

# 階層化設計指向データベースの概要と階層展開処理

柿沼守男 辻裕生 村井真一  
(三菱電機株式会社)

## 1. はじめに

L S I化される論理規模の増大に伴い、L S I全体を一括して論理・物理設計する作業は、人間の誤り多発、プログラムの計算時間の増大等により、実質的には不可能になると思われる。またL S I化される論理範囲の増大に伴い、L S I全体に同一物理設計手法のみを適用するのでは、集積度が低下する恐れもある。従って前者に対して設計の階層化による処理対象の小規模化、後者に対して論理の特性に適した種々の論理・物理設計手法の採用が必要になると思われる。(図1) そこで我々は設計の階層化、多手法化を指向したL S I設計データベースを開発した。このデータベースは当社の商用DMS(データマネージメントシステム)により管理されるランダムアクセスファイルである。ここでは当データベース及びレイアウト設計向け階層展開処理について述べる。

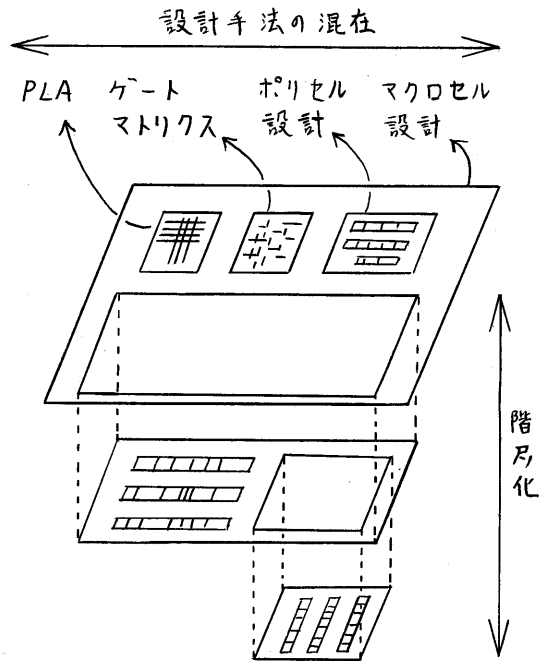


図1. 物理設計の階層化, 多手法化

## 2. システム概要

当データベースシステムはデータベース及びデータベースを囲む4種のプログラム群から構成されており、L S Iの設計方式(マスタースライス, ビルディングブロック)やデバイス技術(MOS, ECL)によらず共通に使用される。

### i) 入力プログラム群

グラフィック端末で作画した図面あるいは設計言語で記述した設計データがデータベースに登録される。

### ii) アプリケーションインターフェイスプログラム群

論理・物理設計システム向けに、データベースから設計データを抽出する。詳細配線遅延値の算出、冗長素子の削除等の処理も合わせて実施される。

### iii) フィードバックプログラム群

物理設計の結果(配線容量, 物理的サイズ, 端子位置等)がデータベースに格納される。これにより既設計部の詳細な遅延シミュレーションや、より上位階層の物理設計を開始できる。

### iv) ユーティリティプログラム群

当データベース専用で作成されたプログラムや当社DMSのプロセッサが、種々のユーティリティを提供する。

### 3. 設計データの表現

LSIは論理的・物理的に多階層的に表現される。すなわち低レベル要素のみの接続による一階層表現ではなく、まず高レベル論理のまとまりを設定し、それらの接続としてLSIを表現し、次にそこで使用した各まとまりをさらに低レベルにブレイクダウンする。これを繰り返して最終的にLSI全体を詳細に表現する。以後、一階層を表現する“何かのまとまり”をブロックと呼ぶ。

ブロックは具体的には以下のごとく意味付けられる。物理設計の観点に立てばLSI、ポリセルブロック、PLA、ゲートマトリクス、メモリ、セル、トランジスタ等に、論理設計の観点に立てばLSI、高機能素子、PLA、ROM、RAM、一般的ゲート、トランジスタ等に対応し得る。さらに以上のブロックの中間には単に下位ブロックの集合化として使用される単なるブロックが存在し得る。

ブロックのネスティングは意味的に矛盾が無い限り、完全に自由である。また意味付けに応じて記述する情報は取捨選択されるが、どのブロックも意味付けにより共通言語で記述され、入力プログラム群によりデータベースに登録され、さらに無矛盾性のチェック等のユーティリティプログラムのサービスを受ける。

