

高位合成技術を用いたプリンターエンジン用LSIの開発

Development of LSI ICs for Printer Engines using High-Level-Synthesis

藤森 春 充*
Harumitsu FUJIMORI

橘 優 太*
Yuta TACHIBANA

鈴木 大 地**
Daichi SUZUKI

高橋 厚*
Atsushi TAKAHASHI

要旨

コニカミノルタでは、より高機能な製品をより早くお客様へ提供することを目指している。

様々な機能を搭載するLSI (Large Scale Integration) の開発において、その開発期間を短縮可能な技術である高位合成 (HLS: High Level Synthesis) の活用が進んでいる。コニカミノルタでも、画像処理用LSIの開発では高位合成技術を活用してきた。レーザープリンターエンジン用LSIの開発においても、高機能化が進み、より複雑な入出力信号処理が必要となった事から、高位合成技術をはじめ活用し、新製品への搭載を実現した。

今回、複雑な制御アルゴリズムをもったモーター制御モジュールに対して高位合成を実施している。これにより、LSIの開発を効率的に進める事を目指した。しかし、高位合成したRTL (Register Transfer Level) デザインには、手書きしたRTLデザインでは見る事のない特徴的な記述があり、その特徴が要因となって、レイアウト設計時に配線混雑を引き起こした。この特徴的な記述を生み出した原因は、高位合成ツールの設定や、モーター制御モジュールの構成により、演算モジュールなどが過度に共有され、配線が集中した事にあった。そこで、共有できるレジスタ数の上限値を設定するなど、高位合成ツールに適切な設定を行う事で配線混雑を改善した。

この結果、手書きしたRTLデザインに高位合成したRTLデザインを組み込み、LSI化を進める中で実施したいくつかの改善手法を、高位合成技術を活用した効率的で新しい開発フローとして確立する事ができた。

Abstract

Konica Minolta is committed to providing continuously advancing products, and in shorter time. An example, presented here, is a new and efficient development method using HLS (high-level synthesis).

In the development of LSI (large scale integration) ICs (integrated circuits) with various functions, HLS technology is widely utilized during development. At Konica Minolta, HLS technology has been utilized in the development of LSI ICs for image processing. Because laser printer engines handle multiple input/output signals, HLS technology is now also utilized in the development of LSI ICs for the laser printer engines found in Konica Minolta's new products.

Currently, HLS technology has been applied successfully to laser printer engine motor control modules having a complicated control algorithm. However, we found that when HLS technology is used in the conventional way, the synthesized RTL (register transfer level) design includes characteristic descriptions not used in hand-coded RTL design. These characteristic descriptions cause an over-concentration of interconnects in the process of layout design. This concentration of interconnects is due to inappropriate settings of the HLS tool so that calculation modules and other modules are excessively shared. By appropriately configuring the settings of the HLS tool (e. g. setting an upper limit to the number of sharable registers), the concentration of interconnects was reduced.

As a result, we have established a new and efficient development method using HLS technology and including measures taken in the development of LSI ICs through the implementation of HLS RTL designs employing hand-coded RTLs.

* 情報機器開発本部 エンジン制御開発センター 第1エンジン制御開発部
** 情報機器開発本部 オプトメカトロサービス開発部

1 はじめに

大規模化や複雑化が進むLSIの開発において、高位合成技術を活用した開発が進んでいる。Fig. 1 に従来の開発フローと高位合成技術を活用した開発フローを示す。これまでVerilog HDLなどのハードウェア記述言語 (HDL: Hardware Description Language) を使ってRTLデザインを手書きし、ハードウェア回路を設計していた。この様な従来のRTL設計では、レジスタ間のデータ転送について、クロックタイミング毎に記述し設計している。一方で、高位合成技術を活用すると、高位合成ツールが、C言語などの高級プログラミング言語で記述されたソースコードからRTLデザインを自動生成し、ハードウェア回路を設計してくれる。これにより、設計者は、クロックタイミングを意識せず、実行命令を主とする少ない記述での設計が可能となる。そして、HDLよりも高い抽象度での設計が可能となり、開発効率が向上できるというメリットがある。また、設計したRTLデザインをFPGA (Field Programmable Gate Array) へ実装して検証した結果に不具合があり、その原因がそもそも制御アルゴリズムにあった場合でも、制御アルゴリズムの見直しや修正がし易いというメリットもある。

