

講座

# コンピュータの設計自動化(1)\*

倉地 正\*\*

## 1. はじめに

近年の電子計算機の普及ぶりは目ざましいものがあるが、その内容においても機能の増大の傾向は著しい。これに伴い新しい計算機を設計するのに必要な工数も増大するので、大規模な設計自動化システムが不可欠な要素となっている。

初期の設計自動化(以後 DA と略記)の対象は個々の人手設計作業をプログラムに置きかえる単発的なものであり、個別布線の布線設計など製造資料を作成する分野から始められたが、その後次第にプリント板の配置・パターン設計、論理シミュレーション、プリント板の機能試験データの自動作成等設計の各分野が対象にされ、処理が連続して行なわれるようになった。

最初の一貫した DA システムとしては IBM 社の Solid Logic Design Automation<sup>1)</sup> が有名である。この頃のシステムは各種の処理をするプログラムが直列に実行され、それらの間を中間ファイルで接続する形式のシステムであったが、処理が多様化し設計過程のフィードバックも必要になってきたため、最近のシステムは設計情報を記憶するデータベースを中心に構成され、処理の順序に自由度を持たせている。更に運用の容易化、信頼性の向上のため専用のモニタを開発した例<sup>2)</sup>もある。

設計手順も DA の影響を大きく受けるようになっており、設計者にとっては DA に関する知識は必須である。

DA システムを使う利益としては

- (1) 設計者の創造性向上
- (2) 設計期間の短縮
- (3) 設計内容の信頼性向上(チェック・評価)
- (4) 製造・検査資料の自動作成

\* Design Automation of Computers by Tadashi KURACHI (Computer Hardware Development Dept., Ome works, Tokyo Shibaura Electric Co., Ltd.)

\*\* 東京芝浦電気(株)青梅工場ハードウェア開発部

- (5) 各種資料間の統一性(無矛盾性)
- (6) 標準化推進
- (7) 設計資料管理の容易化
- (8) 設計費用削減

等がある。ある試算によれば DA システムを使った場合の設計工数は人手設計の場合の 25% 以下、費用は計算機の使用料を含めても 50% 以下といった数字が出ている。設計ミスによるやり直しの費用、製造等の自動化による省力費用を加えると DA の効果ははるかに大きなものとなるであろう。

本稿は DA システムの一般的な構成及び代表的な処理アルゴリズムについて説明を行うものであるが、紙面の都合上関係論文を全てサーベイする性質のものではなく、代表的な手法及び最近の動向について簡略に理解しやすい形で紹介するものである。従って厳密なアルゴリズムについては文献リストに掲げた原論文を参照していただきたい。

## 2. 設計自動化システムの構成

電子計算機の設計の流れは対象とする装置の規模、使用するテクノロジー(実装方式)により異なるが、図-1 に示すような流れを取るのが普通である。仕様検討、機能設計の部分は主として人手により進められ、論理設計、ファームウェア(マイクロプログラム)、診断設計、ソ

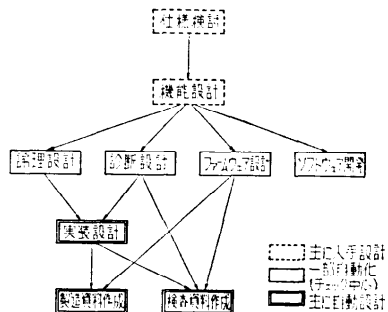


図-1 電子計算機設計の流れ

ソフトウェア開発の部分は処理の一部が自動化されているが人手設計が主体でシミュレーション等設計のチェック (Verification) に計算機が使用される場合が多い。システム的にも独立している場合が多いが、実装設計の情報とのミスマッチを防ぐために変換プログラム等を通して実装設計システムと接続されていることも多い。実装設計及び製造・検査資料作成の段階は古くから設計自動化の対象になっていた部分であり、多くのメーカーで実用化されている。データベースを中心とした統合システムになっている場合が多く、次第に論理設計、ファームウェア設計、診断設計の部分も統合システムの中へ組み込んでゆく傾向が見られる。

