画像処理用LSIの設計技術高度化

The Advance of Design Technology of LSI for Image Processing

遠藤靖彦＊ Yasuhiko ENDO
髙木　潔＊ Kiyoshi TAKAGI
工藤俊樹＊ Toshiki KUDO
江口俊哉＊ Toshiya EGUCHI

要旨

MFP（Multi Function Printers）やプリンタなどのイメージング機器開発では，常に高画質化を求めて画像処理アルゴリズム検討が行なわれており，製品競争力を決める重要技術となっている。その画像処理アルゴリズムをLSI化して製品に搭載するための効率的な設計技術として，高位合成技術を適用した開発手法を提案した。

高位合成技術を効果的に利用するため，画像処理回路のインターフェースを標準化したプラットフォーム型の設計手法を選択し，更にインターフェース回路の検証にアサーション検証技術を適用することで，効率的な開発環境を確立した。この開発手法の考え方と，効率化を実現した基盤技術について紹介する。

Abstract

In the development of imaging systems such as multi-function printers (MFPs), image processing algorithms are consistently investigated to enhance the quality of images; these algorithms are important in determining product competitiveness. A development method, to which high-level logic synthesis technologies were applied, was proposed as an efficient design technology to pack image processing algorithms onto a single LSI integrated with imaging systems.

A platform type design method, with standardized image processing circuit interfaces, was chosen to apply the high-level logic synthesis technologies effectively, and assertion-based verification technologies were applied to verify their interfaces. As a result, an efficient development environment was established. In this article, the concept of this development method and the fundamental technologies realizing efficiency are introduced.

＊コニカミノルタテクノロジーセンター㈱
システム技術研究所　アーキテクチャ開発室

1　はじめに

MFP（Multi Function Printers）やプリンタの画像処理用LSIの開発では，大規模化と複雑化が急速に進み，開発工数の増加が課題となっており，高画質化と高機能化の要望に応え続けていくため，設計の効率化が強く求められている。

そこで，コニカミノルタの画像処理技術の強みを生かし，LSI開発効率を向上する技術として，アルゴリズム開発で使われるC言語から，LSI設計用のRTL（Register Transfer Level）記述を自動生成する”高位合成技術”に注目した。設計工程への適用を検討した結果，開発効率化に有効であることが確認され，すでに製品開発への適用が始まっている。現在では，この技術をより効果的に利用するため，回路検証作業を効率化する”アサーションベース検証技術”を組み合わせた，回路設計のプラットフォーム化を進めている。

2　画像処理回路の設計

2. 1　画像処理回路設計の課題

従来，高画質化を目指した画像処理アルゴリズムの回路化検討の作業では，まず，アルゴリズム設計者がC言語で設計した様々な画像処理アルゴリズムを検討した後に，回路設計者にRTL設計を依頼するための仕様書を作成していた。回路設計者は，その仕様書を元にRTL設計を行い，設計した回路と，アルゴリズムの等価性を検証後，FPGA（Field Programmable Gate Array：回路書き換え可能なLSI）を搭載した実基板で，回路性能（画質，回路規模，速度）を確認していた。

これらの作業工程に，通常1週間程度の時間がかかるが，最終的に回路性能が仕様を満たさなかった場合，最初のアルゴリズムを修正する手戻りが発生する。このように，アルゴリズムの回路化検討は，この作業を繰り返すことで最適化していく工程のため多くの時間を要しており，これを効率化する設計技術が求められていた。

2. 2　高位合成技術

前述のアルゴリズムの回路化プロセスを効率化する手法として，高位合成技術の利用が近年注目されている

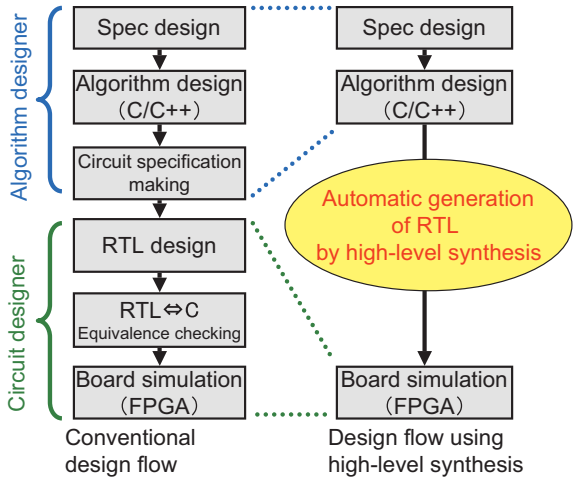

Fig.1 Conventional design flow and design flow using high level synthesis

(Fig.1)。高位合成技術は，C言語から直接RTL記述を自動生成することができ，画像処理アルゴリズムのような演算処理系のRTL記述生成に適しているが，データの入出力を行う転送制御系の回路には不向きであるという特徴がある。

