALIGNED, TEMPLATE\_RANK

機能. 変数の最終 *align-target* に関する情報を返す。HPF\_ALIGNMENT が変数の観点からの情報を返すのに対し、HPF\_TEMPLATE は最終 *align-target* の観点からの変数に関する情報を返す。

分類. マッピング問合せサブルーチン

引数.

**ALIGNEE** どの型でもよく、スカラでも配列でもよい。大きさ引継ぎ配列であってはならない。結合変数グループのメンバであるとき、そのグループの結合被覆でなければならない(「結合変数グループ」及び「結合被覆」の定義については 3.8 項を参照されたい)<sup>3</sup>。空状態のポインタ又は割り付けられていない割付け配列であってはならない。INTENT (IN) 引数とする。

ALIGNEE がポインタであるときは、指示先の整列に関する情報が返されるものとする。指示先は大きさ引継ぎの仮引数又は大きさ引継ぎの仮引数の部分配列であってはならない。

**TEMPLATE\_RANK** (省略可能) 基本整数型スカラでなければならない。INTENT (OUT) 引数とする。最終 *align-target* の次元数が設定される。この値は、縮退や複製により ALIGNEE の次元数と異なることがある。

**LB** (省略可能) 基本整数型の 1 次元配列でなければならない。少なくとも ALIGNEE が最終的に整列している *align-target* の次元数と等しい大きさを持たなければならない(これは TEMPLATE\_RANK に設定される値である)。INTENT (OUT) 引数とする。LB の *i* 番目の要素には、テンプレートの *i* 番目の次元軸に対する *align-target* の宣言時の下限が設定される。

**UB** (省略可能) 基本整数型の 1 次元配列でなければならない。少なくとも ALIGNEE が最終的に整列している *align-target* の次元数と等しい大きさを持たなければならない(これは TEMPLATE\_RANK に設定される値である)。INTENT (OUT) 引数とする。UB の *i* 番目の要素には、テンプレートの *i* 番目の次元軸に対する *align-target* の宣言時の上限が設定される。

<sup>3</sup>訳注: 原文の誤り。HPF 2.0 では「結合被覆」の定義は抹消されている。ALIGNEE は結合変数グループのメンバであってはならないと解釈すべきである。

1        **AXIS\_TYPE** (省略可能) 基本文字型の 1 次元配列でなければならない。長さは任意で  
 2        あるが、完全な値を得るためには少なくとも長さが 10 必要である。各要素  
 3        には以下の値が文字組み代入文と同じ方法で設定される。少なくとも  
 4        も **ALIGNEE** が最終的に整列している *align-target* の次元数と等しい大きさを  
 5        持たなければならない (これは **TEMPLATE\_RANK** に設定される値である)。  
 6        **INTENT** (OUT) 引数とする。 **AXIS\_TYPE** の *i* 番目の要素には、 *align-target*  
 7        の *i* 番目の次元軸に関する情報が設定される。 **HPF** では以下の値が定義さ  
 8        れている (実装によりその他の値が定義されていることがある)。

10        '**NORMAL**' *align-target* のこの次元軸には、 **ALIGNEE** のある次元軸が整列し  
 11        ている。この値が代入された **AXIS\_TYPE** の要素に対しては、対応する  
 12        **AXIS\_INFO** の要素に、この *align-target* の次元軸に整列する **ALIGNEE**  
 13        の次元軸の番号が設定される。

15        '**REPLICATED**' *align-target* のこの次元軸には、 **ALIGNEE** が複製されている。  
 16        この値が代入された **AXIS\_TYPE** の要素に対しては、対応する **AXIS\_INFO**  
 17        の要素に、この *align-target* の次元軸に複製された **ALIGNEE** の複製の  
 18        数が設定される。

19        '**SINGLE**' *align-target* のこの次元軸のある座標に、 **ALIGNEE** が整列して  
 20        いる。この値が代入された **AXIS\_TYPE** の要素に対しては、対応する  
 21        **AXIS\_INFO** の要素に、 **ALIGNEE** が整列している *align-target* の座標が  
 22        設定される。

24        **AXIS\_INFO** (省略可能) 基本整数型の 1 次元配列でなければならない。少なくとも  
 25        **ALIGNEE** が最終的に整列している *align-target* の次元数と等しい大きさを  
 26        持たなければならない (これは **TEMPLATE\_RANK** に設定される値である)。  
 27        **INTENT** (OUT) 引数とする。上記 **AXIS\_TYPE** の項を参照されたい。

29        **NUMBER\_ALIGNED** (省略可能) 基本整数型スカラでなければならない。 **INTENT** (OUT)  
 30        引数とする。この最終 *align-target* に整列している変数の総数が設定され  
 31        る。これは、この *align-target* が再分散されたときに同時に移動する変数  
 32        の数である。

34        例示. **HPF\_ALIGNMENT** の例に示す宣言が与えられ、実際のマッピングが宣言通りに行  
 35        なわれたとすると、 **HPF\_TEMPLATE** の呼出しの結果は以下ようになる。

	A	C	D
<b>LB</b>	[1, 1]	[1, 1]	[1, 1]
<b>UB</b>	[40, 20]	[40, 20]	[40, 20]
<b>AXIS_TYPE</b>	['NORMAL', 'NORMAL']	['NORMAL', 'NORMAL']	['NORMAL', 'SINGLE']
<b>AXIS_INFO</b>	[1, 2]	[3, 1]	[1, 4]
<b>NUMBER_ALIGNED</b>	3	3	3
<b>TEMPLATE_RANK</b>	2	2	2

# IALL(ARRAY, DIM, MASK)

省略可能な引数. DIM, MASK

機能. DIM 番目の次元について、ARRAY の要素のビット論理積を集計する。

分類. 変型関数

引数.

ARRAY 整数型でなければならず、スカラであってはならない。

DIM (省略可能) 整数型スカラであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならず。  
ここで  $n$  は、ARRAY の次元数とする。対応する実引数は、省略可能な仮引数であってはならない。

MASK (省略可能) 論理型でなければならず、ARRAY と形状適合しなければならない。

結果の型、型パラメタ 及び 形状. 結果は、ARRAY と同じ種別型パラメタの整数型とする。DIM が省略されているか、又は ARRAY が 1 次元配列であるときは、スカラとする。それ以外るとき、結果は  $n-1$  次元配列とし、形状は  $(d_1, d_2, \dots, d_{\text{DIM}-1}, d_{\text{DIM}+1}, \dots, d_n)$  とする。ここで  $(d_1, d_2, \dots, d_n)$  は、ARRAY の形状とする。

結果の値.

場合 (i): IALL(ARRAY) の結果の値は、ARRAY の全要素の IAND 集計とする。ただし、ARRAY の大きさが 0 であるときは、ARRAY と同じ型種別パラメタのすべての整数 I に対して  $\text{IAND}(I, x) = I$  なる性質を持つ実装依存の値  $x$  とする。7.4.3 項を参照のこと。

場合 (ii): IALL(ARRAY, MASK=MASK) の結果の値は、MASK 中の真である要素と対応している ARRAY の要素の IAND 集計とする。ただし、真である要素がないときは、ARRAY と同じ型種別パラメタのすべての整数 I に対して  $\text{IAND}(I, x) = I$  なる性質を持つ実装依存の値  $x$  とする。

場合 (iii): ARRAY が 1 次元配列のとき、IALL(ARRAY, DIM [,MASK]) の結果の値は IALL(ARRAY [,MASK]) とする。2 次元以上するとき、IALL(ARRAY, DIM [,MASK]) の要素  $(s_1, s_2, \dots, s_{\text{DIM}-1}, s_{\text{DIM}+1}, \dots, s_n)$  の値は  $\text{IALL}(\text{ARRAY}(s_1, s_2, \dots, s_{\text{DIM}-1}, :, s_{\text{DIM}+1}, \dots, s_n) [, \text{MASK} = \text{MASK}(s_1, s_2, \dots, s_{\text{DIM}-1}, :, s_{\text{DIM}+1}, \dots, s_n)])$  とする。

例示.

