

FPGA 設計用統合環境を用いた ASIC 設計事例

小林和淑† 神原弘之†† 小野寺秀俊† 田丸啓吉†

†京都大学大学院情報学研究科 †京都高度技術研究所
 京都市左京区吉田本町 京都市下京区中堂寺南町 17 番地
 †TEL: 075-753-5313
 †E-mail: kobayasi@tamaru.kuee.kyoto-u.ac.jp

あらまし 本稿では、MAX+PLUSII (MAX) で AHDL を用いて設計した ASIC と、Verilog-HDL, VHDL を用いて Design Compiler で設計した ASIC との性能の比較を行う。MAX を用いて ASIC の上流設計ができれば、FPGA 上に実現された回路をそのまま ASIC にすることができ、ツールの使用法も簡便で、設計者への初期負担が非常に小さい。学生や社会人の LSI 導入教育へも利用できる。設計事例として、「BCD 電卓」、「教育用マイクロプロセッサ Kuechip2」を例にあげ、AHDL で設計した回路を ASIC 向けにリターゲットしても、その性能、回路面積が Verilog-HDL や、VHDL で設計した回路と遜色ないことを示す。

キーワード FPGA, Altera, MAX+PLUSII, ASIC, 論理合成, 統合環境

An ASIC Design Methodology Using an Integrated Design Tool for FPGAs

K, Kobayashi† H, Kanbara††, H, Onodera† K, Tamaru†
 †Graduate School of Informatics, Kyoto University
 Yoshida honmachi, Sakyo-ku, Kyoto, 606-8501, Japan
 †Advanced Software Technology & Mechatronics
 Research Institute of KYOTO

Chudo-ji Minami-cho 17, Simogyo-ku, Kyoto, 600-8813, Japan
 †TEL: +81-75-753-5313, E-mail: kobayasi@tamaru.kuee.kyoto-u.ac.jp

Abstract Here, we compare ASIC design methodologies between MAX+PLUSII for FPGAs and conventional ASIC design environment by Verilog-HDL or VHDL. MAX+PLUSII is an FPGA design framework that integrates a schematic entry, an HDL Compiler and a simulator. If MAX can be applied to ASIC designs, designers can easily master the design flow owing to its user-friendly GUI environment. We apply the proposed MAX+PLUSII-based ASIC design methodology to two designs, "a BCD calculator", and "A Microprocessor for Education of Computer Hardware and LSI Design: Kuechip2". The proposed method achieves good performance and small area.

key words FPGA, Altera, MAX+PLUSII, Integrated Design Tool, ASIC

1 はじめに

近年のVLSIの発展は目覚ましく、数年前まで夢のごとく語られていたHDLによる設計が現在では、ASIC設計の主流となっている。HDLを用いたLSI設計は、「HDLそのものの記述、HDLレベルでのシミュレーション」、「論理合成(テクノロジーマッピング)」、「ゲートレベルシミュレーション」、「自動配置配線」、「ポストレイアウトシミュレーション」の各段階に大きく分類することができる。VLSI用CADでは、これらのツールを個々に動かさなければならず、ライブラリの管理や、環境設定等の設計そのものに関係のないところでの負荷が大きい。学生の設計教育に使うにはツールそのものの教育に手間を取られて、設計手法を教えているのか、ツールの使い方を教えているのかわからないような状況になってしまう。

他方では、簡単に所望の論理を実現でき、その構成を変更できるFPGAの利用範囲は年々広がっている。FPGAを積極的に教育に利用しようという動きも活発である[1]。一般にFPGAの開発ツールはそのベンダ自体から提供されている。Altera社の提供するMAX+PLUSII(以下MAX)は開発用途がFPGAに絞られるものの、HDLからの論理合成、回路図編集、シミュレーション、FPGAへの配置配線、ポストレイアウトシミュレーションまでが提供される統合環境となっている。教本を1冊用意するか、講師が半日講義すれば、だれでもその使い方が比較的簡単にマスターできる。ここでは、このMAXを用いてHDLからのFPGA設計を行い、そのままその回路をASICにマッピングするという設計手法の性能について論ずる。FPGAの設計者には1. ASIC設計ツールを用いてFPGA設計を行うASIC設計経験者、2. FPGA専用ツールを用いて設計する設計者に大きく二分されるが、本手法を取ることで、2.の設計者が簡単にASICを設計できるのみならず、FPGAからASICへのリターゲットが容易になる。

