

Title: Ternary Digital System: Concepts and Applications

Authors: A P Dhande, V T Ingole, V R Ghiye

Published by SM Online Publishers LLC

Copyright © 2014 SM Online Publishers LLC

ISBN: 978-0-9962745-0-0

All book chapters are Open Access distributed under the Creative Commons Attribution 3.0 license, which allows users to download, copy and build upon published articles even for commercial purposes, as long as the author and publisher are properly credited, which ensures maximum dissemination and a wider impact of the publication. Upon publication of the eBook, authors have the right to republish it, in whole or part, in any publication of which they are the author, and to make other personal use of the work, identifying the original source.

Statements and opinions expressed in the book are these of the individual contributors and not necessarily those of the editors or publisher. No responsibility is accepted for the accuracy of information contained in the published chapters. The publisher assumes no responsibility for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained in the book.

First published October, 2014

Online Edition available at www.smgebooks.com

For reprints, please contact us at ebooks@esciencemedicine.com

Fundamental Concept of Ternary Logic

INTRODUCTION

The fundament of today's digital technology age is binary logic. The time when Shannon expressed the behavior of electrical switches in Boolean algebra, he overlay the ramp to an industrial development which is recognized as beginning of one of the most revolutionary economic changes ever.

Binary logic technology has come across the dramatic changes and advances. Earlier from electro-mechanical to electronic switches by using electronic tubes (1919) like triode, pentodes, then from tubes to transistors (1948) and from transistors to LSI (1958) and VLSI (1970) circuits. Although efficient and powerful, binary logic is not the *most* efficient & *powerful* switching logic. Non-binary logic or Multiple Valued Logic ($\text{radix} > 2$) has been around for quite a while and is known as Multi-Valued Logic or Many Valued Logic. In this book it will be referred to as MVL hereafter.

The subject of MVL is also known as Multi-Valued, Multiple-Valued or Many-Valued logic. In case of 3-Valued logic ($\text{radix} = 3$) the term 'Ternary' logic is used & term 'Quaternary' logic ($\text{radix} = 4$) for 4-Valued logic and so on up to 'n' values. Multi-Value logic is regarded as a switch with more than two states. Such as a 3-value switch with logic states '0', '1' and '2', 4-value switch with logic states '0', '1', '2' and '3' and so on up to 'n' values.

MVL has been the topic of most interest of many researchers over the last 50 years. From 1971 there has been an annual symposium devoted exclusively to the object. Moreover, a large number of technical papers have published together with numerous survey articles. Much of the ancient work is purely theoretical nature concerned with the completeness of the function with sets of operator, function minimization and similar problems from the switching theory and logic design. Work on hardware implementation of multiple value devices has been more recent. The use of Multi-Valued logic ranges from various applications to VLSI technology and design techniques.

There are three directions for the work in MVL. Due to pressure to reduce interconnection complexity and reduce chip area on VLSI, it is giving motivation for the investigation of many different hardware implementations of MVL systems. The largest commercial use of Multiple-Valued logic is in the area of MVL memories. The MVL can be used to overcome existing difficulties

in the analysis of problems in binary digital systems, such as the design of fault simulators. Finally there is still ongoing work in the general area of switching theory to yield the best methodologies for the implementation of multi-valued systems.

There are two modes of operation in MVL. Those are voltage mode operation and current mode operation Figure 1.1 shows operation modes of MVL. In voltage mode operation, logic state is specified in terms of distinct voltage levels i.e. $V_1, V_2, \dots, V_n$ and in current mode logic, state is multiple of lowest logic current level state i.e. Logic state for current mode is $xI_1, xI_2, \dots, xI_n$, where x is reference current.

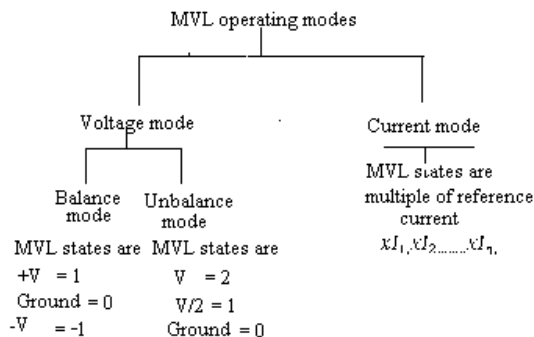


Figure 1.1: MVL operating modes.

Where, 1, -1, 0 and 2 ...n are the logic levels for operation in voltage mode and $xI_1, xI_2, \dots, xI_n$ for current mode. The figure 1.2 is a representation for Ternary logic.

Modern technical work have shown advantages of using Multi-Valued logic where the natural question is whether there exists a practical radix other than 2 that would produce circuits with greater saving in components, without loss of speed? Answer to this is 'Yes' and it is Multi Valued Logic.

