特集1

第5章

システムLSIの論理合成から配置設計まで

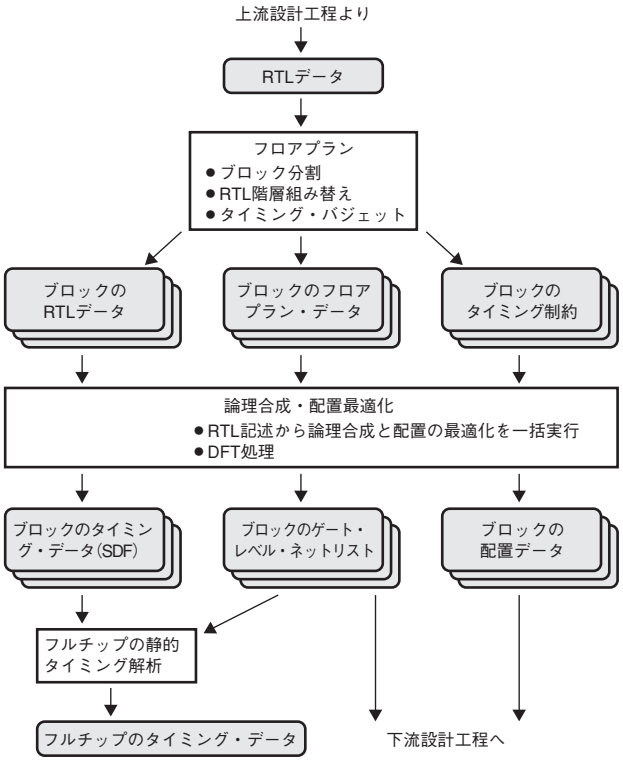
―― RTLとレイアウトの間に「中流設計工程」を定義

細田浩希，島沢貴美

ここではシステムLSIの論理合成から配置までの工程について解説する．筆者らは，論理合成から配置までの工程を独立させ，「中流設計工程」と呼んでいる．プロセス技術の微細化が進み，タイミングに対する配線遅延の影響が支配的になってきたため，従来の上流設計工程・下流設計工程という分担のままではLSIをうまく開発できなくなってきたという．（編集部）

複雑で回路規模の大きいシステムLSIの開発では，従来の上流設計工程（論理設計），下流設計工程（レイアウト設計）という区分けが業務の実態と合わなくなってきています．筆者らは，従来の区分けにはない「中流設計」とでも呼ぶ工程を定義して，上流設計工程（RTL設計），中流設計工程（論理合成，配置設計），下流設計工程（電源設計，クロック設計，信号配線設計）という3段階で作業を進めています．

ここではRTL設計と配線設計の間の橋渡しをする中流設計工程の作業に必要な知識や課題について述べます[注1]．



〔図1〕中流設計工程のフロー

上流設計工程から渡された（リリースされた）RTLデータからタイミング違反のない配置データを作成するまでの工程を筆者らは「中流設計工程」と呼んでいる．階層設計を採用したため，ブロック分割を行って論理合成・配置最適化ツールを適用した．

●中流設計工程とは？

システムLSIの開発では，上流設計工程からRTLデータが渡されてから，実際のチップとしての機能を実現するインプリメンテーション（実装）の作業が始まります．

一般に，論理LSI（ASICやマイクロプロセッサなど）の設計では，論理設計を行うフロントエンド設計者と，レイアウト設計を行うバックエンド設計者が，ゲート・レベル・ネットリストを介して分業を行っています．ところが，システムLSIの開発を成功させるためには，後に述べるいくつかの理由により，論理合成，フロアプラン（メモリなどのハード・マクロの配置や基幹電源の配線），そしてレイアウト設計の一部である配置設計の工程を一貫して行うことが重要になると筆者らは考えています．ここではこうした作業を，RTLデータを作成する上流設計工程と，クロック・ツリー生成や配線以降の作業を行う下流設計工程の中間に位置するという意味で，「中流設計工程」と呼ぶことにします（これは一般的な用語ではなく，あくまでも筆者らが内輪で使っている用語である）．

注1：ここで述べる内容の一部は，2001年12月の日本シノプシスのセミナ[1]でも発表された．

中流設計工程のフローを**図1**に示します．中流設計工程の入力はRTLデータで，出力がゲート・レベル･ネットリストと配置データです．まずフロアプランを行い，チップ・レイアウトの概要を決めます．筆者らはMePモジュールを搭載するMPEG-2 CODEC LSI「DEFiant」を開発するにあたって階層設計を行ったので，ここでブロック分割（チップを複数のブロックに分割する作業）を行いました．得られたフロアプラン・データに対して，分割されたブロックごとに論理合成と配置を行います．配置後のタイミング・データを組み合わせてチップ全体のタイミング解析を行い，タイミング違反がないことを確認できれば，続く下流設計工程へ配置データを渡します．この中流設計工程ではクロック･ツリー生成や信号配線は行いません．一貫して理想クロック，理想配線（詳しくは後述）として扱います．

