第4章 Cプログラムでアルゴリズムを検証して HDL化により100倍高速化！

JPEG画像の処理アルゴリズムとJPEGデコーダの設計と実装

石原 ひでみ Hidemi Ishihara

ここでは，JPEG画像フォーマットについて解説した後，JPEGデコーダ処理のCプログラムを作成し，それをCPUで実行して動作を確認します．そして処理に時間がかかる部分をハードウェア化して，どれだけ高速化されたかを確認します．JPEGデコーダ・コアは，AXIバスに接続するアクセラレータとして動作します．

1．JPEGデコーダを開発する

今回はFPGAでJPEGファイルをデコードし，画像をRGB出力するJPEGデコーダを開発します．CPU＋FPGAの醍醐味であるソフトウェア・アルゴリズムをハードウェアにオフロードする開発プロセスを解説します．さらに，開発したJPEGデコーダはZynqのPL部（FPGA部分）に実装し，PS部（ARMプロセッサ部分）から制御を行います．

Zynqへの実装と実機検証には，Digilent社から販売されているZynq評価ボードのZYBO（**写真1**）を使用します．

● ソフトウェア処理のハードウェア化

まず一般的に，ソフトウェア処理をハードウェアにオフロードする場合，ハードウェアの原理モデルになるソフトウェア原理モデルを作成します．ソフトウェア原理モデルはあくまでソフトウェアなので，そのままハードウェアへ変換できることは多くありません．

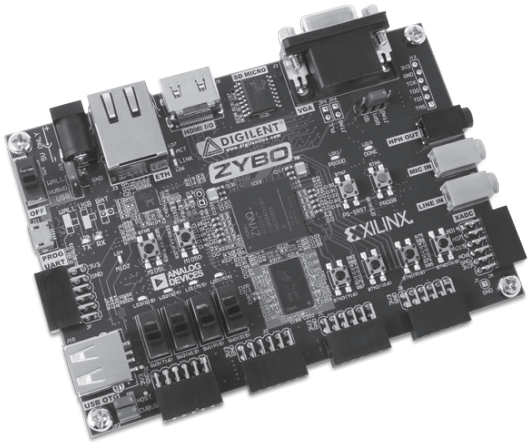

写真1　Zynq搭載評価ボードZYBO（Digilent社）

そして，ソフトウェア原理モデルからどのようにハードウェアにオフロードするかを検討します．実現が難しそうであればソフトウェア原理モデルの修正を繰り返して，実際にハードウェアへの落とし込みが可能なレベルのソフトウェア原理モデルを作り上げていきます．

ハードウェアに落とし込みができそうなソフトウェア原理モデルが完成すると，ようやくハードウェアへの実装のフェーズに入ります．今回はソフトウェア原理モデルをC言語，ハードウェアをVerilog HDLで実装し，両者の言語体系に沿ったハードウェア・オフロードを解説していきます．

今回は言語トランスレーションでのハードウェア・オフロードを実現するので，まず，ソフトウェアとハードウェアの言語体型が似ているものを選びました．C言語とVerilog HDLはよく似た言語と言われているので，C言語でソフトウェア原理モデルを作成し，そのままの気分でVerilog HDLへの設計に入れることで，できる限り言語トランスレーションでの負荷・変換ミスを少なくすることを目的としています．

ソフトウェア・アルゴリズムをハードウェア・オフロードする際に，C言語やVerilog HDLでなければいけない理由はありません．PerlやRuby，PHP，JavaやC++でもかまいません．ハードウェア言語もVHDLを使ってもかまいません．

● 対応JPEGデータの仕様

今回開発するJPEGデコーダがデコードできるJPEGデータを次のように決めました．

- ベースラインのみ
- サンプリングは4：2：0
- サムネイルはなし
- リセットマーカは処理しない

この仕様はJPEGファイルの中でも，ベーシックなものになります．