

# 並列アルゴリズムの IP コア ライブラリの構築に関する研究

伊 藤 靖 朗      広島大学大学院工学研究科助教

## 1 はじめに

デバイス技術の進歩により、高速なプロセッサが開発され、様々な計算処理が短時間でできるようになってきた。しかし、近年プロセッサの動作周波数の増加も鈍化しており、高速化に限界が見えてきた。そのような中、FPGA(Field Programmable Gate Array)を用いた計算の高速化手法が注目されている。FPGA は書き換え可能な LSI であり、計算機からダウンロードされた回路情報データに基づいて結線を実現することによって任意の回路をチップ内に実現するというものである。従来 FPGA は ASIC のプロトタイプ設計用として利用されていたが、近年の高速・大規模化に伴い、書き換え可能である特徴とともに高速計算の用途への利用が期待できるようになった。

一方、過去の研究において、並列アルゴリズムが数多く発表されてきた。しかしそれらの多くは理論のみで、並列処理をおこなうハードウェアアルゴリズムでもチップ化まで至らず、実用化されることはほとんどなかった。これは、理論上有効だが実用的でないアルゴリズムの場合もあるが、それらのなかには当時の計算機環境やハードウェア技術の制約のため実用化できなかったものも数多く存在すると考えられる。

そこで本研究では、様々な並列処理アルゴリズムを、ハードウェアの機能単位である IP (Intellectual Property) コアとして実装し、それをハードウェア全体の一部品として FPGA 上で動作させ、性能評価をおこなう。そしてその結果を並列アルゴリズム IP コアライブラリとしホームページ上で公開し、広く利用可能にすることを目的とする。

本研究では基本回路の IP コア化と二値画像の並列ラベリングハードウェアの設計を行った。次にこれらについて説明する。

## 2 基本回路の IP コア化

並列処理アルゴリズムでは、それらを実装する上で必要となる様々な基本的な回路が必要となる。そこで本研究では、基本的な回路として、加算器、乗算器、マルチプレクサ、シフタ、カウンタ等の基本的な回路の IP コア化を行った。このとき、作成する並列処理アルゴリズムによって回路サイズや動作スピードなど、必要とする性質が異なる実装の IP コアを設計した。例として、加算器では Carry-lookahead adder や Ripple-carry Adder, Carry-save Adder など様々な実装で IP コアを設計した。

## 3 二値画像の並列ラベリングハードウェアの設計

### 3-1 二値画像のラベリング

二値画像のラベリング処理は、オブジェクトの特徴を求める前段階の処理として画像認識の分野において基本的な処理の一つである。画像のラベリングの方法は Rosenfeld と Pfaltz らが最初に提案し [1]、その後、様々な方法が考案されてきた。画像の連結成分のラベリングの代表的な方法を挙げると

- ・ ラスタスキャン順に画像を走査し仮ラベルを割り当て、異なる仮ラベルを割り当てられた画素と隣接したときに仮ラベルの連結関係を記録する。走査の途中、または走査後に、連結関係から最終的に割り当てるラベルを計算し、もう一度走査し仮ラベルを付け替えることによって最終的なラベルを割り当てる方法 [1] [2]。
- ・ オブジェクトの輪郭を見つけ、その内部に同じラベルを割り当てる方法 [3] [4]。
- ・ 4 分木などの木構造を用いて画像を階層的に表し、各階層で画像の連結関係を調べてラベリングを行う方法 [5] [6]。

上記の方法をハードウェアで実装することで高速にラベリングを行う方法が提案されてきた [7] [8] [9]。しかし、これらの手法は 1 画像分の入力に対して 2 回以上画像の走査が必要で、画像中のオブジェクトの形状に計算時間が依存していた。

カメラからの入力等，連続的に画像が入力される場合，1つの画像を処理するために，いくつかの画像を処理できず，入力される画像すべてにラベリングを行うことができなかった．そこで本研究では，画像の形状に制限がある場合に，連続的に入力される画像に対して，1画素当たりの処理を1クロックサイクルで行い，FPGA(Field Programmable Gate Array)上で実装可能なラベリングのハードウェアアルゴリズムを提案する．

