

TM-0124

AI テクノロジー
— LS レイアウト設計 —

後藤 敏, 森 啓, 光本圭子, 藤田友之
(日本電気)

July, 1985

©1985, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

AIテクノロジー-

5. LSI回路設計

日本電気株式会社・C&Cシステム研究所

後藤 敏， 森 浩， 藤田 友文， 光本 孝子

5.1 はじめに

LSIの集積度の飛躍的な向上、チップサイズの増大に伴い、1チップ上に搭載可能な回路規模は、この数年間に、大幅に増加した。このような集積回路製造技術の発展に即応して、電子通信機器のLSI化がエレクトロニクスにおける分野で急速に進んでいる。特に、特定用途向けのLSI（カスタム LSI）は生産量は少ないが、需要の品種と開発スピードは少なく、設計からLSI生産のネットワーク化されている。このため、インベスタータの導入により、設計の省カ化、自動化を徹底的に図り、信頼度の高いLSIを短期間に、単一コストで設計できるが、新しいエレクトロニクス装置の開発競争に打ち勝つ決り手となった。

然れ、LS工設計りから、最大の課題はレイアウト設計にあるといわれている。
 しかるに^今同書によれば、他の設計フェーズに比べて、設計と頻繁に起し、膨大な設計工数と必要とすることによる。更に、レイアウト用CADシステムの開発が、
 急務にゆえ、利用レベルで本格的に使用されており、その有用性が立証された格
 わいたことによる。

利用レベルでCADシステムが使用ゆえ、多くの解決すべき新しい課題が
 出現して来た。そのらは従来技術では解決困難であり、新しい技術の登場を待た
 望んでいる。

最近、人肉の思考理解といった知的活動をつかに形式化して、インピータ
 化を実現するかどうかという知識情報処理の所見も発表になっている。CADの分野にお

いても、知能処理技術を用いて、従来技術では解決困難な課題を解決しようとする気運が強い。従来技術では解決困難な課題が、知能処理技術の導入によって、完全に解決するといえることも、部分的解決、あるいはより良い解決法を求めて、知能処理技術の可能性を追求しているところもある。

ソフトウェア危機と呼ばれて久しいが、現在、CADシステム(ソフトウェア)も、次のような問題に直面している。

(1) インピーターが設計可能な分野、領域が狭く、多くの部分は未だ人間によらず、設計がなされており、多数の設計専門家を必要としている。

(2) インピーターによる設計結果は、人間の設計に比べて、はるかに良くなる。

(3) CADの運用は、設計の身内設計が行なわなくてはならない程難しく、設計身内設計に、本来の設計作家以外の時間を多く費やさせている。

(4) 製造技術の著しい変化と、設計技術への多様多岐な要求により、CADプログラムは大型化と、複雑な改造を強いられ、プログラムの開発と保守作業が膨大化している。

(5) CAD運用端において、大半の時間には設計データの入力とその妥当性のチェック作業にいられている。

このような課題を、新しい技術として知識処理で解決したいのである。

本件では、CAD/DAへ応用した知識情報処理技術(知的CADと呼ぶ)の基本的な考えを明らかにし、実際に開発したVLSI配線設計用エキスパートシス

アム(WIREX)の針を通して、知物CADをどのように実現したかを説明する。

5.2 知物CADの方法

インジペンタによる設計支援(CAD/DA)は、設計過程における種々の作業を、インジペンタ上で実行することにより、設計者の負担を軽くすることを目指すとしている。最近の超高度回路(VLSI)の設計などに見られる大規模な設計においては、人間だけで行なうことが困難となり、インジペンタによる設計支援技術の不可欠なものとなっている。

従来の設計支援システムは、以下のような順により開発されてきた。

- (1) 設計手順の標準化を図る。
- (2) 設計問題を図表化したモデルで表現する。

(3) 問題解決の方法モデルについて記述する。

すなわち、問題解決の手順を標準化して、列挙しているモデルが実際の問題を正しく反映している場合には、この設計交換方法が極めて効果的のものである。しかし、このような設計交換システムは、次のような問題点をかかえている。

(1) 設計の負担

求まるモデルを限定しているため、コンピュータで処理可能な範囲、設計作業負担が増加して少ない。標準化が困難な設計過程は、現在も人肉に委ねられている。

(2) システムの柔軟性

問題を表現したモデル、及び解決の方法がプログラムに埋め込まれているため、

最近の主要な問題である。とくに、設計の対象を大規模化、個別化されるにつれて、柔軟性をもたせ課題となっている。

(3) 設計判断知識の記述容易性

複数のトシードオットも存在するような状況を固定納にモデル化し、そのモデルの上で、与息をも標準化して記述することは困難である。状況に応じた判断を、設計の場面に対応させて、素直に記述できることも望まれる。

(4) システム利用の容易性

ほとんどの設計支援プログラムは、複雑なパラメータを要求するため、所望の結果を得るには多くの経験と知識を必要とする。

これらの課題を克服する方法として、知能CADシステムが考えられる。

てもとと向の信をう設計には、今までのインジエ一タの交換による設計と比べて、幾かに面がわるが、他のCADシステムでは、人工知能手法により、人肉の設計者の持つ知能をもつ入れたCADシステムを構成しようとするものである。

現在、設計交換システムとして知られているのは、CADシステムのアーキテクチャは、知能を格納する知識ベースと、それらを利用して問題解決を行う推論機構とでなるシステム(知識ベースシステム)である。これは、問題の解決も問題解決の知能も知識ベースに蓄え、状況に応じて、それらの知識を適用して適切な推論機能により、設計を進めるものである。このようなアーキテクチャにより構成される他のCADシステムの特徴として、次のような利点を挙げることもできる。

5.2

(1) システムの拡張性

対応の表現や問題解決について何らかの知識を知識ベースに独立に記述できると、システムの拡張性変更も容易である。

(2) 設計支援領域の拡大

問題解決のヒューリスティックを柔軟に記述できるため、固守したモデルでは表現できなかった経験則を用いることもできる。その結果、より幅広い領域の設計を支援することになる。

