

# Optimization of 1-Bit ALU using Ternary Logic

A.Swetha Priya<sup>1</sup>, A.Sathish Kumar<sup>2</sup>

<sup>1</sup>Lead Engineer, Cerium Systems, Bangalore

<sup>2</sup>IC Design Engineer, Broadcom, Bangalore

\*\*\*

**Abstract** - In this paper, we presented a novel approach for implementation of 1-bit ALU using Ternary logic. The ternary logic or Three Valued Logic (3VL) is the next alternative approach offering several advantages over existing conventional binary digital logic. The proposed 3VL based ALU is designed for 1-bit operation and can be used for real time applications with better hardware reduction in turn minimizing the number of gates over binary logic. The 3VL based ALU is designed using CMOS ternary logic gates (T-Gates) for ternary based arithmetic and logical circuits which is suitable for LSI/VLSI implementation. The design can be extended to n-bit operation for real time applications. Today's existing simulation tools do not permit the simulation of 3VL circuits. VHDL is used as a simulation tool for 3VL verification. An objective of this study is to describe how existing 3VL system can be optimized in terms of transistors over binary logic. Ternary logic resulted in 25% reduction of transistor utility in 1-Bit ALU design when compared to binary logic.

**Key Words:** CMOS; T-ALU; T-Gates; Ternary logic; VHDL

## 1. INTRODUCTION

Is it time to move beyond zeroes and ones? This very thought of Bernard Cole's [1] article published in 2003 on the official site of the Embedded Development Community brought many researches to work upon Multi-Valued Logic to bring a new era of technology. Design of combinational and sequential circuits traditionally uses binary value. Two-valued binary logic has provided efficient two-state devices and circuits with its current status of complexity and sophistication reached mainly because of the continual development of microelectronics [1]. Interconnection problem exists as a intricate problem on both on-chip and between the chips for two-valued logic. Difficulty on chip for placement and routing (P&R) of digital logic elements increase in terms of capability per chip escalated on a complete chip. Also, silicon area used for inter connections are greater than that of the active logic elements used in the chip.

The first known classical father of logic, Aristotle found logical calculus where traditionally only two possible values (i.e., true and false) was obliged to be used for any proposition which includes or assumes the law of the excluded middle until the 20<sup>th</sup> century. Extension to mainstream formulaic two-valued logic ( $n = 2$ ) called  $n$ -valued logic is brought out for  $n > 2$  with the 20<sup>th</sup> century. As a measure of the cost or complexity of three valued circuits, the mathematical work of product of the radix and the

number of signals has been proposed by Godel. Godel [2] has stipulated that the most efficient base for reducing the hardware in a circuitry is Euler constant ( $e = 2.718285...$ ). Since decadic 3 is the digit nearer to Euler Constant than base 2 offering great significance over the design of binary digital systems.

In fact, in reality, there is no clarity on the yes/no conditions. Real world scenarios like true/false/may be, open/close/half open or half close, yes/no/may be, left/right/straight or up/down/straight are observed conditions with uncertainty in decision making values. Multiple decision making scenarios are encountered to understand the certainty level for the acceptance or rejection of the situations [3-7]. Unreliability-Reliability probabilistic model is as shown in Fig-1. The deterministic model explains that the reliability of the model is interpreted within the boundaries of  $0 \leq R \leq 1$ . Vicinity of  $R$  being 0.5 could be interpreted as may be true or may be false and above 0.5 is said to be reliability level and below 0.5 is said to be unreliability level. It is interpreted to be certainly happening if  $R=1$ . And if  $R=0$ , it is interpreted to be not happening certainly.

Higher processing rates are achieved going up from unary logic level to higher logic level. Traditionally, there are many bases identified by the mankind from base-1 (Unary) to base 109. As shown in Fig-2, any value apart from binary logic is called non-binary logic which is expressed in terms of  $n$ -value and so is called Multi-Valued Logic (MVL). Higher processing levels gives advantage for memory management, communication throughput and domain specific computations. Economy of digits is an evident advantage of a ternary representation. It needs 58% more digits for binary to represent a number compared to ternary logic. These benefits have shown to be useful in the design of ternary computers, for digital filtering [8]. Ternary logic allows sign conversions. These advantages of ternary are casted in many applications in the field of Data Mining, Fuzzy logic, Machine Learning, Robotics, Artificial Intelligence, Digital signal processing, Image Processing and Digital control systems. Mainly, Ternary is used in State Assignments and decision diagrams, error correction, error encoding and decoding, data compression and representation of discrete information, Automatic Theorem proving and in automatic telephony.

