

仮想周辺機器を備えたウェブブラウザベースハードウェア 実機検証環境における高位合成フロントエンドの開発

吉塚友美¹ 山脇彰²

概要：組み込みシステムが搭載された製品開発の試作では FPGA が使用されているが、FPGA を使用した開発は様々な問題を抱えており、開発者に深い知識と技術、および経験が求められている。これらの問題を解決するために、複数の開発者がネットワーク経由で FPGA ボードを共有でき、実際の FPGA ボードや各種周辺機器に依存しないブラウザベースのハードウェア実機検証環境を開発している。本論文では、高位合成フロントエンド機能の導入を試みる。

キーワード：FPGA, 仮想周辺機器, リモート, ブラウザ, 開発環境, 高位合成

Development of high-level synthesis front-end for web browser-based hardware verification system with virtual peripheral

TOMOMI YOSHIZUKA^{†1} AKIRA YAMAWAKI^{†2}

Abstract: FPGAs are used in prototyping for product development with embedded systems. However, development using FPGAs has various problems and requires developers to have in-depth knowledge, skills, and experience. To solve these problems, we are developing a browser-based hardware verification environment that allows multiple developers to share an FPGA board via a network and is independent of the actual FPGA board and various peripheral devices. In this paper, we attempt to introduce a high-level synthesis front-end function.

Keywords: FPGA, virtual peripherals, remote, browser, development environment, rapid prototyping, high-level synthesis

1. はじめに

スマートフォンや自動車などの組み込みシステムが搭載された製品開発の試作では、回路を何度も書き換えられる FPGA が使用されている[1][2][3][4]。FPGA を使用することで試作の度に新規にマイコンを開発しなくてよいため、設計期間・開発コストが抑えられる。しかしながら、FPGA を用いた試作では次のようなハードウェア開発の問題点が挙げられる。開発者が必要なハードウェア処理に加え、その動作を実機で検証するために周辺機器のインタフェースもハードウェアとして新規に開発する必要があり、製品ごとに異なる周辺機器を用意する必要がある。また、ハードウェア開発にはハードウェア記述言語を用いてクロックレベルでデジタル回路の動作を詳細に記述する必要がある。さらに、ハードウェア開発をするにあたって、様々な開発ソフトやそれらを利用する知識が必要である。以上より、開発者には深い知識と技術、および経験が求められ、負担がとても大きいことがわかる。近年では、組み込みエンジニアが人手不足であるため、これらの問題を解決し FPGA を用いた開発のさらなる負荷削減は非常に重要な課題である。

そこで、我々は複数の開発者がネットワークを介して FPGA ボードを共有でき、実際の FPGA ボードや各種周辺機器に依存しない仮想周辺機器を備えたブラウザをユーザインタフェースとしたハードウェア実機検証環境を開発している。具体的には、CPU 内蔵 FPGA を搭載した FPGA ボー

ドに Linux を導入し、Web サーバーを構築した。そして、遠隔ハードウェア検証の基本機能の実装を行い、データアップロード機能と回路の切替機能を導入した。本稿では、ソフトウェアプログラムから HDL プログラムへの変換を容易かつ効率的に行うために高位合成フロントエンド機能の導入を試みる。

2. 提案の実機検証環境

提案の実機検証環境の概要図を図 1 に示す。開発後の製品に搭載される実周辺機器は各種市販の USB 機器に置き換えることで仮想化する。また、インタフェースはパソコンに置き換えられるため、開発者はそれらに必要なインタフェースやファームウェアの開発は必要なくなる。そして、CPU 内蔵 FPGA とパソコンはネットワークを使って協調動作し、実際の製品と近い環境で試作が可能である。これにより、開発者に必要なものは、CPU 内蔵 FPGA ボードと USB 機器を備えたパソコン、任意のソフトウェア環境、インストールされた高位合成ツールと FPGA 実装ツールになる。

図 1 の左側は提案のユーザインタフェースである。作業ツールをウェブブラウザに統一することで、開発者がハードウェア開発に必要な各種ツールの複雑な使用方法を意識しなくてよくなる。さらに、各種 OS に合わせてスクリプトや Docker イメージを用意し、OS に依存しないクロスプラットフォームな開発環境を実現する。

