

# 数クロック単位で互換性を担保する ASIC (Application Specific Integrated Circuit) リメイク手法

古石 憲男

近年、国内半導体業界は競争激化や半導体不足によるサプライチェーンの混乱に直面し、古いプロセスの ASIC の生産終了が進む中、我々は、顧客への商品供給維持のために ASIC 生産終了時の対応に苦慮している。一般的な対応策としては、一括購入（ラストバイ）か、再設計（リメイク）が考えられるが、リメイクでは、しばしば、設計資産が一部残っていないという課題や、古い ASIC に広く採用されてきた非同期回路の技術的な課題に直面する。

このような課題に対して、我々は、設計資産が一部残っていない中で非同期回路から同期回路への再設計を可能とする開発手法を提案する。具体的には、RTL 検証において、非同期回路である元の RTL と同期回路に再設計された RTL に対して同時に信号を入力し、それぞれの RTL からの出力信号をクロック毎に比較することで、これらの課題を同時に克服し、機能と性能の互換性を確保している。我々はこの手法を用いてすでに 6 個の ASIC をリメイクし、市場クレームは 0 件という成果を上げた。

## ASIC (Application Specific Integrated Circuit) Remake Method Guaranteeing Compatibility at the Clock Cycle Level

*FURUISHI Norio*

In recent years, the domestic semiconductor industry has faced intense competition and supply chain disruptions due to semiconductor shortages. Amid the phase-out of ASIC production using older processes, we have struggled with how to respond when ASIC production is discontinued to maintain product supply to customers. Common responses include bulk purchases (last buys) or redesigns (remakes). However, redesigns often face challenges such as the lack of design assets and technical issues with asynchronous circuits widely used in older ASICs.

To address these challenges, we propose a development method that enables redesign from asynchronous to synchronous circuits even when design assets are unavailable. Specifically, in RTL verification, we simultaneously input signals to both the original RTL circuit, which is asynchronous, and the RTL redesigned into synchronous circuits. By comparing the output signals from each RTL circuit clock by clock, we overcome these challenges simultaneously, ensuring functional and performance compatibility. We have already remade six ASICs using this method and achieved zero market claims.

### 1. まえがき

#### 1.1 半導体業界の動向について

近年、国外の半導体ベンダとの競争激化により国内の多くの半導体ベンダの縮小・撤退が進んでいる<sup>1,2)</sup>。また、

COVID-19 パンデミック、洪水・地震などによる自然災害、米中デカップリングなどの地政学的要因などを要因とした半導体不足は、サプライチェーンに大きな混乱をもたらしている。このような状況下で国内外を問わず一部の半導体ベンダは付加価値率の高い先端プロセスにリソースを集中し<sup>3)</sup>、古いプロセスの生産終了が加速すると思われる。

Contact : FURUISHI Norio norio.furuishi@omron.com

## 1.2 半導体部品生産終了に対する弊社の対応

我々の商品である CS/CJ シリーズにも、多くの ASIC と呼ばれる半導体部品を使用している。ASIC とは特定の用途や目的に合わせて最適化された半導体集積回路の一種である。特定の機能および処理に最適化されて設計されているため、商品の性能向上や差別化を図ることができる。

CS/CJ シリーズは、機械制御や工場の自動化、プラント制御など産業機器向けに使用される汎用プログラマブルコントローラの商品群である。図 1 に CJ シリーズの外観を示す。



図 1 CJ シリーズの外観

CS/CJ シリーズなどの汎用プログラマブルコントローラ商品群は、工場の自動化やプラント制御などの用途から、製品サイクルが非常に長くなる傾向にある。事実、CS/CJ シリーズも 1990 年半ばの販売開始から現在に至るまで、30 年近く顧客にご愛顧いただいております、今後も生産を継続する予定である。

しかしながら、先ほど述べたように、近年 ASIC の生産終了が頻発しており、その対応に直面している。我々は、顧客が我々の商品を安心して長期間にわたって使用いただけるように、部品の長期的な確保を行い、顧客への商品の供給責任を果たす必要がある。

## 1.3 ASIC リメイクの必要性について

顧客への供給責任を果たすために、生産終了される ASIC への対応方法は、大きく分けて、ラストバイ (Last Buy) と呼ばれる生涯使用数量の一括購入か、ASIC リメイクと呼ばれる ASIC の再設計のいずれかである。

ラストバイは、ASIC 生産終了時に将来の需要予測から搭載商品が生産終了するまでに必要な数量を算出し、一括購入し在庫確保する方法である。ASIC の生産終了時期と、当該 ASIC を使用している製品の生産終了時期が比較的近い場合に採用される方法である。本方法は、将来の製品需要を正確に予測し、適切な数量を購入する必要がある。過剰に購入した場合は製品の生産終了時に余った ASIC の廃棄ロスが発生するリスク、逆に不足した場合は製品の供給ができなくなるというリスクがある。またラストバイ数量

