

OLIMPIC - LSI パターン修正システム

吉田憲司 倉持矩忠
(東京芝浦電気(株)総合研究所)

1. まえがき

最近、卓上計算機用 IC を始めとして、数千素子よりなる大規模集積回路(いわゆる LSI) が広く用いられるようになってきた。これとともに集積回路の設計に要する時間と労力もぼう大なものとなり、CAD (Computer Aided Design) システムに対する要求が一段と強まってきた。これに対してさまざまなアプローチによる CAD システムが開発されているが、現実問題としては、設計結果の質や経済性を考えるとその適用範囲は限られており、人手による設計もまだまだ多く用いられている。

一般に集積回路(IC)の設計(特にパターン設計)とは IC の製造工程中の写真蝕刻に用いるマスクの設計を意味するが、このマスクは 6~9 枚で一組となっており、それぞれ数千~数万個の多角形パターンを含むものである。このマスクを製作するには自動製図機により精密なパターンを作図することが必要であるが、IC の CAD システムの出力はこの製図機の入力データである。また人手設計の場合も設計図よりデジタルタイザで座標を読みとり、製図機入力データに変換されるのが普通である。ここで報告する OLIMPIC (On-Line Inspection and Modification of Pattern Data for Integrated Circuits) システムは、CAD であれ人手設計であれ、設計が一旦終了し数値データに変換されたパターンを検査・修正することを目的とするシステムである。機能を限定しているため、複雑で大規模ないわゆる CAD システムに比べると、開発コストも安く、応答も速く、使いやすい実用システムである。

このシステムでは、マン・マシン・インターフェースとして、リフレッシュ型の CRT ディスプレイとライトペンを用いているが、システム設計上最も問題となったのは扱うデータ量が多いこととハードウェアから来る各種の制限である。扱うデータが多いので外部記憶装置を使わざるを得ないが、これによりシステムの応答速度が遅くなりがちである。また、CRT ディスプレイの機能は LSI のような大量の図形を扱うのには十分というにはほど遠い。しかし、このような制限にもかかわらず、それらをよく理解した上で、人間の役割と機械の役割をバランスよくわり分けることにより、実用的なシステムを設計することは可能である。このようなマン・マシン・システムの一例として OLIMPIC システムを紹介する。

2. システムの概要

図 1 は OLIMPIC システムを使った、マスクの設計・製図工程を表わしている。CAD システムにより自動設計されたものも、人手設計の後デジタルタイザで数値データに変換されたものも同一形式の磁気テープ(これをマスター・ファイルと呼ぶ)に変換される。このマスター・ファイルはさらにそれぞれ適当なデータ変換を行なって、カーブ・プロッタやマスク作成用の高精度作図機を駆動するのに用いられる。OLIMPIC システムが検査・修正する対象はこのマスター・ファイルである。IC マスク・パターン用ということから、取り扱う図形は座標軸に平

行な線分よりなる直角多角形と座標軸に対して斜めの矩形の2種類に限定することができる。システムに必要な機能としては、基本的には、

- (i) ファイルへの入出力。
 - (ii) パターンの任意部分を任意倍率でCRT上に表示すること。
 - (iii) 図形をタイプライタまたはカードから追加入力すること。
 - (iv) 図形の削除、位置修正、複製、一部の变形などの修正。
- などである。

システムの設計目標としては、

- (i) 大規模なLSIのパターンも処理できること。
- (ii) 応答時間が短いこと。
- (iii) 操作が簡単であること。
- (iv) 在来システムとの compatibility があること。
- (v) システムの機能拡張が容易であること。

などを重要と考えた。(i)(ii)は相反する要求であるが、このシステムでは一つのLSIをいくつかのセクションに分け、各修正作業はすべてこのセクションごとにとまとめて行なうこととし、データ構造もそれに適するように設計することによって、これらの問題を解決した。

使用計算機は TOSBAC-3400/M41 (96K語) で、通常の周辺装置の他にマン・マシン・インターフェースとして DDS-300 グラフィック・ディスプレイ装置を用いている。DDS-300 は4K語のリフレッシュ用バッファメモリ、16inのCRT、ライトペン、ファンクション・キー・ボードおよびタイプライタを持っている。図2には、これらの間の関係がデータの流れるとともに示されている。

