

# 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)

曾我正和・田中千代治・田淵謙也・清尾克彦・国岡美千子・辻裕生

(三菱電機 (株) 計算機製作所)

## 1 まえがき

現在論理装置の設計・製造・検査に計算機の利用は欠かせないものとなっており、設計の各ステップでの自動化の範囲も拡大し、取扱う設計データも飛躍的に増大している。このように種々雑多な設計情報が各ステップで利用され、蓄積されると必然的に重複作業が発生し、スペースと時間の浪費をまねくとともに、誤用・見落としなど作業の効率が著しく低下する傾向が生ずる。これらのDA作業の多様化と運用の煩雑さに基因するロスを少なくする対策には、二つの側面がある。

第1は、各設計ステップに適した設計言語を用いて記述されるデータおよび自動設計サポートを介して、生成されるデータを一括してファイル化し、そのデータを多目的に利用できる形で維持する。いわゆる設計のデータベースを構成することである。第2の点は、運用作業そのものの自動化ないしは操作の簡単化、すなわち作業管理サービスの強化 (DAモータの強化) である。たとえば、プログラム・共通データの管理・データの秘密保全・エラーからの回復・情報媒体管理・実行管理・連続JOBの実行などがこれに相当する。本システムは、実装設計データ・リソース (情報媒体・プログラム) を一括管理するためのデータベースを構成するもので、既存の自動設計プログラムを情報システムとして体系化する試みであり、対象として電子計算機の開発設計における主として実装設計サポートと、関連する製造、検査工程の省力化を計ることを目的としている。また本システムは、計算機には MELCOM-7700 をデータベースの構成には、ネットワーク構造を取扱うことのできるはん用ファイル処理システム MELCOM-DMS(1) を使用している。本文では、この EDMS (Engineering Data Management System) の概要と自動設計プログラムとの関係、各サブシステムの構成と機能について述べ、実施例にもとづく性能について概説する。

## 2 データベースシステム (EDMS) の概要

図 2.1 に示すように、システムは大別して四つのサブシステムと対応するファイル群か

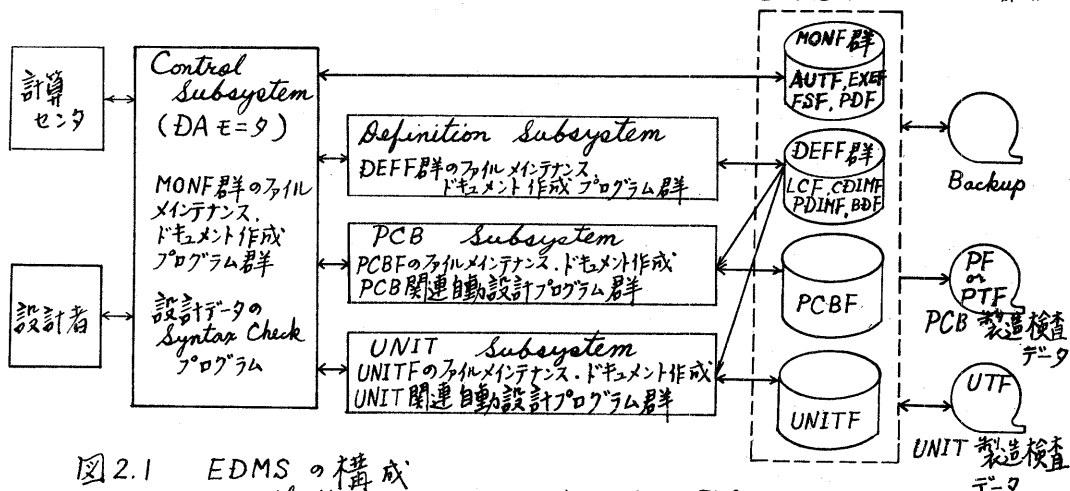


図 2.1 EDMS の構成

AUTF - Authority File    EXEF - Execution File    FSF - Format Spec. File  
 PDF - Program Definition File    LCF - Logic Component File    CDIMF - Comp. Dim. File  
 PDIMF - PCB Dim. File    BDF - Backpanel Dim. File

ら構成される。各サブシステムには、関連する自動設計サポートプログラムが含まれ、全体として論理装置の実装設計自動化システムを構成するが、本章では、主としてデータベースの構成要素であるファイルとその生成、変更処理を行なう部分に注目し、個々の自動設計プログラムには触れない。

## 2.1 データベースの構成

### 2.1.1 用語の定義(1)

(1) ファイル ある論理的構造をもつ閉じたデータ群の集合で、ここで閉じているとは、他のファイルのデータへのポインタを直接持たないことを言い、Keyed Accessは相互に可能であってもよい。ファイルの集合をファイル群という。