The scope of the paper is to implement the novel idea to explore the possibilities and advantages in realizing switching circuits which reduces T-gates for ALU slice. Section 2 briefs out our preliminary study in the earlier papers on ternary logic to recognize and to bring out its

importance. Section 3 gives the basic study of ternary logic gates. In Section 4, gives the ternary 1-bit ALU operation. The simulation of the ALU and its results is described in section 5. Section 6 shows the advantages of ternary over binary and the conclusion is given in Section 7.

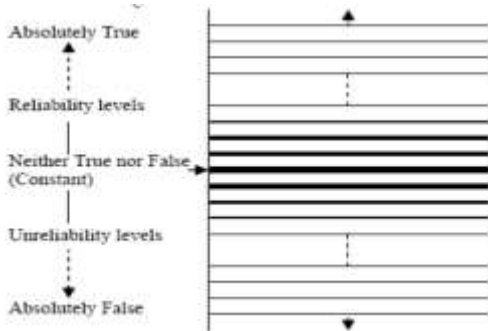


Fig -1: Implementation of ternary function

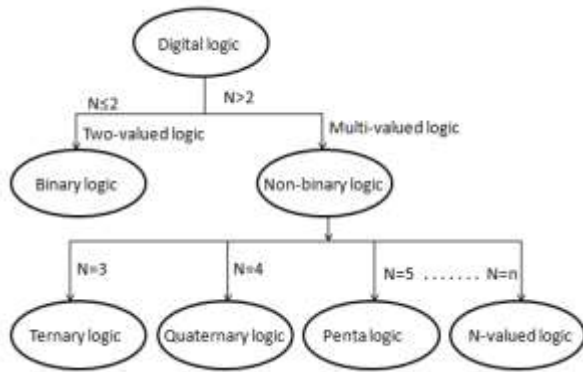


Fig -2: Levels of Switching Algebra and its extension to N-valued logic

## 2. TERNARY SWITCHING ALGEBRA

For a ternary switching algebra, a suitable set of operations ought to be selected from the set of all possible variables [1-5]. Let a system be L whose elements called propositions or statements are valued in the set  $\{\alpha, \beta, \gamma\}$ . If X is a proposition, the value of X can be seen as a mapping  $V : L \rightarrow \{\alpha, \beta, \gamma\}$  such that

$$V(X) = \begin{cases} \gamma & \text{if X is true} \\ \beta & \text{if X is intermediate} \\ \alpha & \text{if X is false} \end{cases} \quad (1)$$

Objective of using general variables  $\alpha, \beta, \gamma$  is to consider any type of present or future device technology and also for positive or negative logic representation. Ternary logic level ' $\alpha$ ' generally corresponds to logic-0 also called low voltage which is well defined in binary, ' $\beta$ ' corresponds to an intermediate or uncertainty stage and ' $\gamma$ ' corresponds to logic-1 also called high voltage which is well defined in binary. The intermediate state ' $\beta$ ' can be metaphorically thought of partially true and partially false condition.

### 2.1 Basic Laws of Ternary Logic

For  $X, Y, Z \in L$ , there exists an equivalence (=) operation, such that  $X = X$ , If  $X = Y$ , then  $Y = X$ , If  $X=Y$  &  $Y=Z$ , then  $X = Z$ .

