

COBOLフローチャート ジェネレータ “HIFLOW/C”の開発

The Development of COBOL Flowchart Generator HIFLOW/C(Hitachi Autoflow/COBOL)

COBOLフローチャート ジェネレータ“HIFLOW/C”(“Hitachi Autoflow/COBOL”)は、COBOL言語で書かれたソース プログラムから、正確で実用的なフローチャートをHITAC 8000シリーズ 電子計算機とX-Yプロッタを用いて、自動的に作成するコンピュータ ドキュメンテーション システムである。

“HIFLOW/C”はフローライン処理、ランチャートの出力、ロジカル フロー機能、フローチャートの部分出力機能など種々の特長を有しており、ドキュメント整備のための一つの道具として使用するとともに、ドキュメントの標準化としての意義をもっており、プログラム作業の標準化、自動化に大きく貢献するものと考えている。

鈴木 健二* Kenji Suzuki
佐々木 静男* Sizuo Sasaki
大貫 政男* Masao Ônuki
浜田 謙蔵* Kenzô Hamada

1 緒 言

日立製作所日立工場(以下、日立工場と略す)に電子計算機が導入されてから14年目になる。この間に設計部門、製造部門、管理部門と各分野の機械化システムが開発されてきた。更にこれらの機械化システムは有機的に結合したトータル システムへと展開している。

このように、システムが相互の有機的結合により大型化し、発展的・流動的に変化してくると、新規システム計画の開発もさることながら、既成システムの維持・管理に要する作業も膨大なものとなり、プログラマはプログラムの作成や修正に追われ、コンピュータ ドキュメンテーションの中でも特にプログラム管理に不可欠なフローチャートを処理内容の変更の都度修正することは、ともすれば怠りがちになる。

そのうえ、システムの開発を担当したプランナやプログラマは、固定的なものでなく、数年を経過したシステムの維持・管理にはコンピュータ ドキュメントの完備と標準化が不可欠であり、その必要性・重要性が叫ばれているのが現状である。

このような背景のもとに、日立工場においてはCOBOLソース プログラムから逆にフローチャートを自動的に作成するジェネレータを作成すべく、昭和48年よりフローチャート パッケージの調査・検討を開始し、同年2月より設計、5月よりプログラミングに入り、10月に“HIFLOW/C” VER-01システムとして完成した。以降、“HIFLOW/C”の活用段階に入っているが、昭和49年1月よりバージョン アップを行ない、同年4月に“HIFLOW/C” VER-02が完成、現在このVER-02を最終バージョンと考え使用している。

2 “HIFLOW/C”の概要

“HIFLOW/C”とは、“Hitachi Autoflow/COBOL”の略称であり、図1に示すようにHITAC 8000シリーズ 電子計算機及びX-Yプロッタを用いて、COBOLソース プログラムから実用的な“HIFLOW/C”フローチャートを作成するコンピュータ ドキュメンテーション システムである。

COBOLソース プログラムの入力方式としては、ダイレク

ト カード入力と磁気テープ入力の二通りがあり、磁気テープ入力としてカード ツー テープしたカード イメージのもの、ソース ステートメント ライブラリ(SSL)のフォーマットのものなどが可能となっている。

使用する電子計算機は、HITAC 8300/8400/8500(8350/8450)、メモリ サイズは262KB(196KBも可)、オペレーティング システム(OS)にはExtended Disk Operating System(EDOS)を使用している。また、X-Yプロッタにはカルコン社の900/937/1136型X-Yプロッタを使用しており、出力結果としてランチャート、ディテール フローチャート若しくはロジカル フローチャートを得ることができる。