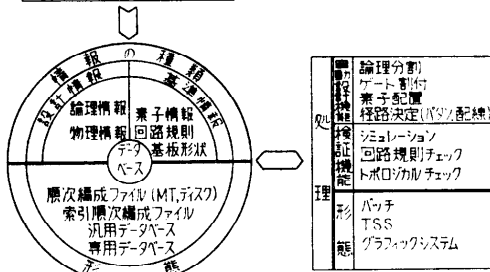
最近発表されている統合された設計自動化システムは図-2 に示す如く四つの部分——入力、データベース、処理、出力——から構成されている。以下夫々についての説明を行う。

2.1 設計自動化システムへの入力

システムへの入力の形はシステムにより種々のものが発表されているが大別して

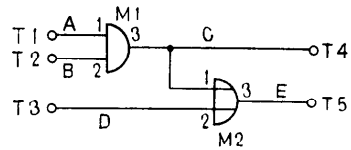
- (a) 論理カード形式
  - 論理式形式
  - 素子形式
  - ネット形式

情報の種類	形	態
入 力	論理カード	カード
	(素子形式)	端末
	(ネット形式)	グラフィックディスプレイ (グラフィクス)
	(論理式形式)	ライター
	(論理回路図)	デザイナー
レイアウト図		



情報の種類	形	態
出 力	論理回路図	LP用紙
	タイムチャート	MT
	堆積リスト	紙テープ
	エラーリスト	カセットテープ
	パッドイメージ	フキキリ/グラフィック
	自動作画テープ	カーブノット
	穴明けテープ	COM
	自動配線テープ	グラフィック
	自動挿入テープ	ディスプレイ
	配線試験テープ	
機能試験テープ		

図-2 設計自動化システムの構成



(a) 回路例

- INPUT A, B, D      INPUT (A, B, D)
- OUTPUT C, E      OUTPUT (C, E)
- C = A \* B          C = AND (A, B)
- E = A \* B + D      E = OR (C, D)
- (b) 論理式形式      (c) 素子形式

- AND M1
- OR M2
- A (T1, M1.1)
- B (T2, M1.2)
- C (M1.3, T4, M2.1)
- D (T3, M2.2)
- E (M2.3, T5)
- (d) ネット形式

図-3 入力形式の例

- (b) 論理回路図形式
- (c) レイアウト図形式

がある。

論理カード形式は回路の論理接続を主体に入力するもので、論理機能を式の形で入力し論理素子への分解の段階から自動化した形式、基本論理素子を単位として入力する形式、逆にネット (信号線) を単位とした形式がある (図-3 参照)。

素子形式とネット形式は同じ情報を着目点を変えて表現したものであるが、論理回路図イメージとの対応が付き易い利点があるため素子形式を採用する場合が多い。設計上の要求によりゲートと IC との対応を入力したり、基板上での配置を部分指定したりすることも可能にしている場合もある。入力情報量を減らすためにアセンブラ言語等で使われるマクロ機能を導入したり、機能ブロック単位で入力して繰り返回路を容易に入力できるようにしたシステム<sup>3,4)</sup>もある。また論理情報や実装情報を必要な程度に混在させて記述出来る設計入力言語を採用したシステム<sup>5)</sup>も報告されている。

論理情報をカードの形に変換して入力する代りに論理回路図のイメージを直接デザイナーで読み取り設計自動化システムへ入力するシステム<sup>6)</sup>も報告されている。このシステムは論理回路図イメージを設計データベース中に記憶しておき実装情報と一致した回路図を自動作成することをも目的としたシステムである。

また人手設計されたプリント基板のレイアウト図をデザイナーを使って入力し、別にカードから入力し

た素子の配置情報とをつき合わせて設計データベースを作り、論理カードから作成したものと同等のチェックを行い設計・製造資料を作るようにしたシステム<sup>7,8)</sup>もある。プリント板の自動作画テープ及びドリルテープを作る単能プログラムにはしばしばレイアウト図を入力とするものが見られるが、論理情報と混ぜ合わせるようにした例は少ない。