2 FPGA統合環境を用いたASIC設計

図1に、本稿で提案する設計手順を示す。FPGAを設計する場合は、MAXのみを用いて、AHDL(Altera HDL)での記述、FPGAへのマッピング、シミュレーション、FPGAへの実装といった一連の流れを行う。これらはすべてGUI上で行うことができ、特別な設定等もなく非常に簡便に行うことができる。MAXは、全く同一のGUIをUNIXとWindows上に実装しているため、家ではWindows、職場ではUNIXといったシームレスな設計環境を実現できる。MAXには、外部シミュレータを用いてポストレイアウトシミュレーションを行うために、Verilog、VHDLのネットリストを出力する機能を持つ。Verilogネットリストの場合、シミュレーションを行うための各セルの機能まで含めた記述が出力される。このネットリストを、簡単なフィルターにより変換して論理合成ツールに入力すれば、他のプロセス用に回路をリターゲットすることができる。

MAXを使った設計環境と、通常のASIC設計用のCADを用いたFPGA設計環境と比較してみる。図2に、Verilog-HDLやVHDLで記述して、合成ツール、シミュレーションツールを用いてネットリストレベルまでの設計を行い、EDIFを通じてMAXに入力する設計フローを示す。この方法は一般に広く行われている。しかし、すべて別のツールであり、個々にその使用法を学ばねばならず、各ツール毎に、ライブラリや環境変数等の設定も必要である。Verilog-XL等のシミュレータを用いるさいの最初の難関は、シミュレーション用の最上位階層の回路モジュール中に、波形ベクトルを書かなければならないことである(図3)。直感で考えると、回路があつて、その外側から波形を与えるのがわかりやすいやり方である。さらにHDLレベルでのシミュレーションは、何も考えずにできるが、合成後のネットリストレベルでのシミュレーションでは、ライブラリ設定、入力ポート順の確認等の、初心者のがはまりやすい項目が待ち構えている。