THE TERNARY LOGIC

In existing binary digital system, the output of the system is decided by considering two input conditions i.e. either ON (Favorable or true logical level 1) or OFF (unfavorable or false logic at logic level 0) leaving behind the third conditions i.e. when both the input conditions are same, here decision is consider as don't care or it is discarded by the system. Such situation generally occurs in sequential circuit design. Consider a digital system where both the inputs are same i.e. either 00 or 11 as shown in figure 1.2. Here in binary system output will be uncertain or will be same as that of previous state of the system but in practice, system must give the output that will satisfy both the input conditions mentioned above. It is shown in figure 1.2 here the system gives the output which is balanced and this state is regarded as third state i.e. can't say or can't make any decision. So to make third decision the radix of the system must be greater than 2. Here the third logic level is introduced whose system radix is greater than 2.

Alexander [1964] showed that natural base ($e \approx 2.71828$) is the most efficient radix for implementation of switching circuits. It seems that most efficient radix for the implementation of digital system is 3 than 2. Ternary logic system, meaning that it has 3 valued switching. Ternary system has several important merits over binary. It can be listed as reductions in the interconnections require to implement logic functions, thereby reducing chip area, more information can be transmitted over a given set of lines, lesser memory requirement for a given data length. Besides this serial & some serial-parallel operations can be carried out at higher speed [1-3]. Its advantages have been confirmed in the application like memories, communications and digital signal processing etc [4].

Russian first ternary computer “SETUN” and “SETUN 70” was developed at Moscow state university in 1960. It was found that ternary computer is very favorable for seizing of application simplicity of programming in codes, other than permitted to design a few interpreter. Few of the example of implementation of ternary logic systems are three value counter which greatly simplifies the counter circuitry based on two value logic, Three valued memory based on multi valued logic can considerably reduce the memory size required to store than it requires by using two value logic, the implementation of cyclic convolution where significant advantage can be gained by using ternary digital hardware namely an increased maximum sequence length and can be achieved without increasing the complexity of digital hardware. The current mode CMOS circuits have application in digital signal processing and computing.

The three value logic offers particular advantages in digital signal processing applications (Convolution, FFT) etc. For example, an increased maximum sequence of length can be achieved by implementing ternary logic system in DSP.

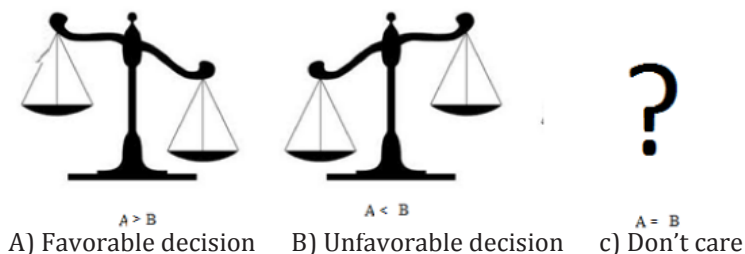


Figure 1.2: Binary decision making.

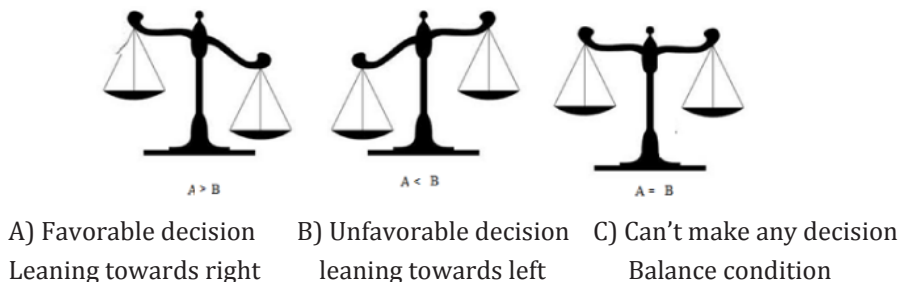


Figure 1.3: Natural decision making.

There are several advantages using ternary logic digital system which can be summarized as follows.

Number Representation

For representing decimal number 16 in binary, 5 bits are required (10000) whereas for representing same number in ternary require 3 bits (121). Similarly, signed number representation in binary ranges from -2^{n-1} to $2^{n-1} - 1$ for negative & positive numbers. In ternary same range spans from -3^{n-1} to $3^{n-1} - 1$. Thus number representation in ternary facilitates to develop algorithm for arithmetic operations for high speed and/or area efficient computation.

Processors

If the case of binary 8-bit microprocessor is considered, it has $2^8 = 256$ instructions but same ternary processor will have $3^8 = 6561$ instructions. It means that increasing radix of the system can increase processing capacity of the processor or to have 256 instructions in ternary processor, only 5 bits are required. Thus it reduces design complexity, number of interconnections & power consumption of the system. No ternary processor has been reported till date.