が非常に多い場合は、購入した ASIC の保管場所の確保、湿度管理などの適切な管理が必要となり追加コストが必要になるリスクがある。

ASIC リメイクは、異なった半導体ベンダや比較的新しい半導体プロセスを使用して、ASIC を再開発する方法である。ASIC の開発方法については、「1.4 一般的な ASIC 開発フロー」にて簡単に説明する。ASIC リメイクは、ASIC の生産終了時期と当該 ASIC を搭載している製品の生産終了時期が離れており正確な将来の需要予測が困難な場合、そもそも製品の生産終了時期が決まっておらず必要な数量が算出できない場合、購入すべき数量が非常に多く在庫管理コストが膨大になる場合などに採用される。本方法のメリットとして、比較的新しい半導体プロセスを採用することにより、供給の持続性を確保しコスト効率を向上させる効果が期待される

## 1.4 一般的な ASIC 開発フロー

ASIC 開発の主要な工程について「図 2 一般的な ASIC 開発フロー」を用いて簡単に説明する。

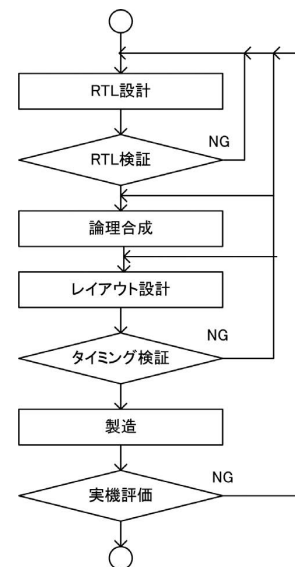


図 2 一般的な ASIC 開発フロー

RTL 設計 (Register Transfer Level Design) : ASIC 内部の回路設計を行う工程である。回路の実現方法や設計意図を設計仕様書に回路図やタイミングチャートなどで記載し、その内容を Verilog HDL や VHDL といったハードウェア記述言語を用いて、論理回路よりも抽象度の高い、フリップフロップと呼ばれる記憶素子間の処理を明確化したレジスタ転送レベル (RTL : Register Transfer Level) でシステムの機能と動作をモデル化する<sup>4)</sup>。

RTL 検証 (Register Transfer Level Verification) : RTL 設計工程で設計された RTL モデルの正確性を検証する工程である。論理シミュレーション、フォーマル検証などの技術

を活用し、設計が指定された仕様に適合していることを確認する。設計が仕様に適合していない場合は、RTL 設計工程に戻り修正が行われる。本工程には、テストモデルなどの検証環境の作成や検証計画の実行が含まれる。

**論理合成 (Logic Synthesis) :** RTL モデルを物理的なゲートレベルのネットリストに変換する工程である。本工程により、高レベルの抽象化された設計が、実際の論理ゲートとしての表現へと変換される。この論理ゲート表現に変換された設計データをネットリストと呼ぶ。変換時には、回路の最適化、エリアとパフォーマンスのトレードオフ、電力消費の最小化などが重要な考慮事項となる。

**レイアウト設計 (Layout Design) :** レイアウト設計は、論理合成によって得られたネットリストを基に、ASIC 内の物理的な構成要素 (トランジスタ、配線など) の配置と配線を行う工程である。本工程は半導体ベンダにて実施される。本工程では、チップの面積効率、配線の最適化、熱管理、電力分配などの物理的な要件を考慮しながら、設計の物理的実装が行われネットリストに反映する。

**タイミング検証 (Timing Verification) :** タイミング検証は、設計された回路が特定のタイミング要件を満たすことを保証するための工程である。本工程は半導体ベンダによる検証の後、設計者側でも実施される。本工程は、レイアウト工程で行われた配置や配線の情報を反映したクロックや信号の伝播遅延情報をもとに、各タイミング要件を検証する。回路が特定のタイミング要件を満たさない場合は、レイアウト工程に戻り回路の配置や配線を見直す。それでもタイミング要件を満たさない場合は RTL 設計工程に戻る。

**製造 (Manufacturing) :** ASIC を製造する工程である。これまでの工程を経て生成された設計データをもとに、半導体製造工場にて製造される。製造された ASIC は、機能テストと品質検証のプロセスを経て、出荷される。

**実機評価 (Device Evaluation) :** 製造された ASIC を製品に搭載し評価する工程である。機能評価、性能評価、耐環境評価などが行われる。本工程により、ASIC は仕様通りの機能、性能および信頼性を確保する。

## 2. ASIC リメイク時の課題について

### 2.1 設計資産の問題

過去に開発された ASIC の設計資産は一部存在しないか、または正確性に欠ける不完全な状態のものであることが多い。ここで言う設計資産とは、「1.4 一般的な ASIC 開発フロー」で述べた各工程で作成された設計仕様、回路図や RTL、検証環境や検証結果、各種の検討結果書などである。これら設計資料の不足は、ASIC リメイク実行時において重大な障害となり得る。また設計資産の不確実性は、過去の設計に関わった技術者やチームの専門知識の不足に起因することがある。特に、設計の意図や過程が文書化さ

れていない場合、ASIC リメイク時に元の回路の正確性を確認し、必要に応じて修正を行うことがより困難になる。

さらに、知的財産権の対応も重要である。ASIC リメイクには、元の設計で使用された特許や他の知的財産権に関する権利の再確認が含まれる。特に、ライセンス契約が更新されていないか、期限切れである場合、法務部門と連携し対応に当たる必要がある。

### 2.2 技術的な問題

#### 2.2.1 非同期回路の問題

多くの古い ASIC においては非同期回路が広く用いられていた。非同期回路は同期回路に比べて、消費電力の抑制、性能や回路効率がよいというメリットがある一方、以下に示すようなデメリットがある。