(3) 高品質設計

インジクタ上に蓄積された専門家レベルの知識を利用することであるため、設計の品質を向上させることができる。

以上、従来の内題解決法と知れCADの内題解決法を説明したが、こゝらをも以下に再整理して説明する。

一般に、内題解決のモデルは、 I を入力データの集合、 O を出力データの集合とし、内題解決法を f とすれば、

$$O = f(I) \quad (5.1)$$

と表さる。ここで、 I, O は部々集合から構成され、

$$I = \{i_1, i_2, \dots\} \quad (5.2)$$

$$O = \{o_1, o_2, \dots\} \quad (5.3)$$

f はアルゴリズム A と、過去のデータ (経験) D に与えられる。

$$f = \{A, D\} \quad (5.4)$$

アルゴリズムにより，内題解決を図っているものは，

$$O = A(I) \quad (5.5)$$

と表され，経験データによっているものは，

$$O_k = D(i_k) \quad (k=1, 2, \dots) \quad (5.6)$$

となる。一般に，内題解決は，

$$O_k = \{A, D\}(i_k) \quad (k=1, 2, \dots) \quad (5.7)$$

というアルゴリズムと経験データの両面から行っている。従来のCADプログラムは(5.5)式に基づいて作られており，知識(内題解決法)がプログラムの中に埋め込んでいる。このため，内題が高度化，複雑化するにつれ，プログラムが大型化し，開発・保守が困難となり，新しい機能の追加修正は，そのプ

ロプログラムを開発したものでない状況になっている。

一方、知識処理では、経験データ "D" の導入により、問題解決法を制御とデータという形で記述する。制御のために汎用な推論機能を用意し、データのためには一定の形式で経験を格納する知識ベースを用意する。もちろん制限はあるが、機能の追加や修正は推論機能を変えないで、知識ベースのデータの變更によって、行なうことができる。データの修正ならば、本体のプログラムの詳細を知らなくてもできるため、非プログラムの専門家でも、自由に適したシステムを容易にカスタムできることとなる。

これをアルゴリズムと対比させて見れば、図5.3のようになる。アルゴリズムにおける定式化は知識ベースにおける知識表現に対応し、アルゴリズムそのもの

は、知識表現と知識利用（推論機能）に対応する。

また、知識処理で重要な四要素は、以下のものをあげられる。

- (1) 知識獲得： 外部として与えられた知識を人間（エキスパート）から引き出せるか。更に進んだものとして、学習機能（自動的に知識獲得）も可能か。つまり、(5.2), (5.3) 式のデータの収集と分類作業といえる。
- (2) 知識表現： 引き出した知識をインジクタ上に表現できるか。つまり、(5.7) 式の A および D の表現法である。
- (3) 知識利用： インジクタ上に表現した知識も、ユーザの要求に応じて、適切な形でユーザに提供できるか。つまり、(5.

7) 式のデータと推論機能との間の最適な組合せの制御法である。

(4) 2次元ソング法: (1) ~ (3) のすべてにわたって、人間とマシンとの最適な機能の役割分担方式は何か。

以上の夜明け、知的CADを構築する際に解決しておかねばならない課題である。これらの課題を解決することにより、従来の法では何れも新しいインジュータの応用が広がっていく可能性も大きい。現在、上記の(2)および(3)は、かなり活発に研究が行われ、成果も出ているが、(1)に対してはインディュータの初歩的技法しかなく、おもしろ研究の進んでいない。(4)に対しては、新機能の7-7スラ-ジョンの出現等により、急速に技術が進んでいる。

現在では、いくつかの種類のADシステムが発表されてゐる。海外で発表された野矢系の知能CADシステムの内容を表5.1に示す。このシステムの思想的背景にあり、実用化するにはいくつかの基本的問題を解決してゆくことが必要である。

当社でも開発したWIREは実用的規模の問題の解決を組んだ知能CADシステムである。以下に、その実用方法について、具体的に述べる。

5.3 CAD向Prologインタプリタ

問題解決・推論機能は、人間の思考メカニズムを明確にし、コンピュータにより人間に近い知性処理を行わせようというものである。このような機能を実現するための言語として、述語論理型プログラミング言語"Prolog"が注目されている。Prologは、柔軟なリスト処理と問題解決・推論のための基本機能を備えている。このため、エキスパート・システムの構築や、知識を表現する言語として注目され、実用化に向け改良、拡張の研究が行われている [11] [12]。

しかし、大規模問題を取り扱うシステムを既存のPrologでリアルタイム表現するには、現在のハードウェア環境とPrologの処理系では、実行速度とメモリ使用量の点か

らなで問題がある。これは、計算の実行中、データトラップ時に代入をもてなす
ておいたデータの存在することにより、多くのメモリを使うこと、述語はデータ
ン又シンタックスエラーが起るため、述語の数が多くなると、述語の格納・
取出しに時間がかかることと原因と差えられる。

一方、システムが明確な問題に対しては、従来、アルゴリズムによって問題の解
決を回っており、これらはFORTRAN言語などにより、効率的なプログラム
が作成されている。しかし、システムが不明確な問題やシステムをみつけるのに試行
錯誤を要する問題に対しては、FORTRANなどのシステム言語では、プログ
ラム作成の効率が低く、Prologなどの論理型言語も適していると考えられる。

以上の点から、システム型言語と論理型言語とを結合した言語を作り、おの

の制点もいくつか使われ、より高度な知識的処理が効率よく行なわれる。また、すでに手続型言語によって、現在すでに構築された膨大なデータベースのプログラムの変種が、FORTRAN言語とのリンク機能によりProlog上で有効に使用できる。

ここでは、Prolog言語の標準機能は、新たにFORTRAN言語とのリンク機能を加えたPrologインタプリタ(CADLOG)の説明をしよう。

5.3.1 CADLOGの機能 [13, 14, 15]

CADLOGは、一階述言語処理を基礎にして開発されたProlog [16] を基本にしている。構文規則は、DEC-10版Prolog [17] に類似している。文字は、大文字、数字、記号の3種であり、変数は '*' を付加して記述する。プログラムの

は、節によって構成され、この節はいくつかの述語を組み合せて表現する。述語の引数は、定数として、文字列、数値、構造を記述することによって与えられる。算術演算子（加算、減算、乗算、除算）、数値の比較、バックトラックに對する制御を行う REPEAT, ! (カット) などから組込まれた述語として用意されている。

CALC はプロログのプログラムから FORTRAN サブルーチンをユーザで与える機能をもっている。FORTRAN で書かれたプログラムとのリンクは、これに對する述語によって行われる。この述語のことも、外部ファンクション述語とよぶ。この述語は、プロログの中では、頭文に 'F' を付加して、他の言語と區別している。

FORTRAN 言語とのリンク機能をもっているため、この言語で構築されている

データ構造を使用することによって、これによって、大規模なデータが効率よく処理でき、実行時間も短縮される。

5. 3. 2 FORTRAN言語とのインターフェイス

FORTRANで書き出したプログラムとのリンクは、外部ファインクション述語とよばれる述語とFORTRANのサブルーチンと対応づけることによって行う。Prologの中です、この述語の題文は「 P' 」を付けて、他の述語と区別している。1つの外部ファインクション述語に対して、引数の入出力の異なる複数のサブルーチンが対応している。

