

ICを単位としたパッケージ設計データ入力システム

手本雅則 小花信雄 安野威志

(日本電気株式会社)

1. はじめに

電子交換機、電子計算機システムの多様化にともない、これら論理装置の設計量は増加し、DA(設計自動化)システムへの期待も、はじめの“個々の設計作業の自動化・省カ化”といったイメージから、よりトータルな効果を指向するようになってきている。これにしたい、設計工程全般にわたる能率向上の意味から、DAシステムの機能、性能、拡張性(あるいは汎用性)といった面のほかに、信頼性や使いやすさの面からの評価がなされるようになってきた。

通常の装置設計では、DAシステムの使用に際し、多量の入カデータを必要としてきた。このうちの何割かは、本質的に装置設計に必要な情報であろうが、のこりは、DAシステムの事情により、入カを余儀なくさせられているものである。

また、ひとたび入カしたデータを、ふたたび別の形式で入カさせられるといった、DAシステム使用上不都合なことも生じている。これらは、すべてDAシステムの“使いにくさ”に通じ、とくに多種少量生産の装置においては、納期や設計工数の制限が強いことから、実施上の大きな障害となるものであった。

これらの問題に対し、データベース化¹⁾、設計データの言語化²⁾および回路図の直接入カ³⁾等いろいろの試みがなされてきた。

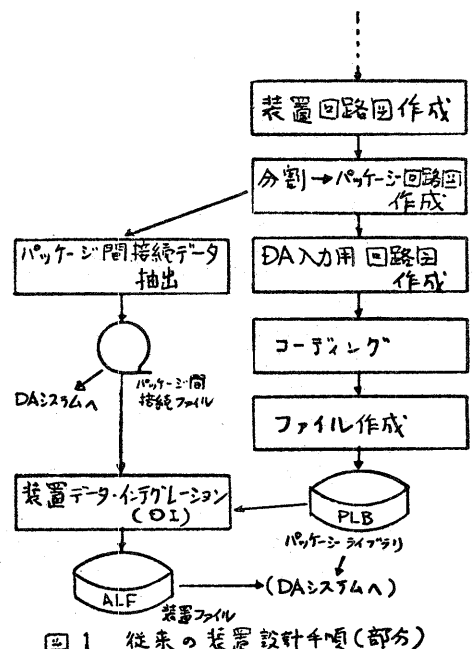
本報告では、DAシステムの使いやすさを、入カデータ量の減少、データ作成の容易化という面から考察し、IC間接続による入カと、そこにともなう図面発生を提案している。なお、本入カシステムは、昭和48年に実用化され、現在当社電子交換事業部で、DAシステムの一部として使われている。

2. 装置設計データ入カの問題点

2.1 入カ手順

当社で従来行われていた、装置設計データのDAシステムへの入カ手順の一部を、図1に示す。

装置回路図作成後、パッケージ分割が行われ、パッケージ単位の回路図が作成される。ついで、素子のICへの仮割付けが行われ、ゲートレベルのDA入カ用回路図が作成される。この回路図は、DAシステム中の回路図印刷プログラムで印刷されるものと1対1に対応し、重要なソースデータであった。ここから、素子毎のコーディングを行い、パッケージライブラリファイル(PLB)を作成する。いっぽう、パッケージ間接続データは、ファイル化された後、PLBとともに、装置ファイル(ALF)を作成する



ために用いられる。このパッケージデータを装置データに展開する処理は、装置DI (Data Integration) と呼ばれ、装置中に存在する、同種類複数のパッケージデータの重複作成をなくし、相互の矛盾を避けるために用いられている。

このようにしてできた3種のファイルは、データベースとして、各種DAプログラムにより処理が行われ、製造・検査データが作成される。

2.2 DA用入力回路図

ICの集積度の向上、その機能化、および装置実装の小型化等の理由から、パッケージの種類は増加する傾向にあり、パッケージ設計に対する能率向上が特に望まれていた。とくに、前節で述べたDA用入力回路図作成作業は、単純な作業ではあるが、多くの工数を要し、設計スケジュール上のネックとなり、また、誤りの混入しやすい箇所でもある。この回路図を書くためには、IC割付作業のほかに、