- ・メタステーブルのリスク：非同期信号をフリップフロップなどの記憶素子で受ける場合、メタステーブルという信号が不安定な状態になる期間が発生する可能性があり<sup>4,5)</sup>、システムの信頼性に重大な影響を与えるリスクがある。このため、非同期信号をフリップフロップなどの記憶素子で受ける場合には、適切な同期化回路を挿入する必要がある。
- ・設計の複雑性の増大：非同期回路は信号間の実時間でのタイミングを個体のばらつきや温度や電圧変化の影響まで考慮した設計が必要であり、設計全体の複雑性を増加させ、保守性や拡張性に悪影響を及ぼす。
- ・検証網羅性の確保の難しさ：非同期回路は、実時間でのタイミングを個体のばらつきや温度や電圧変化の影響まで考慮した挙動を検証する必要があるが、RTL 検証ではすべての組み合わせを検証することが困難であり、一般的なタイミングの組み合わせで済ませることが多い。この結果、RTL 検証の網羅性を確保できていないことがある。タイミング検証についても、タイミング検証ツールでは信号の遅延時間が最大のケース、一般的なケース、最小のケースの3条件しかツールによる自動判定することができない。それ以外のケースは設計者が机上計算にて判断する必要があり、タイミング検証の網羅性が確保できていないリスクがある。

このため、非同期回路で設計された古い ASIC を非同期回路のままリメイクした場合、設計が複雑なため不具合を混入するリスクが高く、設計の妥当性や互換性は、実機評価による波形測定やランニング試験などで確認する必要がある。この結果、実機評価で問題が発生した場合は論理合成やレイアウト設計、最悪の場合 RTL 設計に戻るリスクが非常に大きい。リスクが発現した場合は、製造コストと期間が大きく増加する問題がある。



## 2.2.2 同期化再設計の問題

時代が進むにつれ半導体プロセスが微細化され、ASIC の回路規模が大きくなると、非同期回路のメリットをデメリットが上回るようになった。2003 年には、設計者によって異なる記述スタイルや合成、RTL 検証やタイミング検証などの設計手法を共通化して可読性と流用性の向上を図ることを目的とし、株式会社半導体理工学研究センターにより「RTL 設計スタイルガイド」が発行された。現在では本スタイルガイドが事実上の業界標準となっており、本スタイルガイドで非推奨とされた<sup>6)</sup> 非同期回路設計は次第に採用されなくなり、同期設計が主流となっている。同期回路設計により、非同期回路のような実時間単位での設計および検証は不要になり、クロック単位での設計および検証ですべての挙動が網羅できるというメリットがある。

一方、非同期回路で設計された古い ASIC を同期回路で再設計する場合、元の非同期回路と再設計後の同期回路で挙動が変わるという問題がある。具体的な例を、図 3 に示した実際に古い ASIC で採用されていた非同期回路の例を用いて説明する。

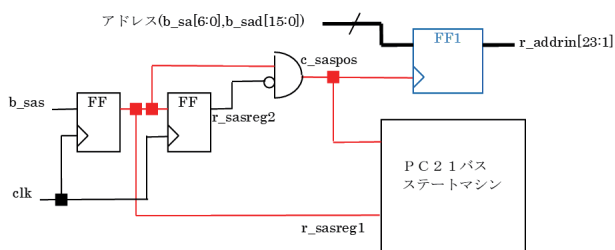


図 3 非同期回路の例

図 3 の非同期回路の問題点は 2 つである。

1. 外部信号である  $b\_sas$  をフリップフロップ (FF : flip-flop) でラッチしているが、同期化用のシリアライザーが挿入されていないため、メタステーブルと呼ばれる信号が不安定な状態が発生するリスクがあり<sup>4,5)</sup>、発生したメタステーブルが赤色のラインを経由して PC21 バスステートマシンに伝搬し、このステートマシンの挙動が不安定になるリスクがある。
2. 青色のフリップフロップ (FF1) のクロック端子にクロック (clk) 以外の信号が接続されているため、青色のフリップフロップ (FF1) の出力である  $r\_addrin[23:1]$  もクロックに同期しておらず、この信号を使用する後段の回路の検証網羅性の確保、およびタイミング検証が複雑になる。

上記の問題を解決するために図 3 の非同期回路を同期回路に修正した例を図 4 に示す。図 4 の同期回路では、図 3

の非同期回路の問題点がそれぞれ以下のように解決する。

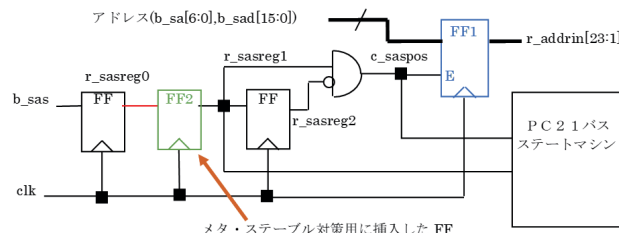


図 4 同期回路への修正例

1. 同期化用のシリアライザーを構築するために緑色のフリップフロップ (FF2) を挿入し、外部信号である  $b\_sas$  を同期化させる。これによりメタステーブルの発生は赤色のラインで示した区間のみで発生し、メタステーブルは後段の PC21 バスステートマシンには伝搬せず、当該ステートマシンの挙動が不安定になることはない。
2. 青色のフリップフロップ (FF1) をイネーブル端子付きのフリップフロップに変更し、クロック端子にクロックである clk 信号を接続、イネーブル端子に以前のクロック端子に接続していた  $c\_saspos$  信号を接続する。これにより青色のフリップフロップ (FF1) の出力信号である  $r\_addrin$  はクロックに同期され、検証網羅性の確保、およびタイミング検証が容易になる。

