

電子交換機実装割付けチェック・システム

Block Assignment Check System for Electronic Switching Equipment

菅 野 文 友* 萬 木 敏 昭**
Ayatomo Kanno Toshiaki Manki

要 旨

DEX-2 電子交換機の開発にあたり、電子計算機を利用し、設計の誤りを早期に発見するための設計自動化システムの一つとして、実装設計の誤りをチェックする割付けチェック・システムを開発した。

本システムでは、論理設計によって作成した回路を、人手によって IC やパッケージに割り付けした結果のチェックを目的として自動化を図ったものであり、このチェックのために、論理シミュレータ LSS-6 によって正しいことが確認済みの論理データ・ファイルを利用した。

また、論理データ・ファイルの素子間の接続関係を使用して、パッケージ・ジャック・ピンの割付けを自動的に決定した。

本報告は、この実装設計のための割付けチェック・システムについて、方式および処理内容についての概略を記述したものである。

1. 緒 言

現在、日本電信電話公社電気通信研究所を中心に、日本電気株式会社、沖電気工業株式会社、富士通株式会社、および株式会社日立製作所の交換機メーカー 4 社の共同研究として、室内実験用 DEX-1 電子交換機、現場試験用 DEX-2 電子交換機が開発されてきた。

これらの電子交換機および電子計算機などの複雑な情報処理装置はますます高性能、大規模化が要求されている。従来、これらの装置の設計、製造は、人手により行なわれてきたが、装置が巨大化、複雑化するにしたがって、設計作業で取り扱う情報量は膨大なものとなり、人手による設計作業が限界に達している。

このため、装置の設計の過程を電子計算機によって自動的に処理する技術が各方面で開発されている⁽¹⁾。たとえば、論理設計、回路設計、印刷配線パターン設計、回路パッケージ実装配置設計、布線設計が考えられる。これらの設計の過程を電子計算機を利用して処理することを設計の自動化 (Design Automation) と呼ばれる。

現在、IC の設計から計算機システムの動作解析にまで及ぶ各種シミュレータや、設計作業で必要とする大量のデータを処理し、作表、作図などを行なって設計能率の向上を図るなど多種多様な自動化システムが多方面で開発されている。

日本電信電話公社電気通信研究所においても、電子交換機 DEX の設計に電子計算機を利用する設計の自動化システムの開発が進められてきた。この種のものとしては、すでに論理設計結果の機能確認と論理デバッグを目的とした論理シミュレータ LSS-6 が開発され、DEX-1 CC (中央制御装置) の論理設計においてすでに活用されている⁽²⁾⁽³⁾。

DEX-2 電子交換機の開発にあたり、電気通信研究所を中心として、交換機メーカー 4 社が協力し、HITAC 5020 を使用した実装設計自動化のための割付けチェック・システム (BACS: Block Assignment Check System)、および布線設計システム (WACS: Wiring Assignment and Check System) が開発された⁽⁴⁾⁽⁵⁾。

