A Ternary Arithmetic and Logic

Ion Profeanu

***Abstract*—This paper is only a chapter, not very detailed, of a larger work aimed at developing a theoretical tool to investigate first electromagnetic fields but not only that, (an imaginative researcher might use the same tool in very unusual areas of research) with the stated aim of providing a new perspective in understanding older or recent research in "free energy". I read somewhere that devices which generate "free energy" works by laws and principles that can not be explained within the framework of classical physics, and that is why they are kept far away from public eye. So in the absence of an adequate theory to explain these phenomena, these devices can not reach the design tables of some plants in order to produce them in greater number. The tool I am talking about is a logical-mathematical one, and has many facets which I will gradually reveal. For the beginning I propose a ternary arithmetic and logic and I will bring arguments for their use.**

***Index Terms*—free energy, ternary arithmetic, ternary logic.**

I. Introduction

It is well known that the numeral system used in most if not all modern computer systems is the binary numerical system, as again it is known that binary logic, also known as Boolean algebra, also brings his significant contribution to the building of the same computer systems which I will name them more briefly as computers. In other words we distinguish here, on the one hand between a binary arithmetic in which we do basic calculations such as addition, subtraction, multiplication, division in a manner as possible like we do arithmetic calculations with numbers written in the decimal numerical system and on the other hand we call into question this time an algebra, which has also two digits that are the truth values of sentences and I mean by this the propositional calculus of the above mentioned Boole algebra.

Why does the binary numeral system and binary logic have so much significance in the construction of computers? The answer is at hand if we think about how easy it is to build electronic circuits with two states and I mention here the known bistable circuit. In the same manner we can easily distinguish between a closed electrical circuit and an open one, and going further, between a forward biased LED and a switched off one, as again, one is a magnetized portion and a different thing is a non-magnetized portion of the same magnetic tape. In the spiritual plane we discover the dualist philosophies, among which the two religious concepts of Yin and Yang with roots in Chinese philosophy and metaphysics, is prominently situated, and also Manichaeism which supports radical ontological dualism between the two eternal principles, Good and Evil, which oppose each other in the course of history, in an endless confrontation. A key element of Manichean doctrine is the non-omnipotence of the power of God, denying the infinite perfection of divinity that has they say a dual nature, consisting of two equal but opposite sides (Good-Bad). I confess that this kind of dualism, which I think is harmful, made me to seek another numeral system and another logic by means of which I will praise and bring honor owed to God's name; and because God is One Being in three personal dimensions (the Father, the Son and the Holy Spirit) I stopped as expected at the ternary numeral system and ternary logic.

II. Arithmetic in the Ternary Numeral System

A. Ternary digits

The two figures of the binary numeral system are 0 and 1, each bearing the name of **B**inary dig**IT,** hence the name of BIT with its plural **bits**. The figures of the ternary numeral system are 0, 1 and 2. As each of these figures would be known in English as a **T**ernary dig**IT** we get the word **TIT** (with the plural **TITs**) that I propose to use from now on, when we refer to the figures of the ternary numeral system.

B. Counting and converting

Let's count in the ternary numeral system: 0, 1, 2, 10, 11, 12, 20, 21, 22, 100, 101, 102, etc. Let's convert into the base-10 number system, the number 2102_3 written in base-3 number system: $2102_3 = 2 \cdot 3^3 + 1 \cdot 3^2 + 0 \cdot 3^1 + 2 \cdot 3^0 = 54 + 9 + 0 + 2 = 65_{10}$. The resources of the Word Editor does not allow me to convert in base-3 a number written in base-10, but the procedure is similar to that used in the passage in base-2 of the numbers written in base-10, namely we do the integer division of the number with three and then successively of the obtained quotients with the same number three until the moment the quotient is zero, then we write the obtained remainders one after another from the last and reaching up to the first.

C. Encoding

How much distinct information can we encode using n TITs? The answer is 3^n. As I said, n = 3 has a particular significance for us. There are $3^3 = 27$ distinct combinations of the three TITs and it remains 26 if we exclude the combination 000. By fortune (really?) the number of characters of the English alphabet is 26 and we are very ready to associate the 26 different combinations of numbers 0, 1 and 2 with the letters of the English alphabet, adding combination 000 for the character @ as in Table I:

Manuscript received March 2, 2010.
Ion Profeanu is with "Grupul Şcolar Gheorghe Lazăr", Baia Mare, str. Dr. Victor Babeş, nr. 64 Romania, phone: 004-0729-030161; e-mail: ion.profeanu@yahoo.com.