一般にブロックの構造は図2のごとく周辺を囲む矩形の内部に小さな矩形を散在させ、それらを線で結ぶことで表現し得る。以下周囲の矩形を定義ブロック、内部の矩形を参照ブロック、結線を信号、信号が定義・参照ブロックに接続する点を端子と呼ぶ。当データベースではブロックを以下の情報で定義する。

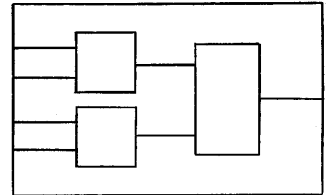


図2. ブロックの表現

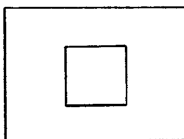
- i) ブロック認識情報
- ii) 定義ブロック情報
- iii) 参照ブロック情報
- iv) 信号情報

#### (i) ブロック認識情報

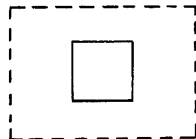
ブロックの定義、参照、信号情報すべてが、どのアプリケーションに対しても有意(必要)であるとは限らない。むしろアプリケーション毎に有意性は異なるであろう。さて参照ブロックが、あるアプリケーション(APと略)には有意だが別のAPには無意である場合、その別のAPへのインターフェースプログラム(APIと略)が参照ブロックを認識しない様な制御が必要になる。これはブロック毎にその意味付け(LSI、セル等)を明示的に宣言しても可能であるが、当システムでは以下の方法を採用した。

一般に階層関係の要素であるブロックは図3に示す4通りの認識を受ける。これを4Eと呼ぶコードで区別する。4Eコードは、あくまで情報の有意性を示すものであり、情報の存在/非存在を示すものではない。全てのブロックに対し、アプリケーション毎に〔API識別キー=4Eコード〕の形式で宣言する。

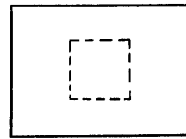
EXistence



ENtry



ELement



EMpty

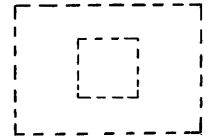


図3. 4Eコード(定義, 参照ブロック有意性の組合せ 実線: 有意)

4Eコードによる認識制御は明示的意味宣言に比べてデータベースシステムの融通性(拡張性)を以下の理由により高めるものと思われる。データベースに接続すべきAPは今後、継続的に増加し、同時にブロックの意味付けも多岐に渡るであろうことは仮定せざるを得ない。制御の種類数をC, APの数をaとすればブロックの意味付け数は基本的にはCのa乗になり得る。従ってAPを1個増やすには  $C^a(C-1)$ 個の意味付けを新たに定義する必要があり、制御数4でAP数が2が3になる場合意味宣言用キーワードは48増加するかもしれない。これは悪くすれば既存APIの修正、良くてAPIのコントロールデータの修正を強制する。いずれにしてもキーワードの決定において、全てのAPIの調査とドキュメント類の改訂は必要になるであろう。

具体的には、4Eコードは以下の利点を生む。

- i) 1個のデータベースが各サブシステム毎に専用な複数個のデータベースに相当するので、データを共有するにもががわらずAPI間の相互干渉がない。
- ii) 新たなAPを接続する場合、既に格納された設計データに対しては、新たに必要になった情報を付加するだけで済む。
- iii) 多階層的に表現された設計データをAPの仕様に合わせて一階層に圧縮する処理(階層展開)を汎用的に実施可能になり得る。

マクロセルを含むポリセルレイアウトシステム(AP識別=L)及び、高機能素子を含むゲートレベルシミュレーションシステム(AP識別=S)を例にとり、種々のブロックの4Eコード表現を、図4に示す。

4Eコードの変更はきわめて容易に実施可能である。そこであるブロックをポリセルブロックにするべきか、あるいは、そのブロックは単なるブロックにし、より上位のブロックをポリセルブロックにした方が、良い物理設計が得られるか等の試行錯誤が容易である。

また4Eコードはブロックを使用する時すなわち参照ブロックに属する情報として宣言できる。これはその参照ブロックに対してのみ有効な4Eコード修正宣言として扱われる。そこで、あるブロックにおいて2ヶ所で同一ブロックを参照する場合、一方をポリセルブロックとして、他方を単なるブロックとして設計することや、一方を単なるブロックとして、他方をゲートマトリクスとして設計することも可能であり、きめ細かな設計に対処し得る。

