Ternary Mathematics Principles Truth Tables and Logical Operators 3 D Placement of Logical Elements Extensions of Boolean Algebra

Ruslan Pozinkevych[1*]

[1]*Faculty of Information Technologies and Mathematics, The Eastern European National University, 43021, Lutsk, Potapov str.9, Ukraine.*

Author's contribution

The sole author designed, analyzed, interpreted and prepared the manuscript.

Original Research Article

ABSTRACT

The approach presented in this article aims at transition between two systems of counting binary and ternary I propose to use ternary math principle in coding the signal Instead of using duos of numbers 01 I propose to use triplets (1,0-1) and make a transition from binary to ternary so that the binary code is converted to ternary and vice versa That same principle can be used for building microcircuits where logical elements are placed in a 3 D space instead of a layer.

Keywords: Propositional operators; truth table; combinatorics; associaive/distributive principle; truth functional operation.

1. INTRODUCTION

As we know from the course of discrete mathematics in logics and in mathematical logics in particular we build statements using operators such as conjunction (and), disjunction (or), negation (not), and (existential or universal) quantifiers. We formalize statements using the laws of logics or mathematical logics to simplify them and manipulate in accordance Since 19th century especially with the works on algorithms of discrete mathematics and contributors such as

---

Corresponding author: E-mail: galagut@protonmail.com;

George Bull ((1815-1864), F. Freher (1848-1929), Bertran Russel (1872-1970), Peano (1858-1932), Whitehead (1861-1947) this formal approach laid the foundation for the new science named discrete mathematics and in the second half of the 20th century with the appearance of first programming machines and use of binary math the contribution of those mathematicians is invaluable Boole [1], Kennedy & 177-180 [2].

With the development of computers however there stood a need in faster data processing and efficient use of hardware We all experienced development of computer device from primitive electric circuits and microcircuits to the integrated circuits and potentially powerful Nano computers Mendelson [3].

With that idea in mind there appears to be a need in new approach towards data processing and we are trying to present such an approach.

2. METHODOLOGY

Let's take a look at some of the propositional operators and the truths tables of conjunction for example Hilbert [4].

The compound statement in the last row is *A&B* which is only true if both literals *A* and *B* are true.

Table 1. List of compound statement

A	B	A&B
T	T	T
F	F	F
F	T	F
T	F	F

Let's take a look at the disjunction table

Table 2. Disjunction table

A	B	A $\vee$ B
T	T	T
F	F	F
F	T	T
T	F	T

Why do we choose these two operators Basically because all other logical operators can be presented as combination of the two or finally a True/False statement.

The conjunction disjunction and negation statements are the abstract representations of the algorithms of building processing machines

or circuits in integral integrated circuits Shannon [5]. That is why the propositional logic is so important in engineering and computer maths.

As it appears modern data processing requires algorithms in a form different from 1D linear circuits It might stand in need for 3 Dimensional or 3 D placement of the logical elements for which the Truths tables will look different Gould, W. E. (1974).

We propose a new type of relation between the elements of the logical expression. They are built using principles of association and distribution for example.

In this abstract the author just tries to justify the use of triples of literals to make logical statements Further research is necessary for a transition from binary to ternary maths Lukasiewicz [6]. Here we go

Table 3. A distributive principle

A(B+C)	AB+BC
$T(\overline{T}+F)$	1(-1+0)

One should mention here that instead of two operators True and False in our ternary table we use three operands T, $\overline{T}$, F (Truth, Truth negation, False) It will become clear when we substitute those by the corresponding numbers (1,-1,0).

In other words for the literals here we use not binary but the ternary which can be easily converted to the Bullean math as subtraction.

(1) Example:

Let's say we have a string of 01 numbers presented by the sequence 010010+our last element *A(B+C)*

or as we agreed $T(\overline{T}+F)$ *=1(-1+0)=-1*

Let's write it as 010010 *-1* = 010000

And so on we can add subtract multiply binary numbers converting our ternary formulas into the binary What for? So that we could place our logical elements on the two planes Plane of binary numbers *(x,y)=(0,1)* and the plane of *-1's*.

As previously mentioned There are two offered principles used for logical operations in the

ternary system: one is distributive mentioned above and the other one is associative Before we start to talk about the second principle lets remind ourselves of the main maths operations used in discrete maths when dealing with logical expressions They are logical addition and logical multiplication These two are most easily recognizable when joining statements by the symbols $\cup$ and $\cap$ We are trying to do something similar and use two principles Gallier [7].

Back to the associative principle we have three operands *ABC* that can be presented as combinations.

ABC ACB CBA CAB BAC BCA =3!

Now when we substitute them by numbers that is *1,-1* and *0* we will obviously get one result provided of course that we use multiplication as an operation for our elements that is *0.*

$(0 \cdot 1 \cdot -1) = 0$ but let it not mislead you as we can have a chain *ABB*= $1 \cdot (-1) \cdot (-1)$ which equals *1.*

And if we combine our first method that is distributional and our second approach then we can write any number written in binary code as a combination of the first and the second.