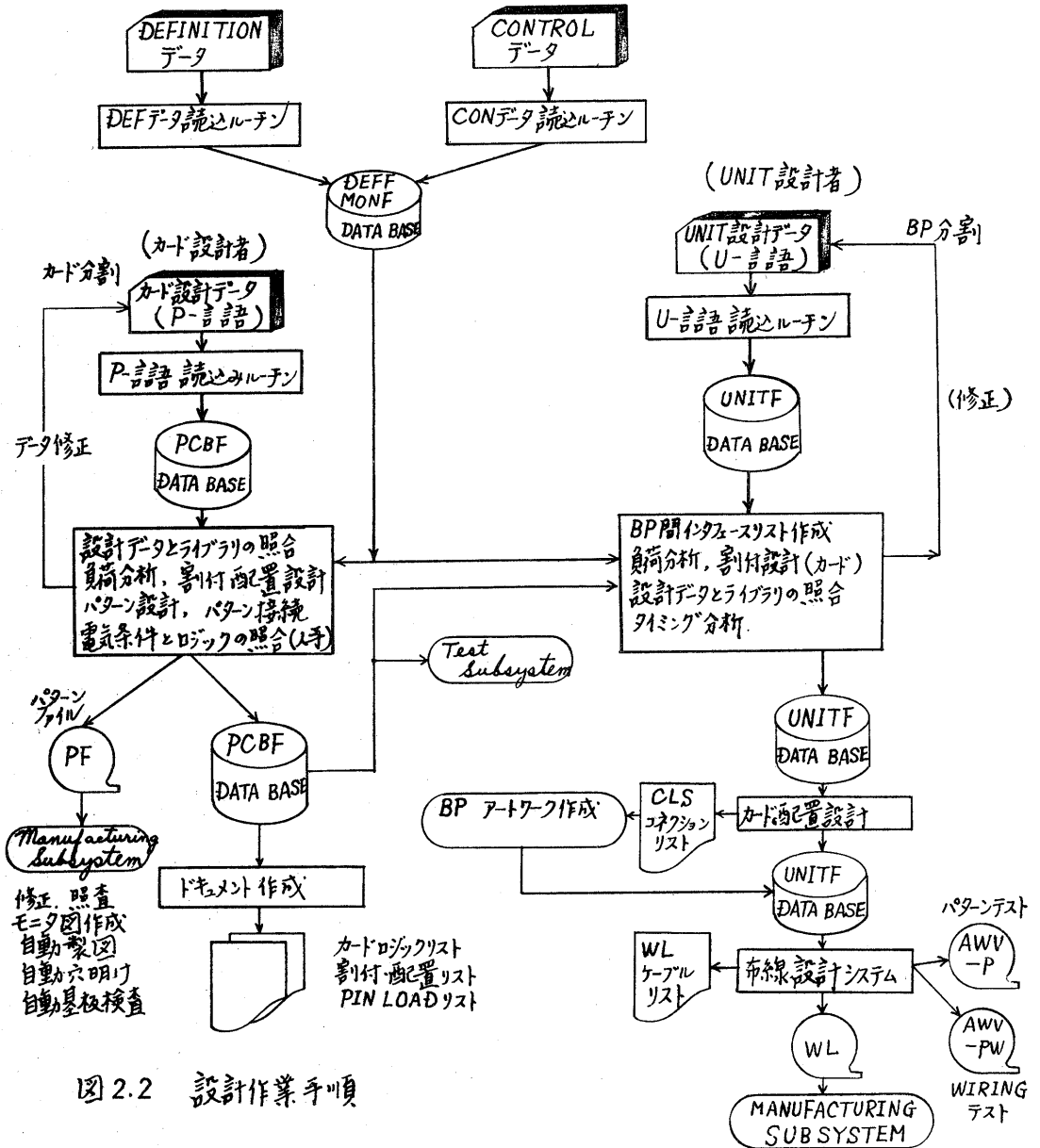


図 2.2 設計作業手順

(2) メンバ 同じ論理的構造（ファイル）で内容の異なるデータ群を、そのファイルのメンバと呼ぶ。

(3) スキーマ データベースを構成するファイルの構造定義ファイルをいう。

(4) サブスキーマ スキーマの論理的サブセットで大きなデータベースの一部のみをアクセスする場合には、その部分を指定するサブスキーマを指定する。

(5) データベース ファイルとスキーマ（サブスキーマ）の一对からなるロジカルファイルをいう。

## 2.1.2 設計データベースの構成

論理装置の設計データは、種々のファイルを含み、設計から製造に至る一貫した流れの中で、次第にデータが増殖してゆく性質をもち、設計の各段階で要求するデータは、その処理内容によって、任意の詳しさ（レベル）で効率よくアクセスできることが要求される。図2.1の構成では、処理対象によって必要とするファイルの部分集合を個別に定義しておき、大きなデータベースの部分領域のみをアクセスするようにしている。こうすることにより、プログラムの作業領域を小さくでき、データ構造の変更に伴うプログラム変更を少なくできる。また、サブシステムの開発の都度、サブスキーマを定義してデータベースを除々に拡大してゆくことができる。

