

制御用計算機標準入出力ソフトウェアシステム

Standard Software System of I/O Processor for Computer Control System

最近、計算制御システムの分野では、システム規模が年々大となり、特にソフトウェアの分野では、人件費の上昇及び有能人材の不足が切実な問題となっている。これらの解決手段としては、ソフトウェアの標準化が最も効果的である。

また、ソフトウェアの中でも、多様化した入出力機器をアクセスするプログラムは、システム全体に占めるウエートが大きく、特に標準化のニーズは強い。

本稿では、ユーザー ニーズをもとに開発したHIDIC 80用標準入出力システムについて、機能とその機構及び動作を中心に概要を説明し、更に入出力機器をアクセスする標準入出力システムの今後の展望についても触れる。

政井 賢二* Masai Kenji
林 利弘* Hayashi Toshihiro
桜川正三郎** Sakuragawa Syôzaburô
彦坂 利和** Hikosaka Toshikazu

1 緒 言

ソフトウェアの標準化は、プログラムの信頼性及び生産性を向上させることがねらいである。標準化といっても、小さなパッケージ単位での標準化には限度があり、なかなか大きな効果は期待できないので、今後は、あるまとまった機能を持ち、複数パッケージから成るプログラム群（サブシステムとも呼ぶ）としての標準化、更に一步進んで問題向き言語化といった形で進むべきと考えられる。

このような背景をもとに、複雑な入出力処理をユーザーから解放することを目的として、日立製作所ではHIDIC 80用標準入出力システムInput-Output Processor with Formatting function System（以下、IOPFと略す）を開発した。機能面では、COBOL、PL/Iなどの帳票作成と同等レベルの機能、及び入出力機器故障時の処理機能などを具備し、複数パッケージから成るプログラム群として標準化し、更に徹底したモジュール化を行なった。

本稿では、IOPFの概要説明を中心に、入出力機器をアクセスする標準入出力システムの今後の展望についても記述する。

2 標準入出力システムに対するニーズ

計算制御における標準入出力システムについてのユーザーニーズを調査した結果をまとめると、次のようになる。

- (1) 高級言語記述によりリンケージされ、ユーザータスクとは独立に処理すること。
- (2) 機能選択により、コンパクトなものからデラックスなものまで自由に構成でき、メモリ容量、処理速度の点で最適化が図れること。
- (3) 実際に入出力したタイミングが、ユーザー側でとれること。
- (4) 入出力機器故障時のきめ細かな処理ができること。
- (5) ユーザー特有の機能が組み込めること。
- (6) ユーザーは、ユーザータスクとのリンケージを考えるだけで使いこなせること。
- (7) 入出力データの収集機能までもたせること。

以上のうち、特に(1)、(2)、(5)、(6)は、単に入出力処理だけでなく、一般のソフトウェア標準品に対するニーズとしても受け取ることができる。すなわち、ソフトウェア標準品は、

できる限りのブラック・ボックス性が要求され、更に、ユーザー側の機能選択及び専用機能の組込みにより、最適システムの構成ができるものでなければならない。また、(3)、(4)から、標準入出力システムの場合は、入出力機器故障時の対策が特に重要であり、ユーザーが自由に故障処理できるようにすべきである。(7)は、入出力処理の機能のわくを広げて、データベースの域まで対処することにより、ユーザー側負担を更に軽減したいというニーズであるが、この場合は、特に位置づけが問題となろう。

3 IOPF

前記のユーザー ニーズをもとに、HIDIC 80用IOPFを開発したが、その概要を次に記述する。

3.1 位置づけ

IOPFの位置づけを図1に示す。まず、広義のアプリケーション プログラムは、オペレーティング システムを介してハードウェアとリンケージされるが、これらは言語プロセッサを通して作成される。広義のアプリケーション プログラムを分類すると、標準と非標準とに大きく分けられ、また標準部分は、製品分野別標準サブシステムと共通標準サブシステムに分類される。共通標準サブシステムは、非標準プログラム、製品分野別標準サブシステムと、オペレーティング システムの間に位置し、非標準プログラムと製品分野別標準サブシステムの機能のうち、あるまとまった機能を肩代わりし分担する。共通標準サブシステムの中には、複数のプログラムパッケージがまとまって、ある機能を持っているものと、各各独立した機能を持つプログラム パッケージが集まって、ある機能集団となっているものとがあるが、IOPFは前者の形の共通標準サブシステムの一つである。

共通標準サブシステムは、オペレーティング システムの一部と、時々勘違いされることがあるが、オペレーティング システムは、主にハード機能をカバーする目的を持つものに対して、共通標準サブシステムは、主にアプリケーション プログラムの標準部分を拡張する目的を持つものであり、根本的相違がある。したがって、例えばハードウェアの仕様が変更された場合、基本的には、オペレーティング システム内でカ