しかし、高位合成技術は、画像処理アルゴリズムの様なパイプライン型の演算処理系ハードウェア回路のRTLデザインを生成する事には適しているが、データの入出力を行う様なシステム制御系ハードウェア回路のRTLデザインを生成する事には不向きである、という特徴を持っている。その理由のひとつは、システム制御系ハードウェア回路が逐次処理ではない事にあり、複雑な制御構成であるほど、回路規模は大きくなり易い。そのため、設計者は所望のハードウェア回路が生成される様に、ハードウェア回路を意識した記述へソースコードを変換したり、あるいは、高位合成ツールに様々な合成制約を設定したり、が必要となる。

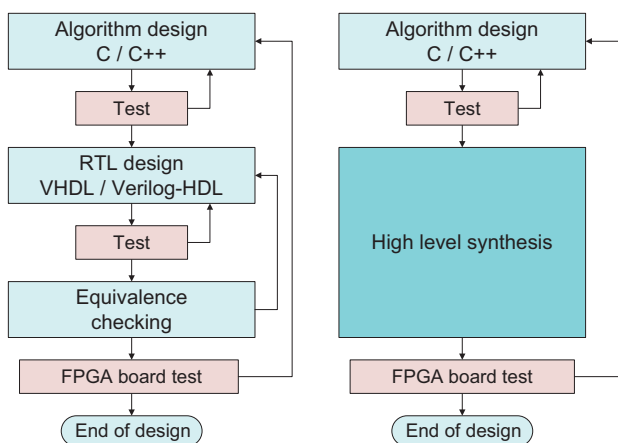


Fig. 1 Conventional HDL (hardware description language) versus HLS (high-level synthesis) technology.

HLS technology allows logic design generally to be performed at a higher level of abstraction than does conventional HDL-based design, with accordingly increased development efficiency.

今回、コニカミノルタでは初めて、画像処理用LSIの開発で実績のあった高位合成技術を、レーザープリンターエンジン用LSIの開発にも活用した。その結果、レーザープリンターエンジン用LSIの開発期間が短縮可能となる基盤を構築する事ができた。本稿では、その開発事例を紹介する。

2 システム概要

2.1 LSIシステム構成

今回開発したLSIのシステム構成をFig. 2 に示す。原稿用紙や印刷用紙を搬送するモーターを制御する機能、レーザー書き込み装置に使用するポリゴンミラースキャナーモーターを制御する機能、各種センサーからのアナログ信号をデジタル値へ変換するA/D変換器 (ADC: Analog to Digital Converter), RAM (Random Access Memory), レジスタ, などを搭載し、画像処理用LSIと協調したメカニカルな制御を担当している。

これまでオフィス向けプリンターのエンジン制御では、ASIC (Application Specific Integrated Circuit) とMCU (Micro Control Unit) が分散して制御を行ってきた。今回の開発では、これら2つのデバイスを統合して搭載機能の連携性を高め、より高機能でより低コストとなるLSIを目指した。この目的を実現するため、MCUを使ってソフトウェアが実現してきた機能を、ハードウェア化する必要がある、その多くの機能はRTLデザインを手書きして設計した。しかし、制御アルゴリズムが複雑なモーター制御に対しては、その一部において高位合成を実施し、制御アルゴリズムの検討からハードウェア回路の設計までを効率的に進める事とした。