## 2.2 設計手順

図2.2に、論理装置の実装設計を中心とした自動化システムの設計フローを示す。この図からわかるように、現状では自動化システムが実装オリエンテッドな形態のものとなることが多い。この理由としては、次の2点があげられよう。まず論理分割設計の評価基準が複雑であるため、実用に耐える有効な設計アルゴリズムが見つかっていない。このために、分割単位を人手で指定して設計データを記述する必要があるため、図2.2のようにPCB設計フロー・Unit設計フローが分離してしまうことになる。第二の点は、ブール式から使用できる論理素子レパートリに翻訳する論理合成の有効なアルゴリズムが見つかっていないために、論理設計では、使用IC(MSI)に対応した記述から出発せざるを得ない事情がある。

図2.2の設計手順は、次の三つの作業に大別できる。

### 2.2.1 共通データ・管理データの作成

MONFならびにDEFFを実設計作業を始める前に、まず作成しておかなければならない。この作業には、設計者の登録、設計者の実行作業範囲、共通部品ライブラリ、標準実装仕様の登録などが含まれる。

2.2.2 カード設計 カード設計データをもとに、割付設計・基板設計・診断テスト生成を行なう作業で、基板の製造・検査、カードの試験システムへ接続され、必要なドキュメントが作成される。

2.6.3 Unit設計 Unit設計データをもとにバックパネルへの分割・負荷分析・カード配置設計・布線設計を行なう作業で、バックパネルの製造・自動配線・自動配線検査システムに接続される。

## 3. DA モニタ

DA モニタは、DAデータベースシステム（以下DAシステムと略す）の円滑な運営を計り、DAシステムの使いやすさを向上させ、自動設計作業の管理を行なうことをおもな目的としている。DAモニタを中心としたDAシステムの概略を図3.1に示す。

### 3.1 機能

DAモニタは以下の機能を行う。

(1) Authority Check DAシステムを利用す

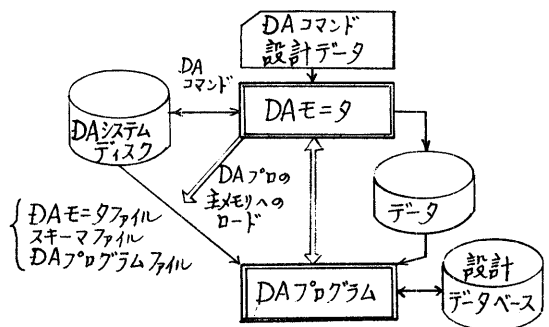


図3.1 DAモニタの概念図

る使用者が、あらかじめ登録されているかどうか、および使用者の実行が、あらかじめ登録されたDAシステムにおける実行許可範囲内にあるかどうかのチェックを行なう。これは使用者の勝手な使用を禁止し、DAシステムのプロテクションおよび秘密保持のために行なわれる。

(2) Catalogued Procedure DAプログラムの実行を指示し、実行時に必要な各種情報を与えるDAコマンドの組を一つのプロシージャとして定義し、実行時にDAコマンドでそのプロシージャを指定するだけで、正しくDAシステムの実行を行なうことができる。定義されたプロシージャの内容は、実行時に任意に修正することができる。

これによりDAコマンドの量は大幅に減少し、使用者によるつまらないミスを除くことができる。

(3) 実行管理 DAモニタのもとで実行される複数のプロシージャは、それぞれのプロシージャで指定される優先度にしたがって実行される。この優先度はDAシステムのサブシステムごとに定められており、たとえば、Definition サブシステムのプロシージャは、PCB サブシステムのプロシージャより先に実行されるようになっている。また、各DAプログラムの実行は、以前に行なわれたDAプログラムの結果（発生エラー数・最高エラーレベル）が、DAコマンドで指定した値よりも小さい場合にのみ行なわれる。

(4) 入力ストリームの編集 入力ストリーム中の任意の位置にあらかじめファイルに登録されたデータをそう（挿）入したり、入力ストリーム中の任意の一部をファイルに登録することができる。