ただしこの場合、同期化用のフリップフロップを追加したことにより  $r\_sasreg1$  信号や  $c\_saspos$  信号の出力が 1 クロック分遅れ、オリジナルの非同期回路との動作が異なるという課題が発生する。「2.1 設計資産の問題」でも述べたように古い ASIC の設計資産は正確性のリスクがあり、同期化による動作タイミング変更の影響を正確に判断できないリスクがある。また、同期化による動作タイミングの変更箇所が複数ある場合は、その影響が相互作用するため、その判断はさらに困難になる。

## 2.3 ASIC リメイク時の課題

従来の ASIC リメイクでは、「2.1 設計資産の問題」がある中では、「2.2.2 同期化再設計の問題」を解決することができないと判断し、非同期回路を非同期回路のままリメイクしていた。その結果、「2.2.1 非同期回路の問題」で述べた実機評価での不具合発覚によるイタレーションが発生し、開発期間とコストが計画時から大幅に増加するという問題が複数の ASIC リメイクで発生した。

そこで筆者らは、従来手法の問題を解決するために、「2.1 設計資産の問題」がある中で、「2.2.2 同期化再設計の問題」を解決できる、すなわち同期化による動作タイミング変更の影響が十分に検証できる ASIC リメイク手法を

提案する。

### 3. 提案する ASIC リメイク手法

今回提案する手法は、まず「2.3 ASIC リメイク時の課題」で述べた従来の ASIC リメイク手法の課題を回避するために、非同期回路を同期回路に修正する。すなわち非同期設計手法で開発された古い ASIC を、現在主流である同期設計手法<sup>6)</sup>を用いてリメイクする。これにより、従来の ASIC リメイク手法で課題となっていたイタレーションの発生リスクを抑える。また、従来の ASIC リメイク手法で同期設計が採用できなかった理由である同期化による動作タイミング変更の影響が十分に見極められなかった課題を解決するために、本提案手法では RTL 検証に新たな作業を追加する。それぞれの工程の作業と確認内容について以下に述べる。

#### 3.1 設計資産の正確性の確認

本提案手法の RTL 設計工程の最初の作業は、現存する RTL ソースコードの信頼性の確認である。「2.1 設計資産の問題」で述べたように、現存する設計資産が正確性に欠ける不完全な状態であるリスクがある。そこで、現存する RTL ソースコードと半導体ベンダで保管されているレイアウト後のネットリストが一致するか、等価検証ツールを用いて確認する。等価検証ツールは、本来、合成工程やレイアウト工程内で合成結果やネットリストが元の RTL と機能的に等価であるかを確認するためのツールであるが、本提案手法では、RTL 設計工程で、RTL が元のネットリストと等価であるかを確認するためにも使用する。RTL と元のネットリストの等価性を確認することにより、少なくとも現存する RTL の正確性は担保される。図 5 に Synopsys 社の等価検証ツールである Formality<sup>7)</sup> による等価検証結果の出力例を示す。

```

***** Verification Results *****
Verification SUCCEEDED
ATTENTION: RTL interpretation messages were produced during link
of reference and implementation designs.
Verification results may disagree with a logic simulator.

-----
Reference design: r:/WORK/s_msmc_top
Implementation design: i:/WORK/s_msmc_top
1105 Passing compare points

-----
Matched Compare Points
-----
Passing (equivalent)      144      0      59      0      85      359      458      1105
Failing (not equivalent)  0         0         0         0         0         0         0         0
    
```

図 5 等価検証結果の出力例

RTL ソースコードと半導体ベンダに保管されているネットリストとが一致しなかった場合は ASIC リメイクの継続を含めて検討が必要である。少量の不一致の場合は RTL ソースコードを修正して一致させる。大量の不一致が発生している場合は、RTL ソースコードは使用できず、ラストバイや同等以上の機能や性能をもつ後継機種の開発など代

替策を検討する必要がある。

#### 3.2 同期化再設計の妥当性確認

RTL ソースコードとネットリストの一致がとれた場合、次の作業は非同期回路の洗い出しと同期化再設計の実行である。近年では、非同期回路の検出は人手で行わず CDC (Clock Domain Crossing) チェックツールと呼ばれるツールを用いて非同期回路の検出を行い、人手による検出漏れを防いでいる。本提案手法でも同様に CDC ツールを用いて非同期回路の検出を行う。代表的な CDC ツールの 1 つである Synopsys 社の SpyGlass CDC<sup>8)</sup> で指摘された非同期回路の例を図 6 に示す。