Clearly  $(L, +, \bullet)$  is a distributive lattice with zero element(0 or  $\alpha$ ) and universal element(2 or  $\gamma$ ). Thus  $X, Y, Z \in L$  [1-5]:

$$\text{Idempotent Law : } X+X = X \quad X \bullet X = X \quad (2)$$

$$\text{Commutative Law: } X+Y = Y+X \quad X \bullet Y = Y \bullet X \quad (3)$$

$$\text{Associative Law : } (X+Y) + Z = X+(Y+Z) \quad (4)$$

$$X \bullet (Y \bullet Z) = (X \bullet Y) \bullet Z \quad (5)$$

$$\text{Absorption Law : } X+X \bullet Y = X \quad X \bullet (X+Y) = X \quad (6)$$

$$\text{Distributive Law: } X+Y \bullet Z = (X+Y) \bullet (X+Z) \quad (7)$$

$$X \bullet (Y+Z) = X \bullet Y+X \bullet Z \quad (8)$$

To implement three valued logic function, unary variables are primarily selected through a Ternary decoder approach as shown in the Fig-3, which converts all the unary variables to suitable conditions of ternary variables to formulate a ternary function.

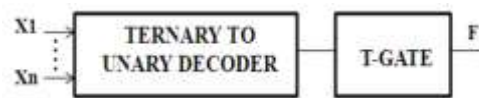


Fig -3:

Implementation of ternary function

Literal is denoted by  $X_i^{\alpha_i}$ , where  $\alpha_i = \alpha, \beta, \gamma, \alpha\beta, \beta\gamma$  and  $\alpha\gamma$  is defined as given Table -1.

Table -1: Function Table of Unary Functions

X	X $\alpha$	X $\beta$	X $\gamma$	X $\alpha\beta$	X $\beta\gamma$	X $\alpha\gamma$
$\alpha$	$\gamma$	$\alpha$	$\alpha$	$\gamma$	$\alpha$	$\gamma$
$\beta$	$\alpha$	$\gamma$	$\alpha$	$\gamma$	$\gamma$	$\alpha$
$\gamma$	$\alpha$	$\alpha$	$\gamma$	$\alpha$	$\gamma$	$\gamma$

Some expressions useful in ternary logic are:

$$X^i = \begin{cases} \alpha & \text{if } X \neq i \\ \gamma & \text{if } X = i \end{cases} \quad \text{Where } i = \alpha, \beta \& \gamma \quad (9)$$

$$X^{\alpha\beta} = X^\alpha + X^\beta = X^\gamma \quad (10)$$

$$X^{\beta\gamma} = X^\beta + X^\gamma = X^\alpha \quad (11)$$

$$X^{\alpha\gamma} = X^\alpha + X^\gamma = X^\beta \quad (12)$$

$$X^{\alpha\beta} \bullet X^{\beta\gamma} = X^\beta \quad (13)$$

$$X^{\beta\alpha} \bullet X^{\alpha\gamma} = X^\alpha \quad X^{\alpha\gamma} \bullet X^{\beta\gamma} = X^\gamma \quad (14)$$

$$X + \alpha = X \quad X \bullet \alpha = \alpha \quad (15)$$

$$X + \gamma = \gamma \quad X \bullet \gamma = X \quad (16)$$

Universally formulated De Morgan's Theorem [5] also holds good for ternary logic. Three types of inverters are used to implement De Morgan's theorem.

$$\overline{(X+Y)^\alpha} = \overline{X^\alpha} \bullet \overline{Y^\alpha} \quad (17)$$

$$\overline{(X \bullet Y)^\alpha} = \overline{X^\alpha} + \overline{Y^\alpha} \quad (18)$$

$$\overline{(X+Y)^\beta} = \overline{X^\beta} \bullet \overline{Y^\beta} \quad (19)$$

$$\overline{(X \bullet Y)^\beta} = \overline{X^\beta} + \overline{Y^\beta} \quad (20)$$

$$\overline{(X+Y)^\gamma} = \overline{X^\gamma} \bullet \overline{Y^\gamma} \quad (21)$$

$$\overline{(X \bullet Y)^\gamma} = \overline{X^\gamma} + \overline{Y^\gamma} \quad \overline{\overline{X^\beta}^\beta} = X \quad (22)$$