^{†1}TOMOMI YOSHIZUKA, Kyushu Institute of Technology

^{†2}AKIRA YAMAWAKI, Kyushu Institute of Technology

本論文では画像処理を例に挙げてユーザインタフェースの概要を説明する。開発者は自分の慣れ親しんだ環境で画像処理を行うソフトウェアプログラムを開発する。そして、高位合成の対象として入力画像と共にブラウザで指定する。ブラウザからの指示を受け、高位合成ツールはソフトウェアプログラムをHDLプログラムに変換する。さらに、FPGA実装ツールがそれを回路データに変換する。実行ボタンを押すとCGIプログラムによりFPGA上で実装されたハードウェアを起動し、回路データと画像データがネットワーク越しにFPGAに送られる。そして、FPGAにハードウェアが構成され、画像データが処理される。CGIはハードウェアの

処理結果（画像処理の結果や実行時間等）をHTMLファイルで返し、ブラウザに表示する。

3. 高位合成フロントエンドの開発

高位合成フロントエンドの構成を図2に示す。基盤環境として、CPU搭載FPGAのCPU上にLinuxのUbuntu18.04とウェブサーバー(Apache HTTP Server 2)、高位合成ツール(Xilinx社 Vivado HLS 2018.3)を導入した。LinuxイメージはSDカードにかかれ、FPGAに挿入される。そして、FPGAインタフェースを起動する。この環境下に、HTMLで作成したユーザインタフェース、PHPで作成したデータアップロード機能、CGIによるHTMLとFPGAファームウェアとの