Communication

In communication based on ternary digital system, a minimum of 3 digits is required to code ten decimal digits. The possible states of three variables in 3-valued system results in $27! / (27-10)! \approx 3.1 \times 10^{12}$ ternary coded decimal. Not all of these codes are fundamentally different. Some exhibits useful properties for coded arithmetic, encoding techniques, error detection etc.

Converters

In case of digital converters, for e.g. 8 bit binary ADC has $2^8 = 256$ values for representing analog signal whereas same ternary 8 bit ADC has $3^8 = 6561$ discrete values for same analog signal. It leads to accurate & efficient processing of digital signal.

Memory

As mentioned above, number of bits required to code a decimal number in binary is more than ternary, ultimately memory required to store the coded decimal in ternary is lesser than binary which has advantages like cost/bit ratio reduction, reduced access time & increased storage capacity.

TERNARY DIGITAL SIGNALS

As mentioned in the introductory part, there are two modes of operation in ternary system. Those are voltage mode operation and current mode operation. In voltage mode operation, logic state is specified in terms of distinct voltage levels i.e. $v_1, v_2, \dots, v_n$. And in current mode logic state is multiple of lowest logic current level state i.e. Logic state for current mode is $xI_1, xI_2, \dots, xI_n$, where x is reference current.

Voltage mode operation is further classified in to balance and unbalance voltage mode operation depending on supply voltage to logic circuit. In balance logic for voltage mode operation, the signal level takes the value as $-v$ ($-v$ i.e. lowest logic $0 = v_L$), ground ($0v$ i.e. intermediate logic $1 = v_I$) and $+v$ ($+v$ i.e. highest logic $2 = V_H$) whereas in unbalance condition the voltage levels are ground ($0v$ logic $0 = v_L$), $v/2$ (intermediate logic $1 = v_I$) and v (highest logic $2 = V_H$). In figure 1.4 (a) shows unbalance signal and figure 1.4 (b) shows balance signal. In each of two signals we observe that the voltage corresponding to given level; is not fixed, rather voltage in a limited range are designated as level. As long as a voltage belong to a level will be taken as level and exact value of the voltage is immaterial.

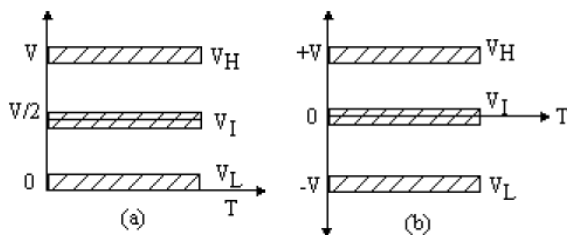


Figure 1.4: logic levels in Ternary system.

(a) When supply voltage is 0 to $+v$ volt: Unbalance system (b) When supply voltage is $-v$ to $+v$ volt: Balance system.

In current mode of operation for balance system current will be either sink, source or will be idle or very small current flows through circuit and for unbalance the current level is multiple of lowest reference current level. The waveforms are same as in figure 1.4 except voltage term will be replace by current term.

TERNARY NUMBER SYSTEM

A digital system represents information with discrete symbols rather than with continuously carrying quantity as in an analog system. Digital binary systems use just two symbols, 0 and 1.

To represent all information leavening aside for the moment the problem of circuit realization, one may ask whether the binary number representation is an optimum choice?. The real world is not binary. It is more intuitive to reason about a system, especially at higher levels of abstractions, in terms of variables with symbolic values. In many practical engineering situations, a device can be not only in "off" or "on" state, but also in "idle" state. When arithmetic operations are involved, computing in a decimal system would match best our experience.

For representing decimal number 16 in binary, 5 bits are required (10000) whereas for representing same number in ternary require 3 bits (121). Similarly, signed number representation in binary ranges from -2^{n-1} to $2^{n-1} - 1$ for negative and positive numbers. In ternary same range spans from -3^{n-1} to $3^{n-1} - 1$. Thus number representation in ternary facilitates to develop algorithm for arithmetic operations for high speed and/or area efficient computation.

Number Representation

There are two major conventions for labeling values in a multiple-valued logic system over a set of m values. The most common is $0, 1, 2, \dots, m-2, m-1$, extending binary notation in one direction only. It is called unbalanced (or unsigned, or positive). The second one requires an odd $m = 2r + 1$. It extends binary notation in both directions as $-r, 1-r, \dots, -1, 0, 1, \dots, r-1, r$. It is called balanced (or Signed).

A string of digits $(a_{n-1} \dots a_0)$ over a set of m values represents the number $a_{n-1}m^{n-1} + a_{n-2}m^{n-2} + \dots + a_0$.

For example, in the binary case of $m = 2$, $a_i \in \{0, 1\}$, in the ternary case $m = 3$, $a_i \in \{0, 1, 2\}$ for the unbalanced system, and $a_i \in \{-1, 0, 1\}$ for the balanced system.

