
LINEAR NON-COMPARISON SORT

Michael You*
youmichaelc@gmail.com

September 18, 2022

ABSTRACT

We introduce a non-comparison based sort that runs in linear time, with fixed number bit size, and logic gates of linear size. We show that this sort can be implemented with very simple hardware components, which makes it a good candidate for a sorting algorithm on ASICs or FPGAs.

Keywords Computational Complexity · Circuits · Sorting · Transistors · Parallel

1 Introduction

When we compare two numbers to find the larger one, we are essentially checking digits from the most to least significant bit, and the first bit that is less than the corresponding bit in the other number, is the number that is less. To give an example, consider the following algorithm:

```
LARGER_NUMBER( $a, b$ ):  
  Suppose  $a = a_1a_2a_3 \dots a_k$  and similar for  $b$   
  
  for  $i \in (1, 2, \dots, n)$ :  
    if  $a_i > b_i$ :  
      return  $a$   
    else if  $a_i < b_i$ :  
      return  $b$   
  
  return EQUAL
```

With k binary bits, we can represent 2^k numbers, so it is usually the case that we consider fixed-size integer comparisons on computers constant time, since for $O(k) = O(1)$ for k constant.

A question we may think about with this number comparison algorithm is, *can we compare an arbitrary number of integers and find the largest one in k rounds?*

The idea for this paper is to develop such a parallel comparison algorithm, and apply it to sorting, where we can create a $O(n)$ time algorithm.

This sort was already developed by Ghosh et al. [2019] in 2019, but lacked formal details about the correctness of the proof, and justifications for time (and space) complexity, so this paper goes more in depth in those areas². Their paper demonstrated that this algorithm can be implemented on FPGAs, with small area size and with fast runtimes.

*Github: mikinty, graduate of Carnegie Mellon University, M.S. in ECE

²I also came up with this sorting algorithm independently, but that's not really a big deal because I doubt this algorithm is a new algorithm.

2 Algorithm

For the rest of the paper, we are going to assume that we are sorting binary numbers with k bits.

We first introduce the algorithm for finding the largest of n numbers.

References

Surajeet Ghosh, Shaon Dasgupta, and Sanchita Saha Ray. A comparison-free hardware sorting engine. In *2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 586–591, 2019. doi:10.1109/ISVLSI.2019.00110.