IMPLEMENTATION OF COMBINATIONAL CIRCUITS USING TERNARY MULTIPLEXER

Sweta Giri
Dept of ece
Srm univesity

Mrs.N.Saraswathi
Assistant Prof. (S.G)
Dept Of ece
Srm Univesity

Abstract

This is the paper presenting a novel method for defining, analyzing and implementing the basic combinational circuitry with less number of ternary multiplexers. Multilplexer is used as basic building block to realize all the combinational and sequential circuitry providing complete, concise, implementation-free description of the ternary function involved. This shows the potential of VHDL modeling and simulation which can be applied to Ternary switching circuits for verifying its functionality and timing specifications. This is the method which is used in analyzing the complex ternary functions and reduction of gate count

Keyword:- Gate count, Multivalued logic, Reability-unreability model ,VHDL, Ternary switching levels.

1. Introduction

In the Multi-valued logic, a many-valued logic is a propositional calculus in which there are more than two truth values. Traditionally, there was only two possible values (i.e. “true” and “false”).An obvious extension to classical two values logic is an n-valued logic for n greater than 2. The most popular in the literature are the three valued logic, the finite value with the more than three values and the infinite (e.g. Fuzzy logic) logics. The paper emphasis on the three valued logic which has three values (true ,false and intermediate) mathematically. Nowadays, VLSI us the technology which is used for the realization of MVL circuits in order to bring their full potential into many of the operational circuits. Three valued logic has many more advantages over the binary logic. It is possible for the ternary logic to reduce the computation steps, complexity of interconnect, chip area and chip delay. In this logic, higher information carried by the each line offers better utilization of the transmission channels. Serial and serial-parallel operations can be carried out faster and also gives efficient error detection and correction codes when the ternary logic is applied. Section 2 covers the preliminaries to ternary logic to recognize and to bring out its importance. In Section 3, ternary switching algebra with the basic operation of T-gates and its conceptual laws and theorems are presented. Design of combinational and sequential circuits with minimum number of multiplexers are presented in section 4. Section 5 gives the conclusion with future work.

2. Preliminaries Of Ternary Logic

The ternary logic (also called three-valued or trivalent logic and abbreviated 3VL) is a promising alternative to the conventional binary logic design technique. It is possible for ternary logic to achieve simplicity and energy efficiency in digital design since the logic reduces the complexity of interconnects and chip area, in turn reducing the chip delay. It offers better utilization of transmission channels because of the higher

information content carried by each line, also gives more efficient error detection and correction codes and possess potentially higher density of information storage. In principle, MVL can provide a means of increasing data processing capability per unit chip area. Furthermore, serial and serial-parallel arithmetic operations can be carried out faster if the ternary logic is employed. One of the main advantages of ternary logic is that it reduces the number of required computation steps. Since each signal can have three distinct values, the number of digits required in a ternary family is $\log_3 2$ times less than that required in binary logic. It is assumed that ternary-logic elements can operate at a speed approaching that of the corresponding binary-logic elements. However, if the ternary and binary logic gates are used to take advantage of their respective merits, performance could be significantly improved because ternary logic gates are good candidate for decoding block since it requires less number of gates while binary logic gates are a good candidate for fast computation modules. Thus, ternary design technique combined with the conventional binary logic gate design technique also provides an excellent speed and power consumption characteristics in data path circuit such as full adder and multiplier.