(5) DAシステムの管理情報の保持 DAモニタは3.2節のDAモニタファイルにより、各使用者・各プロシージャの使用実績と、各プロシージャの実行によるDAモニタの履歴情報を保持する。これらの情報は各種管理・運営資料となる。

### 3.2 DAモニタ

DAモニタは、その実行にあたって、次の四つのDMSによるファイルを必要とする。

(1) Authority ファイル DAシステムを利用する使用者に関する情報（アカウント番号・マンナンバ・氏名・所属部課など）、その使用者が実行を許可されたDAシステム内の範囲（サブシステム、プロシージャなど）に関する情報、および各使用者の使用実績に関する情報（登録日、最近実行日・実行回数・総実行時間）を保持する。

Authority Check および各使用者の使用実績は（把握）のため使用される。

(2) Execution ファイル DAコマンドにより実行時にDAモニタにより作成され、DAモニタの実行を制御する。このファイルは実行終了後もそのまま残され、DAシステムの履歴情報となる。

(3) Format Definition ファイル

DAコマンドのフォーマットの定義情報を保持し、DAコマンドのシンタックスチェックに使用される。また他のDAプログラムの入力データに対する定義情報も保持し、それらに対しても統一的にシンタックスチェックが行なわれる。

(4) Program Definition ファイル DAシステムで利用されるプロシージャに関する定義情報（一連のDAコマンド、所属サブシステム、作成者など）と、そのプロシージャの使用実績に関する情報（登録日・最近実行日・総実行回数）を保持する。Catalogued Procedure機能とプロシージャの使用実績は握のため使用される。

### 3.3 DAコマンド

DAモニタの実行は、すべて以下のDAコマンドによって制御される。

(1) SYS コマンド DAシステムディスクの定義を行なう。

(2) EXEC コマンド 実行すべきプロシージャを指定し、使用者のアカウント番号・マンナンバなどを与える。

(3) STEP コマンド DAプログラムの実行を指示し、実行打切時間・実行打切エラー数・実行打切エラーレベルなどを与えることができる。

(4) DCB コマンド DAプログラムで使用するI/Oに関する情報を与える。

(5) COND コマンド DAプログラムを実行する場合の条件（以前に実行されたDAプログラムのエラー数あるいは最大エラーレベルが指定した値より小さい）を与える。

(6) DATA コマンド 以下にデータが続くことを示す。

(7) INCLコマンド 入力ストリーム中のこの部分に、指定したファイルの内容を組み込むことを指示する。

(8) FILE, EOFコマンド 両者のコマンドには含まれた入力ストリームの部分を指定したファイルに出力することを指示する。

(9) ENDコマンド データの区切りを示す。

#### 4. Definition サブシステム

Definition サブシステムは、論理実装設計において共通に使用される標準データ（基本論理要素定義データ、部品定義データ、部品・カード構造データなど）をファイルに登録し、その管理を行なうサブシステムである。共通データをファイルとして管理することにより、入力（特に重複）作業は軽減し、データミスも減少し、さらに異なった機種間で共用できるために標準化の促進をはかることができる。また共通データの改訂による波及効果をは握することが容易になる。図4.1にDefinitionサブシステムの概略を示す。

##### 4.1 構成ファイル

Definition サブシステムを構成するおもなファイルは以下のもので、それぞれのファイルには、メンテナンスプログラムおよびドキュメントプログラムが用意されている。

(1) Logic Component ファイル (LCF) AND, ORなどの論理要素 (LE), 100Ωの抵抗等の非論理要素 (NLE), LEを実装するIC・MSIなどの論理部品 (LO), および NLE を実装する抵抗などの非論理部品 (NLO)の電気的特性（供給電圧・消費電力・平均遅延時間など）、物理的特性（形状など）および論理的特性（論理式など）に関する情報を保持したライブラリファイルである。これらの情報は、LE (NLE)のLC (NLC)への自動割付プログラム、論理シミュレータ・カード機能検査のための故障診断プログラム、負荷分析プログラムおよびカード基板設計プログラムなどに供給される。

(2) PCB Dimension ファイル (PCBDF) カードの構造情報（外形寸法、パターン領域、基本格子、電源・グランド信号パターン、共通孔、IC配列間隔など）を保持し、カード基板設計プログラムにそのデータを供給する。