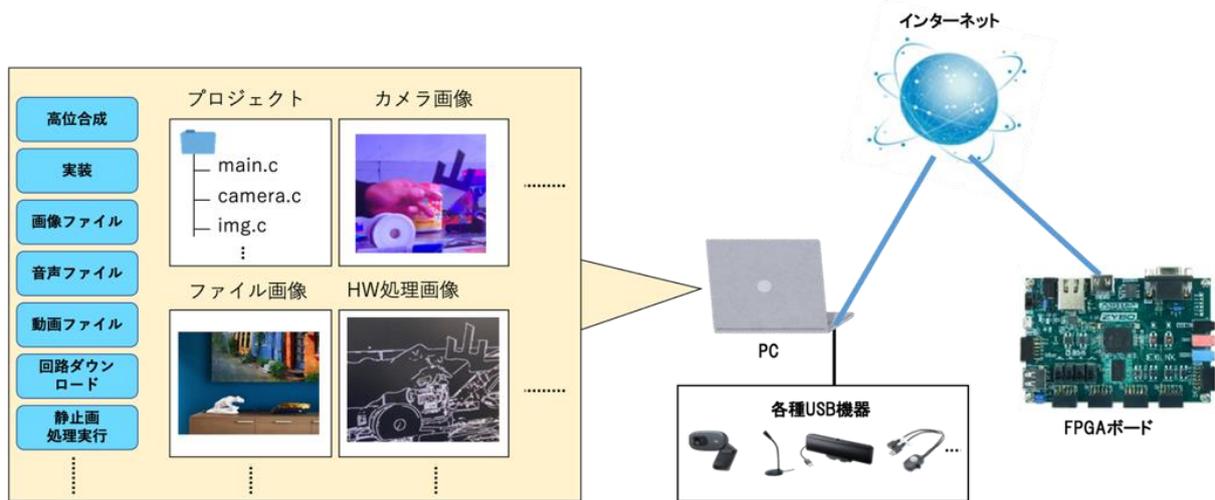


図1 提案の実機検証環境

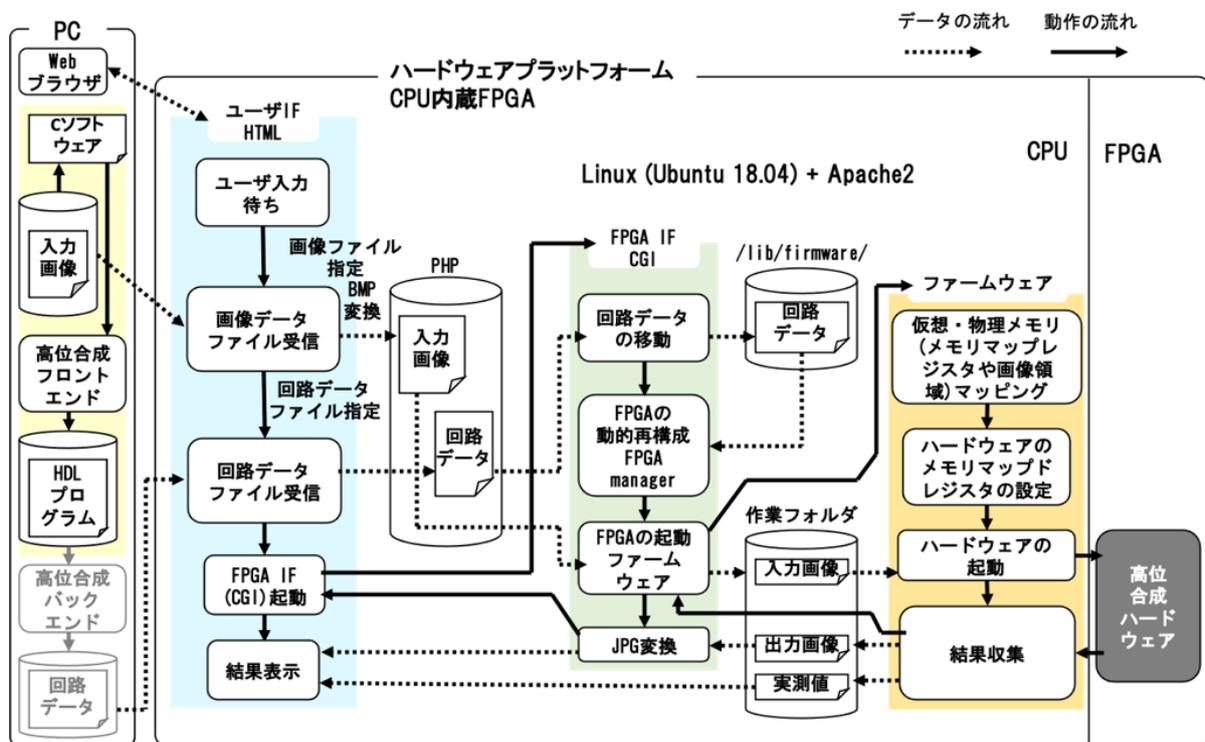


図2 高位合成フロントエンドの構成

```

1 my $function_name = $ARGV[0];
2 my $file_name     = $ARGV[1];
3
4 open(DATAFILE,">> Tclスクリプト") or die("error :$!");
5
6 print DATAFILE "書き込みたいデータ";
7
8 my $cmd = 'vivado_hls -f Tclスクリプト';
9
10 system $cmd;
    
```

図 3 Perl プログラム

```

1 #####
2 ## This file is generated automatically by Vivado HLS.
3 ## Please DO NOT edit it.
4 ## Copyright (C) 1986-2018 Xilinx, Inc. All Rights Reserved.
5 #####
6
7 open_project -reset プロジェクト名
8 set_top ハードウェア化対象関数名
9 add_files 対象Cソースファイル名
10
11 open_solution -reset "ソリューション名"
12 set_part { デバイス名 } -tool vivado
13 create_clock -period 10 -name default
14 csynth_design
15
16 exit
    
```

図 4 Tcl スクリプト

インタフェース、C 言語による FPGA ファームウェアが存在する。

HTML で作成したユーザインタフェースでは、Web ブラウザから受け取った画像データと回路データを PHP 経由でローカルフォルダに保存する。そして、CGI からの指示で FPGA インタフェースを起動する。

高位合成フロントエンドでは、高位合成を用いてユーザが開発したソフトウェアプログラムを HDL プログラムに変

換する。ユーザは、開発したソフトウェアプログラムのファイル名とハードウェア化対象の関数名の入力のみで高位合成フロントエンドを利用できる。

高位合成フロントエンドは、Perl で高位合成ツールを動かすための Tcl スクリプトファイルを作成する。図 3 に Perl ファイルの概要を示す。1・2 行目はコマンドライン引数を利用するための宣言をしている。4 行目では、書き込む Tcl スクリプトファイルを書き込み専用で開く。6 行目で DATAFILE (Tcl ファイル) にデータを書き込む。8 行目で実行したいコマンドを引数に入力し、10 行目で実行している。これにより、Vivado HLS のバッチモードでの Tcl スクリプトの実行が完了する。次に Perl に書き込まれた Tcl スクリプトファイルを図 4 に示す。Tcl は Vivado ツール環境に統合されているスクリプト言語である。7 行目は既存のプロジェクトを開く記述であり、指定したプロジェクトがない場合は、新規に作成される。前情報を残さないために、-reset オプションをつける。8 行目では、高位合成する対象の関数を指定している。9 行目は、ハードウェア化対象の C ソースファイルをプロジェクトに追加している。11 行目ではプロジェクト内でソリューションを開き、12 行目ではプロジェクトで使用するパーツを指定している。13 行目では、ソリューションの仮想クロックを作成しており、クロック周期を 10ns に指定している。14 行目で、プロジェクトを合成している。