2. 3　高位合成技術の適用検討

画像処理回路設計の効率化のために，高位合成技術導入の検討を行った。その結果，実製品開発への適用にはアルゴリズムのC言語記述の最適化と，画像入出力制御方法の明確化の2つの課題を解決することが重要であると判断した。

2. 3. 1　アルゴリズムのC言語記述の最適化

高位合成で生成されたRTL記述と，従来手法の回路設計者による手書きRTL記述との性能比較を行った（Table 1)。アルゴリズム設計者が，生成される回路構造を考慮せずに設計したC言語記述を高位合成すると，回路性能を満たせないRTL記述が生成される場合が多い。今回検討した回路の例では，回路規模が手書きRTL記述に比べ40%増加した（Table 1のB)。

Table 1 Comparison of design performances

	RTL	Size (Library area)	Size ratio (Hand writing is assumed to 100.)	Speed ratio (%)
A	Hand writing	2259	100	105
B	High-level synthesis (No optimaization)	3167	140	115
C	High-level synthesis (Ciruit optimization)	2354	104	118

回路規模増加要因のひとつは，通常のアルゴリズム開発のC言語記述で多用するループ処理を高位合成すると，ループ回数分の同一回路が生成されることが挙げられる。これに対しては，多重ループ（ループ処理の中でループ処理する）の分解により回路規模を削減できた。同様に下記のC言語の修正により回路規模と動作速度を最適化した。

・シリアル動作の並列化（高速化）
・不要なループ処理の削減（回路規模削減）
・マクロ使用による機能の共通化（回路規模削減）

上記の最適化の結果（Table 1のC)，手書きRTLとほぼ同性能の回路を生成することができた（動作速度12%アップ，回路規模+5%以内)。このような最適化手法は，C言語の記述スタイルガイドとして整備し，高位合成を利用した設計手法の標準化を進めている。

2. 3. 2　画像入出力の制御方法

LSI開発では，様々な画像処理アルゴリズムが回路化された画像処理部を搭載する。同時に処理すべき複数の画像処理部が，並列に動作する場合，レイテンシ（信号を回路に入力後，出力されるまでの遅延時間）を合わせなければならないため，調整回路が必要となる。レイテンシ調整回路は制御系の回路であるため，高位合成では生成できず，回路設計者が手書きのRTL記述による設計を行っている。

従来は，画像処理機能毎にレイテンシ調整回路が画像処理内部に含まれていたため，C言語修正よる高位合成時に画像処理部のレイテンシが変化し，レイテンシ調整回路も，その都度修正が必要であった。

そこで，レイテンシ調整回路を画像入出力部として画像処理部から分離した以下の構成に統一し，レイテンシを任意に設定する機能を付加することで，汎用的な画像入出力制御を実現した（Fig.2)。

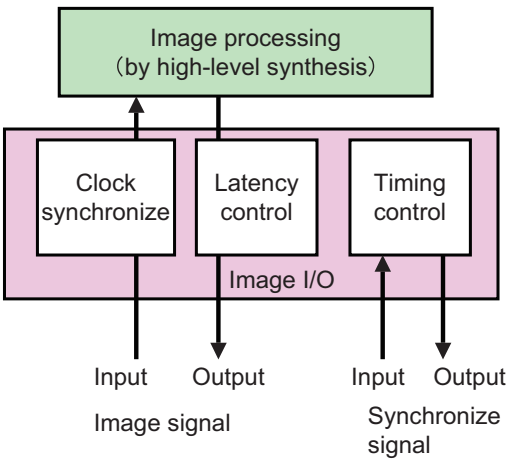

Fig.2 Image I/O interface

レイテンシ調整部は，外部からのパラメータ設定により，回路修正せずに任意の遅延を発生する。また，クロック同期部は，画像入力信号を画像処理部の動作クロックに同期させ，さらにタイミング調整部は，同期出力信号を画像出力信号のレイテンシに合わせる。

その結果，画像出力信号と同期出力信号を同じタイミングで出力することを可能にした。

2. 4　高位合成技術の適用効果

画像処理部と画像入出力部を分離し，画像処理部の設計に高位合成技術を適用することで，通常1週間を要していた画像処理回路の設計が，1日で出来ることが確認された。そして，この設計プロセスの改善により，限られた開発期間に従来よりも多くの画像処理のアイデアを試し，高画質化を追求できるという効果が期待されることから，実製品向けの画像処理回路開発への適用が始まった。