## 2.2 データベース

最近の設計自動化システムはデータベースを中心に構成されたものが多く、かつデータベースに記憶される情報は次第に増加する傾向がある。

通常性格の異なる二種類のデータベースにより構成されるが、それは基準情報データベースと設計情報データベースである。前者は設計を進める上で基準になるデータを記憶・登録しているもので、旧世代の設計自動化システムにおいてはプログラムの中に組み込まれていた実装技術に依存するデータを共通ファイルに抽出しプログラムの汎用性を増す目的で採用された。共通ライブラリデータの名で呼ばれることもある。ライブラリデータとして記憶される情報には、

- 使用素子に関する情報
  - 素子名, ピン名称, ピンの機能, ピンごとの電流量, 素子の内部論理構造等
- 回路使用規則
  - ネットの回路規則コード決定法, コードごとの制限及びエラーチェック項目等
- 自動設計時のルール
  - ゲート割付の優先順序等
- 基板形状に関する情報
  - 外形, 素子配置位置, コネクタ位置, 障害物の寸法等

の如く多様である。

後者の設計情報データベースは設計対象の種類ごとに異なる情報を記憶するもので、プリント基板の例をとると

- 配置位置ごとの素子名称
- 接続情報 (論理的及び物理的)
- 基板のパターン情報 (上記接続パターン以外)
- 図面情報

等を含んでいる。

設計情報データベースは通常計算機が IC, プリント板, バックパネル, ユニット (装置) の如く階層構造になっているのに対応して階層的に扱われることが多い。この場合下位のモジュールのデータベースは上

位のモジュールのデータベースにとって素子情報の如き役割を果たす。設計情報データベースはモジュールごとに別の物理的媒体に記憶される場合もあるし、全体を一つの媒体に記憶する場合もある。夫々一長一短があるがローカルにアクセスする場合の方が多く、媒体を分けた方が能率的である場合が多い。

データベースの形態としては順次編成ファイル, 索引順次編成ファイル, 汎用ネットワーク型データベース, 専用ネットワーク型データベース等が用いられる。バッチ処理では順次編成ファイルが能率が良いが, インタラクティブ処理が増えるにつれネットワーク型のデータベースのランダムアクセス機能が要求されるようになるであろう。

## 2.3 処理内容と形態

設計自動化システム中で行われる処理内容は電子計算機の論理機能情報を装置を構成するハードウェアの形に組立ててゆく実装設計が主体であるが, 並行して設計内容のチェックをする検証機能も活用される。設計の進行にともない各機能プログラムが選択的に使用され, また何度もフィードバックが行われるので処理順序には自由度を持たせ, また任意の段階で設計者の人手介入を許している。特に近年 ECL (エミッタ結合論理), CML (電流モード論理) 等の高速論理回路を使用した装置が出現するにつれ, 回路設計規則が複雑になり自動設計のみでは対処しにくくなって来ており, 実装設計の途中の段階での設計者の介入が必要になって来ている。

各処理内容に対する自動設計プログラムは発見的 (Heuristic) な方法とアルゴリズムによる方法とを併用している場合が多い。処理時間は通常前者が早いのでまず前者により処理を行い, 残った比較的難しい処理を後者により完成させるという組み合わせが一般的である。

設計自動化としての処理の種類は実装形態により差があるがプリント板実装方式を採用した計算機を対象とする場合には

- 論理プリント板単位への分割
- ゲートの IC への割付
- IC のプリント板上での配置
- プリントパターンの発生
- バックパネル配線径路決定
- 各種変更機能

等がある。最初の論理分割は SSI (Small Scale Integration) の IC が多かった時は処理し易かったが, MSI,

LSI が主体になると人手に対抗できるような処理が難しく、また診断の見地から機能単位を実装単位と一致させる事が好ましいという事情もあって人手で処理されることが多い。しかし IC 中の集積度が高くなりカスタマイズの LSI が使われるようになるとまた論理分割の自動化が重要になるであろう。