4. 高位合成フロントエンドの有用性の評価

開発した高位合成フロントエンドの有用性を評価するために、従来の高位合成ツールの GUI を用いたハードウェア化と高位合成フロントエンドでのハードウェア化の変換時

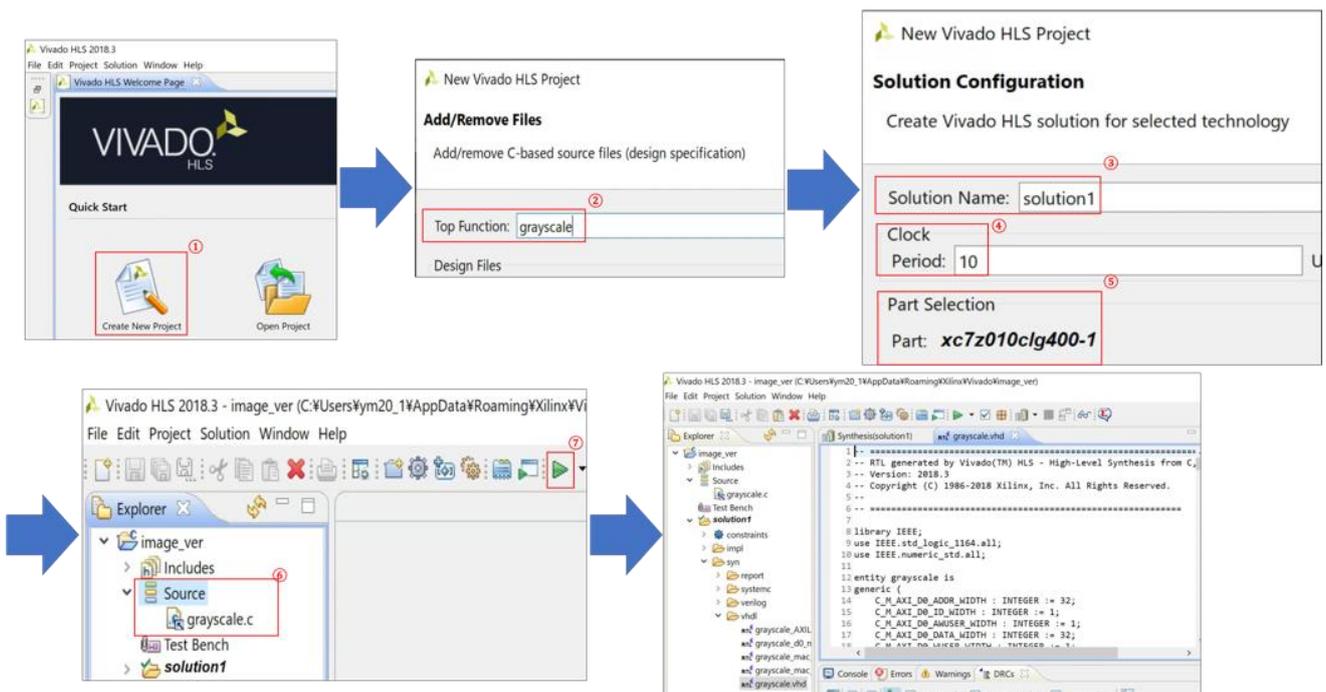


図 5 従来のハードウェア化の流れ

```

C:\Xilinx_work\tutorial_yama\posterization\Pipelined>perl write.pl grayscale grayscale.c
***** Vivado(TM) HLS - High-Level Synthesis from C, C++ and SystemC v2018.3 (64-bit)
**** SW Build 2405991 on Thu Dec 6 23:38:27 MST 2018
**** IP Build 2404404 on Fri Dec 7 01:43:56 MST 2018
** Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.
    
```