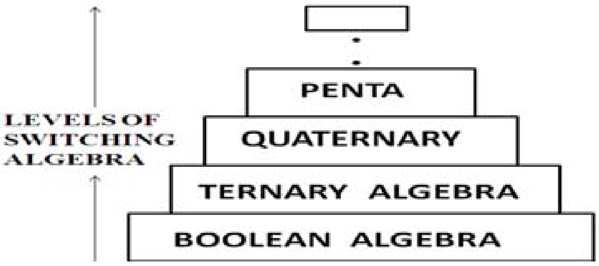


Fig1: Levels of switching algebra

3. Ternary Switching Algebra

Ternary has the logic levels ‘0’ corresponding to logic-0 in binary (also called zero element or low voltage), ‘1’ corresponding to an intermediate stage (also called Meta stable state) and ‘2’ corresponds to logic-1 in binary (also called universal element or high voltage). The intermediate state can be metaphorically thought of as either true or false. The binary logic is limited to only two states ‘1’ and ‘0’, where as MVL is a set of finite or infinite number of values. In a standard CMOS process, the three supply voltages are vdd, vdd/2 and ground. Ternary logic gates are the basic building blocks in realizing combinational and sequential logic functions. The implementation is based around (bipolar transistors, MOSFETs etc.) a basic switching elements, which is referred to as T-Gates . The Ternary gate called T-gate qualifies as a universal element in several different senses. Firstly, it should be logically complete with simple operation. Secondly, it should be easily implemented with its straightforward construction. Thirdly, it should possess two essential elements that must be embodied in any logic gate, namely, logic-value thresholding and logic-signal connection of switching . This functional completeness of T-gate is the property of a set of compositions which enables one to synthesize any arbitrary switching function within a particular class. There are several algebras available for the design of ternary switching functions among which, the Post and the Modular algebra have the advantages of similarity with ordinary algebra.

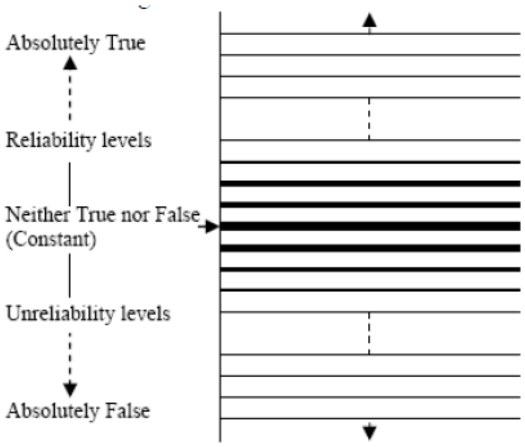


Fig 2: Truth table decision based on Reability-Unreability model

4.Design And Verification Of Ternary Circuits

This is the paper in which 3:1 MUX is taken as a basic building block to explore the realization of circuits with minimum number of ternary 3:1 MUX . Ternary K map method is the concept for the implementation of the ternary function minimization.

A.DESIGN OF THE BASIC GATES

For designing the ternary multiplexers, we start with the basic gates and design of decoder which are building blocks for any circuitry. The basic building gates are Ternary Inverter, Ternary OR (TOR), Ternary AND (TAND) and Ternary XOR (TXOR) which is symbolized and represented as given in Fig and Equations respectively .

$$STI = \overline{X^1} = 2 - X$$

$$PTI, NTI = \overline{X^i} = \begin{cases} i \; if \; X \neq i \\ 2 - i \; if \; x \neq i \end{cases}$$

A general ternary inverter (GTI) is a basic unary operator with one input *x* and three outputs. Therefore, the implementation of ternary inverter requires three inverters namely negative ternary inverter (NTI), standard or simple ternary inverter (STI) and positive ternary inverter (PTI) forming an operator set that is complete in logic sense. This basic ternary inverter is used for constructing ternary AND/NAND, ternary OR/NOR etc. The VHDL Modeling is used for unary functions as well as shown the simulation results.

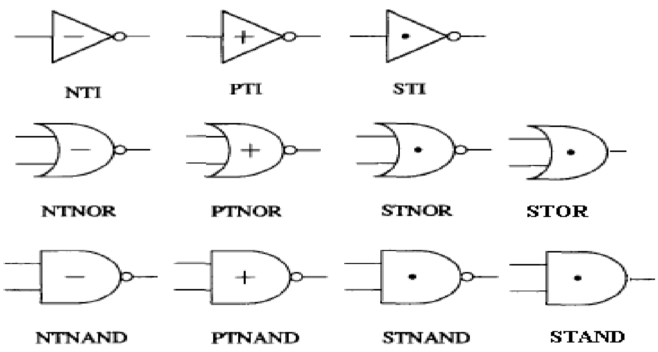


Fig3: Symbol for the basic gates

Table 4.1. Function Table For Ternary Inverter

X	$\overline{X^0}(NTI)$	$\overline{X^2}$(PTI)	$\overline{X^1}(STI)$
0	2	2	2
1	0	2	1
2	0	0	0

It can be proved that the complement or negation of literals (Xi) give the following observed which are helpful in reduction of ternary gates during implementation. The negation is defined and represented as shown

$$X^2 = \overrightarrow{X^{01}} \& X^{01} = \overrightarrow{X^2}$$
$$X^1 = \overrightarrow{X^{02}} \& X^{02} = \overrightarrow{X^1}$$
$$X^0 = \overrightarrow{X^{12}} \& X^{12} = \overrightarrow{X^0}$$
$$\vec{0} = 2 \& \vec{2} = 0$$