3　LSI設計におけるプラットフォーム化構想

3. 1　画像処理部とシステムバスの接続

画像処理部とプロセッサーやメモリを接続するバスなどで構成されるLSIシステムを設計する場合，アーキテクチャの基本的土台となるプラットフォームを用意することにより，汎用性の高い効率的なシステム設計が実現できる。プラットフォーム構築にあたり，高位合成した画像処理部のRTL記述と，画像入出力部のRTL記述をシステムバスへ接続するには，インターフェース部を新たに設ける構成が望ましい（Fig.3）。

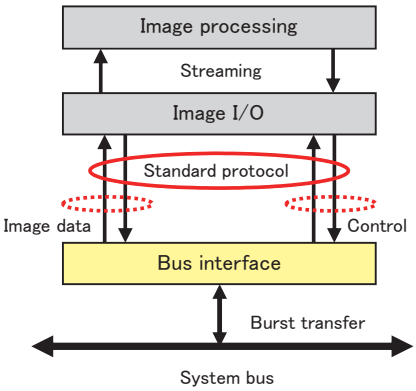

Fig.3 Interface control block diagram

インターフェース部の機能と，3つの処理部の接続について説明する。

インターフェース部は，システムバスの規格化されたデータ転送手順に従い，バースト転送と呼ばれる一定量の画像データをまとめて転送する機能を持つ。インターフェース部と画像入出力部間は，固定ビット長画像信号と同期信号による，標準化されたプロトコル（通信制御手順）で転送する。また，画像入出力部と画像処理部間は，画像信号の連続するストリーミングデータで転送する。

このように3つの処理部で構成することにより，インターフェース部は同一システムバスに対して共通の回路となり，流用が可能となるため設計工数を削減できる。

3. 2　プラットフォーム型の設計構想

高位合成した画像処理部のRTL記述を活用し，システムレベルでプラットフォーム化した構成図を2つの異なる画像処理機能を持つ例としてFig.4 に示す。

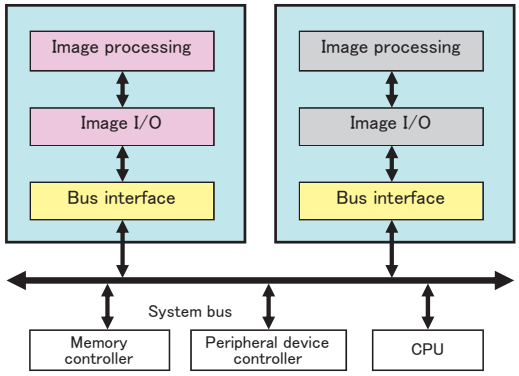

Fig.4 Platform block diagram

プラットフォームは画像処理部，画像入出力部とインターフェース部を1つのブロックとして，システムバスに接続する構成となる。複数のブロックは，共通のインターフェースを介してシステムバスに接続しているため，画像処理機能の追加や変更が容易になる。また，各画像処理部の並列動作が可能になるため，画像処理スループットの向上も期待できる。

4　インターフェース部の設計

4. 1　インターフェース部設計の課題

近年，画像処理部の高機能化と高速化に伴うインターフェース部の複雑化で検証作業が増加する傾向にあり，最近の設計事例では，検証作業が設計開発工数の70%以上に達する場合もある。インターフェース部のような制御回路では，信号間の関係をシミュレーション波形で目視確認（モニタリング）する必要があるため，膨大な工数がかかっており，検証漏れを起さず効率的に検証する新たな手法が求められていた。

4. 2　アサーションベース検証

アサーションベース検証は，検証対象に期待する動作（アサーション）を記述し，ツールにより回路のエラー動作を自動チェックした結果をレポート出力する，新規の検証方法である。波形モニタリングを自動化できるため，大量のテストパターンの検証が可能になり，検証のレベルアップ効果が期待できる。アサーション記述とモニタリングの例をFig.5 に示す。

Fig.5 のアサーション記述は，x_AckのLow期間にx_ReqがHighになることを期待した記述であり，シミュレーション実行中にアサーション記述に違反した事象が発生すると，エラーを自動検出する。一方，Fig.5 の波