図 6 コマンドラインでの実行

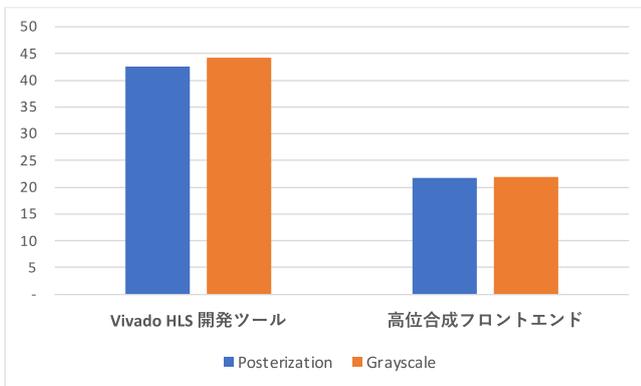


図 7 ソフトウェアからハードウェアへの変換時間

間を比較した。実験用に画像のポストリゼーションとグレースケール化のプログラムを C 言語で記述した。必要なツールのパスは設定済みとする。まず、それぞれの変換作業の流れについて説明する。

高位合成ツール GUI を用いたハードウェア化の流れを図 5 に示す。グレースケール化のプログラムの高位合成を例に挙げて説明する。高位合成ツールは Xilinx 社 Vivado HLS 2018.3 を使用する。ツールを立ち上げたら、新規プロジェクトを作成する (①)。次に高位合成する対象の関数名を指定し (②)、ソリューション名を指定する (③)。そして、ソリューションの仮想クロックの設定を行い (④)、プロジェクトで使用するパーツを指定する (⑤)。プロジェクトができたら、高位合成対象の C ソースファイルを追加し (⑥)、高位合成を行う (⑦)。高位合成が終了すると図のような HDL ファイルが作成される。

次に、我々が開発した高位合成フロントエンドを用いたハードウェア化の流れを図 6 に示す。高位合成対象の関数名とハードウェア化対象の C ソースを引数で指定し、Perl ファイルを実行している。Perl ファイルの中身には Tcl スクリプトを実行するコマンドが記載されているため、Tcl スクリプト作成完了後に HDL への変換が行われる。

図 7 に変換時間の測定結果を示す。ポストリゼーション・グレースケール化共に従来法より高位合成フロントエンドの方が、変換が高速に実施できることを確認した。

5. おわりに

仮想周辺機器を備えたウェブブラウザベースハードウェア実機検証環境においてソフトウェアプログラムから HDL プログラムへの変換を容易かつ効率的に行うための高位合

成フロントエンドを開発した。開発には Perl を用いて高位合成ツールを動かすための Tcl スクリプトを開発し、それを実現した。ソフトウェアプログラムから HDL プログラムへの変換を容易かつ効率的に行うことが目的である高位合成フロントエンドの効果を確かめるために、C プログラムから HDL プログラムに変換するまでの時間を測定した。その結果、変換速度は従来の変換方法より高位合成フロントエンドの方が早くなり、その効果を明らかにできた。今後は、HDL プログラムを回路データに変換する高位合成バックエンドの開発を行い、提案システムのさらなる実現をめざす。

参考文献

- [1]秋山雅裕 他：「FPGA を用いたコンパクトで高性能なパルスパワー発生装置の開発」, J. Plasma Fusion Res, Vol. 85, No. 9 pp. 631-635 (2009)
- [2]照屋大地 他：「PyJer：高位合成ツールと SoC を用いた IoT 向けデバイスプロトタイピングのためのフレームワーク」, 電子情報通信学会論文誌 D, Vol. J100-D, No. 3 pp. 287-297 (2017)
- [3]E. Tang and E. Giacomini : "OpenFPGA : An Opensource Framework Ending Rapid Prototyping of Customizable FPGAs", 29th International Conference on Field Programmable Logic and Applications (FPL), pp. 367-374 (2019)
- [4]S. J. H. Pirzada and A. Murtaza : "A Reconfigurable Model-Based Design for Rapid Prototyping on FPGA", International Journal of Computer Theory and Engineering, Vol. 12, No. 3 pp. 80-84 (2020)
- [5]Tomomi Y, Akira Y, "Building Primitive Functionality of Web browser-based FPGA Hardware Verification System", Proc. of 27th Int'l Symp. on Artificial Life and Robotics 2022 (AROB 2022), pp. 706-709 (2022)
- [6]Atsushi S, Akira Y, "Development of simple verification environment using FPGA for image processing hardware created by High-Level-Synthesis using TCP/IP", Proc. of the 8th IIAE International Conference on Intelligent Systems and Image Processing, pp. 221-226, 2021
- [7]Wicaksana A, Charf A, Andriamisaina C, Ventroux N, "Hybrid prototyping methodology for rapid system validation in HW/SW co-design", The Conference on Design and Architectures for Signal and Image Processing 2019 (DASIP 2019), Montreal, Canada. cea-02494007 (2019)