

9N-3

カスタム LSI 自動設計システムにおける設計工程管理
- オブジェクト指向データベースを用いた自動管理機能 -

津幡 真太郎 松本 典子 植田 雅彦 得能 誠司 西山 保
松下電器産業 (株) 半導体研究センター

1 はじめに

現在、カスタム設計を要する高性能な LSI の開発期間の短縮が求められており、カスタム設計技術の資産化、およびそれを再利用することによるカスタム設計の効率化・自動化が重要である。これらの実現のためには、設計資産のデータベース化、設計工程管理が課題となる。

そこで我々は、オブジェクト指向に基づいて既設計回路の再利用および設計工程管理を行う自動設計システムを開発している [1]。本稿では、そこで用いている設計工程管理手法について報告する。

2 カスタム設計工程管理の課題

カスタム設計においては、高性能を実現するために、専用セル設計や人手による論理設計など、回路の種類に応じた最適な設計手順の適用が必要となる。このため、設計手順が多様になり、設計者にとっては負担が大きい。

従って、カスタム LSI 設計工程管理の課題として、

- (1) 回路の種類に応じて最適な設計手順が適用されること
- (2) 設計者が設計手順の詳細を意識する必要がないことが挙げられる。

また、設計工程管理には、処理順序の最適化、負荷の分散なども考慮しなければならない。

3 オブジェクトモデルによる工程管理

上述の課題を解決するためのアプローチとしてはオブジェクトモデルが有効である。オブジェクトモデルでは、データとその手続き(メソッド)を一つのオブジェクトにまとめることができ、またメソッドの実現方法の隠蔽もできる。そこで、回路部品をオブジェクトとすることにより、回路データと設計手順を統合管理し、回路データに応じた最適な設計手順を適用することができる。また、設計手順の詳細を隠蔽することにより、設計者が設計手順の詳細を意識する必要がなくなる。

3.1 回路部品のモデル化

3.1.1 回路部品オブジェクト

本システムでは、回路部品を以下のようなオブジェクトにモデル化した。

オブジェクト	回路部品
データ	各種回路データ
メソッド	各種設計手順

図 1 にオブジェクトの一例を示す。図 1 では、回路部品オブジェクト FPU は、機能記述、論理ネットリスト、レイアウトデータなど各種の回路データを保持し、設計手

順(メソッド)として機能検証は Verilog-XL を、論理設計は LODES などを用いる。

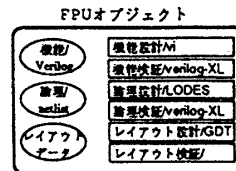


図 1 オブジェクトの例

3.1.2 設計手順のカプセル化

オブジェクトモデルにおいては、オブジェクトは内部の状態に応じた処理実行が可能である。このような、状態に応じた処理実行を設計手順メソッドとすれば、設計回路の状態に応じた処理が可能となる。例えば、回路が論理設計が完了していない状態であれば、論理検証を実行する際それに先だって論理設計を実行することができる。

また、設計手順に、最適な処理順序や負荷分散を考慮するようにカプセル化することも可能である。

3.1.3 クラスによる管理

オブジェクト間の関係では、一般に is-a 関係と part-of 関係がある。例えば、前者は「単精度 FPU は FPU の一種である」や「fpu1 は単精度 FPU である」という関係を表し、後者は「FPU は REGISTER を含む」という関係を表す。

本システムでは、種々の回路部品を、その機能により類別し、これをオブジェクトモデルにおけるクラス、またそれを再利用した際の設計対象回路をそのクラスのインスタンスとして捉える。また、クラス間にもその機能の抽象レベルにより、is-a 関係つまりクラス階層がある。クラスの階層の一例を以下に挙げる。



ここで回路部品クラスは、全ての回路部品の抽象クラスであり、データとメソッドの宣言のみがある。下位のクラスで、データやメソッドは具体化される。

また、クラスである回路部品は通常のオブジェクト指向プログラミング言語におけるクラスにくらべて、静的に part-of 階層構造を保持しているという特徴がある。

3.2 再利用

オブジェクトモデルでは、クラスの性質をそのインスタンスが持つので、回路部品のインスタンスを生成することにより、その回路部品の回路データ及び設計手順を再利用することができる。

Design Process Management in the Custom LSI Design Automation System
Shintaro TSUBATA, Noriko MATSUMOTO, Masahiko UEDA, Seiji TOKUNOH, Tamotsu NISHIYAMA
Matsushita Electric Industrial Co., Ltd.