Table I. Encoding of the English alphabet letters

000 @	012 E	101 J	120 O	202 T	221 Y
001 A	020 F	102 K	121 P	210 U	222 Z
002 B	021 G	110 L	122 Q	211 V	
010 C	022 H	111 M	200 R	212 W	
011 D	100 I	112 N	201 S	220 X	

D. A three-dimensional array

Starting from a two-dimensional array with three rows and three columns (Fig. 1) I will build a so-called three-dimensional array which will have a total of 27 items:

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{pmatrix}$$

Fig. 1. A two-dimensional array with three rows and three columns

Each of the 9 elements of the array can now be considered a vector of three items. For example a_{00} is the vector which has the elements a_{000}, a_{001} and a_{002}. The first element that is a_{000} is a small cube sitting on the plane of the paper, the second element that is a_{001} is a small cube placed over the first, coming towards us and a_{002} is the small cube situated closest to our eyes. This way we obtained a large cube consisting of 27 small cubes, each of which can be identified by a combination of three TITs written as subscript near letter *a* or simply, as I did above the correspondence with the English alphabet letters, by a character. For example the small cube located in the heart of the large cube can be identified by a_{111} or by the letter M.

The American Standard Code for Information Interchange (ASCII code) is a binary code with initial bytes length (byte meaning a series of binary digits used for encoding of an alphanumeric character) of 7 bits with the help of which 128 characters could be coded. An extended version of the same code is in use that using 8 bits encodes 256 alphanumeric characters. It is easy to see that if we use a **tyte** (ternary equivalent of byte) of only 5 TITs, all possible combinations of it encode $3^5 = 243$ characters which is very close to 256, wanting to say by this that 5-TITS tyte is large enough and is also saving 3 lines of data.

E. Physical implementation

It is the time to present the way I recommend the physical implementation of the three ternary digits, so we should first recall how the binary digits are implemented. In positive logic, bit "1" means a positive potential versus the ground (e.g. + 5 V if using TTL gates) and bit "0" is the potential of the ground (reference point) which means 0 V. For the three ternary digits I recommend the following assignment: "0" = 0 V, "1" = + 5 V and "2"= – 5 V. I can also accept, based on technological considerations or of another nature, other distribution of the three potentials to the three TITs, but basically when I imagine the future digital circuits to implement ternary arithmetic I think of circuits that use voltages: – 5 V, 0 V and + 5 V or the ground potential and any other two appropriate voltages equal in modulus but of different signs.

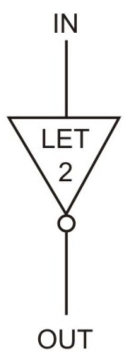
III. Ternary Logic

A. Designing logic gates

Arithmetic using the base-3 number system was for me a relatively easy matter, having nothing to do but to move base-2 arithmetic to the arithmetic that uses three digits instead of two. The real problem arose when I had to design the logic gates which would be a difficult undertaking if we give up the logic of Aristotle. I present in short the stages of the development and I propose the solution I have reached. I watched the logic gates and "forgetting" Aristotle's logic that they implement I have managed to find "the truth". I saw if you like, the "design" and found out that, for example, the AND gate is a gate with two inputs and one output. I wondered why there were two entries and found the answer to be because it is a binary logic. I then asked myself why there was only a single output and realized that the gate combines the two inputs. The next step was to explore the inverter (NOT gate) that has only one input and one output and I had a revelation. All I had to do was to build gates with one, two or three inputs for which the output (possibly outputs) meet a certain "logic", but other than that of Aristotle. And thus I designed gates with one input and one output, a gate with two inputs and one output, and gates with three inputs and one output.

B. Gates with one input and one output

I describe first the so called "logical" gates that have one input and one output. The equivalent gate from binary logic with one input and one output is the NOT gate or the inverter gate. If the input is at the logical level 0, the output will be at the logical level 1 and if the input is at the logical level 1, the output will be at the logical level 0. And now let us "play" a little with the three TITs of the ternary logic. I inform you in advance that in ternary logic we can design at least five gates with an input and an output. Let us make two groups with ternary numbers, a group that has two digits and a group that has a single figure, the easier being to group TIT 0 with TIT 1 that leaves TIT 2 single in group. And here's how this first gate works: it switches between the TITs from the first group and leaves unchanged the TIT of the group with a single figure that is TIT 2. And because the gate allows (lets) figure 2 to pass through the gate unchanged, switching between them the other two, I chose as symbol of this gate, the sign used to represent the inverter gate in which I wrote "LET 2". You can watch the graphical representation and the truth table in Fig. 2.



