

画像データを使用した ASIC 設計の論理検証

—Konica7728 用画像判別ゲートアレイの開発事例紹介—

ASIC Design Verification Using Image Data as Inputs for Logic Simulation

新妻 徹也*

Niitsuma, Tetsuya

東井 满男†

Azumai, Mitsuo

Whereas the quality of ASIC design becomes better with use of HDL these days, there are no good methods for verifying HDL description itself. In this paper, we tested the method which uses image data as inputs for logic simulation of the ASIC and compares the simulation output with the data produced by the program for developing image process algorithm. The method verifies the correctness of the ASIC design with at least the same accuracy of the algorithm design.

1 背景

高速な画像処理が要求される、デジタル複写機、プリンタ、FAXなどでは、画像処理アルゴリズムをハード化し、処理を高速化することが必要である。

汎用的な画像処理については、市販の DSP[‡]を利用することも一部可能である。しかし、製品の差別化を図るために、特徴ある画像処理アルゴリズムを採用しようとすると、新たに LSI を設計することが必要となる。例えば、デジタルカラー複写機 Konica7728 の画像処理部には、数千ゲートから数万ゲート規模の新規に設計されたゲートアレイが十数種類も搭載されている。

大規模な ASIC[§]を開発する場合に、必要とされる工数のうち、テストパターン作成と論理検証に半分以上が費やされている[¶]。このような現状から、テストパターン作成と論理検証にかかる工数を削減する強い要求がある。最近では、ATPG[¶]などのツールの発達によって、テストパターン作成の自動化がはかられているが、これらのツールは、回路の仕様となる論理記述から故障検出用のテストパターンを発生するものであり、論理記述の正当性については、検証できない。

画像処理用 ASIC の開発では、画像処理アルゴリズムの開発者と、そのアルゴリズムをハード化する ASIC 開発者とは、別人であることが多い。

ところが、従来は、ASIC 論理検証のテストパターンは ASIC 開発者が作成していた。このため、このパターンで検証できるのは、ASIC 開発者の理解した回路仕様に対する、設計された回路の正当性であり、画像処理アルゴリズムに対する正当性ではなかった。言い替えれば、アルゴリズム開発者が作成した仕様書の誤記や、ASIC 開発者による仕様書の誤読については、これを検証することが

できなかつたのである。

2 目的

画像処理アルゴリズムに対する ASIC 設計の正しさを論理検証するために、以下のような手法を試みた。すなわち、画像データを論理シミュレーションの入力データとし、出力データを、画像処理アルゴリズム検討用のソフトの出力と比較することによって、論理検証を行なう方法である。本論文では、この方法によって論理検証を行なった ASIC の開発事例として、画像判別ゲートアレイを取り上げ、事例を紹介する中で、本手法の有用性と有効性について明らかにする。

3 画像判別ゲートアレイの開発

画像判別アルゴリズムの開発からそのハード化までの一連の流れについて始めに説明する。

3.1 画像判別とは

画像にフィルタリングなどの処理を加える場合、画像をその特徴をもとに、複数の領域に分割し、領域ごとに異なった処理を行なうことによって、より品質の高い画像を得ることができることが多い。この、画像をその特徴をもとに複数の領域に分割する手法を画像判別と呼んでいる。

デジタルカラー複写機 Konica 7728 は、ビジネスカラー文書の複写をターゲットとしており、カラー印刷や写真的複写品位が高いことはもちろんだが、特に黒文字の再現に高い要求がある。なかでも、黒文字を黒トナーのみによって再現することが、黒文字品位を高める上で、重要な要素であった。

今回開発した画像判別ゲートアレイはこの画像処理機能を達成するためのもので、画像を文字領域とそれ以外の領域に分割する信号を出力するものである。この出力は、別に設けられた色判定回路によって得られた無彩色判別信号と組み合わせて黒文字判別信号として出力される。この出力によって、黒文字と判別された領域では、

* 機器開発統括部第二開発センター

† 機器開発統括部第一開発センター

‡ Digital Signal Processor

§ Application Specific Integrated Circuit

¶ Automatic Test Pattern Generator

画像を黒トナーのみによって再現し、すっきりとした黒文字画像を得ることができる。

3.2 画像判別アルゴリズム開発

3.2.1 アルゴリズム開発の前提

民生用機器である複写機には、製品コストとの兼ね合いでから、使用できるメモリの容量に制限がある。このため、局所的な画像を参照しながら、リアルタイムに画像を処理していく形をとることが多い。この方法は、ラインセンサによって画像をスキャンする画像の入力方法、レーザービームによって感光体をラインごとにスキャンする画像の形成方法とも整合性がある。本アルゴリズムも、注目画素近傍を参照しつつ画素ごとに領域を判別していく形とした。