システムLSIの開発スケジュールの中で，中流設計工程の作業が本格的に始まるのはRTLデータの最初の暫定版ができ上がるころです．以後，RTLデータの作成と並行して最適なフロアプランの模索，ツールの動作確認，スクリプトの作成，結果の品質の改善，RTL設計へのフィードバックなどを行います．完全に検証が終わったRTLデータが提供されるころには，動作確認の済んだフローを実行するだけでよい状態になっています．こうして，開発期間の短縮を目指しました．

●目標性能達成と設計期間短縮の両立が要求される

中流設計工程には，「目標性能の達成」と「設計期間の短縮」という二つの重い課題が課せられています．というのも，配置が確定した段階でおおよそのチップの性能が決まってしまうからです．この時点で目標性能を達成できなければ，あるいはスケジュールを守ることができなければ，製品の市場投入時期を逸してしまいます．

システムLSIは，複雑で回路規模が大きいのが特徴です．さらにプロセス技術の微細化に伴って，この二つの課題を達成することがいっそう困難になってきています．時間のかかる機能検証の結果をぎりぎりまでRTLデータに反映させたいという開発現場の事情もあります．

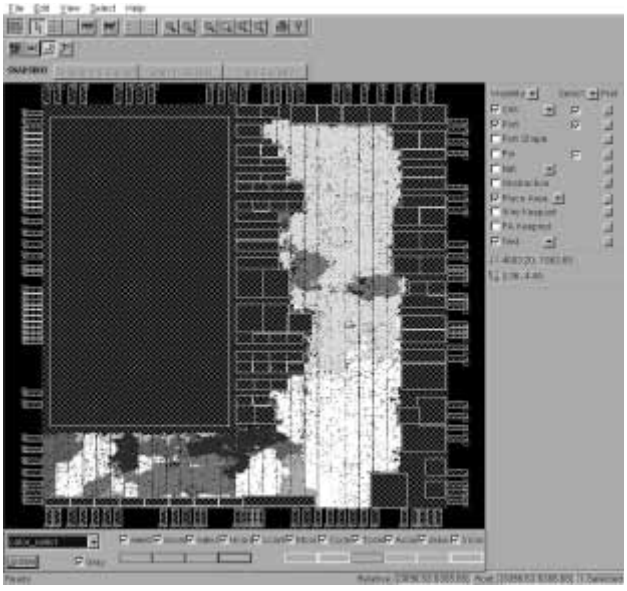
このような状態を克服するために筆者らがDEFiantの中流設計工程で重点を置いて取り組んだのが，「タイミング収束」と「階層設計」です．

●タイミング修正のもぐらたたき状態が発生

論理LSIのタイミングに対して，以前はセル遅延（ゲート遅延）が支配的でした．しかし，半導体製造技術の微細化によって，現在では配線依存の遅延が支配的です．かつて1μm程度のプロセスのころは，レイアウト前のタイミングの確認にワイヤ・ロード・モデル（ネットに接続されるセルの数と駆動能力をもとに信号遅延時間を見積もるタイミング・モデル）を利用していました．ワイヤ・ロード・モデルによってタイミングを確認すれば，レイアウトの際にあまり大きな修正を行わなくてもタイミング違反を解消できていました．今から考えると，ゲート・レベル・ネットリストとワイヤ・ロード・モデルを仲介として論理設計とレイアウト設計の分業がうまくいっていた「古き良き時代」でした．

ところが，現在主流の0.18μmや0.13μm以降のプロセスでは，ワイヤ・ロード・モデルによるタイミングの見積もりが大きく外れ，レイアウトの修正（ECO：engineering change order）を繰り返してもタイミングがいっこうに収束しないという状況があたりまえとなっています．

例として，筆者らの部門で開発されたMPEG-4ビデオ・オーディオCODEC LSI「T3（**図2**）」をモチーフとしたベンチマークにおける，従来手法の工程ごとのタイミング・スラックの推移を**図3**に示します．ここでタイミング・スラ



〔図2〕MPEG-4ビデオ・オーディオCODEC LSI「T3」
0.18μmルールのセル・ベースIC「TC260C」を利用．配線層数は4．DRAMコアを搭載している．