3. セクション分けとファイル構成

マスター・ファイルには一つのLSIのための全マスク・パターンが含まれており、そのデータは大きな記憶領域を必要とするので外部メモリを利用しなければならない。システムの応答時間を短くするためには、コアと外部メモリのデータのやりとりを極力少くしなければならないが、そのためには、システム使用中のある期間内に必要とするデータの範囲がなるべく限定されるようにシステムの外部仕様を決め、またデータ構造としてはそのようなデータの集合がまとめて記憶されるように考慮を払うことが重要である。

このようなデータの集合、あるいは処理の単位としては、一枚のマスクの、しかもある座標範囲内にあるパターン群とすることができる。一般に手設計によるマスク図面は何枚かのA1方眼紙に分けて書かれ、すべての図形はその方眼紙内部で閉じている。したがって処理単位として設計図と同じセクション分けを用いると、セクション分けによる不便はあまりない。

実際の操作としては、まず一つのセクション(方眼紙)とマスク番号を指定し

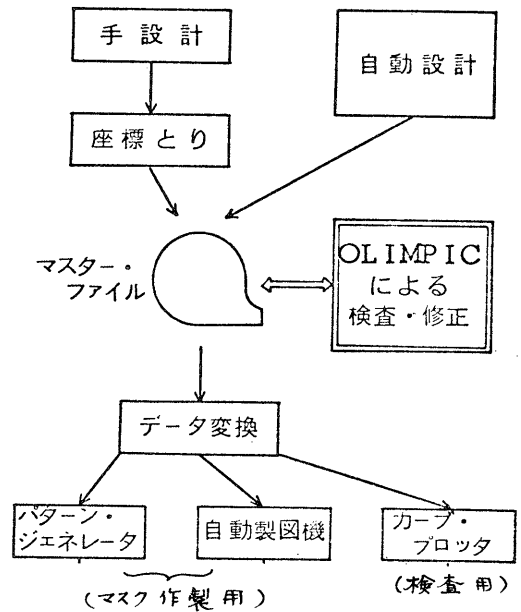


図1 マスク作成工程

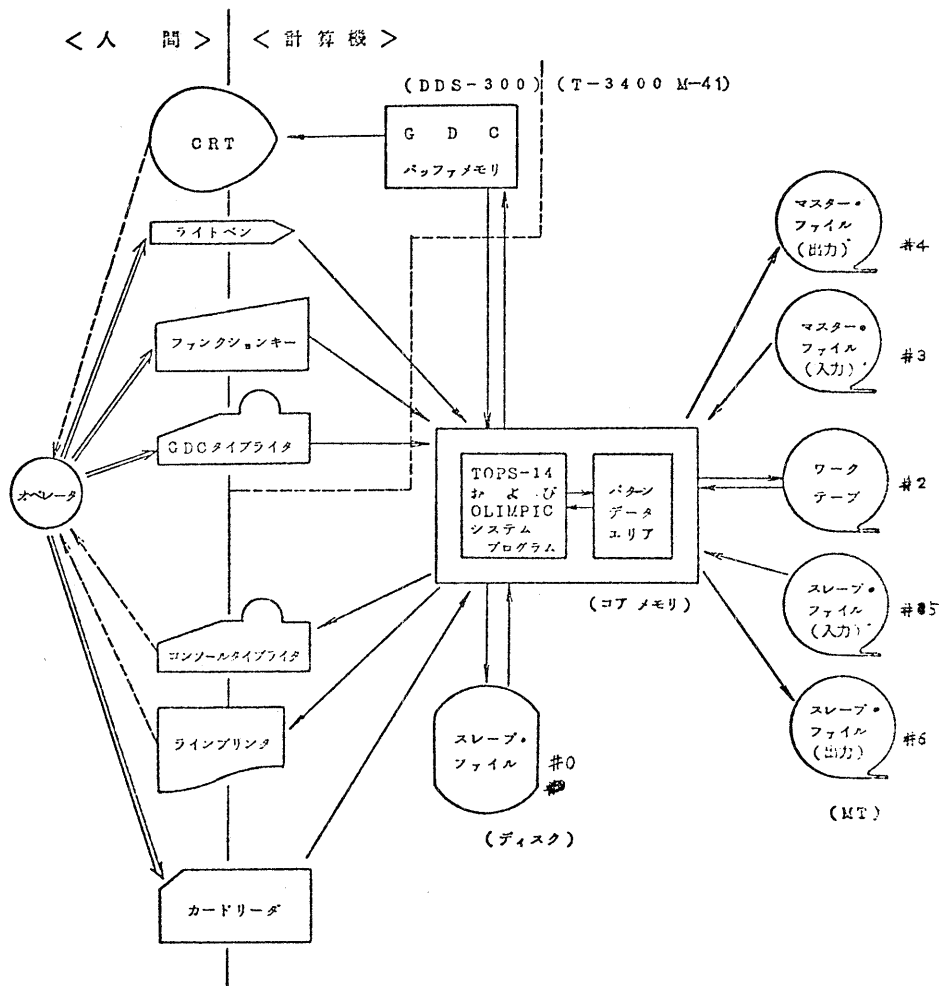


図 2 使用ハードウェアとデータの流れ

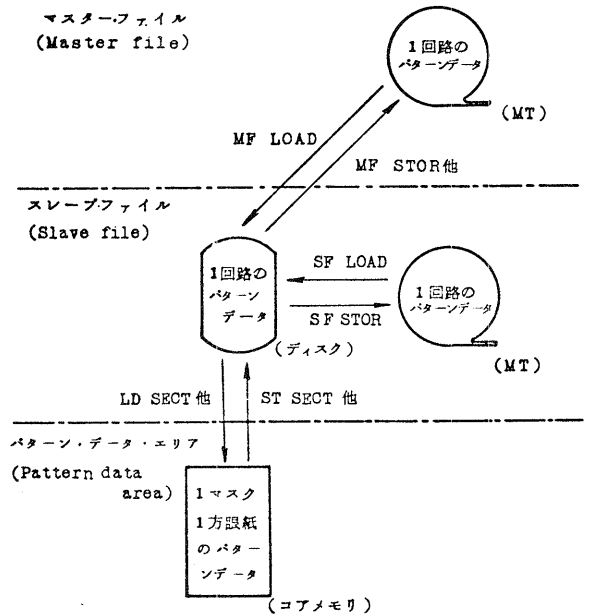


図 3 ファイルとパターンデータエリアの関係