The canonical Sum of Product (SOP) and Product of Sum (POS) Product of Sum (POS) can be written as

$$f(x_1, x_2, \dots, x_n) = \gamma \cdot g(x_1, x_2, \dots, x_n) + \beta \cdot h(x_1, x_2, \dots, x_n)$$

i.e.,  $f = g + \beta \cdot h$  (23)

$$f(x_1, x_2, \dots, x_n) = (\gamma \cdot g(x_1, x_2, \dots, x_n)) \cdot (\beta + h(x_1, x_2, \dots, x_n))$$

i.e.,  $f = g \cdot (\beta + h)$  (24)

### 3. Ternary Gate Design

The basic building block of ternary decoder is a general ternary inverter (GTI). A GTI takes a unary operator with one input  $X$  and produces 3 possible outputs. The implementation of ternary inverter requires three types of inverters namely negative ternary inverter (NTI), simple ternary inverter (STI) and positive ternary inverter (PTI) forming an operator set that is complete in logic sense[6-9]. These T-inverter acts as a basic fundamental element for constructing ternary AND/NAND (T-AND/NAND), ternary OR/NOR( T-OR/NOR) and soon. Table -2 summarized the functioning of three types of ternary Inverters.

$$STI = \overline{X^\beta} = \gamma - X \quad (25)$$

$$PTI, NTI = \overline{X^i} = \begin{cases} i & \text{if } X \neq i \\ \gamma - i & \text{if } X = i \end{cases} \quad (26)$$

where  $i$  takes the value of  $\gamma$  for PTI and  $\alpha$  for the NTI operator. The minus sign represents arithmetic subtraction. Symbols of Ternary Inverters are given in Fig-4.

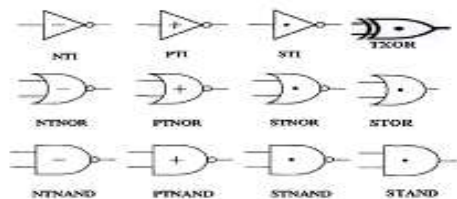


Fig -4: Basic Ternary logic gates

Table -2: Function Table For Ternary Inverter

X	$\overline{X^\alpha}$ (NTI)	$\overline{X^\gamma}$ (PTI)	$\overline{X^\beta}$ (STI)
$\alpha$	$\gamma$	$\gamma$	$\gamma$
$\beta$	$\alpha$	$\gamma$	$\beta$
$\gamma$	$\alpha$	$\alpha$	$\alpha$

Ternary OR (TOR) and Ternary AND (TAND) represent ORing (+) and ANDing (.) respectively on L in Ternary terminology. These gates are as shown in Fig-4. and basically it requires two or more multiple input operators [1-6]. Functioning of these gates are represented by following equations and are tabulated as given in Table -3.

Binary logic is restricted to only two states '0' and '1' whereas three-valued logics have three truth values. And so Ternary logic is also called three-valued or trivalent logic and is abbreviated to 3VL as it has three distinct voltage levels. 3VL, in general a multi-valued logic(MVL) is implemented in two modes i.e. current mode and voltage mode[10].

Table -3: Truth Table For Basic Gates

A	B	TAN D	SINA ND	FINAN D	NINAN D	TOR	SINO R	FINO R	NINO R
$\alpha$	$\alpha$	$\alpha$	$\gamma$	$\gamma$	$\gamma$	$\alpha$	$\gamma$	$\gamma$	$\gamma$
$\alpha$	$\beta$	$\alpha$	$\gamma$	$\gamma$	$\gamma$	$\beta$	$\beta$	$\gamma$	$\alpha$
$\alpha$	$\gamma$	$\alpha$	$\gamma$	$\gamma$	$\gamma$	$\gamma$	$\alpha$	$\alpha$	$\alpha$
$\beta$	$\alpha$	$\alpha$	$\gamma$	$\gamma$	$\gamma$	$\beta$	$\beta$	$\gamma$	$\alpha$
$\beta$	$\beta$	$\beta$	$\beta$	$\gamma$	$\alpha$	$\beta$	$\beta$	$\gamma$	$\alpha$
$\beta$	$\gamma$	$\beta$	$\beta$	$\gamma$	$\alpha$	$\gamma$	$\alpha$	$\alpha$	$\alpha$
$\gamma$	$\alpha$	$\alpha$	$\gamma$	$\gamma$	$\gamma$	$\gamma$	$\alpha$	$\alpha$	$\alpha$
$\gamma$	$\beta$	$\beta$	$\beta$	$\gamma$	$\alpha$	$\gamma$	$\alpha$	$\alpha$	$\alpha$
$\gamma$	$\gamma$	$\gamma$	$\alpha$	$\alpha$	$\alpha$	$\gamma$	$\alpha$	$\alpha$	$\alpha$