(1) 図面分割---パッケージ全体を、どのように複数の回路図に分けるか。

(2) 素子の図面位置の決定---見やすく、かつ(3)がうまくいくように。

(3) 図面上での素子間の接続

等の図面上の作業を必要とした。とくに、(2)において決められる図面位置は素子名として、コーディング時の接続表現に用いられるため、この回路図作成は、DA使用上必須とされていた。

さらに、この回路図は、ゲートレベルのため、IC内の素子や接続までも記載しなければならず、このために、同一種類ICが食い違った形で入力されたり、MSI等の使用による素子数の増加が作業を長びかせたりした。

2.3 IC

MSI、LSIなどの機能的ICが多く供給され、設計者は、設計段階のかなり初期から、使用する回路とそれに対応するICを、念頭において作業しているといえる。よって、IC割付のイメージは、全体ではないがかなりの部分DA入力以前に明確になっていると考えられる。

いっぽう、ICの内部に関するデータは、割付チェック等のため、DAミステム中に持っているのが普通で、設計者にとっては、仮割付にせよ、IC情報をPLB作成のために入力するのは、2度手間という感じがあった。

3. 入力システムの構成

3.1 目標

2.で述べたような状況のもとで検討を行い、DA用入力回路図を作成しないで、DA入力が可能な方法が検討された。このような試みは、以前にもなされたことが報告されており文献1)では、簡便な言語でゲートレベルの装置回路図を発生している。また文献2)は、実装の実現があらかじめ可能なように定義された機能ブロックという概念を階層的に用い、機能ブロック構成要素の相対図面位置を用いて、図面レイアウトを行っている。

われわれは、DA入力作業者が、単純作業で入力が作成でき、かつその量が少ないことを目標とした結果、

(1) ICのライブラリ(ILB)を用い、IC(端子)間接続をもととした、パッケージ設計データの展開(パッケージDI)。

(2) ゲートレベル素子の図面位置の決定

の2点を自動化するのが、作業能率を上げるうえで得策と考えた。

入力単位として、機能ブロックではなく、ICを選んだ理由として、

- 設計者がIC割付イメージを持っているため、これを尊重した。
- すでにICライブラリファイル(ILB)が存在し、この後、さらにブロックライブラリ等の管理対象が増加するのは好ましくない。
- 入力カード形式を簡単にでき、設計者以外の者がコーディングできる。また、ICに付随するデータは、そう多くないので、カード枚数を減らすことが期待できる。
- 集積度の向上によっても、入力データ量の増加がない。

等があったが、図面への割付では、ICより小さな単位の使用も許している。

さらに、具体的目標としては

- 1 IC 当り 1 ~ 2 枚のカードとすること。
- 作成作業管理上固定カラム形式のカードを用い、かつコーディングしやすいこと。
- 図面への割付は、ICより小さな単位でも行えること。
- 図面の割付状況をすぐ知ることができること。(回路図印刷はデータベースロード後可能)
- チェックが充分なされること。
- 処理時間が短いこと。

等が挙げられた。

3. 2 システム構成

本システム(以降PDI-Package Library Data Integration-システムと略す。)の存在により、ファイルの生成過程は、図. 2 のようになり、一切の割付チェックプログラムが不要となっている。

図. 3 に、本PDIシステムの構成を示す。IC間接続カード、図面指定カード、およびILBを入力とし、PLBロード用の中間ファイル(従来の入力カードイメージ)、チェックリスト、および図面割付リストを出力する。

これら出力リストの結果により、入力カードを修正し、PLBを作成する。

PLBからの回路図印刷の後、この入力カードを修正する頻度を少なくするため、空端子や未使用素子の処理、および、IC使用上のチェック等を行っており、ユーザーは、IC間の接続とその図面への割付けさえ注意していれば、DA入力の正当性が保てるようになっている。