場合 (i): IALL((/ 7, 6, 3, 2 /)) の値は、2 になる。

場合 (ii): IALL(C, MASK = BTEST(C,0)) の値は、C の奇数である要素の IAND 集計になる。

場合 (iii): 配列 B が  $\begin{bmatrix} 2 & 3 & 5 \\ 3 & 7 & 7 \end{bmatrix}$  であるとき、IALL(B, DIM = 1) の値は  $\begin{bmatrix} 2 & 3 & 5 \end{bmatrix}$  になり、IALL(B, DIM = 2) の値は  $\begin{bmatrix} 0 & 3 \end{bmatrix}$  になる。

# IALL\_PREFIX (ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)

省略可能な引数. DIM, MASK, SEGMENT, EXCLUSIVE

機能. ARRAY の DIM 番目の次元について、区分ごとに順方向のビット論理積累計を計算する。

分類. 変型関数

引数.

ARRAY 整数型でなければならず、スカラであってはならない。

DIM (省略可能) 整数型スカラであって、 $1 \leq DIM \leq n$  の範囲の値でなければならず。  
ここで  $n$  は、ARRAY の次元数とする。

MASK (省略可能) 論理型でなければならず、ARRAY と形状適合しなければならない。

SEGMENT (省略可能) 論理型でなければならず、ARRAY と同じ形状でなければならず。

EXCLUSIVE (省略可能) 論理型スカラでなければならず。

結果の型、型パラメタ 及び 形状. ARRAY と同じとする。

結果の値. 結果の要素  $r$  の値は、 $IALL((/ a_1, \dots, a_m /))$  とする。ここで、 $(a_1, \dots, a_m)$  は、7.4.5 項に述べた規則により、 $r$  に寄与すべく選択された ARRAY の要素の集合である (空集合のこともある)。

例示.         $IALL\_PREFIX( (/ 1, 3, 2, 4, 5 /), \text{SEGMENT} = (/ F, F, F, T, T /) )$  の値は  
$$\begin{bmatrix} 1 & 1 & 0 & 4 & 4 \end{bmatrix}$$
 となる。

# IALL\_SCATTER (ARRAY, BASE, INDX1, ..., INDXn, MASK)

省略可能な引数. MASK

機能. MASK で選択される ARRAY の要素を、指標配列  $INDX1, \dots, INDXn$  で指定される結果の位置に拡散する。結果の要素の第  $j$  ビットは、BASE の対応する要素及びその位置に拡散される ARRAY の要素の第  $j$  ビットがすべて 1 であるとき、かつこのときに限り、1 となる。

分類. 変型関数

引数.

ARRAY 整数型でなければならず、スカラであってはならない。

BASE ARRAY と同じ種別型パラメタの整数型でなければならず、スカラであってはならない。

$INDX1, \dots, INDXn$  整数型でなければならず、ARRAY と形状適合しなければならない。  
引数 INDX の数は、BASE の次元数と等しくなければならない。

MASK (省略可能) 論理型でなければならず、ARRAY と形状適合しなければならない。

結果の型、型パラメタ 及び 形状. BASE と同じとする。

結果の値. BASE の要素  $b$  に対応する結果の要素の値は、 $IALL( (/a_1, a_2, \dots, a_m, b/ ) )$  とする。ここで  $(a_1, \dots, a_m)$  は 7.4.4 項に述べた方法で  $b$  と関連付けられる ARRAY の要素とする。

例示.  $IALL\_SCATTER( (/ 1, 2, 3, 6 / ), (/ 1, 3, 7 / ), (/ 1, 1, 2, 2 / ) )$   
は  $\begin{bmatrix} 0 & 2 & 7 \end{bmatrix}$  となる。

## IALL\_SUFFIX(ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)

省略可能な引数. DIM, MASK, SEGMENT, EXCLUSIVE

機能. ARRAY の DIM 番目の次元について、区分ごとに逆方向のビット論理積累計を計算する。

分類. 変型関数

引数.

ARRAY 整数型でなければならず、スカラであってはならない。

DIM (省略可能) 整数型スカラであって、 $1 \leq DIM \leq n$  の範囲の値でなければならない。  
ここで  $n$  は、ARRAY の次元数とする。

MASK (省略可能) 論理型でなければならず、ARRAY と形状適合しなければならない。

SEGMENT (省略可能) 論理型でなければならず、ARRAY と同じ形状でなければならない。

EXCLUSIVE (省略可能) 論理型スカラでなければならない。

結果の型、型パラメタ 及び 形状. ARRAY と同じとする。

結果の値. 結果の要素  $r$  の値は、 $IALL( (/ a_1, \dots, a_m / ) )$  とする。ここで、 $(a_1, \dots, a_m)$  は、7.4.5 項に述べた規則により、 $r$  に寄与すべく選択された ARRAY の要素の集合である (空集合のこともある)。

例示.  $IALL\_SUFFIX( (/1,3,2,4,5/ ), SEGMENT= (/F,F,F,T,T/ ) )$  の値は  
 $\begin{bmatrix} 0 & 2 & 2 & 4 & 5 \end{bmatrix}$  となる。

## IANY(ARRAY, DIM, MASK)

省略可能な引数. DIM, MASK

機能. DIM 番目の次元について、ARRAY の要素のビット論理和を集計する。

分類. 変型関数



1 引数.

2 ARRAY 整数型でなければならず、スカラであってはならない。

3  
4 DIM (省略可能) 整数型スカラであって、 $1 \leq DIM \leq n$  の範囲の値でなければならず。  
5 ここで  $n$  は、ARRAY の次元数とする。対応する実引数は、省略可能な仮引  
6 数であってはならない。

7  
8 MASK (省略可能) 論理型でなければならず、ARRAY と形状適合しなければならず。

9  
10 結果の型、型パラメタ 及び 形状. 結果は、ARRAY と同じ種別型パラメタの整数型とす  
11 る。DIM が省略されているか、又は ARRAY が 1 次元配列であるときは、スカラとする。そ  
12 れ以外のとき、結果は  $n-1$  次元配列とし、形状は  $(d_1, d_2, \dots, d_{DIM-1}, d_{DIM+1}, \dots, d_n)$   
13 とする。ここで  $(d_1, d_2, \dots, d_n)$  は、ARRAY の形状とする。

14  
15 結果の値.

16  
17 場合 (i): IANY(ARRAY) の結果の値は、ARRAY の全要素の IOR 集計とする。ただし、  
18 ARRAY の大きさが 0 であるときは、0 とする。7.4.3 項を参照のこと。

19  
20 場合 (ii): IANY(ARRAY, MASK=MASK) の結果の値は、MASK 中の真である要素と対応  
21 している ARRAY の要素の IOR 集計とする。ただし、真である要素がないと  
22 きは、0 とする。

23  
24 場合 (iii): ARRAY が 1 次元配列のとき、IANY(ARRAY, DIM [,MASK]) の結果の値  
25 は IANY(ARRAY [,MASK]) とする。2 次元以上のとき、IANY(ARRAY,  
26 DIM [,MASK]) の要素  $(s_1, s_2, \dots, s_{DIM-1}, s_{DIM+1}, \dots, s_n)$  の値  
27 は、IANY(ARRAY( $s_1, s_2, \dots, s_{DIM-1}, :, s_{DIM+1}, \dots, s_n$ ) [,MASK =  
28 MASK( $s_1, s_2, \dots, s_{DIM-1}, :, s_{DIM+1}, \dots, s_n$ )]) とする。

29  
30 例示.