FORTRAN言語でも、サブルーチン内の引数はあらかじめ入出力の区別をしておかなければならない。一方、Prologの外部ファインクション述語

り変数も、実行時に値を決る。Prologの実行中に、変数の値が決まるに従い、
 凡対TPCNグラフの引数の入出力の区別が決まり、Prologからコールされる
 グラフの引数も区別される。もし、入出力が違ふ場合ある。サブルーチンがなけれども、
 Prolog上でね、その外部ファイングは述語をFailしたとみなし、バックトラッ
 クを有する。

例218, FUNCという節を下記のように表現する。

FUNC(#X, #Y) :- \$PROC1(#X, #Y). ... サブルーチンPROC1(X, Y)に対応

する外部ファンクション述語

FUNC(#X, #Y) :- \$PROC2(#X, #Y). ... サブルーチンPROC2(X, Y)に対応

する外部ファンクション述語

FUNC(#X, #Y) :- \$PROC3(#X, #Y). ... サブルーチンPROC3(X, Y)に対応

する外部ファンクション述語

*X, *Y

(注) ~~X~~ は入力変数、~~Y~~ は出力変数を表わす。

このFUNC述語を用いて, SUB_GOALという節を次のように定義する。

SUB_GOAL1(#X, #Y, #Z) :- FUNC(#X, #Y), FUNC(#Z, #Y).

この節において, 第1引数#Xが変数, #2, 3引数#Zが変数で1ニファイされた判

を国5.5に示す。この例では、\$PROC1、\$PROC2に外えるFORTRAN サンプルが

ンが置に実行され、オ2、3引数を決する。
 C ADLOGは、FORTRAN言語でのリンク機能をもっているが、この言語で表現されたデータ構造を利用することができる。
 この場合、システム全体の状態は、FORTRANで構築されているデータ構造を扱う変数と、

Prolog内に表わされる変数として表現されている。このため、Prologも実行中に、バックトラップも生じた際、Prologの変数に対してのみ、エニファイされる前の状態に戻り処理を行なう。Prologの変数が表現される状態と、FORTRANのデータ構造を扱う変数を表現する状態とに違いが生じる。したがって、この外部プログラムの記述がデータ構造内のデータも更新するかを常に意識し、バックトラップも生じる。このような処理を行なわねばならないかを考えて、プログラムの修正が必要がある。これら、プログラマに大きな負担をかけるばかりでなく、プ

ログラムを設計にくくする。CADLOGでは、データを更新する外部ファンク
ション述語に対応するPORTALサブルーチンに、常に更新前のデータを保持する
処理を組込込人でおさ、バックトラックが全じてこの述語もで戻った時に、保持
しているデータを用いて、データの改変処理を行なう機能をもっている。1つの
例を図5.6に示す。このように、PORTALで構築されたシステム内リデータ構
造を用いる場合でも、CADLOGはデータの改変処理用サブルーチンを組み込
むことにより、常にデータ構造内のデータの整合性を保つことができる。

5.4 VLSI配線設計問題への応用

図5.7

LSI設計の一般的な流れを  に示す。製造プロセス条件などの基本的な設計ルールが確立されていると、あらかじめトランジスタなどのデバイス設計、電子回路設計をおえた論理機能単位（ブロックとよび）をライブラリとして登録しておく。設計すべき個別の品種の論理仕様が決まれば、このブロックを用いて論理接続情報を作成する。遅延時間やフリティカルな回路に対しては、回路解析を行い、必要であればブロックの再割付をする。同時に、論理の正当性を論理シミュレーションによりチェックする。正しいと認められた論理回路は、製造検査のためにテストパターン系列の発生と故障対策のためにシミュレーションと

行う。一方、レイアウト設計において、ブロックの配置場所と信号線、電源線の配置場所をプログラム上で決定し、マスキパターン製造用テープを作成する。

ここで、レイアウト設計問題とは

(1) 回路接続条件：論理的な接続関係

(2) 幾何学的条件：製造プロセスより決まるもの（配線幅、コンタクトサイズ、配線間隔、コンタクト間隔、配線—コンタクト間隔）

(3) 電氣的条件：遅延時間、雑音

等の条件を守るとして、与えられたチップサイズ内に、又は可能な限り小さいチップサイズに納まるように、ブロックの配置と信号線、電源線の配線経路と決

$$E_{\mathcal{F}^{(2)}} = E_{\mathcal{A}^{(2)}}$$

レイアウト設計問題は、大きく分けて干渉上において、回路の構成要素である素子の配置場所を決めること（配置問題）と、素子間の信号系の配線場所を決めること（配線問題）の2つに分けらる。本文で扱う問題は、配線問題であり、

1621 1915. 8 10 示。

图 5.8

(a)は、配線が通れる格子を設定したマップであり、A, B, Cの3種類の名前や格子点二につけられている。同じ名前が付いた点を接続し、異なる名前が付いた点間にはポートしないように、配線パターンを定める。配線パターンは、
図5.8(ア)のように使用して接続し、2つの層-要素にそれぞれ1回、スルーホールの穴を通り抜けて接続する。図5.8(b)は1つの実現例を示している。

この場合、3種類の配線要求に対して、すべての配線パターンから見つけられる。

一般に、多変数関数の最適化問題を考えるとき、配属法（ダイナミック・プログラミング）を用いる。これは、問題を小さな部分問題に分解し、それらの部分問題を逐次解いていくことで、全体の最適解を求める方法である。配属法の適用範囲は、動的計画法（Dynamic Programming）と呼ばれる。これは、最適解の構造が最適部分解と一致する（最適性原理）という性質を持つ問題に適用される。配属法の適用範囲は、動的計画法（Dynamic Programming）と呼ばれる。これは、最適解の構造が最適部分解と一致する（最適性原理）という性質を持つ問題に適用される。

自明の理である。このプログラムには、セーリス・プログラム法が組込まれており、このプログラムで、配列計算を完成させることは現実には不可能である。

ところが現在のCADシステムでは、自動配線プログラムが結線できなかつた信託線を結線したために回路ファイルがディスプレイ上で専門家に会談的に図形配線パターンへ追加、修正を行なっており、この作業は設計者にとっても大きな負担となり、設計時間の多くを費している。この未結線処理のために専門設計者は、設計対象となるVLSI自体の知識、ルールとなるCADシステムについての知識、配線を行なうための結線解決のための知識を駆使して、短時間で質の高い設計を行なう。