また、本システムでは、再利用する際にカスタマイズすることも可能である。

また、回路部品は静的に part-of 階層構造を保持しているので、その構造を変化させるようなカスタマイズが行われれば、その設計対象回路つまりインスタンスは元のクラスとは構造が異なり、新たな別のクラスに属する。

3.3 設計工程管理

本モデルにおいて各設計者は、設計対象回路を回路部品オブジェクトとして扱い、その回路部品オブジェクトにメッセージを送付することにより、設計を行う。回路部品オブジェクトにメッセージが送付されれば、そのメッセージに対応したメソッド(設計手順)が実行される。part-of 階層を持つ回路部品オブジェクトにメッセージが送付されれば、その回路部品オブジェクトに含まれる全ての下位の回路部品オブジェクトにメッセージが送付され設計手順が実行され、階層部品の設計も容易になる。

4 システムの実現方法

4.1 オブジェクト指向データベースの利用

オブジェクト指向に基づいたシステムで用いるデータベースとしてはオブジェクト指向データベースが有効である。オブジェクト指向データベースの特徴としては、(a) オブジェクト識別性、(b) クラス、(c) カプセル化、(d) バージョン管理などが挙げられる [2]。本システムでは、これらの特徴を利用し、設計工程管理を実現する。

4.2 回路部品オブジェクト

実際の設計手順は、CAD ツールなどの設計手段をいくつか実行することである。現状の CAD ツールでは設計データはファイルであるので、回路部品オブジェクトには、回路データとして、回路データファイルを識別するための情報(パス名など)を保持する。また、設計手順として、その回路部品の設計手順を実行する CAD ツールの情報を保持する。

4.3 階層構造を持つ回路部品オブジェクト

(1) part-of 階層構造

回路部品オブジェクトが、内部に他の回路部品オブジェクトを含む part-of 階層構造を持つ場合、回路部品オブジェクトは、下位階層の回路部品オブジェクトと、その部品を結合する部分を持つ。例えばハードウェア記述言語ではメインモジュールである。この部品を結合する部分は、図 2 に示すように、その回路部品オブジェクトの内部に保持する。図 2 において、回路部品オブジェクト MCU は回路部品オブジェクト IU、FPU と結合モジュールを持つ。

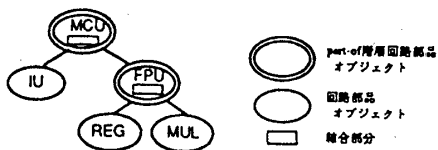


図 2 part-of 階層を持つ回路部品オブジェクト

(2) part-of 階層構造でのメソッド

回路部品オブジェクトが part-of 階層構造を持つ場合、そのメソッドは、以下のいずれかで実現される。

- 下位の回路部品オブジェクトに対してメッセージを送付し、下位オブジェクトのそれぞれでメソッドを実行する。
- 下位に含む全回路部品オブジェクトの回路データを集め、その全て設計データを対象にメソッドを実行する。

4.4 設計手段のオブジェクト化

(1) 設計手順の動的な変更

実際の設計においては、CAD ツールなど設計手段が変更され得る。そこで、設計手段を実現するオブジェクトを設け、設計手順はそのオブジェクトにメッセージを送ることにより実現する。これにより、設計手段を実行する場合のメッセージの送付先を変更すれば、動的に設計手順が変更できる。

(2) CAD ツールのオブジェクト化

現状では、各 CAD ツールはネットワーク環境において分散して配置されている。これらの CAD ツールをツールオブジェクトとしてモデル化し、このツールオブジェクトにメッセージを送ることにより、設計手段を実行する。

ツールオブジェクトは Unix 上での独立のプロセスであると考えられ、この設計オブジェクトとツールオブジェクト間のメッセージ送付はプロセス間通信により実現する。図 3 にその概要を示す。図 3 では「Message Manager」は上述のプロセス間でのメッセージ送付を実現するオブジェクトである。

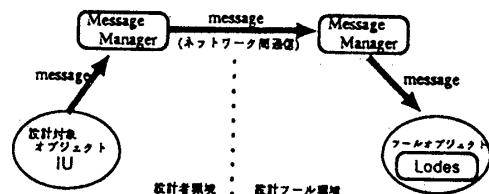


図 3 設計対象オブジェクトとツールオブジェクト

5 まとめ

回路部品をオブジェクトとしてモデル化することにより、設計手順と設計データを統合管理・保持ができ、回路の再利用が図れ、設計の自動化が可能となった。また、オブジェクトの状態に応じた処理が可能となり、設計処理が最適化された。

参考文献

- [1] 松本他:オブジェクト指向によるカスタム LSI 設計自動化手法, 情報処理学会第 45 回全国大会講演論文集, vol. 6, pp.19-20, 1992.
- [2] Atkinson, M. et al.:The Object-Oriented Database Manifesto, Proc. of Deductive and Object-Oriented Databases, pp. 40-57, 1989.