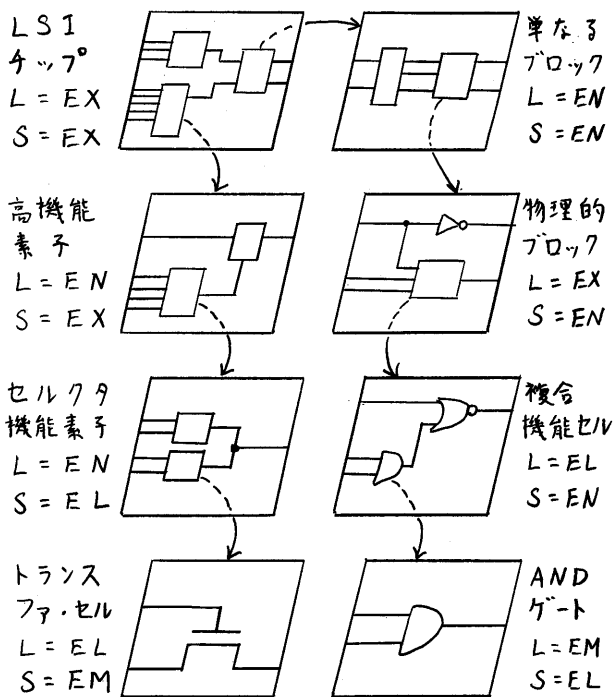


図4. ブロックの意味付けと4Eコード

(2) 定義ブロック情報

定義ブロックに関する情報を表1に示す。いかなるブロックに対して意味があるか、値を決定するのは人間が(M)、プログラムが(P)を付記する。

表1. 定義ブロックに関する情報

定義ブロックの名称, 設計者, 日付	全ブロックに必須	(M)
パッケージ, チップマスター, 型名	LSI	(M)
APIの特別な制御用コード	APIの4EコードがEL, EXに必須	(M)
APで使われる当ブロックの種類	..	(M)
物理的サイズ	物理設計のEL, EXに可	(M,P)
配線の通過可能域	..	(M)
ディレイ算出パラメータ	ディレイ算出すべき素子に必須	(M)
固定的ディレイ値, 容量値	特別なセルのみ	(M)
論理の入出力端子名	全ブロックに必須	(M)
上記端子に接続する信号名	参照ブロック又は物理的端子に接続	(M)
上記端子の論理的等価性	論理設計におけるEL, EXに可	(M)
物理的端子名と信号名	物理設計のEL, EXに必須	(M)
上記端子の位置	..	(M,P)
上記端子のディレイ算出パラメータ	論理設計のELに可	(M)
上記端子の電氣的等価性	物理設計のEL, EXに可	(M)

(3) 参照ブロック情報

表2. 参照ブロックに関する情報

参照ブロックの名称, 識別名	参照ブロックが存在するなら必須	(M,P)
4Eコード修正宣言	矛盾が無い限り可	(M,P)
識別名チェーン	物理設計のEL, EXに可	(P)
配置位置と方向	物理設計のEL, EXに可	(P)
論理的端子名と信号名	必須	(M)
論理的端子が割付けられた物理端子	物理設計のEL, EXに可	(P)

(4) 信号情報

表3. 信号に関する情報

信号名	信号が存在するなら必須	(M,P)
特別処理コード (VCC, GND, BUS, CLK 等)		(M,P)
配線の重み	物理設計で処理される信号に可	(M,P)
配線容量	..	(P)
配線長	..	(P)

なお、当データベースはその構造として定義ブロック, 参照ブロック, 信号に対しては、無制限にデータを格納できるので、拡張性は高いと思われる。

#### 4. レイアウト設計向け階層展開

当データベースには現在、ビルディングブロック方式LSIとCMOS、ECLマスタースライス方式LSIのレイアウト設計システムが接続されている。いずれも単なるブロックが使用されるので、多階層設計データを各システム向け一階層設計データに圧縮する必要がある。このレイアウト設計向け階層展開処理は、いずれのLSIも同一プログラムで実行される。アルゴリズムも同一であり、ただビルディングブロック方式は複数回の展開を受けただけである。