したがって、未結線処理の専門家の設計ノウハウを計算機上に表現することはでき、設計時にそのノウハウを有効に利用できるとすれば、非専門家でも容易に専門家のような設計を行なえることとなり、現在のVLSI設計にとっても最

大の線路局で、その設計者の意思に對する突破口となる時期が過ぎる。ここではVLSIの配線レイアウト設計に限って議論を進めるが、本気で述べる知識工学の手法はVLSI設計の全体であり、配線設計領域の全般にわたって適用できる。

*1

5.4.1 配線設計エキスパートシステム

VLSIレイアウト設計、とくに配線レイアウト設計においては、CAD化がかなり早くから進められており、すでに、数多くの優れたパッケージを持つCADプログラム資産が蓄積されている。これらのプログラムのほとんどでは多大の費用をかけて開発されて、実際の運用に供されたものであり、運用中にはいかに問題点を改良等も行なわれ、使用上のノウハウも集積されている。また既

*1

本システムの開発は新世代VLSI技術の発展機構(EDOT)の第5世代コンピュータプロジェクトの一環として進められている。

そのCADプログラム自体、貴重な知識の集まりと見なすことができる。実用的知識ベースのCADシステムを構築するためには、これらの蓄積された膨大なプログラム資産やデータベースを無視することはできず、これらを有効に活用できることが必要である。

従来、エキスパートシステムあるいはコンサルテーションシステムとして開発されてきた種々のシステムでは、図5.9(a)に示すような構成が一般的である。ここでは、人間がエキスパートシステムからコンサルテーションを受けるという形態となっており、いわば、システム上位の形で、人間がシステムからサービスを受けている。知的CADシステムの場合には、設計という人間の創造活動を支援するものや要求され、設計者には与えられる限りの自由度や許されるべきである。

既存の CAD ツールを用いて設計を行なう設計者と前提に考えと、知的 CAD システムとしては、図 5.9 (b) のような構成が望ましいと考えらる。これは、CAD システムの操作自体は、機械的なものが多く、エキスパートシステムで得た結論と連結することからである。設計対象の大型化に対処する手法の一つは、いわゆる人海戦隊であった。知的 CAD に期待されているのは、人海戦隊の兵士の役割であり、ここでは、設計者は指揮官の役割をするのである。

知的 VLSI 配線設計システム WIEREX の構成を図 5.10 に示す。本システムは既存の会話型 LSI レイアウト CAD システム LAMBDAL [19] と接続している。WIEREX システムでは、知識ベースと CAD データベースの 2 種類のデータベースをもっている。CAD データベースには、設計対象の物理的な情報

であるLSIの設計ツールが格納されている。VLSIの設計では、膨大な量のデータを取扱ひ、かつ複雑にアクセスするため、データベース構造、アクセスプログラム等に設計支援、既存のCADシステムを通じて、データの読み書きが行なわれる。知識ベースには、問題解決のための設計者の知識ノウハウが格納されている。推論システムはこの知識を利用して、既存CADシステムLAMBDAを通じて高度な設計を進める。推論システムとしてProlog型インテプリタであるCADLOGインタプリタを採用し、知識ベースの記述はCADLOGを用いて行なっている。

WIREXシステムは、NEC MSミニコンピュータ上で稼動し、グラフィックディスプレイエディタマシンインタフェースに用いている。図5.11にハードウ

エア構成, 図5.12にハードウェアの外観を示す。述語論理型プログラミング言語 Prolog は柔軟なリスト処理と問題解決, 推論のための基本機能を備えているためエキスパートシステムの構築や知識を表現する言語として注目されている。しかし, 大規模問題を扱うシステムを Prolog ですべて表現するには, 実行速度やメモリ使用量の点から見て問題がある。一方, 手続き的な明確な問題に対しては, 従来アルゴリズムによって問題の解決をばかっている, ことから FORTRAN 言語などにより, 効率の良いプログラムが作成されている。そこで,

高度な知識処理と効率良く行なうには, Prolog 言語の標準機能に, 新たに

FORTRAN 言語とのリンク機能を付け加えた CAD 向けの Prolog 型言語 CADLOG を開発した。FORTRAN プログラムとのリンクは CADLOG の外

部フアンクショナル述語によってとられる。例えば、CONNECT (XNET) は未結線端子間の接続を行う外部フアンクショナル述語で、変数XNETは結線可能な未結線を表わす。DELETE (XNET) は既配線XNETの削除を行う外部フアンクショナル述語である。これらのフアンクショナル述語は、FORTRANサブルーティンによってプログラムし、効率的に操作している。

5. 4. 2 配線設計における知識とその表現

配線設計のような複雑な組合せ問題を短期間で解決するためには、経路探索を

通して得られたヒューリスティックな問題解決の知識を状況に応じて適用することが必要である。そこで、このような設計過程に人工知能的手法を取り入れ、従来設計者が判断していた操作をシステムで実行できるようにする。

れば、設計工数の大幅な削減が可能になると考えられる。自動配線後の配線レイアウト完成のために、設計者は経験に基づいた知識により、会話型レイアウトCADシステムを用いて既存配線パターンの変更および未結線の接続を行っていき、本システムでは、このような設計者行なっている一連の手続きと知識の基本要素と考え、ルールとして表現している。

例として図5.13に示すような状況では、信号ネットBの配線系によって信号ネットAの一方の端子が妨害されているため、信号ネットAの経路を見直さなければならぬ。この未結線状態は次のような一連の手続きで解決できる。

- (1) 信号ネットAの結線と妨害する既配線Bを見直す
- (2) ネットBを削除する

(3) ネットAを接続する

(4) 削除したネットBを接続Bを接続する

この手順は図5.14に示すようにルールとしてCADDLOGで記述される。*で始まる文字列は変数であり、*NET-Aは未接続ネット、*NET-Bは未接続ネットを妨害しているネットを表わしている。BLOCKINGは未接続ネットや条件的に与えられたときに妨害ネットと求める処理を表わしている。一般には妨害ネットBの候補は複数であり、上の(3)ある(4)のステップで失敗(fail)したときにバックアップが起こり、別の妨害ネットを試みる。このとき配線削除をする前のインターフェイスに戻すことも必要となるが、FORTRANで書かれたインターフェイス用のルーチンを用意することによってこれを実現している。

