Ternary Impartiality-Based Computational System: Part I A Mathematical Treatise on Three-valued Logical Elements

Madhav Khirwar
Panchkula, India
madhavkhirwar49@gmail.com

***Abstract*—This whitepaper is a description of the mathematical side of making a functional and miniaturize-able circuit that can impartially sort three different and discrete input values into a "truth table", thus effectively acting as a ternary logic gate. It deduces an algorithm that can quickly and effectively do so. The algorithm itself is the fastest possible yet as known to the author, as it has a fixed input domain and size: three variables x, y and z, each one of 0, 1, 2, and a one-step function in all three input variables.**

This paper is part of a larger effort by the author, called Project TRIBUS, which involves constructing a fully-functioning, miniaturize-able ternary logic gate that can be scaled up for use in day-to-day computation.

Keywords - ternary;impartial; number-base; logic-gates

I. INTRODUCTION

A. Overview

For the past many decades, computational technology has been evolving at an ever-accelerating pace. Every year, engineers are able to develop circuits with smaller and smaller process distances, and with exponentially greater computing power. Moore's Law has been quite representative of this growth pattern for a long time, but this decade is seeing a new problem arising: circuit processes are becoming too close to each other; and this is causing an over ow of currents-thus limiting the size to which circuits can be reduced and still be user-end reliable.

Project TRIBUS is an effort to counter-act this 'ceiling' that conventional computers are approaching in their processing power by use of mathematics and simple but unusual engineering. (The acronym is derived from the title of this whitepaper, "TeRnary Impartiality-Based CompUtational System"; and serendipitously, '*tribus*' is Latin for three.)

B. Rationale for Research

Moreover, circuits based in binary logic are, due to ever-increasing complexity, becoming more and more expensive. Many innovative solutions have been proposed for this, the most popular being 'quantum computing'. However, as these quantum computers rely on 'exotic physics', they tend to be hit-and-miss in the real world, not to mention their commercial unavailability, their immense building and maintenance expenditure, and the engineering problems that they harbor.

Thus, the author has undertaken Project TRIBUS, as it will be referred to in the following sections for succinctness, to come up with a solution that uses Mathematical theory rather than 'exotic physics', which will not only have a huge impact on production cost, but also on public availability.

Also, mathematically speaking, operating in a higher base will boost processing power by leaps and bounds. Thus, an extremely viable solution is the adoption of unconventional logic-operation methods. This paper explores the possibility of using ternary logic-based circuits in order to boost the computational power of computer-system architecture using non-conventional circuit design logic.

II. MATHEMATICAL BASES: BINARY AND TERNARY

A number base is simply a way of representing, using different numerals, a numerical value.

A. Making a number base

One of the simplest things to achieve in Mathematical theory is to formulate a number base- the theory that lies behind it has only to do with the basics of number theory. The consecutive number places from the right to the left are multiplied with the base, raised to the power of the number place itself.

Theories suggest that the only reasons we humans choose to operate in the number base of 10, is because we have ten fingers. Speaking from a strictly mathematical viewpoint, there is absolutely nothing significant about the number ten. One of the most brilliantly intuitive, yet easily comprehensible explanations about making a number base can be found in Jeff Duntemann's [2] chapter on number bases.

Thus, the author believes that there is no need to expound upon this, since the reader is expected to have a basic understanding of it.

B. Boolean Algebra

Binary Logic is based in Boolean Algebra, which deals in dual-natured quantities only, which is limited to perfect truisms or fallacies. One of the greatest mistakes a researcher can make in terms of thought is to try and extrapolate the very same mathematical structure into Ternary Logic.

The reader is, for the purpose of comprehending this paper, advised to shed all notions of Boolean Logic; for Ternary calls for a completely new means of mathematical manipulation. TRIBUS deals with revolutionizing both the mathematical structure that governs computation, as well as the physical construction of gates that can perform these operations.

C. Ternary Logic

The first step we must undertake is recognizing that the number base 3 gives us the options (*0; 1; 2)*. Just as Binary Logic calls its values bits, for the purpose of this whitepaper, we shall hereafter refer to Ternary values as **trits**.

Now we may do a few ternary-decimal conversions, so as to familiarize our-selves with it. Let us take the number 27_{10}. This implies the sum of 7×10^0 and 2×10^1.

Let us now convert this to ternary. Since $3^3 = 27$, we have the quantity twenty-seven as: 1000_3.

To check, we shall take each trit, beginning from the right side, and multiply it with 3^n, beginning with n = 0 until the *n*th trit. We shall then add all of them together to obtain our result in decimal.

$$0\times3^0+0\times3^1+0\times3^2+1\times3^3 = 27$$

III. LOGIC

Now, since we are, for TRIBUS, inventing a new mathematical base, we must see how we can maximize the computational power of our circuit. Arpasi [1] has been able to make theoretical truth tables for ternary elements, but he has used Boolean Logic in their manipulation.