One concern in binary number representation is the treatment of the -ve numbers. There are three common techniques:

- (1) sign-magnitude, where a sign is explicitly attached to the front of the string of digits;
- (2) 1's complement, where the representation for a negative number is obtained by subtracting each digit from 1;
- (3) 2's complement, where the representation for a negative number is obtained as in (2) but with a final addition of 1 to the number.

There are disadvantages of all three of these techniques. Both (1) and (2) have two representations for 0 (-0 and +0), while (3) permits the representation of one more negative number than positive. Alternatively, in a balanced system over a set of m values, all the numbers can be represented without using an explicit sign. The sign of a number is the sign of the most significant non-zero digit. Furthermore, in a ternary system, the negative of a number can be found by interchanging 1 and -1 throughout, leaving all zeros unchanged. Hence, addition and subtraction can be performed with the same hardware by sign changes of the addend and subtrahend, respectively, as required. One other advantage of a balanced system is that the procedure of rounding a number is identical to truncation. In a binary system it is not possible, because there is no way for negative correction being applied by digits of lower significance. Therefore, the correct value of the number must be approached from lower digits.

Another concern in binary number representation is that in performing addition (or subtraction), the sum bits depend on the carry from lower bits. Two alternative multiple-valued number systems have been extensively studied in order to reduce or eliminate the ripple through carries. The first one is *residue* number system, in which there are no carries between bits. In such a representation, operations occur at each digit independently of the other digits, resulting in fast arithmetic operations [5,6]. A disadvantage is that the size of the digits may vary, and thus different circuit designs might be needed for different digits.

The second number representation which has potential performance attractions is a number

system with *redundancy*. In such a system, all numbers except 0 are not uniquely represented by a string of digits. Instead, two or more representations for a given number are available. The most significant digit does not depend on the least significant bit. The carry into a digit is computed only from (at most) the next two lower digits, but no other, enabling fast arithmetic operations. Multiple-valued arithmetic in redundant balanced number system [1,7] as well as in redundant unbalanced number system [8,9] has been presented.

Some other number representations which have potential advantages over binary have been studied, including overlap resolution number system based on signed continuous valued digits, allowing to perform arithmetic operations by analog digit manipulation circuitry [10] and redundant complex number system [5], allowing to perform addition and multiplication of complex numbers without treating real part and imaginary part separately as well as enabling carry-free addition and binary-tree multiple-operand addition.

Sign magnitude representation of ternary number system

In decimal number system a plus (+) sign is used to denote a positive number and minus (-) sign for negative number. The plus sign is usually dropped, and absence of any sign means that number is positive value. This representation of number is known as sign number. As is ternary circuits there are two configurations i.e. balance and unbalance ternary systems. In unbalance ternary system, the positive number representation is same as that of binary system where as negative numbers are represented in its 1's and 3' complement form. The example below represents the number representation in ternary system.

Suppose decimal number 64 is to be represented in its positive and negative form.

So positive $(64)_{10}$ is to be represented in unbalance ternary form it will be $(02101)_3$ here the left most 0 (MSB) indicates that number is positive. On other hand in the negative signed ternary $(64)_{10}$ represents as $(20122)_3$ considering 3' complement of 64.

Similarly decimal $(+48)_{10}$ can be represented as $(01210)_3$ and its negative $(-48)_{10}$ as $(21021)_3$.

1' Complement representation

In a ternary number if each 2 is replaced by 0 and each 0 by 2 by keeping 1 as is the resulting number is known as the 1's complement of given number. For example $(52)_{10}$ is represented as 1221 and its 1's complement is represented as 1001.

3' Complement representation

If 1 is added to 1's complement of a ternary number the resultant number is known as 3's complement of ternary number. For example 3' complement of 2102 will be 0121. For n trit number the maximum positive number which can be represented in 3's complement form is $(3^{n-1}-1)$ and the maximum negative number in 3's complement form is -3^{n-1} .

FLOATING POINT REPRESENTATION OF TERNARY NUMBER

Floating point representation in ternary is same as that of binary except one tryte (groop of 9 trits) is use to represent exponent and two trits for mantissa for 27 trit word which is composed of three trytes.The maximum exponent value, heptavintimal (base 27) ZZZ, is set aside to encode values that are “not a number” or NAN. Aside from the representations reserved for undefined and infinite values, other NAN values have no predefined meaning. In general conventions for floating point representation is [S] M E where S is sign of a number is mantissa and E is in an exponent.

Sign	Mantissa (9 trits)	Exponent(18 trits)
------	--------------------	--------------------

27-trits representation for ternary number

Balanced ternary is a natural choice for both exponent and mantissa. As there is no concept of a sign trit in balanced ternary, we cannot follow the almost-universal convention used on binary system of separating the sign bit from the rest of the mantissa.