そのパターンを表示してから個々の修正を行なうことになる。この指定をした時に外部メモリ（ディスク）よりコアにデータを転送し、個々の修正はコア内部で処理する。このセクション・マスク分のデータを保存するコア内の領域をパターン・データ・エリアと呼んでいる。

マスター・ファイル（磁気テープ）には、一般にランダムな順序でデータが入っているので、マスター・ファイルを読み込む時に、マスク毎、方眼紙毎に分類し、本システムの処理に便利な形で保存する必要がある。これをスレーブ・ファイルと呼び、磁気ディスクの中に定義する。図3は以上のファイルとパターン・データ・エリアの関係を示している。

このようなファイルの構造は、個々の修正の時間を短くしているばかりでなく、修正ミス・操作ミスによる時間の損失を少なくするのにも役立つ。すなわち、個々の修正処理はパターン・データ・エリアのデータに限られ、スレーブ・ファイルにまで及ばないのでスレーブ・ファイルには一段階古いデータが入っていることになる。これをパターン・データ・エリアに揃えて来ることにより、システムを一段階古い状態にもどすことができる。

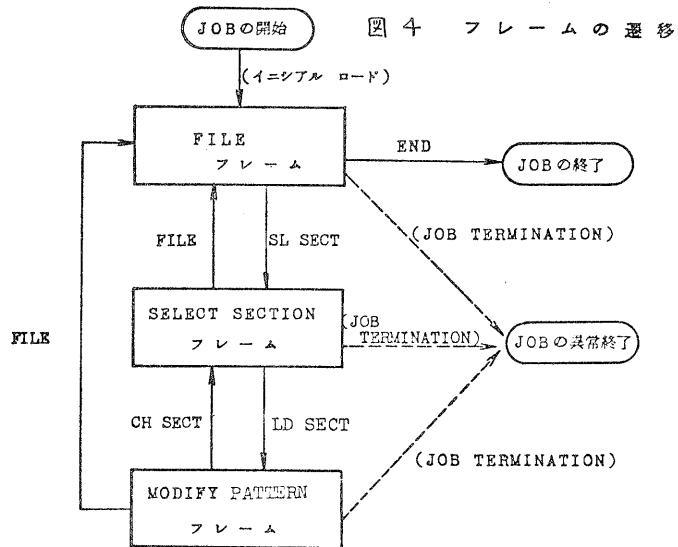
4. 人間-機械の対話のしかた

2. で述べたような修正機能を使うために3種類の画面が用意されている。これをフレームと呼ぶ。すなわち、ファイルの入出力や各セクションにわたる処理を行なうための“FILE フレーム”、修正しようとするセクションおよびマスク番号を指定するための“SELECT SECTION フレーム”（図5）およびパターンを表示し検査・修正するための“MODIFY PATTERN フレーム”（図6）である。