一方、MAXは先ほども述べた通り、FPGA

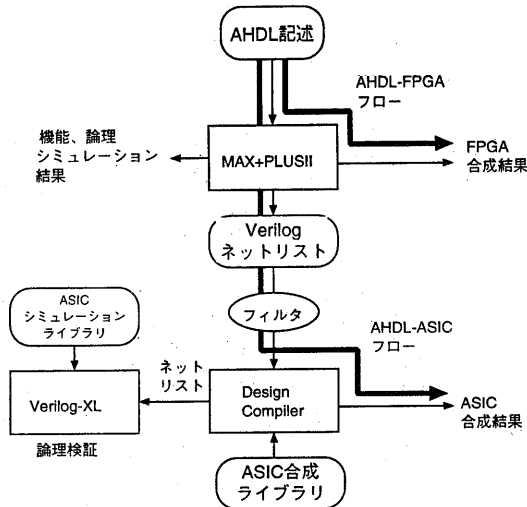


図 1: 提案する FPGA, ASIC 設計フロー

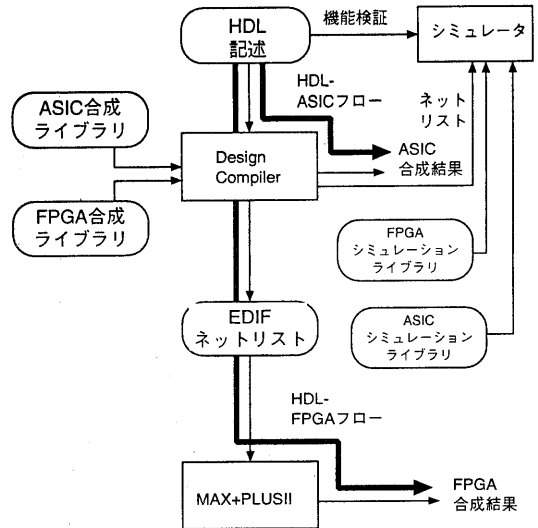


図 2: 従来の FPGA, ASIC 設計フロー

設計に必要なすべてのことが一つのツール内ですべて完結している。設計者はツールを立ち上げるだけで、何も設定することはない(もちろん、ライセンス関係の設定は必要である)。そのインタフェースは初心者にも優しいだけでなく、上級者でも使いやすいようになっている。例えば、シミュレーションへの波形ベクトルの与え方を見てみると、

1. 波形エディタによる GUI 入力
2. テキストファイルによる入力

の両方をサポートしており、初心者は GUI を、上級者はテキスト入力を用いることで、効率よい回路シミュレーションを行うことができる。大学等の教育機関の場合は、その用途を報告するだけで、無償でライセンスの交付を受けることができる。Altera の FPGA は、そのツールの使いやすさから信奉者が多く、同社の FPGA を用いたアプリケーションボードが多数サードパーティから発売されている。その中には、FPGA にプッシュスイッチ、LED 等をつけた教育向けのボードがあり、4節で述べる電卓や、Kuechip2 を実装してその動作を確かめることもできる。

3 合成環境

ここでは、

AHDL-FPGA フロー AHDL で記述し、MAX を用いて合成、マッピングを行ったもの。

HDL-FPGA フロー HDL で記述し、Design Compiler で合成、MAX を用いてマッピングを行ったもの。

および

AHDL-ASIC フロー AHDL で記述し、MAX を用いて合成・マッピングを行い、シミュレーション用に出力された Verilog ネットリスト記述を、Design Compiler に入力し、ASIC 向けにリターゲットングを行ったもの。

HDL-ASIC フロー HDL を用いて、Design Compiler で ASIC 用に合成したもの。

について、その性能を比較する。

それぞれのフローは、図 1、図 2 に記載してある。

```

module main;
  reg [9:0] decimal;
  reg plus,minus,equal,CLK,reset;
  // 中略
  extsyncro9 I4(_decimal,decimal,CLK,reset);
  bintobcd I5(out,lower,upper);
  initial $monitor("%d LCD=%d%d SIGN=%b",
    $time,upper,lower,sign,overflow,
    IO.p,IO.q,IO.state);
  // 入出力モニタ
  always #10 CLK = ~CLK ;// クロック
  initial // 入力波形
  begin
    decimal=0;plus=0;minus=0;
    equal=0;reset=1;CLK=0;
    #200 decimal=0;plus=0;minus=0;
    equal=0;reset=0;
    #200 decimal=1<<0;
    plus=0;minus=0;equal=0;reset=1;
  // 中略
endmodule

```

図 3: Verilog におけるシミュレーション用
トップモジュール

ASIC 用のライブラリとして、当研究グループで開発したプロセスポータブルライブラリ P2LIB[2] を用いた。ターゲットプロセスは、0.5 μ m² 層メタル CMOS プロセスとした。FPGA は、ALTERA 社の FLEX10K シリーズを用い、Design Compiler 用のライブラリも同社提供のものを利用した。

4 回路合成実験結果

ここでは、FPGA 上で実現できる回路として、2K ゲート程度の「加減算のみ可能な 2 桁 BCD 電卓」の Verilog-HDL および AHDL 記述、4k ゲート規模の「教育用マイクロプロセッサ kuechip2」[3, 4, 5] の VHDL および AHDL 記述に対する合成実験を行った。

4.1 BCD 電卓

図 4, 5 に電卓のブロック図および状態遷移図を示す。この電卓を、AHDL, Verilog-HDL の両者で同一の設計者が記述した。AHDL で

は 354 行、Verilog-HDL では 293 行となった。AHDL の記述量が大きいが、これは AHDL が、状態遷移時のすべてのレジスタの値を明示的に書かなければならないためである。