なお、X-Yプロッタに関しては他機種であってもカルコン1136型と同機能をもつものであれば、“HIFLOW/C”の使用は可能である。

2.1 “HIFLOW/C”の構成

“HIFLOW/C”は図2に示すように、大きくPHASE-1と

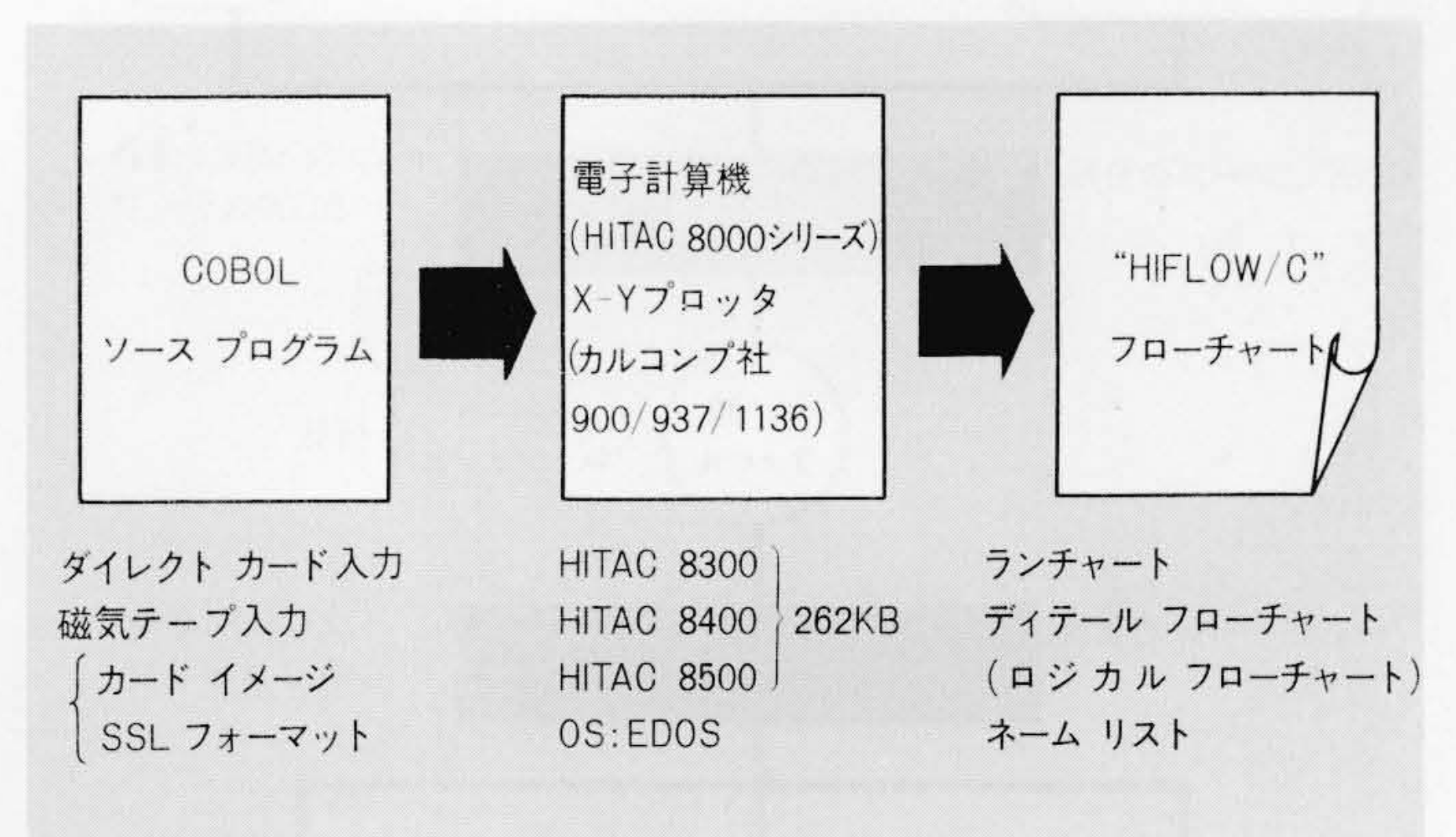


図1 “HIFLOW/C”の概要 “HIFLOW/C”は、HITAC 8000シリーズ及びX-Yプロッタを用いて実用的なフローチャートを作成するコンピュータ ドキュメンテーション システムである。

*日立製作所日立工場

PHASE-2 の二つのプログラムから構成される。

PHASE-1 は、COBOL ソース プログラムをカード又は磁気テープから読み込み、主に COBOL ソースの分解を行なう。COBOL ソースの分解結果は、ランチャート情報及びフローチャート情報に分けて編集し、中間ファイルである磁気テープに出力する。

PHASE-2 は、PHASE-1 が出力した中間ファイルを読み込み、ここで実際にフローチャートへ展開することになる。

PHASE-1 は、アセンブリ言語で書かれたプログラムで約 5,000 ステップ、PHASE-2 は FORTRAN を使用しており約 2,000 ステップとなっている。但し、PHASE-2 のステップ数には、プロット関係のサブルーチンは含んでいない。

2.2 “HIFLOW/C” の特長

(1) ランチャートの出力

“HIFLOW/C” が出力するフローチャートの表紙の下半分に書かれるランチャートは、プログラムの入・出力ファイル情報が一目で分かるようにした構成図である(図 6 参照)。

ランチャートの中央には、プログラム名称と使用する計算機名称を記入し、上側にはインプット ファイル情報を、下側にはアウトプット ファイル情報をそれぞれ最大七つまで記入することができる。

入・出力情報ファイルとして、ランチャート記号の中にはファイル名称を、その上側には recording mode, block size, record size, label 情報を、更に下側には装置名称を記入する。

なお、これらのファイル情報は、ユーザーがパラメータで指定するのではなく、COBOL ソースの DATA DIVISION のファイル記述(FD)や OPEN の I/O(INPUT/OUTPUT)情報などにより “HIFLOW/C” が自動的に作成するものである。

(2) BOX No. 方式

“HIFLOW/C” には二つのバージョンがある。表 1 は VER-01 と VER-02 の比較表であるが、当初 VER-01 では、フローチャート上のブランチ方式としてコネクタを使用する場合、

035 の 120, A-RTN というように COBOL ソースの PAGE No., LINE No. 及びパラグラフ名でブランチ情報を表現していたが、これではブランチ先である A-RTN がフローチャート上のどの部分に書かれているのかを見つけるのが容易でなく、VER-02 にてこのブランチ情報に大改造を加えたわけである。

VER-02 では、7 の 24, EOF というようにフローチャート上の PAGE No., BOX No. 及びパラグラフ名をブランチ情報とした。これによると、フローチャートの 7 ページ目、頭から 24 番目の BOX が EOF であるというように容易にブランチ先を探すことができる。

また、VER-02 では OS を Tape Disk Operating System / Disk Operating System (TDOS/DOS) から EDOS へ移行したことも挙げられる。

(3) フローラインの処理

フローライン(流れ線)の処理が、フローチャート作図のテクニック上、最も難しい部分であるといえる。

“HIFLOW/C” におけるフローラインは、図 3 のように各列の左右にそれぞれ 4 本、フローラインを引く順序は内側よ

表 1 VER-01, VER-02 の比較 VER-02 にて、フローチャート上のブランチ情報に大改造を加え、BOX No. 方式を採用した。

バージョン 項 目	VER-01	VER-02
ブランチ方式	035 120 A-RTN COBOL ソースの PAGE, LINE No. パラグラフ名	7 24 EOF フローチャート上の PAGE, BOX No. パラグラフ名
OS	TDOS/DOS	EDOS