本報告では、このうち電気通信研究所の指導のもとに日立製作所が担当した BACS についてその概要を記述する。

2. 設計自動化システムの概要

DEX-2 CC 開発に際し、適用した電子計算機による設計自動化シ

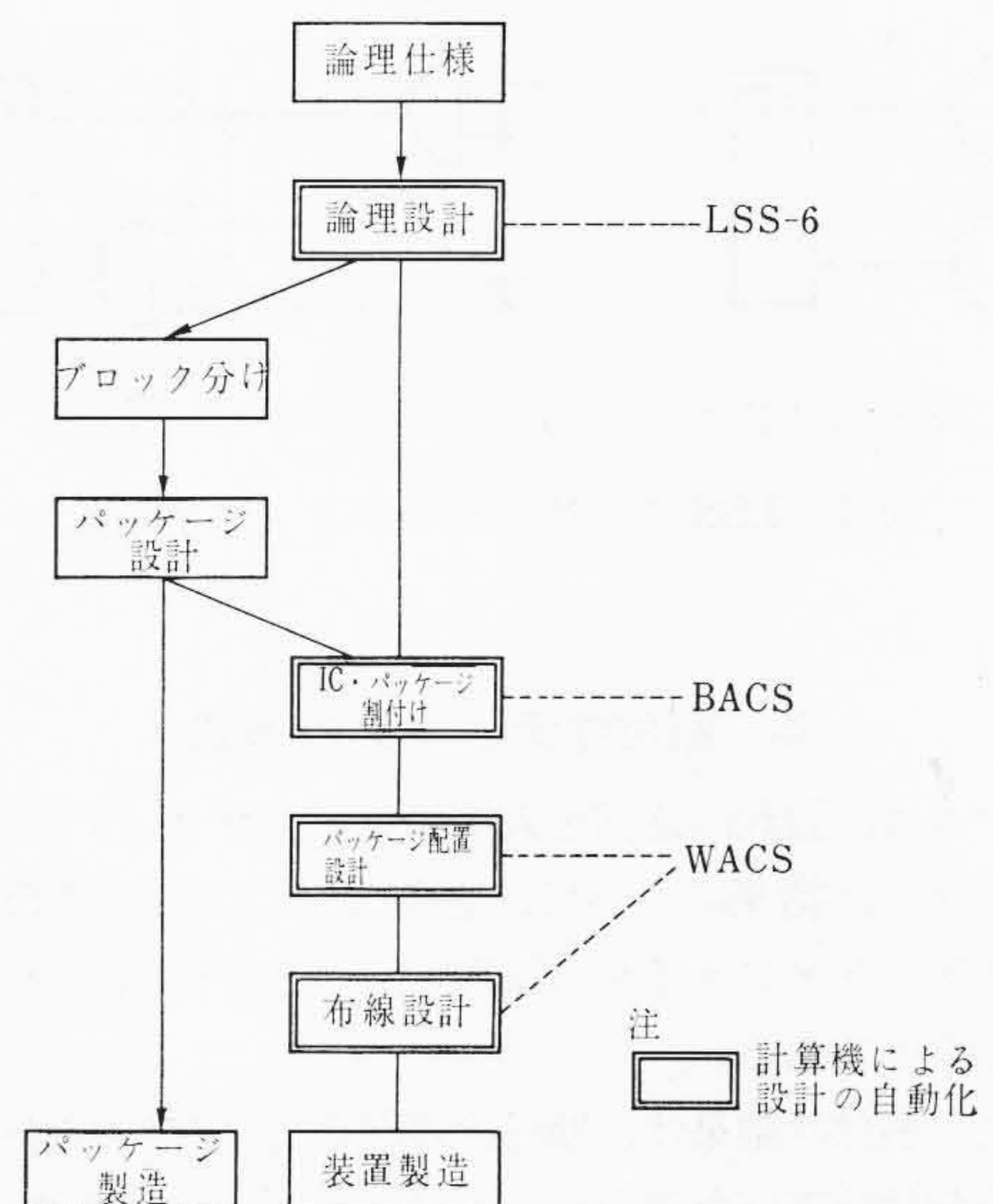


図1 設計自動化システムの概略図

ステムと、設計の流れの関係は図1のようになっている。

最初に、論理仕様ができ上がると、論理設計者が論理回路図を作成し、その結果を論理素子対応の論理カードにパンチする。これを入力として、論理シミュレータ LSS-6 によりシミュレーションが実行され、論理機能の確認および論理デバッグが進められる。ブロック分けは、論理回路を機能分類して分割し、パッケージ 1 枚にのせる回路を決める作業で、パッケージ設計とともに、人手により行なわれる。パッケージ設計は、ブロック分けされた論理回路を IC またはトランジスタなどの部品に割り付け、1 枚のパッケージに積載し、パッケージ上の部品の配置を決定する。IC・パッケージ割付けは、論理設計により設計された論理素子を先にパッケージ設計で完成したパッケージ、およびパッケージ内の部品に割り付ける作業であり、本報告の BACS によって部品の二重割付け、部品間接続状態などをチェックする。その後、人手により、パッケージの架段列配置が決定され、WACS により割付けもれ、二重割付けおよびコネクタ端子割付けなどの状態をチェックし、さらに布線接続順序、布線表、布線種が決定される。