This observed result is used to show the reduction in gate count and also in simplification of ternary function. 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. It is represented by following equations and tabulated as shown in Table 3 and Fig.5 and 6. Logic Sum or TOR:

X1+X2+…+Xn=MAX(X1,X2,…,Xn)

Logic Product or TAND:

X1 . X2 ….. Xn =MIN(X1,X2,…,Xn)

Similarly, TNAND is

$$\overline{X1.X2.X3\ \ldots..Xn} = \min \overline{X1.X2.X3\ \ldots\ldots.Xn}$$

TNOR is

$$\overline{X1 + X2 + X3\ \ldots..+Xn} =$$
$$\min \overline{X1 + X2 + X3\ \ldots\ldots.+Xn}$$

Clearly (L, +,.) is a distributive lattice with zero element(0) and universal element(2). Thus the following laws hold for any x, y, z ∈L:

Idempotent: X+X=X X . X=X

Commutative: X+Y=Y+X X. Y=Y. X

Associative: (X+Y)+Z=X+(Y+Z)
X . (Y . Z)= (X . Y) . Z

Absorption: X+ X . Y =X X . (X+Y) = X

Distributive: X+Y . Z= (X+Y) . (X+Z)
X. (Y+Z)=X. Y+X.Z

It is evident that laws of identity elements, holds here.

X + 0 = X
X • 0 = 0
X + 2 = 2
X ≠ 0)
X+1 = 1(for cases X ≠ 2) & 2(for x=2) X • 2 = X
X • 1= 1(for cases DeMorgan's Theorem holds for ternary logic when the three types of inverters are used.)

$$\overline{(X+Y)^0} = \overline{X^0}.\overline{Y^0}$$
$$\overline{(X.Y)^0} = \overline{X^0} + \overline{Y^0}$$
$$\overline{(X+Y)^1} = \overline{X^1}.\overline{Y^1}$$
$$\overline{(X+Y)^1} = \overline{X^1} + \overline{Y^1}$$
$$\overline{(X+Y)^2} = \overline{X^2}.\overline{Y^2}$$
$$\overline{(X.Y)^2} = \overline{X^2} + \overline{Y^2}$$
$$\bar{\bar{X}}1^1 = X$$

Ternary Ex-OR function is mod-3 addition of ternary numbers. Modulo 3 sum is the sum of two integers ignoring the carry digits in the addition. Modulo-3 addition is an important function, since so many redundant code techniques use half-adding functions . It is denoted and expressed as given in Fig.7 and Table 4.

X + *Y* = *MODSUM* (*X* ,*Y*) = (*X* + *Y*)mod 3

Ternary functions of one or more variables may be represented in truth table or K-map form or algebraically in canonical form as a product of sum or sum of product. According to Expansion theorem [21], any ternary function f(Xl,X2,...,Xn) may be generated from (Xl, X2,...,Xn) by means of (+), (.) and the unary functions *X* 0 , *X* 1 , *X* 2 as given below:

f(Xl,X2, ..., Xn) = 2 • F2(Xl,X2 ..... Xn) + 1 •F1(Xi,X2, ...,x n) + 0 • F0(Xl,X 2 ..... Xn)
i.e., f = 2 • F2 + 1 • F1 + 0 • F0

where F_K equals 2, when value of the function f equals k, otherwise, it is 0. Applying equations to the above equation, the function may be represented by f = F2+1 • F1 for canonical Sum of Product form and f= F2 • (1+ F1) for canonical Product of Sum form

B. Design Of 3:1 Ternary Multiplexer

A ternary multiplexer is a combinational circuit that selects one of the 3n input lines based, on a set of n selection lines and directs it to a single output line. The design of 3x1 multiplexer (MUX) is as presented in Fig.11 and operates as given in Table 5. In this paper, 3x1 MUX is taken as a basic building block to explore the realization of 9x1 MUX, 27x1 MUX, Half Adder, Half Subtractor, Full Adder, Full Subtractor, Multiplier, 1-bit Comparator, 1-bit and 2-bit Position shifter ,Ripple carry Adder and position shifter.