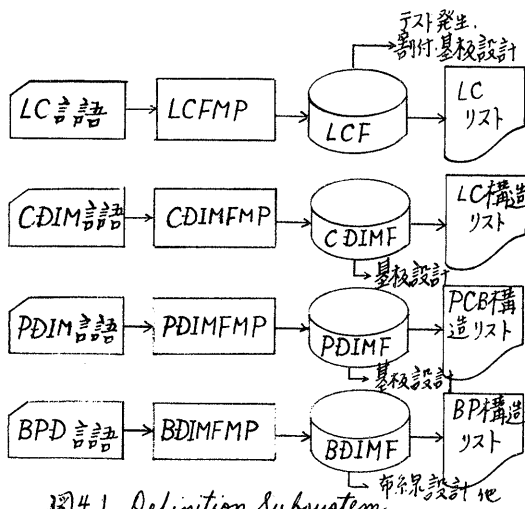


図4.1 Definition Subsystemの概略

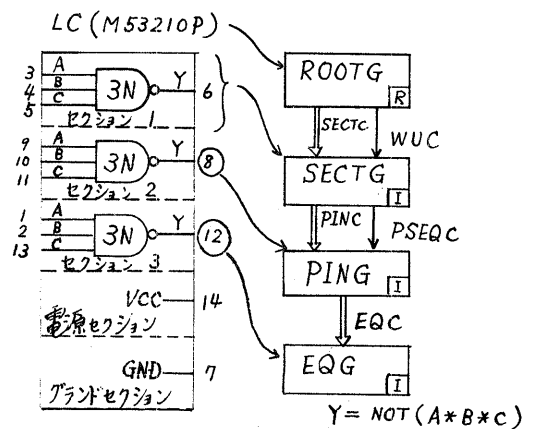


図4.2 LCファイルの構造

(3) Back Panel Dimension ファイル (BPDF) バックパネルの構造情報 (外形寸法, パターン領域, 基本格子, 電源・グランド信号パターン, 共通孔, カード配列間隔, 設列情報など) を保持し布線設計プログラムにそのデータを供給する。

4.2 Logic Component ファイル LCファイルはDMSによるファイルで, その構造と保持されるモジュール (LE, NLE, LC, NLCからなる) との関係を図4.2に示す。以下LCファイルのおもなグループとチェーンについて説明を行なう。

- (1) Root グループ (ROOTG) モジュールを定義し, そのモジュールに関する情報 (略名・製作メーカー・外形・タイプ・消費電力・平均遅延時間・部品コードなど) を保持している。
- (2) Section グループ (SECTG) モジュール内の分割可能な回路部分 (セクション) を定義し, そのセクションに関する情報 (回路番号・回路タイプなど) を保持する。LE (NLE) のLC (NLC) への割付けはセクション単位に行なわれる。
- (3) Pin グループ (PING) セクションに含まれる端子を定義し, その端子に関する情報 (端子名, 端子番号, 端子クラス, ファン・イン/アウト, オープン時の論理値, PCB言語のLEデータ上での端子順序など) を保持する。
- (4) Equation グループ (EQG) 端子間の論理関係を論理式で定義し, 出力端子のもとに生成される。
- (5) Section チェーン (SECTC) モジュールに含まれるセクションをグループ化する。
- (6) Where Used チェイン (WUC) LE (NLE) を含む LC (NLC) をグループ化する。

### 5. PCB サブシステム

PCBファイルは, プリントカードについての内部論理情報および実装情報を保持しているファイルであり, P言語と呼ばれる入力言語で記述されるPCB設計データにより, メンテナンスされる。PCB設計データには create と update の二つのモードがあり, P言語は update 指示も create と同じ形式で記述することができる。PCBの負荷分析・割付設計・配置設計・パターン設計などを行なう各DAプログラムは, PCBファイルから必要なデータを受けとる。また, PCBファイルは, 故障診断サブシステムへデータを供給する。言語の一般形については, 文献(4)に述べられているので, ここでは具体例をあげて説明する。

#### 5.1 P言語 (PCB設計データ)

PCB設計データは, PCB単位に論理分割された後に, 各カードごとに作成される。P言語は, PCB設計データを記述するための言語であり, 次のような機能を持っている。

- (1) カード内の論理分割 (セクション) の指定 (SECCARD)
- (2) 論理素子 (LE) の宣言と接続関係の記述 (LECARD)