FPGA の合成結果を検証する。Verilog-HDL で合成したものが、AHDL に比べて面積で 80%、速度で 70% 良くなっている。両者とも面積優先で合成を行ったが、MAX では、面積優先にしても、速度優先の場合と比べてほとんど回路規模が変化なかった。両者を、ASIC ライブラリで合成したところ、速度面積ともに、ほとんど大差ない回路が作成できた。これは、Design Compiler の最適化能力の高さをはからずも示す結果となった。しかし AHDL で FPGA 用に設計した回路が、Verilog のネットリスト記述を介して Design Compiler に渡すだけで、最初から Verilog で記述した回路と遜色ない回路が合成されることが証明できた。

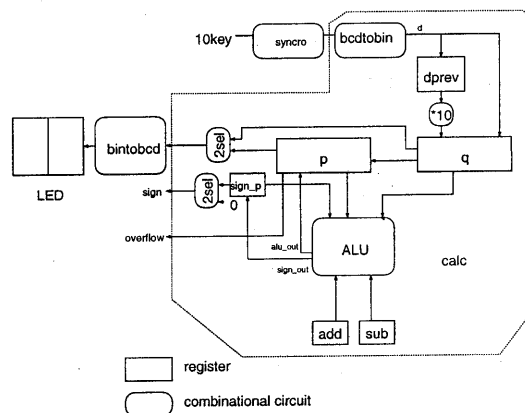


図 4: 電卓のブロック図

4.2 Kuechip2 の合成結果

Kuechip2 は、8 ビットの教育用マイクロプロセッサである。その仕様は一般に公開され様々なデザインコンテストの教材や、ボードを使ったマイコン教育等に広く利用されている。その LSI は、回路図レベルからの手設計により設計されたが、論理合成技術の進歩により、一人の設計者が HDL を用いて記述できるように

表 1: BCD 電卓の合成結果

	AHDL		Verilog-HDL	
	面積	動作速度	面積	動作速度
FPGA	294LCs	16.6MHz	250LCs	25.8MHz
ASIC	2783	129MHz	2748	120MHz

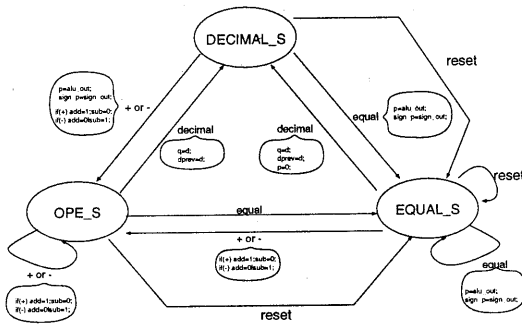


図 5: 電卓の状態遷移図

なった。ここでは、既存の VHDL RTL レベル記述と、今回新たに記述を行った AHDL 記述とで、その回路規模、動作速度、合成にかかる時間の検証を行う。両記述とも、Kuechip2 の企画からボード設計まですべてをマネジメントした同一の設計者によって作成された。表 2 に両記述を比べる。

表 2: Kuechip2 の VHDL, AHDL 記述の比較

	VHDL	AHDL
記述量	68kB	35kB
設計期間	80 時間	40 時間

表 3 に、結果をまとめる。合成時間は、すべて Sun の Ultra SPARC2 300MHz 上での結果である。結果を見ると、面積、動作速度、合成時間のすべての項目において、AHDL での記述が勝っていることがわかる。面積では、FPGA の場合で、VHDL 記述は AHDL の 1.4 倍となったが、速度面ではほぼ同等である。ASIC の場

合では、面積、速度ともに 1.3 倍となった。特筆すべきなのは、その合成時間である。AHDL で記述したものが 30 秒とほんのわずかの時間で合成できるのに対し、VHDL ではその 25 倍もの時間がかかってしまう。MAX が出力した Verilog ネットリスト記述を、Design Compiler でリターゲットした場合でも、VHDL をそのまま合成するのに比べて約 1/5 の時間で済んでいる。