010010 *-1* = 010000=010010+ *A(B+C)+ABC*= 010010+ *AB+AC+ABC*

Let's have a look at what the new Truth tables with two operators we shall call them ternary addition and ternary multiplication.

Let's take a look at ternary multiplication first the number of entries is different It will be 9 obviously Choosing an appropriate sign would be necessary For the time being let's mark it as (*)

Table 4. Ternary multiplication

A	B	A*B
T	T	T
F	F	F
$\overline{T}$	$\overline{T}$	T
T	F	F
F	T	F
$\overline{T}$	T	$\overline{T}$
$\overline{T}$	F	F
F	$\overline{T}$	F
T	$\overline{T}$	$\overline{T}$

Ternary addition will be a challenge from the maths standpoint Let's use our distributive law -1 is a product of $\overline{T}$ and T therefore we would present the sum of $\overline{T}$ and $\overline{T}$ as the product of T and $\overline{T}$.

Let's see what we can accomplish by doing so:

$$\overline{T} + \overline{T} \equiv (T^* \overline{T}) + (T^* \overline{T}) \equiv \overline{T} (T+T)$$

Note:

We have to accept mathematical signs of addition + and multiplication as representatives of conjunction & and disjunction $\vee$ in our Tables 1 and 2 So what will be the table of ternary addition?

This is 9 entries table the same as ternary multiplication with the results column different of course.

Coming back to our formula:

(T+T)=T;

Respectively: $\overline{T} (T+T) \equiv \overline{T} *(T\&T) \Rightarrow \overline{T} * T = \overline{T}$

Now it's time to present our ternary addition table using logical operators A ternary addition sign is chosen randomly just to differ it from conventional addition In our table it's being presented as $\oplus$

Table 5. Ternary addition table using logical operators

A	B	A $\oplus$ B
T	T	T
F	F	F
$\overline{T}$	$\overline{T}$	$\overline{T}$
T	F	T
F	T	T
$\overline{T}$	T	F
$\overline{T}$	F	$\overline{T}$
F	$\overline{T}$	$\overline{T}$
T	$\overline{T}$	F

Our next step is an attempt to derive the formula for the arrays of ordered pairs (x, y). As shown before our ternary tables consist of 9 rows

corresponding to permutations with replacement $_nP_r = n^r$ In our case the total number of permutations with replacement n^r equals a decimal number Let's call it d where $d = n^r$ where r=2,3,4…. depending on which system we use (binary, ternary, etc.) From here $n = \sqrt{d}\ (n^2 = d) \rightarrow n = \sqrt{d}$ What does our number n stand for? It stands for the number of arrays we choose from Respectively we can present our decimal number as the number n^r; number of arrays $n = \sqrt[r]{d}$ or which is the same $n = d^{\frac{1}{r}}$ which we can also present as a binomial distribution

$$\sum_{k=0}^{n}\binom{n}{k}d^k \rightarrow \sum_{k=0}^{\frac{1}{r}}\begin{pmatrix}\frac{1}{r}\\k\end{pmatrix}d^k$$

3. RESULTS AND DISCUSSION

Such an approach will give us a distribution of arrays (1;0). For example (1,2,1); (1;3;3;1) etc. To conclude what has just been said we present in our research a transition from one system of counting to another Whitehead & 1962 [8]. When we operate decimal numbers we might switch to binary ternary or any other system of numbers to convert from Such an approach is designed to make processors that implement new principles of transition and processing of the data In their turn these principles can facilitate the development of already existing computers and their architecture This example shows how logic can be updated to the needs of the developing industry and facilitate the development of existing technology.

4. CONCLUSION

The approach presented in this article aims at transition between two systems of counting binary and ternary This is not a secret that with the industry mathematics methods of calculations are developing and with their development there is a need in new ways of transmission storing and processing of the data I deliberately omit mentioning physical approach in my introductory research It's well known that development of new systems of counting needs new technical means and sometimes the gap between the first and the second to close can count decades of research calculations etc. However I might suggest that for the virtual machines and modeling as well as cryptography perhaps and adaptation to Nano computer systems this approach can add some value.

COMPETING INTERESTS

Author has declared that no competing interests exist.

REFERENCES

1. Boole G. Mathematical analysis of logic. MacMillan, Barclay & MacMillan Cambridge; 1847. Republished by Open Court, La Salle; 1952.
2. Kennedy, Hubert C. & Peano, Giuseppe. Selected Works of Giuseppe Peano. Journal of Symbolic Logic. 1980;45(1): 177-180.
3. Boollean Algebra and switching circuits Elliot Mendelson Shaum's outline series.
4. David Hilbert. The Foundations of mathematics; 1927.
5. Shannon CE. A Symbolic Analysis of Relay and Switching Circuits; 1938.
6. Lukasiewicz JAN. Aristotle's Syllogistic From the Standpoint of Modern Formal Logic. Garland; 1951.
7. Jean Gallier Discrete Mathematics, Second edition in progress, February 27; 2019.
8. Whitehead, Alfred North & Russell, Bertrand. Principia Mathematica, to *56. Cambridge University Press; 1962.

Peer-review history:
The peer review history for this paper can be accessed here:
http://www.sdiarticle4.com/review-history/63247