はじめに

論理回路設計の回路規模が増大するに従い、設計資産を有効的に再利用していくことが重要になってきています。設計資産はIP (Intellectual Property) とも呼ばれ、システム・オン・チップ(System-on-a-Chip)を実現する上での重要な技術になっています。

IPは、現在、VSI(Virtual Socket Interface)という組織において共通化が計られています。IPを共通化することで、IPを複数のベンダーから購入してシステムを構築し、1つのチップ内に大規模なシステムを容易に統合することができます。そのため、従来より大規模な回路を、従来と同様の期間で設計することが可能になります。

ソフトIPは、主にHDL(Hardware Description Language)を用いて、RTL(Register Transfer Level)で記述されています。設計資産をRTL記述で残すことで、HDL記述を変更することが容易になります。また、パラメータ化された記述では、アプリケーションの目的に応じて回路を容易に変更することができます。

このようなソフトIPは、論理合成ツールを用いて、設計制約条件とテクノロジを指定することで、最終的にゲートレベル回路に実現します。設計制約条件は、要求する回路性能としてタイミング条件や面積、動作環境などを指定します。合成時に与える制約条件が適切でない場合には、適切ではない回路として合成される恐れがあり、目的の性能を満足させることが難しくなります。また、記述が適切でない場合、目標とする性能を満足することができない可能性もあります。

本設計スタイルガイドは、設計資産の再利用手法に基づきながら、HDL設計を進めるための設計スタイルを定義したものです。設計スタイルを共通化することで、設計者によって異なる記述スタイルや合成、検証などの設計手法を共通化することができます。そのため、記述者以外の設計者がその記述を変更したり、理解するための読解性を改善することができます。

本設計スタイルガイドでは、Verilog-HDLの構文規則を理解していることを前提として記述されています。Verilog-HDLの構文規則については、IEEE1364Verilog Hardware Description Language Reference Manualまたは、各種Verilog-HDL書籍を参照してください。Verilog-HDLによる論理回路設計では、Verilog-HDL対応のシミュレータによる検証、Verilog-HDL対応の論理合成ツールを使用することが前提となります。それらは各ツールのマニュアルを参照してください。

論理回路設計に使用する言語としては、Verilog-HDLの他にVHDLがあります。本設計スタイルガイドは、Verilog-HDLにそって記述されています。VHDLをお使いの方は、設計スタイルガイドVHDL版をお読みください。

本設計スタイルガイドは、日本の各半導体および電子機器製造会社が、自社内で適用しているVerilog-HDL(あるいはVHDL)による論理回路設計ルールを一般化し、IP流通を促進するために編纂されたものです。編集にあたりましては、株式会社 半導体理工学研究センターが参画企業からの意見をまとめ、IP技術支援小委員会の合意のもと作成されています。編集にあたりご尽力頂いた半導体理工学研究センターの方々に感謝の意を表します。

2002年2月1日

長谷川　裕恭
株式会社 エッチ・ディー・ラボ

設計スタイルガイド 2002 年度版発行に関して

設計スタイルガイドは、技術革新に対応するため、毎年１度見直しを行ないます。今回は、初版2001 年度版で見送りとなった内容が大幅に加わりました。その他にも、理解をより深めて頂くために、解説文の強化も図りました。その結果、2001 年度版に比べ、ほぼ倍のページ数に増加しています。主な追加変更ポイントは下記の通りです。

① 「第３章 3.3. テスト容易化設計」にLSI 製造試験に関するテスト容易化に、ロジック回路設計としてどのような考慮が必要か、細かい規定と解説が追加されています。（9 節構成、新規 37 項目）

② 付録として、シノプシス社の論理合成ツール Design Compiler の使用方法、ノウハウ集、ケイデンス社の論理合成ツールBuild Gates の使用方法とノウハウが追加されています。

③ 「第３章 3.4. 低消費電力設計」で低消費電力化の考え方、3.5.7. ～ 3.5.9. にバージョン管理ツールCVS の使用方法が新たに追加されています。（新規 14 項目）

④ 各項目のレベルを「必須」、「禁止」、「推奨」、「注意」、「参考」から、「必須」、「推奨 1」、「推奨 2」、「推奨 3」、「参考」に変更し、各項目の明確性を図りました。

⑤ 「第３章 3.3, 3.4, 3.5.7」以外でも各項目の見直しを行ないました。その結果、14 項目が削除され、新たに 27 項目が追加されています。既存項目での 59 項目の内容が変更されています。追加、変更された項目に関しては、本文解説も変更されています。