VHDL の回路規模が大きくなるのは、そもそも記述の量自体が多いことに起因している。VHDL は、もともと設計ドキュメントのための動作記述言語から出発している。その意味定義が曖昧なため、確固とした記述法が存在せず、設計者は各論理合成ツールに合わせた記述をしているのみである。記述そのものから、合成される回路を予測することは困難である。一方、AHDL は、FPGA に実装する回路を合成するための設計言語である。SFL と同様、記述からどのような回路が合成されるかが比較的読みやすい。

設計者は、純国産の HDL 合成環境である UDL/I 処理系 [6] を構築した経験があり、HDL そのものの理解度は非常に高い。両記述を策定した当初は、VHDL, AHDL ともに何も知らない状態からはじめていた。したがって、多少の記述の最適性の問題はあるが、VHDL は回路合成時のオーバーヘッドが大きいといえるだろう。蛇足ではあるが、設計中に MAX からのエラーメッセージにより、VHDL 記述のバグを発見することができた。

5 結論

本稿では、FPGA 用統合環境を用いた、ASIC 設計の性能を議論した。2K ゲート程度

表 3: Kuechip2 の合成結果

	AHDL			VHDL		
	面積	動作速度	合成時間	面積	動作速度	合成時間
FPGA	485LCs	20.0MHz	30 秒	680LCs	19.4MHz	754 秒
ASIC	4025	152MHz	368 秒	5313	119MHz	1585 秒

の回路規模を有する「BCD 電卓」を、

1. AHDL で記述し、MAX で、FPGA 用に設計した後、Verilog ネットリストに変換し、Design Compiler で、ASIC 用に再合成したもの。
2. Verilog-HDL で記述し、Design Compiler で ASIC, FPGA 用に合成したもの。

の 2 種類の手法で設計した結果、ASIC では、面積、動作速度がほぼ同等な回路が得られた。「教育用マイクロプロセッサ Kuechip2」を、VHDL, AHDL の両記述で比べたところ、AHDL で記述した方が面積、動作速度ともに良好な回路が得られた。

今後は、本設計手法を利用した FPGA, ASIC 設計教育プログラムの開発を行う予定である。なお、Verilog-HDL で記述した「BCD 電卓」は、98 年 12 月上旬に行われる「VDEC リフレッシュ教育 VLSI 設計教育コース」で利用する予定である。

参考文献

- [1] 清水尚彦, 吉田正廣, 野間泉: FPGA を用いた集積回路設計教育システムの開発ならびに大人数を対象とする集積回路設計教育カリキュラムの検討, 電子情報通信学会技術報告, VLD97-103, CPSY97-91(1997-12)
- [2] H. Onodera, A. Hirata, T. Kitamura and K. Tamaru: "P2Lib: Process-Portable Library and Its Generation System", *Proc. of the IEEE 1997 Custom Integrated Circuits Conference*, pp.341-344, May (1997).
- [3] 神原 弘之: 教育用マイクロプロセッサ: KUE-CHIP, 電子情報通信学会技術報告, ET 90-106, 1990
- [4] 神原 弘之: 教育用マイクロプロセッサ: KUE-CHIP, 情報処理学会研究報告, 90-ARC-82, 1990
- [5] 越智 裕之, 澤田 宏, 岡田 和久, 濱口 清治, 上嶋 明, 神原 弘之, 安浦 寛人: 計算機工学・集積回路工学教育用マイクロプロセッサ KUE-CHIP2, 情報通処理学会研究報告, 90-ARC-80, 1993
- [6] Hiroyuki Kanbara, Satoshi Yokota: Validation of UDL/I Test Suites and UDL/I Simulation/Synthesis Environment, 電子情報通信学会英文論文誌 A, Vol. E78-A, No.12, pp.1749-1754, 1995