

高性能演算実現のための
光 Race Logic の提案と評価

浅井里奈

平成 28 年 12 月

情報知能工学科

高性能演算実現のための 光 Race Logic の提案と評価

浅井里奈

内容梗概

配列アラインメントナノフォトニクス高性能化

目次

1	はじめに	1
2	Race Logic とその実装に関する課題	3
2.1	Race Logic	3
2.2	CMOS によるレースロジック実装	3
2.3	解決すべき課題	3
3	ナノフォトニック・デバイスを用いた Race Logic 実装	4
3.1	光デバイスについて	4
3.2	ナノフォトニック・デバイスに夜 Race Logic 実装	4
4	ケーススタディ：配列アラインメント	5
4.1	配列アラインメントの基本原理	5
4.2	設計選択肢	5
4.3	提案回路	5
5	評価・考察	6
5.1	評価	6
5.2	考察	6
6	おわりに	7
	謝辞	8
	参考文献	9

第 1 章

はじめに

高性能と低消費電力を実現させるために、多くのアプリケーションにおいて専用アクセラレータが検討されてきた。その中でも、動的計画法（Dynamic Programming, DP）によって解決されるような広範なクラスの最適化問題を加速するために、“Race Logic”と呼ばれる新しいコンピューティングのアプローチが提案された [1]。Race Logic の基本概念は、回路に設定された競合条件を使用して有用な計算を実行することである。Race Logic はその実装において複数の設計選択肢がある。しかしながら、現状では従来の相補型金属酸化物半導体（CMOS）テクノロジーで実装可能な Race Logic のみしかその有効性が明らかになっていない。

一方、ナノフォトニクスと呼ばれる新しい光素子技術が注目を集めている。このナノフォトニクスを用いて機能を実現したデバイスをナノフォトニック・デバイスという。ナノフォトニック・デバイスは光速度で演算を実現できる素子として注目されており、パターン検出機構や加算器などのアーキテクチャの検討がされている。

本論文では、Race Logic の更なる高性能化を目的とし、ナノフォトニック・デバイスを用いた実装を提案する。より具体的に評価を行うために、ナノフォトニック・デバイスを用いた光 Race Logic（以下、光 Race Logic）の性能を DNA グローバル配列アライメントタスクの例を用いて検討する。

本論文の構成は以下の通りである。第 2 章で Race Logic の基本原理、及び実装に関する課題を説明する。第 3 章では、光デバイスの基礎事項と共にナノフォトニクスの基本事項を説明し、光 Race Logic の実装について述べる。第 4 章ではケーススタディとして配列アライン

メントを取り上げる．配列アラインメントの原理説明を行った後，設計選択枝の整理，回路の提案を行う．第 5 章では提案した回路に対して評価と考察を行い，第 6 章でまとめを行う．

第 2 章

Race Logic とその実装に関する課題

本章では,

2.1 Race Logic

2.2 CMOS によるレースロジック実装

2.3 解決すべき課題

第 3 章

ナノフォトリック・デバイスを用いた Race Logic 実装

3.1 光デバイスについて

3.2 ナノフォトリック・デバイスに於 Race Logic 実装

第 4 章

ケーススタディ：配列アラインメント

4.1 配列アラインメントの基本原理

4.2 設計選択肢

4.3 提案回路

第 5 章

評価・考察

5.1 評価

5.2 考察

第 6 章

おわりに

謝辞

本研究の進行および本論文執筆にあたりまして，懇切丁寧なご指導を頂きました井上弘士教授に心より感謝申し上げます．本論文執筆にあたり，多大なご指導を頂きました小野貴継助教に心より感謝申し上げます．本研究を行うにあたり，多大なご指導を賜りました日本電信電話株式会社 物性科学基礎研究所 ナノフォトニクスセンタ 主幹研究員 新家昭彦様に感謝の意を表するとともに厚く御礼申し上げます．本研究に際し，データの提供を頂きました京都大学大学院 情報学研究科 通信情報システム専攻 石原亨准教授に心より御礼申し上げます．本論文執筆にあたり多大な指導頂きました，井上研究室大学院修士1年磯部聖氏に深く感謝致します．

最後に，井上研究室の皆様の御意見，御厚意に感謝の意を表します．

参考文献

- [1] Advait Madhavan, Timothy Sherwood, and Dmitri Strukov. Race logic: A hardware acceleration for dynamic programming algorithms. *ACM SIGARCH Computer Architecture News*, Vol. 42, No. 3, pp. 517–528, 2014.