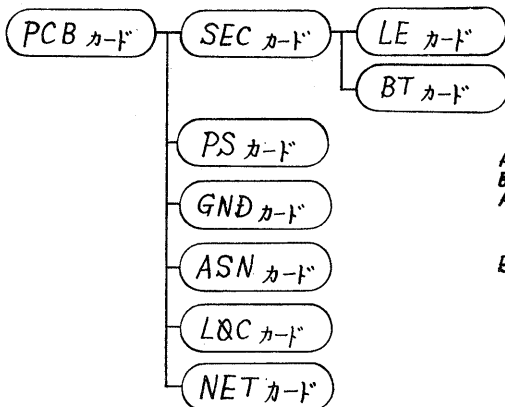


図5.1 P-言語の階層構造

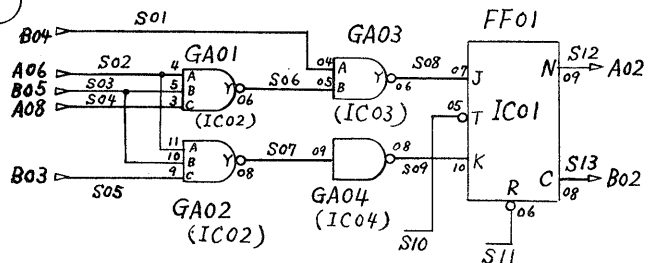


図5.2 カード接続区の例 (カードX)

# MELCOM INPUT DATA FORM

図 5.3 P言語による記述例

	0	1	2	3	4	5
	1	2	3	4	5	6
1	A	PCB=X	TYPE=PT04		PCB	カード
2	A	SEC=1			SEC	カード
3	A	3N	GA01	S06	S02	S03
4	A	3N	GA02	S07	S02	S03
5	A	2N	GA03	S08	S01	S06
6	A	1N	GA04	S09	S07	
7	A	LE=(DJKF,FF01)	J=S08	T.=S10	K=S09	
8	/		R.=S11	N=S12	C=S13	
9	A	BT=(C&N,)	B04=S01	A06=S02	B05=S03	BT
10	/		A08=S04	B03=S05	A02=S12	B02=S13
11	A	ASN	GA01	M210	IC02	2
12	A	ASN	GA02	M210	IC02	3
13	A	ASN	GA03	M200	IC03	2
14	A	ASN	GA04	M204	IC04	4
15	A	ASN	FF01	M273	IC01	2
16	A	LOC	IC01=D01	IC02=B01	IC03=A01	LOC
17	/		IC04=A02			
18	A	PS=(5,P5V)	B01=(C&N,)	B24=(C&N,)		PS
19	A	GND=GND	A01=(C&N,)	A24=(C&N,)		GND
20						

- (3) 境界端子 (BT) の宣言と接続関係の記述 (BTカード)
- (4) 電源信号の指定と接続関係の記述 (PSカード)
- (5) GND信号の指定と接続関係の記述 (GNDカード)
- (6) 接続関係の変更指示 (NETカード)
- (7) 人手による実装条件の指定
  - 部品配置の指定 (LOCカード)
  - 論理素子の実装素子への割付指定 (ASNカード)

以上の8種類のカードのほか、PCBカードがあり、これは一組のPCB設計データの最初に必ず置かれ、そのPCBの登録・検索・削除を指示する。P言語を構成するこれらのtransactionカードは、図5.1に示すような階層構造をしている。図5.3にP言語によるPCB設計データの記述例を示す。これは、図5.2のXカードについて記述したものである。

- 1 ; X というカードを PCBファイルに登録する。
- 2 ; 以下の LE と BT は セクション1 に属することを示す。
- 3 ; LE, GA01 は 3NAND で出力信号が S06, 入力信号が, S02, S03, S04 であることを示す。  
LEカードには, free format (3~6) と alignment format (7) の2種類の形式がある。
- 9 ; B04 はコネクタ端子であり信号 S01 に接続することを示す。
- 11 ; GA01 という LE を, IC02 の第2回路に割付けることを指示する。M210 は, ICタイプをあらわす mnemonic である。
- 16 ; IC01 という IC を D01 の位置に配置することを指示する。
- 18 ; B01 と B24 はコネクタ端子で, 電源 (電源値 5V) に接続していることを示す。P5V は default 信号名である。

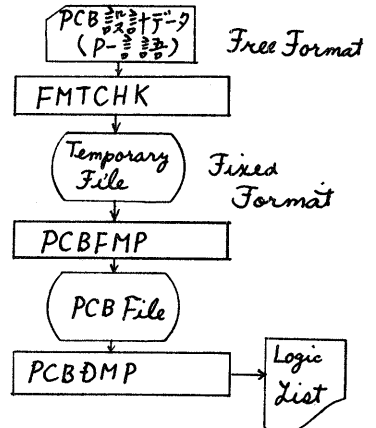


図 5.4 プロシージャ 'PCBFMP'

19 ; A01 と A24 はコネクタ端子で、GND に接続していることを示す。GND は、default 信号名である。

### 5.3 使用方法と実施例

図 5.4 に示すように、プロセス P C B F M P (P C B F i l e M a i n t e n a n c e P r o g r a m) は三つのステップから成っている。