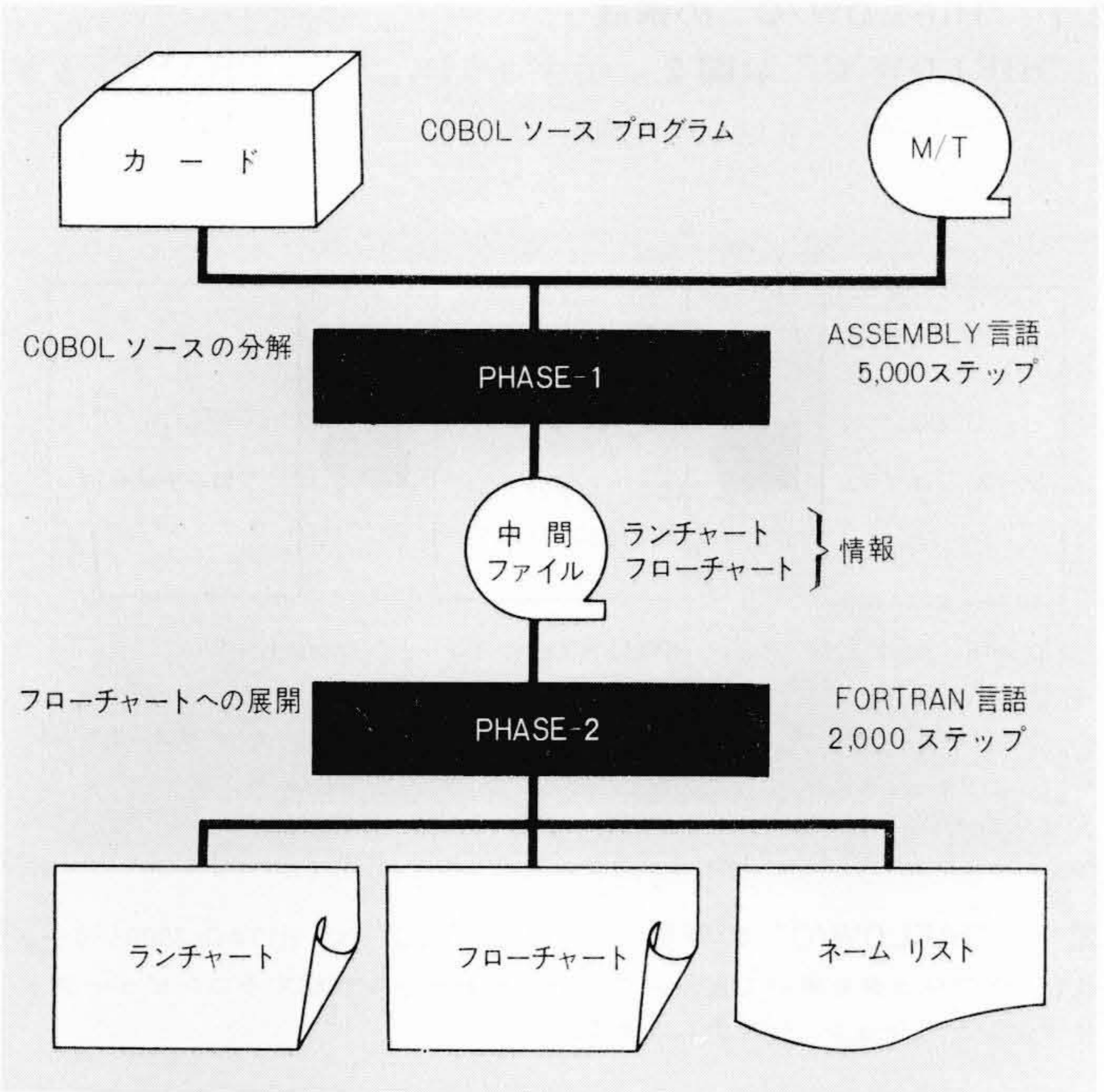


図 2 “HIFLOW/C” の処理 “HIFLOW/C” は大きく PHASE-1 と PHASE-2 の二つのプログラムから構成される。

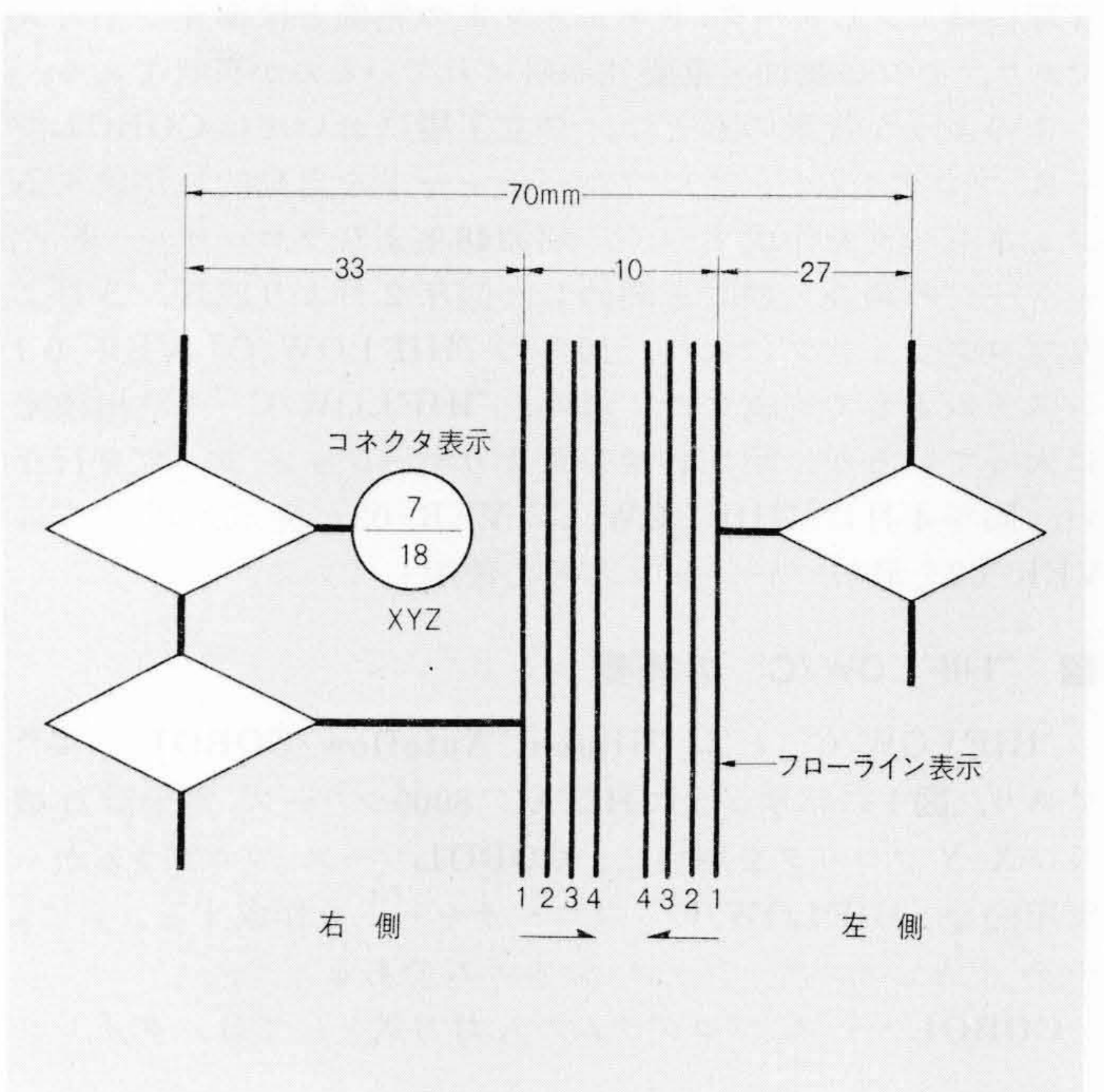


図 3 フローライン “HIFLOW/C” は、各列の左右に最大 4 本のフローラインを引くことができる。

り1本、2本、3本、4本と引いていき、既に4本とも使用済みの場合は、コネクタでブランチ先を表示することになる。

フローラインの本数については、現在までの出力結果を調査したところ、せいぜい2、3本しか使用されていないことから、4本で十分と考えている。

(4) フローチャート用紙の節約

フローチャートは、フローチャート用紙の上から下へ、左から右へと順次展開していくが、第4列目を書き終わると次のページに移ることになる。

“HIFLOW/C”ではこの時点で、フローチャート上の残りスペース(下方スペース)をチェックし、残りスペースが必要以上あれば、第4列目の続きを下方スペース上に続けて書く機能をもっている。すなわち、第4列目から第1列目に移り、また同様に同一ページ内にフローチャートを展開していくわけである。

このような方法を取ることで、n段書きが可能となった。これにより高価なフローチャート用紙(プロッタ用紙)の大幅な節約ができ、更に同一ページ内のBOXの数が増えることにより、ブランチ先が追いやすく、見やすいフローチャートにすることができた。図7は2段書きの例である。