Logic Sum or TOR and Logic Product or TAND:

$$X1 + X2 + \dots + Xn = \text{MAX} (X1, X2, \dots, Xn) \quad (27)$$

$$X1 \cdot X2 \cdot \dots \cdot Xn = \text{MIN} (X1, X2, \dots, Xn) \quad (28)$$

Similarly, TNAND and TNOR is

$$\overline{X1 \cdot X2 \cdot \dots \cdot Xn} = \text{MIN} (\overline{X1 \cdot X2 \cdot \dots \cdot Xn}) \quad (29)$$

$$\overline{X1 + X2 + \dots + Xn} = \text{MAX} (\overline{X1 + X2 + \dots + Xn}) \quad (30)$$

In Voltage mode, logic state are specified in terms of distinct voltage levels i.e.,  $V1, V2, V3, \dots, Vn$ . Voltage modes are further divided into balance modes and unbalanced modes [11]. Balanced modes are the logic states defined in terms of distinct voltage levels i.e.,  $+V = 1$ (logic high), Ground = 0(intermediate state) &  $-V = -1$ (logic low). Balanced mode is more efficiently identified as standard in the industry. And based on the "N" value defined, we have these voltage levels like 1, 0, -1 for ternary logic. Unbalanced modes are the logic states defined in terms of voltage levels i.e,  $V = 2$ (logic high),  $V/2 = 1$ (intermediate logic) & Ground = 0(logic low), (equivalently it is  $V = \gamma, V/2 = \beta$  & Ground =  $\alpha$  or  $V = 1, V/2 = z$  & Ground = 0). It uses non-negative numbers for representation. In Current mode [12], logic states are specified in terms of multiples of reference current states i.e.,  $x1, x2, x3, \dots, xn$  where  $x$  is reference current.

### 4. Design of 1-Bit Ternary ALU

ALU (Arithmetic Logic Unit) which is the central execution unit of a CPU is a combinatorial circuit performing arithmetic and logical operations and its complexity keeps varying [13]. Ternary ALU (T-ALU) slice has been designed for the 1-bit ternary operations. This can be extended for n-bit operations also and can also be implemented by cascading smaller basic  $n/2$  ALU slices as well. This proposed ALU is designed to have CMOS logic T-Gates for ternary arithmetic & logic operations which is suitable for LSI/VLSI implementation. The design of ALU slice is shown in Fig-5 and works as per the functional Table -4. Depending on the selection line (S) and the bits X & Y, the corresponding operations will be performed by T-ALU. Arithmetic operations are as given in the Table -5.

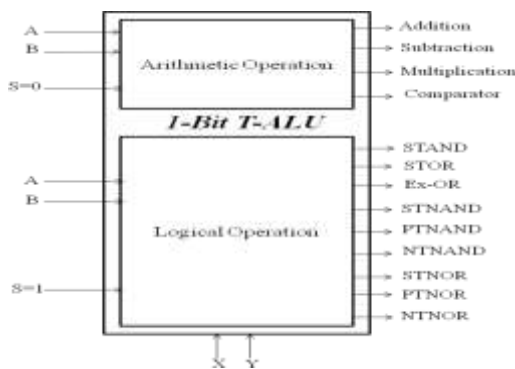


Fig -5: Block diagram of 1-bit T- ALU Slice

A	B	Operation	Operation
$\alpha$	$\alpha$	Addition	TAND
$\alpha$	$\beta$	Subtraction	TOR
$\alpha$	$\gamma$	Multiplication	Ex-OR
$\beta$	$\alpha$	Comparator	STNAND
$\beta$	$\beta$	-	PTNAND
$\beta$	$\gamma$	-	NTNAND
$\gamma$	$\alpha$	-	STNOR
$\gamma$	$\beta$	-	PTNOR
$\gamma$	$\gamma$	-	NTNOR