* 日立製作所大みか工場 ** 日立エンジニアリング株式会社

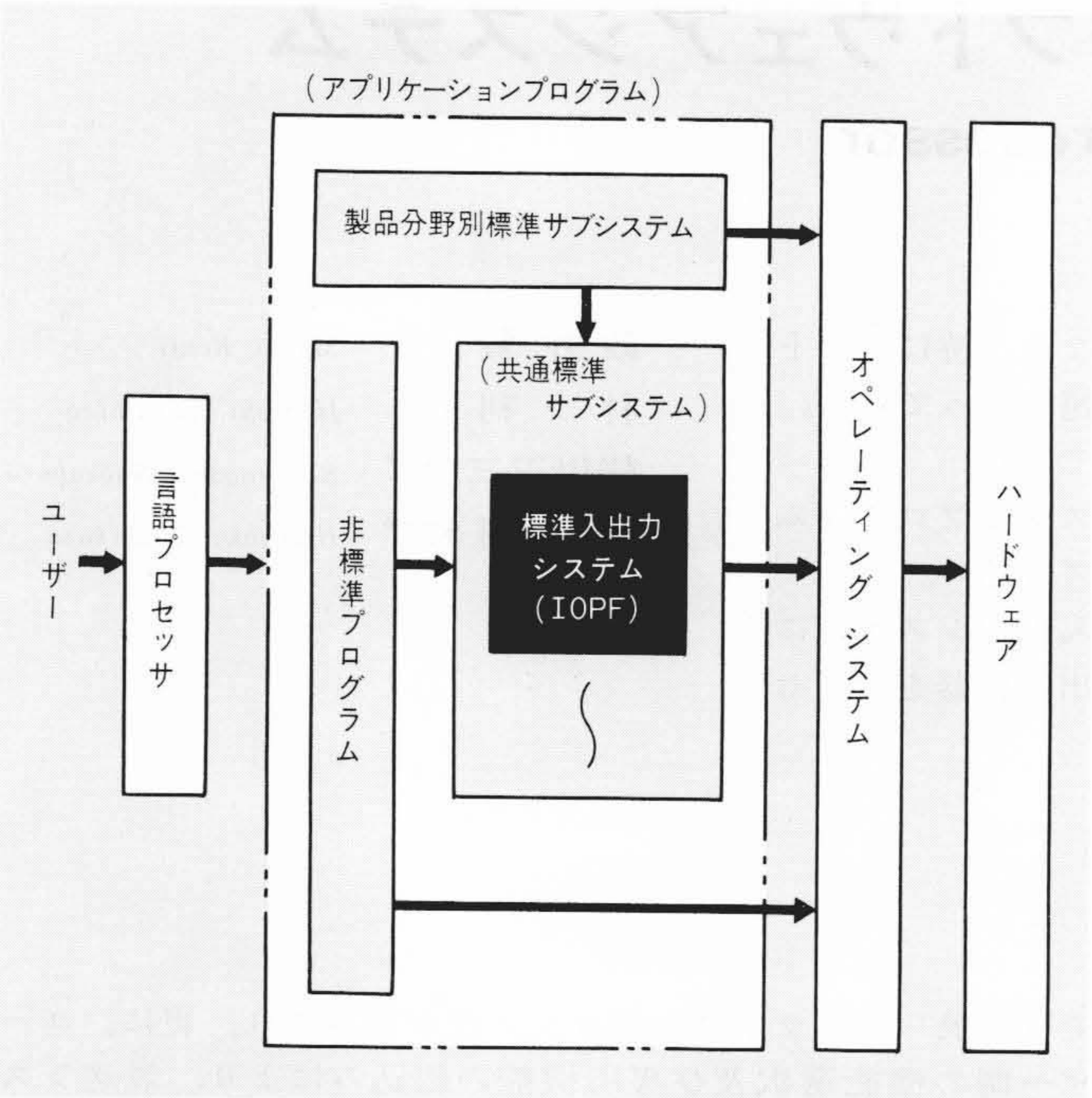


図1 IOPFの位置づけ IOPFは広義のアプリケーション プログラムに属した共通標準サブシステムの一つである。

バーすべきであって、アプリケーション プログラム、特に共通標準サブシステムに影響を与えないようにすべきであるということになろう。

3.2 特 徴

- IOPFは、次のような特徴を持っている。
- (1) 制御用高級言語 **Process Control Language**（以下、**PCL**と略す）のステートメントである、入出力文（**READ**文・**WRITE**文）の記述により、自動的にユーザープログラムとリンケージされる。
 - (2) 比較的低速の入出力機器をサポートし、入出力機器の同時処理、入出力要求の待ち行列管理により、ユーザータスクとは独立して入出力処理を実行する。
 - (3) 入出力機器の各種コードと、計算機内部コードとの変換はもちろん、データの編集機能をも持ち、**COBOL**、**PL/I**などの帳票作成機能と同レベルの機能がある。
 - (4) モジュール化の徹底により、ユーザーはモジュールの選択を行なうことにより、コンパクトタイプからデラックスタイプまでの最適システムを自由に構成できる。
 - (5) 入出力機器故障時の処理は、きめ細かく、更にユーザープログラムにリンクするので、入出力機器のバックアップなど、ユーザー特有の異常処理を付加することができる。
 - (6) ユーザーは実際の入出力実行完了のタイミング、及びどの入出力文で入出力機器が故障となったかを知ることができる。

3.3 機 能

IOPFは、表1に示すような機能を持っており、ほとんどの機能はユーザーが自由に取捨選択できるようになっている。次に各分類別に機能の概要を説明する。

(1) 通常処理

IOPFの単独イニシャライズができ、**PCL**のステートメントである入出力文、及びフォーマット文に対する処理を行なう。フォーマット変換としては27種を用意している。入出力のデータ型（整数・実数・論理・文字・16進の型や、1語・2語などの語長を含む）は、あらかじめ各データごとに指定