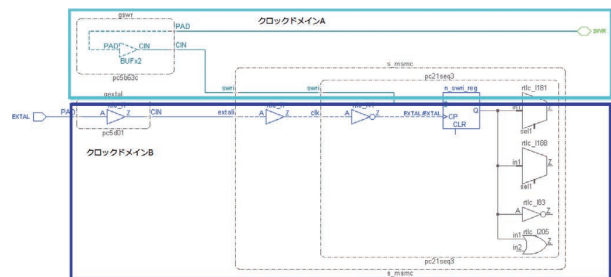


図 6 CDC による非同期箇所解析例

図 6 では、空色の信号 (クロックドメイン A) と青色の信号 (クロックドメイン B) がそれぞれ異なったクロックを持つため、青色の最終段のフリップフロップの出力にメタステーブルが発生するリスクがあり、後段の論理がその信号をそのまま使用しているため、システムに悪影響を及ぼす可能性があることを示している。検出された非同期回路を同期回路に修正し、修正が正しいか再度 CDC チェックを実施する。図 7 に SpyGlass CDC の結果出力を示す。本作業は、図 7 に示すようにすべての CDC エラーが無くなるか、当該エラーが問題ないことを確認できるまで繰り返す。これにより、同期化再設計の妥当性が担保される。

```

Results Summary:
-----
Command-line read : 0 error, 0 warning, 0 information message
Design Read       : 0 error, 16 warnings, 2 information messages
Found 1 top module:
s_msmc_top (file: ../src/s_msmc_top.v)

-----
Blackbox Resolution: 0 error, 0 warning, 14 information messages
SDGC Checks         : 0 error, 0 warning, 0 information message
Policy lint         : 0 error, 177 warnings, 2 information messages
Policy clock-reset : 0 error, 2 warnings, 14 information messages

-----
Total : 0 error, 195 warnings, 32 information messages

-----
Total Number of Generated Messages : 227
Number of Reported Messages       : 227
    
```

図 7 CDC 結果の出力例

#### 3.3 同期化再設計による互換性への影響確認

RTL 検証工程では、「2.2.2 同期化再設計の問題」で述べたように同期化による動作タイミング変更の影響が十分に見極められなかった課題を解決し、機能的および性能的な

互換性を担保するために図 8 に示す様な検証環境を構築する。緑色でハッチングした部分が本提案手法で追加されたモジュールである。

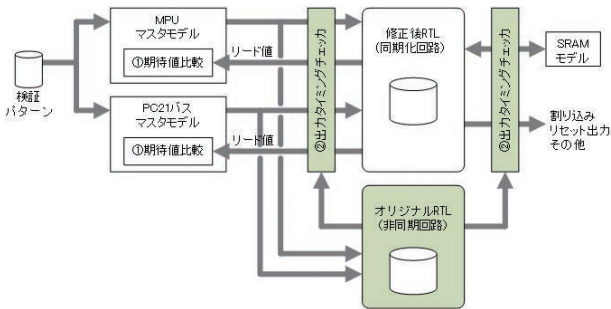


図 8 リメイク用検証環境構成の例

本環境の各モジュールの説明を以下に記述する。

- ・MPU マスタモデル：マイコンのリードアクセスタイミングの生成、ライトアクセスのタイミングの生成を行う。リードアクセス時には検証対象ユニットである修正後 RTL からのリードデータとあらかじめ検証パターンで示された期待値との比較を行いリードアクセスが正しく実行されたか判定を行う。
- ・PC21 バスマスタモデル：弊社独自のバスプロトコルである PC21 バスのシングルリードアクセスおよびバーストリードアクセスのタイミング生成、シングルライトアクセスおよびバーストライトアクセスのタイミングの生成を行う。シングルリードアクセスおよびバーストリードアクセス時には検証対象ユニットである修正後 RTL からのリードデータとあらかじめ検証パターンで示された期待値との比較を行いリードアクセスが正しく実行されたか判定を行う。
- ・修正後 RTL：検証対象のモジュール。リメイク対象 ASIC の RTL を前述の RTL 設計で同期化再設計したもの。外部 SRAM を経由して PC21 バスマスタモデルと MPU マスタモデル間でデータのやり取りを行う。
- ・SRAM モデル：修正後 RTL からのライトアクセスに対して、指定されたアドレスにデータを格納する。また、修正後 RTL からのリードアクセスに対して、指定されたアドレスのデータを返す。
- ・オリジナル RTL：本提案手法で追加されたモジュール、すなわちリメイク対象 ASIC の RTL である。機能的には修正後 RTL と同じであるが、非同期回路で構成されているため信号の出力タイミングが異なる。図 8 に示すように本モジュールにも修正後 RTL モデルと同様に PC21 バスマスタモデルからの入力信号が接続されているが、本モデルからの出力信号は各マスタモデルには接続されておらず、後述の出力タイミングチェッカに接続されている。このため、本モジュール

からのリードデータは、各マスタモデルとの期待値比較には使用されない。

- ・出力タイミングチェッカ：本提案手法で追加されたモジュールである。図 8 に示すように修正前 RTL モジュールと修正後 RTL モジュールの出力信号が接続されており、それぞれの信号の値および出力タイミングの差をクロック単位で確認する。出力信号の差が一定のクロック数以上、または値が一致しない場合にシミュレーションエラーを通知する。