自動設計機能と同様、各種の検証機能もシステムの中で重要な役割を占める。通常

- 論理シミュレーション
- 論理解析
- 回路規則チェック
- 回路遅延時間計算
- プリント板設計規則チェック

等が使われる。これらの具体的内容については 4. 以下で説明する。

設計自動化システムの運用形態としては従来単機能システムを除いては大型計算機によるバッチ処理が中心であったが、エラーチェックや修正機能等ではターンアラウンド時間と使い易さを改善するために TSS (Time Sharing System) 機能も使われるようになっている。

またプリント板の設計や修正のように図形情報が主体となる部分ではグラフィック端末またはフリースタンドイングのグラフィックシステムにより処理を行う例も多い。

## 2.4 設計自動化システムの出力

設計自動化システムからの出力は設計者に設計内容をフィードバックするための資料と製造・検査工程で使用する資料とに分類することが出来る。一部の資料は設計・製造の両工程にわたって使用されるものもあり、更に製品の出荷後保守用資料として使われるものもある。

設計段階用に出力される代表的な資料には

- ピンソートリスト
- ロジックソートリスト
- 設計規則チェックリスト
- 論理回路図
- タイムチャート
- プリントパターンイメージ

等がある。

ピンソートリストは各構成要素ごとにそのピン番号とピン機能、信号名を対にして表現したもので各要素がどの信号と接続されているかを知るのに便利である。ロジックソートリストは

逆に信号線ごとに着目してどの素子に接続されるかを探す時に使用する。配線長や負荷容量等も同時に表示することが多い。設計規則チェックリストは各種の回路規則、設計規則をチェックした結果を表示したもので、通常エラーのある素子あるいは信号線のみを示している。設計対象の論理動作を確認するためには論理シミュレータによるタイムチャートが使われるし、プリント板の配線パターンを確認するためにはパターンイメージリストが使われる。

一方、製造・検査資料としてはリストの形のものや図の形のもの、または製造・検査機械を制御する媒体等があるがこれらについては 7. で説明する。

出力の形態としてはラインプリンタによるリスト、紙テープ等が古くから用いられてきたがカーブプロッタやグラフィック COM による図形出力、磁気テープ、カセットテープ、フレキシブルディスク等の制御媒体も使われている。また結果の一時的な確認の道具としてグラフィックディスプレイの比重も高まっている。

## 3. 設計情報データベース

### 3.1 データベースの発展過程

設計自動化システムにおけるデータベースは急に発生したものではなく、アプリケーションごとに存在した独立のファイルが次第に進化したものとして捕えることができる<sup>9)</sup>。

#### (1) 独立時代 (図-4(a)参照)

各アプリケーションごとに全く独立にデータベースが存在し、相互間には全く連絡がない。従ってデータの重複度による無駄やデータ間の矛盾が問題になった。

#### (2) データベースネットワーク (図-4(b)参照)

重複したデータを別々に入力し、別々に管理する無

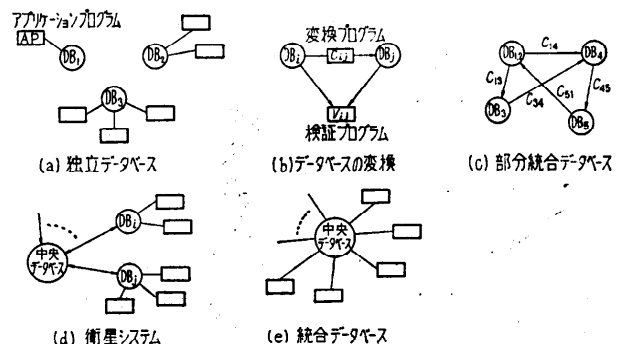


図-4 データベースの発展過程<sup>9)</sup>

駄をさけるためにあるデータベースから別のデータベースへの変換プログラムが作られるようになる。しかし個々のデータベースへ独立な変更を加えることにより、やはり矛盾が生ずるので検証プログラムも同時に作られ統一性を保証した。このようにデータベース間の変換プログラムが作られることによりデータベースのネットワーク関係が発生することになり必要に応じて一つのデータベースから他のデータベースを作ることができるようになった。