既知の配置パターンを変更するためにルーチンとして図5.15に示すような系統
 的な知識がある。また適切なガイダンス—表ネを行なうことは会話型システム
 では非常に重要であるが、表ネのための知識としてルール化され、それらにより
 表ネを制御している。図5.16(a)は自動配線後の状態である。この例に示さ
 れるシステムの部ミを説明する。本結線の解決においてシステムはまず、解決すべ
 き本結線にコンパを提案する。とくに本結線にコンパ相互に絡み合っている場
 合には、本結線にコンパの処理手順は重要な意味を持っている。本結線にコンパ
 の処理手順の決定に関しては、コンパ間の距離によるもの、コンパに属する
 コン同士の相対的位置関係によるもの等、いくつかの因子が存在する。と
 りの例に示す、用いているCADによる性質に依存したものである。例えは、
 5.15
 5.16

ンダの距離や長さ、ディスプレイの一面に入り込む、全体の状況からかみにといったために、先に処理を行う。あるいは、コンピュータA, Bに属するセンサが図5.17に示すような位置関係にある場合、Bの方がAより優先する等である。

図5.16(a)においては、矢印で示した末結線の処理をシステムが要求している。オペレータが承認すると、その末結線に対する根拠経路を提示する。概略経路とは、いま考えている末結線コンピュータE、このような配線の引き回し方を示すものやということ、会話型設計においては最も重要な部分であり、主としてパターンの認識の能力やオペレータに要求される。また、配線の混雑箇所の中で、解決できそうな箇所を選択する直観的能力である。その中で最も基本的な判断基準は、混雑度である。そこで本システムでは、概略経路の提示において、

図5.18に示すような経路パターンに沿った複雑度を評価することにより行われる。オペレーターが、これを配線図の複雑の解決が困難と考えて却下した場合には、その別の概略経路の提案をさせる。オペレーターが概略経路について承く認識すると、その概略経路に沿っての未結線の解決が試行される。端点周辺でブロッキングあるいは配線干渉やエネルギーのホットスポットの解消に対して、種々の状況に対する戦略がルールとしてある。先に(1)～(4)に示した未結線解決の系統では、各ステップで用いる知識にさまざまなバリエーションがあり、現在60余りのルールを用いられている。システムはこの状況に適用できるルール群を評価し、最も適当と認められる未結線解決のルールをオペレーターに提案する。この列の場合には、図5.15(b)の1で示す場所にある経路を修正すべきだという提案である。これを

受け入れると、妨害既配線の削除、標識線の結線や行なわれる。結線が成功すれば、次の未結線解決に移る。失敗した場合には、別の妨害既配線の修正を試みられる。この制御にはPrologのパック機能が利用されている。以上は人間のオペレーションをプログラムしてあると考える。

システムは未結線解決のための戦略をオペレーターに提案し、承認を得られるば、それを実行する。もし、承認を得られない場合は、その次に適当と判断した戦略を提案する。オペレーターはシステムが提案している戦略をより大局的な見地から適不適判断して、システムに実行させる。またこの際、システムはオペレーターを判断しやよいディスプレイ画面表示をしてくれるにより制御している。従来の会話型CADシステムでは、ディスプレイ画面^{の制御}はオペレーターに行きわた

いかに適切な価値設定を認識しに基づいて行なうこととは、実用上大きな意味を持っている。図5.16(a)の状況に対して、本系統の解決のルールを用いて適用した結果を図5.16(b)に示す。ここで①は削除された故障配線系統の経路であり、②は図5.16(a)の矢印で示す本系統に対して配線経路である。③は故障配線の代替経路である。これらの経路のルールによって自動的に決定される。

このように、従来のCADシステムでは故障系統の提示、配線経路の指示を細かく与える必要があったが、本システムでは、配線経路をシステムで提案し、オペレーターは煩雑なシステム操作から解放される。これにより、設計対象について詳しく知らない非専門設計者でも、専門家並みの高度な設計を行なうことが可能となる。このようなプロダクトとすることにより、次のような特徴をもつ。

のシステムの構築が可能となった。

(1) 専門家の設計ノウハウを知識ベースに蓄え、利用できるように、非専門家で高度な設計が可能である。

(2) ソフトウェア開発の専門家でないLSI設計者から、自分自身で設計ノウハウを知識ベースに格納できるように、CADシステムのアダプタプログラムの構築、改良が容易になる。

(3) 実用規模の問題を高速に処理できるエキスパートシステムである。

① 問題解決の連続的な知識

② 状況判断の知識

③ 情報表示のための知識

- ①は未結線系を解決するための手続き的知識と配線パターンを改良するための手続き的知識である。現在、ロビン周辺のプロッキング解決ルールとして42, 10パターンを改良ルールとして15のルールがある。②の状況判断の知識は、どのような未結線系状況に対して、最も適切な未結線ルールを選択するための知識等である。現在61のルールがある。③の情報出力は会話型CADシステムでほととくになる。本システムでは一画面の表示領域、表示項目等の画面制御ルールにより行っている。現在、このようなコントロール用ルールとして103のルールがある。未結線系画面の2つの端3間の距離を大きく、適切な画面設定やでよい場合には、図5.19のように線散画面に経路を分割し、各画面で仮想的

トランペア S'T'を結ぶ部分経路を①のルールにより飛生する。但し、S'T'の位置の入力は現在オペレータのコンピュータ入力である。

6. おわりに

ゲートアレーコストアライシステムについて、MOS系、バイポーラ系を比較し、数品種づつテストを行なった。2100ゲート CMOSゲートアレーの例では、2140の信号のピンから1448のピンまでの接続要求を持ち、自形配線では、12本の未結線が残った。この未結線処理にWIREXシステムでは、約30分で行って結線できた。1191ピンの接続要求を持つバイポーラゲートアレーの場合は、自動配線の6本の未結線と処理するのに25分を要した。

スト結果をとりあつかうように述べることもできる。

(1) 処理時間、従来の会話型CADシステムを用いた場合の $\frac{1}{2}$ から $\frac{1}{3}$ は同等程度であった。但し、本システムでは殆んど予備知識が不用であること、^{従来の}会話型CADの場合にはかなりの熟練を要する。

(2) システム操作に要する手間をタッチ数で表わすとすれば、 $\frac{1}{10} \sim \frac{1}{5}$ と少なかった。

(3) 配線混雑度、以下aものについて、非常に効果的であった。1本の未結線解決に人手で数十分も要するようでは配線混雑度が高いものについては、あまり有効では無い。

(4) ルールによる画面制御は、実用上非常に有効である。

既知問題は、組合せ探索の問題であり、問題解決の知識によつて探索範囲を狭めるが、システムの動きをトレースすると、明らかに無駄と見わゆるサージや数多くみられ、(1)のようになり、とくは時間的に短縮できないう原因とわかつてゐる。無駄なサージを生じた原因を明らかにし、新たな知識して追加してやることにより、時間の短縮が可能になる。(3)においても、配線混雑度が高く70%に上り、考慮すべき組合せの数も爆発的に増加し、サージの効率も落ちるからである。適当なルールにより、サージのスペースを小さくしてやらねばならない。また、システム開発におけるメンテの利益として、次のようなことにかゝつた。