表1 IOPFの機能分類 ほとんどの機能は、ユーザーが自由に取捨選択できる。

番号	分 類	機 能	選 択
1	通 常 処 理	(1)IOPFイニシャルスタート機能	—
		(2)PCL順アクセス入出力文処理機能	○
		(3)フォーマット変換機能	○
		(4)フォーマット優先処理機能	◎
		(5)フォーマットNo.指定機能	◎
		(6)ページ コントロール機能	—
2	特 殊 処 理	(7)出力データ途中打切処理機能	◎
		(8)指定文字出力処理機能	◎
		(9)出力文の実行完了待ち機能	—
3	異 常 処 理	(10)出力機器異常時のレベル分け機能	◎
		(11)入出力機器故障時の特定タスク起動機能	◎
		(12)タスク強制停止時の処理機能	◎
		(13)入力データチェック機能	◎
4	ユーティリティ	(14)フォーマット登録・削除機能	◎
		(15)入出力機器の状態通知・修正機能	◎
		(16)入出力要求待ち行列の状態通知・キャンセル機能	◎

注：選択の欄の記号の意味
— ユーザーの選択は不可（基本機能）
○ ユーザーの選択は部分的に可（基本機能＋拡張機能）
◎ ユーザーの選択は全面的に可（拡張機能）

しておき、これに従ってフォーマット変換を行なうが、入出力文発行時にこれを一時的に変更できると、1個のデータに対し、*n*個のデータ エリアを用意せずに、何種ものデータ型で出力することができる。これを行なうのが、あらかじめ指定した入出力データ型よりフォーマットで指定した型を優先する「フォーマット優先処理機能」である。

フォーマット データはユーザータスク内に持つだけでなく、補助記憶装置に一括して全タスク共通に持つと、フォーマット変更時のユーザー側フォーマット管理がしやすくなり、フォーマット指定は、そのフォーマット データの登録No.だけでよくなる。これが、「フォーマットNo.指定機能」である。また、出力機器の改行・改頁の処理をするのが、「ページ コントロール機能」である。

(2) 特殊処理

1 出力文に対する出力データ及びデータ量をプロセス状態により、ダイナミックに変えることができると、複数の出力文、複数のフォーマットを用意せずともよくなる。これを行なうために、出力データの内容があらかじめ決めておいた値、又はパターンと一致したとき、その時点で、その出力文の処理を打ち切る「出力データ途中打切処理機能」、及び出力データの代わりにあらかじめ指定された文字を出力する「指定文字出力処理機能」がある。

また、出力文の場合は通常、出力要求受付完了の時点で、ユーザータスクは他の処理を実行できるが、これを出力文単位の指定により、その出力文の実行が完了するまでユーザータスクを待たせるようにすると、実際に出力したタイミングをユーザー側でつかむことができる。これが「出力文の実行完了待ち機能」である。

(3) 異常処理

出力機器の異常は、その内容により無視してよいものと、再試行により救えるもの、故障とすべきもの、及び機器その

ものの故障でなく単なるローカル状態のものなどあり、これらを分けて処理する必要がある。これが「出力機器異常時のレベル分け機能」である。

また、入出力機器が故障したとき、ユーザー側で自由な故障時処理ができるように、「入出力機器故障時の特定タスク起動機能」がある。更に、ユーザータスクがIOPFとリンケージ中に強制的にタスク停止されたとき、リンケージを解除する「タスク強制停止時の処理機能」、及び入力したデータの合理性チェックを行なう「入力データチェック機能」がある。

(4) ユーティリティ

前記「フォーマットNo.指定機能」を使用するためには、補助記憶装置にフォーマットデータを一括して登録しておかねばならない。これを行なうのが「フォーマット登録・削除機能」である。

また、ユーザー側できめ細かな使い方をするためには、入出力機器の異常状態及び入出力要求待ち行列の空き状態などを、ユーザー側で知り、これらを修正することが必要となる。また、入出力機器の故障を検知した入出力文及びその入出力文を発行したタスクなどをユーザー側で知ることにも必要となる。これを行なうのが「入出力機器の状態通知・修正機能」及び「入出力要求待ち行列の状態通知・キャンセル機能」である。

3.4 CRT表示例

IOPFを使用して、Cathode Ray Tube (以下、CRTと略す) に表示した例について説明する。表2には入出力文の一般形を、表3にはCRT表示のコーディング例を、また図2に表3のコーディングによって実際に表示されたCRT画面を示す。図2の画面は、タイトル①②、プラントデータ及び日時と3回の出力文によって表示されている。本例で使

(1) IOPFイニシャル スタート機能

IOPFを動作可能状態とするためにあらかじめ使用した機能

(2) PCL順アクセス入出力文処理機能

WRITE文を3回発行している。

(3) フォーマット変換機能

(a) !変換〔!β〕

βで示される制御文字から制御コードに変換する。使用した制御文字は、画面消去(!ERASE)、改ページ(!PAGE)、白色指定(!WHITE)である。

表2 入出力文一般形 パラメータの指定順は決まっているが、不要なパラメータは指定せずともよい。

項 目	一 般 形
入 力 文	READ (a:s, b:m, e, c, d) l
出 力 文	WRITE (a:s, b:m, e, c, d) l
説 明	a: ファイル参照番号(I/Oの識別番号) s: サブファイル参照番号(CRTのメモリ指定) b: フォーマットデータエリアアドレス m: モード(CRTのページ指定) e: エラー情報セットエリアアドレス c: 現在使用せず d: 出力完了待ちの指定 l: 入出力データエリアアドレス