IN	LET 2
0	1
1	0
2	2

(a) (b)

Fig. 2. (a) The gate LET 2 (b)The truth table of the gate LET 2

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Following the same "logic" we get another two gates with one input and one output, namely the gate "LET 0" and gate "LET 1" you see represented in Fig. 3 (a), (c). The gate "LET 0" allows TIT 0 to pass it unchanged and switches the TITs 1 and 2 between them, whereas the gate "LET 1" allows TIT 1 to go through it unchanged and switches the TITs 0 and 2 between them (see the truth tables in Fig. 3 (b), (d)).

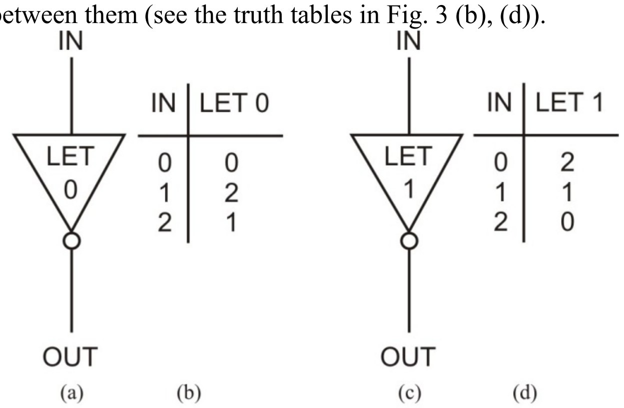


Fig. 3. (a) The gate LET 0 (b) The truth table of the gate LET 0
(c) The gate LET 1 (d) The truth table of the gate LET 1

Let us now do a cyclical transformation of the TITs between them, namely to turn 0 to 1, 1 to 2 and 2 to 0 as again let's turn 0 to 2, 2 to 1 and 1 to 0. I will call these gates "ROT" that stands for the word "ROTATION" and therefore there are two gates "ROT 1" and "ROT 2" as shown in Fig. 4 (a), (c) (see the truth tables in Fig. 4 (b), (d)):

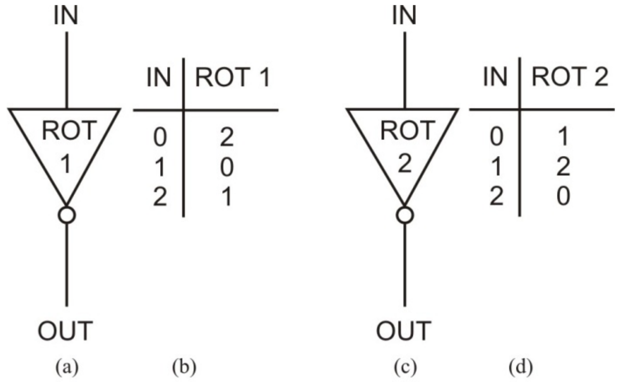


Fig. 4. (a) The gate ROT 1 (b) The truth table of the gate ROT 1
(c) The gate ROT 2 (d) The truth table of the gate ROT 2

A. Gates with two inputs and one output

To describe the gate, this has two inputs and one output, which I called the gate "when two quarrel the third wins". The "truth" table (as for the previous gates the word "truth" is suitable for classical gates and less here, but I have not yet found another appropriate term) warrants the name and I invite you to see how in Fig. 5 (a) and (b). When "0" quarrels with "1", "2" wins, when "0" quarrels with "2", "1" wins etc. When "0" quarrels with "0", "0" obviously wins and the same happens with the quarrel between "1" and "1" and the quarrel between "2" and "2".

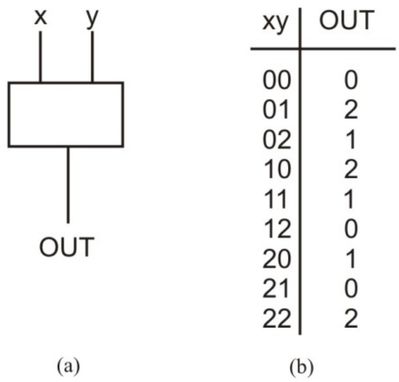


Fig. 5. (a) The gate "when two quarrel the third wins"
(b) The truth table of this gate

If we read the output in groups of three digits i.e. 021, 210 and 102 we obtain the "word" GUK according to the encoding from the beginning of this paper. This suggests us that we can design as many gates with two inputs and an exit as many combinations of three English letters there exist.