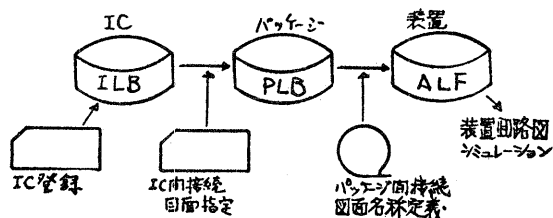


図2. ファイル生成過程

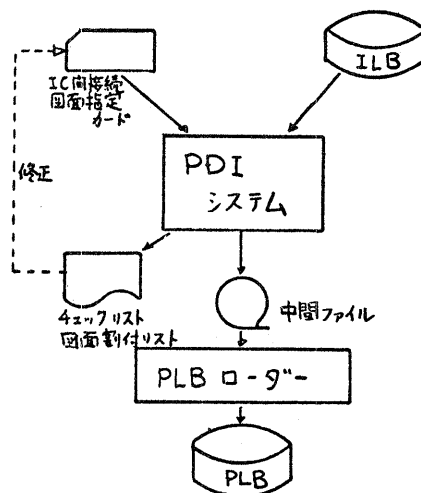


図3. システム構成

3.3 プログラム構成

図4に、本システムのプログラム構成を示す。パッケージ種類ごとに、入力処理、展開処理、図面位置割付処理、そして出力処理の順で実行される。以下に、各処理の内容を順を追って説明する。

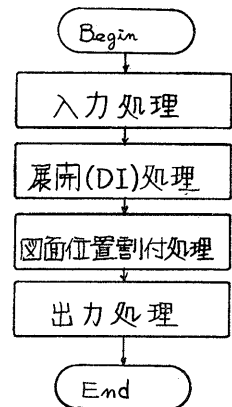


図4. プログラム構成

4. 入力データと入力処理

4.1 I L B (I C ライブラリ) とグループ

I C の組み合わせによって、パッケージ論理を構成するためには、その前提とする I C について、十分な情報を有するライブラリが必要となる。I L B は、I C についてのデータベースであり、I C 種類ごとの、構成要素、要素間および I C ピンとの接続等が含まれる。

現在、装置設計に使用されている I C の中には、同一の機能を有する回路が複数個含まれているものがある。(図5) これらの回路は、通常論理的には独立なものとして扱われ、もっぱら割付上の都合でその使用が決められるようなものである。

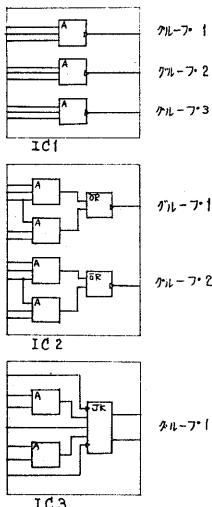


図5. グループ

ここでは、このおのおのの回路を“グループ”と呼び、I C としては、お互いに切りはなせないが、図面指定時に個別のものとして扱えるようにしている。

各グループを構成する要素には、図面上での相対的位置関係を示す図面座標が付与されている。

4.2 I C 間接続カード

パッケージ内の論理接続のタイプは、a) I C ピン相互、b) 入力パッケージピンから I C ピン、c) I C ピンから出力パッケージピンの3通りがある。(図6) これを、ファンイン形式で記述すると、a) と b) は I C ピンからみた入力先、c) は出力パッケージピンからみた入力先との接続、ということになり、2種類の形式を用いる必要がある。

(1) I C ピンからの接続

この接続を定義する場合、図7 に示したようなコーディングシートを用いる。上部には、この I C の素子レベルの回路図(相対的な図面位置も示している。)がプリントしてあり、下方に接続データを、入力ピン番号に対応して記入するようになってい。記入内容は、入力先 I C の名称(2文字、パッケージ内ユニーク)およびその I C ピン番号である。また、入力先がパッケージピンであるときは、パッケージピン表示文字“J J”とそのピン番号を記入する。この I C の出力ピン番号に対応する欄は、コーディングシートのデザイン時に、使用不可にしておく。この形式では、14ピン I C のとき、80カラムのカードで1枚、16ピン ~ 22ピン I C で2枚の入力カードを作成すればよい。