このように、LSS-6、BACS、WACS によって、論理設計から布線設計に至る設計自動化システムを構成している。

* 日立製作所ソフトウェア工場 工学博士

** 日立製作所ソフトウェア工場

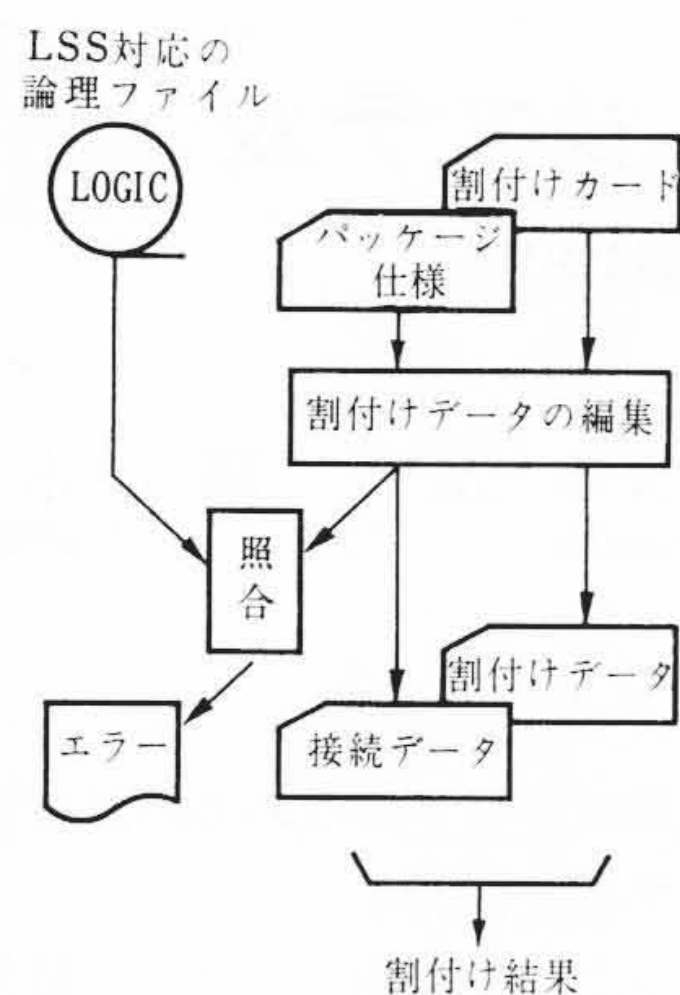


図2 割付けデータのチェック方式

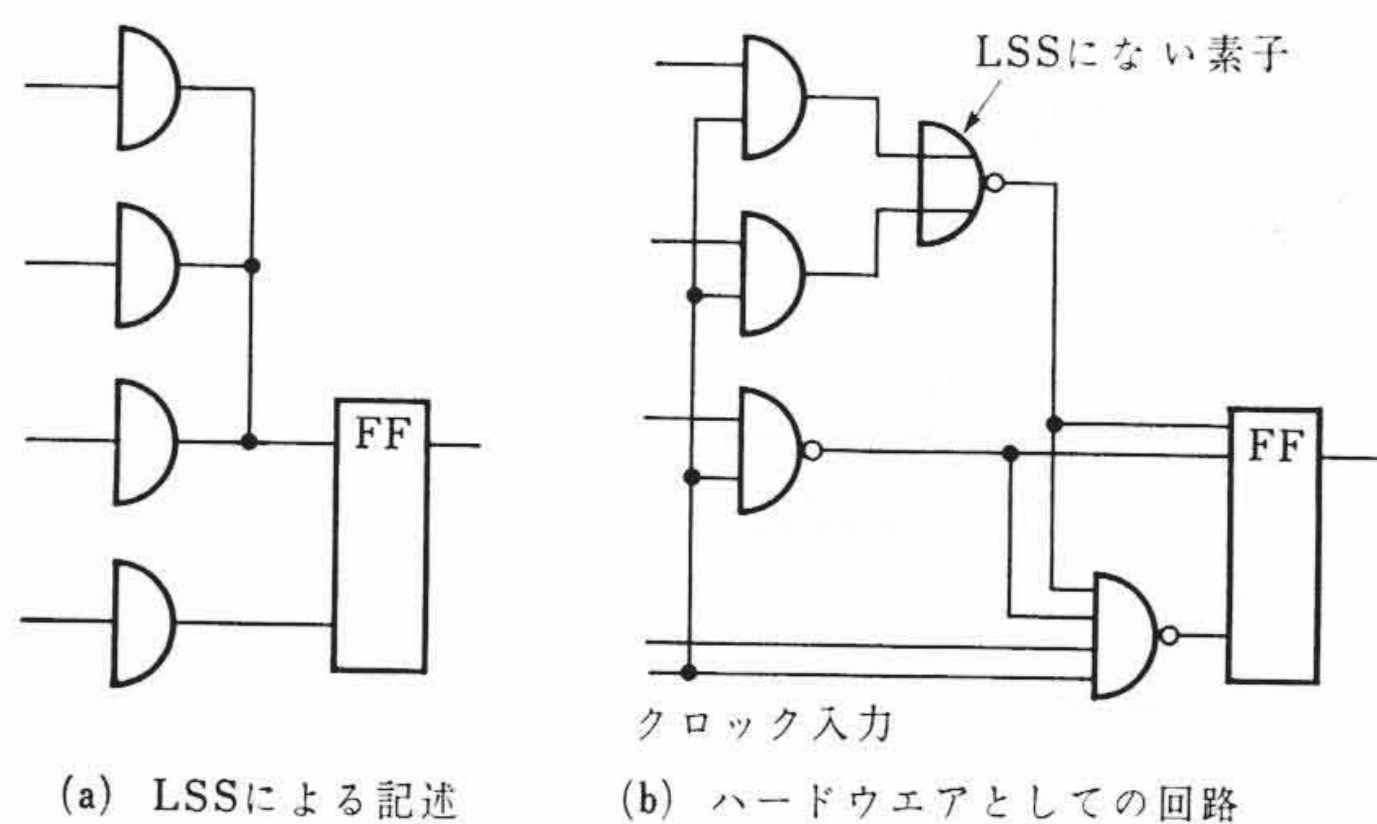


図3 LSSの記述とハードウェアの差異

3. 割付けチェックの方式

本システムは、論理回路図を人手によってブロック分け、および割付けを行なった結果について、論理シミュレータLSS-6で作成された論理データ・ファイル（論理ファイルと略称）を利用してチェックするものである。

人手による割付け結果は、割付け規則違反すなわち同一素子の二重割付けや割付け不能な部分への割付けなど、割付け結果だけでチェックできる部分と、論理的な誤りを生ずるような割付けもれや割付けの余剰など論理回路と照合しなければ誤りが発見できない部分とがある。後者のチェックを論理ファイルを利用してチェックを行なうものである。

割付けデータのチェック方式は図2に示す照合方式であり、割付けデータ側で論理的な接続情報も含めたすべての情報をデータとして作成し、論理ファイルと照合する。

この場合、割付けデータは、ハードウェアに対応した素子によって記述されるが、論理ファイルでは論理機能に対応した素子で記述される。このため、実際には図3に示すフリップ・フロップの場合やエキスパンダ、共通リセット・ゲートなどでハードウェア対応の素子とLSS素子が単純に1対1の対応となっていない。

したがって、このままでは、照合する素子の単位が一致しないため、照合する単位を一致させる必要がある。この照合方法として、幾つかの方法が考えられるが、本システムでは、論理素子の機械的な集合、たとえばICのようなブロックを考え、ハードウェア素子とLSS素子の両者をこのブロックに変換して照合する方法を採用した。この方法は、現在の論理設計が、ICやLSIの導入によって、従来のNAND単位の設計からブロック単位の設計へ変わりつつあることにより考えられたものである。

