

## 高位合成手法を用いた C ベース設計による LSI 開発事例

松田 昭信<sup>†</sup>南谷 崇<sup>‡</sup>放送大学教養学部<sup>†</sup>東京大学先端科学技術研究センター<sup>‡</sup>

## 1. はじめに

現在、各種デジタルシステムの高機能化に従い、LSI 設計における回路の大規模化が進むと共に、開発工数が増大し、開発工程を含め複雑化している。その一方で、開発期間は一層の短縮が求められている。こうした要求における開発期間短縮を実現するため、HDL を用いてハードウェアの設計記述を行う従来の設計手法から、C ベース言語を用いてシステムをより高い抽象度で記述する手法に設計手法が移行しつつある<sup>(1)</sup>。本稿では、システムレベル記述から詳細設計まで一貫して C 言語を用いて設計可能となる、高位合成手法を用いた効率的な LSI 開発事例について述べる。

## 2. システムレベル設計の課題

システムレベル設計において、例えば、設計言語一つとってみても、言語仕様が少しずつ異なる SystemC、SpecC など様々な記述言語がある。そして、これらの言語記述スタイルはソフトウェアプログラムのスタイルとも異なり、ソフトウェア記述との統合についても課題を残しているため、設計ツールを含めた高位合成手法が確立していない。よって、ANSI-C 言語でシステム LSI を設計することにより、RTL モデルに比べ、10~100 倍の速さでシミュレーションが可能となり、検証にかかる工数が削減できる。これにより、全体の工数が削減でき、早期のバグ発見、修正、検証が可能となる。

## 3. システムレベル設計手法

今回、システムレベル設計に対応した高位合成ツールを用いて、FPGA を設計した。C 言語を用いてハードウェア設計を行うためには、記述された C 言語を高位合成して、RTL 記述に変換しなければならない。一旦 C 言語で記述されたハードウェア記述を RTL まで落としてしまえば、従来の論理合成ツールを使用できる。図 1 では、

```
#include "cbased.h"

void counter (
    unsigned char clk, // pclk:1
    unsigned char reset, // anrst:1
    unsigned int *dout // output:4
)
{
    unsigned int tmp; // bit size : 4
    *dout = 0;
    tmp = 0;
    main_loop :
        while(1) {
            tmp = tmp + 1;
            *dout = tmp;
        }
}

void main ()
{
    unsigned char clk;
    unsigned char reset;
    signed int dout;
    counter (clk, reset, &dout); //module
}
```

図 1. 高位合成可能な C 言語コード

Fig.1. High-Level Synthesizable C Language Code

高位合成可能な非同期リセット付 4 ビットカウンタの簡単な記述例を示す。C 言語コードに、入出力ポート及び信号のビット幅などを定義した。例えば、pclk はポジティブエッジクロック、anrst はローアクティブ非同期リセットを示す。また、main\_loop はリセットブロックとメイン処理を区別している。これら高位合成ツールを活用することにより、設計者は C 言語を RTL 記述に書換える工数や人為的ミスがなくなるというメリットがある。

## 4. LSI 開発事例

今回の開発した JPEG エンコーダには、ターゲットデバイスの FPGA である XILINX 社 Spartan™ シリーズを使用した。図 2 は、今回の開発事例

A Development Example of LSI on C-based Design with High-Synthesis Methods

<sup>†</sup> Akitoshi Matsuda : The University of the Air

<sup>‡</sup> Takashi Nanya : The University of Tokyo

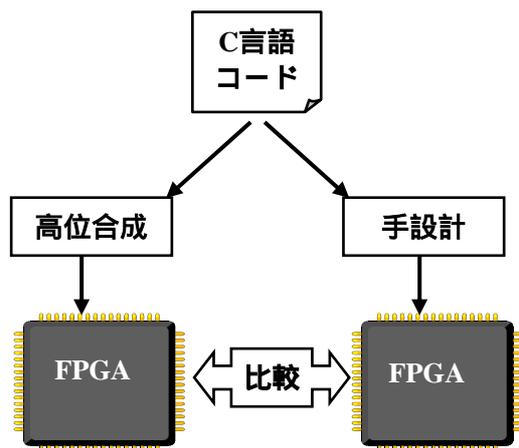


図 2 . 評価方法  
Fig.2. Evaluation Methods

手設計工数

仕様検討	RTL作成	Sim検証	全体検証
------	-------	-------	------

C ベース設計工数

仕様検討	Sim検証	全体検証
------	-------	------

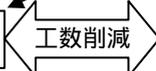


図 3 . 設計工数の比較  
Fig.3. Design Works Comparison

を比較した評価方法を簡易的に示した。図 3 は、今回用いた C ベース設計手法により、従来の設計工数からどれだけ短縮したかを簡単に表記した。C ベース設計手法を用いることにより、C 言語から RTL の書き換え作業が不要となり、また、シミュレーションに要する時間も短縮され設計工数が短縮される。さらに、C ベース設計手法を用いることにより、システムレベル設計が終了すると同時に、トータルの検証・ハードウェア設計・ソフトウェア設計が同時に進行できる。これにより、イタレーション及びリスピが減り、全体的な設計工数も短縮される。今回の設計事例では、設計工数が 20 人日から 15 人日に減少しており、全体の 25% の工数削減が実現できた。また、C 言語から RTL の書き換え作業において、高位合成手法を用いて自動生成した回路と、人為的に RTL を記述して生成された回路の性能比較を実施して、その結果を図 4 に示す。周波数は最大動作周波数を用い、面積は FPGA における Logic Cell 数によって比較した。また、要求最大動作周波数は 100MHz をターゲットにして論理合成を実行した。この結果からわかるように、最大動作周波数で約 20%、面積で約 30% 向上した。

	高位合成	手設計
最大周波数 (MHz)	134.15	111.12
面積 (LCs)	9358	13021
スループット	5	5
レイテンシ	118	156
高位合成時間 (s)	215	

図 4 . 回路性能比較  
Fig.4. Circuit Performance Comparison

	ビット幅指定	ループ展開
最大周波数 (MHz)	171.25	143.76
面積 (LCs)	7375	8867
スループット	5	5
レイテンシ	118	97
高位合成時間 (s)	215	215

図 5 . 回路性能比較  
Fig.5. Circuit Performance Comparison

さらに、この高位合成手法において、ビット幅の最適化やループ展開を事前に効果的に設定してやることにより、さらに性能のよい回路が生成されることが判明した。この結果を図 5 に示した。

5 . まとめ

C ベース設計手法により、C 言語を FPGA のシステム記述から詳細設計まで一貫して用いることによって、システムレベル検証が効果的かつスピーディに実行できることがわかった。詳細結果については、当日報告する。また、この手法はデザインのターゲットによっても、メリットの度合いが大きく違う。よって、今後は手続き的な処理やデータフロー的な処理など、様々なターゲットに対して比較検討していく予定である。

6 . 今後の課題

現状のレベルでは、まだ完全にソフトウェア C 言語をサポートしておらず、例えば、グローバル変数の使用、関数の再帰呼び出しなどの対応が出来ていない。よって、依然ハードウェアを意識した記述が必要であるケースもある。また、入力した C 言語コードと出力された RTL コードの等価性のプロパティチェックが十分に実施できない。これらの課題対応についても今後検討を継続していく予定である。

参考文献

- (1) 松田昭信：信学技報、Vol.101.No.473. 37 ~ 42 (2001)