3.2.2 画像判別手法

Fig. 1 に画像判別手法の概要を示す。画像からエッジ領域と網点領域をそれぞれ独立に抽出し、「エッジ領域」かつ「網点領域でない」部分を文字領域として判別するのが基本的な考え方である。エッジ領域の抽出は、Fig. 2 に示すように、隣接画素間の差の最大値をもとに行なっている。

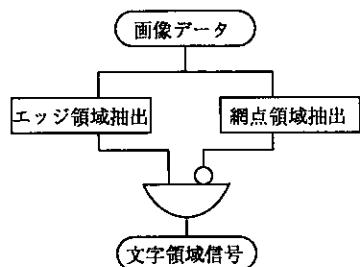


Fig. 1 画像判別手法の概要

判定の確度を高めるため、コンボリューションフィルタやガンマ変換による前処理、領域の拡張処理などが加えられている。



Fig. 2 エッジ領域抽出

一方網点領域の抽出は Fig. 3 のような手順で行なう。印刷物の網版のスクリーン角度を特徴として抽出するため、右斜め方向のエッジと左斜め方向のエッジとを独立

|| Hardware Description Language

にフィルタで検出し、両方の成分を持つ画素を網点候補画素とする。さらに、平滑、拡張処理を加えて、近傍に網点候補画素の多い画素を網点領域と判別する。

以上のアルゴリズムは、ワークステーション上に C 言語を用いてプログラムされ、画像データを入力とする画像処理のシミュレーションを行なってその正しさを検証した。

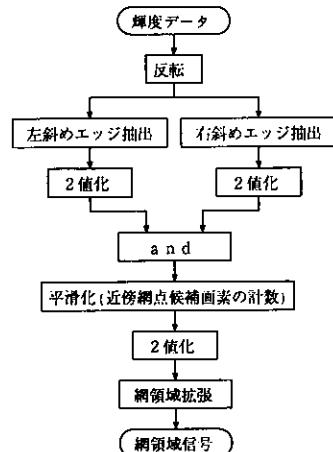


Fig. 3 网点領域抽出

3.3 画像判別ゲートアレイの設計

以上のアルゴリズムの検証後、画像処理アルゴリズム開発者は、アルゴリズムのフローチャートや、ハードのブロック図などから成る ASIC 仕様書を作成する。ASIC 開発者は、この仕様書によりアルゴリズムの理解をはかり、アルゴリズム開発者の作成した、C 言語のソースプログラムを参考にして、HDL¹による回路記述を行う。この過程で、アルゴリズム開発者の提案したブロック図は、ASIC 開発者によって大幅に修正され、回路規模と遅延が改善されていく (Fig. 4)。エッジ抽出部を例にとり、この様子を示す。

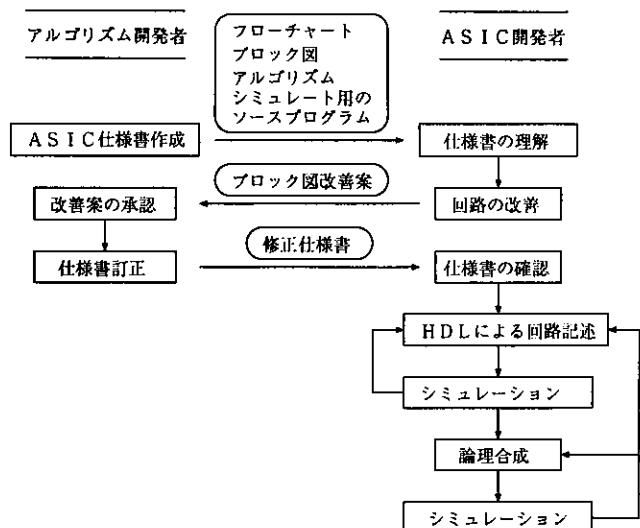


Fig. 4 ASIC のデザインフロー

Fig. 5 は、アルゴリズム開発者が最初に描いたエッジ抽出部のブロック図で、Fig. 2 に示したアルゴリズムフローを素直にハード化したものとなっている。一方 ASIC 開発者によって改善されたブロック図は Fig. 6 のようなものとなった。コンボリューションフィルタ部と隣接画素間の差の最大値を抽出する部分を融合することにより、網点領域抽出部で使用している FIFO** メモリ 4 ライン分が共用され、総合的なコストと I/O 数の大幅な削減が図られている。

