

ハードウェア演算による高速信号処理 —FPGA を用いたシステム開発—

キーワード：FPGA、ハードウェア演算、高速信号処理、デジタル回路

はじめに

近年、携帯電話等の通信機器の発達やゲーム機、ロボット、医療機器のための画像処理等への要求から、デジタル信号処理の必要性がますます高まっています。多くの場合、CPU(Central Processing Unit)を用いたソフトウェアによる演算がデジタル信号処理を担ってきました。しかし近年、安価でかつ現場開発が可能なハードウェアである FPGA

(Field Programmable Gate Array) と呼ばれるデジタル信号処理用集積回路が登場し、ハードウェアによるデジタル信号処理が取り扱易くなってきました。ここでは、ソフトウェアとハードウェアそれぞれの処理についての比較と FPGA を用いる利点を述べ、FPGA の構造や設計について簡単に解説します。

ソフトとハードの信号処理の違い

CPU は一般的に fetch(命令の取り出し)、decode(命令の解釈)、execute(命令の実行)、writeback(実行結果の書き戻し)の動作を繰り返しています。CPU に与える命令を変えることで、多様で極めて複雑な処理を行わせることができます。このサイクルは高速なクロックで動作しますが、複数の処理を同時に行うことはできません。また、大容量のメモリを使用可能ですが、アドレスバスとデータバスのバス数が限定されており、同時に多数のメモリにアクセスすることはできません。そこで処理の高速化のため、バス速度がどんどん高速になっており、基板設計が非常に難しくなっています。

一方、ハードウェアによる処理はそれぞれの目的に応じた専用回路で行われ、CPU で行われる命令の取り出しと解釈の必要がありません。また、必要としている演算に対して集積回路のリソースが許すだけ多数の演算部と

メモリを搭載し、データへの複数同時アクセスと並列演算が可能です。特定の処理しかできませんが、同一クロックであれば、CPU による処理よりはるかに高速に処理が可能です。現在では、512 次の FFT 等も容易にハードウェア化が可能です。

ハードウェア処理の種類

ハードウェア処理を行う回路で代表的なものが ASIC(Application Specific Integrated Circuit)です。ASIC は特定用途向け集積回路で専用の回路を作り込むため、高速、小型、低消費電力化が可能です。また、デジタルとアナログ両方の回路を搭載することも可能です。大量に生産した場合、製造単価は低くなります。その反面、半導体メーカーに製造を委託する必要があり、開発に時間と費用が必要で、再設計の損失も大きくなります。

一方、FPGA はプリント基板に実装後も内部回路を変更できるデジタル専用集積回路です。回路動作だけでなく外部入出力ピンの役割も変更できます。自分で設計した専用デジタル回路をその場で IC にすることが可能で、手軽にハードウェア信号処理を実現できます。しかしながら回路に冗長部を多く含み、実行速度、消費電力および単価では ASIC と比較して不利になります。

以上の点を考慮し、開発時には FPGA を用い、大量生産時には FPGA で開発した演算回路を ASIC で実現することが一般的になっています。

FPGA の内部構造

図 1 に FPGA の内部構成を示します。FPGA の内部は論理を変更可能な論理ユニット (以下 LU)、内蔵メモリ、入出力部、乗算器およびこれらを繋ぐ配線などからなります。LU とは小規模な論理を構成する最小単位で、

論理ゲートではなく SRAM を用いたテーブルで入出力間の論理を表します。内蔵メモリは、FPGA の規模によりますが、数十 Mbit まで利用可能です。このメモリはメモリ幅とメモリ深さを変更可能で、例えば 8bit 幅で 512 ワードや 32bit 幅で 128 ワード等に指定して利用可能です。入出力部は FPGA を基板に実装した後でも入出力や規格の変更が可能で、種々の入出力規格(PCI、LVDS 等)に対応しています。乗算器は LU で構成することも可能ですが、最近の FPGA では専用回路として用意されており、LU で構成するよりも高速に動作します。配線は LU、メモリ及び入出力を繋ぐためのもので、多数の断片的な配線とスイッチからなります。クロックスキュー(クロックが回路の各部に到達する時間がずれること)をなくすための専用のクロック用配線もあります。多くの FPGA では外部から与えるクロックを通倍(n 倍すること)する PLL 機能も搭載しています。

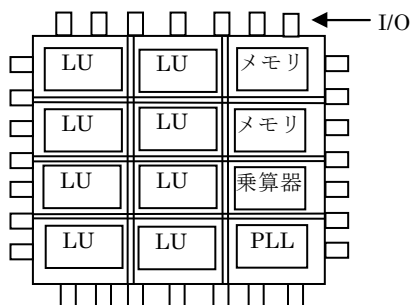


図1 FPGA の内部構成イメージ

FPGA の設計の流れ

設計の流れを図2に示します。

1. 論理設計

ハードウェア記述言語(HDL)を用いて設計を行います。四則演算はそのまま式で記述可能です。データがクロックとともに変化するので、クロック毎の動作を記述します。

2. 論理合成

HDL で記述された論理を回路に変換します。(実際は設計用 CAD ソフトが変換)

3. 配置配線

合成された回路を実際の IC のどの部分に配置し、どのように配線するかを決めます。

4. シミュレーション

設計 CAD ソフト上で入力データパターンを与えて動作のシミュレーションを行い、得られる出力結果や動作速度が所望のものか検査します。所望の結果が得られるまで、上記の手順を繰り返します。

5. コンフィグレーション

FPGA に設計データを書き込み、動作可能にします。一般に JTAG と呼ばれる規格のピンを通して書き込みます。FPGA は電源 OFF で回路情報が失われてしまうため、外付けの ROM に回路情報を書き込み、電源 ON 時に ROM から読み込むようにすることが一般的です。

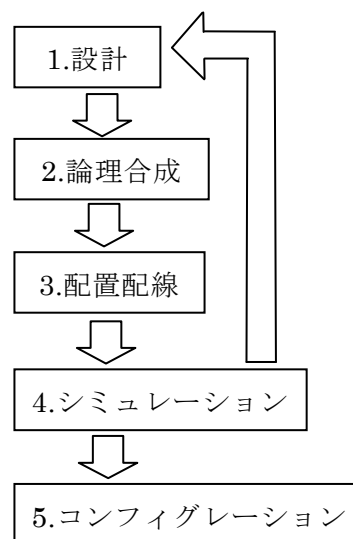


図2 FPGA の設計の流れ

おわりに

数万円以下で FPGA ボード付トレーニングブックが市販されており、その開発ソフトウェアは FPGA メーカーから無償でダウンロードできます。現在では、FPGA の開発・トレーニング環境は整っています。産技研でも、超音波画像処理システムの研究開発等に应用しています。詳細はテクニカルシート(No.09013)をご参照下さい。また、FPGA を用いた信号処理に関して相談・指導等の支援も行っています。お気軽にご相談下さい。