### (3) 部分的統合 (図-4(c)参照)

更に経験が積まれると変換の無駄や独立な変更による矛盾の問題が強く認識されるようになり、論理シミュレーションと論理回路図作成システム間のような極めて類似したデータを必要とするシステムとの間で同一データベースが使われるようになった。

### (4) 衛星システム (図-4(d)参照)

部分的統合による効果が認識されるにつれて、更に統合する必要が痛感されるようになり、中心的な新しいデータベースを定義しこれに各アプリケーションが接続される構成を採用するようになった。これは割合少ない手間できかつ従来の問題点がカバーされるので、しばらくの間満足して利用された。

### (5) 統合データベース (図-4(e)参照)

しかし非常に簡単な出力プログラムを実行する場合でも中央データベースと衛星データベースの変換を行う必要があるため、実行費用の経済性が問題になるようになり衛星データベースを対象とするプログラムが直接中央データベースを対象として実行できるよう改造された。この過程を繰り返してデータベースの統合が行われ全てのプログラムが中央データベースから直接実行されるようになった。同時にアプリケーション間で重複した機能を持つプログラムが整備され、統合データベースシステムが確立された。

## 3.2 データの種類

設計情報データベースに記憶される情報は対象となるユニット(プリント板、装置等)に関する全情報が理想であり、次第に種類が増加する傾向にある。それらを大別すると表-1に示すように四種類に分類できる<sup>10)</sup>。

第一のタイプは設計者が直接入力する情報であり通常アプリケーションプログラムによっては変更されない。対象ユニットの論理的記述が主要な内容となる。論理の単純化等を行う場合にはプログラムにより変更されることもある。

表-1 データベース中の情報の種類

情報タイプ	内容	発生源
1	回路の論理的記述 (論理接続関係)	設計者が直接入力。 人手のみにより変更。
2	回路の物理的記述 配線径路、パタン径路、素子の配置、論理回路図イメージ	アプリケーションプログラムにより発生。
3	回路記述以外の情報 テストパタン シミュレーション出力	アプリケーションプログラムにより発生。 データベースに記憶されぬことも多い。
4	各種補助情報 配線材料、パタンサイズ、 説明用コメント	人手入力、自動発生等多様。

第二のタイプは回路の物理的記述であり、通常設計自動化システムの対象としてアプリケーションプログラムにより発生される。この中には素子の配置位置やプリント配線径路、個別配線の径路、論理回路図のイメージ等が存在する。プログラムで発生した後設計者により修正されることもあり、最初から人手で入力されることもある。

第三のタイプもプログラムにより発生されるが、これは回路の構造以外の情報で、検査用のテストパタンとか論理シミュレーションの結果等である。また回路規則コードとかエラーコード等の情報もこの範囲に属する。

最後のタイプは配線の太さやパットサイズ等の製造に関係するデータと設計者の入力した説明用のコメント等である。後者は単に記憶されるだけで処理の対象にはならないが、設計情報の内容を理解する上では重要な役割を有する。

### 3.3 データベースの構成

設計情報データベースの構成方法としては順次編成ファイルによるものが多いが、ネットワーク型のデータベースを採用した例や、リレーショナルデータベースも研究されている。設計自動化システムにおいては設計情報の部分的な検索や修正は比較的少なく全体の情報を一度に処理することが多い。このような場合には順次編成ファイルが有利と考えられている。一方ネットワーク型データベースは論理処理が主体となったシステムに有利と考えられ、階層構造を有する装置をいろいろな規模で取り出し展開して論理シミュレーション等の処理を実行する場合能率が良い。汎用のMIS型データベースを設計自動化に適用した例<sup>11),12)</sup>やDA向きの専用システムを開発した例<sup>2),10),13),14)</sup>が報告されている。MIS型の汎用データベースシステムは部分的な情報の変更や検索に能率が良いように設計され