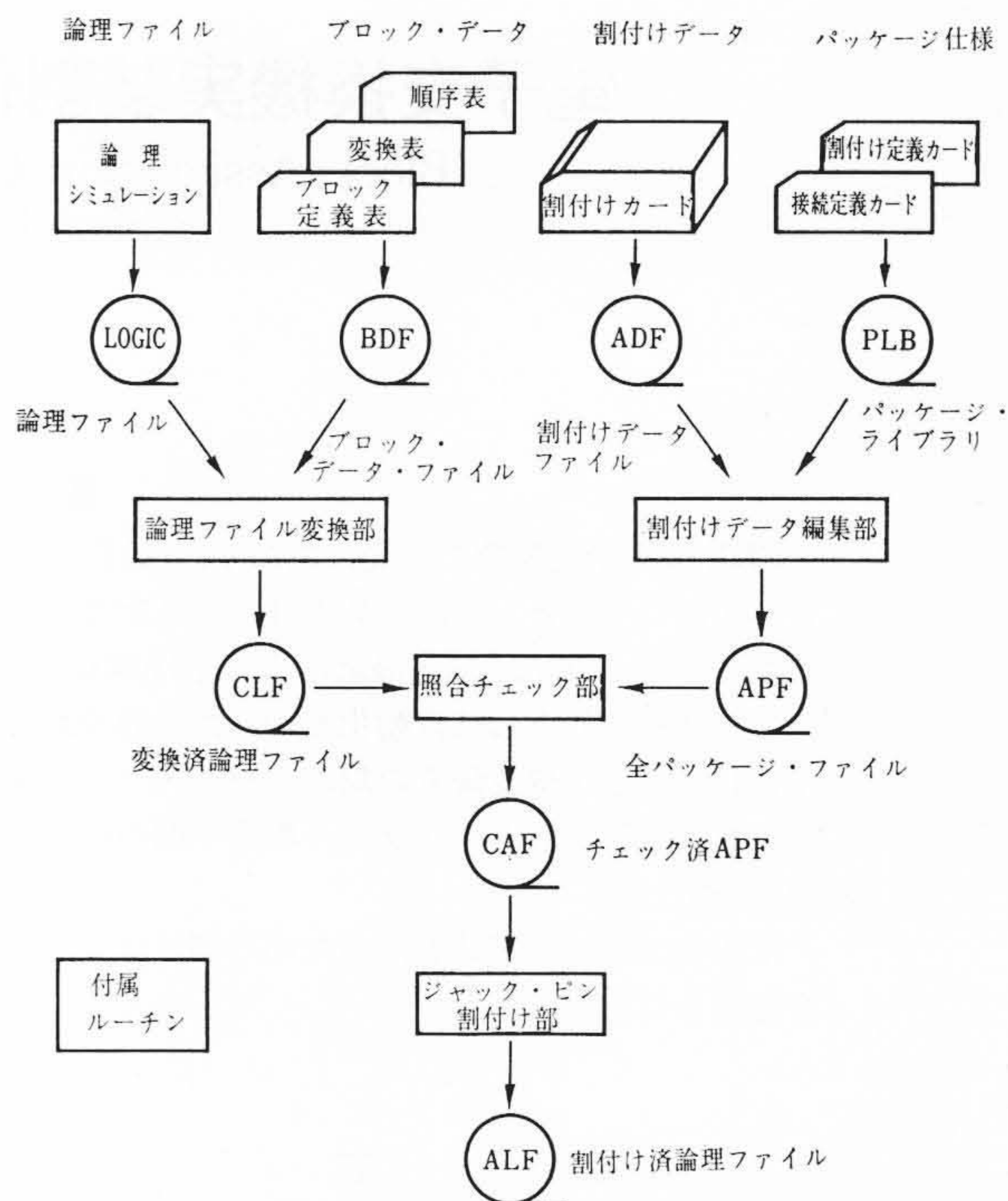


図4 割付けチェック・システムの処理概要

4. 処理方法

ブロック変換方式による割付けチェック・システムの処理の概要は図4に示すとおり、大別して次の5部分から構成される。

- (1) 論理ファイルをブロック・データによって割付けデータと照合できる形のファイルに変換する部分
- (2) 割付けデータをパッケージ定義データと併合して素子間の接続情報を与える編集部分
- (3) 上記により作成された二つのファイルからブロック単位ごとに照合チェックする部分
- (4) パッケージ・コネクタのジャック・ピンを割り付ける部分
- (5) 付属ルーチン

4.1 論理ファイル変換部

論理ファイルの内容を素子の論理機能名をキイとして、変換表を利用してブロックへ変換する部分であるが、1素子に割付け可能なブロックは一般に1通りとは限らない。たとえば、2入力NANDを割り付ける場合を考えてみると、対応するブロックは2入力NANDはもちろん、3入力NANDでも、4入力NANDでも、余分の入力を論理定数“1”すなわち開放端子にしておくことによって、金物的には割付け可能である。

したがって、論理ファイルを変換する場合、割付け可能なすべてのブロックへの変換を考慮しなければならない。このため、これらのNANDブロックに注目すると、2入力NANDは3入力または4入力NANDでも割付け可能であるが、3入力NANDは2入力NANDでは割り付けられず、4入力NANDで割り付けできるという入力数による包含関係がある。この包含関係にしたがって、変換可能なブロックを順序表として記述し、ブロックへの変換は割付け可能なブロックのうち、最小のものへ1対1に変換表によって行なう。さらに、NAND以外の複雑なブロックについても、入力数や入力側素子数を手がかりとして、包含関係を調べ順序表を作成する。

本システムでDEX-2のために用意したブロックは、NAND系、AND-NOR系、フリップ・フロップ系、および強制指定ブロックの4種類である。初めの3種類は、自動変換を行なうためのものであり、この3種類以外のブロックを定義して使いたい場合は、人手

によって強制的に指定したブロックへ変換することができる。

(1) NAND系

NAND素子など、主として1素子1ブロックになるものであって、端子やケーブルドライバなどの個別部品による素子を表わす場合もある。この場合、変換は論理機能名と入力数をキイとして行なわれる。

(2) AND-NOR系

AND素子とNOR素子を組み合わせたICのためのブロックであって、変換は、機能名がNORである素子があった場合、その入力側に接続するAND素子を捜し出して、1ブロック分の素子を集めることによって行なわれる。

なお、この場合の順序表は、入力側AND素子の数とこれらのAND素子への入力数をキイとして作成される。

(3) フリップ・フロップ系

論理ファイルではフリップ・フロップの記述が、入力ゲートのみであって、金物対応ではない。このゲートだけが記述されたものをフリップ・フロップとして変換するブロックである。

この場合の順序表は、セット側およびリセット側それぞれのゲート数と、これらゲートへの入力数をキイとして作成される。

(4) 強制指定ブロック系

上記(1)～(3)に属さないものすべてを強制指定ブロックとする。これは、NAND素子ばかりで構成するパリティ・チェック回路がこれに該当し、これが1個のICとして作られている場合、これらのNAND素子の集まりを一つのブロックとして人手により強制的に変換する。このようなブロックの場合、論理ファイルを見ただけでは、パリティ・チェック回路であるというパターン認識を行なわない限り、ほかのNANDブロックと区別できない。現状では、パターン認識による自動変換は、処理が複雑となる割には利点も少なく、人手による指定で補うことにした。

このように(1)～(4)の方法で変換した結果は、論理ファイルの各素子に変換したブロック名を追加したものであり、変換済論理ファイル(略称CLF)と呼ぶ。

4.2 割付けデータ編集部

人手によって割付けを行なった結果から、AND、NANDなどの最小論理単位ごとに割付けカードを作り、これに素子名称と実装位置の対応関係を記入して、入力データとする。

この割付けデータ編集部の処理は、割付けカードに記入した割付けデータを、別に用意してあるパッケージ仕様を記入したパッケージ・ライブラリ(略称PLB)と照合し、実装位置をキイとして、上のせ方式によって素子名をパッケージ・ライブラリにはめこんでゆく作業である。この処理によって、割付けデータだけでわかる誤りと、パッケージ仕様との照合によってわかる誤りはすべて検出される。

この処理の結果は、実装されるパッケージすべてについての割付けデータと、パッケージ内部での素子間の論理的な接続情報を含んでいる。これを全パッケージ・ファイル(略称APF)と呼ぶ。

4.3 照合チェック部

本割付けチェック・システムの中心部であって、変換済論理ファイルに記録されている論理的に正しいデータと全パッケージ・ファイルに記録されている人手による割付け結果とを、照合チェックする。