⑥ シミュレーションによって結果が異なる原因について、検証手法については、解説が大幅に変更、追加されています。

⑦ クロックツリーシンセシスに関して、同期RAM に関しての解説が大幅に変更されています。

本設計スタイルガイドは、以下のような章によって構成されています。

「第１章 基本設計制約」では、設計時に考慮する必要がある命名規則や考慮すべき設計スタイル、クロックの設計、同期設計や非同期設計時の考慮事項、および階層設計時の考え方など、設計を開始する際に考慮すべき基本的な設計制約について紹介します。

「第２章 RTL 記述テクニック」では、Verilog-HDL を用いて RTL 記述を作成するために必要な基本的な RTL 記述スタイルについて紹介します。記述スタイルでは、組み合わせ回路や順序回路の記述スタイル、および always 文や function 文、if 文や case 文などの使用方法についても紹介しています。

「第３章 RTL 設計手法」では、回路の分割設計を前提として、機能ライブラリの作成方法、設計資産のパラメータ化、テスト容易化設計、低消費電力設計、設計データの管理などを紹介します。この設計手法に従うことで、設計資産の再利用性が向上します。

「第４章 検証のテクニック」では、シミュレーションを実行するためのテクニックについて紹介します。紹介するテクニックには、テストベンチを記述する際のパラメータ化、タスクの使用、また、検証の進め方などがあります。

また、付録として
「A-5 Design Compiler による論理合成」と「A-6 Build Gates による論理合成」に、論理合成ツールの使用方法とノウハウを紹介しています。

本マニュアルで紹介している項目を以下のように分類しています。

項　目	内　　容
必　須	必ず守らなければならない項目を示します。
推奨 1	守らなければならない項目だが、違反した場合でも適切に対応策がとられていれば間違いない項目を示します。
推奨 2	回路品質、可読性、様々なツール使用上の問題を考え、推奨される設計スタイルの項目を示します。
推奨 3	推奨されるスタイルだが、注意を促す内容で厳守する必要がない項目を示します。
参　考	知っておいた方がよい設計ノウハウの参考事項です。

目次

3.3.1. DFT 向きのクロック、リセットを考える

① クロックは、LSI 外部入力端子から直接制御可能にする	必　須
② 2 系統のクロックを切り替えて使用している場合、テスト時はどちらか片方のクロックに固定する	必　須
③ ランダムロジックの出力をクロックにしない	推奨 1
④ FF のリセットは、LSI 外部入力端子から直接制御可能とする	必　須

解　説

スキャンを挿入するにあたって、もっとも重要なことは、スキャンテスト時にスキャンシフトが安全に行なわれる回路構造をとることです。スキャンシフトが安全に行なわれる回路構造とは、スキャンシフト時にスキャンパスに接続される全てのFFのクロック端子が、外部入力端子で直接制御可能なことと、スキャンパスに接続される全てのFFの非同期セット、リセット信号が、スキャンシフト中にアクティブにならないような回路構造をいいます。もし、非同期セット、リセット信号がスキャンシフト中にアクティブになるような回路構造だった場合、スキャンシフト中にFFのデータを破壊（FFをセットまたはリセットしてしまう）してしまい、正常にデータのシフトが行なえなくなってしまいます。したがいまして、非同期セット、リセット信号は、スキャンシフト中に非アクティブとなるような回路構造が必要となります。

● クロックは、LSI 外部入力端子から直接制御可能にする

スキャンを挿入するためには、スキャンテスト時にFFのクロック端子を外部入力端子から直接制御できるようにする必要があります。FFのクロック端子が外部入力端子から直接制御できない場合、スキャン挿入ツールは、この FF をスキャンの対象から除外してしまいます。また、スキャン挿入されていない部分に関しては、ATPG ツールによって故障を検出することが困難になってしまいます。