The mantissa is represented as a pure fraction strictly less than +1.0 and strictly greater than -1.0; there is no hidden bit, so the point is immediately to the left of the most significant trit of the mantissa. The lack of a hidden bit allows non-normalized values with any exponent, but all arithmetic operations on floating point values should return normalized results. Thus, when the exponent is greater than ----- (9 values = heptavintimal 000), the most significant trit of the mantissa should be nonzero. For normalized numbers, therefore, the absolute value of the mantissa is greater than or equal to 1/3.[26].

ternary		heptavintimal	meaning
+++++++	+++++++	ZZZ ZZZZZZ	positive infinity
+++++++	000000000000000000	ZZZ DDDDDD	undefined
+++++++	+++++++	ZZZ 000000	negative infinity
-----	000000000000000000	000 DDDDDD	zero

BASIC TERNARY LOGIC CIRCUITS

In the digital systems there are few basic operations performed irrespective of complexity of system. These operations may be required to perform a number of times in a large digital system like computer or any digital control system etc. These basic operations are AND, OR, NOT and flip flop. In ternary logic system, these operations are carried out by ternary gate (T-Logic gate) which is labeled as T-AND, T-OR, T-NOT and Flip-Flap-Flop. In recent days T-gates are implemented by using switching elements like CMOS, Resonant tunneling Diodes (RTD) and Carbon Nano Tubes (CNT). Device implementation is considered in chapter 3.

The Ternary Inverter

Ternary inverter is a circuit that gives the output in inverted form of input. Three types of inverter operations are possible in ternary logic. These are STI (simple ternary inverter) PTI (positive ternary inverter) & NTI (Negative ternary inverter) such that,

$$\begin{aligned} STI &\equiv \overline{X^i} = 2 - x \\ PTI, NTI &\equiv \overline{X^i} = \begin{cases} ii\bar{x} \neq i \\ 2-ii\bar{x} = i \end{cases} \end{aligned} \quad (1)$$

Where i take the value of 2 for PTI & 0 for NTI inverter.

Symbols for PTI STI of NTI are shown in Figure 1.5

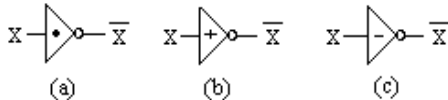


Figure 1.5 Symbols for Inverters (a) STI (b) PTI (c) NTI.

The symbols ‘.’, ‘+’, and ‘-’ are used to represent simple, positive and negative logic ternary inverters. Device level construction, operation and truth tables for each type of inverters illustrated in chapter 3.

Ternary OR/NOR Logic Gate

In general OR operation is defined as $(y) = \max(x_1, x_2 \dots x_n)$ i.e. where y is an output and x's inputs.

Ternary OR is a circuit that have $X_1 \dots X_n$ as input & Y_0 as output such that

$$T - OR = Y_0 = X_1 + X_2 + \dots + X_n = \text{Max}[X_1, X_2 \dots X_n] \quad (2)$$

& Ternary NOR has an output that is compliment as OR function i.e.

$$T - NOR = \overline{Y_0} = \overline{X_1 + X_2 + \dots + X_n} = \text{Max} [\overline{X_1}, \overline{X_2} \dots \overline{X_n}] \quad (3)$$

The sign ‘+’ indicates logical ternary OR logic operation.

Depending upon the type of inverter used, the logic functions T-OR/NOR can be Simple ternary OR/NOR [ST-OR/NOR], Positive ternary OR/NOR [PT-OR/NOR], Negative ternary OR/NOR [NT-OR/NOR]

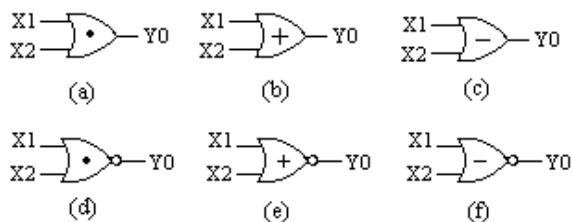


Figure 1.6: Shows Symbols for T-OR/NOR logic gates (a) ST-OR (b) PT-OR (c) NT-OR (d) ST-NOR (e) PT-NOR (f) NT-NOR

Figure 1.6 is a representation of symbols for T-OR/NOR operation. The device level construction, operation and truth table for each type of OR/NOR logic gate is covered in chapter 3.