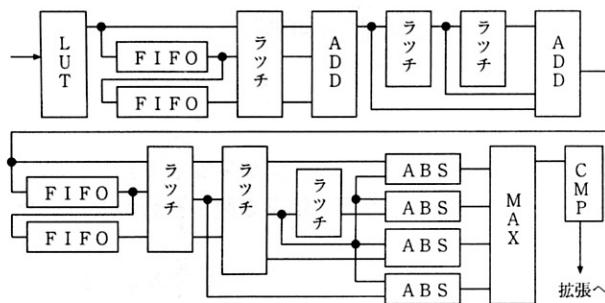


Fig. 5 アルゴリズム開発者の設計したハードブロック図

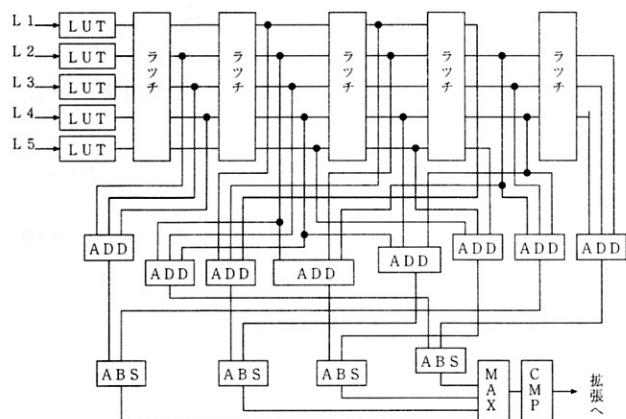


Fig. 6 ASIC開発者により改善されたブロック図

このような最適化プロセスは、アルゴリズム設計者の了解を得て進められて行くものであるが、アルゴリズムと、設計された ASIC とが、論理的に等価であるかどうかが判断しにくくなり、設計誤りが混入する可能性を否定できない。

3.4 論理検証

HDL に対し主に論理検証のためのシミュレーションを行うとともに、使用ゲート数、I/O 数の見積りを行い、ゲートアレイを選択する。今回は、 $0.8 \mu m$ ルール CMOS シリコンゲートメタル 2 層配線、160pin QFP、敷き詰めゲート数 4 万ゲートのものを使用した。HDL の論理検証後、論理合成ツールによって論理合成を行い、合成された回路に対し、主にタイミング検証のためのシミュレーションを行う。本ゲートアレイでは、論理合成後のゲート数は、約 2 万ゲートであり、従来の手法を用いる

** First In First Out

と、テスト設計の工数は 264 人日と見積もられる。

我々は、画像データをシミュレーションの入力とし、出力を画像処理アルゴリズム検討用プログラムのシミュレーション結果と比較することにより、アルゴリズムに対する、ASIC の動作の正当性を検証する方法を試み、検証の確度を高めるとともに、論理検証工数の大幅な削減をはかることができた。HDL の使用や、スキャンパス手法の適用ともあいまって、テスト設計の工数は、最終的には 75 人日まで低減された。

4 画像データを利用した論理検証方法

次に、実際に用いられた画像データを用いた論理検証プロセスについて詳細に説明する。

4.1 論理合成による ASIC 開発

本論文冒頭に述べたように、テストパターン作成やシミュレーションの実行には、ASIC 開発工数の半分以上が当たられており、この部分の工数削減を行なうことに対する、強い要求がある。我々は、この要求を満たすために、HDL 論理合成による ASIC 開発ができる環境を導入した。論理合成のツールを使用することにより、論理記述から回路合成までの間での論理誤りの混入は、根絶されることになる。論理合成ツールは、米シノプシス社の Design Compiler、シミュレータは米ケイデンス社の Verilog-XL、HDL は Verilog-HDL である。また、画像処理用 ASIC に対しては、これらの汎用ツールの特徴をいかして、専用の環境を構築してきた。

4.2 シミュレーションデータの作成

スキャナから読み込んだ画像をシミュレーションデータとして用いる。今回は、Fig. 7 に示す 512×512 の 8 bit モノクロ輝度画像を使用した。



し。たけき者も
只春の夜の夢の
れる人も久しか
祇園精舎の鐘
の塵に同じ。遠
ほろびぬ、偏に
し。たけき者も
只春の夜の夢の
れる人も久しか
祇園精舎の鐘
の塵に同じ。遠
ほろびぬ、偏に
し。たけき者も
口をのぞの夢の