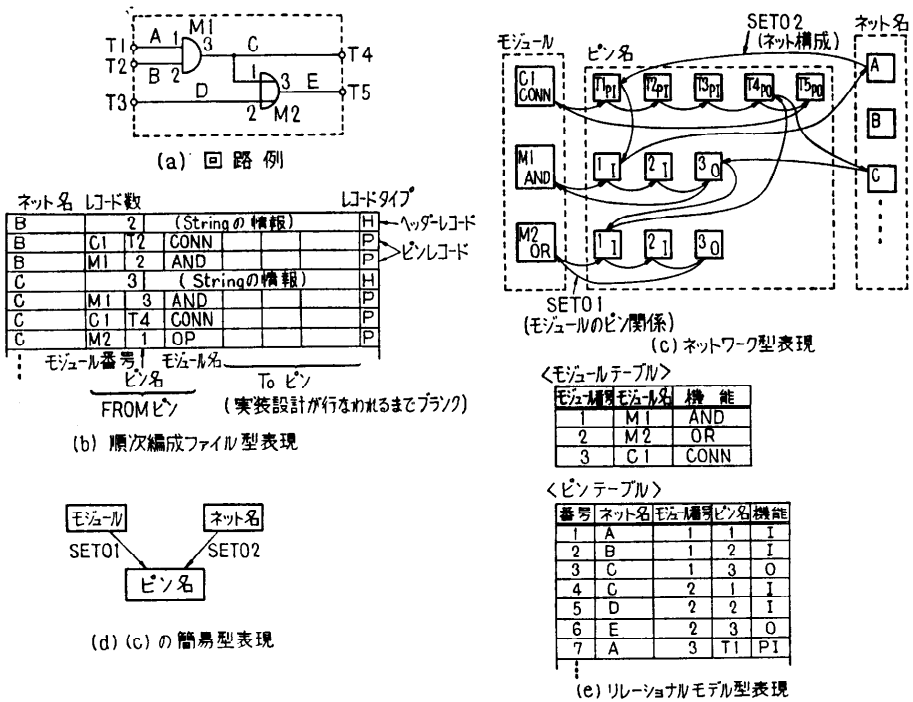


図-5 各種形式による回路表現

ており、前節で述べた第二のタイプに対応する処理についてはあまり考慮されていないため設計自動化システムにとっては能率が良くないという意見もある。リレーショナルデータベースはその理論的な研究が近年さかんに行われており、設計自動化システムへの適用も研究されているが<sup>15,16)</sup>性能に関しては今のところ、未知数である。

各種のデータベースにおける回路の表現の仕方を図-5に示す。(a)はサンプル回路であり、(b)はそれを順次編成ファイルにより表現した例である。ネット名でソートされており、各ネットごとに二種のレコードタイプを有する。ヘッダレコードはネット全体に関する情報を含んでおり、ピンレコードには各ピンのモジュール名、ピン名、モジュール番号等を含む。このピンレコードはFromピンとToピンを含む形式になっており設計の初期の段階ではToピンはブランクになっているが、配線径路またはパターン径路が決るとFrom-Toの形でパターンセグメントを表現することができる。

(c)はネットワーク型のデータベースにおける表現の例であり素子名、ピン名、ネット名等のレコードタイプが定義されており、各レコードタイプ間はリング

構造のポインタで結ばれている。この関係をセットと呼び(d)に示す簡易型表現で簡潔に表現することができる。

同じ回路をリレーショナルモデルを用いて表現したのが(e)である。モジュールテーブルとピンテーブルの二つのテーブルで表現されており、両者を共通のキー(この場合モジュール番号)で結ぶことにより関係づけている。外部記憶装置への記憶方式等については研究段階であるが、ユーザから見たデータ表現は規則的であり理解しやすい。

3.4 データベースへのアクセス

データベースに情報を記憶したり情報の検索を行うためにデータベース管理システムが開発されることがある。順次編成ファイルの場合には割合構造が簡単のためアプリケーションプログラムで直接アクセスするか、割合簡単な入出力ルーチンを通して処理する例が多いが、ネットワーク型データベースの場合には通常データベース管理システムを通してアクセスが行われる。

DA 向き専用データベース<sup>2)</sup>を例にとり管理システムの構成について説明を行う。まずシステムの中心となるのはDMS(Data Management System)であり、