31  
32 場合 (i): IANY((/ 9, 8, 3, 2 /)) の値は、11 になる。

33  
34 場合 (ii): IANY(C, MASK = BTEST(C,0)) の値は、C の奇数である要素の IOR 集計に  
35 なる。

36  
37 場合 (iii): 配列 B が  $\begin{bmatrix} 2 & 3 & 5 \\ 0 & 4 & 2 \end{bmatrix}$  であるとき、IANY(B, DIM = 1) の値は  $\begin{bmatrix} 2 & 7 & 7 \end{bmatrix}$   
38  
39 になり、IANY(B, DIM = 2) の値は  $\begin{bmatrix} 7 & 6 \end{bmatrix}$  になる。

40  
41 IANY\_PREFIX(ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)

42 省略可能な引数. DIM, MASK, SEGMENT, EXCLUSIVE

43  
44 機能. ARRAY の DIM 番目の次元について、区分ごとに順方向のビット論理和累計を計  
45 算する。

46  
47 分類. 変型関数  
48

引数. 1

ARRAY 整数型でなければならず、スカラーであってはならない。 2

DIM (省略可能) 整数型スカラーであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならず。 3

ここで  $n$  は、ARRAY の次元数とする。 4

MASK (省略可能) 論理型でなければならず、ARRAY と形状適合しななければならない。 5

SEGMENT (省略可能) 論理型でなければならず、ARRAY と同じ形状でなければならず。 6

EXCLUSIVE (省略可能) 論理型スカラーでなければならず。 7

結果の型、型パラメタ 及び 形状. ARRAY と同じとする。 8

結果の値. 結果の要素  $r$  の値は、 $\text{IANY}((/ a_1, \dots, a_m /))$  とする。ここで、 $(a_1, \dots, a_m)$  9

は、7.4.5 項に述べた規則により、 $r$  に寄与すべく選択された ARRAY の要素の集合であ 10

る (空集合のこともある)。 11

例示.  $\text{IANY\_PREFIX}((/1,2,3,2,5/), \text{SEGMENT}=(/F,F,F,T,T/))$  の値は 12

$\begin{bmatrix} 1 & 3 & 3 & 2 & 7 \end{bmatrix}$  となる。 13

## IANY\_SCATTER(ARRAY,BASE,INDX1, ..., INDXn, MASK)

省略可能な引数. MASK

機能. MASK で選択される ARRAY の要素を、指標配列  $\text{INDX1}, \dots, \text{INDXn}$  で指定される 21

結果の位置に拡散する。結果の要素の第  $j$  ビットは、BASE の対応する要素またはその 22

位置に拡散される ARRAY の要素の第  $j$  ビットの中に一つでも 1 があるとき、かつこの 23

ときに限り、1 となる。 24

分類. 変型関数

引数.

ARRAY 整数型でなければならず、スカラーであってはならない。 25

BASE ARRAY と同じ種別型パラメタの整数型でなければならず、スカラーであって 26

はならない。 27

$\text{INDX1}, \dots, \text{INDXn}$  整数型でなければならず、ARRAY と形状適合しななければならない。 28

引数 INDX の数は、BASE の次元数と等しくなければならない。 29

MASK (省略可能) 論理型でなければならず、ARRAY と形状適合しななければならない。 30

結果の型、型パラメタ 及び 形状. BASE と同じとする。 31

結果の値. BASE の要素  $b$  に対応する結果の要素の値は、 $\text{IANY}((/a_1, a_2, \dots, a_m, b/))$  32

とする。ここで  $(a_1, \dots, a_m)$  は 7.4.4 項に述べた方法で  $b$  と関連付けられる ARRAY の要 33

素とする。 34

例示.  $\text{IANY\_SCATTER}((/ 1, 2, 3, 6 /), (/ 1, 3, 7 /), (/ 1, 1, 2, 2 /))$  35

は  $\begin{bmatrix} 3 & 7 & 7 \end{bmatrix}$  となる。 36

## 1 IANY\_SUFFIX(ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)

2 省略可能な引数. DIM, MASK, SEGMENT, EXCLUSIVE

3  
4 機能. ARRAY の DIM 番目の次元について、区分ごとに逆方向のビット論理和累計を計  
5 算する。

6  
7 分類. 変型関数

8  
9 引数.

10  
11 ARRAY 整数型でなければならず、スカラーであってはならない。

12  
13 DIM (省略可能) 整数型スカラーであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならず。  
14 ここで  $n$  は、ARRAY の次元数とする。

15  
16 MASK (省略可能) 論理型でなければならず、ARRAY と形状適合しなければならず。

17  
18 SEGMENT (省略可能) 論理型でなければならず、ARRAY と同じ形状でなければならず。

19  
20 EXCLUSIVE (省略可能) 論理型スカラーでなければならず。

21  
22 結果の型、型パラメタ 及び 形状. ARRAY と同じとする。

23  
24 結果の値. 結果の要素  $r$  の値は、 $\text{IANY}((/ a_1, \dots, a_m /))$  とする。ここで、 $(a_1, \dots, a_m)$   
25 は、7.4.5 項に述べた規則により、 $r$  に寄与すべく選択された ARRAY の要素の集合であ  
26 る (空集合のこともある)。

27 例示.  $\text{IANY\_SUFFIX}((/4, 2, 3, 2, 5/), \text{SEGMENT}=(/F, F, F, T, T/))$  の値は  
28  $\begin{bmatrix} 7 & 3 & 3 & 7 & 5 \end{bmatrix}$  となる。  
29

## 30 IPARITY(ARRAY, DIM, MASK)

31  
32 省略可能な引数. DIM, MASK

33  
34 機能. DIM 番目の次元について、ARRAY の要素のビット排他的論理和を集計する。

35  
36 分類. 変型関数

37  
38 引数.

39  
40 ARRAY 整数型でなければならず、スカラーであってはならない。

41  
42 DIM (省略可能) 整数型スカラーであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならず。  
43 ここで  $n$  は、ARRAY の次元数とする。対応する実引数は、省略可能な仮引  
44 数であってはならない。

45  
46 MASK (省略可能) 論理型でなければならず、ARRAY と形状適合しなければならず。  
47  
48

結果の型、型パラメタ及び形状。結果は、ARRAYと同じ種別型パラメタの整数型とする。DIMが省略されているか、又はARRAYが1次元配列であるときは、スカラとする。それ以外の場合、結果は $n-1$ 次元配列とし、形状は $(d_1, d_2, \dots, d_{DIM-1}, d_{DIM+1}, \dots, d_n)$ とする。ここで $(d_1, d_2, \dots, d_n)$ は、ARRAYの形状とする。

結果の値。

場合 (i): IPARITY(ARRAY)の結果の値は、ARRAYの全要素のIEOR集計とする。ただし、ARRAYの大きさが0であるときは、0とする。7.4.3項を参照のこと。

場合 (ii): IPARITY(ARRAY, MASK=MASK)の結果の値は、MASK中の真である要素と対応しているARRAYの要素のIEOR集計とする。ただし、真である要素がないときは、0とする。

場合 (iii): ARRAYが1次元配列のとき、IPARITY(ARRAY, DIM [,MASK])の結果の値はIPARITY(ARRAY [,MASK])とする。2次元以上のとき、IPARITY(ARRAY, DIM [,MASK])の要素 $(s_1, s_2, \dots, s_{DIM-1}, s_{DIM+1}, \dots, s_n)$ の値は、 $IPARITY(ARRAY(s_1, s_2, \dots, s_{DIM-1}, :, s_{DIM+1}, \dots, s_n) [,MASK = MASK(s_1, s_2, \dots, s_{DIM-1}, :, s_{DIM+1}, \dots, s_n)])$ とする。