(5) ロジカル フロー機能

図4において、左側の一般のディテール フローにロジカル フロー機能を使用すると、右側のようなロジカル フローが作成できる。すなわち、ロジカル フロー機能とはフローチャートを展開する段階で、特定のステートメント、例えばMOVE、ADD、COMPUTEなどが現われた場合、これらを“PROCESS”という一つのBOXにまとめて表示することである。

従って、フローチャートの出力量を削減することができる。同時に、IF、GO TOなどの流れ主体のフローチャートが作成でき、プログラムのロジックを理解するうえで有効なものになる。

(6) フローチャートの部分出力機能

この機能はPHASE-1入力であるパラメータに、COBOL

ソースの「範囲」をPAGE、LINE No.で指定し、プログラムの一部分(例えば、修正部分のみとか、ある特定のSECTIONのみ)をフローチャート化するフローチャートの部分出力機能である。この機能は、主に修正頻度の高いプログラムに適用すると有効である。

(7) ネーム リストの出力

ライン プリンタの出力結果であるネーム リストは、COBOLソースのPGLN(6けた)及び Procedure Nameとフローチャート上のPAGE No.との関連を Procedure Nameのアルファベット順に記述したインデックス リストである。

ネーム リストの出力要求は、パラメータにて指定可能であり、フローチャートと併用することにより、参照したいステートメントがフローチャート上のどの部分に書かれているかを即座に見付けることができる。