データベースを定義するためのデータ記述言語とアプリケーションプログラムからデータベース中のデータを操作するデータ操作言語を有する。これにより複雑な構造のデータベースを一貫した方法で取り扱うと同時にアプリケーションプログラムの開発者がデータベースの具体的な構造を意識しなくても良いようにする。この他プログラムの実行を監視するDAモニタ、データベースの内容を初期状態に戻すための再編成プログラム、リカバリプログラム等各種のユーティリティプログラムがシステムの効率を高めている。

### 3.5 データベースの問題点

データベースシステムを利用する上での問題点の一つは関連する情報が複数存在するがそれらの一方を変更したとき、いかにして残りを最新状態にするかという点である。入力情報である論理情報を変更する度に実装情報を作り出すプログラムをランするのでは費用が高くなり過ぎる上、変更しない箇所まで影響を受けて変ってしまう可能性がある。また、実装情報の方を変更した場合に対応する論理情報を自動的に変更する能率の良い方法は知られていない。一般的にはこのような問題に対応するため両者の内容に矛盾がないかどうかを調べる検証プログラムが用いられる。

機械の誤動作あるいは操作ミスに対するバックアップの問題も重要である。順次編成ファイルの場合には通常入力ファイルと出力ファイルを別々にして出力ファイルの内容が確認されるまで2~3世代の旧ファイルを残すのが一般的であるが、ネットワーク型データベースの場合には時々バックアップを取りその間の全入力データを記録しておく方法が採られるが、リカバリ操作は簡単でない。

### 参 考 文 献

- 1) P. W. Case et al.: Solid Logic Design Automation, IBM Journal, pp. 127~140 (Apr. 1964).
- 2) 中林 操他: DA のための設計データ管理システム, 情報処理学会設計自動化研究会資料, DA 30-3 (1976).
- 3) 渡会隆志他: 機能ブロックによる論理装置の設計言語の一例, 情報処理学会設計自動化研究会資料, 74-11 (1974).
- 4) M. Umemura et al.: A Digital System Design Language Using Functional Block, Proc. USA-Japan Design Automation Symposium '75, pp. 15~23 (1975).
- 5) 星 元雄他: 論理装置設計のための設計データ記述言語について, 情報処理学会設計自動化研究会資料, DA 30-2 (1976).
- 6) 山田, 杉本: 論理回路図の自動作成, 昭和48年度情報処理学会第14回大会, pp. 21~22 (1974).
- 7) 上田正二他: パターンレイアウトを主入力とする実装 DA システム, 昭和49年度情報処理学会第15回大会, pp. 507-508 (1974).
- 8) A. Inao et al.: Design Automation System for Manually Routed Circuit Board, USA-Japan Design Automation Symposium '75 Proc., pp. 114~118 (1975).
- 9) J. C. Foster: The Evolution of an Integrated Data Base, 12th Design Automation Conf. Proc., pp. 114~118 (1975).
- 10) P. L. Ciampi et al.: Concepts in CAD Data Base Structures, 13th Design Automation Conf. Proc., pp. 290~294 (1976).
- 11) 小島, 田中他: 設計自動化のためのデータベースとそのサポートシステム, 電子通信学会電子計算機研究会資料 EC 72-37 (1972).
- 12) 曾我, 田中他: 電子計算機設計自動化のためのデータベースシステム (EDMS), 情報処理学会設計自動化研究会資料 74-13.
- 13) P. L. Ciampi et al.: Control and Integration of a CAD Data Base, 13th Design Automation Conf. Proc., pp. 285~289 (1976).
- 14) A. J. Korenjak et al.: An Integrated CAD Data Base System, 12th Design Automation Conf. Proc., pp. 399~406 (1975).
- 15) G. Valle: Relational Data Handling Techniques in Integrated Circuit Mask Layout Procedures, 12th Design Automation Conf. Proc., pp. 407~413 (1975).
- 16) 松島, 元岡他: 論理装置設計自動化システム用リレーショナル・データベース, 情報処理学会昭和51年度第17回全国大会論文集 pp. 553~554 (1976).

(昭和51年11月30日受付)