例示。

場合 (i): IPARITY((/ 13, 8, 3, 2 /))の値は、4になる。

場合 (ii): IPARITY(C, MASK = BTEST(C,0))の値は、Cの奇数である要素のIEOR集計になる。

場合 (iii): 配列Bが  $\begin{bmatrix} 2 & 3 & 7 \\ 0 & 4 & 2 \end{bmatrix}$  であるとき、IPARITY(B, DIM = 1)の値は  $\begin{bmatrix} 2 & 7 & 5 \end{bmatrix}$  になり、IPARITY(B, DIM = 2)の値は  $\begin{bmatrix} 6 & 6 \end{bmatrix}$  になる。

## IPARITY\_PREFIX(ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)

省略可能な引数。DIM, MASK, SEGMENT, EXCLUSIVE

機能。ARRAYのDIM番目の次元について、区分ごとに順方向のビット排他的論理和累計を計算する。

分類。変型関数

引数。

ARRAY 整数型でなければならず、スカラであってはならない。

DIM (省略可能) 整数型スカラであって、 $1 \leq DIM \leq n$ の範囲の値でなければならず、ここで $n$ は、ARRAYの次元数とする。

MASK (省略可能) 論理型でなければならず、ARRAYと形状適合しななければならない。

SEGMENT (省略可能) 論理型でなければならず、ARRAYと同じ形状でなければならない。

1 EXCLUSIVE (省略可能) 論理型スカラでなければならない。

2 結果の型、型パラメタ 及び 形状. ARRAYと同じとする。

3  
4  
5 結果の値. 結果の要素  $r$  の値は、 $\text{IPARITY}((/ a_1, \dots, a_m /))$  とする。ここで、  
6  $(a_1, \dots, a_m)$  は、7.4.5 項に述べた規則により、 $r$  に寄与すべく選択された ARRAY の  
7 要素の集合である (空集合のこともある)。

8  
9 例示.  $\text{IPARITY\_PREFIX}((/1,2,3,4,5/), \text{SEGMENT}=(/F,F,F,T,T/))$  の値は  
10  $\begin{bmatrix} 1 & 3 & 0 & 4 & 1 \end{bmatrix}$  となる。  
11

## 12 IPARITY\_SCATTER(ARRAY,BASE,INDX1, ..., INDXn, MASK)

13  
14 省略可能な引数. MASK

15  
16 機能. MASK で選択される ARRAY の要素を、指標配列  $\text{INDX1}, \dots, \text{INDXn}$  で指定される  
17 結果の位置に拡散する。結果の要素の第  $j$  ビットは、BASE の対応する要素とその位置  
18 に拡散される ARRAY の要素の第  $j$  ビットの中に奇数個の 1 があるとき、かつこのとき  
19 に限り、1 となる。  
20

21 分類. 変型関数

22 引数.

23  
24  
25 ARRAY 整数型でなければならず、スカラであってはならない。

26  
27 BASE ARRAY と同じ種別型パラメタの整数型でなければならず、スカラであって  
28 はならない。

29  
30 INDX1, ..., INDXn 整数型でなければならず、ARRAY と形状適合しなければならない。  
31 引数 INDX の数は、BASE の次元数と等しくなければならない。

32  
33 MASK (省略可能) 論理型でなければならず、ARRAY と形状適合しなければならない。

34  
35 結果の型、型パラメタ 及び 形状. BASEと同じとする。

36  
37 結果の値. BASE の要素  $b$  に対応する結果の要素の値は、 $\text{IPARITY}((/a_1, a_2, \dots, a_m, b/))$   
38 とする。ここで  $(a_1, \dots, a_m)$  は 7.4.4 項に述べた方法で  $b$  と関連付けられる ARRAY の  
39 要素とする。

40  
41 例示.  $\text{IPARITY\_SCATTER}((/ 1, 2, 3, 6 /), (/ 1, 3, 7 /), (/ 1, 1, 2, 2$   
42  $/))$  は  $\begin{bmatrix} 2 & 6 & 7 \end{bmatrix}$  となる。  
43  
44  
45  
46  
47  
48

## IPARITY\_SUFFIX(ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)

省略可能な引数. DIM, MASK, SEGMENT, EXCLUSIVE

機能. ARRAY の DIM 番目の次元について、区分ごとに逆方向のビット排他的論理和累計を計算する。

分類. 変型関数

引数.

ARRAY 整数型でなければならず、スカラであってはならない。

DIM (省略可能) 整数型スカラであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならず、ここで  $n$  は、ARRAY の次元数とする。

MASK (省略可能) 論理型でなければならず、ARRAY と形状適合しなければならない。

SEGMENT (省略可能) 論理型でなければならず、ARRAY と同じ形状でなければならない。

EXCLUSIVE (省略可能) 論理型スカラでなければならない。

結果の型、型パラメタ 及び 形状. ARRAY と同じとする。

結果の値. 結果の要素  $r$  の値は、 $\text{IPARITY}((/ a_1, \dots, a_m /))$  とする。ここで、 $(a_1, \dots, a_m)$  は、7.4.5 項に述べた規則により、 $r$  に寄与すべく選択された ARRAY の要素の集合である (空集合のこともある)。

例示.  $\text{IPARITY\_SUFFIX}((/1,2,3,4,5/), \text{SEGMENT}=(/F,F,F,T,T/))$  の値は  
 $\begin{bmatrix} 0 & 1 & 3 & 1 & 5 \end{bmatrix}$  となる。

## LEADZ(I)

機能. 整数の先頭の 0 であるビットの個数を返す。

分類. 要素別処理関数

引数. I は整数型でなければならない。

結果の型 及び 型パラメタ. I と同じとする。

結果の値. 結果は整数 I の先頭の 0 であるビットの個数とする。整数値をビット列として解釈する数体系は、F95:13.5.7 項による。LEADZ(0) の値は、BIT\_SIZE(I) とする。I が 0 でないとき、立っている最も左のビットが第  $k-1$  ビットであれば (ただし右端を第 0 ビットとする)、LEADZ(I) の結果は  $\text{BIT\_SIZE}(I) - k$  とする。

例示. LEADZ(3) の値は  $\text{BIT\_SIZE}(3) - 2$  になる。スカラ I に対し、LEADZ(I) .EQ. MINVAL((/ (J, J=0, BIT\_SIZE(I)) /), MASK=M) になる。ここで、 $M=(/ (\text{BTEST}(I,J), J=\text{BIT\_SIZE}(I)-1, 0, -1), \text{.TRUE.} /))$  とする。ある整数 I について、LEADZ(I) の結果は、整数の表現形式のビット数 (BIT\_SIZE(I)) により異なることがある。これは、LEADZ がビット数を最上位ビットから数えるためである。ILEN と比較されたい。

## MAXVAL\_PREFIX (ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)

省略可能な引数. DIM, MASK, SEGMENT, EXCLUSIVE

機能. ARRAY の DIM 番目の次元について、区分ごとに順方向の MAXVAL 累計を計算する。

分類. 変型関数

引数.