(1) ステップ1 : P 言語で記述された PCB 設計データのフォーマットチェックを行ない、free format から fixed format へ変換する。

(2) ステップ2 : P 言語を解説して PCB ファイルのメンテナンスを行なう。

(3) ステップ3 : ロジックリストを出力する。

### 6 Unit サブシステム

Unit サブシステムは、装置 (Unit) 単位に、各バックパネルの設計補助、設計自動化さらに製造検査工程の自動化・省力化を計るために使用される一群の DA プログラム、およびファイルメンテナンスプログラムの集合からなり、図 2.2 に示す手順で使用される。以下 Unit サブシステムの機能について概説する。

(1) 設計データの維持 Unit 設計は、バックパネルへの論理分割を手で行なうことを前提とし、論理設計結果を、設計済みの機能カードの外部コネクタ端子から見た相互接続情報として表現する (U-言語)。またははん用カードで実現する論理を、未割付けのままで記述し、あらかじめ設計されているはん用カードへの割付けは計算機にまかせることもできる。このとき未割付けの論理ブロック (回路 Section) はカードレポート (PCB ファイル) に定義されているものでなければならない。

この初期設計データの登録およびファイルの修正を U-言語を用いて行ない、図訂履歴を設計者の指示により保存することができる。

(2) バックパネル (BP) 単位への分割設計 バックパネルへの分割設計サポートとして、仮分割の行なわれた設計結果から、同一 Unit の BP 間インタフェース信号を抽出し、リストを出力する。図 2.2 に示すように、このリストを見ながら、分割を変更してはリストを印刷し、妥当なインタフェースケーブル数と BP 間インタフェース信号を定める。BP 間インタフェース信号の定まった時点で、その信号をインタフェースピンに割付けることにより、インタフェース設計を終了する。

(3) 自動割付け設計 はん用カードを使用するとき、特定のカード中の回路 Section を指定しないで、設計結果を記述した場合に、割付けを線長の減少、カード数の減少という人手設計に準じた基準にもとづき設計する。

(4) 負荷分析 設計結果の直流的な負荷のバランス (Fan In/Out) を、Unit 設計のレベルにおいてチェックし、Fan Out Over の信号についてはエラー表示を行なう。

(5) カード配置設計 (MAPSBP) バックパネル単位にカードピン間の相互接続情報が与えられているとき、カードの最適配置を、信号に重みづけを許した相互接続線長を最短にすることを設計基準として求める。同一 BP 内でサイズの異なるカードの混在する場合や、カードの集合を特定のブロック (複数の位置) に、最適配置する問題にも適用することが可能である。(位置のあらかじめ限定されたインタフェースコネクタの最適配置など)。

(6) 布線設計 IC の高速化に対応して、設計上考慮すべき要因が増加した。布線設計品質の向上を計るために、新たに考慮している要因には、下記のものがある。

(a) 特定の信号間干渉 (平行線長) を計算し、より線を少なくするように設計しより線のリターンワイヤの接続指示も行なう。

(b) 布線密度は均一になるようにし、布線条件によってバックパネル上に終端回路



を自動的に選択してそう入できる。

(c) 図訂に伴う再設計計算は、信号ストリングごとに再計算し、変更リストを出力できる。

(d) 布線設計上考慮すべき、カード内パラメータは、自動的に組込まれる。

(7) ドキュメントの作成 (1)～(6)で述べたDAプログラムの生成する設計、製造、検査資料も含めてUnitファイルから次の資料を作成する。

(a) Unit単位のロジックリストおよび割付けリスト

(b) カード配置図およびBP間インタフェースリスト

(c) PIN LOAD リストおよび布線表(WL)

(d) 検査(AWV・・・Automatic Wiring Verifier) テープ

(e) 配線(AWP・・・Automatic Wiring Positioner)紙テープ

## 7. 実施例の評価

データベースシステムの性能は、ただ単に所要記憶容量、メンテナンスの実行時間だけで評価されるべきものではなく、データの一元化による入力(重複)作業の軽減化、データの修正による図訂管理・波及効果測定、データの同時性、プロテクション機能、システムの拡張性、運用形態などの総合的な面から評価されるべきものである。しかしシステムをこれらの面から定量的に評価することは困難なので、ここでは、PCBファイルに関して一つの側面(所要記憶容量・メンテナンス時間・バッファサイズなど)から、評価を行なってみた。