表3 CRT表示のコーディング例 タイトル出力、プラントデータ出力及び日時出力と3回のWRITE文で表示している(図2とペアで参照)。

INTEGER	DATA, TIME, CRT	初期設定
単精度整数型指定		
DIMENSION	DATA(10, 9), TIME(5), CRT(1)	
エリア大きさ指定	9W×10ケース 5W 1W	
DATA	CRT/50/	
定数指定	CRT=50: ファイル参照番号	
WRITE(CRT, 100)	出力文	
フォーマット文のステートメントNO		
100 FORMAT(!ERASE, !PAGE, !WHITE,	フォーマット文	タイトル出力
ステートメントNO 画面クリア カーソルを 白色		
TOPへ		
1 'GAMEN NO=05 (TANK STATUS);'		
タイトル①		
2 V3, T3,		
カーソルを3行目の3文字目へ		
3 'TKNO LEVEL PRES VOL TEMP RATE ADD S'		
タイトル②		
WRITE(CRT, 200) DATA	出力文	
プラントデータ格納場所		
200 FORMAT<V4, T5,		
フォーマット優先指定		
1 10(Z2, 1X, T5, 1X, F4.2, 4(1X, T3, 1X), A1, 6X) >		
繰返し TKNO LEVEL PRES 繰返し VOL		
TEMP		
RATE		
ADD		
S 次の行 フォーマット		
位置へ 優先指定		
WRITE(CRT, 300) TIME	出力文	
日時データ格納場所		
300 FORMAT(V16, T7,		
'S', 3(P' 99.'), P' 99' ' : ', P' 99')		
年 月 日		
時		
分		

注: □で囲まれた部分がオンラインで変化するデータである。

タイトル	① GAMEN NO=05 (TANK STATUS)。
②	TKNO LEVEL PRES VOL TEMP RATE ADD S
プラントデータ	01 1200 1.15 4 240 300 50 B 02 10000 1.28 190 255 350 80 A 03 7000 1.20 55 230 400 100 B 04 5000 1.19 29 230 330 70 C 05 8000 1.18 72 225 380 60 B 06 12345 1.25 247 230 360 135 A 07 2050 1.22 9 260 375 10 C 08 1230 1.15 5 230 399 50 C 09 1900 1.10 9 225 348 20 C 10 5500 1.18 31 230 320 30 C
日時	S77.02.15.14:23