ARRAY 整数型又は実数型でなければならず、スカラーであってはならない。

DIM (省略可能) 整数型スカラーであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならず、ここで  $n$  は、ARRAY の次元数とする。

MASK (省略可能) 論理型でなければならず、ARRAY と形状適合しなければならない。

SEGMENT (省略可能) 論理型でなければならず、ARRAY と同じ形状でなければならず。

EXCLUSIVE (省略可能) 論理型スカラーでなければならず。

結果の型、型パラメタ 及び 形状. ARRAY と同じとする。

結果の値. 結果の要素  $r$  の値は、 $\text{MAXVAL}((/ a_1, \dots, a_m /))$  とする。ここで、 $(a_1, \dots, a_m)$  は、7.4.5 項に述べた規則により、 $r$  に寄与すべく選択された ARRAY の要素の集合である (空集合のこともある)。

例示.  $\text{MAXVAL\_PREFIX} ( (/3, 4, -5, 2, 5/), \text{SEGMENT} = (/F, F, F, T, T/) )$  の値は  
 $\begin{bmatrix} 3 & 4 & 4 & 2 & 5 \end{bmatrix}$  となる。

## MAXVAL\_SCATTER (ARRAY, BASE, INDX1, ..., INDXn, MASK)

省略可能な引数. MASK

機能. MASK で選択される ARRAY の要素を、指標配列  $\text{INDX1}, \dots, \text{INDXn}$  で指定される結果の位置に拡散する。結果の各要素は、BASE の対応する要素とその位置に拡散される ARRAY の要素の最大値となる。

分類. 変型関数

引数.

ARRAY 整数型又は実数型でなければならず、スカラーであってはならない。

BASE ARRAY と同じ型及び同じ種別型パラメタでなければならず、スカラーであってはならない。

$\text{INDX1}, \dots, \text{INDXn}$  整数型でなければならず、ARRAY と形状適合しなければならない。  
引数 INDX の数は、BASE の次元数と等しくなければならない。

MASK (省略可能) 論理型でなければならず、ARRAY と形状適合しなければならない。

結果の型、型パラメタ 及び 形状. BASEと同じとする。

結果の値. BASEの要素  $b$  に対応する結果の要素の値は、 $\text{MAXVAL}((/a_1, a_2, \dots, a_m, b/))$  とする。ここで  $(a_1, \dots, a_m)$  は 7.4.4 項に述べた方法で  $b$  と関連付けられる ARRAY の要素とする。

例示.  $\text{MAXVAL\_SCATTER}((/ 1, 2, 3, 1 /), (/ 4, -5, 7 /), (/ 1, 1, 2, 2 /))$  は  $\begin{bmatrix} 4 & 3 & 7 \end{bmatrix}$  となる。

## MAXVAL\_SUFFIX(ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)

省略可能な引数. DIM, MASK, SEGMENT, EXCLUSIVE

機能. ARRAYの DIM 番目の次元について、区分ごとに逆方向の MAXVAL 累計を計算する。

分類. 変型関数

引数.

ARRAY 整数型又は実数型でなければならず、スカラであってはならない。

DIM (省略可能) 整数型スカラであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならず。ここで  $n$  は、ARRAY の次元数とする。

MASK (省略可能) 論理型でなければならず、ARRAY と形状適合しなければならない。

SEGMENT (省略可能) 論理型でなければならず、ARRAY と同じ形状でなければならない。

EXCLUSIVE (省略可能) 論理型スカラでなければならない。

結果の型、型パラメタ 及び 形状. ARRAYと同じとする。

結果の値. 結果の要素  $r$  の値は、 $\text{MAXVAL}((/ a_1, \dots, a_m /))$  とする。ここで、 $(a_1, \dots, a_m)$  は、7.4.5 項に述べた規則により、 $r$  に寄与すべく選択された ARRAY の要素の集合である (空集合のこともある)。

例示.  $\text{MAXVAL\_SUFFIX}((/3, 4, -5, 2, 5/), \text{SEGMENT}=(/F, F, F, T, T/))$  の値は  $\begin{bmatrix} 4 & 4 & -5 & 5 & 5 \end{bmatrix}$  となる。

## MINVAL\_PREFIX(ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)

省略可能な引数. DIM, MASK, SEGMENT, EXCLUSIVE

機能. ARRAYの DIM 番目の次元について、区分ごとに順方向の MINVAL 累計を計算する。

分類. 変型関数

引数.



1 ARRAY 整数型又は実数型でなければならず、スカラーであってはならない。  
2 DIM (省略可能) 整数型スカラーであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならず。  
3 ここで  $n$  は、ARRAY の次元数とする。  
4  
5 MASK (省略可能) 論理型でなければならず、ARRAY と形状適合しなければならず。  
6 SEGMENT (省略可能) 論理型でなければならず、ARRAY と同じ形状でなければならず。  
7  
8 EXCLUSIVE (省略可能) 論理型スカラーでなければならず。  
9

10 結果の型、型パラメタ 及び 形状. ARRAY と同じとする。

11  
12 結果の値. 結果の要素  $r$  の値は、 $\text{MINVAL}((/ a_1, \dots, a_m /))$  とする。ここで、  
13  $(a_1, \dots, a_m)$  は、7.4.5 項に述べた規則により、 $r$  に寄与すべく選択された ARRAY の  
14 要素の集合である (空集合のこともある)。  
15

16 例示.  $\text{MINVAL\_PREFIX}((/1, 2, -3, 4, 5/), \text{SEGMENT}=(/F, F, F, T, T/))$  の値は  
17  $\begin{bmatrix} 1 & 1 & -3 & 4 & 4 \end{bmatrix}$  となる。  
18

## 19 MINVAL\_SCATTER(ARRAY, BASE, INDX1, ..., INDXn, MASK)

20  
21 省略可能な引数. MASK

22  
23 機能. MASK で選択される ARRAY の要素を、指標配列  $\text{INDX1}, \dots, \text{INDXn}$  で指定される  
24 結果の位置に拡散する。結果の各要素は、BASE の対応する要素とその位置に拡散され  
25 る ARRAY の要素の最小値となる。  
26

27 分類. 変型関数

28  
29 引数.

30  
31 ARRAY 整数型又は実数型でなければならず、スカラーであってはならない。  
32  
33 BASE ARRAY と同じ型及び同じ種別型パラメタでなければならず、スカラーであ  
34 てはならない。  
35  
36  $\text{INDX1}, \dots, \text{INDXn}$  整数型でなければならず、ARRAY と形状適合しなければならず。  
37 引数 INDX の数は、BASE の次元数と等しくなければならない。  
38  
39 MASK (省略可能) 論理型でなければならず、ARRAY と形状適合しなければならず。  
40

41 結果の型、型パラメタ 及び 形状. BASE と同じとする。

42 結果の値. BASE の要素  $b$  に対応する結果の要素の値は、 $\text{MINVAL}((/a_1, a_2, \dots, a_m, b/))$   
43 とする。ここで  $(a_1, \dots, a_m)$  は 7.4.4 項に述べた方法で  $b$  と関連付けられる ARRAY の  
44 要素とする。  
45

46 例示.  $\text{MINVAL\_SCATTER}((/ 1, -2, -3, 6 /), (/ 4, 3, 7 /), (/ 1, 1, 2, 2$   
47  $/))$  は  $\begin{bmatrix} -2 & -3 & 7 \end{bmatrix}$  となる。  
48

## MINVAL\_SUFFIX (ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)

省略可能な引数. DIM, MASK, SEGMENT, EXCLUSIVE

機能. ARRAY の DIM 番目の次元について、区分ごとに逆方向の MINVAL 累計を計算する。

分類. 変型関数

引数.