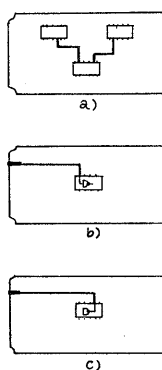


図6. IC間接続

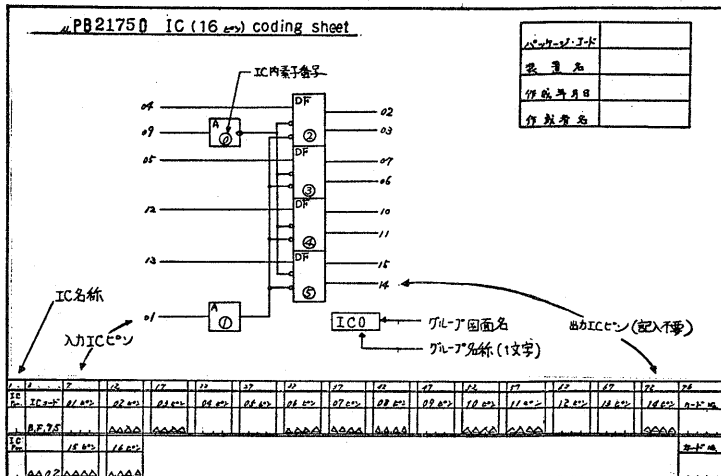


図7. コーディングシート

(2) 出力パッケージピンからの接続

この接続は、固定欄のカードに、出力パッケージピン番号と、入力先ICのIC名称-ICピン番号をペアにして記入する。(4~7枚/パッケージ。)

4.3 図面指定カード

パッケージの論理は、通常複数の回路図で表現されるため、このどれにどの回路を割りふるかを指定する必要がある。こ

の指定の単位としては、以下の2通りが許される。

- (1) IC単位-----その図面に含まれるべきICを、IC名称で指定。
- (2) グループ単位----その図面に含まれるべきグループを、IC名称、グループ名で指定する。

ひとつのICがある図面に指定されていても、そのICを構成するグループの一部を、(2)の指定法を用いて他の図面に割り当てることのできる。(各1枚/図面)

4.4 入力処理

入力処理では、パッケージ種類名、IC間接続カード、図面指定カードを読み込み、形式チェック、IC名称やICピンの2重指定チェック等を行った後ILBの情報をもとにして、未定義のIC種類の使用、ICピンの誤使用等がないかをチェックし、リストを出力する。

5. 展開処理

IC間接続カードによって与えられた接続情報とILBより、素子レベルに展開したパッケージの論理データを作成する。このとき、IC内で使用されているグループのみを、接続関係をたどることにより抽出して、図面位置割付処理のための準備をおこなう。また、どこからも接続のないグループを削除している。

使用されているグループ内でも、接続のない入力ICピンに対しては、空端子の処理が行われる。指定された空端子、およびICピンの接続も、この段階で処理される。

展開の際、IC内の接続はそのままにし、IC間の接続は、ICの端子(ICピン)を消去することにより、素子間の接続を作成する。この段階では、素子は正式の名称を持たない(図面位置割付処理ではじめて決められる。)ので、仮の名称として、IC名称 IC内素子番号(IC内ユニーク)が用いられる。

ワイヤードロジックについては、ILBにこの機能を実現する仮素子を登録しておき、ICと同様に接続を定義するので、展開処理においても全くICと変ることなく扱われる。

6. 図面位置割付

図.9 に示すような 75×70 枠の図面に、図面指定されたグループを、I L B中に定義された素子の相互位置をかえることなく割付ける。このとき、図面の使用密度をあげ、かつ、図面を見やすくするため、できるだけ論理が左から右へ流れ、信号線の交差等が少なくなるようにする。

図.8 に図面位置割付の処理フローを示す。

6.1 グラフ作成

グループ間のトポロジカルな関係を取り扱えるように、グループ間の接続情報から、処理図面毎のグラフを作成する。このグラフの節点には、処理図面に含まれるように指定されたグループ、それに接続する入力パッケージピン、および、この図面に含まれない入力先グループを対応させ、それらの間の接続を有向枝とする。入力パッケージピンと他図面の入力先グループとを節点に含めることにより、グラフにサーキットがある場合でも、これらに接続するグループを、図面の左側によせる効果が生じる。図.10 にグラフの一例を示す。