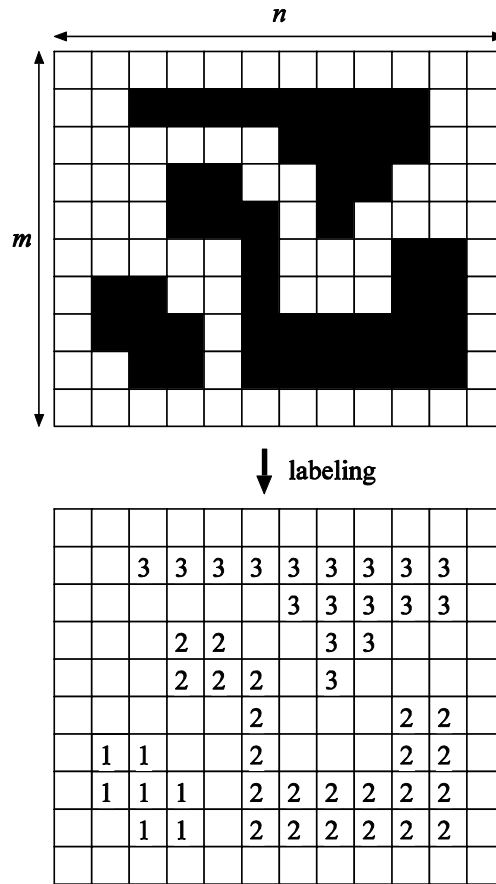


図 1：二値画像のラベリング

### 3-2 定義

k-concave な画像とは，二値画像の連結成分の形状を制限したもので，連結成分中のすべての2画素の組について，それらを結ぶパス上の画素が，下側の端点より k 画素下側に存在しないとき，その二値画像を k-concave な画像とする．図 2に 3-concave な画像の例を示す．図中の2点 A, B のそれらを結ぶパス上の画素が，下側の端点より 3 画素下側に存在しないことがわかる．

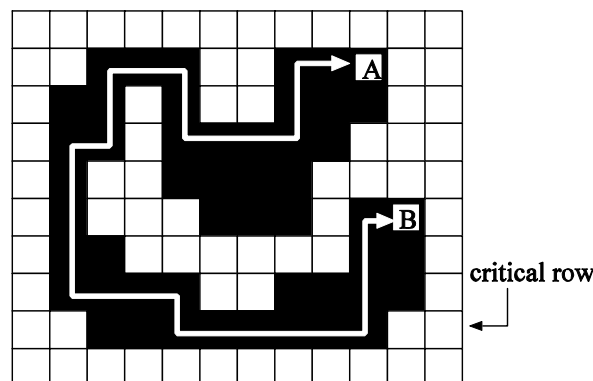


図 2：3-concave 画像

本手法では、ラストスキャン順に入力される  $k$ -concave な二値画像に対して、最初に二値画像の基本的な構造を表す仮ラベル Prefix LMRS labels を割り当てる。これを用いて 0-prefix tentative labels, 1-prefix tentative labels, 2-prefix tentative labels, ...,  $k$ -prefix tentative labels を割り当て、その結果から最終的なラベリングを行う (図 3)。次にこれらの仮ラベルについて説明する。