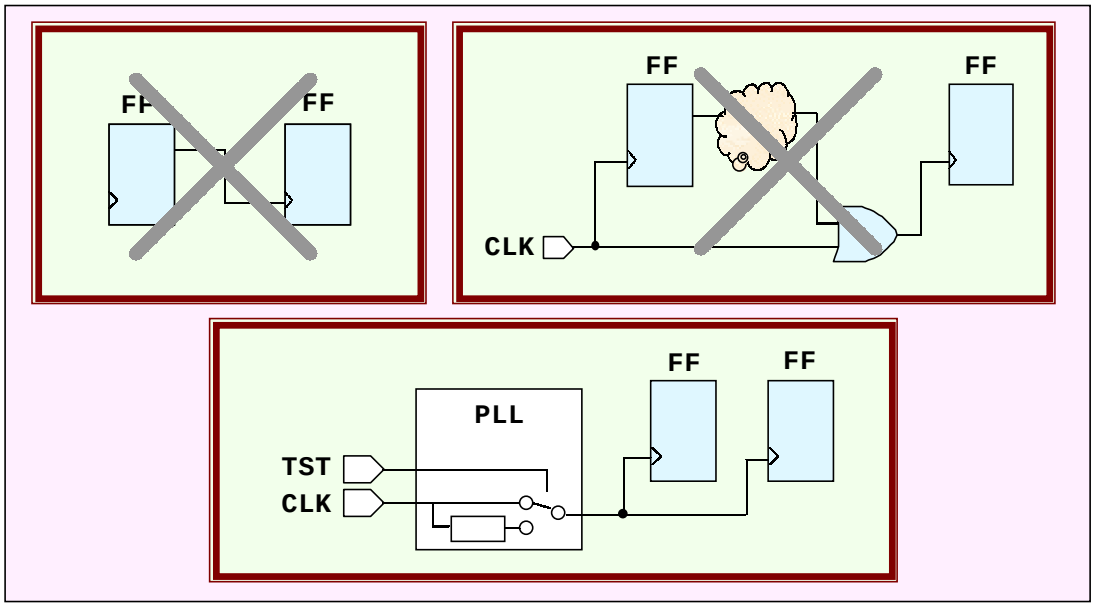

図 3-3　内部で発生したクロック、直接制御できないクロック、PLL の例

図3-3の上図のように、FFの出力がFFのクロック端子に接続されている場合や、FFのクロックラインがゲーティングされているような場合、そのFFはスキャンの対象から除外されてしまいます。このような場合は、スキャンテスト時に、クロックラインを切り替えるなどして、外部入力端子から直接、制御可能にする必要があります。また、LSI内部のハードマクロの出力がクロックになっているものは、スキャン挿入ができません。LSI内部では、PLLやDLLといったクロック発生マクロを利用します。通常このようなマクロには、図3-3の下図のようにテスト端子がついていて、テストモード時には、PLLに入力した基準クロックが、そのままクロックラインに出力されます。もし、このような切り替え回路がついていないPLL、DLLを利用する場合には、設計者が、外部クロックとの切り替え回路を挿入する必要があります。

「クロックは外部入力端子から直接制御可能にする」に関して、詳しい対策方法は「3.3.5. クロックラインのDFT対策例」で解説します。

● 2系統のクロックを切り替えて使用している場合、テスト時はどちらか片方のクロックに固定する

クロックが、LSI外部入力端子から制御可能であっても、図3-4のように2つのクロックを切り替えて使用している場合、スキャン挿入ができません。図の様にセレクタの選択信号をテスト信号によって固定し、スキャンテスト時は、常時どちらか一方のクロックを選択している必要があります。