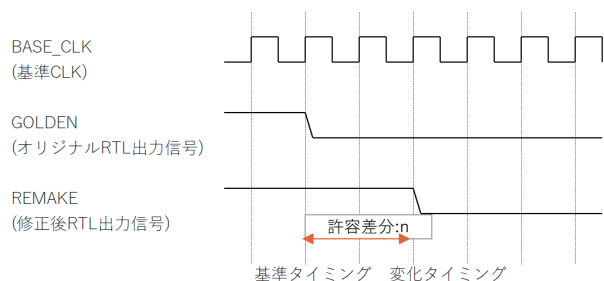
本提案手法では、MPU マスタモデルや PC21 バスマスタモデルなどの各モデルから同時に修正後 RTL モジュールおよび修正前 RTL モジュールに対してアクセスを行う。リードアクセスの場合は、一般的な RTL 検証と同様に各モデルで実施する修正後 RTL モジュールからのリードデータと期待値比較の他に、本提案手法で追加された出力タイミングチェッカモジュールにて修正後 RTL モジュールからの出力と修正前 RTL モジュールからの出力を比較する。モデルによる期待値チェックおよび出力タイミングチェッカモジュールでの出力タイミングの差が一定のクロック数以内であれば問題がないと判定する。モデルによる期待値チェックは一致しているが出力タイミングチェッカモジュールでの出力タイミングの差が一定のクロック数以上の場合は、問題ありと判定し修正後 RTL モジュールの修正を行う。また、モデルによる期待値チェックは不一致であるが出力タイミングチェッカモジュールでの出力タイミングの差が一定のクロック数以内の場合は、既存の検証パターンの問題である可能性が高く、検証パターンを確認し必要に応じて検証パターンを修正する。また、各モデルから外部 SRAM へのアクセスの場合も同様に、出力タイミングチェッカモジュールにて修正後 RTL モジュールおよび修正前 RTL モジュールからの SRAM モデルへの出力を比較し、出力タイミングの差をクロック単位で判定する。上記の他に必要機能については、出力タイミングチェッカにて出力タイミングの比較を行い、出力タイミングの差が問題ないか判定を行う。

本提案手法の特徴の 1 つである出力タイミングチェッカでの出力タイミングの比較方法について以下に述べる。これまでも再三述べたように、非同期回路を同期設計した場合、同期回路の出力タイミングは元の非同期回路の出力と比べて数クロック遅くなる。どの程度までのクロック数の差を許容できるかはシステムに依存するが、MPU やメモリなどのアクセス時間が変わらない範囲であれば問題はない。もし出力の差が許容できない場合には、当該出力タイミングを変更するために内部回路を修正する必要がある。リメイク開発の場合、リメイクに使用する ASIC の半導体プロセスが元の ASIC の半導体プロセスよりも微細化されているケースが多く、その結果内部回路の伝搬遅延が早く



なっているため、内部回路の修正による出力タイミングの変更が可能な場合が多い。また、出力の差が許容できず、かつ出力タイミングの変更が不可能な場合は、顧客にリメイク前の ASIC と同様に使用して頂くことができないため、リメイクを継続するか再考する必要がある。

出力タイミングの比較はアサーションベース検証 (ABV: Assertion-Based Verification) と呼ばれる手法を適用し、動作クロック単位で自動的に比較する。本来、アサーションベース検証では回路内部やインタフェース信号の挙動を、SVA (System Verilog Assertion)、PSL (Property Specification Language) といったアサーション専用言語で記述する必要がある<sup>9)</sup>。本提案手法では、内部回路やインタフェース信号の挙動はオリジナル RTL の内部信号や出力信号を使用するため、新たに SVA や PSL などのアサーション言語で記述する必要はない。本提案手法で必要な記述を図 9 で示す。



```
unique_name : assert property(
  @(posedge BASE_CLK)
  $fell(GOLDEN) | => ##[0:n] !REMAKE
) else sim_top.err = sim_top.err+1;
```

図 9 出力波形比較のアサーション言語記述例

図 9 に示すように、修正後 RTL と修正前 RTL の出力をクロック毎に比較し、何クロックまでの信号の差を許容するかを記述するのみである。このため、アサーション記述言語の習得に苦勞することなく、比較的容易に適用することができる。

このように本提案手法では、非同期回路である修正前 RTL の出力と同期回路に修正した RTL との出力をクロック毎に比較することにより、機能の互換性および性能の互換性を数クロックの範囲内で担保する。

### 3.4 検証網羅性の確保について

最後に検証網羅性の確保について述べる。「2.2.1 非同期回路の問題」でも述べたが、非同期回路は個体ばらつきや温度および電圧変化を考慮したタイミングを含めて検証する必要があり、RTL 検証ではそれらを考慮した検証ができないため代表的なタイミングの組み合わせのみを検証し、RTL 検証での網羅性はあまり重要視されなかった。一

方、一般的に同期回路は回路構造に合わせて、適切なクロック単位でのタイミングの組み合わせで検証パターンを用いて検証した場合、ほぼすべての回路の検証が可能である。また、非同期回路時に考慮が必要であった個体ばらつきや温度および電圧変化の影響については、タイミング検証時に解析ツールで自動的に判定される。

本提案手法では、現存する検証パターンを図 8 で示した検証環境を用いて検証する際にカバレッジ測定ツールを用いて検証網羅性を測定し、一般的な同期回路の検証と同様に十分な検証網羅性が確保されるまで、検証パターンの追加とカバレッジ測定を行い品質の確保を行う。図 10 に Cadence 社の論理シミュレータ Xcelium<sup>10)</sup> を用いたカバレッジ測定結果の例を示す。