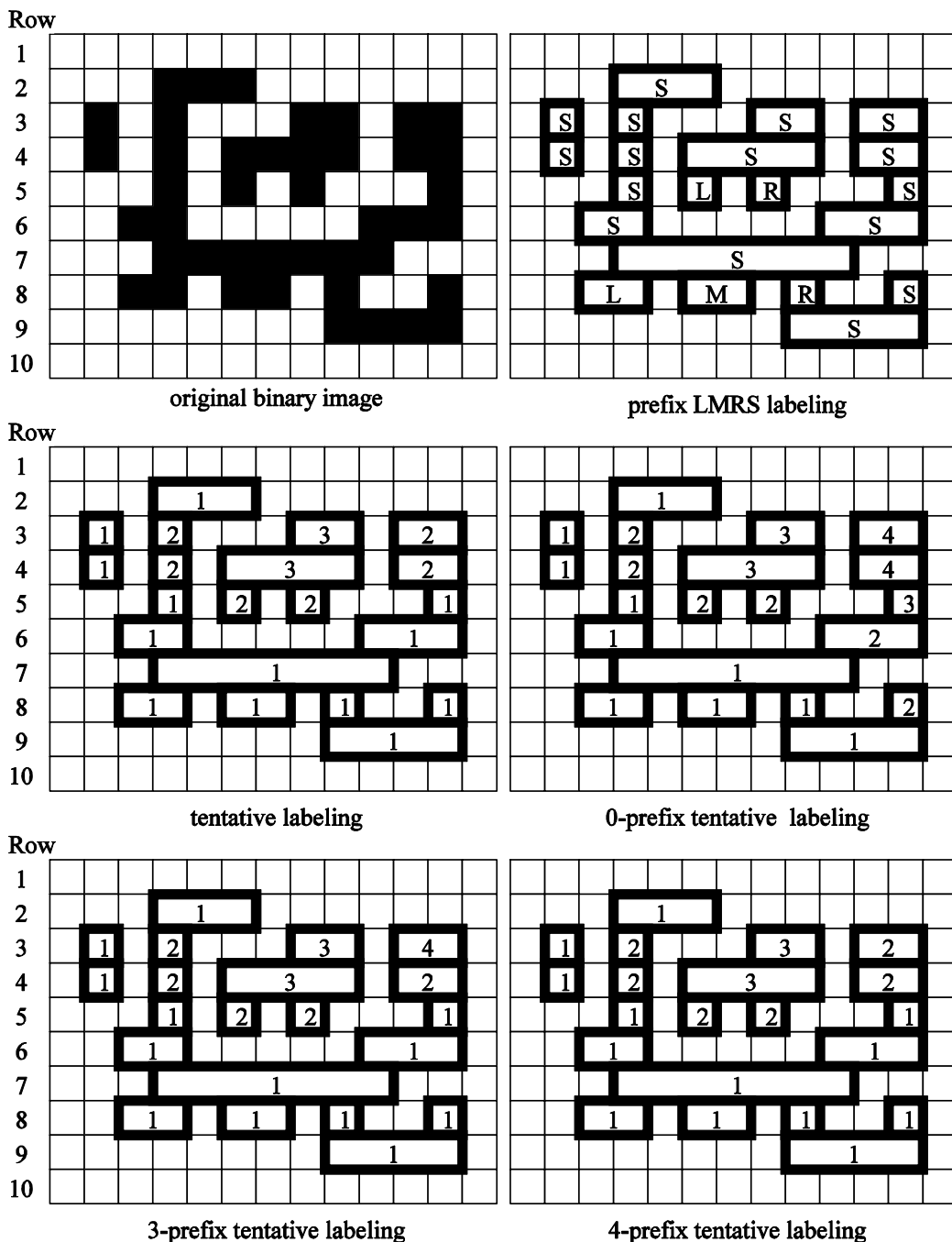


図 3 : 二値画像とその仮ラベル

#### (1) Prefix LMRS labels

Prefix LMRS labels とは、1つの行における連結成分の構造を表す4つの記号 S, L, M, R から構成され、各 run に記号の一つを割り当てる。このとき run は、行中で連続するピクセルの集合とする。Prefix LMRS Labels は  $i$  行目の記号を割り当てるときに、1行目から  $i$  行目までの部分画像の連結成分の構造から記号を割り当てる。つまり、Prefix LMRS Labels は、割り当てる run が、ある行より下の画像の構造に依存せずに一意に決めることができる。

(2) tentative labels

tentative labels は、各行において、その行の各 run が同じ連結成分に属するときに同じラベルを割り当てたものである。

(3) k-prefix tentative labels

k-prefix tentative labels は、 $i$  行目の各 run に 1 行目から  $i+k$  行目の部分画像で同じ連結成分毎にラベルを割り当てたものである。つまり、 $k$ -concave な画像の k-prefix tentative labels を求めることは、その画像の tentative labels を求めることと同義である。