図2 CRT表示画面 表3のコーディング例により、実際に表示されたCRT画面を示す(表3とペアで参照)。

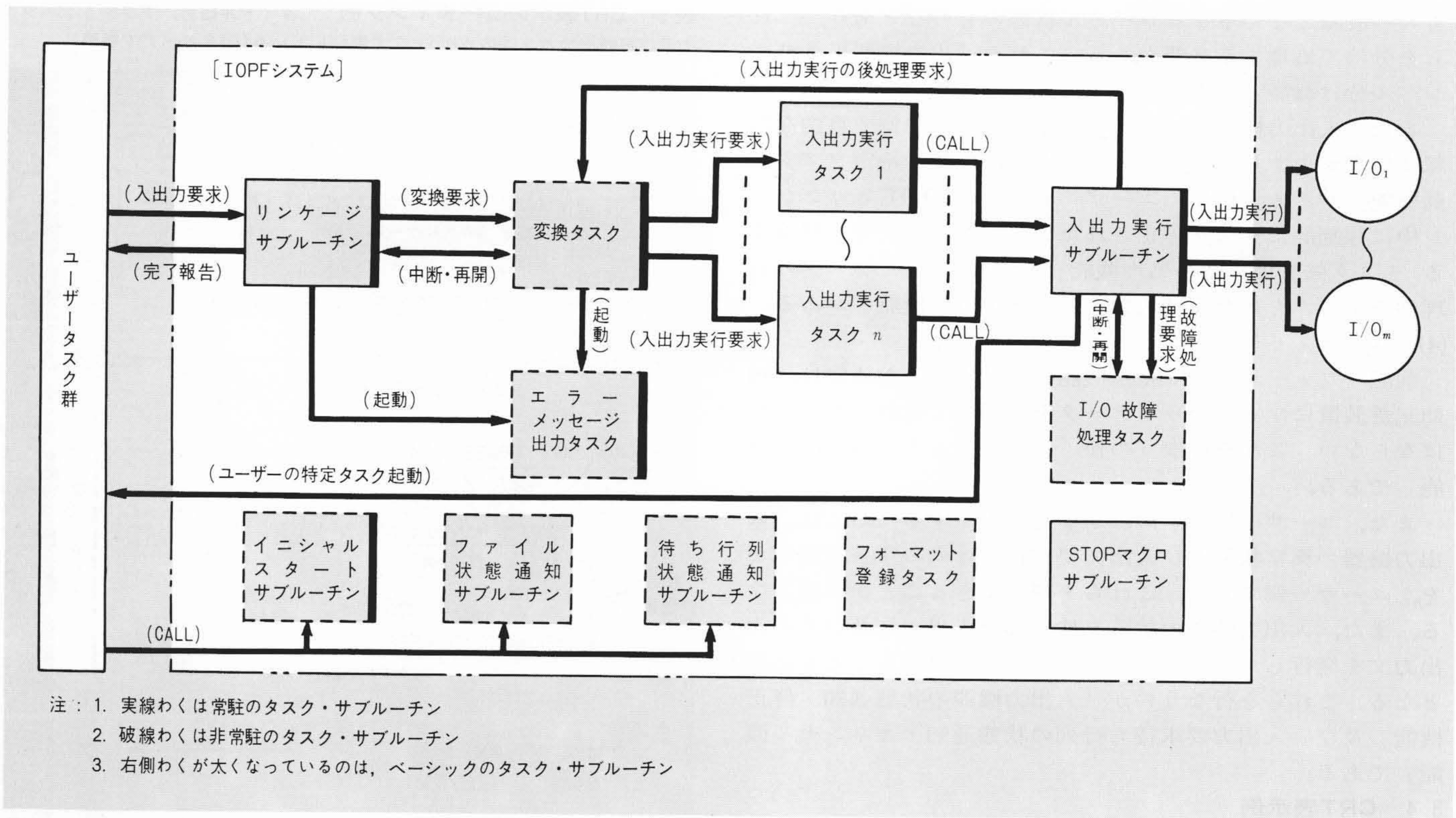


図3 IOPFのプログラム構成 11個のプログラムパッケージから構成され、入出力処理を行なう形となっている。

- (b) リテラル変換〔 $x_1x_2\cdots x_n$ 〕
 $x_1x_2\cdots x_n$ を文字データとして、そのまま文字コードに変換する。
- (c) V変換〔 V_p 〕
カーソルの垂直方向位置を上から p 行目とする。
- (d) T変換〔 T_p 〕
カーソルの水平方向位置を左から p 文字目とする。
- (e) Z変換〔 Z_ω 〕
16進数データとみて ω 個の数値文字に変換する。
- (f) X変換〔 nX 〕
 n 個の空白コードを付加する。
- (g) I変換〔 I_ω 〕
単精度整数型データとみて ω 個の数値文字に変換する。
ゼロサプレスは自動的に行なわれる。
- (h) F変換〔 $F_\omega \cdot d$ 〕
実数型データとみて小数部が d けたで ω 個の数値文字に変換する。
- (i) A変換〔 A_ω 〕
文字コードデータとみて ω 個の文字コードに変換する。
- (j) P変換〔 $P'x_1x_2\cdots x_n$ 〕
ピクチャ変換と呼び、 $x_1x_2\cdots x_n$ はピクチャ文字を使用する。表示例では、ピクチャ文字として9と.を使用している。9は単精度整数型データ用ピクチャ文字であり、1個が1けたに対応する。 .は挿入文字である。
- (4) フォーマット優先処理機能
フォーマット文の全体の括弧()を<>に変えて記述した。これは、初期設定でDATAを単精度整数型としたが、DATAの中のPRESに相当するデータが実数型であるために使用した。

3.5 構成

IOPFのプログラム構成を図3に示す。同図のように11個のプログラムパッケージから構成され、入出力処理を行なう形となっている。そのうちの一つである入出力実行タスク1から n は、基本的にはDFW(データフリーウェイ)を介して接続された入出力機器用に1個、DFWを介さずに直接接続された入出力機器用に1個用意すればよいが、1個の入出力実行タスクでの同時処理可能台数15台を超える場合などは、

表4 IOPFモジュール分類 拡張モジュール31個は、ユーザーが自由に取捨選択できる。

番号	大 分 類	小 分 類	モジュール個数	
			基 本	拡 張
1	入 出 力 機 能 モ ジ ュ ー ル 群	PCLステートメントに 対応するモジュール	1	1
		入出力機器種類に 対応するモジュール	1	4
2	フォーマット変換機能 モ ジ ュ ー ル 群	PCLフォーマット文に 対応するモジュール	7	12
		その他モジュール	0	3
3	特 殊 機 能 モ ジ ュ ー ル 群	異常処理モジュール	2	6
		その他モジュール	1	5
合 計			12	31

