Asymptotically Faster Circuit Topologies

Michael You*
youmichaelc@gmail.com

November 12, 2020

Abstract

We are familiar with using the Turing paradigm to create computers, representing information in binary bits. However, there are more natural ways to solve problems without using bits. For example, if you have a graph, why not just find a natural way to represent it so the computation can be done better? That's the motivation of the paper. We will explore designing more natural circuit topologies to solve problems, and analyze their computational complexity. We will find that for sorting, we can come up with an algorithm that is faster than existing algorithms, and shortest paths in graphs that we can perform asymptotically better than existing algorithms. We will then explore the possibility of extending more natural circuit topologies to solve other problems faster, and applications of using this circuit topology technique to solve algorithmic problems in general.

Keywords Computational Complexity · Circuits · Diodes · Graphs

1 Introduction

Turing machines

efficient for information storage, but is it the best way to compute things? Best way to represent everything?

There can be more natural ways to represent problems.

2 Definitions

Definition 1. An **ideal diode** is a circuit element that has infinite current when on, and 0 current when off. The threshold voltage[2] for when it is on and off is called V_{D} and is measured from the $+$ to the $-$ terminal.

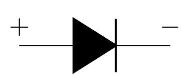

Figure 1: A diode, with the positive and negative terminals marked.

Definition 2. A **voltage source** is a circuit element that maintains a voltage of V between 2 nodes.

In this paper, we will be using the following element to represent a voltage source.

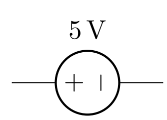

Figure 2: A voltage source

*Github: mikinty

[2]we sometimes call this the "turn-on voltage"

Definition 3. An **ammeter** is a device used to measure the current at some node of a circuit. We will use the notation

$$I(A_i) \tag{1}$$

to describe the current measured by ammeter A_i.

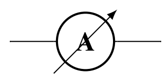

Figure 3: An ammeter

Definition 4. A **switch** is a device that can be set to 2 configurations, either closed, which behaves as a wire in a circuit, or open, which will behave as an open circuit, so no current can flow through.

Figure 4: An open switch

3 Circuit Topologies

Here, we will explore two circuit topologies, their computational complexity in the problem they solve, and their costs.

- Sorting numbers
- Shortest path in a graph

3.1 Sorting

We define the sorting problem to be
Given a list of n numbers,
The fastest algorithms for sorting numbers are

- **QuickSort:** $O(n \log n)$
- **Radix Sort:** $O(b(n + k))$

3.1.1 Circuit

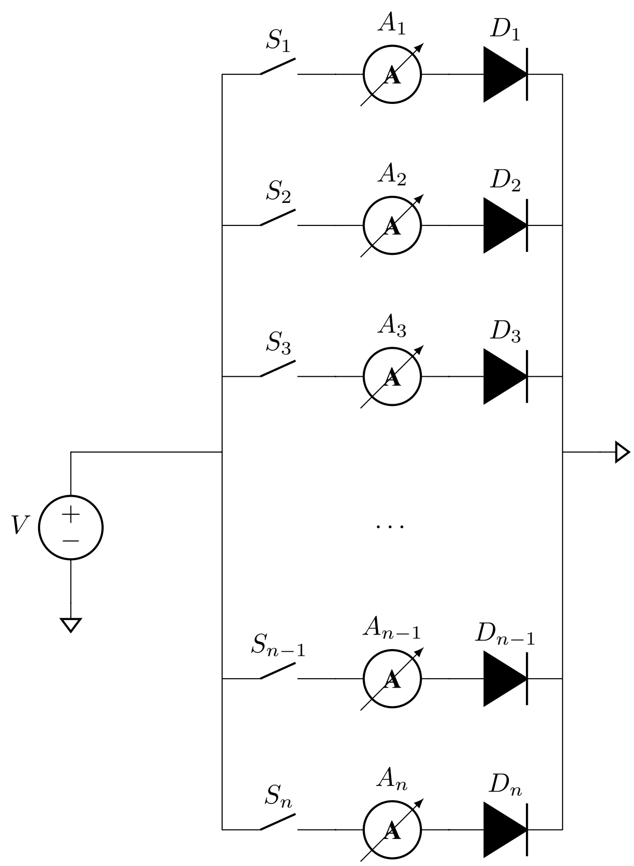

Figure 5: Circuit for sorting numbers

3.1.2 Algorithm

```
SORT:
  // Build circuit
  Get a voltage source V, with nodes A at the positive, B at the negative
  for i = 1 : n:
    Build a series circuit of a switch S_i, ammeter A_i, and diode D_i
    and put it between A, B

  // Sort
  queue Q = []
  for n iterations:
    set V = V_max

    for i such that I(A_i) > 0:
      turn off S_i
      Q.push(V_i)

  return Q
```

We will now prove this algorithm is correct and runs in $O(n)$ time.

First, we will prove the following lemma,

Lemma 1. If there are diodes $D_1, D_2, \ldots, D_n$ in parallel, and a voltage of $V > \max(V_{D_1}, V_{D_2}, \ldots, V_{D_n})$ is applied across these diodes, the voltage across the diodes is $V_{D_i} = \min(V_{D_1}, V_{D_2}, \ldots, V_{D_n})$, and only D_i is on.

Proof. We will prove this statement by contradiction.

Suppose some other D_k where $k \neq i$ is on, then the voltage across the diodes is V_{D_k}. But since we know that $V_{D_k} \geq V_{D_i}$, it must be the case that D_i is also on. However, if D_i is on, then the voltage across the diodes is $V_{D_i} < V_{D_k}$, and therefore D_k cannot be on. We have reached a contradiction, and therefore only V_{D_i} can be one. □

Now, the proof is simple,

Theorem 1. *Algorithm 3.1.2 correctly sorts diodes $D_1, D_2, \ldots, D_n$ from smallest to greatest threshold voltage V_{D_i}.*

Proof. Once we have built our circuit, for each of n iterations, we are only detecting the D_i with the smallest V_{D_i} that turns on, and by turning the switch S_i off, in the next iteration, the smallest D_k of the remaining diodes will be selected next.

Therefore, since we originally pick the smallest D_i, then the second smallest D_k, and so on, we will return the diodes in increasing order. □

3.1.3 Complexity

Theorem 2. *Algorithm 3.1.2 runs in $O(n)$ time.*

Proof. Building the circuit takes 1 operation for putting the voltage source, and $3n$ operations for attaching the switch, ammeter, and diode in parallel.

Then, to run the algorithm, we are just turning off S_i one by one, n times, and each time adding V_{D_i} to our queue, which is a total of 2 operations.

Therefore, the total number of operations is

$$1 + 3n + 2n = 5n + 1 \in O(n). \tag{2}$$

□

Notice that although the complexity is $O(n)$, which matches radix sort, the constant $kn \in O(n)$ is for a $k \approx 2$, which outperforms radix sort.

3.2 Shortest Path

A natural extension to the sorting circuit is to create graphs out of diodes, and then look for shortest paths.

3.2.1 Circuit

3.2.2 Algorithm

First, we need to prove the following

Lemma 2. For a chain of diodes $D_1, D_2, \ldots, D_n$, the turn on voltage for the chain is

$$V_D = \sum_{i=1}^{n} V_{D_i} \tag{3}$$

Proof. □

4 Real-Life Implementations of Circuits

To demonstrate that these ideas can work in practice, I built these circuits in real life and tested them out.

5 Discussion

How do we get diodes of arbitrary threshold voltage? This is unlikely, and can vastly limit the scopes of the problems we'd like to solve.

Physical limitations with electrons traveling through a wire.

What other problems can we solve with this idea?

References