6.2 レベル付け

(1) レベル付け

図面内の論理が左から右へ流れるように、ラベリング手法を用いて、各グループ単位にレベルをつける。論理的に接続していないグループ相互を分離し、それらの間に同一レベルが生じないようにする。

(2) サーキット解消

上記ラベリング手法でレベル付けを行うとき、同時に、サーキットを検出し、サーキットの解消を行っている。このとき、レベルが付けられている節点のみを調べて、処理の高速性をはかっている。

6.3 割付処理

付与されたレベルにしたがって、各グループを図面に

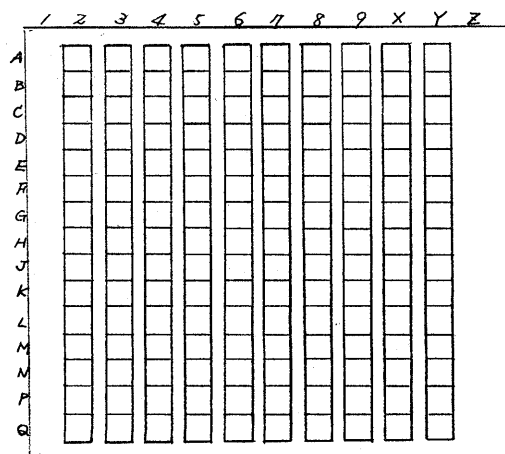


図9. 図面位置

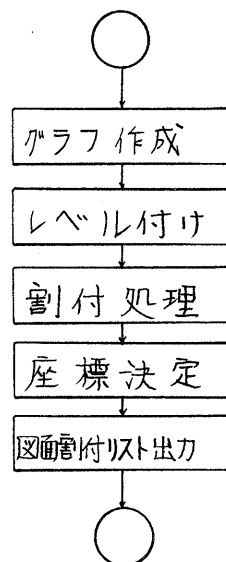


図8. 処理フロー

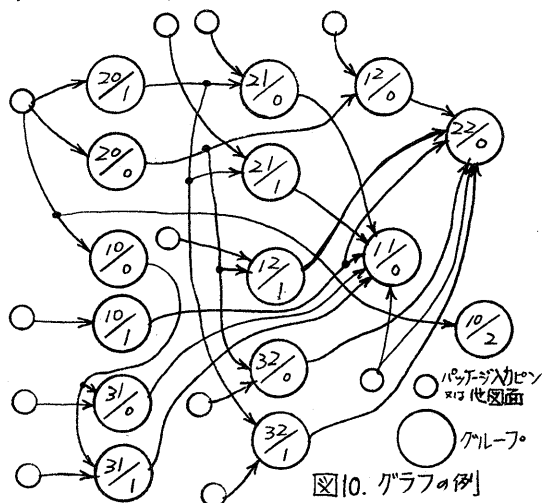


図10. グラフの例

割付け, その図面位置を決定する。このとき, 各グループの縦方向の間隔を一単位ずつあけ, 同一列の各グループの左辺をそろえる。列と列との間隔は, 含まれる最大の幅のグループによって決められる。このような前提のもとで, 以下の順にしたがって, 図面位置の割付けを行う。

(1) 初期割付

同一レベルをもつグループを, 同一列に割付ける。その例を図.11に示す。全グループが図面に入りできれば(3)へ。

(2) つめこみ処理

初期割付で全グループが図面上に入りきれない場合, レベルにしたがって図面の左側から順に縦方向につめて割付ける。その際, できるだけ信号線のもどりが少なくなるように考慮して割付ける。図.12にその例を示す。

(3) 交差減少処理

図の右列から順に, 発見的方法により, 列内のグループの順序を入れかえることによって, 信号線の交差をできるだけ少なくする。

(4) 列間調整

以上の処理の結果, 図面上でスペース的に余裕がある場合, 入力側の信号線総数の多い列から, 列間の間隔を広げる処理を行う。これにより回路図印刷プログラムでの未結線を少なくすることが出来る。