B. Gates with three inputs and one output

I have already presented the gates with one input and one output and the gate with two inputs and one output so I will go to the presentation of the gates with three inputs and one output. For the beginning let’s think of the graphic representations of accounts used in accounting, called bilateral forms, in tables with two sides like the letter T (balance sheet) or better think to the scales (fig. 6).

Fig. 6. The scales

We first associate the TITs 0, 1 and 2 to the left pan, central axle and right pan of the scales and later we will see that we can also do other combinations, but now we take the simplest case. For simplicity we also say that the TIT 1 is "heavier" than the TIT 0 and the TIT 2 is "heavier" than the tit 1. The corresponding gate is finally doing the work of the scales due to the fact of which I will appoint it a "LIBRA". This will be of type 012, 012 where the first group of TITs indicated the association between them and the three components of the scales. The way the TITs in the second group are written indicates the increasing order of their weights (fig. 7).

Fig. 7. The gate LIBRA 012, 012

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Now suppose that three TITs x, y and z reach the input of the gate. The TIT x reaches the left pan which corresponds to number 0, the TIT z reaches the right pan that corresponds to digit 2 and the TIT y could correspond to the central axle, i.e. digit 1. The gate assesses the weights and found for instance that the TIT z is "heavier" than the TIT x and so the pan which lowers corresponds to tit 2 means that the output of the gate will be set to 2. If the TIT x is "heavier" the pan that goes down corresponds to digit 0 and the output is set to 0. If x and z have the same weight then the pans are balanced and the central axle becomes significant to which I associated the number 1, so the output will be set to 1. In Table II, I present the truth table for the gate "LIBRA 012, 012".

Table II. The truth table of the gate LIBRA 012, 012

xyz	OUT	xyz	OUT	xyz	OUT
000	1	100	0	200	0
001	2	101	1	201	0
002	2	102	2	202	1
010	1	110	0	210	0
011	2	111	1	211	0
012	2	112	2	212	1
020	1	120	0	220	0
021	2	121	1	221	0
022	2	122	2	222	1

I have established a two-way relationship between the 27 different combinations of ternary digits 0, 1 and 2 and the 26 English alphabet letters to which I have added the character @ to get the same number 27. The combination 000 corresponds to the character @, the combination 001 corresponds to character A, the combination 002 corresponds to character B, etc. This way the combination 012 corresponds to the character E and thus the gate described above could be called "LIBRA EE". For the sake of exercise and to reinforce better that said so far, I will deal below with the gate "LIBRA 201, 210" or more briefly "LIBRA SU". As I said, first group of tits are assigned to the parts of the balance: TIT 2 corresponds to the left pan, TIT 0 corresponds to the central axle and TIT 1 corresponds to the right pan. The second group of tits says that 1 is "heavier" than 2 and that 0 is "heavier" than 1. These being said, track in Table III the truth table of this gate.

Table III. The truth table of the gate LIBRA SU

xyz	OUT	xyz	OUT	xyz	OUT
000	0	100	1	200	1
001	2	101	0	201	1
002	2	102	2	202	0
010	0	110	1	210	1
011	2	111	0	211	1
012	2	112	2	212	0
020	0	120	1	220	1
021	2	121	0	221	1
022	2	122	2	222	0

I have also tried to design gates with three inputs and two outputs but ultimately they are reduced to gates with one input and one output and gates with two inputs and one output, and therefore I do not dwell on them here.

IV. Conclusion

Finally I point out that the described arithmetic and logic can have applications in very diverse areas. But remaining at the IT domain I also add that using a number of only 5 TITs we can encode 243 alphanumeric characters, a number that is close to the 256 characters that are encoded using 8 bits, that means the ternary arithmetic and the ternary logic extend very much the ability to exploit the hardware resources, by simply introducing a negative electric potential.

References

[1] Palma, Raoul de, "Algebra binara a lui Boole si aplicatiile ei in informatica," Ed. Tehnica Bucuresti, 1976.
[2] Arpasi, Jorge Pedraza, "A Brief Introduction to Ternary Logic," 7th November 2003, http://aymara.org/ternary/ternary.pdf
[3] "Ternary logic, From Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Ternary_logic
[4] "Ternary computer, From Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Ternary_computer

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)