なお、今回開発したLSIは、コニカミノルタのオフィス向けプリンターとプロダクションプリンターのどちらにも搭載できる様な、様々な機能拡張を行っているが、本稿では、高位合成技術を活用したモーター制御モジュールに的を絞って紹介していく。

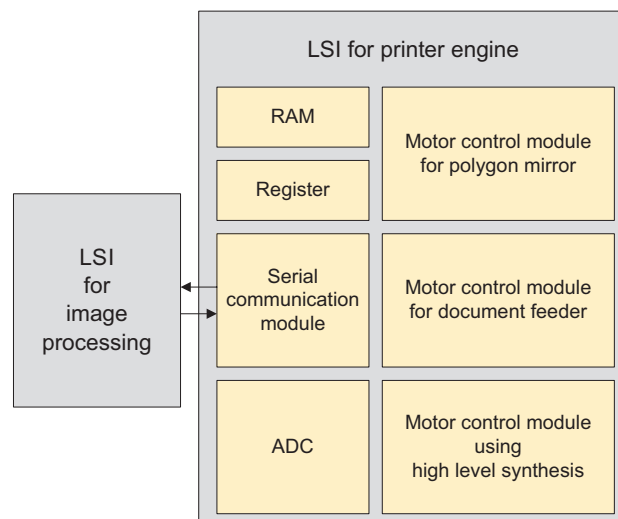


Fig. 2 LSI system configuration for laser printer engine.

2.2 高位合成モジュール構成

今回、高位合成技術を活用したモーター制御モジュールは、3相ブラシレスDCモーター4つを同時に駆動可能な機能である。Fig. 3にそのモジュール構成を示す。2つのA/D変換器、制御部、を備えており、制御部の内部には演算部、および4つのパルス幅変調 (PWM: Pulse Width Modulation) 波形生成部を備えている。

2つのA/D変換器は、各3相ブラシレスDCモーターのU相巻線とV相巻線に流れる電流量を検出しており、電流検出用抵抗器と増幅回路で構成した電流検出回路を介して得られる電圧値を同期したタイミングで取得する事で、あるタイミングにおける各巻線の電流量を検出している。一方、W相巻線に流れた電流量は、検出した2つの電流量から制御部で算出している。

制御部は、各3相ブラシレスDCモーターに流れた電流量と、RAMに記憶するモーター制御パラメーターから現在のモーター回転速度を算出している。そして、現在のモーター回転速度を目標値と一致させるため、パルス幅変調したモーター駆動信号を出力する。

出力したモーター駆動信号がLSI外部のドライブ回路を駆動し、その際にU相巻線とV相巻線に流れる電流量を2つのA/D変換器が再び検出する、というフィードバック制御を行っている。

尚、制御部の内部にある演算部が高位合成で生成したRTLデザインであり、手書きしたRTLデザインに組み込む構成としている。

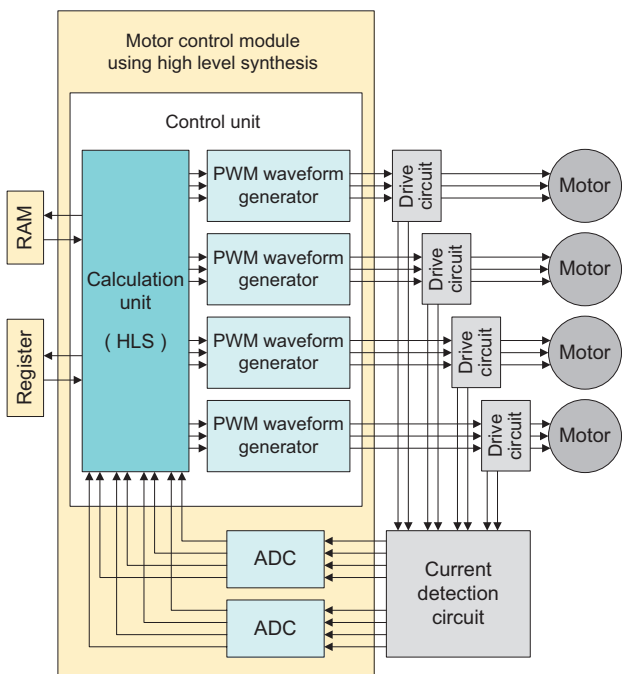


Fig. 3 Motor control module configuration designed via HLS technology.