(8) フローチャート記号

“HIFLOW/C”のフローチャート用紙は、A3サイズの縦長とし、横方向4列方式を採用して1列目から2列、3列、4列と順次展開していく。フローチャート上のMOVE、COMPUTEなどの処理、IF、AT ENDなどの判断、ALTERを示す準備記号などをフローチャート記号と呼び、“HIFLOW/C”ではJIS規格に準じた9種類の記号を用意している(表2参照)。

フローチャート記号の中に書く文字の高さは、すべて2mmコンスタントにし、フローチャート記号の大きさは「整然とした美しさ」を出すために横幅を固定(40mm)し、縦方向を文字数による可変長とした。

3 “HIFLOW/C”の能力

図5に“HIFLOW/C”の能力について示す。

“HIFLOW/C”では、COBOLソースプログラムのProcedure Division 40~50ステップが、A3サイズのフローチャート用紙1枚になる。COBOLソースの40~50ステップといえば、コンパイル リスト約1ページ分に相当するので、容

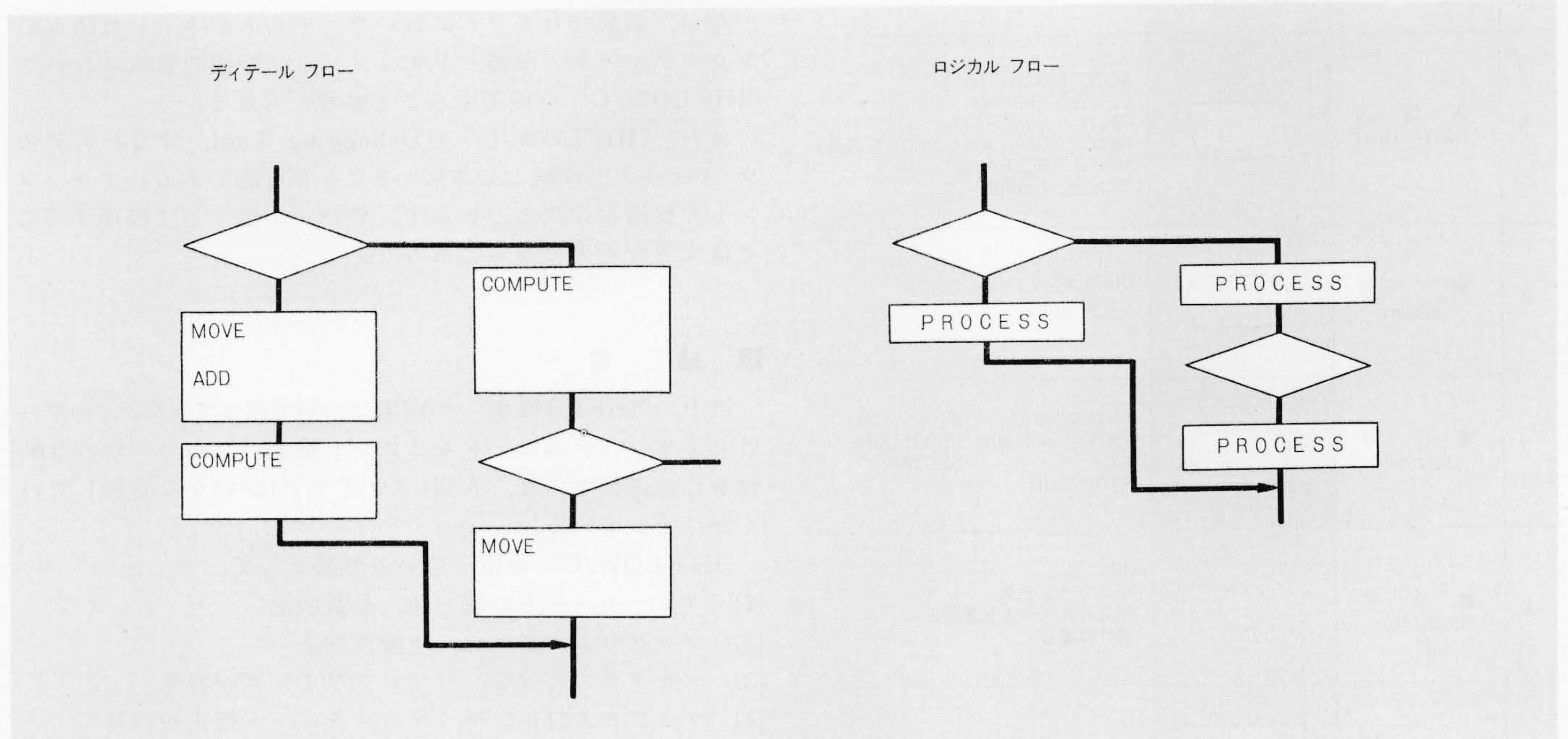


図4 ロジカル フロー機能 左側のディテール フローにロジカル フロー機能を使用すると、右側のようなロジカル フローが作成できる。

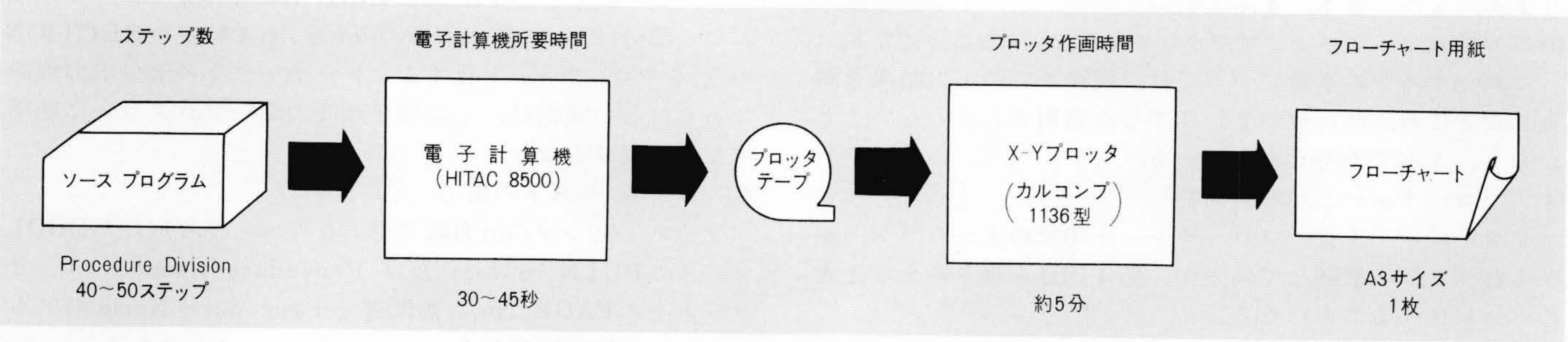


図 5 “HIFLOW/C” の能力 COBOL ソース プログラム 40~50 ステップが、A 3 サイズのフローチャート用紙 1 枚になる。

表 2 フローチャート記号 “HIFLOW/C” は、JIS 規格に準じた 9 種類の記号を用意している。

タイプ	意 味	フローチャート記号	対 象 命 令
1	処 理 process		ADD COMPUTE DIVIDE EXAMINE MOVE MULTIPLY SUBTRACT TRANSFORM USE INITIATE TERMINATE GO(ALTERED) GO(DEPENDENT)
2	判 断 decision		IF ON AT END INVALID ON SIZE ERROR
3	準 備 preparation		ALTER
4	定義済み処理 predefined process		ENTER(CALL) PERFORM SORT INCLUDE
5	入・出 力 input/output		ACCEPT REWRITE CLOSE RETURN OPEN GENERATE READ DISPLAY (CONSOLE 以外) WRITE RELEASE TRACE EXHIBIT
6	表 示 display		DISPLAY STOP literal