照合チェックは次の手順で行なわれる。

(1) 変換済論理ファイルから、一つのブロックを取り出す。

(2) (1)のブロックの先頭の素子名(もっとも出力側に位置する素子)をキイとして、全パッケージ・ファイルから対応するブロックを取り出す。

(3) 両ブロックについて、ブロック単位、素子名称および機能などを照合し、対応するブロックであることをチェックする。もし、対応しなければ、順序表を参照して、対応するブロックを捜し出す。変換可能なブロックすべてを参照しても対応しなければ、割付けの誤りである。

(4) 論理ファイル側から取り出したブロックを、ブロック定義表を参照して、ハードウェア対応の最小論理単位に分解する。このときブロック定義表に記録されているICコードなど論理接続以外の情報も付加される。

(5) 最小論理単位のレベルで、論理ファイル側と割付けデータ側の情報を照合チェックする。チェックは接続情報が中心であるが、このほかにも、フリップ・フロップ用ゲートの図面位置、ICコードなど、使用者の要求に応じてチェック項目をふやしている。

(6) パッケージ間の接続情報は論理ファイル側にあるだけであり、この場合の照合チェックは行なわれない。この部分のチェックは対応する素子があることを確かめるのみで、論理ファイル側の接続情報をパッケージ間接続として、別のファイルに出力する。これを入力ジャック・ファイル(略称IJF)と呼び、次のジャック・ピン割付けで、パッケージ・コネクタ・ピン間の接続すなわち裏面布線を決めるデータとして利用する。

(7) 一つのブロックについてチェックが終わると、チェック済マークを付けて次のブロックへ進み、上記の(1)～(6)の処理を繰り返す。

ブロックの取り出しは、変換済論理ファイルに記録されている順にしたがう。

また、処理の途中でチェック・エラーになった場合は、そのブロックのチェックをすべて中止し、エラー・データを保存して、次のブロックへ進む。

(8) すべてのブロックについてチェックが終わると、全パッケージ・ファイルの内容を最小論理単位ごとに取りまとめて、チェック済全パッケージ・ファイル(略称CAF)に出力する。

(9) チェック中のエラー・データは、すべての処理が終わったのち、エラーの種類、程度、内容などによって編集し、リストとして出力する。

なお、特別な条件として、最小論理単位で接続の照合チェックを行なう場合、NAND素子の入力については、割付けデータ側の入力線に余分なものがあっても、それがパッケージ・コネクタに接続されているものであれば、チェック・エラーとはしない。これは、この端子を開放し、空(あ)きの端子として処理すれば、論理定数1が接続されているものとみなせるためである。この空き端子もチェック済全パッケージ・ファイルに記録しておき、次のジャック・ピン割付けで利用する。

4.4 ジャック・ピン割付け部

照合チェックでは、同一パッケージ内の接続は、完全に処理されるが、パッケージ・コネクタのジャック・ピンからジャック・ピンへの接続、すなわち裏面布線の部分については、素子の対応がチェックされるだけであって、どのジャック・ピンを使用するか決定していない。たとえば、図5(a)に示す回路を図5(b)に示すパッケージに割り付けた場合、素子Xからの信号が素子Aに接続されるにあたって、ジャック・ピンJ2を通るかJ3を通るか決定されていない。これを自動的に決定する処理がジャック・ピン割付けである。

図5(a)および(b)の例で、接続方法としては2個の解があり、J1-J2, J3=空き, J4=空きの場合と、J1-J3-J4, J2=空きの場合とがあるが、論理的には同じである。しかし、裏面布線をできるだけ少なくするためには、パッケージ内マルチ接続を利用した前者