Ternary AND/NAND Logic Gate

AND operation is defined as $(y) = \min(x_1, x_2, \dots, x_n)$ i.e. where y is an output and x 's are inputs i.e.

$$T - \text{AND} = Y_0 = X_1.X_2 \dots X_n = \text{Min}[x_1x_2x_n] \quad (4)$$

$$T - \text{NAND} = \overline{Y_0} = \overline{X_1.X_2 \dots X_n} = \text{Min}[\overline{x_1x_2x_n}] \quad (5)$$

PT-AND/NAND & NT-AND/NAND are constructed by using PTI & NTI

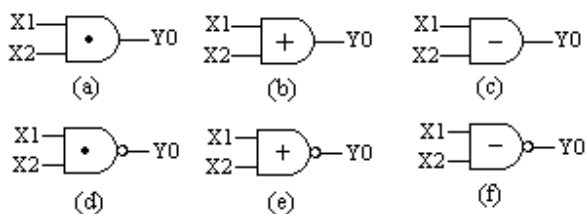


Figure 1.7: Shows symbols for T-AND/ NAND logic gate. (a) ST-AND ([b) PT-AND (c) NTAND (d) T-AND (e) PT-NAND (f) NT-NAND.

Where Y = output of T-Logic AND/NAND gate and X = output of T-Logic AND/NAND gate.

Figure 1.7 is a representation of symbols for T-AND/NAND gates. The device level construction, operation and truth table for each type of AND/NAND logic gate is covered in chapter 3.

Ternary EX-OR/EX-NOR

Ternary Ex-OR is ternary addition neglecting carry.

It is defined as

$$T - \text{EX} - \text{OR} = X_1 \oplus X_2 \quad (6)$$

$$T - \text{EX} - \text{NOR} = \overline{X_1 \oplus X_2} \quad (7)$$

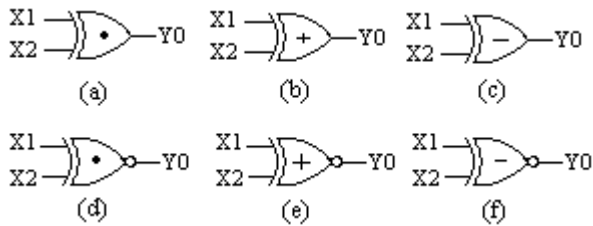


Figure 1.8 shows symbol for T-EX-OR/NOR logic gates.

Figure 1.8 Symbol for T-Ex-OR/ Ex-NOR gate. (a) T- Ex-OR (b) T-Ex-NOR

Chapter 3 covers construction, operation and truth table for each type of OR/NOR logic gate.

BOOLEAN ALGEBRA FOR TERNARY LOGIC

Algebra proposed by George Boole is also valid for ternary logic system with some additional rules for ternary. The complete theorem and proofs for algebra are given below.

Let the three voltage levels of ternary logic circuits be represented by 0, 1 and 2. The 0 represents the low, 1 the intermediate and 2 the high level. Let $L = \{0, 1, 2\}$. In L a set of operators are defined. For $x, y, z \in L$ there exists an equivalence ($=$) operation, that is:

$$x = x$$

$$\text{If } x = y, \text{ then } y = x$$

$$\text{If } x = y \text{ and } y = z, \text{ then } x = z$$

The simple ternary inverter (STI), positive ternary inverter (PTI), negative ternary inverter (NTI), forward diode (FD) and reverse diode (RD) are considered as basic unary operators. These are represented by equations (1)-(3),

$$STI \equiv \overline{X}^i = 2 - x \quad (8)$$

$$PTI, NTI \equiv \overline{X}^i = \begin{cases} i \text{ if } x \neq i \\ 2 - i \text{ if } x = i \end{cases} \quad (9)$$

Where, i take the value of 2 for PTI and 0 for the NTI operator. The minus sign in equations (1) and (2) represents arithmetic subtraction.

$$FD, RD \equiv x^{k_i} = \begin{cases} 1 \text{ if } x \neq i \\ i \text{ if } x = i \end{cases} \quad (10)$$

Where i can be 2 or 0, k_2 represents the FD operator ($\neg$), and k_0 the RD operator ($\bar{}$). The operation of addition (+) and multiplication (.) on L , which can be called ternary OR(TOR) and ternary AND(TAND) respectively, represent two multiple input operators. These are given in following equations:

$$TOR \equiv x + y = MAX(x, y) \quad (11)$$

$$TAND \equiv x \cdot y = MIN(x, y) \quad (12)$$

The two-element operations obey the idempotent, commutative, associative, distributive and absorption laws.

It is evident that laws of the identity elements hold also here.

$$x + 0 = x \quad (14)$$

$$x \cdot 2 = x \quad (15)$$

$$x + 2 = 2 \quad (16)$$

$$x \cdot 0 = 0 \quad (17)$$

Theorem: DeMorgan's Theorem holds for ternary logic when the three types of inverters are used:

$$\overline{(x + y)^0} = \overline{x^0} \cdot \overline{y^0} \quad (18)$$

$$\overline{(x \cdot y)^0} = \overline{x^0} + \overline{y^0} \quad (19)$$

$$\overline{(x + y)^1} = \overline{x^1} \cdot \overline{y^1} \quad (20)$$

$$\overline{(x \cdot y)^1} = \overline{x^1} + \overline{y^1} \quad (21)$$

$$\overline{(x + y)^2} = \overline{x^2} \cdot \overline{y^2} \quad (22)$$

$$\overline{(x \cdot y)^2} = \overline{x^2} + \overline{y^2} \quad (23)$$

All theorems, laws and relationships presented in this book have been proven in reference [5].