各フレームのCRT画面は中央の枠に囲まれた working area とそのまわりの control area とに分れている。左側の control area に表示されているのがコマンドと呼ぶ命令で、ライトペンでピクچすることによって指定できる。

フレームの移り変りもコマンドによって行なうが、移り変りは全く自由ではなく、制限が設けられている。この様子は図4に示されている。

コマンドには、ライトペンで指定するものとファンクションキーで指定するものがあるが、ライトペンで指定する場合は指定が正しく行われたかどうかをオペレータが確認した後に、実行が開始される必要がある。また、コマンドの処理を規定するために、補助的なデータ（アージメントと呼ぶ）を指定する 경우가多い。例之は、図形を平行移動させるためには、移動すべき図形と移動方向・距離をアージメントとして指定する必要がある。アージメントとしての指定方法としては、



- (i) 図形を指定する。(ライトペンでピック)
- (ii) 臭(位置)を指定する。(ライトペンで図形ピック, ポインティング_{マウス}タイプラフ)
- (iii) リスト(例えばマスク番号リスト)の中からいずれかを選ぶ(ライトペンでピック)などがある。なお、これらアーギュメントの入力順は一般に任意であって、ファンクションキー(EXキー)を押した時臭にコマンドの実行が開始される。なお、ライトペンで図形や文字をピックする場合は、ライトペンが光を感じた時臭でその図形を一時的に明示し、オペレータが確認の後ペンのスイッチをオフにした時臭で、その明示を固定し同時に、指定データとして正式に登録する方式をとっている。これにより、自然な動作で正確なピックが可能である。

5. フレーム

各フレームの機能の概要を述べる。

5.1 FILE フレーム

仕事を始める時、最初に出るのがこのフレームで、マスター・ファイルやスレーブ・ファイル(磁気テープ形式)からのデータ入力や出力を行なう。

'MF LOAD' コマンドは、マスター・ファイルよりデータを入力してスレーブ・ファイルを作るもので、アーギュメントとして、セクション分けのレカたを指定する。

5.2 SELECT SECTION フレーム

図5に示すようにCRTのworking areaにはセクション分け図が表示される。この中の一臭をライトペンでさすことにより、これから修正しようとするセクションを、また右側control areaのマスク番号リストより1つをピックすることによりマスク番号を指定する。また表示位置や倍率も指定でき、それ以後、MODIFY PATTERN フレームで表示される部分を示す目安枠が表示される。

5.3 MODIFY PATTERN フレーム

ここで図6に示すように、選ばれた、セクション・マスクに所属するパターンを表示して各種の修正を行なう。ここで修正できるパターンは選ばれたセクション・マスクに属するもののみであるが比較・参照のために、隣接する地のセクションや別のマスクのパターンも同時に表示することもでき