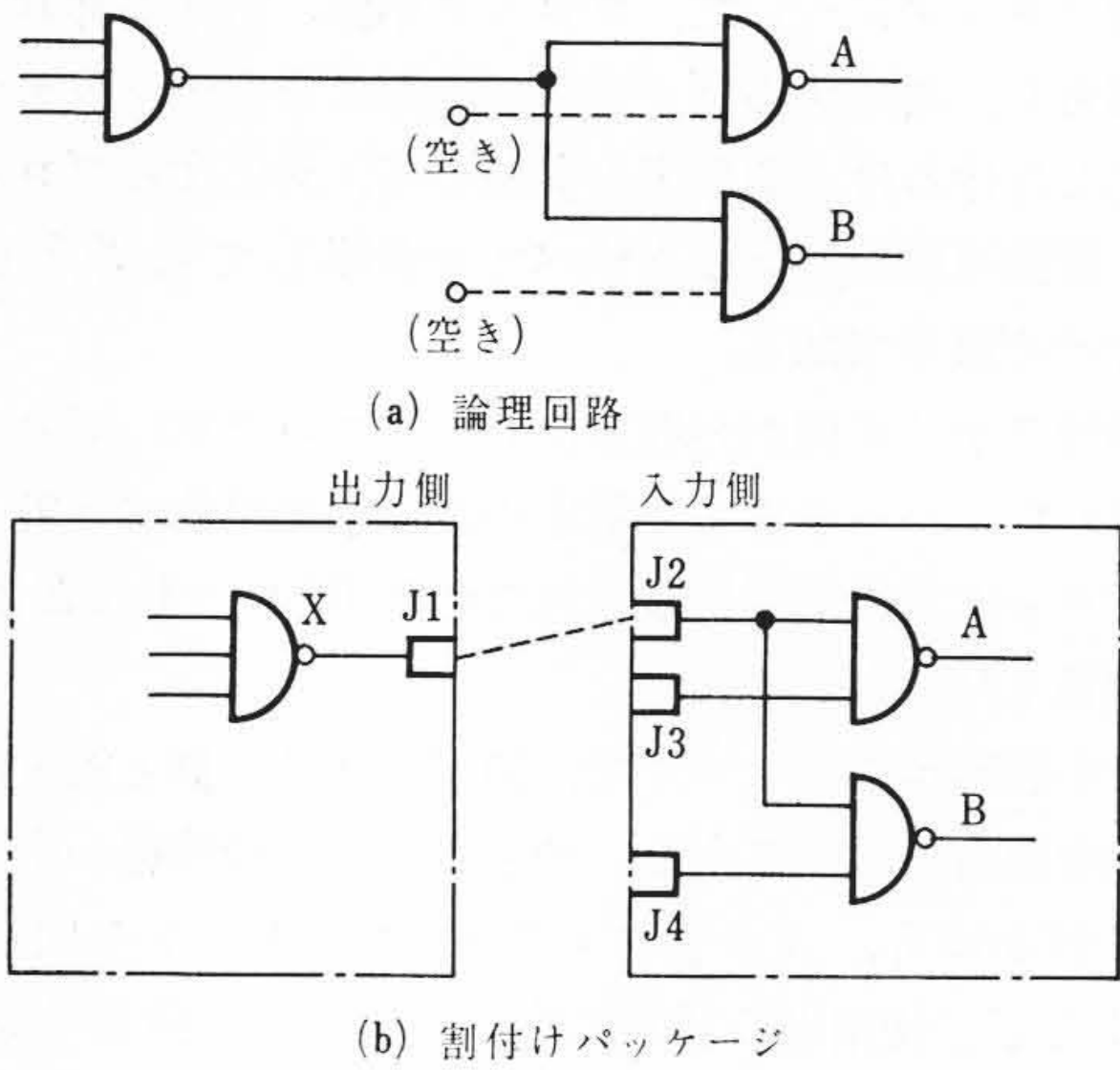


図 5 ジャック・ピン割付けの例

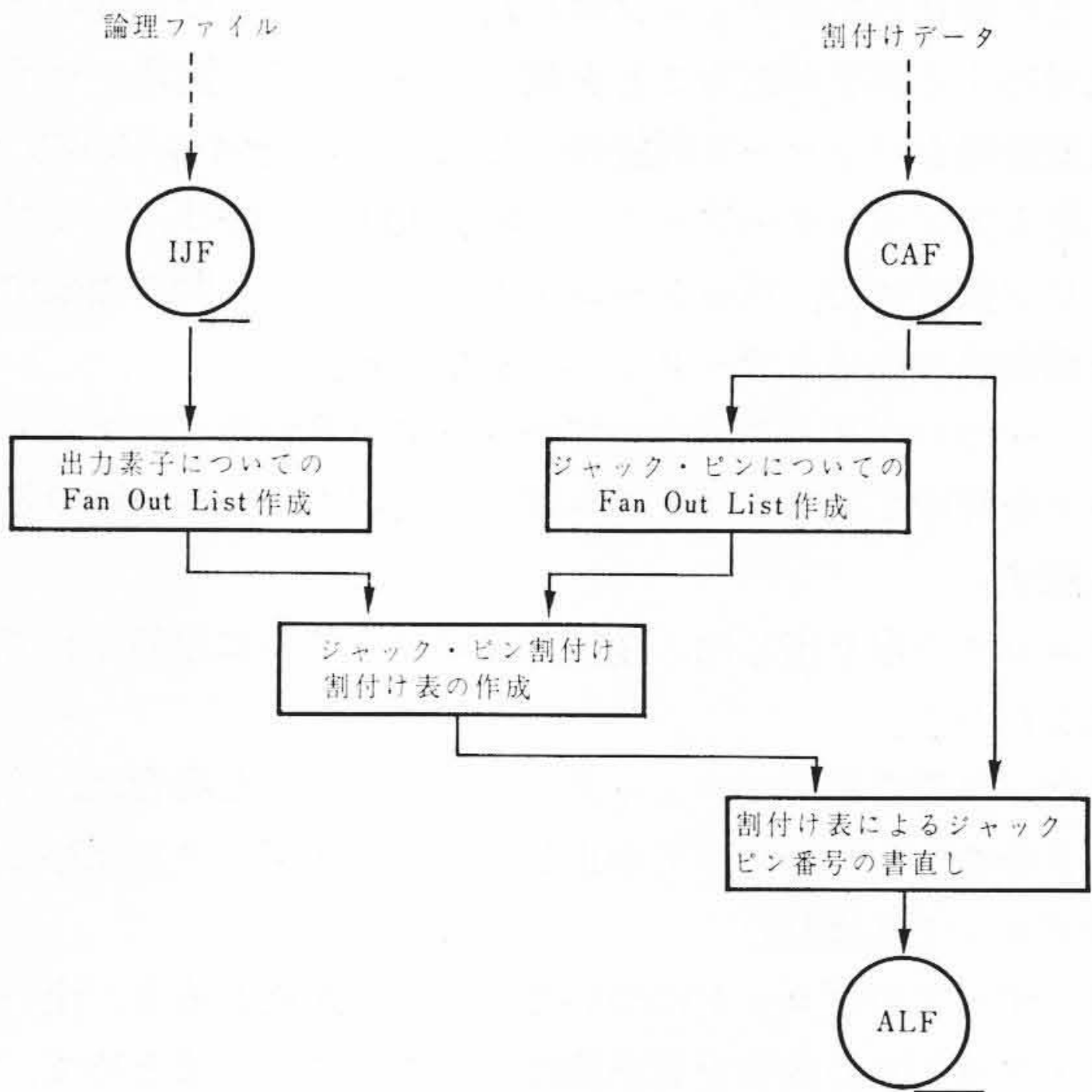


図 6 ジャック・ピンの割付けの処理概要

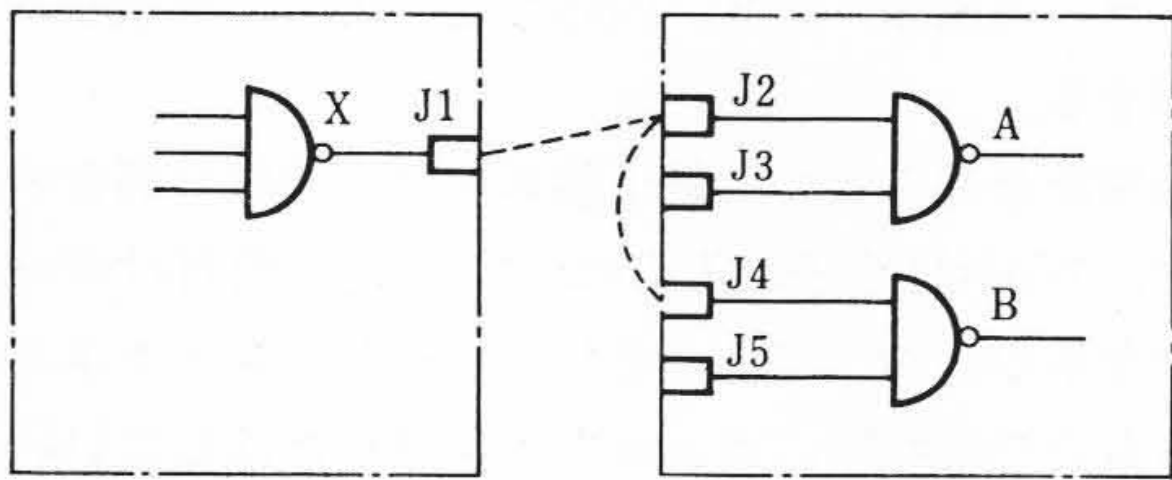


図 7 パッケージ外マルチ布線の例

の解、すなわち図 5 (b) の点線の接続がよい。

このような解を求めるにあたって、出力素子と出力ジャック・ピンは 1 対 1 に対応するというパッケージ設計上の制限を利用し、処理している。

なお、ここで用語の定義をすると、出力素子とは、パッケージから外へ信号を出力する素子であり、図 5 (b) の場合、信号は、出力素子 X から出力ジャック・ピン J1 を通り、裏面布線によって入力ジャック・ピン J2 にいたり、J2 から入力素子 A および B に入力することになる。

ジャック・ピン割付けは次の手順によって行なわれる。

(1) 入力ジャック・ピンと入力素子の接続関係はパッケージ仕様から与えられ、照合チェックの結果として、全パッケージ・ファイルに記入してある。これはすべての入力ジャック・ピンを含んでおり、図 5 (b) の右側の部分が該当する。

表 1 入力ジャック・ピンについての FAN OUT LIST	
入力ジャック・ピン	FAN OUT 素子名
J 2	A, B
J 3	A
J 4	B

表 2 出力素子についての FAN OUT LIST	
出力素子名	FAN OUT 素子名
X	A, B
空 き	A
空 き	B

表 3 割 付 け 表	
出力素子名	入力ジャック・ピン
X	J 2

これを入力ジャック・ピンごとに取りまとめて、入力ジャック・ピンの FAN OUT LIST (展開表) を作成する。例の場合について作成すると表 1 となる。

(2) 素子間の論理的接続関係は、照合チェックの結果として入力ジャック・ファイルに記入してある。これを出力素子ごとにとりまとめて、出力素子についての FAN OUT LIST を作成する。例の場合について作成すると表 2 となる。