6.4 位置決定

割付処理によって決定したグループの図面位置と, ILBに登録されているグループ内の素子の相対位置とから, 各素子の図面位置を最終的に決定する。

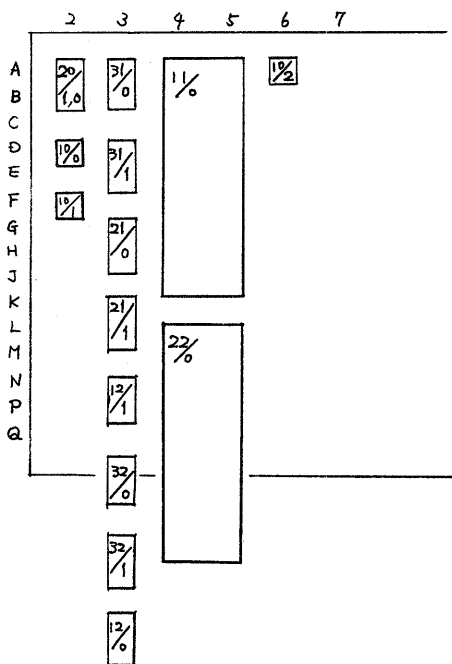


図.11 初期割付

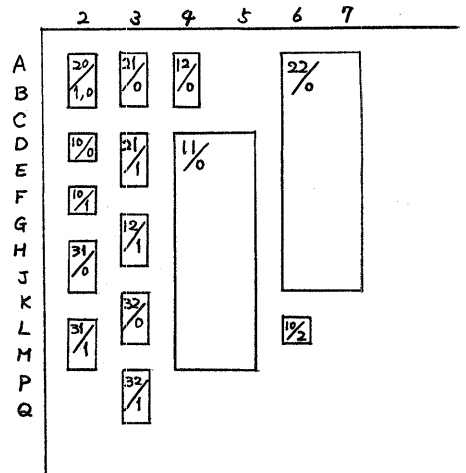


図.12 つめこみ処理

6.5 図面割付リスト出力

グループの割付結果のレイアウト図を、図面割付リストとして出力する。また、割付けることのできなかったグループを、未割付リストとして出力する。図面割付リストの例を図.13に示す。

IGAC	*****	IC - G R O U P	A S S I G N	C H A N T	*****	C H A R T - C O D E (C S Q)	P a C K E T
1	2	3	4	5	6	7	8
A		BH04					
		20-1					
B		BH04		B214		B214	
		20-0		21-0		12-0	
C							
D		BH04					
		10-0					
E				B214			BF75
				21-1			22-0
F		BH04					
		10-1					
G							
H				B214			
				12-1		BF75	
J		B214				11-0	
		31-0					
K							
L				B214			BH04
				32-0			10-2
M		B214					
		31-1					
N							
P				B214			
				32-1			
Q							

図.13 図面割付リスト

7. 出力処理 他

図面位置割付処理によって決定された図面位置をもとにして、各素子に素子名を付与する。素子名は、図面名称および図面位置から成る。素子名をもとに、素子間の接続データが作られ、IC割付情報とともに中間ファイルへ出力される。

この中間ファイルは、以前の素子レベルの入力と同一形式なので、細部にわたる修正等は、過去のプログラムを用いておこなうことができる。ただし、この方法による修正は、再び各種チェックプログラムのランを必要とするため、あまりおこなわれない。

このファイルを用いて、PLBにデータをロードし、パッケージ回路図印刷、パッケージパターン設計、検査データ生成等を行うことができる。図.13に示されたグループのレイアウトに従ったパッケージ回路図を、図.14に示す。