The theorems and laws presented above are nearly the same as those of the Boolean algebra with a little generalization. But there will be some differences due to the existence of three types of complements in the algebra presented here. This is shown clearly in the following theorems.

For any $x, y, z \in L$, the following theorems hold:

$$x + \overline{x^2} = 2 \quad (24)$$

$$x \cdot \overline{x^0} = 0 \quad (25)$$

$$(x \cdot y) + \left(x \cdot \overline{y^2} \right) = x \quad (26)$$

$$(x + y) \cdot \left(x + \overline{y^0} \right) = x \quad (27)$$

$$x + \left(\overline{x^2} \cdot y \right) = x + y \quad (28)$$

$$x \cdot \left(\overline{x^0} + y \right) = x \cdot y \quad (29)$$

$$\overline{(z \cdot x) + (z \cdot \overline{x^2} \cdot y)} = \overline{(z \cdot x) + (z \cdot y)} \quad (30)$$

$$(z + x) \cdot \left(z + \overline{x^0} + y \right) = (z + x) \cdot (z + y) \quad (31)$$

$$(x \cdot y) + \left(\overline{x^2} \cdot z \right) + (y \cdot z) = (x \cdot y) + (\overline{x^2} \cdot z) \quad (32)$$

$$(x + y) \cdot \left(\overline{x^0} + z \right) \cdot (y + z) = (x + y) \cdot (\overline{x^0} + z) \quad (33)$$

There is a set of relationships which interrelate any inverter with the two others. These can be expressed by the following equations:

$$\overline{x^0} + \overline{x^1} + \overline{x^2} = \overline{x^2} \quad (34)$$

$$\overline{x^0} \cdot \overline{x^1} \cdot \overline{x^2} = \overline{x^0} \quad (35)$$

$$\overline{x^0} + \overline{x^1} = \overline{x^1} \quad (36)$$

$$\overline{x^0} \cdot \overline{x^1} = \overline{x^0} \quad (37)$$

$$\overline{x^2} + \overline{x^1} = \overline{x^2} \quad (38)$$

$$\overline{x^2} \cdot \overline{x^1} = \overline{x^1} \quad (39)$$

$$\overline{\overline{\overline{x^1}}} = x x^1 \quad (40)$$

$$\overline{\overline{\overline{x^0}{}^i}} = \overline{x^0} \quad (41)$$

$$\overline{\overline{x^i}} = \overline{x^2} \quad (42)$$

Where $i = 0, 1$, or 2 .

$$\overline{x^2} = \overline{x^1} \quad (43)$$

$$\overline{x^1} = \overline{x^2} \quad (44)$$

$$\overline{x^2} = \overline{x^2} = \overline{x^2} \quad (45)$$

$$\overline{x^0} = \overline{x^0} = \overline{x^0} \quad (46)$$

$$\overline{x^0} + \overline{x^0} = 2 \quad (47)$$

$$\overline{x^0} \cdot \overline{x^0} = 0 \quad (48)$$

$$\overline{x^2} + \overline{x^2} = 2 \quad (49)$$

$$\overline{x^2} \cdot \overline{x^2} = 0 \quad (50)$$

Where $i = 0, 1$, or 2 .

For any $x, y \in L$ there is also a set of relationships governing the manipulations of the FD and RD operators:

$$x^\neg = x + 1 \quad (51)$$

$$x^- = x \cdot 1 \quad (52)$$

$$\overline{(X^1)^\neg} = \overline{(X^0)^\neg} \quad (53)$$

$$\overline{(X^1)^-} = \overline{(X^2)^-} \quad (54)$$

$$(\overline{X^1})^\neg = (\overline{X^-})^1 \quad (55)$$

$$\overline{(X^1)^-} = \overline{(X^-)^1} \quad (56)$$

$$\overline{(X^-)^2} = \overline{x^2} \quad (57)$$

$$\overline{(X^-)^0} = \overline{x^0} \quad (58)$$

$$\overline{(X^-)^2} = 2 \quad (59)$$

$$\overline{(X^-)^0} = 0 \quad (60)$$

$$(x + y)^- = X^- + Y^- \quad (61)$$

$$(x \cdot y)^- = X^- \cdot Y^- \quad (62)$$

$$(x + y)^- = X^- + Y^- \quad (63)$$

$$(x \cdot y)^- = X^- \cdot Y^- \quad (64)$$

Ternary functions of one or more variables may be represented in truth table or map form or algebraically in canonical form as a product of sums or sum of products.