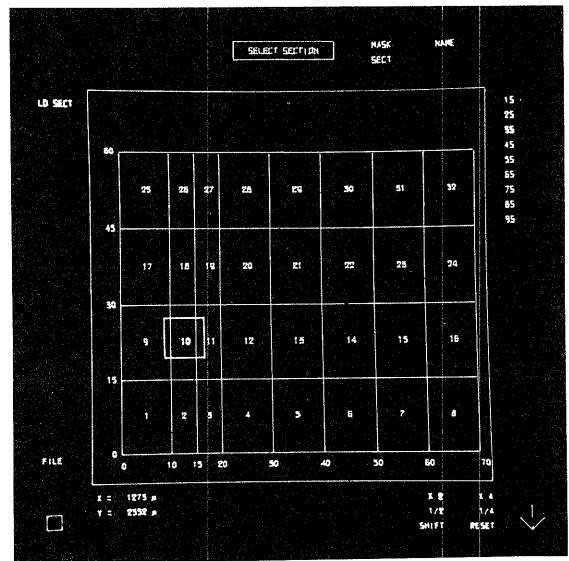


図5 SELECT SECTION フレーム

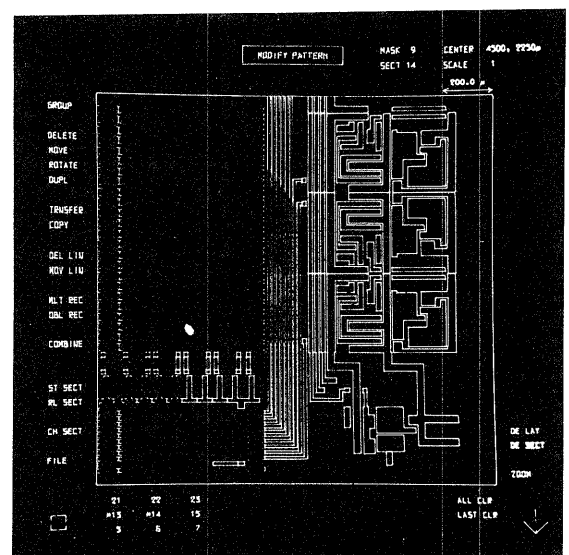


図6 MODIFY PATTERN フレーム

る。また、パターンの任意の部分拡大したり縮小したりするために 'ZOOM' コマンドが用意されている。

パターンの修正のためには 12 個のコマンドが用意されている。次にいくつかの例を示す。

'DELETE' : 指定した図形をとり除く。

'MOVE' : 指定した図形を平行移動する。移動量はタコプライタで指定する。

'DUPL' : 指定した図形と同じものを、指定した距離だけ離れた所に発生する。

'DEL LIN' : 指定した図形(閉図形)の指定した線分をとり除き残った部分を接続して閉図形とする。(図7)

'COMBINE' : 2個の閉図形を合成して1個(又は複数個)の閉図形とする。(図8)

'MLT REC' : 直角多角形を発生する。座標値は1度おきにタコプライタより入力する。

また、グループを定義するコマンドも用意されており、同一グループの図形は同時に、同じしかたで修正される。

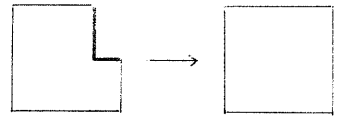


図7 DEL LIN の機能

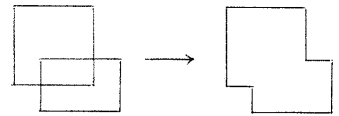


図8 COMBINE の機能

6. データ構造とその処理プログラム

このシステムでまとめたデータを保存する形態としては、

- (i) マスター・ファイル
- (ii) スレーブ・ファイル
- (iii) パターン・データ・エリア

の三種がある。これらはそれぞれ異った構造を持っており、それぞれの処理ルーチンが用意されている。マスター・ファイルは設計あるいは座標とりの段階で作られた順序でデータが入っているが、スレーブ・ファイルは各方眼紙・マスク毎のアクセスが容易であるような形でディスクの中に保存されている。また、パターン・データ・エリアは1つの方眼紙・マスクに所属するデータを保存するエリア中のエリアで、閉図形単位の追加、削除、修正が自由にできるような構造になっている。ここではデータ構造の一例として、スレーブ・ファイルの構造について述べる。

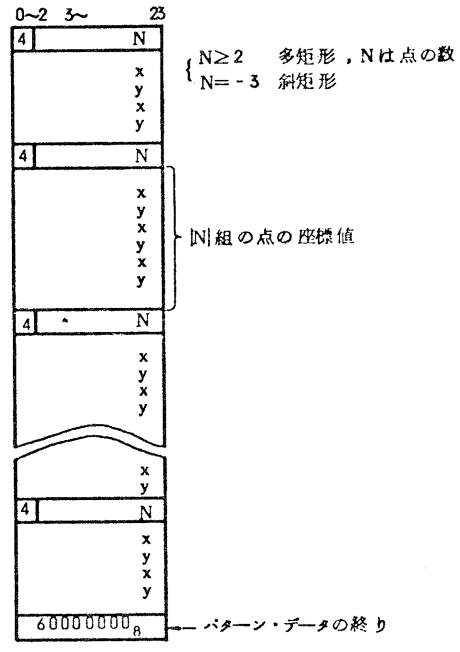
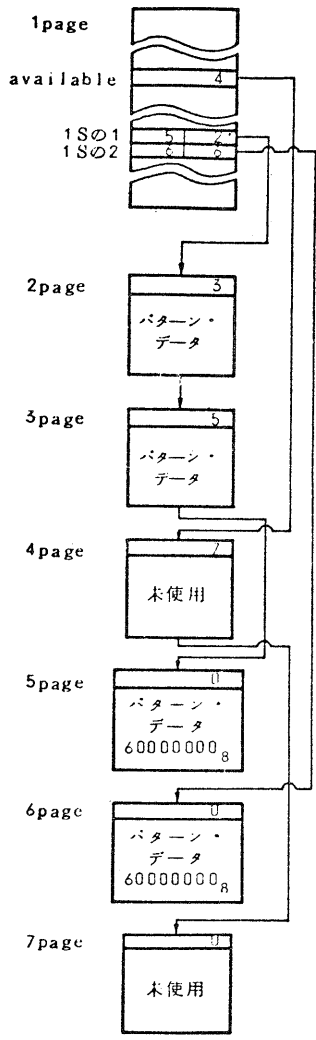
スレーブ・ファイルは、600語よりなるページを単位としており、現在800ページよりなっている。各セクション・マスクのデータを保存するページはそれぞれページ・ポイントによって接続されている。ページ・ポイントはそのページに続くページの番号で、各ページの先頭に保存されている。ファイルの最初のページには制御情報が入っているが、その中には各セクション・マスクの先頭ページ番号と最終ページ番号が入っている。また、未使用ページもポイントによって接続されている。図9(a)はこのようすを示している。ICパターンはマスクによってデータ量が非常にばらついており、しかも修正によりデータ量の増加や減少が起るので、このような構造は有効である。2ページ目以降には実際のパターン・データが入っているが、これは図9(b)に示すように、一閉図形に対して