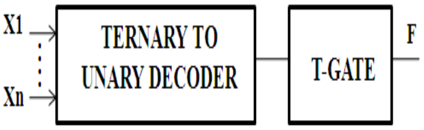


Fig 4: Implementation of the Ternary function

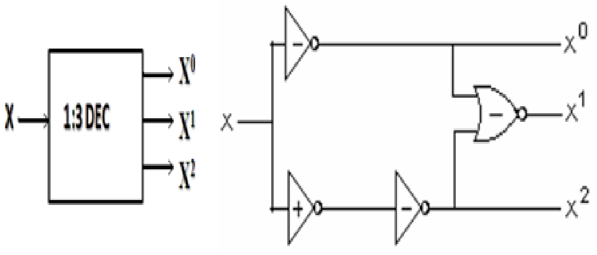


Fig 5: Block Diagram of 1:3 Decoder

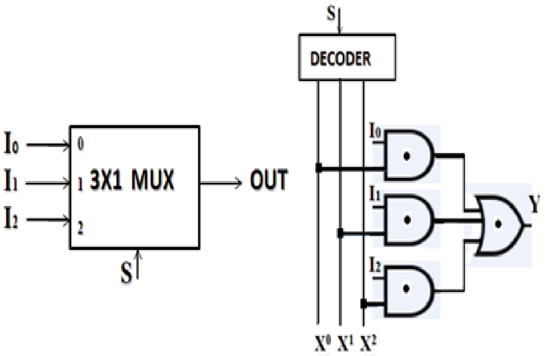


Fig 6: Block Diagram of 3:1 Multiplexer

C. Design Of 9:1 Mux Using 3:1 Mux

A 9x1 MUX is built using four 3x1 MUX as shown in Fig.11(a). A 9x1 MUX has 2 select lines to give an output among 9 inputs as given in Table VI and Fig.11(a) shows the simulation result for structural coding.

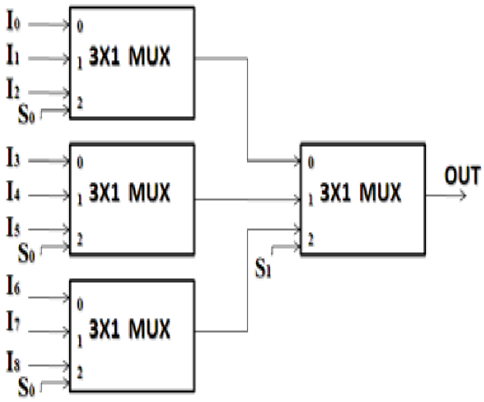


Fig 7: Block Diagram of 9:1 Multiplexer

Table 4.2 of 9:1 mux

S1	S0	OUT
0	0	I0
0	1	I1
0	2	I2
1	0	I3
1	1	I4
1	2	I5
2	0	I6
2	1	I7
2	2	I8

D.Design Of Half Adder And Half Subtractor Using 3:1 Mux

Ternary half adders (HA) is a circuit that adds two bits and generates a sum and carry using Modulo-3 addition. There are two inputs and two outputs and, consequently, two decoders are required. The half adder functions as shown in Table 7 can be realized as given in Fig.13(a-c) using 3x1 MUX.

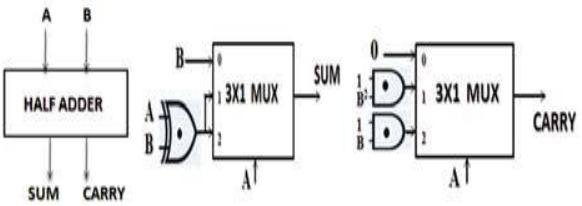


Fig 8: Block Diagram of Half Adder

Table 4.3: for half adder

A	B	SUM	CARRY
0	0	0	0
0	1	1	0
0	2	2	0
1	0	1	0
1	1	2	0
1	2	0	1
2	0	2	0
2	1	0	1
2	2	1	1

Ternary half subtractor (HS) is a circuit that will subtract one from the other number (i.e., A-B) and generate a difference and borrow using ternary logic. The half subtractor function is as shown in Table 8 and can be realized as given in Fig.14(a-c) using 3x1 mux.

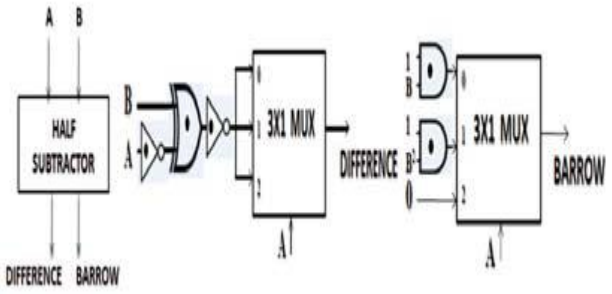


Fig 9: Block Diagram of Half Subtractor

Table 4.4: for half subtractor

A	*B*	*DIFF*	*BORROW*
0	*0*	*0*	*0*
0	*1*	*2*	1
0	*2*	*1*	*1*
1	*0*	*1*	*0*
1	*1*	*0*	*0*
1	*2*	*2*	*1*
2	*0*	*2*	*0*
2	*1*	*1*	*0*
2	*2*	*0*	*0*

E. DESIGN OF FULL ADDER AND FULL SUBTRACTOR USING 3:1 MUX

A full adder (FA) is a circuit that will add three bits and generates a sum and a carry. Fig. shows the design of full adder realization with 3x1mux .

