連載 HDLの記述スタイル

第8回

階層構造の決めかた

長谷川裕恭

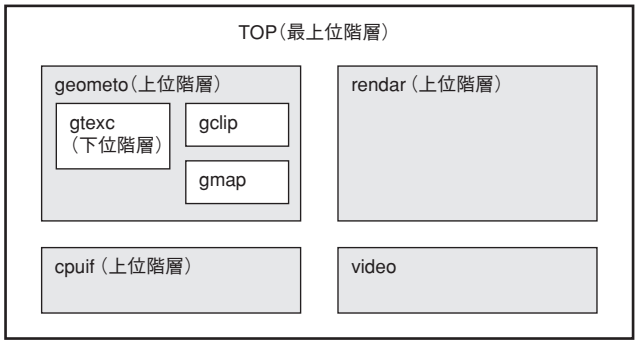
大規模LSIの設計において，ブロックの分割と階層化の問題は非常に重要なポイントである．これをしくじると，論理合成に時間がかかったり，性能が上がらなかったり，プロジェクトの進捗を管理しにくくなる．ここでは，一つのブロックの回路規模，複数の階層(ブロック)をまたぐパス，フロアプラン，オンチップ・バスなどの問題について解説する．（編集部）

大規模LSIを開発する場合，論理合成やレイアウトを考慮して階層構造を決定する必要があります．今回は，階層構造をどのように決定したらよいかについて解説します．

●基本ブロックの規模は2,000ゲート～2万ゲート

ASICを設計する場合，機能的な問題や各設計者への配分などを考えて，複数のブロックに分割します．

図1のように，論理機能をgeometo，rendar，cpuif，videoという四つのブロックに分割し，このうちのgeometoの回路規模が大きい場合には，これをさらにgtexc，gclip，gmapなどのブロックに分割していきます．このとき，TOP（最上位階層）－geometo（上位階層）－gtexc（下位階層）という階層構造を持つことになります．



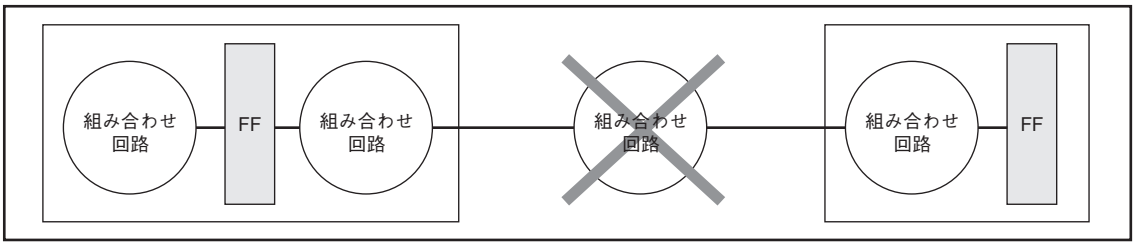
〔図1〕階層構造の例

geometoの回路規模が2,000ゲート～2万ゲートであれば基本ブロックになる．geometoが2万ゲートより大きい場合，さらにその下のgtexcなどが基本ブロックになる．

ASICの設計では，この階層構造を考える場合，二つの重要な問題があります．一つは検証しやすい，あるいは再利用しやすい階層構造（ブロックの分割）です．もう一つは，論理合成ツールやレイアウト･ツールによって扱いやすい階層構造です．

論理合成やレイアウトを考慮して階層を考える場合，まず基本階層（基本ブロック）という概念を持ってください． 基本ブロックは，回路規模と構造について制約があります．基本ブロックの規模は，2,000ゲート～2万ゲート程度になります．**図1**のように，TOP（最上位階層）の四つのブロック（geometo，rendar，cpuif，video）の大きさが2,000ゲート～2万ゲートであれば，この四つのブロックが基本ブロックになります．geometo，rendar，cpuif，videoの各ブロックの大きさが2万ゲートよりも大きい場合，その下の階層（gtexc，gclip，gmap）が基本ブロックになります．論理合成は，通常この基本ブロック単位に実施していきます．論理合成ツールは，この階層構造を維持したまま，ネットリストを出力し，これをレイアウト･ツールに渡します．