Fig. 4 にパルス幅変調したモーター駆動信号の生成方法を示す。演算部では、現在のモーター回転速度を目標値と一致させる指令値を生成し、PWM波形生成部に通知している。一方、PWM波形生成部では、一定周期で

発振するキャリア信号を内部で生成しており、演算部から通知された指令値とキャリア信号の比較を行っている。比較した結果、キャリア信号の方が高い期間をHigh、指令値の方が高い期間をLowとするPWM波形を生成し、このPWM波形をモーター駆動信号として出力している。

今回のモーター制御モジュールでは、キャリア信号が谷となるタイミングをトリガーとして、モーターの各巻線に流れる電流量をA/D変換器が検出している。また、A/D変換を完了した後からは、その検出した電流量を基にモーター回転速度を算出して、指令値を生成する制御を演算部が実施している。この生成した指令値は、次のキャリア周期から反映している。

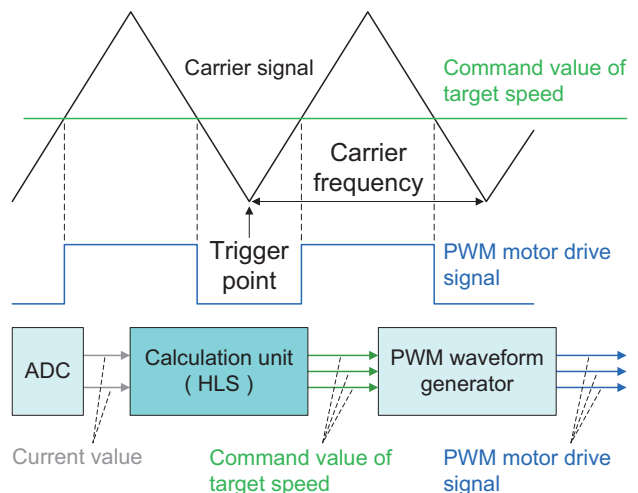


Fig. 4 Pulse-width modulated motor drive signal generated by motor control module.

このモーター制御モジュールの特徴のひとつとして、A/D変換器と演算部を、4つのモーター用に個別では持たず、共有している点が挙げられる。

Fig. 5 は、モーター駆動信号を生成するためのキャリア信号を、4つのモーターで同期させた例であり、今回のモーター制御モジュールでは実施していない方式である。この方式では、各モーターのキャリア信号が谷となるタイミングは重なっているため、A/D変換の開始タイミングや、その後の指令値を生成する演算のタイミングが重なってしまう。このような場合、A/D変換器と演算部を、4つのモーター用に個別で持つ必要がある。そのため、回路規模が増加して、目的のひとつである低コスト化は難しくなると考えた。

そこで、モーター駆動信号を生成するためのキャリア信号を、モーター間でずらして生成する方式を採用した。Fig. 6 は、各キャリア信号を1/4周期ずつずらして生成した例であり、今回のモーター制御モジュールで実施している方式である。この方式では、各モーターのキャリア信号が谷となるタイミングが異なるため、A/D変換の開始タイミングや、その後の指令値を生成する演算のタイミングが重なる事がない。これにより、A/D変換器と演算部を4つのモーターが共有する事を可能にした。