##### 4.1 リダクション

階層展開の機能の一つに、階層表現に伴って発生し得る不要素子を削除する機能がある。この機能は、もしある論理が既存ブロックの論理の一部で実現可能な場合、その論理専用のブロックを定義しないで既存ブロックを流用する手段を提供する。この機能が無い場合、もし新たな専用ブロックを定義しないならばレイアウトセルを無駄に使用することになり、逆にセルの冗長使用も避けようとするれば、ブロックの数が増大して、データベースのパフォーマンスが悪くなる。

リダクションはその程度により4種の操作に分けられる。

- i) 不要と見なしたセルを削除する。(セルリダクション)
- ii) あるセルの入力端子や出力端子が不要で、かつ代替え可能な小さなセルがあれば、そのセルに置き換える。(セルリプレイス)
- iii) あるセルの一部の出力端子が不要で、代替えセルが無いならば端子だけ削除する。(ピンリダクション)
- iv) あるセルの一部の入力端子が不要で、代替えセルがなく、かつ論理的Low信号又はHigh信号を出力するセルがあるならば、そのセルを一部の入力端子に接続する。(セルクリエイト)

##### (1) リダクション仕様ファイル

一定デバイス技術におけるセルライブラリの増加傾向、新たなデバイス技術の発生等を仮定すると、リダクションの仕様を記述するファイルが必須と思われる。そこでセルの状態遷移を記述するファイルを定義し、これをプログラムのコントロールデータとすることで汎用化を図った。図5にリダクション仕様記述の一部を示す。

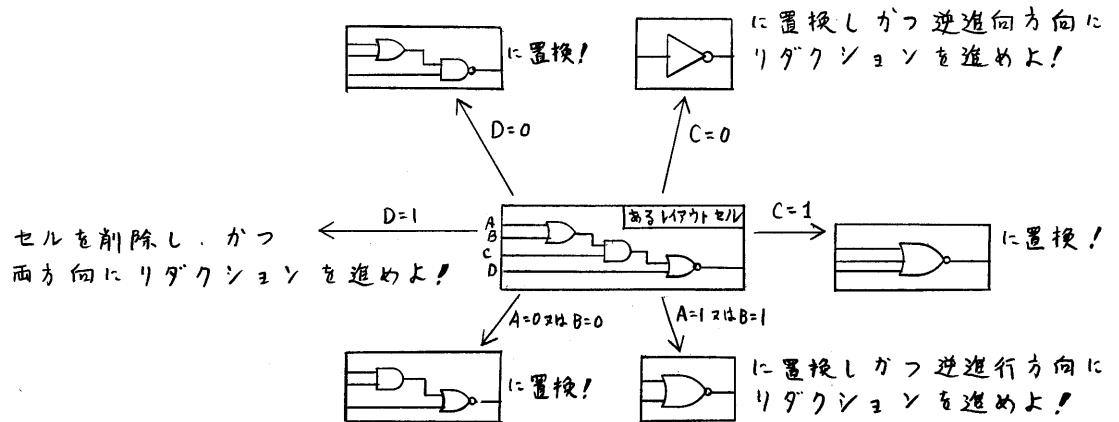


図5. リダクション仕様記述

リダクション仕様ファイルは80バイト/レコードのコンパクトファイルであり、 $n$ 入力 $m$ 出力セルは最大 $(5 + 2n + m)$ レコードで記述できる。セルライブラリが十分でない場合、セルの状態遷移が途切れることも起こり得る。その場合にはダミーセルを定義して解決している。例えば、もし3入力NORと5入力NORは存在するが4入力NORは作られていない場合、5入力NORの1本の入力端子が論理0ならセルクリエイ、2本の入力端子が論理0なら3入力NORへのセルリプレイスが起動される。

## (2) リダクションの手順

### i) バックワードリダクション

ブロックの出力端子が不利用ならブロック内部を逆進行方向(バックワード)にサーチして、その信号を出力するセルを求め、そのセルの出力端子に存在意義があるかないかをチェックする。存在意義があれば終了するが、意義がなければリダクション仕様ファイルに従う。その結果ピンリダクションであれば当該操作後終了するが、セルリダクションなら必ず、セルリプレイスなら必要に応じて当該操作後さらにバックワードトレースを続行する。