各
 尖の数と長の座標よりなり、これが1セクション・マスクを連続している。直角
 多角形は1尖おきの座標、斜矩形については、一边を表わす2尖とそれに平行な
 線上の1尖により表現している。

スレーブ・ファイルでは、閉図形単位でアクセスする必要がないので、1セク
 ション・マスク単位で sequential に読み出し、書き込みを行なうルーチンのみ
 用意されている。このルーチンはページ・ポインタの処理やI/Oのバッファリング
 の処理を行なっているので、ファイルの構造を考えないで使用することができる。

コア内のパターン・データ・エリアは閉図形単位で、削除や追加を行なう必要
 があり、構造と処理ルーチンはそのように設計されている。

表示図形に関しては、図形あるいはそのグループに対して名前(転送コード)
 をつけ、リフレッシュ・バッファ中のロケーションとの対応関係を記憶しておく。
 名前のつけ方により、図形の分類、パターン・データ・エリアとの対応を表現す
 ることができる。図形間の関係は、このシステム
 では、Tree 構造だけであるので、これで十分
 表現できる。



(b) パターンデータの形式
 (a) ページ構造とポインタ

図9 スレーブ・ファイルのデータ構造

7. ソフトウェア・システムの構成

このソフトウェア・システムは次のようなプログラムから構成されている。

- (1) モニタ・プログラム
- (2) コマンド処理プログラム
- (3) データ構造処理プログラム
- (4) 図形処理プログラム

これらのプログラム（実際には各々がサブルーチンの集合体と考えてよい）間の関係は、図10に示すようになっており、ソフトウェア・システム全体は(1)のモニタ・プログラムが統括している。これらの働きを簡単に説明すると、(1)は基本的な入出力処理や割込み処理およびエラー処理の役目とオペレータからの入力を解析してしかるべきコマンド処理プログラムにコントロールを引渡す制御の役目とを持っている。(2)は入力の内容に従ってデータの表示をかえたり、各種処理プログラムを実行させることによりコマンドの機能を実行する。いわゆる設計用の各種応用プログラムはこのカテゴリーに入る。(3)はデータを処理するもので、6.で述べたように3種類のデータ構造の処理とそれらの変換を行なう。(4)では、グラフィック・ディスプレイ装置を制御するための基本プログラム（GUP）を利用して、図形の表示、シザリング、拡大などの処理を行なう。

これらのプログラム群をバッチ処理のプログラムと比較してみると、(1)が主プログラムで、(2)(3)(4)が副プログラムと考えられる。(3)(4)はともにサブルーチン・パッケージであるが、外部仕様はかなり専用化してある。