7	端 子 terminal		ENTER(ENTRY) STOP literal ENTER(RETURN) PROCEDURE EXIT SECTION STOP RUN
8	結 合 子 connector		GO IF のブランチ先 用紙のスペース上連続して 書けぬ場合
9	注 釈 comment		NOTE

易にフローチャート枚数を推定できる。

フローチャート 1 枚当たりの電子計算機所要時間は、HITAC 8500 を使用した場合、約 30~45 秒、プロッタでの作画時間はカルコンプ 1136 型を使用して 1 枚当たり約 5 分程度である。

4 “HIFLOW/C” の出力例(抜粋)

“HIFLOW/C” の出力例として、表紙(ランチャート) 及びディテール フローチャートの実例を図 6, 7 に示す。

5 “HIFLOW/C” の運用

“HIFLOW/C” は次のような考えを基にして運用され则认为る。図 8 のように新規プログラムと既成プログラムとに分けて考えてみると、新規プログラムに関しては、プログラム仕様書、フローチャート(ブロック ダイアグラム)、コーディング、テストなどの一連の流れにおいて、フローチャートはあくまでも「手書き」で作成することになり、この部分は従来となんら変わらない。

“HIFLOW/C” は既成プログラムにおいて、プログラムの修正、改造などを行なった場合、若しくは該当プログラムの担当者のローテーション対策に使用し、自動フローチャートを作成することになる。

なお、新規プログラムにおいて、テストが完了した時点でフローチャートの清書、ドキュメントの整備の意味において“HIFLOW/C” を使用することも考えられる。

また、“HIFLOW/C” を Debugging Tool, すなわちデスク チェック用資料として用いることも可能である。ドキュメントの整備だけにとどまらず、デバッグ用として使用することは大きな効果となるであろう。

6 結 言

以上、“HIFLOW/C” の概要について述べてきたが、いずれにせよ本システムは、ドキュメント整備のための一つの自動化された道具であり、人間(プログラマ)がいかに活用していくかに掛っている。

“HIFLOW/C” が狙っている効果として、

- (1) フローチャートの統一化、品質向上
- (2) プログラムの標準化、生産性向上
- (3) プログラムの保守、リプログラミングが容易
- (4) プログラム訂正に伴うフローチャート修正が容易

などが挙げられるが、“HIFLOW/C” の効果はなんといっても「フローチャートの標準化」であるといえる。フローチャートの標準化により、フローチャートから属人性を排除する