3-3 並列ラベリングハードウェア

本手法では、入力二値画像はラスタスキャン順に入力されるものとし、数行毎にパイプライン処理を行う。このとき、各行の処理の順序を図 4 に示す。

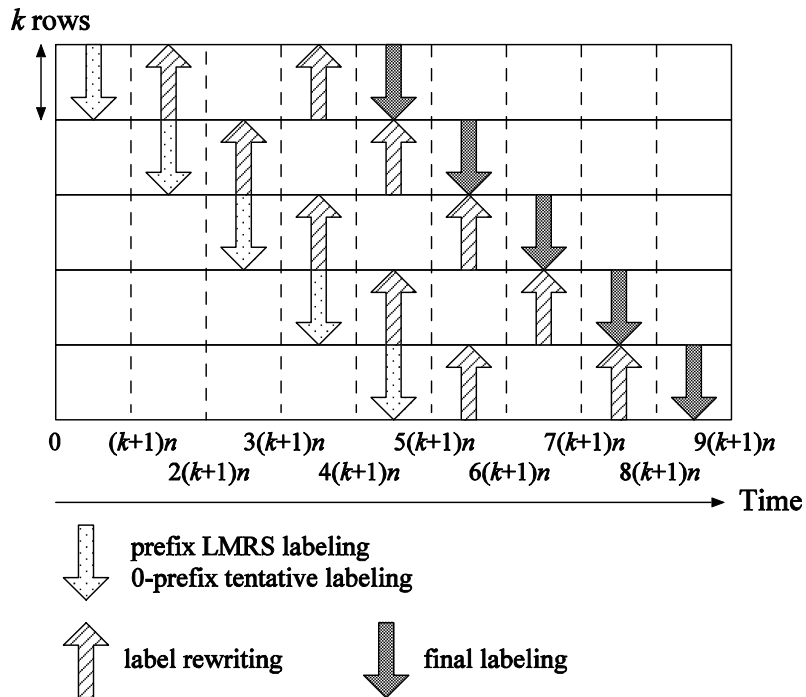


図 4：各行の処理の順序

このように数行毎にパイプライン処理することによって、回路規模が大きくなることを抑えつつレイテンシ、スループットを向上させることに成功した。このときのハードウェアの構成を図 5 に示す。

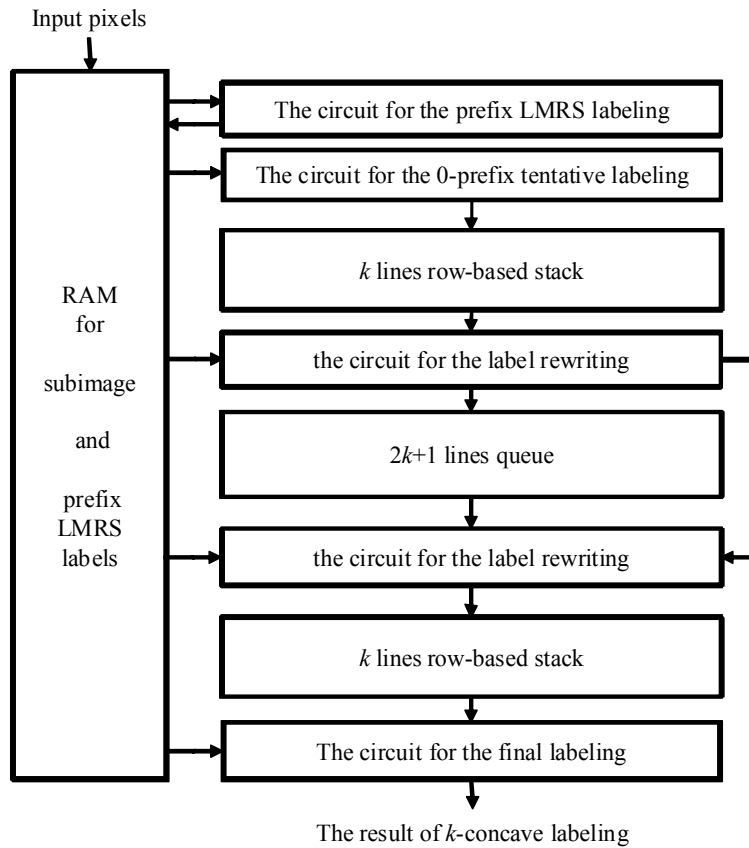


図 5 : ハードウェア構成

### 3-4 性能評価

作成したハードウェアの性能評価を行った. FPGA は 250 万ゲート相当の Altera 社の Stratix シリーズである EP1S25 を使い, 論理合成ツールとして Altera 社の Quartus II を使用した. サイズが  $2048 \times 2048$  の二値画像に対して 0 から 20-concave な画像に対する回路規模 (Logic Elements) と最大動作周波数の論理合成結果を表 1 に示す. 論理合成結果をより, 本手法では, サイズが  $2048 \times 2048$  の 20-concave な二値画像に対してラベリングを行った場合, 実行時間が約 71ms, レイテンシが 1.4ms で処理可能なことを確認した.