(3) 上記で作成した FAN OUT LIST を FAN OUT (展開素子数) でソートする。この処理は、次の照合を容易にするとともに、パッケージ内マルチ布線を多く利用するために役だつ。

(4) これらの FAN OUT LIST について、FAN OUT の素子数と素子名称の照合を行ない、一致するものを捜す。一致した場合、これに対応する入力ジャック・ピンと出力素子が接続されることになる。上述の例では、表 1 の 1 行目と表 2 の 1 行目の FAN OUT が一致するので、X の出力を J2 に割り付けることになる。

(5) ここで求めた出力素子と入力ジャック・ピンの対応をジャック・ピン割付け表に記入して、一つの出力素子についての割付けを終わる。上述の例では表 3 となる。

(6) 上記(3)、(4)の処理を繰り返し、すべての出力素子についての割付けを行なう。

(7) 割付け終了後、ジャック・ピン割付け表を使って、チェック済全パッケージ・ファイルのジャック・ピン番号欄にピン番号を記入し、割付け済論理ファイル(略称 ALF)を作成する。なお、本システムの最終出力となる ALF は必要に応じてソートしたものである。上記の処理概要を示したのが図 6 である。

このようにして、ジャック・ピンの割付けを行なうが、場合により、上記(3)において、FAN OUT の一致が得られないことがある。第 1 の原因は、入力ジャック・ピンに接続している複数の入力素子があって、そのうちのいくつかが論理設計上使われていない場合、たとえば、図 5 の例で素子 B が使われていない場合である。このときは、入力ジャック・ピンについての FAN OUT が出力素子についての FAN OUT を包含しており、しかもその差の素子が論理設計上使われていないことを確かめれば割付けてよい。

第 2 の原因は、FAN OUT がパッケージ内のマルチ接続だけでは実現できず、パッケージ外マルチ布線が必要な場合である(図 7 参照)。この場合は、できるだけパッケージ内のマルチ接続を利用し、やむをえぬ部分だけパッケージ外の布線とし、入力ジャック・