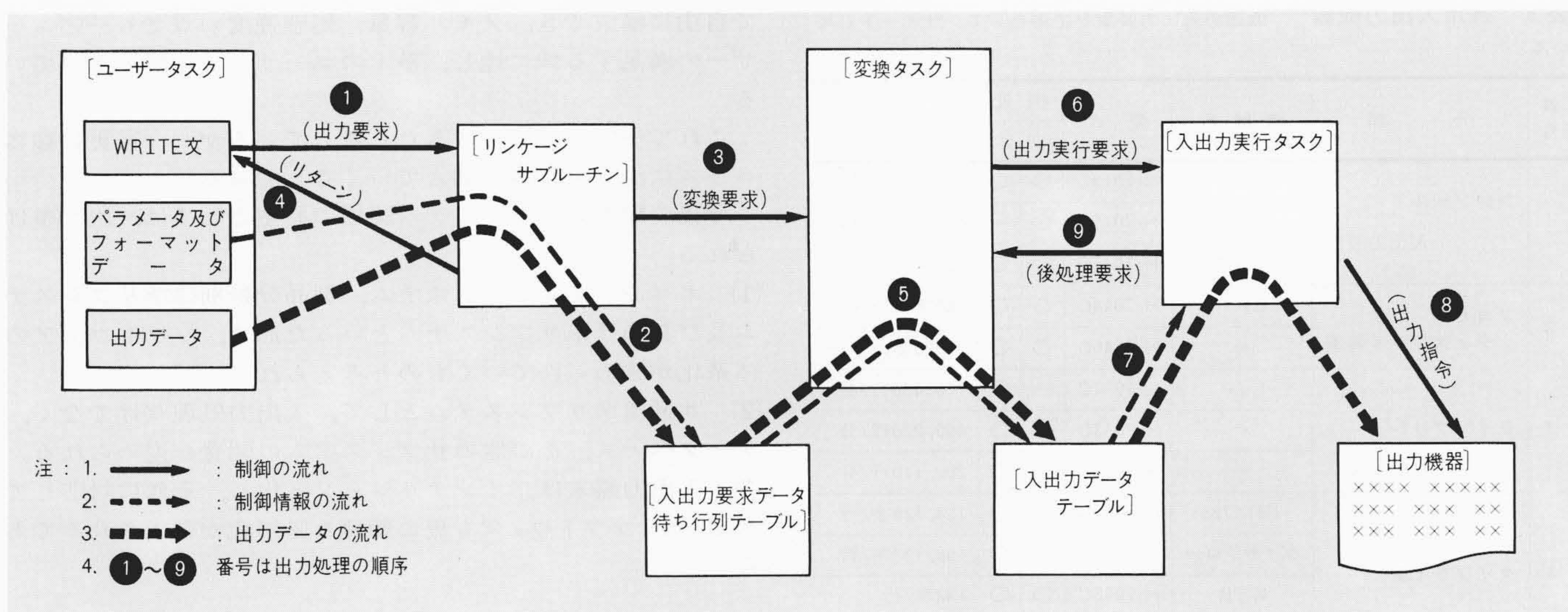


図4 出力処理の流れ ①～⑨の順で出力要求が処理される。

更に入出力実行タスクを1個用意することになる。

なお、これらのプログラム パッケージは、更に機能的に分割され各々モジュールとなっている。表4にそのモジュールの分割を示す。モジュールは、大きく入出力機能、フォーマット変換機能、特殊機能の各モジュール群に分かれ、更に、細かな機能単位に分割されている。同表に示すように基本モジュール12個と拡張モジュール31個、計43個のモジュールに分かれており、各々、各プログラム パッケージの中に分散している。

3.6 動作

次にIOPFの動作について、出力動作を例に取り説明する。図4に出力処理の流れを示す。

まずユーザータスク内でWRITE文を記述すると、PCLコンパイラが、IOPFのリンケージ サブルーチンを、CALL

する形のオブジェクトを作成する。これが出力要求となり、図4に示すように①から⑨の順でその出力要求が処理される。リンケージ サブルーチンにより、出力要求が受け付けられた時点で、ユーザータスクとIOPFは完全に切り離され、各々独立に処理を続行できる。

3.7 異常処理

IOPFの機器異常処理を図5に示す。入出力実行中に入出力機器が異常となった場合は、その機器に対してユーザーがあらかじめ指定した内容を判定し、レベル分けの場合は、異常の内容(異常ビット)まで調べて各々の処理を行なう。故障となり起動されたユーザーの特定タスクは、ユーザー特有のバックアップ処理などを行なうことができる。