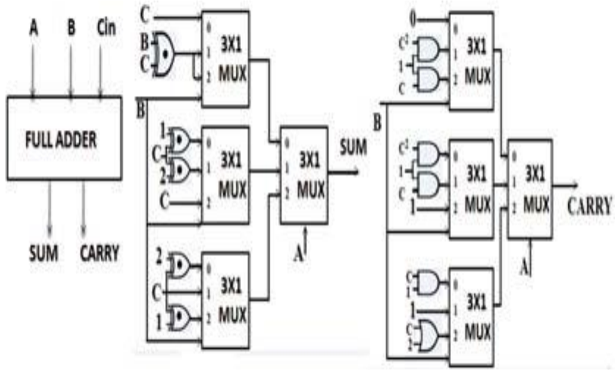


Fig 10: Block Diagram of Full Adder

Full subtractor (FS) is a circuit that will subtract three bits (i.e., (A-(B-Bin)), and generates a difference and a borrow. Fig. shows the design of full subtractor realization with 3x1mux .

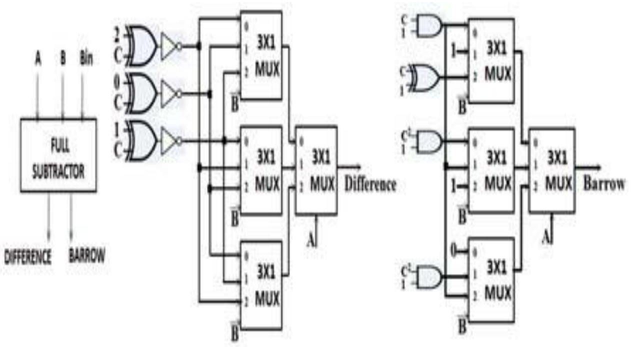


Fig 11: BLOCK Diagram of Full Subtractor

F. DESIGN OF 27:1 MUX USING 3:1 MUX

3x1 MUX is taken as a basic building block to explore the realization of 27x1 MUX in which we have the three selection lines.

G. DESIGN OF 1 BIT MULTIPLIER

A multiplier multiplies two bits and generates the product. Truth table for multiplier is shown in Table 11 and Fig.17(a-b). The gate count of multiplier is

fp+fc=7+2=9.
fp = A^2B^1 +A^1B^2 +1.(A^1B^1+A^2B^2)
fc = 1.(A^2B^2)

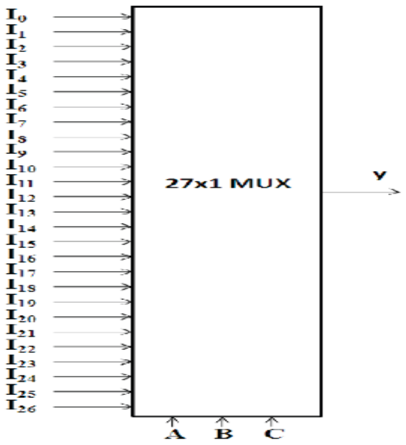


Fig 12: Block Diagram of 27:1 Mux

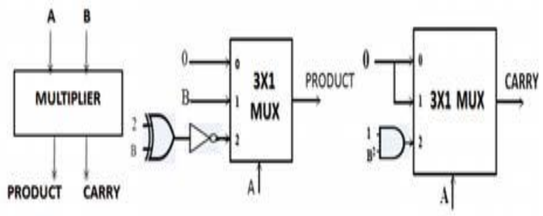


Fig 13: Block Diagram of 1 Bit Multiplier

H. Design Of 1 Bit Comparator

A magnitude comparator is a combinational circuit that compares two bits A & B and determines their relative magnitudes. The comparison of two bits is an operation that determines if one number is greater than, less than or equal to other number as Fig.19(a-b) and Table 12 shows the design of 1-bit comparator which gives