### ii) フォワードリダクション

バックワードリダクション終了後に起動される。ブロックの入力端子で論理0又は1が与えられているもののうちバックワードリダクションされていない端子があればブロック内部をフォワードにサーチする。セルを求め、リダクション仕様ファイルに従う。その結果ピンリダクション又はセルクリエイトなら当該操作後終了する。セルリプレイスならば当該操作後、必要に応じてバックワードリダクションを起動する。セルリダクションならば必要に応じてバックワードリダクションを起動すると同時にフォワードトレースも続行する。

## 4.2 展開の手順

i) ブロックのネスティンググループが無いことをチェックする。

ii) ブロック間の参照関係に対応する木を作成する。

iii) 木を利用して最良の階層展開順序を決定する。すなわち、あるブロックを展開する時には、それより下位に位置するブロックは以前に展開される様に決める。

iv) 決められた順番に下から上に向けて一階層ずつリダクションによるセル削減に務めながら、展開を進める。

## 4.3 展開結果の格納

データベースにおいて通常採用される階層表現法はAがB, BがC, CがDを参照する場合  $A \rightarrow B, B \rightarrow C, C \rightarrow D$ のごとく二階層参照関係の線り返しであり、当データベースも同様であるが、これに加えて、データベースの構造として明示的に  $A \rightarrow B \rightarrow D, B \rightarrow C, C \rightarrow D$ のごとく三階層参照関係を保持できる。もしAがポリセルブロック, BとCは単なるブロック, Dがセルの場合、当データベースではAの展開結果をAの以前の情報に付加する形で格納することが可能である。そこで展開で得たすべてのセルと信号をデータベースに格納する。これにより物理設計結果を一括管理する用意をする。

## 5. ユーティリティ

以下のユーティリティが提供される。

- (1) ブロック毎にその参照ブロック名称をバッチ処理的にエディットする。
- (2) ブロック毎にその論理的無矛盾性をチェックする。
- (3) データベースのリスティングプログラム  
当データベースでは定義ブロック情報として〔任意キー=任意データ〕  
(各4バイト)の形式でいかなるデータでも格納できる構造になっている。  
一方このプログラムは、(キー=データ)の組を多項論理演算のオペランド  
として入力し、演算結果が真であるすべてのブロックに対して種々のリス  
ト処理を実行する。これによりきめ細かなブロック管理が可能になる。
- (4) (3)と同様の演算により決定されたすべてのブロックに対して、配線長等の  
信号に関する物理設計結果を積算し、ヒストグラムをディスプレイ表示す  
る。
- (5) データベースの既使用ディスクグラニューブル数報告、ディスクページのオ  
ーバーフロー検出報告、データベースの Garbage Collection
- (6) ダンプ/ロード (当社 DMS 提供)

## 6. 実用上の効果

当データベースは、現在 MOSビルディングブロック方式 LSI, CMOSゲートアレイ, ECLゲートアレイに対して実用化されているが、システム拡張はきわめてスムーズに達成された。これは当然アプリケーションの類似性が主要因であるが、ブロック認識情報による統一的制御も大きく寄与している。

さてビルディングブロック方式 LSIの設計において、物理的レイアウト階層は最大5階層で構成されている。あるものは自動で、あるものは半自動で、あるものは人手でレイアウトされているが、いずれもデータベースで管理されているために、設計データの抽出、レイアウトチェック、詳細遅延シミュレーション等すべてのサポートを同様に受けている。またあらかじめ物理的ブロックとして定義しておき、使用する時に物理階層構組みを取りはずす4Eコード修正宣言も多用されている。

ゲートアレイでは、その開発期間が短いという特質をさらに高めるために、一定の論理的まとまりを持つブロックの多用が認められる。特に CMOSゲートアレイでは、ユーザーが多いため、論理ブロックの共同利用が進んでいる。これに伴いセルリダクションも多発している。

## 7. おおひ

大規模化、多様化する LSIの設計データベースとして、階層化設計を指向したデータベースを開発した。開発に当たり特に留意した項目は、データベースとしては汎用性、拡張性、及び保守性であり、設計システムインターフェースとしては、きめ細かなサポート機能である。実験動の範囲では、初期の目的は達成されたと思われる。

今後の向題として、マン・マシンインターフェースの高度化、ブロックのバージョン管理手法の確立、ブロックの分割・統合サポート機能付加が必要と思われる。