3.8 適用入出力機器

IOPFの適用入出力機器を表5に示す。

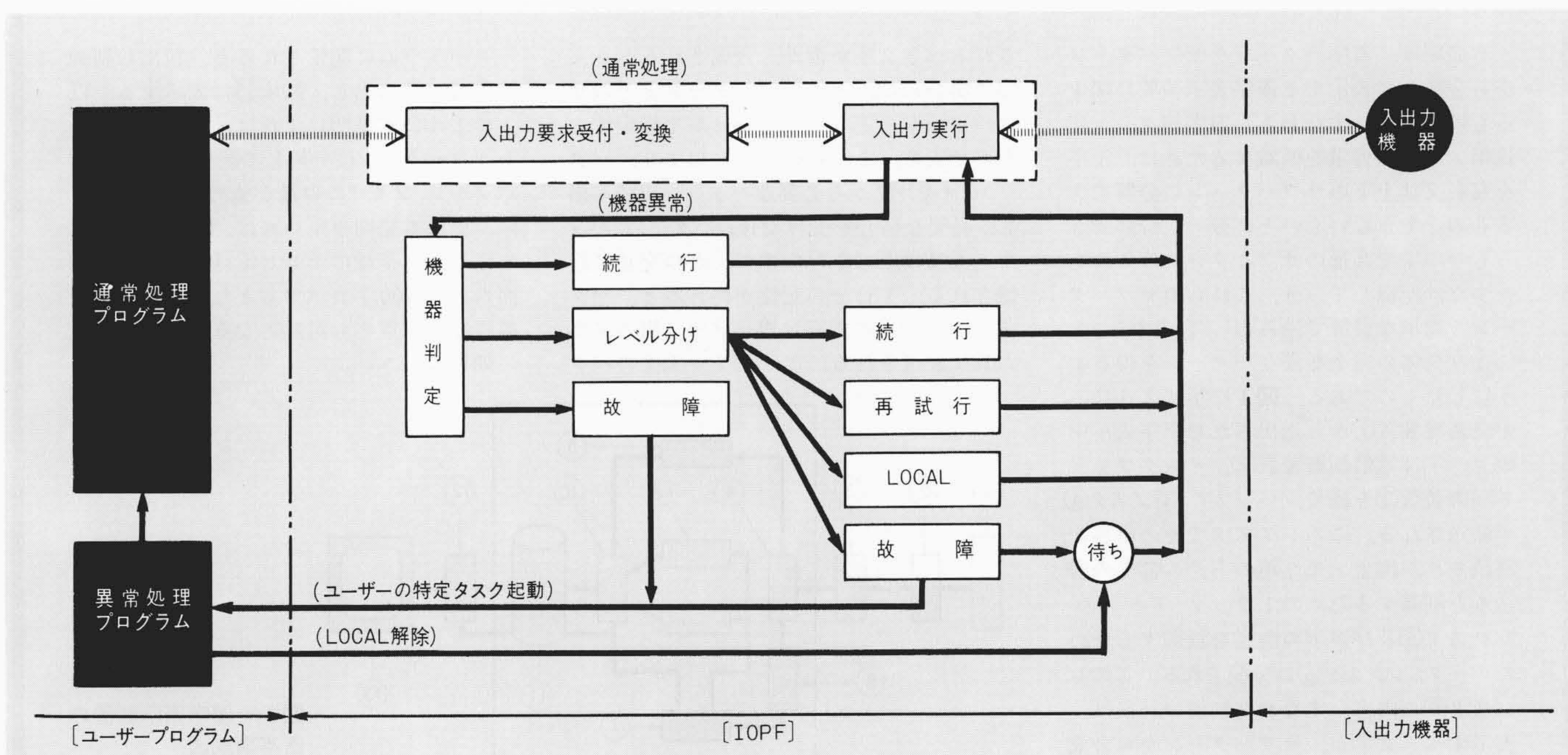


図5 IOPFの機器異常処理 故障となるとユーザーの特定タスクが起動されるので、ユーザー特有の異常処理を付加できる。

表 5 適用入出力機器 低速の入出力装置を主体として、サポートしている。

番号	分類	機器名	型式	機能		備考
				入力	出力	
1	コンソール 入出力装置	ASR	H-7013C	○	○	72字/行
		PTR	H-7015C	○	—	400, 480字/秒
		ASR	H-7017C	○	○	132字/行
2	プロセス ディスプレイ装置	CRT	H-7844C	○	○	960字カラー
		"	H-7845C	○	○	1,920字カラー
3	ラインプリンタ	L/P	H-7032C	—	○	760, 430行/分
		"	H-7033C	—	○	400, 220行/分
		"	H-7034C	—	○	200, 110行/分
4	タイプライタ	IBM 735	H-7863C	—	○	154, 128字/行
		ダイヤブロー	H-7864C	—	○	158, 132字/行
		KSR	H-7898C	○	○	132字/行
		TTY	H-7894C	—	○	72字/行
5	カード読取装置	C/R	H-7861C	○	—	600, 300枚/分
		MC/R	H-7882C	○	—	"
6	紙テープせん孔装置	PTP	H-7866C	—	○	60字/秒
7	カードせん孔装置	C/P	H-7862C	—	○	26枚/分

4 標準入出力システムの今後の展望

以上述べたように、HIDIC 80用IOPFは、豊富な機能を持ち、ユーザーがコンパクトタイプからデラックスタイプま

で自由に構成でき、メモリ容量、処理速度の点でも一応ユーザーの満足する域に達し、多くのユーザーから好評を得ている。

これで一応の基盤はできたと考えているが、今後更に顧客ニーズに沿って改良を加えていく考えである。

今後の標準入出力システム関係の展望としては、次が挙げられる。

- (1) オペレーティングシステム、製品分野別標準サブシステム及び共通標準サブシステムといった形で、ソフトウェアの体系化が強められていくものと考えられる。
- (2) 共通標準サブシステムとして、入出力処理だけでなく、データベース、その他のサブシステムの開発が進められる。
- (3) 入出力端末は、インテリジェント化し、それに対応してサポート ソフトウェアも更に明確な階層化が要求されるであろう。

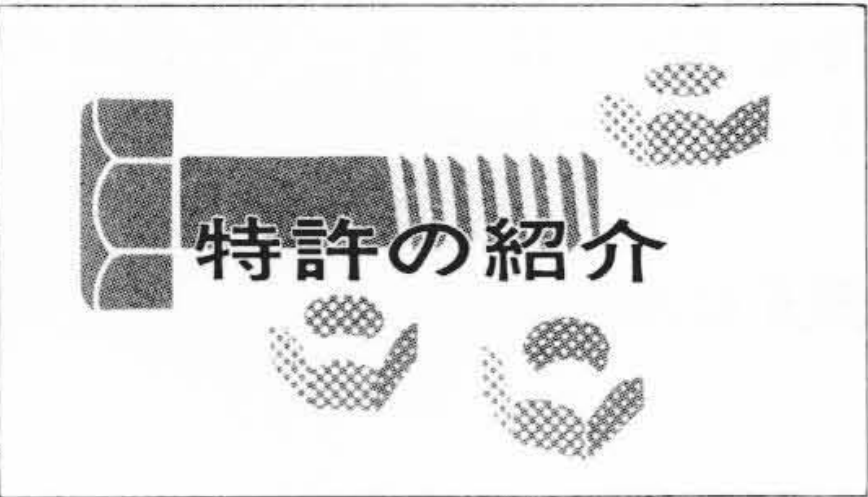
5 結 言

以上、HIDIC 80用標準入出力システムIOPFを中心に説明したが、標準化の考え方はソフトウェア全般に当てはまるものであり、アーキテクチャ、位置づけ及び限界を明確にして標準化すべきであろう。

最後に、HIDIC 80用IOPFを開発するに当たって御協力をいただいたユーザー各位及び関係者の方々に対して深く感謝の意を表わす次第である。

参考文献

- 1) 桑原、平井ほか：HIDIC 80オペレーティングシステム、日立評論、58、653（昭51-6）



漢字表示装置

寺井秀一・中田和男
特許 第807395号（特公昭50—23570号）

本発明は、漢字パターンをサブパターンから合成して表示する漢字表示装置に関するものである。すなわち、漢字パターン記憶用メモリの容量を低減するために、漢字を左右又は上下のサブパターンに分解できるものとできないものとに分け、分解できるものの中で共通のサブパターンはこれを一つだけ記憶しておき、これらのサブパターンを簡単な装置で読み出し、合成することにより全体の完全な漢字パターンを得るようにしたものである。図1に示すように、中央処理装置①から送出された漢字表示用のコードは通信制御装置②、インタフェース制御装置③を経て、バッファレジスタ④へ転送される。このレジスタでその内容が解読され、再生メモリ用のドラム⑪への書き込みを制御するためのトラックアドレスレジスタ⑤及び漢字の内容を制御するためのデータレジスタ⑥に分配される。このレジスタ⑥の内容、すなわち制御コードは、表示すべき文字が、サブパターンから合成