Table - 6: 9- State Logic System[10]

### 5. Simulator for T-ALU

The performance evaluation of 3VL circuits is dealt with the use of Very High-Speed Integrated Circuit Hardware Description Language (VHDL) as a logic simulator[10-14] which is used to model, simulate and describe tristate valued signals of a ternary systems with true, false and with a third transparent high impedance state. An effective way out to connect different distinct logic outputs with characteristics of high impedance state (Z) to a single input allowing the remaining outputs to operate in the normal binary sense can be done with VHDL. Commonly, the concept is used memory banks of computers and other similar devices to a common data bus where a large number of devices can communicate over the same channel ensuring only one to be enabled at a time.

Table -4: Functional Table of Logical Operations

BITS		ADDITION		SUBTRACTION		MULTIPLICATION		COMPARATOR		
A	B	SUM	CARRY	DIFF	BARR	PROD	CARRY	A>B	A=B	A<B
$\alpha$	$\alpha$	$\alpha$	$\alpha$	$\alpha$	$\alpha$	$\alpha$	$\alpha$	$\alpha$	$\gamma$	$\alpha$
$\alpha$	$\beta$	$\beta$	$\alpha$	$\gamma$	$\beta$	$\alpha$	$\alpha$	$\alpha$	$\alpha$	$\gamma$
$\alpha$	$\gamma$	$\gamma$	$\alpha$	$\beta$	$\beta$	$\alpha$	$\alpha$	$\alpha$	$\alpha$	$\gamma$
$\beta$	$\alpha$	$\beta$	$\alpha$	$\beta$	$\alpha$	$\alpha$	$\alpha$	$\gamma$	$\alpha$	$\alpha$
$\beta$	$\beta$	$\gamma$	$\alpha$	$\alpha$	$\alpha$	$\gamma$	$\alpha$	$\alpha$	$\gamma$	$\alpha$
$\beta$	$\gamma$	$\alpha$	$\beta$	$\gamma$	$\beta$	$\gamma$	$\alpha$	$\alpha$	$\alpha$	$\gamma$
$\gamma$	$\alpha$	$\gamma$	$\alpha$	$\gamma$	$\alpha$	$\alpha$	$\alpha$	$\gamma$	$\alpha$	$\alpha$
$\gamma$	$\beta$	$\alpha$	$\beta$	$\beta$	$\alpha$	$\gamma$	$\alpha$	$\gamma$	$\alpha$	$\alpha$
$\gamma$	$\gamma$	$\beta$	$\beta$	$\alpha$	$\alpha$	$\beta$	$\beta$	$\alpha$	$\gamma$	$\alpha$

The proposed VHDL simulator is developed with technology dependent package called 9-state StdLogic\_1164 package [14] whose levels are as listed in Table -6 which is used to synthesize and to verify the performance of ternary logic circuits. This VHDL permits to describe the circuits like TTL, CMOS, GaAs, NMOS, PMOS and ECL[15,16] based on the design implemented for the examination and requires a change in package development to support the design. To demonstrate the use of VHDL as a ternary logic simulator, we have used Logic Low to represent  $\alpha$  or equivalently 0V, High impedance Z to represent  $\beta$  or equivalently 1V and Logic high to represent  $\gamma$  or equivalently 2V as shown in Fig-6. The simulation result of 1-bit T-ALU is as shown in Fig-7.

Table -5: Functional Table of T-ALU

Bits	For S=0	For S=1
------	---------	---------

Symbols	Values
U	Uninitialized
X	Unknown
0	Logic 0
1	Logic 1
Z	High Impedance
W	Weak unknown
L	Weak zero
H	Weak one
-	Don't care

Fig -6: Waveforms of (a) Two-valued logic and (b) Three-valued logic

### 6. Total Number of Transistor Calculation

For binary based two input AND & OR gate requires 4 transistors and for three input requires 6 transistors. The Binary 1-bit ALU can give only 4 different combinations of



operations like addition, subtraction, multiplication, comparator operation and a final multiplexer to select either one of the operation.