実行時のメモリ割当ては図11に示されておりである。コマンド処理プログラムはオーバーレイを行っている。

実行時のメモリ割当ては図11に示されておりである。コマンド処理プログラムはオーバーレイを行っている。

実行時のメモリ割当ては図11に示されておりである。コマンド処理プログラムはオーバーレイを行っている。

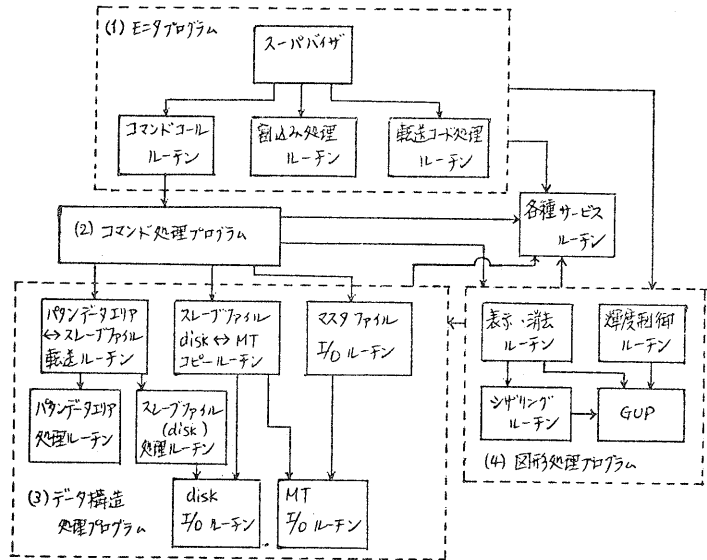


図10 ソフトウェアシステムの構成

モニタプログラム	1.2K
データ構造処理プログラム	8.3K
図形処理プログラム (GUPを含む)	15.4K
各種サービスルーチン	6.0K
モニタエリア (パンデクシア 12.0K)	24.7K
その他	4.5K
コマンド処理プログラム — オーバーレイ領域 —	34.2K
	94.3K

図11 メモリにおけるプログラムのレイアウト

8. ソフトウェアの開発と保守

ソフトウェアの開発は、T-3400/M41の標準オペレーティング・システムであるTOPS-14を用いて行なつた。システムの記述言語としては、これまでに開発したソフトウェアをなるべく使うためと今後の拡張や修正を考慮してFORTRAN言語を主に使い、効率を重んじる部分やFORTRAN言語では記述できない部分(ビット処理など)にアセンブラ言語を使うことにした。例えば図10の(1)および(3)は主としてアセンブラ言語により記述されているが、他のプログラムはほとんどFORTRAN言語により記述されている。ただし(4)で用いるGUPはほとんど汎用として開発されたもので、すべてアセンブラ言語によって記述されている。したがって(1)のモニタ・プログラムを除き、プログラム間の結合はすべてCALL文(アセンブラ言語ではシステムマクロ命令のCALL)を使って行われている。

標準オペレーティング・システムとの関連について述べると、このシステム自身はまったくオペレーティング・システムとは独立であつて、システム中の基本的なサブルーチンのみをオペレーティング・システムから抜粋して組み込んでいるに過ぎない。具体的には、割込み処理ルーチン、入出力処理ルーチン、GUPなどである。これらをまとめてシステムとして編集するプログラムはオブジェクトエディタと呼ばれ、オペレーティング・システムの基本プログラムである。

システムの保守については次のように行われている。プログラムの修正や追加は個々のプログラムを修正または作成したあとで、前述した編集プログラムを用いて再度システムを作り直す手続きをとらねばならない。しかし、システムの外部仕様の変更(機能の変更や追加)は、ほとんど(2)のコマンド処理プログラムの修正又は追加のみ、(1)(3)(4)は変更の必要がほとんどない。

参考のために、ソフトウェア開発に要した期間と人数および概算のプログラムサイズを述べると次のようである。

開発に要した期間	9ヶ月
人数	平均2人(約18人・月)
プログラム・サイズ	FORTTRAN 約8,000枚
	アセンブラ 約2,000枚

(ただし、オペレーティングシステムから抜粋したものは除く)

9. システムの評価

このシステムは、現在実際にルーチンワークで使用されているが、当初の目標機能は十分果している。

設計目標(i)の大規模データ処理については、現在約40万臭よりなるパターンを扱うことができる上、スレーブ・ファイルの合計ページ数を増すだけで簡単に拡張することができる。また寸法は0.1μ単位で最大10mm角まで扱える。これらの値は当箇考えられるLSIに関する要求を十分満たしている。

(ii)の応答時間については、コマンドやアргументの指定の応答は瞬時であるが、各コマンドの実行時間も表1に示すように非常に短い。表1は約5mm角の標準的なLSIについてのデータであるが、修正コマンドはいずれも0.5秒以下と非常に速く、ファイル(MT)の入出力以外は長いものでも十秒台である。大量に図形を表示するコマンドは比較的時間がかかっているが、この時間はディスプレイに図形を転送するのに要する時間で決っており、ディスプレイのデータ