されるべき文字か否か、合成されるべき文字の場合には、どのサブパターンとどのサブパターンのどういう組合せかを指定するものである。この制御コードにより、パターンメモリ⑦から必要なパターンを読み出し、必要な結合を受けた後、パターンバッファ⑧又は⑨に1行分ずつ、かつ交互に記憶される。1行分の記憶が終わると、パターンバッファの内容は再生メモリ用のドラム⑪に記憶される。こうして一画面のパタ

ーンがドラムに記憶されると、読出し制御装置⑫で読み出し、端末⑬に表示する。以上のように本発明によれば、漢字パターンをサブパターンに分解して記憶するために、パターンメモリの容量が極めて小さくてよい。更に本発明を用いれば、例えば、1MHzの信号伝送帯域のテレビ電話の場合、一画面当たり100字表示するとしても約300回線程度の多重化も可能となるなど、実用上の効果は大きい。

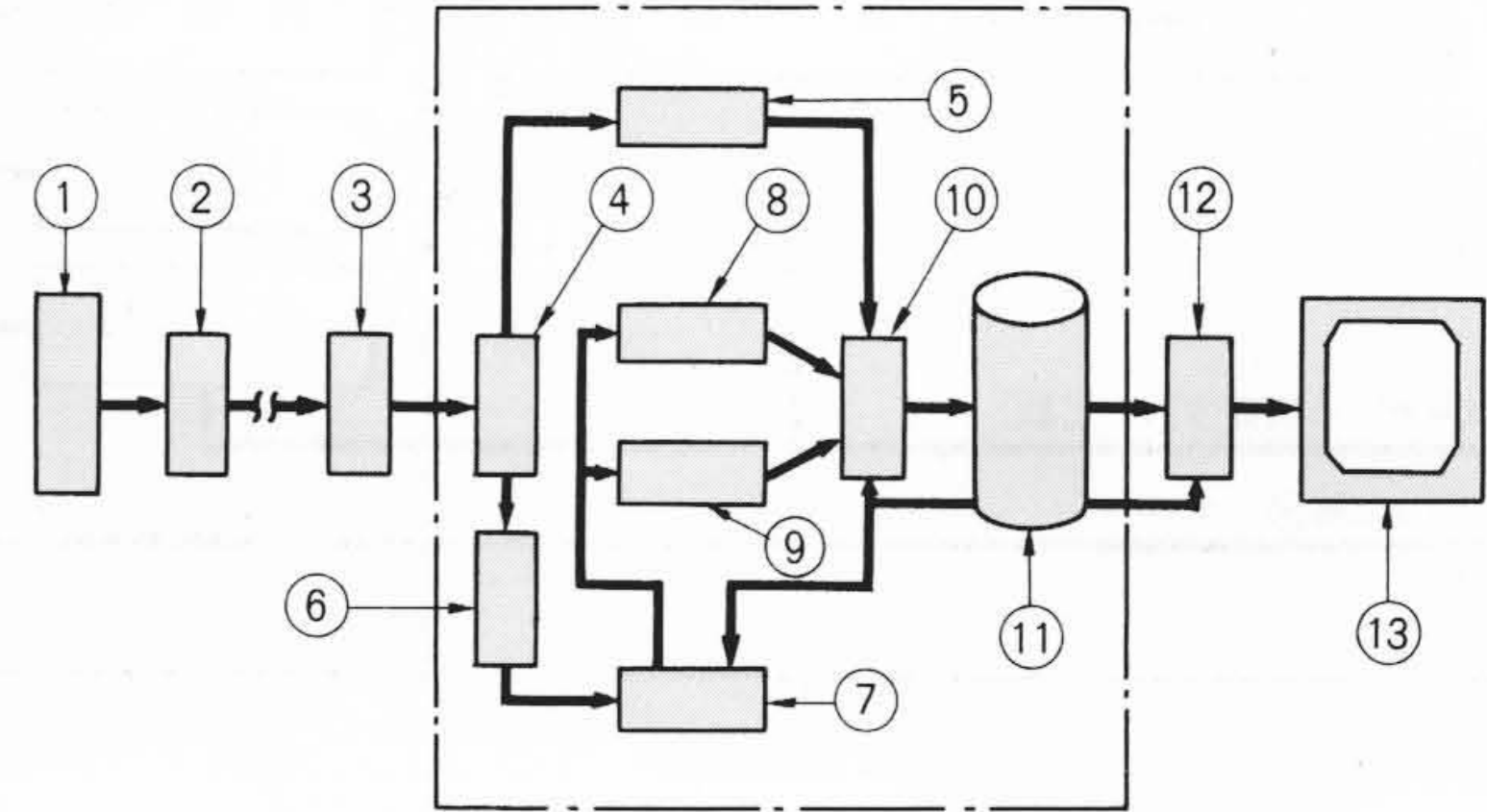


図1 漢字表示装置の基本構成図