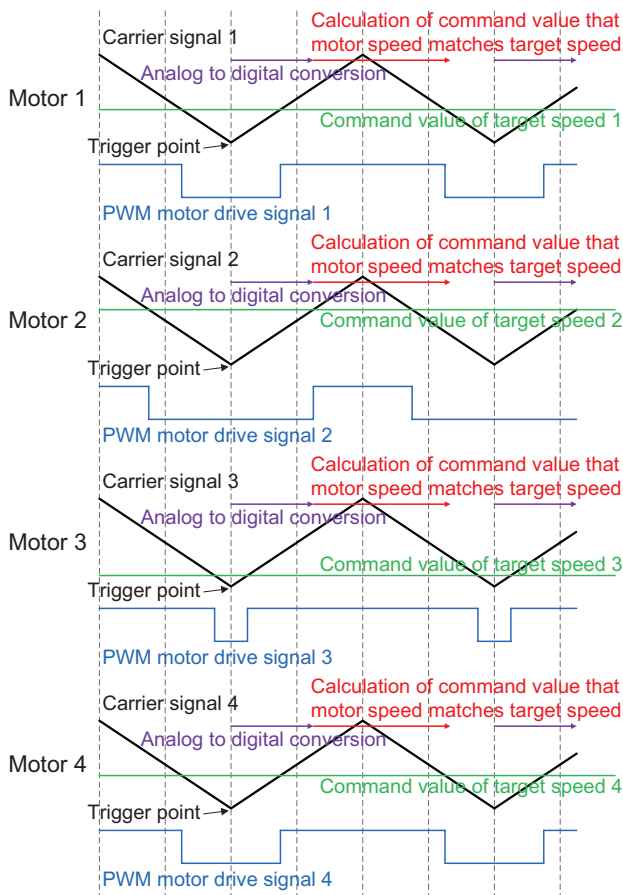


Fig. 5 Conventional generation of motor drive signals utilizing four simultaneous carrier signals.
In this method, four pairs each of an A/D converter and a calculation unit are necessary because A/D conversion and calculation for the generation of command values begin simultaneously.

但し、A/D変換を開始してから、指令値を生成するまでの期間を、キャリア周期の1/4以内に収める必要がある。このため、今回のモーター制御モジュールでは、高い分解能で、変換時間が高速な逐次比較型A/D変換器を搭載している。また、指令値を生成する演算を、A/D変換を完了した後の残り期間で完了させるため、レイテンシーの上限値を高位合成ツールに設定し、RTLデザインを生成している。このレイテンシーとは、演算部にデータを入力してから指令値が出力されるまでの遅延時間であり、このレイテンシーを設定する事で、演算クロックサイクルを適切に調整することができる。この様にして、A/D変換器と演算部の共有を実現している。

3 高位合成したRTLデザインの課題

3.1 高位合成したRTLデザインの特徴

今回のモーター制御モジュールは、逐次処理ではないため、高位合成技術が不向きとされるシステム制御系に高位合成を実施した事になる。また、3相ブラシレスDCモーター4つを同時に駆動可能な機能を、A/D変換器や演算部を共有するなかで実現している。この様な条件で