ARRAY 整数型又は実数型でなければならず、スカラーであってはならない。

DIM (省略可能) 整数型スカラーであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならず、ここで  $n$  は、ARRAY の次元数とする。

MASK (省略可能) 論理型でなければならず、ARRAY と形状適合しなければならない。

SEGMENT (省略可能) 論理型でなければならず、ARRAY と同じ形状でなければならない。

EXCLUSIVE (省略可能) 論理型スカラーでなければならない。

結果の型、型パラメタ 及び 形状. ARRAY と同じとする。

結果の値. 結果の要素  $r$  の値は、 $\text{MINVAL}((/ a_1, \dots, a_m /))$  とする。ここで、 $(a_1, \dots, a_m)$  は、7.4.5 項に述べた規則により、 $r$  に寄与すべく選択された ARRAY の要素の集合である (空集合のこともある)。

例示.  $\text{MINVAL\_SUFFIX} ( (/1, 2, -3, 4, 5/), \text{SEGMENT} = (/F, F, F, T, T/) )$  の値は  
 $\begin{bmatrix} -3 & -3 & -3 & 4 & 5 \end{bmatrix}$  となる。

## PARITY (MASK, DIM)

省略可能な引数. DIM

機能. MASK の第 DIM 次元の要素の値に奇数個の真があるかどうか判定する。

分類. 変型関数

引数.

MASK 論理型でなければならず、スカラーであってはならない。

DIM (省略可能) 整数型スカラーであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならず、ここで  $n$  は、MASK の次元数とする。対応する実引数は、省略可能な仮引数であってはならない。

結果の型、型パラメタ 及び 形状. 結果は、MASK と同じ種別型パラメタの論理型とする。DIM が省略されているか、又は MASK が 1 次元配列であるときは、スカラーとする。それ以外するとき、結果は  $n - 1$  次元配列とし、形状は  $(d_1, d_2, \dots, d_{\text{DIM}-1}, d_{\text{DIM}+1}, \dots, d_n)$  とする。ここで  $(d_1, d_2, \dots, d_n)$  は、MASK の形状とする。

1 結果の値.

2 場合 (i): PARITY(MASK) の結果の値は、MASK の全要素の .NEQV. 集計とする。ただ  
3 し、MASK の大きさが 0 であるときは、偽とする。7.4.3 項を参照のこと。

4  
5 場合 (ii): MASK が 1 次元配列のとき、PARITY(MASK, DIM) の結果の値  
6 は PARITY(MASK) とする。2 次元以上のとき、PARITY(MASK,  
7 DIM) の要素  $(s_1, s_2, \dots, s_{DIM-1}, s_{DIM+1}, \dots, s_n)$  の値は、  
8 PARITY(MASK( $s_1, s_2, \dots, s_{DIM-1}, \cdot, s_{DIM+1}, \dots, s_n$ ))) とする。  
9

10 例示.

11 場合 (i): PARITY((/ T, T, T, F /)) の値は、真になる。

12  
13 場合 (ii): 配列 B が  $\begin{bmatrix} T & T & F \\ T & T & T \end{bmatrix}$  であるとき、PARITY(B, DIM = 1) の値は  
14  $\begin{bmatrix} F & F & T \end{bmatrix}$  になり、PARITY(B, DIM = 2) の値は  $\begin{bmatrix} F & T \end{bmatrix}$  になる。  
15  
16  
17

## 18 PARITY\_PREFIX(MASK, DIM, SEGMENT, EXCLUSIVE)

19 省略可能な引数. DIM, SEGMENT, EXCLUSIVE

20 機能. MASK の DIM 番目の次元について、区分ごとに順方向の排他的論理和累計を計算  
21 する。  
22

23 分類. 変型関数

24 引数.

25 MASK 論理型でなければならず、スカラーであってはならない。

26 DIM (省略可能) 整数型スカラーであって、 $1 \leq DIM \leq n$  の範囲の値でなければならず。  
27 ここで  $n$  は、MASK の次元数とする。

28 SEGMENT (省略可能) 論理型でなければならず、MASK と同じ形状でなければならず。

29 EXCLUSIVE (省略可能) 論理型スカラーでなければならず。

30 結果の型、型パラメタ 及び 形状. MASK と同じとする。

31 結果の値. 結果の要素  $r$  の値は、PARITY((/  $a_1, \dots, a_m$  /)) とする。ここで、  
32  $(a_1, \dots, a_m)$  は、7.4.5 項に述べた規則により、 $r$  に寄与すべく選択された MASK の要  
33 素の集合である (空集合のこともある)。  
34

35 例示. PARITY\_PREFIX( (/T,F,T,T,T/), SEGMENT= (/F,F,F,T,T/) ) の値は  
36  $\begin{bmatrix} T & T & F & T & F \end{bmatrix}$  となる。  
37  
38

## PARITY\_SCATTER(MASK, BASE, INDX1, ..., INDXn)

機能. MASK の要素を、指標配列 INDX1, ..., INDXn で指定される結果の位置に拡散する。結果の要素は、BASE の対応する要素とその位置に拡散される MASK の要素の中に奇数個の真があるとき、かつそのときに限り、真となる。

分類. 変型関数

引数.

MASK 論理型でなければならず、スカラーであってはならない。

BASE MASK と同じ種別型パラメタの論理型でなければならず、スカラーであってはならない。

INDX1, ..., INDXn 整数型でなければならず、MASK と形状適合しなければならない。  
引数 INDX の数は、BASE の次元数と等しくなければならない。

結果の型、型パラメタ 及び 形状. BASE と同じとする。

結果の値. BASE の要素  $b$  に対応する結果の要素の値は、 $\text{PARITY}((/a_1, a_2, \dots, a_m, b/))$  とする。ここで  $(a_1, \dots, a_m)$  は 7.4.4 項に述べた方法で  $b$  と関連付けられる MASK の要素とする。

例示.  $\text{PARITY\_SCATTER}((/ T, T, T, T /), (/ T, F, F /), (/ 1, 1, 1, 2 /))$   
は  $\begin{bmatrix} F & T & F \end{bmatrix}$  となる。

## PARITY\_SUFFIX(MASK, DIM, SEGMENT, EXCLUSIVE)

省略可能な引数. DIM, SEGMENT, EXCLUSIVE

機能. MASK の DIM 番目の次元について、区分ごとに逆方向の排他的論理和累計を計算する。

分類. 変型関数

引数.

MASK 論理型でなければならず、スカラーであってはならない。

DIM (省略可能) 整数型スカラーであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならない。  
ここで  $n$  は、MASK の次元数とする。

SEGMENT (省略可能) 論理型でなければならず、MASK と同じ形状でなければならない。

EXCLUSIVE (省略可能) 論理型スカラーでなければならない。

結果の型、型パラメタ 及び 形状. MASK と同じとする。

1 結果の値. 結果の要素  $r$  の値は、 $\text{PARITY}((/ a_1, \dots, a_m /))$  とする。ここで、  
2  $(a_1, \dots, a_m)$  は、7.4.5 項に述べた規則により、 $r$  に寄与すべく選択された MASK の要  
3 素の集合である (空集合のこともある)。

4  
5 例示.  $\text{PARITY\_SUFFIX}(/T, F, T, T, T/)$ ,  $\text{SEGMENT}(/F, F, F, T, T/)$  の値は  
6  $\begin{bmatrix} F & T & T & F & T \end{bmatrix}$  となる。  
7

## 8 POPCNT(I)

9  
10 機能. 整数中の 1 であるビットの個数を返す。

11  
12 分類. 要素別処理関数

13  
14 引数.  $I$  は整数型でなければならない。

15  
16 結果の型 及び 型パラメタ.  $I$  と同じとする。

17  
18 結果の値.  $\text{POPCNT}(I)$  は、整数  $I$  の 2 進表現中の 1 であるビットの個数とする。整数  
19 値をビット列として解釈する数体系は、F95:13.5.7 項による。  
20

21 例示. スカラ  $I$  に対し、 $\text{POPCNT}(I) = \text{COUNT}((/ (\text{BTEST}(I, J)$ ,  $J=0$ ,  
22  $\text{BIT\_SIZE}(I)-1) /))$  になる。  
23

## 24 POPPAR(I)

25  
26 機能. 整数中のビットの奇偶を返す。

27  
28 分類. 要素別処理関数

29  
30 引数.  $I$  は整数型でなければならない。

31  
32 結果の型 及び 型パラメタ.  $I$  と同じとする。

33  
34 結果の値.  $\text{POPPAR}(I)$  は、整数  $I$  中に 1 であるビットが奇数個あるとき 1 とし、偶数個  
35 あるとき 0 とする。整数値をビット列として解釈する数体系は、F95:13.5.7 項による。  
36

37  
38 例示. スカラ  $I$  に対し、 $\text{POPPAR}(I) = \text{MERGE}(1, 0, \text{BTEST}(\text{POPCNT}(I), 0))$  になる。  
39

## 40 PRODUCT\_PREFIX(ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)

41  
42 省略可能な引数. DIM, MASK, SEGMENT, EXCLUSIVE

43  
44 機能. ARRAY の DIM 番目の次元について、区分ごとに順方向の積の累計を計算する。

45  
46 分類. 変型関数

47  
48 引数.