Ternary Theorem

Any ternary function $f(x_1, x_2, \dots, x_n)$ may be generated from $x_1, x_2, \dots, x_n$ by means of $^+, \cdot$, the unary functions $\overline{x_0}, \overline{x_1}, \overline{x_2}, \overline{x^-}$ and $\overline{x^+}$

It has been proven by Halpern and Yoeli [6] that an algebra composed of the MAX, the MIN and three unary operators $\overline{x}, \overline{x^{10}}$, and $\overline{x^1}$ with the constant 1 is a functionally complete system. Since the set of operators presented here are equivalent to those of the algebra to Halpern and Yoeli except the constant 1 which is to be substituted by the FD and RD operators depending on the relationships given by equations (51) and (52), therefore the above theorem holds.

Moreover it has been shown in reference [8] how these ternary operators realize algebras due to Post, Rosser and Terquette [11], Yoeli and Rosenfeld, Vacca [12] and Mine et al. [10,13], which have been all proven to be functionally complete.

Any ternary function of n variables can be represented by:

$$f(x_1, x_2, \dots, x_n) = 2 \cdot F_2(x_1, x_2, \dots, x_n) + 1 \cdot F_1(x_1, x_2, \dots, x_n) + 0 \cdot (x_1, x_2, \dots, x_n) \quad (65)$$

$$\text{i.e., } f = 2 \cdot F_2 + 1 \cdot F_1 + 0 \cdot F_0 \quad (66)$$

Where F_k equals 2 when the value of the function f equals k , otherwise, it will equal 0.

Applying equations (15) and (17) to the above equation, the function may be represented by:

$$f = F_2 + 1 \cdot F_1 \quad (67)$$

And with the aid of the relationship given in equation (52) the function f can be directly represented by:

$$f = F_2 + \overline{F_1} \quad (68)$$

Methods for minimization of ternary switching functions, previously described by several authors [1,7] and [14,15], can be applied to the algebra presented here. Most of these methods were applied to algebras composed of the MAX and MIN operations unary functions defined:

$$x^i = \begin{cases} 0 & \text{if } x \neq i \\ 2 & \text{if } x = i \end{cases} \quad (69)$$

Where i can be 0, 1 or 2.

To be able to apply these methods the following transformations have to be made.

$$x^0 = \overline{\overline{x^0}} \quad (70)$$

$$x^1 = (x + \overline{x^1})^2 \quad (71)$$

$$x^2 = \overline{\overline{x^2}} \quad (72)$$

Verification for these transformations as well as other methods of minimization is described in reference [5].

References

1. Porat DI. Three-valued Digital Systems. Proc. IEE. 1969; 116: 947-954.
2. Smith KC. The prospects of multivalued logic technology & application view. IEEE transaction on computer. 1981; 30: 619-627.
3. Balla PC, Antoniou A. low power dissipation MOS ternary logic family. IEEE journal on solid state circuits. 1984; 19: 739-749.
4. Chung-Yu-Wu. Design& application of pipelined dynamic CMOS ternary logic & simple ternary differential logic" IEEE journal on solid state circuits. 1993; 28: 895-906.
5. Mouftah HT. Three-valued logic and its implementation with COS/MOS integrated circuits [dissertation]. Laval University, Canada. 1975.
6. Halpern I, Yoeli M. Ternary Arithmetic Unit. Proc. IEE. 1968; 115: 1385-1388.
7. Yoeli M, Rosenfeld G. Logical Design of Ternary Switching Circuits. IEEE Trans. Elect. Comp.1965; 14: 19-29.
8. Mouftah HT, Jordan IB. Integrated Circuits for Ternary Logic. Proceedings of the 1974 Inter. 1974; 285-302.
9. Mouftah HT, Jordan IB. A Design Technique for an Integrable Ternary Arithmetic Unit. Proceedings of the 1975 International Symposium on Multiple-valued Logic. 1975; 359-372.
10. Mine H, Hasegawa T, Ikeda M, Shintani T. A Construction of Ternary Logic Circuits. Electron. Communication in Japan. 51: 133-140.

11. Rosser JB, Turquette AR. Many-valued Logics. North-Holland Publishing Co. Amsterdam. 1952.
12. Vacca R. A Three-valued System of Logic and its Applications to Base Three Digital Circuits. Proc Intern. Conf. Inform. Proceeding, (UNESCO). 1959; 407-414.
13. Post EL. Introduction to a general theory of Elementary Propositions. American Jour Math. 1921; 43: 163-185.
14. Bitran M, Strutt MJO. Minimization of Ternary Logic and Complete Set of Integrated Circuits. Electron. And Comm. 1971; 25: 387-392.
15. Nutter RS, Swartwout RE. A Ternary Logic Minimization Technique. Conference Record of the 1971 Symposium on the Theory and Applications of Multiple-valued Logic Design. 1971; 112-123.