### 7.1 PCBファイル

いろいろなPCBデータに対するファイル生成と検索についての実行結果を、表7.1に示す。

これらの結果により、

(1) ファイル生成は、ネットワーク構造のチェーン生成のために実行時間が長く、ほぼ所

表7.1 PCB設計データの実行例

比較項目	サンプル	シングルカード		ダブルカード		トリプルカード
		A	B	C	D	E
ゲート数		42	58	66	128	210
実装素子数		15	16	31	40	57
端子数 (PIN数)		220	240	423	734	841
信号数		65	88	151	166	≈300
グループ数		357	417	686	1,080	≈1420
所要記憶容量 (KW)		5.3	6.0	10.0	16.5	≈20.8
ファイル生成	Execution Time (min)	2.343	2.802	5.487	11.011	20.977
	I/O Time (min)	1.149	0.565	2.695	10.166	19.823
	Overhead Time (min)	0.586	0.536	1.173	3.087	6.422
	Total Time (min)	4.078	3.903	9.355	24.264	47.222
	ディスクアクセス回数	3373	1845	6784	25664	55122
ファイル検索	Execution time (min)	0.478	0.490	0.825	1.068	1.881
	I/O time (min)	0.069	0.081	0.158	0.185	0.578
	Overhead time (min)	0.143	0.140	0.246	0.330	0.551
	Total time (min)	0.690	0.711	1.229	1.583	3.010
	ディスクアクセス回数	274	293	461	488	1342

要記憶容量の2乗に比例している。

(2) ファイル検索は、チェーンをシーケンシャルに追うのでファイル生成に比べ非常に高速である。またファイル修正も比較的高速に行なわれる。

(3) 所要記憶容量がバッファサイズを越すに従い、ディスクアクセスの回数が増大し実行時間は長くなるということがわかる。

PCBファイルは、カード設計者により論理設計の最初に一度生成され、それ以後は設計作業の進展にともない、修正と実装情報の付加が行なわれ、しだいに増殖されていく。このようなことからファイル生成時間は、幾分長すぎる気はするがほぼ満足の結果である。

7.2 全体の評価と問題点 本システムを試験的に運用した結果、従来のデータベースのDAシステムに比べて、次のようなシステムとしての利点が確認された。

(1) 設計データの一元的管理により、設計者の入力データ作成作業およびミスが大幅に減少した。

(2) 設計データの変更・検索が高速である。

(3) データベースは、設計データの修正に対し常にその最新情報を保持し、設計作業との同時性を保つことができる。

(4) Definitionサブシステムのファイルのように、各機種間の共通データをライブラリとして保持することにより、データ量の削減、標準化の促進を促すことができる。

(5) DAモニタの採用によりシステムの使いやすさを向上させ、運営・作業管理を容易にした。

(6) DMSといはん用データベースシステムを採用したことにより、DAプログラムの開発工期が短縮し、また各種の要求、新技術への対応が容易になった。

(7) 設計言語のフォーマットの統一により、設計者の設計データの作成は容易になりデータのシンタックスチェックを、一つのプログラムで集中的に行なうことができるようになった。

しかしまだまだ解決されねばならない問題が山積されている。以下におもな問題点を列挙する。

(1) データの一元化を徹底的に押し進めた結果、設計データのファイルの構造が複雑なネットワーク構造となり、I/Oおよびオーバーヘッド時間が増大した。

(2) プログラムの虫、データの誤り、オペレーションミスなどによるデータベースの破壊は必ず存在するので、バックアップ体制およびリカバリ体制を十分に確立しておかなければならない。

(3) ライブラリの性格を持つデータは各機種間で共用されるために、使用部品・命名法などについて十分な標準化がなされねばならない。

(4) DAシステムは、現在のようなカードベースによるバッチ処理では限界があり、今後はCRTディスプレイなどによる人と計算機の会話形処理を採用する必要がある。

(5) 本システムでははん用データベースシステム(DMS)を採用したが、サブスキーマ機能が完全でないなど、まだまだ問題があり今後の改良が待たれる。

(6) 情報媒体管理に細心の注意を要すこと。(特にオペレータ間の連絡を厳密に行う必要があること。)

## 8. むすび

DAデータベース確立のため、これまでいろいろお力添えをいただいたかたがたに謝意を表する次第です。

## 参考文献

(1) M-7500/7700 データマネージメントシステム(DMS) 説明書

(2) 三上, 田中, 田淵, 清尾: 論理装置設計自動化のためのデータ構造, 昭47信学全大, 87-6

(3) 論理装置の設計・製造自動化専門委員会: 電気学会技術報告, ([部] Na 94, 1 (昭45))

(4) 小島, 田中, 田淵, 清尾, 岡岡: 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  
昭47電算研資EC72-37