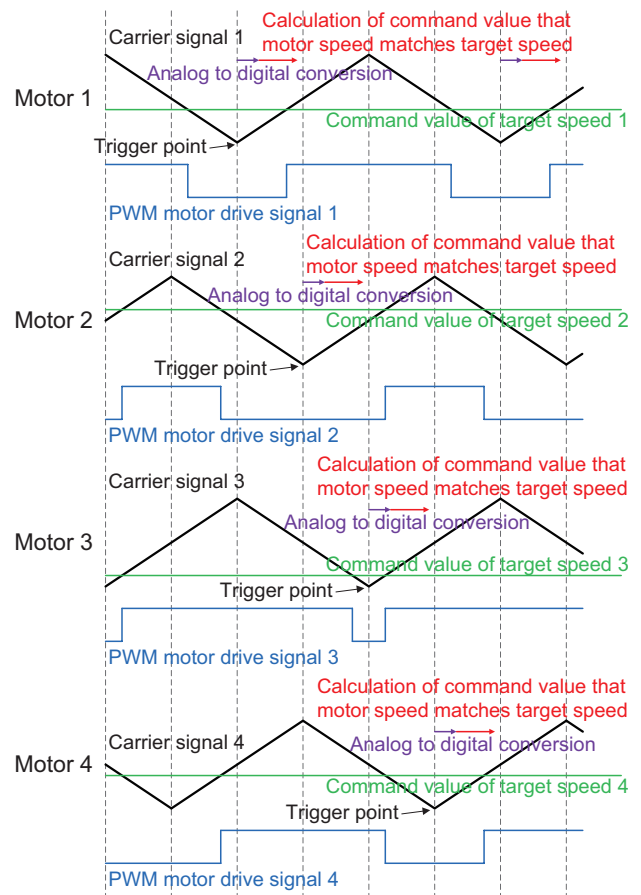


Fig. 6 Achieving smaller circuit size through the generation of motor drive signals by shifting four carrier signals each by 1/4 cycle.
Since the four carrier signals are shifted 1/4 cycle apart, A/D conversions and calculation made to generate command values following those A/D conversions are not performed simultaneously, and the A/D converter and the calculation unit are shared.

高位合成を実施した結果、高位合成ツールが生成したRTLデザインには、手書きしたRTLデザインでは見ることのない特徴的な記述がいくつか確認できた。特に特徴的な点は以下の3つである。

- (1) 入力値によって様々な条件分岐を行いながら、RAMデータやレジスタ値を何度も読み込んで演算処理を行っている。
- (2) RTLデザインの様々な部分からRAMへのアクセスが発生している。
- (3) ソースコードの微修正であっても、高位合成したRTLデザインでは回路規模が大幅に変化してしまう。

これらの特徴を持つ今回の高位合成したRTLデザインは、回路規模が大きく、今回のLSIに搭載した制御モジュールのなかで、ゲート数が一桁違うほど大きなハードウェア回路となった。

また、回路規模が大きいRTLデザインの様々な部分からRAMへのアクセスが発生した事で、論理回路を配置配線するレイアウト設計において、配線混雑を引き起こし、改善を必要とした。

これらの特徴に対する改善は、高位合成技術を活用したモーター制御モジュールのみをFPGAへ実装して行うプロトタイプ検証時から発生しており、高位合成したRTLデザインのゲート数を基に、ロジック規模に余裕のあるFPGAを選定したが、論理回路の配置配線ができず、テクノロジーノードがより微細で、ロジック規模に十分な余裕のあるFPGAを再選定して検証を進める事となった。この事から、システム制御系で高位合成を実施した場合、高位合成したRTLデザインが、ターゲットとするデバイスに対して実装可能であるかを判断するためには、ロジック規模だけではなく要素も考慮する必要があるという知見が得られた。

以下で配線混雑を引き起こした要因を紹介していく。

3.2 配線混雑を引き起こした要因

3.2.1 入力数の多いマルチプレクサー

高位合成したRTLデザインで配線混雑を引き起こした要因のひとつは、A/D変換器や演算部を共有したために、入力信号数の多いマルチプレクサーが生成された事にある。例えばFig. 7の様な、入力信号として16bitのデータが512値あり、これを9bitのセクタ信号で1値を選択し出力するマルチプレクサーである。この様なマルチプレクサーは、演算部のなかで繰り返し使用されており、乗算や除算や飽和加算を実行する演算モジュールの入力経路で生成されている。

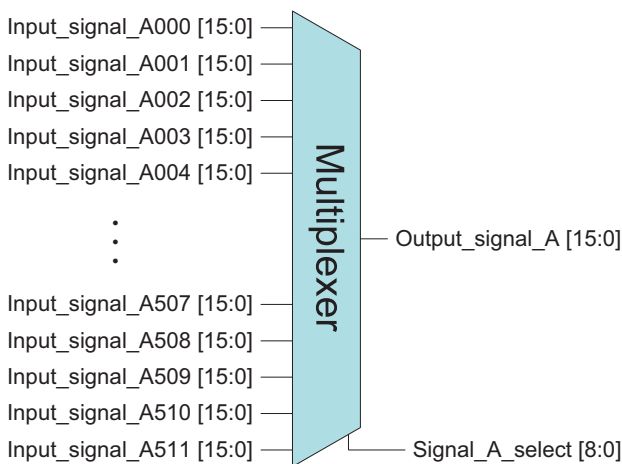


Fig. 7 Structure of multiplexer with multiple input signals.