ARRAY 整数型、実数型又は複素数型でなければならず、スカラであってはならない。

DIM (省略可能) 整数型スカラであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならず、ここで  $n$  は、ARRAY の次元数とする。

MASK (省略可能) 論理型でなければならず、ARRAY と形状適合しなければならない。

SEGMENT (省略可能) 論理型でなければならず、ARRAY と同じ形状でなければならず。

EXCLUSIVE (省略可能) 論理型スカラでなければならず。

結果の型、型パラメタ 及び 形状. ARRAY と同じとする。

結果の値. 結果の要素  $r$  の値は、 $\text{PRODUCT}((/ a_1, \dots, a_m /))$  とする。ここで、 $(a_1, \dots, a_m)$  は、7.4.5 項に述べた規則により、 $r$  に寄与すべく選択された ARRAY の要素の集合である (空集合のこともある)。

例示.  $\text{PRODUCT\_PREFIX}((/ 1, 2, 3, 4, 5 /), \text{SEGMENT} = (/ F, F, F, T, T /))$  の値は  
 $\begin{bmatrix} 1 & 2 & 6 & 4 & 20 \end{bmatrix}$  となる。

## PRODUCT\_SCATTER(ARRAY, BASE, INDX1, ..., INDXn, MASK)

省略可能な引数. MASK

機能. MASK で選択される ARRAY の要素を、指標配列 INDX1, ..., INDXn で指定される結果の位置に拡散する。結果の各要素は、BASE の対応する要素とその位置に拡散される ARRAY の要素の積となる。

分類. 変型関数

引数.

ARRAY 整数型、実数型又は複素数型でなければならず、スカラであってはならない。

BASE ARRAY と同じ型及び同じ種別型パラメタでなければならず、スカラであってはならない。

INDX1, ..., INDXn 整数型でなければならず、ARRAY と形状適合しなければならない。引数 INDX の数は、BASE の次元数と等しくなければならない。

MASK (省略可能) 論理型でなければならず、ARRAY と形状適合しなければならない。

結果の型、型パラメタ 及び 形状. BASE と同じとする。

結果の値. BASE の要素  $b$  に対応する結果の要素の値は、 $\text{PRODUCT}((/ a_1, a_2, \dots, a_m, b /))$  とする。ここで  $(a_1, \dots, a_m)$  は 7.4.4 項に述べた方法で  $b$  と関連付けられる ARRAY の要素とする。

例示.  $\text{PRODUCT\_SCATTER}((/ 1, 2, 3, 1 /), (/ 4, -5, 7 /), (/ 1, 1, 2, 2 /))$  は  $\begin{bmatrix} 8 & -15 & 7 \end{bmatrix}$  となる。

## PRODUCT\_SUFFIX(ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)

省略可能な引数. DIM, MASK, SEGMENT, EXCLUSIVE

機能. ARRAY の DIM 番目の次元について、区分ごとに逆方向の積の累計を計算する。

分類. 変型関数

引数.

ARRAY 整数型、実数型又は複素数型でなければならず、スカラーであってはならない。

DIM (省略可能) 整数型スカラーであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならず、ここで  $n$  は、ARRAY の次元数とする。

MASK (省略可能) 論理型でなければならず、ARRAY と形状適合しなければならない。

SEGMENT (省略可能) 論理型でなければならず、ARRAY と同じ形状でなければならず。

EXCLUSIVE (省略可能) 論理型スカラーでなければならず。

結果の型、型パラメタ 及び 形状. ARRAY と同じとする。

結果の値. 結果の要素  $r$  の値は、 $\text{PRODUCT}((/ a_1, \dots, a_m /))$  とする。ここで、 $(a_1, \dots, a_m)$  は、7.4.5 項に述べた規則により、 $r$  に寄与すべく選択された ARRAY の要素の集合である (空集合のこともある)。

例示.  $\text{PRODUCT\_SUFFIX}((/1,2,3,4,5/), \text{SEGMENT}=(/F,F,F,T,T/))$  の値は  
 $\begin{bmatrix} 6 & 6 & 3 & 20 & 5 \end{bmatrix}$  となる。

## SORT\_DOWN(ARRAY, DIM)

省略可能な引数. DIM

機能. 降順にソートする。

分類. 変型関数

引数.

ARRAY 整数型、実数型又は文字型でなければならず、スカラーであってはならない。

DIM (省略可能) 整数型スカラーであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならず、ここで  $n$  は、ARRAY の次元数とする。対応する実引数は、省略可能な仮引数であってはならない。

結果の型、型パラメタ 及び 形状. ARRAY と同じとする。

結果の値.

場合 (i): ARRAY が 1 次元配列のとき、SORT\_DOWN(ARRAY) の結果は、ARRAY と同じ形状のベクトルで、ARRAY と同じ要素を (同じ出現数だけ) 含み、要素が降順にソートされたものとする。文字型配列に対する大小順序は、Fortran の組込み関数で使われるもの、すなわち ASCII 大小順序とする。

場合 (ii): ARRAY が多次元配列のとき、SORT\_DOWN(ARRAY) の結果は、ARRAY を配列要素順序で 1 次元配列 V に変形し、場合 1 の方法でこの 1 次元配列を降順にソートし、最後にそれを元の形状に戻して得られる結果とする。すなわち、RESHAPE( SORT\_DOWN(V), SHAPE = SHAPE(ARRAY) ) と同じ結果とする。ここで、V = RESHAPE( ARRAY, SHAPE = (/ M /) ) とし、M = SIZE(ARRAY) とする。

場合 (iii): SORT\_DOWN(ARRAY, DIM=k) の結果は、ARRAY と同じ要素を含むが、各々の 1 次元部分配列 ARRAY( $i_1, i_2, \dots, i_{k-1}, :, i_{k+1}, \dots, i_n$ ) が Case(i) の方法で要素の降順にソートされたものとする。ここで、n は ARRAY の次元数とする。

例示.

場合 (i): SORT\_DOWN( (/30, 20, 30, 40, -10/) ) の値は  
 $\begin{bmatrix} 40 & 30 & 30 & 20 & -10 \end{bmatrix}$  になる。

場合 (ii): 配列 A が  $\begin{bmatrix} 1 & 9 & 2 \\ 4 & 5 & 2 \\ 1 & 2 & 4 \end{bmatrix}$  であるとき、SORT\_DOWN(A) の値は  $\begin{bmatrix} 9 & 4 & 2 \\ 5 & 2 & 1 \\ 4 & 2 & 1 \end{bmatrix}$  になる。

場合 (iii): 配列 A が  $\begin{bmatrix} 1 & 9 & 2 \\ 4 & 5 & 2 \\ 1 & 2 & 4 \end{bmatrix}$  であるとき、SORT\_DOWN(A, DIM = 1) の値は  $\begin{bmatrix} 4 & 9 & 4 \\ 1 & 5 & 2 \\ 1 & 2 & 2 \end{bmatrix}$  になる。

## SORT\_UP(ARRAY,DIM)

省略可能な引数. DIM

機能. 昇順にソートする。

分類. 変型関数

引数.