#### 6.1 Binary Logic

For addition operation, totally 6 gates are used.

No. of Inverters are 2  $\rightarrow 2*2 = 4$   
 No. of 2 input AND & OR Gates are 4  $\rightarrow 4*4 = 16$

Total no. of transistors with Inverters  $\rightarrow 20$  transistors  
 For subtraction operation, totally 3 gates are used.  
 No. of Inverters is 1  $\rightarrow 1*2 = 2$   
 No. of 2 input AND is 1  $\rightarrow 1*4 = 4$



No. of 2 input XOR is 1 → 1\*8 = 8

-----

Total no. of transistors with Inverters → **14 transistors**  
 For multiplication operation, totally 7 gates are used.  
 No. of Inverters is 2 → 2\*2 = 4  
 No. of 2 input AND are 4 → 4\*4 = 16  
 No. of 2 input OR is 1 → 1\*4 = 4

-----

Total no. of transistors with Inverters → **24 transistors**  
 For comparator operation, totally 5 gates are used.  
 No. of Inverters is 2 → 2\*2 = 4  
 No. of 2 input AND are 3 → 3\*4 = 12

-----

Total no. of transistors with Inverters → **16 transistors**  
 For multiplexer operation, totally 9 gates are used.  
 No. of Inverters are 4 → 4\*2 = 8  
 No. of 3 input AND are 4 → 4\*4 = 16  
 No. of 4 input OR is 1 → 1\*10 = 10

-----

Total no. of transistors with Inverters → **34 transistors**  
 Other supporting inverters are 5 which would come totally to 118 transistors.

### 6.2 Ternary Logic

For ternary based two input TAND and TOR gate requires 4 transistors and for three input requires 6 transistors.  
 For Addition (uses T- Half Adder) → 4\*4 = 16  
 For Subtraction (uses T- Half subtraction) → 4\*4 = 16  
 For Multiplication (uses T- gates) → 4\*4 = 16  
 For comparator (uses T-gates) → 6\*4 = 24  
 For multiplexer (uses T-gates) → 4\*4 = 16

-----

Total no. of transistors with Inverters → **88 transistors**

**Table -7:** Ternary and Binary Logic Transistor Count Comparison for 1-Bit ALU

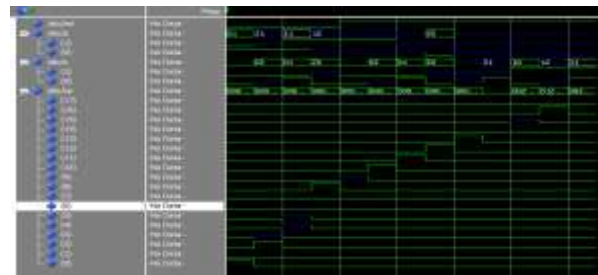
No. Of transistors	Binary logic		Ternary logic	
	1-bit	2-bit	1-bit	2-bit
Without inverters	96	192	78	156
With inverters	118	236	88	176
No. Of combinations	4	16	9	81

**Table -8:** Percentage of Transistor Savings For1-Bit ALU

Total Number of Transistors	Percentage (%) Saving
Without Inverters	19%
With Inverters	25%
No. of wires	18.1%

The total number of transistors comparison between the Ternary and Binary Logic for both 1-bit and 2-bit ALU are given in Table -7. Ternary Logic has showed that fewer numbers of transistors are required to design the ALU when compared to binary logic which as shown in Table -8.

**Fig -7:** The simulation result of 1-bit T-ALU



### 7. Conclusion

In this paper, we developed a ALU using ternary logic with optimum hardware circuitry when compared to binary logic. A scheme based on improved T-gate with formulated design of ALU and also simulation of 1-bit ALU slice is described in this paper. An optimization is performed to compared ternary against the binary to exhibit significance of circuit reduction in complexity, in turn reflecting in low static power consumption due to reduced hardware. Also speed of operation increases as the hardware overhead reduces in non-binary systems compared to binary systems. Moreover, proposed ALU can be easily made to implement several different functions with addition and/or modification in design and selection logic and can be made as realisable system to be incorporated as processing unit in ternary microprocessors. It will reduce the manufacturing cost and complexity and decrease the number of elements required to realize a given ternary function. This Ternary logic helped in about 25% reduction of hardware required in ALU design when compared to binary logic.