形モニタリングの場合は，膨大な波形から違反箇所を目視で検出する必要があり，検証漏れの要因となっていた。

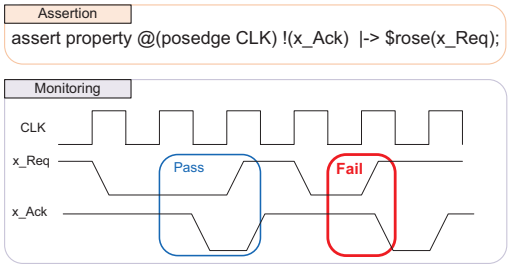

Fig.5 Example of assertion and monitoring

4. 3　アサーションベース検証の適用検討

検証漏れの抑止効果を確認するために，MFP向け画像処理回路のインターフェース部にアサーションベース検証を適用し，バグ（回路不具合）の検出を試みた。適用にあたりアサーション項目の抽出方法と，テストパターンの作成方法の2つの課題を検討した。

(1) アサーション項目の抽出方法

アサーションベース検証は，すべてのアサーション項目をチェックすることを検証のゴールとするため，回路仕様から抽出するアサーション項目の質と量が，検証品質を決定する。検証者が波形モニタリングで着目する箇所は，バグが発生しやすいポイントであるため，バグ検出の可能性が高いアサーション項目となる。

しかし，回路仕様書の記載方法や検証者のスキルや思い込みにより，アサーション項目の抽出漏れが発生する可能性がある。そこで検証者のスキルに依存せず，システマチックにアサーション項目を抽出する方法を検討した結果，インターフェース回路の対象信号が立ち上がりと立ち下がりを起こす，関係信号の条件を抽出しリスト化した（Table 2）。これにより，インターフェース部のアサーション項目を抽出し，実際の波形モニタリングで検証者が着目するポイントをカバーできるようになった。

Table 2 Example of assertion list

No.	Target	Trigger	Operation	Condition	Assertion label
1	x_Req	x_Ack	fall	x_Ack==1	ast_01
2	x_Req	x_Ack	rise	x_Ack==0	ast_02
3	x_Ack	x_Req	fall	x_Req==0	ast_03
4	x_Ack	x_Req	rise	x_Ack==0	ast_04

(2) テストパターンの作成方法

従来手法では，テストパターンは検証技術者が回路の基本動作を想定して作成している。そのため，検証漏れは小規模回路では起こりにくいが，複雑なインターフェース部で発生する可能性が高い。特に，極めて稀な入力条件の組み合わせで発生するバグはコーナーケースバグといい，検証漏れによって検出されない可能性が高い。そこで対策として，検証者が想定していない回路の動作状態を検証できる，ランダムなテストパターン生成手法を適用した。

適用した回路に対し，4項目のパラメータをランダムに生成すると，全ての組み合わせは，17,179,869,184 (2^{34}) 種類であり，回路仕様外の無駄なテストパターンが大量に含まれることになり，検証時間が膨大になる。

$(para1)2^4 \times (para2)2^{13} \times (para3)2^4 \times (para4)2^{13} = 2^{34}$

そこで，仕様の範囲内で起こりえるテストパターンをランダムに生成する制約付きランダム検証手法を採用した。例えばpara2は，8192 (2^{13}) 種類の値を設定できるが，仕様上は60種類の値しか取りえないため，ランダム生成のパラメータ範囲を制約し，検証の効率化を図った。

4. 4　アサーションベース検証の適用効果

前述した画像処理回路のインターフェース部の検証において，従来手法では42種類のテストパターンによる検証を実行し，約60時間のモニタリング作業を行っていた。それに対し，制約付きランダム検証とアサーションベース検証によるモニタリングの自動化により，従来手法と同等の時間で3000種類のテストを実行できた。その結果，従来の手法で見つからなかったコーナーケースバグを検出し，効率的に検証漏れを抑止できる効果を確認できた。

5　まとめ

高画質化を追求するための画像処理アルゴリズム検討と，その回路設計を効率化する手法として高位合成技術に注目し，この技術を効果的に利用するために，以下の技術検討を行い，実開発への適用効果を確認した。

①高位合成技術
- 回路規模と動作速度を最適化し，手書きと同等性能のRTL記述を生成するC言語記述の手法を確立
- 画像処理部と画像入出力部の分離により，アルゴリズム検討後の回路設計を効率化

②プラットフォーム型の回路設計
- 画像処理部とシステムバスの接続を標準化することで，回路の再利用を可能にし，LSI開発を効率化

③アサーションベース検証
- アサーション項目の効率的な抽出方法を考案
- 制約付きランダム検証による，効率的なテストパターンの生成手法により，コーナーケースバグを検出

以上の検討により確立された技術は，LSI開発を効率化するための設計基盤技術として，既に社内の製品開発へ適用が始まっている。今後は，これらの技術をより効果的に活用し，設計資産の再利用化を進めることで，大規模化するLSI開発に対応していく。