基本ブロックより細かい階層はサブブロックと呼びます．サブブロックは，論理合成のときに階層を破壊するケースが多く，レイアウト･ツールに渡す段階では階層を無視することになります．基本ブロックより細かい階層については，階層を作成するかしないか，自由に考えて



FF：フリップフロップ

〔図3〕基本階層より上位の階層には論理ゲートを配置しない

基本ブロックより上位に論理ゲートが存在すると，論理合成の妨げになったり，フロアプランが困難になったりする．

よいと思ってください．

●基本ブロックより上位には論理ゲートを置かない

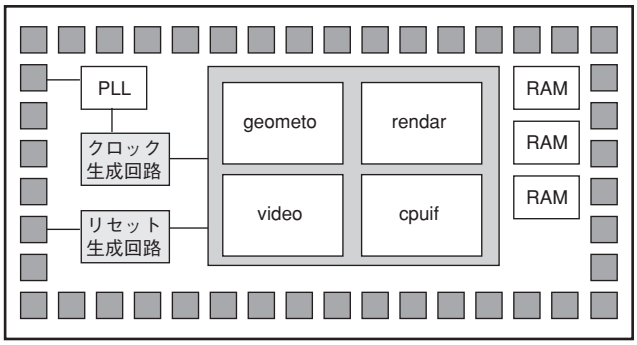
最上位階層（TOP）には，I/Oバッファ，クロック・ドライバ・セル，PLL，RAM，ROMを除いて，いっさい論理ゲート（RTLに論理機能を記述して論理合成ツールで生成するもの）を配置してはいけません（**図2**）．もし，論理ゲートを配置してしまうと，このセルを論理合成するとき問題を起こす可能性があります．また，下位階層（gtexc，gclip，gmap）が基本ブロックである場合，geometoにも論理ゲートを配置してはいけません．

前述のように，論理合成は基本ブロック単位に実施します．規模の大きな階層に論理をもったRTL記述が含まれていると，その部分を論理合成することが困難になることがあります．また，どの階層にも属さない論理ゲートが存在すると，レイアウトの際に適切な場所にセルが配置されにくく，動作速度が遅くなったり，配線効率が悪くなることがあります（**図3**）．

●論理合成ツールの性能と階層管理を考慮する

最近では，論理合成ツールの性能が向上し，一度に10万ゲート～50万ゲート規模の論理合成を実行できるようになっています．回路の面積や動作速度を気にしないのであれば，かなり大きい単位で合成することが可能です．しかし，一度に合成する回路規模が大きくなると，面積を小さくしたり，動作速度を向上させることが難しくなります．一般に，1万ゲート～2万ゲートの回路規模の単位で初期合成を実施しておき，10万ゲート～50万ゲートの単位で再度，階層合成を実施したほうが性能的に有利です．

また，大規模な回路を一度に合成する手法は，設計管理の観点からも好ましくありません．大規模な回路のRTL記述がすべて完成するまで論理合成を行わないと，論理合成後の回路の面積や速度の見積もりが遅れてしまいます．全体の回路が組み上がってから論理合成を行う方法を採ると，もし目標の面積や速度が得られなかったとき，RTL記述を修正して目標を達成することがかなり困難になってしまいます．



〔図2〕最上位階層には論理ゲートを配置しない

最上位階層（TOP）にはI/Oバッファ，クロック・ドライバ・セル，PLL，RAM，ROMだけを配置し，論理ゲートが生成されるような記述を行わない．できれば，RTL記述は最上位階層でまとめ，RTL記述のみの階層

そこで，大規模なLSIを設計する場合，**表1**のような表を作成することをお勧めします．このような表を作成しておくと，面積や速度の問題に気づきやすくなり，早い段階でRTL記述を修正できます．

こうした考えかたは，現在，最新の0.18 μmや0.13 μmのプロセスを利用して，数百万ゲートのLSIを設計されている方にとって，特に重要です．このような設計では，レイアウト後に予想以上に配線が長くなり，動作速度が遅くなることがよくあります．この状態になると，レイアウト側の対応だけでは，なかなか動作速度が改善しません．現在の大規模LSI設計において，目標動作周波数を実現する（タイミングを収束させる）ためには，RTL設計の段階で，基本ブロックのような比較的小さいブロック単位に，