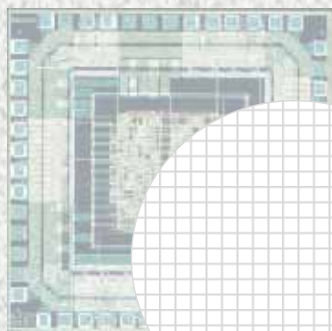


第4章 システムLSI の開発



第4章 システムLSIの開発

1 システムLSIの設計フロー

1-1 全体の設計フロー

プロセスの微細化が進み、大規模なLSIが実現できるようになった。そのため、LSIの開発フロー、およびLSI設計に必要な論理合成、タイミング検証、テスト設計、シミュレータ、およびレイアウトなどのツールの機能や特徴を理解し、効率よく設計することが非常に重要となってきた。図1.1で、一般的なシステムLSIの設計フローを示す。

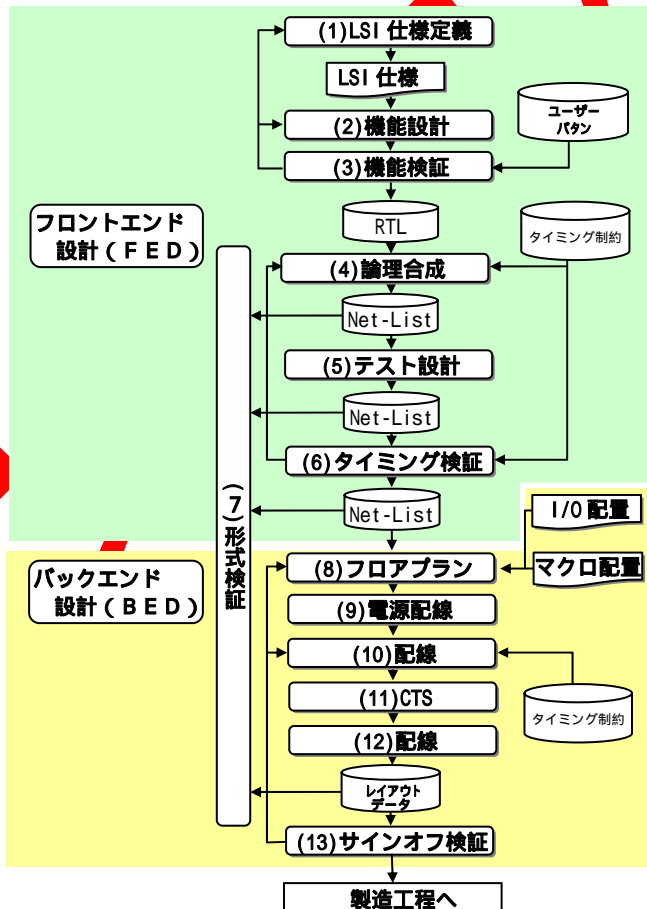


図1.1 LSIの設計フロー

LSI 設計は大まかに 2 つに分けられる。LSI の仕様定義～機能設計～機能検証～テスト設計～論理合成までの設計工程を**フロントエンド設計**（F E D と略す）と呼ぶ。さらに、フロアプランからサインオフ検証までの設計工程を**バックエンド設計**（B E D と略す）と呼ぶ。また、バックエンド設計の中での、フロアプランから配線までは**レイアウト設計**とも呼ばれている。

1 . 1 フロントエンド設計

（ 1 ） LSI 仕様定義

実現したいシステム（装置）のどの部分を、どのように LSI 化するかを定義する。この定義により、システム全体のスループット、消費電力等の性能、およびシステムを実現するトータルコストがほぼ決定する。

（ 2 ） 機能設計

Verilog-HDL、VDHL 等の HDL 言語で、LSI の機能を記述する。HDL は、論理合成ツールで取り扱いが可能な FF（フリップフロップ）と組み合わせ回路で構成する RTL で記述する。

（ 3 ） 機能検証

作成された RTL が期待通り動作するかを検証する。機能を検証するテストベンチを作成し、期待通りの動作をするか確認する方法が一般的となっている、さらに作成した RTL が論理合成、形式検証、テスト設計が可能か RTL チェッカーで事前に実施して確認する。

（ 4 ） 論理合成

RTL ネットリストから、主にタイミングや面積等の制約を与えて、ゲートレベルのネットリストへ変換する。生成されるネットリストは、ラリブラリ・セル（またはユニット）で構成する。

（ 5 ） テスト設計

LSI 製造後の出荷時に行うテストを容易化するためのテスト回路、およびテストパタンの生成を行う。

LSI 外部端子の入出力だけで LSI 内部回路のテストを容易に実現するため、オリジナルの回路に対してさまざまなテスト回路を挿入していく。

（ 6 ） タイミング検証

作成した回路が、要求される動作周波数でタイミング条件を満たしているかタイミング制約をもとに検証する。

（ 7 ） 形式検証

論理合成、テスト設計、レイアウト設計後の各工程でネットリストに回路が追加、変更されていても論理回路の等価性を維持しているかの確認をする。

1.2 バックエンド設計

(8) フロアプラン

I/O ブロック、ユーザーマクロ、およびRAM、IP マクロなどのハードマクロの配置位置をタイミング、配線性（特に迂回配線の状況）を考慮しながら決定する。

(9) 電源配線

ハードマクロ、ライブラリ・セルに必要な電源ラインを付加する。

(10) 配置

ゲートレベルのネットリスト中のライブラリ・セルを、フロアプランを考慮して自動的に配置する。概略配線を行い配線遅延、セル遅延を算出し、タイミング制約を考慮してタイミング収束も行う。

(11) CTS (Clock Tree Synthesis)

クロック信号上のFF間スキュー、およびクロックの遅延時間が極力小さくなるように、クロックラインの分岐やバッファリング（バッファ挿入）を行う。

(12) 配線

配置されたプリミティブゲート、ライブラリ・セル、ハードマクロ間の配線を、配線ルールを考慮しながら配線を行う。

(13) サインオフ検証

LSI 製造後の動作保証、品質保証実現のための検証を行う。検証項目はタイミング、論理、レイアウト等広範囲に及ぶ。

今日の大規模、高速、かつ消費電力が要求される LSI を開発する場合は、LSI 機能設計を行う設計者がフロントエンド設計を、フロントエンド設計を行う設計者がバックエンド設計を十分に理解することが求められる。一方で、バックエンド設計者はフロントエンド設計者にレイアウト上の特性を、逆にフロントエンド設計者は LSI 機能設計者に回路設計上の特性を十分に伝達することも必要となっている。

1 - 2 LSI 仕様定義

LSI 仕様定義では、開発対象となるシステムのどの部分をどのように LSI 化するかを検討する工程である。その製品だけでなく、製品群としての流用性や標準化も考慮していかなければならない。

あるシステムを実現する場合、システム内のどの機能について CPU 等を使用してソフトウェア的