ARRAY 整数型、実数型又は文字型でなければならず、スカラであってはならない。

DIM (省略可能) 整数型スカラであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならず。ここで n は、ARRAY の次元数とする。対応する実引数は、省略可能な仮引数であってはならない。



1 結果の型、型パラメタ 及び 形状. ARRAYと同じとする。

2  
3 結果の値.

4 場合 (i): ARRAY が 1 次元配列のとき、SORT\_UP(ARRAY) の結果は、ARRAY と同じ形  
5 状のベクトルで、ARRAY と同じ要素を (同じ出現数だけ) 含み、要素が昇順  
6 にソートされたものとする。文字型配列に対する大小順序は、Fortran の  
7 組込み関数で使われるもの、すなわち ASCII 大小順序とする。

8  
9 場合 (ii): ARRAY が多次元配列のとき、SORT\_UP(ARRAY) の結果は、ARRAY を配列  
10 要素順序で 1 次元配列 V に変形し、場合 1 の方法でこの 1 次元配列を  
11 昇順にソートし、最後にそれを元の形状に戻して得られる結果とする。  
12 すなわち、RESHAPE( SORT\_UP(V), SHAPE = SHAPE(ARRAY) ) と同じ結  
13 果とする。ここで、V = RESHAPE( ARRAY, SHAPE = (/ M /) ) とし、M =  
14 SIZE(ARRAY) とする。

15  
16 場合 (iii): SORT\_UP(ARRAY, DIM=k) の結果は、ARRAY と同じ要素を含むが、各々の 1  
17 次元部分配列 ARRAY( $i_1, i_2, \dots, i_{k-1}, :, i_{k+1}, \dots, i_n$ ) が Case(i) の方法で要素  
18 の昇順にソートされたものとする。ここで、 $n$  は ARRAY の次元数とする。

19  
20  
21 例示.

22  
23 場合 (i): SORT\_UP( (/30, 20, 30, 40, -10/) ) の値は  $\begin{bmatrix} -10 & 20 & 30 & 30 & 40 \end{bmatrix}$   
24 になる。

25  
26 場合 (ii): 配列 A が  $\begin{bmatrix} 1 & 9 & 2 \\ 4 & 5 & 2 \\ 1 & 2 & 4 \end{bmatrix}$  であるとき、SORT\_UP(A) の値は  $\begin{bmatrix} 1 & 2 & 4 \\ 1 & 2 & 5 \\ 2 & 4 & 9 \end{bmatrix}$   
27  
28 になる。

29  
30 場合 (iii): 配列 A が  $\begin{bmatrix} 1 & 9 & 2 \\ 4 & 5 & 2 \\ 1 & 2 & 4 \end{bmatrix}$   
31  
32 であるとき、SORT\_UP(A, DIM = 1) の値は  $\begin{bmatrix} 1 & 2 & 2 \\ 1 & 5 & 2 \\ 4 & 9 & 4 \end{bmatrix}$  になる。  
33  
34  
35  
36  
37

## 38 SUM\_PREFIX(ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)

39 省略可能な引数. DIM, MASK, SEGMENT, EXCLUSIVE

40 機能. ARRAY の DIM 番目の次元について、区分ごとに順方向の SUM 累計を計算する。

41  
42  
43 分類. 変型関数

44  
45  
46 引数.

47  
48 ARRAY 整数型、実数型又は複素数型でなければならず、スカラであってはならない。

DIM (省略可能) 整数型スカラであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならない。  
ここで  $n$  は、ARRAY の次元数とする。

MASK (省略可能) 論理型でなければならない、ARRAY と形状適合しなければならない。

SEGMENT (省略可能) 論理型でなければならない、ARRAY と同じ形状でなければならない。

EXCLUSIVE (省略可能) 論理型スカラでなければならない。

結果の型、型パラメタ 及び 形状. ARRAY と同じとする。

結果の値. 結果の要素  $r$  の値は、 $\text{SUM}((/ a_1, \dots, a_m /))$  とする。ここで、 $(a_1, \dots, a_m)$  は、7.4.5 項に述べた規則により、 $r$  に寄与すべく選択された ARRAY の要素の集合である (空集合のこともある)。

例示.  $\text{SUM\_PREFIX}((/1, 2, 3, 4, 5/), \text{SEGMENT}=(/F, F, F, T, T/))$  の値は  
 $\begin{bmatrix} 1 & 3 & 6 & 4 & 9 \end{bmatrix}$ . となる。

## SUM\_SCATTER(ARRAY, BASE, INDX1, ..., INDXn, MASK)

省略可能な引数. MASK

機能. MASK で選択される ARRAY の要素を、指標配列 INDX1, ..., INDXn で指定される結果の位置に拡散する。結果の各要素は、BASE の対応する要素とその位置に拡散される ARRAY の要素の和となる。

分類. 変型関数

引数.

ARRAY 整数型、実数型又は複素数型でなければならない、スカラであってはならない。

BASE ARRAY と同じ型及び同じ種別型パラメタでなければならない、スカラであってはならない。

INDX1, ..., INDXn 整数型でなければならない、ARRAY と形状適合しなければならない。  
引数 INDX の数は、BASE の次元数と等しくなければならない。

MASK (省略可能) 論理型でなければならない、ARRAY と形状適合しなければならない。

結果の型、型パラメタ 及び 形状. BASE と同じとする。

結果の値. BASE の要素  $b$  に対応する結果の要素の値は、 $\text{SUM}((/a_1, a_2, \dots, a_m, b/))$  とする。ここで  $(a_1, \dots, a_m)$  は 7.4.4 項に述べた方法で  $b$  と関連付けられる ARRAY の要素とする。

例示.  $\text{SUM\_SCATTER}((/ 1, 2, 3, 1 /), (/ 4, -5, 7 /), (/ 1, 1, 2, 2 /))$   
は  $\begin{bmatrix} 7 & -1 & 7 \end{bmatrix}$  となる。

# SUM\_SUFFIX(ARRAY, DIM, MASK, SEGMENT, EXCLUSIVE)

省略可能な引数. DIM, MASK, SEGMENT, EXCLUSIVE

機能. ARRAY の DIM 番目の次元について、区分ごとに逆方向の SUM 累計を計算する。

分類. 変型関数

引数.

ARRAY 整数型、実数型又は複素数型でなければならず、スカラーであってはならない。

DIM (省略可能) 整数型スカラーであって、 $1 \leq \text{DIM} \leq n$  の範囲の値でなければならず、ここで  $n$  は、ARRAY の次元数とする。

MASK (省略可能) 論理型でなければならず、ARRAY と形状適合しなければならない。

SEGMENT (省略可能) 論理型でなければならず、ARRAY と同じ形状でなければならない。

EXCLUSIVE (省略可能) 論理型スカラーでなければならない。

結果の型、型パラメタ 及び 形状. ARRAY と同じとする。

結果の値. 結果の要素  $r$  の値は、 $\text{SUM}((/ a_1, \dots, a_m /))$  とする。ここで、 $(a_1, \dots, a_m)$  は、7.4.5 項に述べた規則により、 $r$  に寄与すべく選択された ARRAY の要素の集合である (空集合のこともある)。

例示.  $\text{SUM\_SUFFIX} ( (/1,2,3,4,5/), \text{SEGMENT}= (/F,F,F,T,T/))$  の値は  
 $\begin{bmatrix} 6 & 5 & 3 & 9 & 5 \end{bmatrix}$  となる。