Y=f(A>B) when en=0, Y=f(A=B) when en=1and

Y=f(A<B) when en=2

f(A>B) = A^1B^0 +A^2B^0 +$A^{2B-}B^1$ = A^O B^0+A^2B^1

f(A=B) = A^0B^0 +A^1B^1 +A^2B^2

f(A<B) = A^0B^1+A^0B^2+A^1B^2 = A^0B^0+ A^1B^2

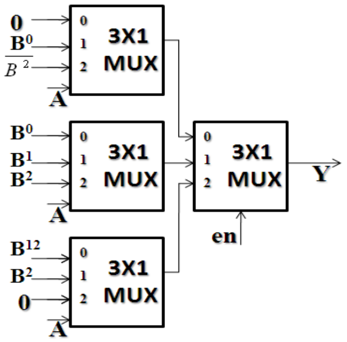


Fig 14: Block Diagram of 1 Bit Comparator

Fig4:5: for 1 Bit comparator

A	B	A>B	A=B	A<B
0	0	0	1	0
0	1	0	0	1
0	Z	0	0	1
1	0	1	0	0
1	1	0	1	0
1	Z	0	0	1
Z	0	1	0	0
Z	1	1	0	0
Z	Z	0	1	0

I.Design Of 1 & 2 Bit Position Shifter

The shifter circuit is as designed in Fig.21 and Table 15 which shifts the bits of an n input vector by 1–bit position to the right. It fills the vacant position on the left side with zero. If S=0, the input is loaded as output which is said to work in parallel mode, if S=1, the input is shifted by one bit position and if S =2, the input is shifted by two-bit positions padded by zeros in the left position. The more versatile shifter circuit will be able to shift by more bit positions at a time. If the bits that are shifted out are placed into the vacated positions on the left, then the circuit effectively rotates the bits of the input vector by a specified number of bit positions

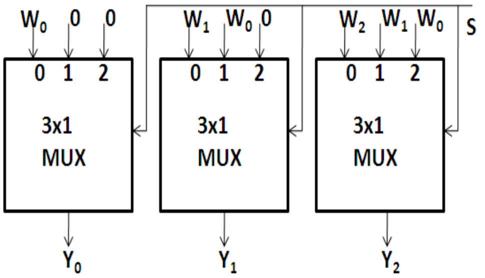


Fig 15: Block Diagram of 1 & 2 Bit Position Shifter

J.Design Of Ripple Carry Adder

Two binary words, each of n bits, can be added using ripple carry adder as shown in Fig.21 (a). The carry input is connected to the least significant bit and the carry output of each full adder is connected to carry input of the next most significant FA. It is a typical example of iterative circuit which is slow, since in the worst case a carry must propagate from least significant FA to the most significant one.

K.Design Of Carry Save Adder Using 3:1 Mux

Carry save adder computes the sum of 3 or more n-bit numbers which gives the sequence of partial sum and carry bits.

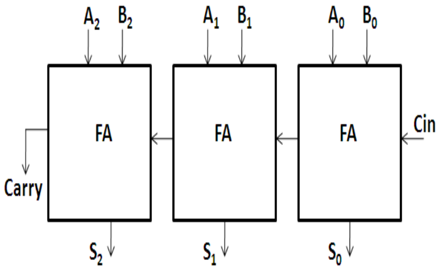


Fig 16: Block Diagram of Ripple carry Adder

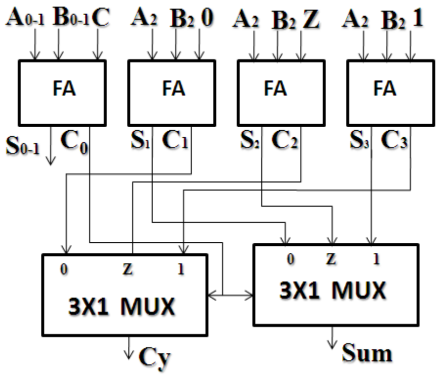
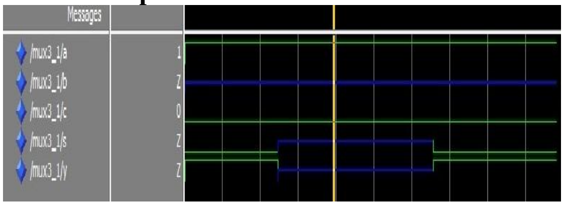


Fig 17: Block Diagram of Carry save Adder

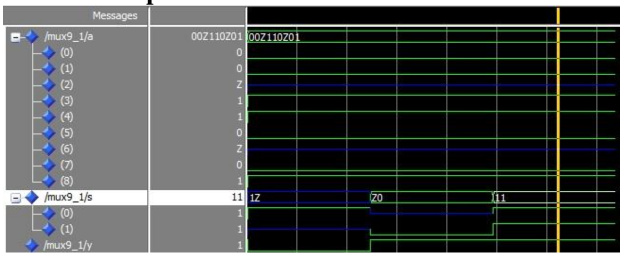
5.Simulation Results:

5.1 3:1 multiplexer:



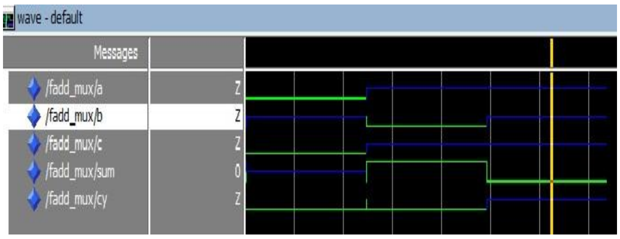
Messages
/mux3_1/a 1
/mux3_1/b Z
/mux3_1/c 0
/mux3_1/s Z
/mux3_1/y Z

5.2 9:1 multiplexer:



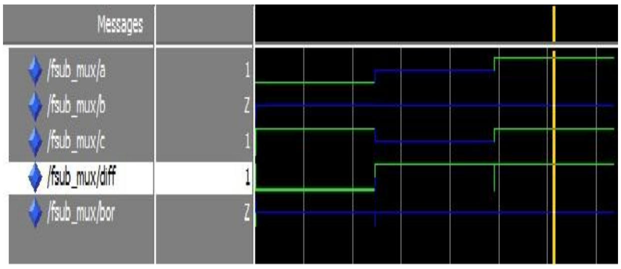
Messages
/mux9_1/a 00Z110Z01 00Z110Z01
(0) 0
(1) 0
(2) Z
(3) 1
(4) 1
(5) 0
(6) Z
(7) 0
(8) 1
/mux9_1/s 11 1Z Z0 11
(0) 1
(1) 1
/mux9_1/y 1

5.3 full adder:



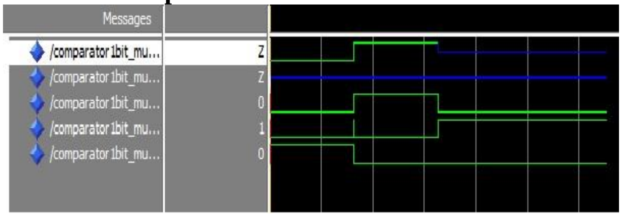
wave - default
Messages
/fadd_mux/a Z
/fadd_mux/b Z
/fadd_mux/c Z
/fadd_mux/sum 0
/fadd_mux/cy Z

5.4 full subtractor:



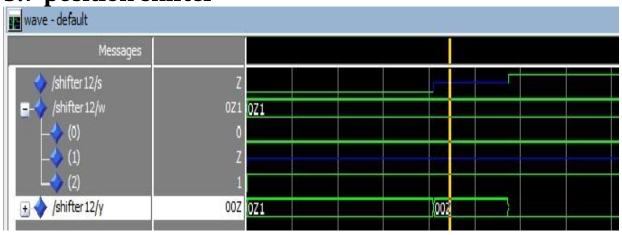
Messages
/fsub_mux/a 1
/fsub_mux/b Z
/fsub_mux/c 1
/fsub_mux/diff 1
/fsub_mux/bor Z

5.6 1-bit multiplier:



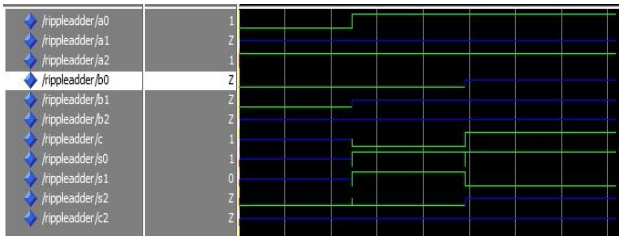
Messages
/comparator1bit_mu... Z
/comparator1bit_mu... Z
/comparator1bit_mu... 0
/comparator1bit_mu... 1
/comparator1bit_mu... 0

5.7 position shifter



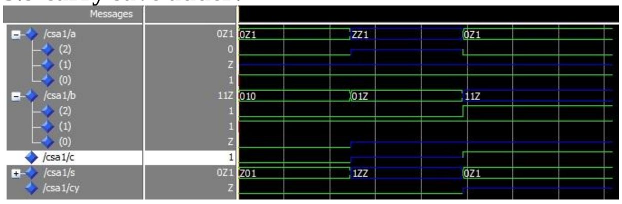
wave - default
Messages
/shifter12/s Z
/shifter12/w 0Z1 0Z1
(0) 0
(1) Z
(2) 1
/shifter12/y 00Z 0Z1 00Z