(1) ルールの追加容易であり、システムの改良も速くできる。

(2) インタプリタであり、従来型のシステム開発に比し、デバッグが楽

である。

(3) プログラムの見やすい。

(4) 自動バリエーションの機能のため記述性が良い。

本システムの開発を通じて、知識を格納する知識ベースと、それらを利用して問題解決を行なう推論機構からなる知識ベースシステムのアーキテクチャが、知的CADシステムのアーキテクチャとして有効であることが確認できた。今後、本文で述べた方法により、現在のCADが直面している様々な問題が解決され、より人間に近いコンピュータ・システムが出現することを目指す。

(人間の思考と理前といった知的活動もコンピュータ上で実現され、

- [1] ^{R.} Richard M. Stallman and ^{G.} Gerald J. Sussman, "Forward reasoning and dependency directed backtracking in a system for computer-aided circuit analysis", Artificial Intelligence, Vol.9, pp.135-198, 1977.
- [2] ^{J.} Johan De Keleer and ^{G. J.} Gerald Jay Sussman, "Propagation of constraints to circuit synthesis", International Journal of circuit theory and applications, Vol.8, pp.127-144, 1980.
- [3] ^{J.} John McDermott, "RI: A rule-based configurer of Computer Systems", Artificial Intelligence, Vol.19, No.1, Sept., 1982.
- [4] T. Mitchell, L. Steinberg, R. G. Smith, P. Schooley, V. Kelly and J. Jacobs, "Representations for Reasoning about digital circuits", Proceedings of the seventh International Conference on Artificial Intelligence, pp.343-344, 1981.

- [5] M. Mark Stefik, D. Daniel Bobrow, A. Alan Bell, H. Harold Brown, L. Lynn Conway and C. Christopher Tong, "The partitioning of concerns in digital system design", Proc. of the 1982 Conference on Advanced Research in VLSI, M.I.T., pp.43-52, 1982.
- [6] M.R.Genesereth, "Diagnosis using hierarchical design models", Proc. of AAAI-82, pp.278, 1982.
- [7] R.Davis et al., "Diagnosis based on description of structure and function", Proc. of AAAI-82, pp.278, 1982.
- [8] D.B.Lenat, W.R.Sutherland and J.Gibbons, "Heuristic search for new micro circuit structures", AI Magazine, Vol.3, pp.17-33, 1982.
- [9] T.J.Kowalski and D.E.Thomas, "The VLSI design automation assistant: First Steps", Proceedings of COMPCON'83, pp.126-130, 1983.

[10] J.H.Kim, J.McDermott, and D.P.Siewiorek, "Exploiting domain knowledge in IC cell layout", IEEE Design and Test, Vol.1, pp.52-64, Aug., 1984.

[11] 中島, "Prolog/KRの概要", 情報処理学会 記号処理研究会, 18-5, 1982.

[12] 新井, 中嶋, 小長谷, 楠田, 島津, 横田, "文字処理を導入したProlog: Shape up に ついて", Proc. of Logic Programming Conference, 1983.

[13] 藤田, 笠越, "CAD向けProlog型インタプリタ", 情報処理学会, 第27回大会, 1983, pp.329-380.

[14] 見本, 森, 森田, 生越, "CAD LOG - CAD 向 Prolog インプ
"9 -", Proc. of Logic Programming Conference, 1974.

[15] 生越, "CAD 用知識処理言語の研究", PIXEL, No. 24,
昭和59年9月.

[16] R. Kowalski, "Logic for problem solving", Elsevier
North-Holland Inc., 1979.

[17] W. F. Clocksin and C. S. Mellish, "Programming in
Prolog", Spring Verlag, 1981.

[18] 梶原, "LSIレイアウト設計のCAD", 電子通信学会誌, 46巻

12号, 昭和56年12月, pp.1274-1281.

[19] S. Goto, et al., "LAMBD A, an Integrated Master-Slice LSI CAD system", INTEGRATION, the VLSI Journal, Vol. 1, No. 1, April 1983, pp. 53-59.

[20] H. Hori, K. Mitsumoto, T. Fujita and S. Goto, "Knowledge based VLSI routing system - WIREX", Proc. of International Conference on Fifth Generation Computer Systems, 1984, pp. 383-388.

- (21) T. Fujita and S. Goto, " Knowledge-base and algorithm for VLSI design", Proc. of International Symposium on Circuits and Systems", 1985, pp. 873-876.

システム名	機能	開発機関	発表年	文献
EL	電子回路解析	MIT	1977	(1)
SYN	電子回路設計	MIT	1980	(2)
RI	コンピュータ構成支援	CMU	1980	(3)
—	回路設計支援	Rutgers	1981	(4)
Palladio	VLSI回路設計支援	Xerox/Stanford	1982	(5)
—	回路診断支援	Stanford	1982	(6)
—	回路診断支援	MIT	1982	(7)
—	三次元回路兼子設計	Stanford	1982	(8)
DAA	設計自動化支援	CMU	1983	(9)
TALIB	セルのレイアウト	CMU	1983	(10)