The VHDL simulator has been used to perform both synthesis and verification of Ternary logic on designs with a defined StdLogic\_1164 package. To develop Ternary circuits and to verify their performance, tools for digital simulations are necessary to be developed. For layout generation and extraction corresponding software tools are not yet developed by industry. Presently it can be done manually for which mathematical foundation has been completed in this project. Further work may be carried out by larger ALU with the concept of signal co-ordination. Major advantage sought and achieved is reduction in gate count using Ternary logic.

### REFERENCES

- [1] A.Sathish Kumar and A.Swetha Priya, "Minimization Of Ternary Combinational Circuits - A Survey", International Journal Of Engineering Science and Technology (IJEST 2010), Vol. 02(8), ISSN:0975-5462, pp. 3576-3589, Aug. 2010.
- [2] A. Sathish Kumar and A. Swetha Priya, "Modeling Of Combinational Circuits Based On Ternary Multiplexer Using VHDL", International Journal On Computer Science and Engineering (IJCSSE 2010), Vol. 02, No. 05, ISSN:0975-3397, pp.1777-1791, Aug. 2010.
- [3] A.Sathish Kumar and A.Swetha Priya, "Modeling of Sequential Circuits Based On 3VL Flip-flops Using VHDL", International Journal of Computer Information Systems (IJCIS 2010), Vol 1, Issue 5, ISSN 2229-5208, pp 38-47, Dec.2010.

- [4] A.Sathish Kumar and A.Swetha Priya, "TDD Realization For Minimization For Combinational Circuitry", International Journal of Computer Information Systems (IJCIS 2010), Vol 1, Issue 4, ISSN 2229-5208, pp 7-15, Nov. 2010.
- [5] M.Yoeli and G. Rosenfeld, "Logical Design of Ternary Switching Circuits", IEEE Transactions on Electronic Computers, Vol.EC-14, Issue.1, Feb.1965, pp.19-29.
- [6] Raymond E. Miller, "Switching Theory, Vol I", John Wiley & Sons, 1966.
- [7] K.S.Gorde <http://www.terna.org/enewsletter/Apr-Jun2010/VLSI.pdf> (Terna Engineering College)
- [8] Robert L. Herrmann, "Selection and Implementation of a Ternary Switching Algebra", Spring Joint Computer Conference, 1968, pp.283-290.
- [9] H.T. Mouftah, "A Study On The Implementation Of Three-valued Logic", University of Toronto, Toronto, Ontario, Canada, pp.123-126.
- [10] A. P. Dhande and V. T. Ingole, "Design and Implementation of 2 Bit Ternary ALU Slice", 3rd International Conference: Sciences of Electronic (SETIT 2005), IEEE Transc., Mar. 2005, pp.1-11.
- [11] WM El-Medany, MR Hussain, "FPGA-Based Advanced Real Traffic Light Controller System Design", IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Dortmund, Germany, Sep.2007.
- [12] Vikram Banerjee, Mrinal Kanti Mandal and DR Anirudha Ghosal, "Automated Traffic Signal controller", Electronics For You, Project Ideas, Nov.2002. Available as: <http://www.electronicsforu.com/efylinux/circuit/nov2002/traffic.pdf>
- [13] C. Rozon, "On the Use of VHDL as a Multi-Valued Logic Simulator", 26th International Symposium on Multiple-Valued Logic (ISMVL '96), IEEE, 1996, pp.110-115.
- [14] K. C. Smith. (1981): The Prospects for Multivalued Logic: A Technology and Applications View, IEEE Transactions on Computers, Vol. C-30, Issue.9, pp.619-634.
- [15] S.L.Hurst. (1984): Multivalued logic - Its status and its future, IEEE Trans .on Computers, vol. C-33, pp. 1160-1179.
- [16] C. Rozon, "On the Use of VHDL as a Multi-Valued Logic Simulator", 26th International Symposium on Multiple-Valued Logic (ISMVL '96), IEEE, 1996, pp.110-115