Fig. 7 シミュレーション用画像データ

4.3 シミュレーションモデルの作成

ASIC を記述した HDL とともに、本 ASIC が動作の際必要とする数種類のメモリのシミュレーションモデルも独自に作成した。

4.4 アルゴリズム検討用ソフトによる画像処理

シミュレーション

上記画像をアルゴリズム検討を行なったソフトで処理し出力データの期待値を得る。(Fig.8)



Fig.8 アルゴリズム検討用ソフトの出力

4.5 Verilog シミュレータによるシミュレーション

Fig.8 の画像データをフォーマット変換して、Verilog シミュレータに、シミュレーション入力データとして取り込ませる。シミュレーション終了後、出力データを 16 進数のデータファイルとして取り出す。

4.6 出力の比較

出力データファイルをポストスクリプトに変換し、画像としてプリントアウトする (Fig.9) とともに、アルゴリズム検討用ソフト出力とのコンピュータ上でのデータ照合を行なう。ハードの遅延分だけ画像をシフトし、両者の差分をとることで評価した結果、両者は完全に一致することがわかった。

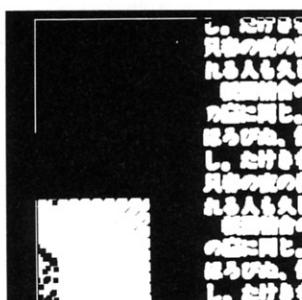


Fig.9 ASIC のシミュレーション出力

画像処理アルゴリズムの性質によっては、設計された ASIC の出力とアルゴリズム検討用プログラムの出力が、厳密には一致せず、しかもそれで ASIC 設計としては正当であることも考えられる。Fig.10 は、変倍ゲートアレイのシミュレーション結果の比較を差分のヒストグラムによって示したものであるが、両者は厳密には一致していない。このような場合にも、画像とヒストグラムを見ることによって、正当な理由^{††}さえあれば、機能としては十分であることが判断できる。これも、入出力が画像であることの利点である。

†† この場合は、ソフトとハードの有効数字の違いによって出力に差が生じることがあらかじめ予想された。

差分	度数
-1	18
0	243822
1	23

Fig.10 シミュレーション結果同士の差分のヒストグラム

5 結 果

アルゴリズム検証と同様の確からしさで、ASIC 設計の論理検証を行えた。さらに、この手法の適用などにより、テスト設計工数を従来の 30% にすることができた。この手法による、工数の削減への貢献度は、削減工数全体の 3 割程度と推定される。

6 考 察

従来、画像データによるシミュレーションが行なわれてこなかった理由と、今回の結果を照らし合わせて考察する。

第一の理由は、検証ツールの能力不足である。最近は、コンピュータの高速化とソフトの充実で、今回検証を行ったような大規模な回路に対するシミュレーションが可能となってきている。

第二の理由は、従来行えたシミュレーションは、小規模な回路に対してのみであり、そのような小規模な回路に対しては、テストパターンの生成が容易であったことがあげられる。ツールの進歩とともに大規模な回路に対してのシミュレーションが可能になった一方、テストパターンの作成は困難となってきたわけであるが、画像処理用のハードの場合、本論文中で詳細に述べたように、アルゴリズム検討自体に画像処理のシミュレーションを多用しており、その出力結果を容易にテストパターンへと転用できる利点がある。

第三の理由は、ライブラリモデルの不足である。従来は、シミュレーションモデルの作成には大変な工数がかかったが、HDL の使用により容易になった。

7 将来の展望

アルゴリズムのハード化の、もう一步進んだ形は、RT^{††} レベルではなく、アルゴリズム検討にも使用できる、より抽象度の高い動作レベルから、直接論理合成功能行えることであろう。こうなれば、もはやアルゴリズム設計と ASIC 設計の違いを検証する目的のシミュレーションの必要性はない。

今回は、メモリを含めた一種のシステムシミュレーションを行ったわけであるが、さらに、ソフトウェアを含むより広い範囲のシステムシミュレーションが行えるような開発環境整備が重要な課題である。

●参考文献

- 1) ASIC ハンドブック、産業調査会事典出版センター, p. 36, Apr. 1991

†† Register Transfer