表5.1. 海外における知的CADの例

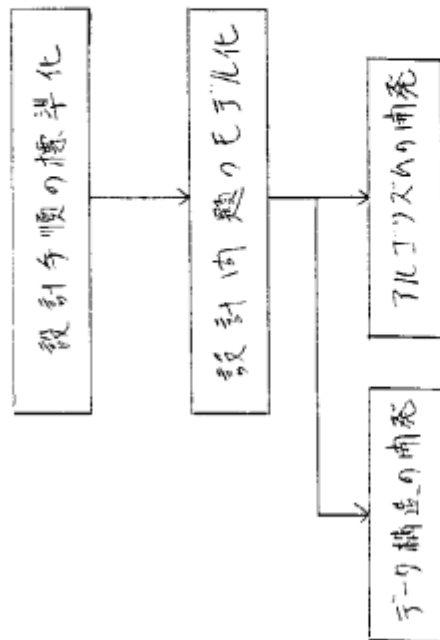


図5.1. 従来のCADシステムの開発

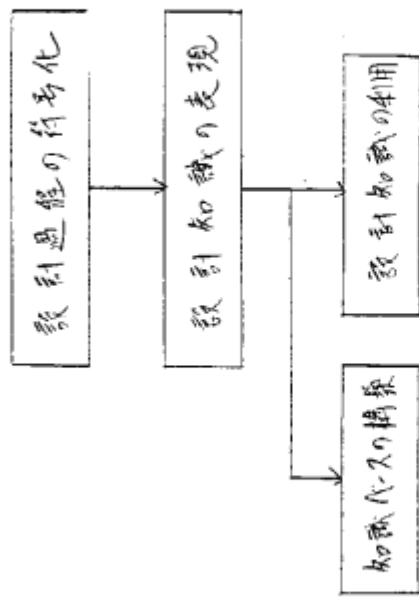
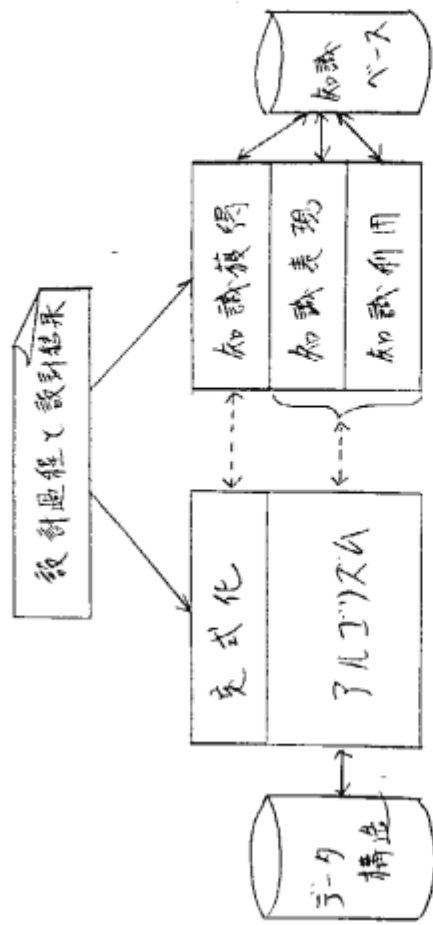


図5.2. 知的CADシステムの開発

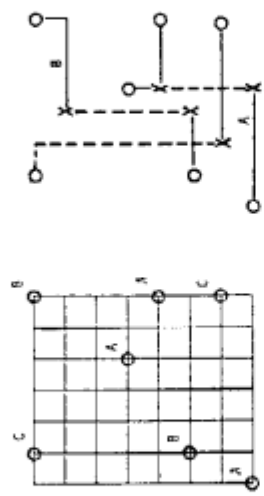


アルゴリズム方式 知識ベース方式

図5.3 従来の方式(アルゴリズム)と知的CAD方式(知識ベース)との解決の対比

図5.4 CADLOGインタープリタ構成

図5.5 外部ファンクション述語に対応する PORTPAN サル・サンバ
コールされた様子



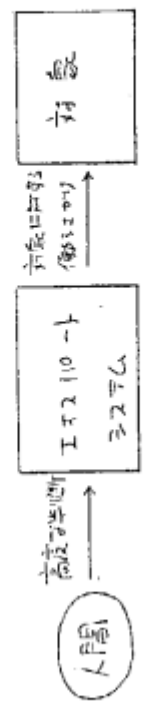
(a) 経路要求 (b) 配線パターン

- O : ヒン
- X : スルポイント
- : 第1層配線
- : 第2層配線

図5.8 配線設計問題



(a)



(b)