5.8 1-bit comparator:



/rippleadder/a0 1
/rippleadder/a1 Z
/rippleadder/a2 1
/rippleadder/b0 Z
/rippleadder/b1 Z
/rippleadder/b2 Z
/rippleadder/c 1
/rippleadder/s0 1
/rippleadder/s1 0
/rippleadder/s2 Z
/rippleadder/c2 Z

5.9 carry save adder:



6. Conclusion

The ternary logic is a promising alternative to the conventional binary logic design technique. The ternary and binary logic gates can be used to take advantage of their respective merits, to improve performance in terms of computation speed and power consumption. Expanding the existing logic levels to higher levels higher processing rates could be achieved. In this paper, we have implemented a technique to reduce the gate count. The main advantages of the ternary logic is that it reduces the power 50% as compare to the ternary logic and also reduces the computation steps, chip area and chip delay. Thus, ternary logic gate design technique combined with the conventional binary logic gate design technique also provides an excellent speed and power consumption characteristics in data path circuit such as full adder and multiplier. VHDL simulator has been used to simulate MVL Systems which provide enough information to verify functionality and timing specifications.

References

[1] http://www.embedded.com/columns/netcentricview/13100886?_requestid=133447

[2] http://en.wikipedia.org/wiki/Ternary_logic

[3] 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.

[4] Sheng Lin et al. (2009): CNTFET-Based Design of Ternary Logic Gates and Arithmetic Circuits, *IEEE Trans. On Nanotechnology*, Vol. PP, Issue.99, pp.1-1.

[5] S.L.Hurst. (1984): Multivalued logic - Its status and its future, *IEEE Trans .on Computers*, vol. C-33, pp. 1160-1179.

[6] Raymond E.Miller. (1966): Switching Theory, Vol. I, John Wiley & Sons, pp.8-9.

[7] D. Venkat Reddy et. al (2008): Sequential Circuits In The Framework Of (2n+1)-ary Discrete Logic, *IJCSNS International Journal of Computer Science and Network Security*, Vol.8 No.7, July, pp.175-181.

[8] Marek Perkowski (2006): Introduction to multivalued logic.Available as: http://web.cecs.pdx.edu/~mperkows/temp/JULY/2006.Introduction-to-MV-logic.ppt

[9] A.P. Dhande et. al. (2005): Design And Implementation Of 2 Bit Ternary ALU Slice, 3rd International Conference, Sciences of Electronic (SETIT), *IEEE Transc.*, pp.1-11.

[10] K.C. Smith. (1988): Multiple-Valued Logic, A Tutorial and Appreciation, Survey & Tutorial Series, *IEEE Transc*. *In computers*,Vol.21, Issue.4, pp.17-27.

[11] H.T. Mouftah. (1975): Three-valued logic and its implementation with COS/MOS integrated circuits, D.Sc. thesis, Laval University, Quebec, Canada.

[12] H.T. Mouftah. (1975): A Study On The Implementation Of Three-valued Logic, University of Toronto, Toronto, Ontario, Canada, pp.123-126., Monograph .

[13] D. I. Porat, “Three-valued digital systems”, *PROC. IEE Inst. Elect Eng*, Vol. 116, No. 6, JUNE 1969, pp.947-954.

[14] R. P. Hallworth and F. G. Heath, “Semiconductor Circuits For Ternary Logic”, The Institution of Electrical Engineers, Monograph No. 482 E, *IEEE trans.*,VOL. 109, PART C.,Nov. 1961, pp.219-225.

[15] Richard F.Tinder, “Engineering Digital Design”, Second Edition, Academic Presss,2000.

[16] Neil H.E.Weste,David Harris, Ayan Banerjee, “CMOS VLSI Design”,A circuit and systems perspective, third Edition, pearon Education,2007.

[17] John F.Wakerly, “Digital Design, principles & practices, third edition, prentice Hall of India Private Limited (EEE),2005.

[18] Nazeih M.Botros, “HDL Programming VHDL and verilog”,dreamtech press, 2006.

[19] Stephen Brown and Zvonko Vranesic, “Fundamentals of Digital Logic with Verilog Design”, second edition, Tata McGraw-Hill Publishing Company Limited, New Delhi,2008.