First off, we shall design a theoretical logic gate; a physical one shall be expounded upon in further sections. Our theoretical gate will have three inputs, and one output. Since we are not using Boolean operations, we will have to come up with both a mathematical method of logic, as well as a type of gate that can support this mathematical structure.

A. The Mathematical Structure

Any novel approach to the issue of eliminating any binary operations in order to derive a 'truly ternary' will have to discard Profeanu's [3] 'three-step' conditional operations. Instead, it becomes imperative to come up with an algorithm that can not only sort trits impartially, but do so fast, and in as little number of steps as possible.

In terms of time-estimates, the new algorithm should also be able to execute in near or exact *O(1)*, because for a real computer, anything higher would visibly slow down processor speed. The author of this whitepaper has derived and expounded upon such an algorithm in the next section.

IV. KHIRWAR'S SORTING ALGORITHM

Let us attempt to create such a function. We must take into account the fact that when we do this, it shall have to be unbiased in its entirety; it must give an equal number of results, all the way from 000_3 to 222_3, or 0_{10} to 26_{10}.

Profeanu [3] has suggested that we use a sort of "scale" to decide which value to pass, which is a novel approach to the issue, but involves three conditional operations, wherein there is a "check" at each input of the gate, and involves setting a default value to each input gate, which can be confusing and slightly cumbersome.

The solution that the author of this whitepaper proposes is that we devise a simple function in three variables for this, one which involves only one algorithmic step and returns a set of impartial values from 000_3 to 222_3, or 0_{10} to 26_{10}.

This involves the use of modular arithmetic:

$$(x + y + z) \equiv \text{ƙ}(x, y, z)(mod\ 3) \qquad (1)$$

Where ƙ(x, y, z) is the Khirwar's Sorting algorithm.

Since this is a single-step operation, it will eliminate the need for any conditional operators- thus in computational terms, tantamount to a single function. The visual flowchart looks like:

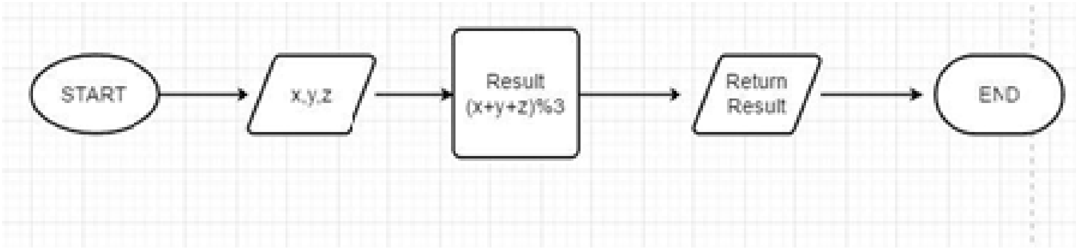

Figure 1. Visual Depiction of Khirwar's Sorting Algorithm

Thus, we have effectively, in a single function, impartially "sorted" the trits into a single truth table.

A. Logical Consistency

To expound upon the algorithm introduced above, it is necessary to look at certain very basic mathematical concepts, the first of which is simple division. Since we are using 3 data types at 3 input gates with 1 output, it is evident from basic permutative theory that the number of possible results is:

$$3^3 = 27$$

Now, it is common and easily provable knowledge that between 0_{10} and 26_{10}, one third of all integers are exactly divisible by 3, one third will leave a remainder of 1, and one third will leave a remainder of 2. Thus, *modulo* 3, every number will evaluate to α, such that $\alpha \in \{0,1,2\}$.

Thus, the impartiality of the system is proven.

B. Algorithmic Design

In terms of Complexity Theory, the computational time-estimate for any algorithm is its order, described by Landau's Symbol *O(1),* since this algorithm is applied to a fixed number of variables $\{x, y, z\}$ that $\in \{0, 1, 2\}$.

This implies, that even in terms of computational time estimates, we have an algorithm that will process in close to zero time.

The fact that its time-estimate is so low means that it can readily be replicated many millions of times over in a short time, which is essentially what a computer works on. It also means that now designing a circuit to do so will involve minimal complexity.

V. CONCLUSION

The first part of TRIBUS has arrived at an algorithm which is currently the fastest the author has come across in sorting trits impartially. The next part of TRIBUS shall be a detailed whitepaper on the structure and implementation of the circuitry being developed by the author to implement with maximum efficiency, the algorithm expounded upon in this paper.

REFERENCES

[1] Jorge Pedraza Arpasi, "A Brief Introduction to Ternary Logic". November 2003.
[2] Jeff Duntemann, "Alien Bases" in Assembly Language Step by Step 1992, pp. 13-38.
[3] Ion Profeanu, "A Ternary Arithmetic and Logic," 2010.