DATE06-04-'74

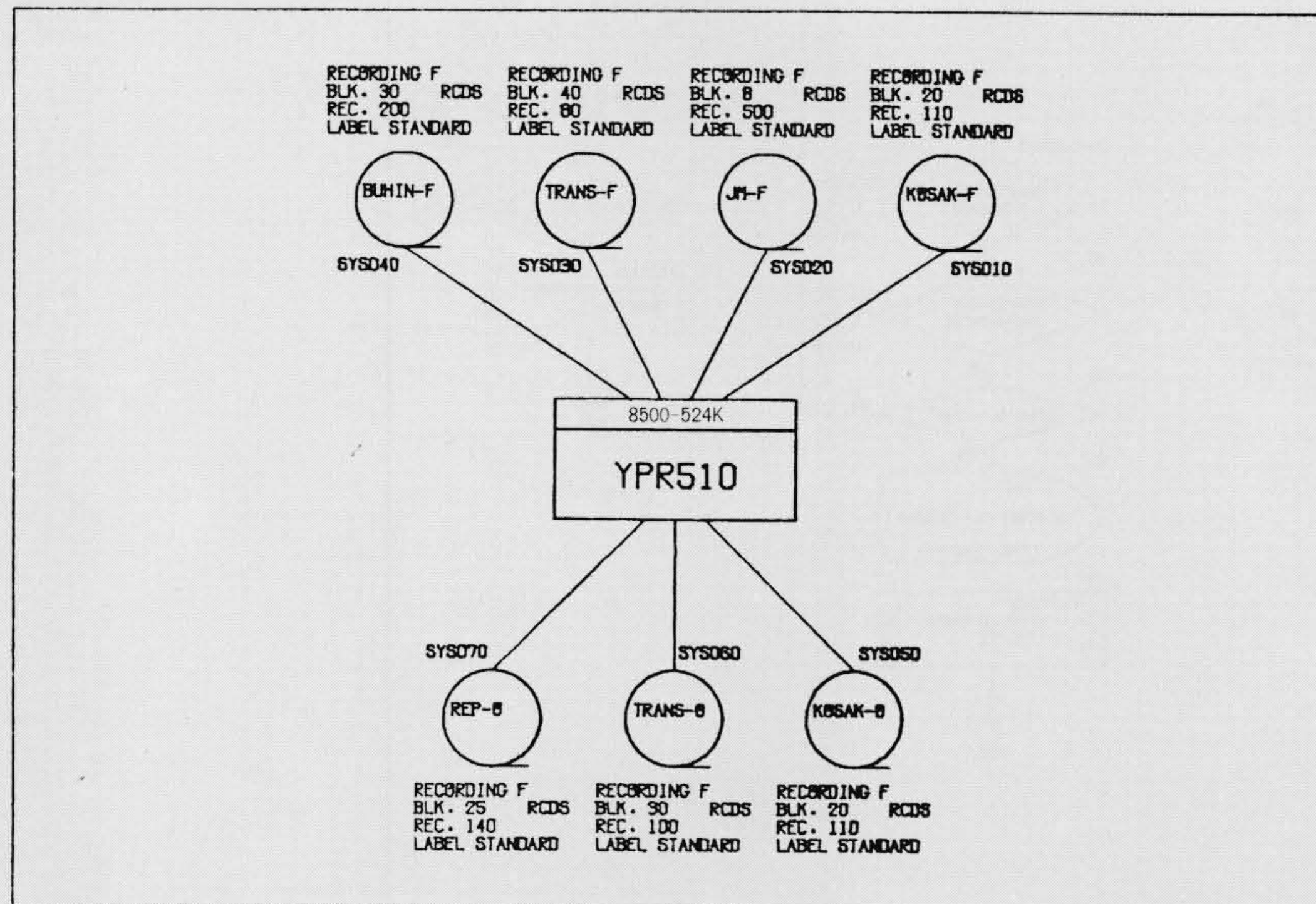
HITACHI AUTOFLOW COBOL

PROGRAM ID. YPR510

AUTHOR.

DATE-WRITTEN. 73-08-22.

RUN CHART



HIFLOW/C VER-02

図6 例1：表紙（ランチャート） ランチャートは、プログラムの入・出力ファイル情報が一目で分かるようにした構成図である。

ことができ、プログラム担当者のローテーション対策にも結び付くことになる。

更に、本システムは COBOL のみならず ASSEMBLY, Programming Language I (PL/I 言語) などへの発展性をもっており、また、図形処理の一つのアプリケーションとして評価できるとともに、X-Yプロッタなどの図形処理装置应用の一つの方向を見いだしたと考えている。

最後に、“HIFLOW/C”は HITAC ユーザー研究会における第11回大会(昭和49年6月)での発表を期に、日立工場のみならず社内外においても、コンピュータドキュメンテーションの有力な Tool として好評を得、広く利用されることを期待するとともに今更ながらこの種の Tool が欠如していたこと、及びカスタマーベースでの開発の必要性を痛感した次第である。