Such a multiplexer is one factor causing a concentration of interconnects.

3.2.2 セクタ数の多いOne-hotマルチプレクサー

高位合成したRTLデザインで配線混雑を引き起こした要因のもうひとつは、RTLデザインの様々な部分からRAMへのアクセスが発生したために、セクタ信号数の多いOne-hotマルチプレクサーが生成された事にある。例えばFig. 8の様な、入力信号として1bitのデータが319値あり、これを319bitのセクタ信号で1値を選択し出力するマルチプレクサーである。この様なマルチプレクサーは、RAMの入力経路で生成されている。

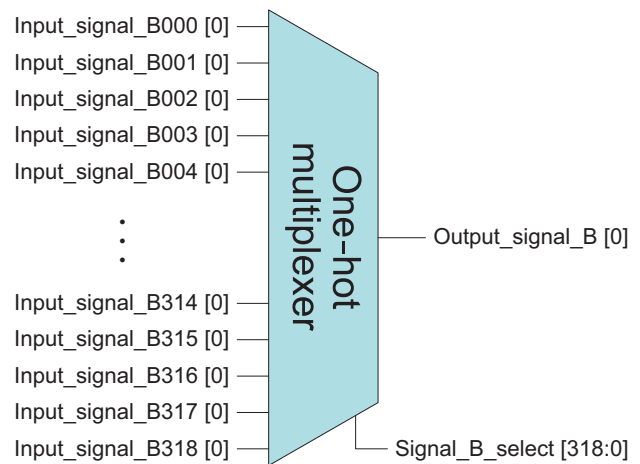


Fig. 8 Structure of one-hot multiplexer with multiple selector signals. Such a one-shot multiplexer is one factor causing a concentration of interconnects.

また、One-hotマルチプレクサーの出力をRAMと接続する部分では、if文の条件式に、多数のOR演算が使われている。例えばFig. 9の様な、One-hotマルチプレクサーの出力信号を、80条件のビット単位OR演算で判定する記述である。

```
always @ ( posedge CLK or negedge nRST ) begin
  if ( ~nRST ) begin
    RAM_Input_A0 <= 1'b0;
  end
  else if ( condition_01 | condition_02 | ...
    ... | condition_79 | condition_80 ) begin
    RAM_Input_A0 <= Output_signal_B [0];
  end
end
```

Fig. 9 An OR operation defined by 80 conditions.

Interconnects are concentrated here because the OR operation is defined by as many as 80 conditions describing under what conditions the output of the one-shot multiplexer is acquired by registers.

3.3 高位合成したRTLデザインへの対応

3.3.1 配線混雑を引き起こした原因と対策

高位合成ツールは、ユーザーが設定した制約条件に基づき、ソースコードをできるだけ最適化し、可能な限りゲート数を削減したRTLデザインの生成を試みる事が知られている。また、高位合成技術を活用したモーター制御モジュールは、A/D変換器と演算部を4つのモーターが共有する構成としていた。さらに、今回は回路規模が想定以上に大きい事から、ゲート数の削減を目的に、高位合成したRTLデザインは、階層構造を持たないフラットなデザインとしていた。

これらにより、演算モジュールやRAMの共有が多いRTLデザインが生成され、演算モジュールやRAMへの配線が集中し、入力信号数の多いマルチプレクサーや、セクタ信号数の多いOne-hotマルチプレクサーを生み出して、レイアウト設計時に配線混雑が発生した、と推定している。