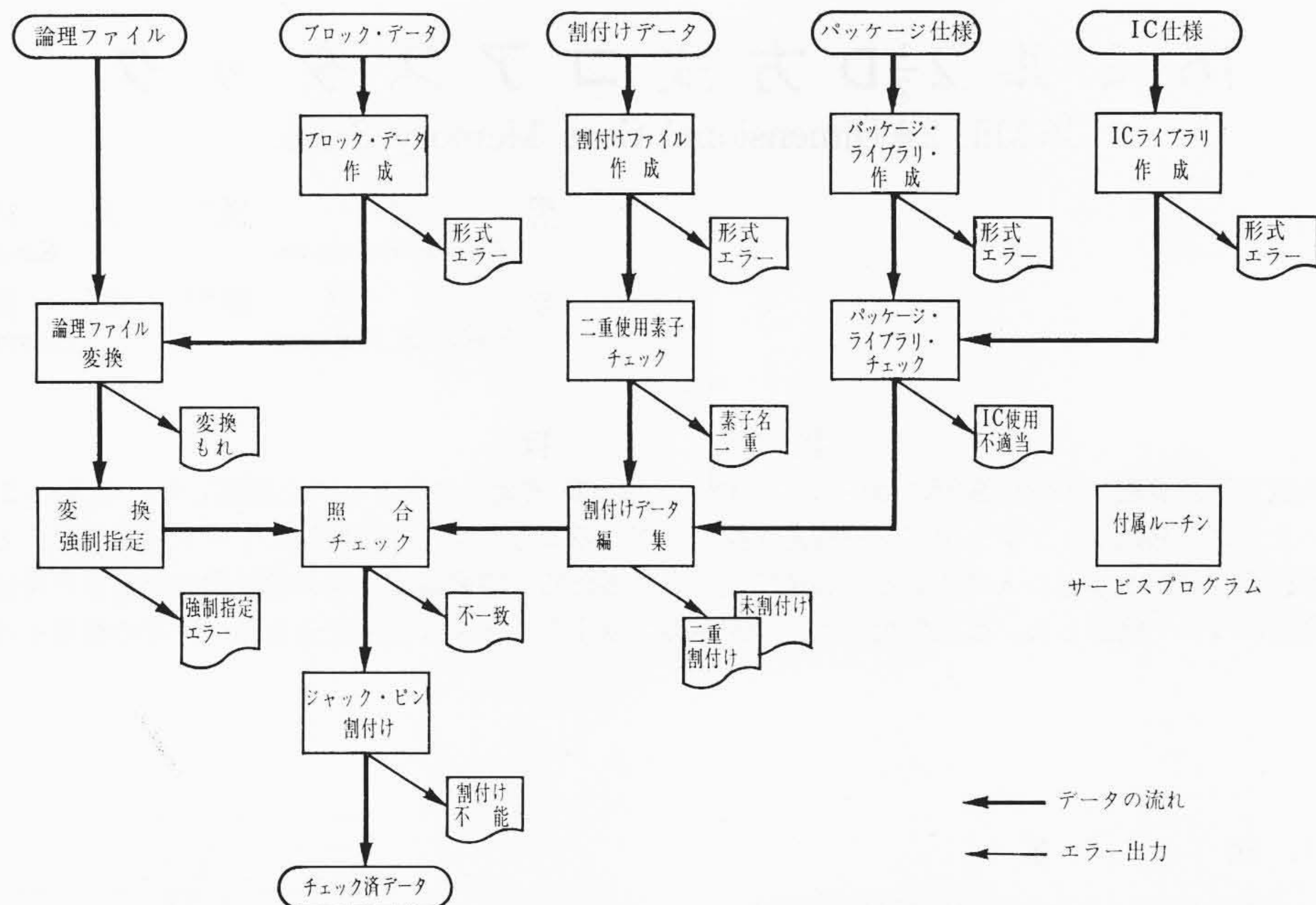


図8 プログラム構成とエラー・リスト

ピンについての FAN OUT の多いものから取り、残り部分を布線で補い、割付けを行なう。

したがって、実際のジャック・ピン割付けでは、FAN OUT の完全に一致するもの、論理設計上使用しない素子を含むもの、およびパッケージ内接続とパッケージ外布線との組合せによるものの3段階に分けて行なう。

4.5 付属ルーチン

本システムで使用する各種の磁気テープ・ファイル用のサービス・プログラムであって、ファイル内容の編集、プリント、訂正を行なうための処理とほかの設計自動化システムとデータの交換を行なうための処理がある。

5. プログラム構成

本システムは、人手による割付けデータをチェックし、これを整理編集して、ほかの設計自動化システムへ与えることを目的としたもので、それにしたがってプログラム構成を定めた。前述のとおり、割付けチェックの処理は、割付けデータのみでチェックできる部分、パッケージ仕様と照合してチェックできる部分、および論理ファイルと照合しなければチェックできない部分、さらに使用する計算機のメモリ容量からくる制約などから、それぞれ別個のプログラムとして全体の構成が決定されている。

プログラム構成をチェックの内容とデータの流れに注目して示すと図8のようになる。

6. プログラム規模

本システムは、電子計算機 HITAC 5020 を使用し、アセンブラ言語である新 HISAP によってプログラム化されている。

本システムで使用する HITAC 5020 の最小機器構成は表4に示すとおりである。しかし、テープ・ソートの処理時間短縮およびラインプリンタによるリスティングのマルチ処理のため、実際には磁気テープ10台、ラインプリンタ2台が使用される。

また、本システムのプログラム・ステップ数は新 HISAP の命令ステップ数で表わすと、

- (1) 変換表作成および論理ファイル変換部 4.6 K ステップ

表4 BACS の最小機器構成

項番	機器装置名	台数
1	H-5020 形 処理装置 65K 語	1
2	H-197 形 磁気ドラム記憶装置	8
3	H-3485 形 磁気テープ装置	8
4	H-329 形 カード読取り機	1
5	H-334 形 カードせん孔機	1
6	H-333 形 ラインプリンタ	1

- (2) パッケージ仕様ファイル作成部 4.8 K ステップ
(3) 割付けデータ編集部 3.2 K ステップ
(4) 照合チェック部 4.9 K ステップ
(5) ジャック・ピン割付け部 1.9 K ステップ

となり、これに付属ルーチンを加えると全体では 26 K ステップに達する。

7. 結 言

実装設計自動化の第一歩として開発したソフトウェア・システムの一つである割付けチェック・システムは、DEX-2 電子交換機中央制御系のいくつかの論理装置の設計に利用され、大量のデータを短期間に処理して設計の誤りをなくすために役だった。

特に、NAND 換算で約 21,000 素子の規模の DEX-2 CC (中央制御装置) の論理装置に適用され、運用されたことにより所期の目的が達せられた。

最後に本システム開発にあたり、技術的なご指導をいただいた日本電信電話公社電気通信研究所岸上データ通信研究部調査役、山本情報処理プログラム研究室長、もと電子制御研究室佐座研究専門調査員、技術局倉地調査員に深く感謝する。

また、本システムの仕様検討において、ご協力いただいた日本電気株式会社、富士通株式会社の関係各位にお礼申し上げます。

参 考 文 献

- (1) 高島：信学誌，50，p.590 (1967-4)
(2) 高島ほか：情報処理，7，p.263 (1966-9)
(3) 飯田，若山：通研実報，17，No.1，p.135 (1968)
(4) 佐座，倉地ほか：交換研究会資料，SE 68-6 (1968-5)
(5) 佐座，中村ほか：交換研究会資料，SE 68-7 (1968-5)