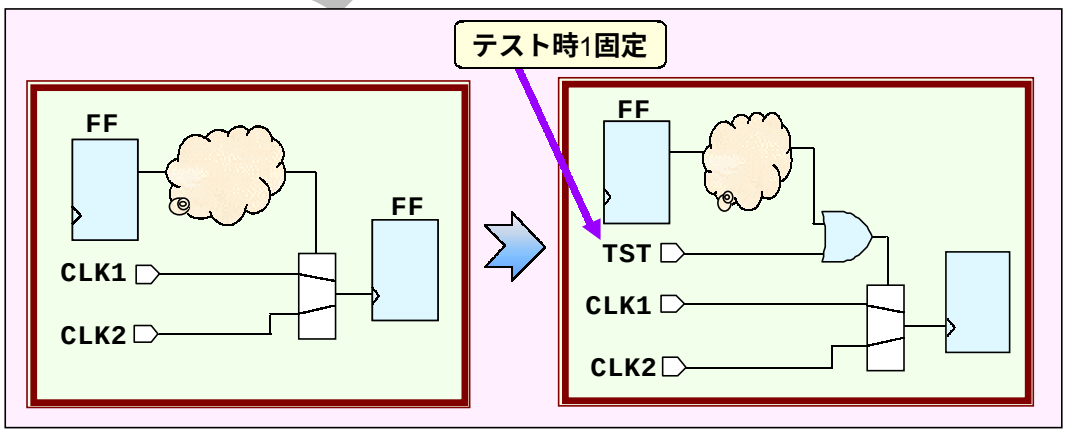

図3-4　2系統のクロックを切り替えている回路のDFT対策例

● ランダムロジックの出力をクロックにしない

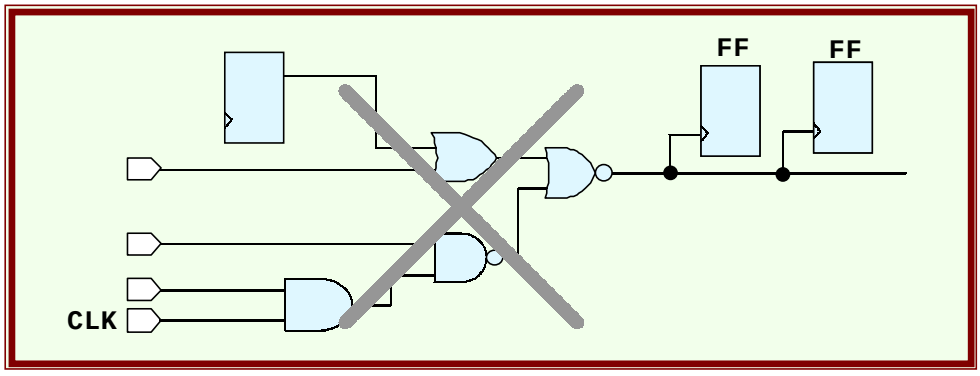

図3-5　ランダムロジックの出力をクロックにしない

図3-5のように、ランダムロジックの出力をクロックに使用している場合は、ランダムロジックの最終出力部分にセレクタを挿入し、外部クロックと切り替えられるようにしてください。

クロックラインにANDゲート、ORゲートあるいはセレクタを利用する方法は「3.3.5. クロックラインのDFT対策例」で解説します。

● FFのリセットは、LSI外部入力端子から直接制御可能とする

スキャンを挿入するためには、クロック系だけでなく、リセットラインにも対策を施す必要があります。リセットラインの対策としては、スキャンシフト中、FFにリセットがかかり、シフトデータを破壊することがないようにする必要があります。テスト信号等により、LSI内部でリセットラインが非アクティブとなるように固定した場合でもスキャン挿入は可能ですが、この場合、ATPGツールによって、リセットラインの故障が検出できないことになります。したがって、リセットラインは外部入力端子から直接制御可能にする必要があります。

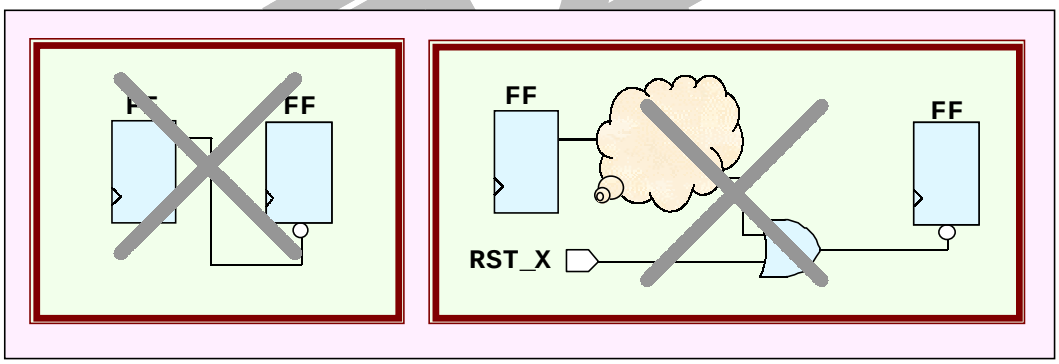

図3-6　内部で発生したリセット、外部入力端子から直接制御できない回路

図3-6の左図ように、FFの出力がFFのリセット端子に接続されている場合、このFFは、スキャンの対象から除外されてしまいます。また、図3-6の右図の回路では、RST_Xを'1'に固定することで、FFのリセットをディセーブルにすることができますので、スキャン挿入は可能ですが、この場合、FFのリセットラインの故障が検出できないことになります。したがって、これらの回路は、外部入力端子から直接制御できるようにしなければなりません。

リセットはノイズ対策、あるいはタイミングの問題でFFを何段も挿入して同期化することが一般的ですが、このような場合も、外部端子から直接制御することができませんので、対策が必要となります。リセットについての諸問題は、「3.3.6.リセットラインのDFT対策」で解説します。