Ex	Unit	Name	Overall Average Grade	Overall Covered
		(no filter)	(no filter)	(no filter)
▲		Verification Metrics	94.83%	6280 / 7278 (86.29%)
▶		Types	95.23%	2984 / 3483 (85.67%)
▲		Instances	94.43%	3296 / 3795 (86.85%)
▶		aiop_ibcif	94.43%	3296 / 3795 (86.85%)
▶		ibcifdelaybuf	100%	549 / 549 (100%)
▶		ibcifavalon	95.06%	362 / 475 (76.21%)
▶		ibcifdmac	96.16%	580 / 714 (81.23%)
▶		ibcifarb	97.92%	424 / 474 (89.45%)
▶		ibcifmpu	98.62%	759 / 794 (95.59%)

図 10 カバレッジ測定結果の例

カバレッジの測定結果が 100% であっても検証網羅性が十分であるかはユースケースや実際の RTL から検討が必要であるが、少なくともカバレッジ測定結果が 100% でない場合は、未検証のコードが残っていることを示す。しかし、デッドコードと呼ばれる実際には動作しない不要なコードが元の RTL 内に残っていると、いくら検証パターンを追加してもカバレッジは 100% にならない。カバレッジが 100% にならない場合、原因が検証パターンの不足によるものか、そのコードがデッドコードであるかを設計者が都度判断する必要がある。

本提案手法では、非同期回路である古い ASIC を同期回路に再設計し、本提案手法の特徴である非同期回路である再設計前の RTL の出力と同期回路に再設計した RTL との出力をクロック毎に比較することにより機能の互換性および性能の互換性を数クロックの範囲内で担保し、一般的な同期回路の検証と同様に十分な検証網羅性が確保されるまで、検証パターンの追加とカバレッジ測定を行い品質の確保を行う。その結果、安定した品質で ASIC をリメイクすることができ、顧客への供給責任が果たせると考える。

## 4. 本提案手法の効果と今後の課題

### 4.1 効果

筆者は過去に従来の ASIC リメイク手法でも 4 件の ASIC

リメイクを経験している。その結果を表 1 に示す。

表 1 従来 ASIC リメイク手法による結果

No	用途	実現手段	実機評価結果	市場不具合
1	ラダーエンジン	FPGA	修正多数	1 件発生 出荷停止半年
2	バス制御用 ASIC	ASIC	基板修正 1 件	問題なし
3	ネットワーク用 ASIC	ASIC	ASIC リワーク 2 件 基板修正 1 件	問題なし
4	カウンタ制御用 ASIC	ASIC	問題なし	問題なし

表 1 の 1 番目に示したラダーエンジン用 ASIC のリメイクでは、イタレーションが多数発生すると見込まれたため ASIC へのリメイクを断念し、FPGA (Field-Programmable Gate Array) でのリメイクを実施したが、結果として市場不具合を発生させ、出荷が約半年程度止まるという事態を招いた。また、その後のリメイクでも市場不具合の発生こそなかったが、表 1 の 2 番目に示したバス制御用 ASIC では、市場投入前の実機評価で不具合が判明し、ASIC の作り直しや基板での対応を行う必要があった。表 1 の 4 番目のカウンタ制御用 ASIC のリメイクは、本提案手法で述べた修正前 RTL モジュールとの出力比較は行っていないが、内部回路は同期回路に修正を行っており、従来手法から今回の提案手法に移る過渡期の ASIC リメイクである。

筆者らは、本提案手法を用いてすでに 6 件のリメイクを ASIC で実現し、リメイクした ASIC を搭載した商品を市場投入した。その結果を表 2 に示す。

表 2 本提案 ASIC リメイク手法による結果

No	用途	実現手段	実機評価結果	市場不具合
1	バス制御用 ASIC	ASIC	問題なし	なし
2	バス制御用 ASIC	ASIC	問題なし	なし
3	ネットワーク用 ASIC	ASIC	問題なし	なし
4	バス制御用 ASIC	ASIC	問題なし	なし
5	バス制御用 ASIC	ASIC	問題なし	なし
6	バス制御用 ASIC	ASIC	ASIC リワーク 1 件	なし

表 2 に示すように、本不具合も含めて市場投入後の不具合報告は発生していない。このように本提案手法は、安定した品質を確保できており有効性の高い開発手法であると

考える。

一方で表 2 の 6 番目のバス制御用 ASIC では、市場投入前に実施した実機評価で 1 件の不具合が発見され、ASIC をリワークしなければならなくなった。これは図 7 で示した検証環境にて修正前 RTL と修正後 RTL の差があることを担当者は認識していたが、その差分は問題ないと誤った判断をしたためである。本不具合は、修正前 RTL と修正後 RTL の差が無いように修正した後、市場投入された。本件は修正前 RTL と修正後 RTL の差がある場合の判断は、複数人で確認し判断しなければならないという示唆であると考ええる。