図5.9 ISPシステム構成

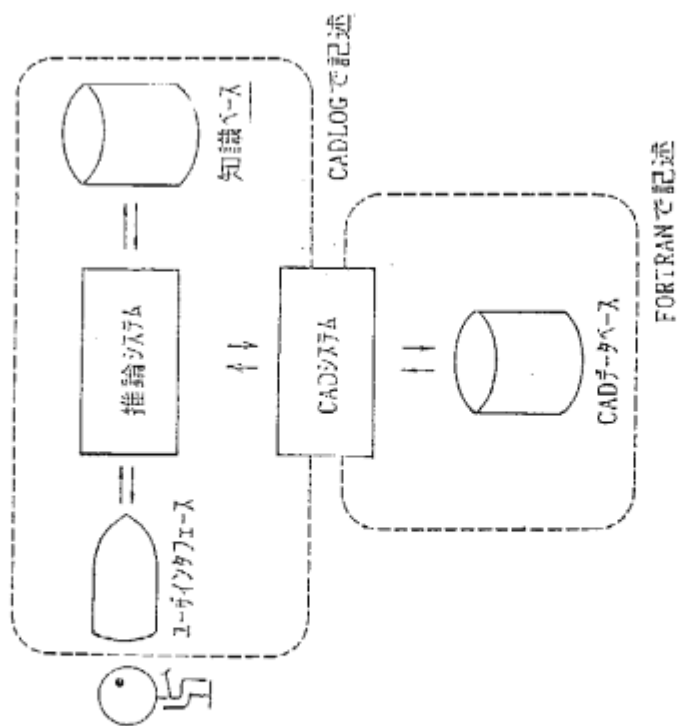


図5.10 WIREX のシステム構成

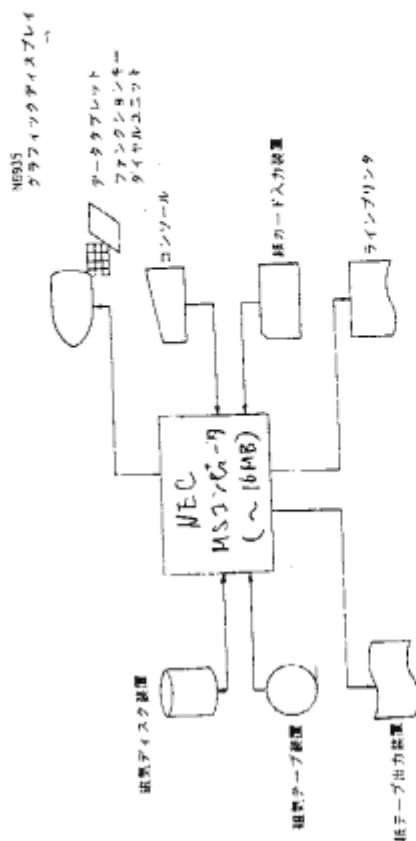


図5.11 ハードウェア構成



图5.12 WIREX 工作站

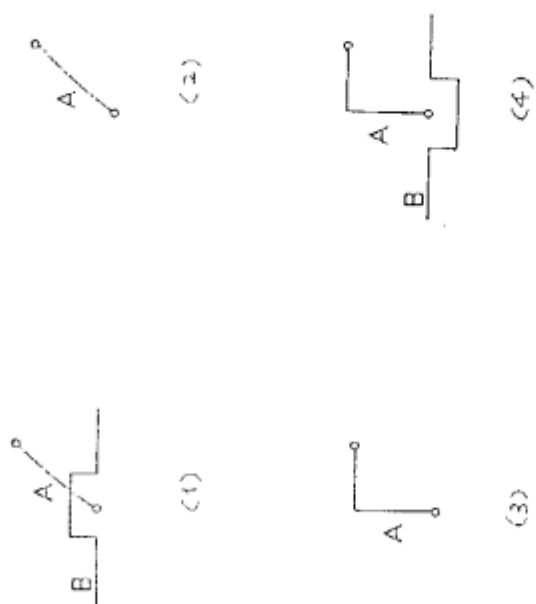


图5.13 未结线解决

```

RULE30(=NET.A):-
  BLOCKING(=NET.A,=NET.B),
  $DELETE(=NET.B),
  $CONNECT(=NET.A),
  $CONNECT(=NET.B),

  BLOCKING(=NET.A,=NET.B):-
    GETPIN(=NET.A,=LAY,=XS,=YS),
    EQ(=LAY, 1),
    DIRECTION(=YS,=DIR),
    $GRIDSEARCH(=DIR,=LAY,=XS,=YS,=NET.B).
  
```

図5.14 CADLOGによる記述例

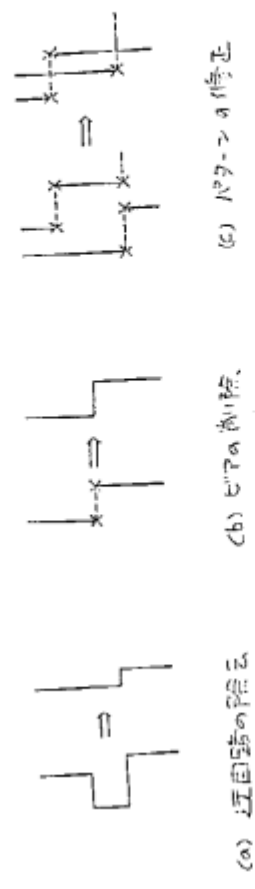
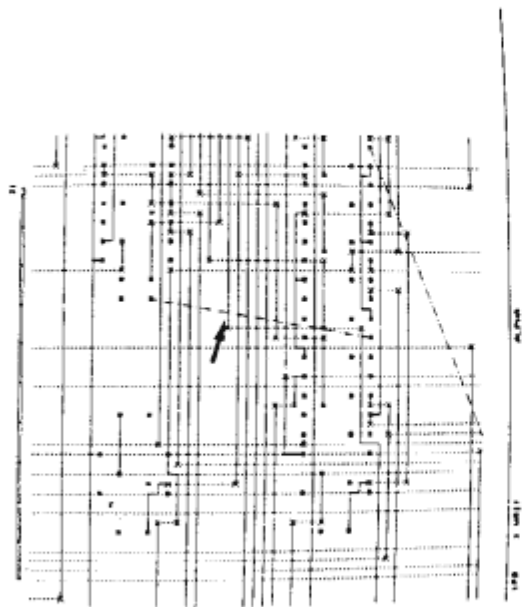
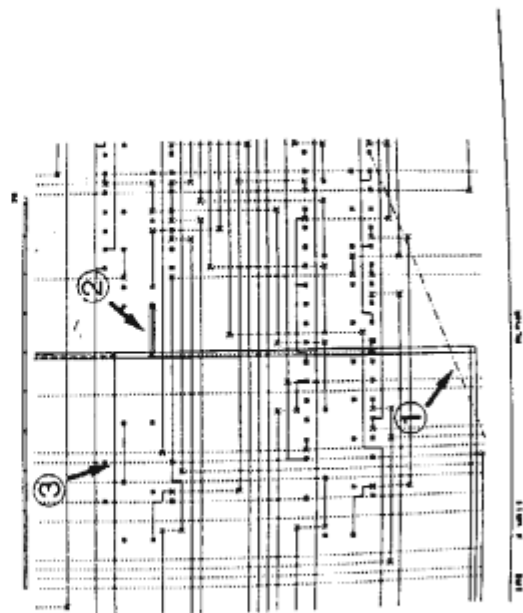


図5.15 ネットワーク変更のルール



(a) 凡点適用前



(b) 凡点適用後

図5.16 未結線部決凡点適用

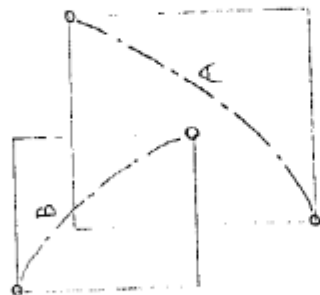


図5.17 点A, Bの相対位置

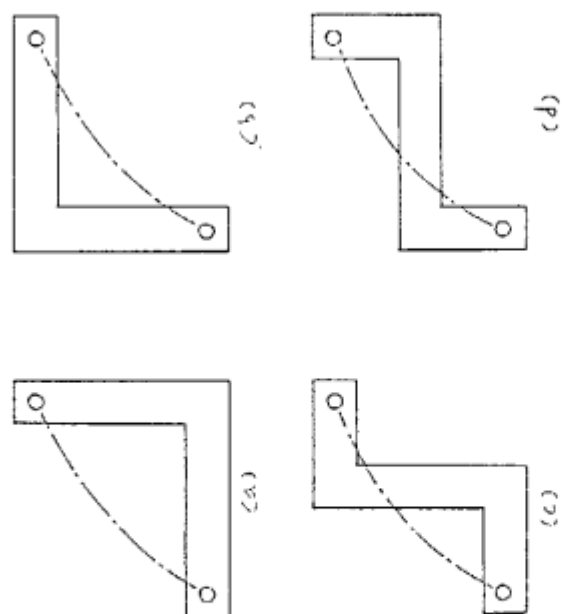
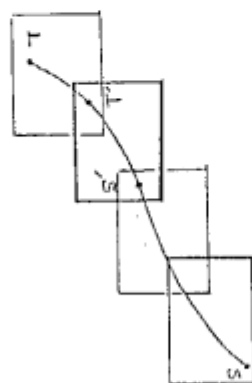


图 5.18 概略配筋



4月5日 画、面移動