これに対応するため、共有できるレジスタ数の上限値を高位合成ツールに設定し、演算モジュールが過度に共有される事を制限して、配線混雑の改善を行っている。また、今回は実施していないが、RAMを分割する事で配線混雑が改善できると考えられる。

但し、共有できるレジスタ数の上限値を設定する事や、RAMを分割する事は、回路規模を増加させる要因となるため、ゲート数と配線混雑度のバランスのとれた適切な制限を高位合成ツールに設定する事が必要となる。今回のLSI開発では、高位合成ツールの設定をいくつか試し、適切な制限の検討を行っている。このような検討が容易に実施できる事も、高位合成技術を活用した場合のメリットと考えられる。

3.3.2 RTLデザイン間の通信インターフェイス

C言語などのソースコードは、クロックタイミングを考慮しない記述となっている。そのため、手書きしたRTLデザインとは異なり、高位合成したRTLデザインは、通信プロトコルの把握が難しくなっている。

今回のモーター制御モジュールでも、高位合成したRTLデザインと手書きしたRTLデザインとの間で、データの受け渡しがあり、データ送受信時の信頼性を確保する必要があった。

これに対応するため、高位合成したRTLデザインとのインターフェイスには、Fig. 10に示す様な、ハンドシェイク通信モジュールを設けている。ハンドシェイク通信モジュールは、接続要求 (Synchronize) とそれに対する応答 (Acknowledge) によって相手の受信状態を確保し、データ送受信時の信頼性を確保する機能である。これを設けた事で、ソースコードの修正によって、高位合成したRTLデザインに大きな変化がある場合でも、通信プロトコルの変化の有無を気にする事なく、高位合成したRTLデザインを、手書きしたRTLデザインに組み込む事ができる。

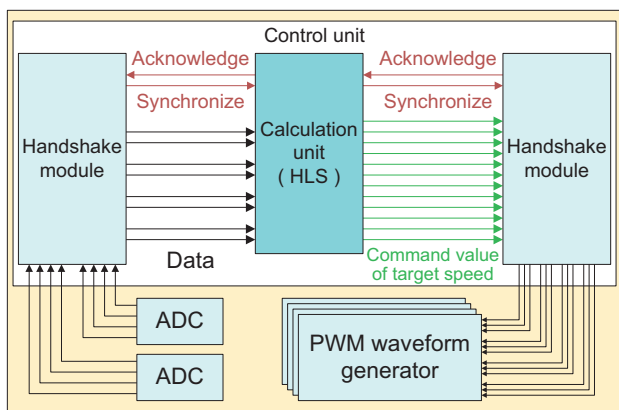


Fig. 10 Communication interface with HLS generated RTL design.

A handshake communication module is used to achieve a secure communication protocol with the hand-coded RTL. A synchronized signal and an acknowledge signal are used to check the reception status of the communication partner so that reliable transmission and reception are secured.

4 まとめ

今回、コニカミノルタでは初めて、高位合成技術を活用してレーザープリンターエンジン用LSIの開発を行い、新製品への搭載を実現した。

但し、高位合成したRTLデザインには、手書きしたRTLデザインでは見る事のない特徴的な記述があり、その特徴が要因となって、レイアウト設計時に配線混雑を引き起こしていた。この原因は、高位合成ツールが演算モジュールやRAMを可能な限り共有する中で、それらを過度に共有した事にあった。そこで、共有できるレジスタ数の上限値を設定するなど、高位合成ツールに適切な制限を設定する事で、配線混雑を改善している。

しかし、高位合成ツールの設定による改善にも限界がある。制御アルゴリズムを設計する段階で、論理回路を考慮して、演算モジュールをどのように使い、RAMやレジスタをどのように配置するのか、といったハードウェアアーキテクチャの検討をより一層実施する事で、さらに効率的な開発が可能になると考えている。