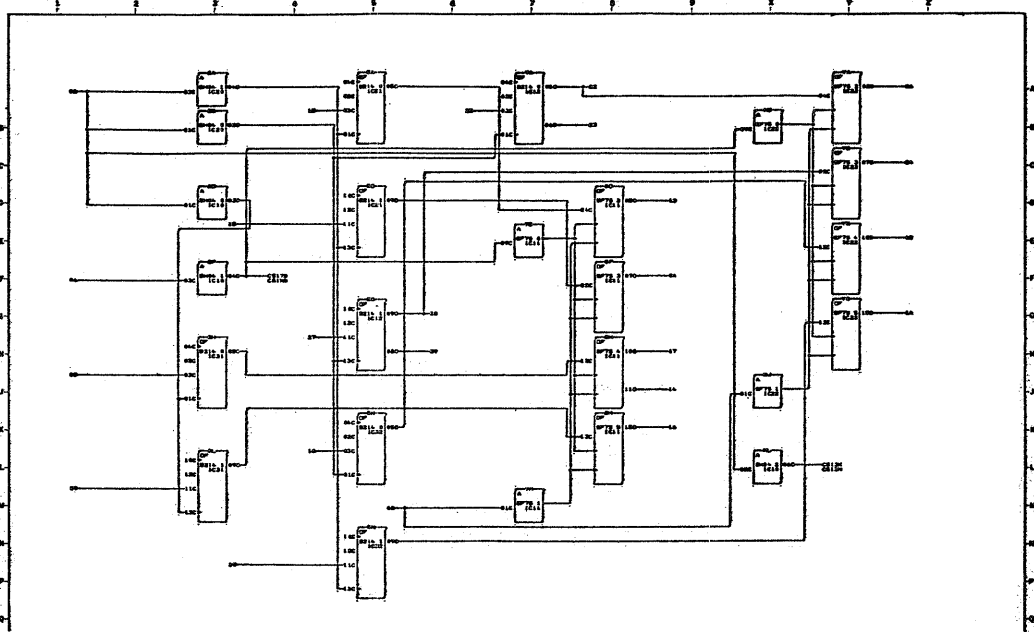


図1.4 パッケージ回路図(図1.3のレイアウト)

8. おわりに

PDIシステムの作成後、いくつかの大型論理装置の設計に適用し、予期通りの効果をあげることができた。従来の当社でおこなわれた素子レベルの入力方法と比較し、

- ・コーディング時間 約1/2 (パッケージ当り)
- ・入力カード枚数 約1/10 (")
- ・ランタイム 約1/4 (" 3~5分以内)

という実績が得られている。このほか、DA用入力回路図を作成する手間がなくなり、誤りが減少するなど、設計進捗上や品質上の効果があった。また、設計者からは、設計上のイメージがそのまま入力できる点が便利という評価を受けた。

いっぽう、図面割付けをすべて自動化しているため、くり返しの多い回路では、設計者のイメージと異った図面が出力されることがあったが、現在は、図面指定をやり直すことで対処している。将来は、適度な人間の介入を取入れていきたい。

本システムは、NEAC-2200 M500 エでインプリメントされ、コンパイル言語(BPL) 6Kステップ、カード枚数11K枚の規模を持つ。

今後の課題として、LSI等の設計に対し、機能ブロック等の新しい入力単位の追加、より見やすい図面レイアウトアルゴリズム、それにマンマシンインタラクションの導入等があげられる。

本システムの開発の機会を与えて下さった奥田部長、高部長代理、有益な助言をいただいた有馬主任、吉田吉憲氏、作成を担当された赤松富雄、小宮末吉、佐藤 保らの関係諸氏に深謝いたします。

参考文献

- 1)堀尾, 山本「論理素子の図面への自動配置」 昭和47年 信学全大予稿(1230)
- 2)渡会, 植村, 森「機能ブロックによる論理装置の設計言語の一例」 計算機設計自動化研究会資料74-11 (1974)
- 3)曾我, 田中, 田淵他「計算機設計自動化のためのデータベースシステム(EDMS)」 設計自動化研究会資料74-13 (1974)
- 4)北村, 他「論理回路図の自動発生」 情報処理 Vol.10 No.6 1969
- 5)後藤, 他「論理回路の自動製図システム」 情報処理学会12回大会予稿(17) (1971)