やりとりに要する時間は無視できる程である。表2は相当熟練したオペレータがコマンドを使うのに要する全時間である。表1と表2からわかることは、実際の作業の速度はオペレータの動作によって決まり、計算機の待ち時間が少なくなっていることで、システム効率から考えると問題となる点である。

表1 コマンドの実行所要時間

MF LOAD	1' 15"	
MF STOR	1' 04"	
LD SECT(表示図形約200個)	18"	
(# 約100個)	8"	
ZOOM(表示図形約200個)	18"	作業方眼紙のみ
(# 約100個)	9"	#
(# 約7個)	5"	#
修正コマンド(MOVE DELETE など)	0.5"以下	

- 注1. いずれもEXキーを押してから実行終了してwaiting loopに入るまでの時間
- 注2. テスト回路は~~XXXXXXXXXX~~チップ面積5010×5240μ, マスク5枚のMOS LSIである。方眼紙分けは6枚である。
- 注3. 作業方眼紙としては4Sの1つの方眼紙を用いたが、その中には長方形ばかり約500個含まれている。上表中図形の個数は長方形の個数である。

表2 典型的な作業所要時間の1例

MF LOADで方眼紙分けをタイプインする(方眼紙6枚, 36文字)	32"
DELETEで次々に図形を削除する(10個連続の平均)	1.5"
MLT RECでタイプインして多矩形を作成する(3点とENDをタイプイン)	43"

- 注1. テスト回路は表1と同じ
- 注2. オペレータはタイプライタに習熟していない。
- 注3. タイプする数値はメモを用意してある。またピクする図形に関しては、探すのに時間がかかったり、迷ったりすることはないものとする。

目標(iii)の操作性については、全然計算機の知識のない女子オペレータが十分使いこなしていることから一応十分であると考えている。

処理をすべてセクション・マスク単位とする事による不都合が設計時に予想されたが、実際には手設計の場合、設計図(A1方眼紙)と同じセクション分けを使うので、ほとんど不便はない。またセクションやマスクを移り変わるのに要する時間も表1(LD SECT)に示す通り短い。

リフレッシュ型のディスプレイを用いているので、図形が多くなるとフリックが出て問題となるが、本システムでは矩形約500個までは、実際に表示可能で、この程度まで表示すると、パターン^(や指定)の識別が困難になるので、普通はもっと拡大

表示して使っている。従ってリフレッシュ型であることによる問題はあまり感じない。たゞ、CRTの視野が狭いことは使用上の大きな制限の一つで、どうしても広い範囲を同時に見たいという場合もある。これはプロット画面を併用することによってカバーしている。^{なお}ピクセルした図形の輝度を変えて確認したり、修正結果を瞬時に表示できること、またライトペンが使えるので操作が簡単である等、リフレッシュ型の利点はすべて活かされており、これを確認することができた。

10. むすび

グラフィック・ディスプレイは図形データの入出力装置として非常に有効なものであるが、LSIパターンのように複雑なものを取り扱う場合には、その機能は十分というにはほど遠い。しかし、その制限をよく理解した上で、人間を含めたシステムの各構成要素の役割と機能をバランスよく組み合わせることにより、実用的なシステムの実現が可能である。

こゝでは、このようなシステムの一例として、OLIMPIC-LSIパターン修正システムを紹介した。

このシステムの特徴は、LSIパターンをセクション分けし、作業はすべてこのセクション毎に行なうこととしていることで、これはオペレータにある程度の余念なオペレーションと制限を要求することになるが、一方これによって速い応答を実現することができたので、全体的には有効なシステムとなっている。

今後の問題としては、計算機の待ち時間が多くあり、一台の計算機で多数の端末を同時に働かせるか、またはミニコンのような多価な計算機で働かせる等の方法で、さらに経済性を向上させることが課題である。

参考文献

1. D. K. Lynn : "Computer-aided layout system for integrated circuits", IEEE Trans., CT-18, 1, p128 (1971)
2. 田丸, 香田他 : "グラフィック・ディスプレイによるオンライン集積回路設計システム", 信学論 (D), 55-D, 9, p579 (1972)
3. C. X. Beardslay : "Computer aids for IC design, artwork and mask generation", IEEE Spectrum, 8, 9, p63 (1971)