## 4.2 今後の課題

筆者らのチームは現在、本提案手法を用いて、さらに 2 件の ASIC リメイクを実行中である。そこで回路規模の拡大に伴い、網羅性を確保するために必要な工数が大幅に増加していることが確認されている。この増加の主因として、「3.2 検証網羅性の確保について」の最後に述べたデッドコードと称される、実行されることのないコードの存在が挙げられる。デッドコードは、特定の条件が恒常的に偽であるために実行されず、また、プログラムの流れが到達することのないコードブロックを指す。ASIC 開発においては、デッドコードは論理合成の際に自動的に削除されるため、通常、回路の規模や挙動、性能に影響を与えないと見なされ、特に対策は講じられていなかった。このため、検証で網羅されない原因が、検証パターンの検討不足によるものかデッドコードによるものかの判断に多くの工数を要している。今後の課題として、リメイク ASIC の回路規模は拡大する傾向にあり、本提案手法を持続的に適用するためにデッドコードの効果的な検出方法の検討が必要であると考ええる。

## 5. むすび

本稿では、ASIC 生産の中止への対応策として、ASIC リメイクの必要性について述べた。ASIC リメイクの課題として「設計資産の問題」と「技術的な問題」から「ASIC リメイク時の課題」を明示し、これらに対処する新たな手法を提案した。提案手法では、古い ASIC に広く採用されている非同期回路を、現在の ASIC の主流である同期回路にリメイクする手続きを具体的に提示し、先に論じた課題に対して柔軟で効果的な対策を講じると同時に、機能と性能の互換性を担保しつつ同期回路への移行が可能であることを示した。具体的な成果として、提案手法を用いて 6 つの ASIC をリメイクし、安定かつ高品質な製品を提供できたことを確認した。これにより、半導体業界が迅速に変化する状況に柔軟かつ持続的に対応し、安定かつ高品質な ASIC を供給することで、顧客への商品の供給責任を果たすことができると考える。



ただし、回路規模の増加に伴う検証工数の増加原因として、デッドコードの存在が浮かび上がった。デッドコードの効果的な検出手法が今後の課題として残る。将来的には、デッドコードの検出手法の向上に加え、提案手法を持続的に適用することでより効率的な開発手法を構築する必要があると考える。

## 参考文献

- 1) 日本経済新聞社. “富士通、三重の半導体工場売却 台湾 UMC に 576 億円。” 日本経済新聞. <https://www.nikkei.com/article/DGXMZO32425480Z20C18A6TJC000/> (Accessed: Feb. 07, 2024).
- 2) ルネサスエレクトロニクス株式会社. “山口工場閉鎖のお知らせ.” ニュース. <https://www.renesas.com/jp/ja/about/press-room/notice-regarding-closure-yamaguchi-factory> (Accessed: Feb. 07, 2024).
- 3) 商務情報政策局. “半導体・デジタル産業戦略.” 経済産業省. <https://www.meti.go.jp/press/2023/06/20230606003/20230606003-1.pdf> (Accessed: Feb. 07, 2024).
- 4) 東芝デバイス & ストレージ株式会社. “メタステーブル対策.” CMOS ロジック IC 使用上の注意. <https://toshiba.semicon-storage.com/jp/semiconductor/knowledge/e-learning/cmos-logic-usage-considerations/usage-11.html> (Accessed: Jan. 22, 2024).
- 5) Intel Corporation. *AN 42: Metastability in Altera Devices*, 第 4.0 版. (2019). Accessed: Jan. 22, 2024. [Online]. Available: <https://www.intel.com/content/www/us/en/content-details/653636/an-42-metastability-in-altera-devices.html?wapkw=an42>
- 6) 株式会社 半導体理工学研究センター 設計技術開発部 IP 技術開発室, *RTL 設計スタイルガイド Verilog-HDL* 編. 初版. 株式会社 半導体理工学研究センター, 2003, 1章, pp. 17-18.
- 7) Synopsys, Inc. “Formality 等 価 検 証.” Synopsys. <https://www.synopsys.com/ja-jp/implementation-and-signoff/signoff/formality-equivalence-checking.html> (Accessed: Feb. 21, 2024).
- 8) Synopsys, Inc. “SpyGlass CDC.” Synopsys. <https://www.synopsys.com/ja-jp/verification/static-and-formal-verification/spyglass/spyglass-cdc.html> (Accessed: Feb. 21, 2024).
- 9) J. Bergeron 他, *ベリフィケーション・メソッドロジ・マニュアル*. 初版. CQ 出版社, 2006, pp. 53-112.
- 10) Cadence Design Systems, Inc. “Xcelium Logic Simulator.” Cadence. [https://www.cadence.com/ja\\_JP/home/tools/system-design-and-verification/simulation-and-testbench-verification/xcelium-simulator.html](https://www.cadence.com/ja_JP/home/tools/system-design-and-verification/simulation-and-testbench-verification/xcelium-simulator.html) (Accessed: Feb. 21, 2024).

## 執筆者紹介



古石 憲男 FURUISHI Norio

インダストリアルオートメーションビジネス  
カンパニー

商品事業本部 コントローラ事業部

第 2 開発部

専門：情報工学

本文に掲載の商品の名称は、各社が商標としている場合があります。Synopsys、Formality および SpyGlass は、Synopsys, Inc. の米国およびその他の国における登録商標または商標です。Cadence および Xcelium は、Cadence Design Systems, Inc. の米国およびその他の国における登録商標または商標です。Altera は Intel Corporation またはその子会社の商標です。