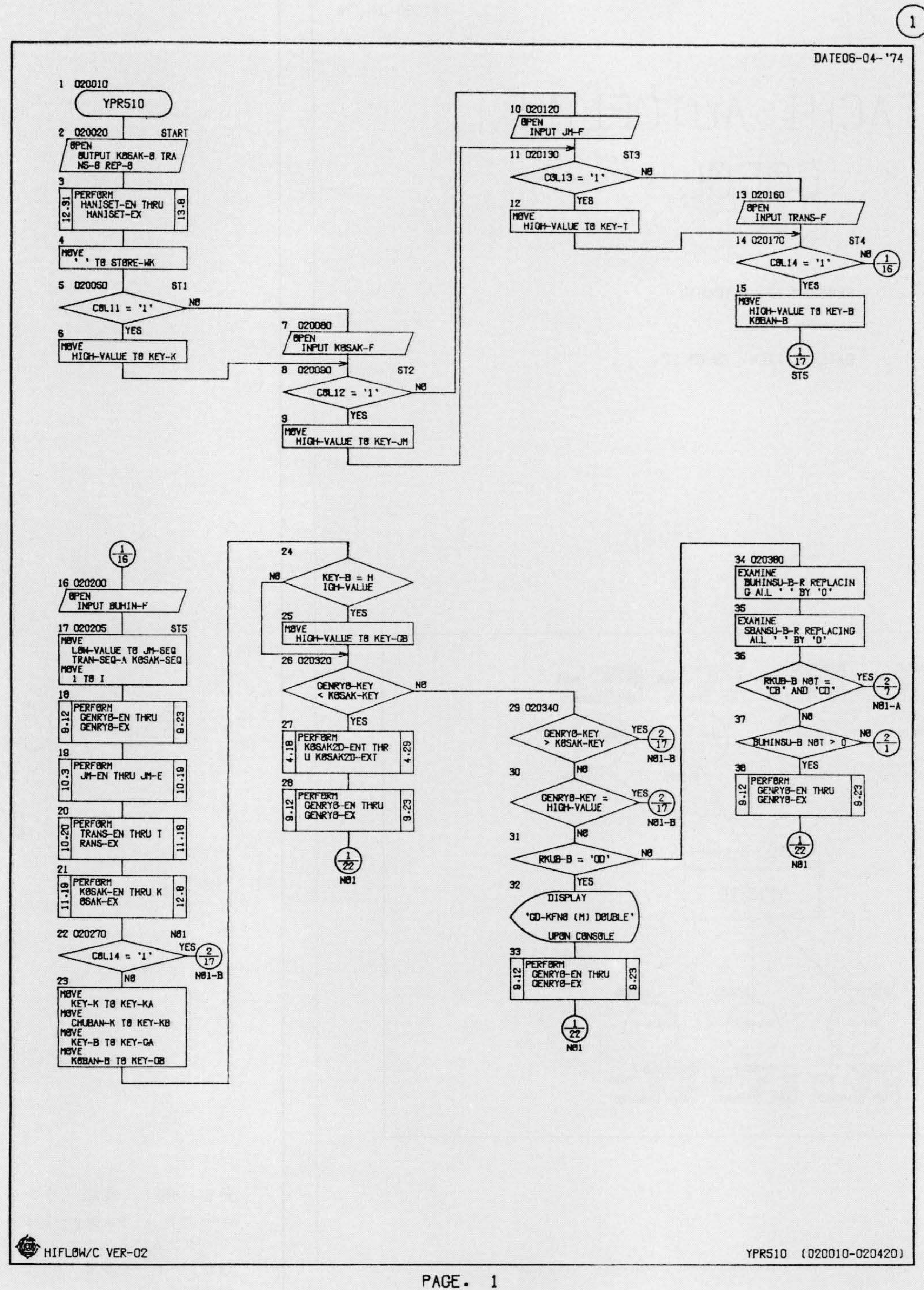


図7 例2：ディテールフローチャート フローチャートは、A 3サイズ縦長、横方向4列方式。本例は特長(4)の2段書きの例である。

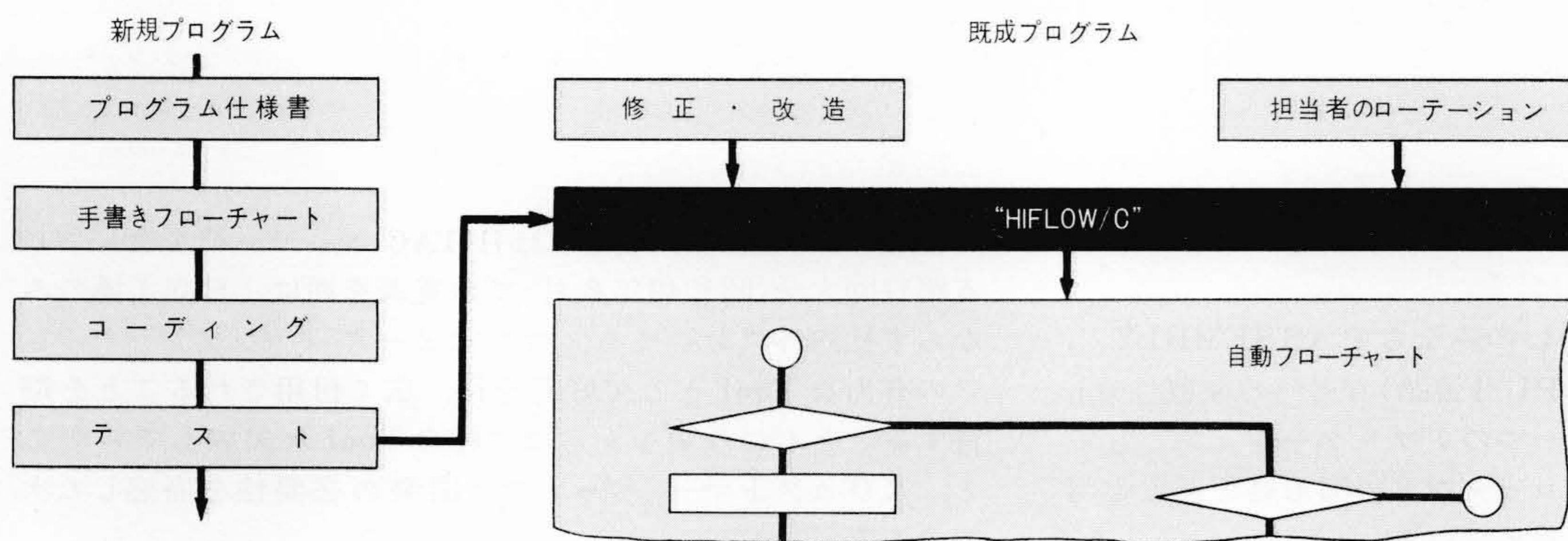


図8 "HIFLOW/C" の運用 既存プログラムにおいてプログラムの修正・改造を行なった場合、若しくは担当者のローテーション対策に"HIFLOW/C"を使用する。