表 1 :  $k$ -concave 画像に対するハードウェアの回路規模と最大動作周波数

| k  | Logic Elements | Maximum Frequency [MHz] |
|----|----------------|-------------------------|
| 0  | 1596           | 72.66                   |
| 1  | 3688           | 65.8                    |
| 2  | 3892           | 67.51                   |
| 3  | 4068           | 63.86                   |
| 4  | 4242           | 66.16                   |
| 5  | 4441           | 64.87                   |
| 6  | 4615           | 67.73                   |
| 7  | 4876           | 65.23                   |
| 8  | 5094           | 63.1                    |
| 9  | 5275           | 64.69                   |
| 10 | 5471           | 64.72                   |
| 11 | 5544           | 63.43                   |
| 12 | 5717           | 66.05                   |
| 13 | 5887           | 65.52                   |

|    |      |       |
|----|------|-------|
| 14 | 6071 | 64.9  |
| 15 | 6251 | 63.14 |
| 16 | 6282 | 65.48 |
| 17 | 6500 | 65.92 |
| 18 | 6706 | 62.81 |
| 19 | 6912 | 64.78 |
| 20 | 7124 | 62.75 |

#### 4 まとめ

IP コアライブラリは、ハードウェアベンダーやオープンソースコミュニティなど、商用・非商用問わず数多く存在する。しかしそれら IP コアライブラリの多くは、CPU や DSP、算術演算などハードウェアで頻繁に利用される機能を IP コア化したものが大半で、並列アルゴリズムが元となる IP コアはほとんど存在しなかった。しかし、組合せ最適化問題などの並列アルゴリズムは多くのアプリケーションにおいて中心的な処理となる場合が少なくない。そこで上記で説明した基本回路をまとめ、IP コアライブラリとして公開し、広く使われるようにしている。今後も IP コアの作成を続け、ライブラリの充実を図る予定である。

#### 【参考文献】

- [1] A. Rosenfeld and J. L. Pfaltz. Sequential operations in digital picture processing. *Journal of the ACM*, 13(4):471–494, 1966.
- [2] K. Wu, E. Otoo, and A. Shoshani. Optimizing connected component labeling algorithms. In *Medical Imaging 2005: Image Processing. Proceedings of the SPIE, Volume 5747*, volume 5747, pages 1965–1976, Jan. 16 2005.
- [3] P.-E. Danielsson. An improved segmentation and coding algorithm for binary and nonbinary images. *IBM Journal of Research and Development*, 26(6):698–707, 1982.
- [4] F. Chang, C. Chen, and C. Lu. A linear-time component labeling algorithm using contour tracing technique. *Computer Vision and Image Understanding*, 93(2):206–220, 2004.
- [5] H. Samet and M. Tamminen, “Efficient component labeling of images of arbitrary dimension represented by linear bintrees,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 4, pp. 579–586, 1988.
- [6] A. Unnikrishnan, P. Shankar, and Y. V. Venkatesh, “Threaded linear hierarchical quadtrees for computation of geometric properties of binary images,” *IEEE Transactions on Software Engineering (SE)*, vol. 14, no. 5, pp. 659–665, May 1988.
- [7] M. Jablonski and M. Gorgon, “Handel-C implementation of classical component labelling algorithm,” *Digital System Design*, 2004. *Euromicro Symposium on*, pp. 387–393, 2004.
- [8] K. Benkrid, S. Sukhsawas, D. Crookes, and A. Benkrid, “An FPGA-based image connected component labeller,” in *Field-Programmable Logic and Applications*, ser. *Lecture Notes in Computer Science*, P. Y. K. Cheung, G. A. Constantinides, and J. T. de Sousa, Eds., vol. 2778. Springer, 2003, pp. 1012–1015.

[9] C. Schmidt and A. Koch, "Fast region labeling on the reconfigurable platform ace-v," in Field Programmable Logic and Application, 13th International Conference, FPL 2003, Lisbon, Portugal, September 1-3, 2003, Proceedings. Springer, 2003.

〈 発 表 資 料 〉

| 題 名  | 掲載誌・学会名等  | 発表年月        |
|--|---|-------------|
| Optimized Component Labeling<br>Algorithm for using in Medium Sized<br>FPGAs | Proceeding of International<br>Conference on Parallel and<br>Distributed Computing,<br>Applications and Technologies,<br>pp. 171-176. | 2008 年 12 月 |