International Journal of Research in Advanced Engineering and Technology
Online ISSN: 2455-0876
www.engineersjournals.com
Volume 1; Issue 2; November 2015; Page No. 82-88

Scheme of quaternary logic for reducing control lines in multiplexer

[1] P. Karthika, [2] Solomon deva doss

[1] Student, Applied Electronics Department, Infant Jesus College of Engineering, TN, India

[2] Associate Professor, ECE Department, Infant Jesus College of Engineering, TN, India

Abstract
Designer face challenges in interconnection due to large number of components. Interconnection and power consumption plays essential position on this paper. The drawbacks of previous paper tells approximately the interconnection occupies greater area and also consume more power. The Binary logic circuits layout is constrained by using the requirement of number of interconnections which increases the chip area, delay and power consumption with boom in logic. So we propose new techniques that going to be reducing interconnection of the circuit as well as power consumptions. Multiple-valued logic can lessen the desired interconnections. On this paper recommended a Quaternary to Binary decoder to reduce the control lines in multiplexer is one of the vital parts of the processing element and therefore it has a focal point of research. Therefore design of adders through multiplexer using multi valued logic can show to be very beneficial. So we introduce a full adder prototype based totally at the designed LUT. It's capable of operate at 195MHz at the same time as power consuming 0.81mW using Xilinx14.2 and getting the simulation results through the Modelsim. The received consequences give an explanation for the perfect quaternary operation and make certain the power efficiency of the new techniques.

Keywords: Multiple-valued logic (MVL), quaternary logic, lookup table (LUT).

Introduction

In a binary VLSI circuit, the processing transistors employ only 10% of the chip but interconnect accounts for 70% of the chip's area and the last 20 % is set apart of the insulation [1]. The requirement of the interconnection is restricted in the binary logic circuits. When increasing the number of connections greatly, it will increase the size of the Chip and also reduce the speed of the circuit. When transistors shrink with new technology, the characteristics of the circuit namely, area, power or delay, decreases in terms to interconnect. The new approach is to solve the interconnection problems using multiple-valued logic (MVL) inside the VLSI chip. Multiple-valued logic (MVL) has accepted increased attention in the last decades since the possibility to represent the information with more than two discrete levels. Representing data in a MVL system is producing more favorable end result than the binary based representation, because the number of interconnections can be appreciably decreased. Multiple-value logic carries more information on a unmarried line. So it reduces the complexity of interconnection and delay. Reducing interconnect be the manner to a direct reduction of line capacitances and the area. It additionally minimizes the required number of gates. In this approach has been design in multivalve system and binary valued output in the circuit. This design is depends on the quaternary voltages for the multi- valued logic. We present a QLUT in an FPGA context. Practically used digital systems are dominantly binary ones. Reasons and possibilities for application and implementation of digital systems that use logic basis greater than 2 (i.e.), MVL are getting real and almost viable with fast development of VLSI technologies. The best realistic hobby exists for research and implementation of ternary (logic basis 3) and quaternary (logic basis 4) MV circuits and systems. There are many well-known benefits of quaternary logic systems and circuits comparing with the binary ones are, greater speed of logic and arithmetic operations, greater density of memorized information, higher utilization of transmission lines and paths, decreasing of interconnections complexity and area, decreasing of pin number of integrated circuits and printed boards, opportunities for less difficult testing of digital systems.

Multi-valued systems are usually proposed to offer advantages by using lowering range of data interconnect lines and processing components. Such logic circuits can represent numbers with fewer bits than binary, e.g. the decimal number 156 is represented as 10011100 in binary and 2130 in quaternary. Multiple-valued logic can provide progressed circuit interconnections, reduced chip area and increased bus efficiency, since more logic levels are used per line as compared to conventional binary logic. Quaternary logic is quite possible because the implementations can be designed using available circuitry, and no additional special components are required. For this reason quaternary radix can be selected to realize the logic circuits.

The quaternary logic we are proposing here consists of quaternary states or variables and operators. The quaternary states are 0 (absolute low), 1 (medium low), 2 (medium high) and 3 (absolute high. When expressed as a number, a single quaternary digit is called a qudit. Since a quaternary variable (Q) is capable to carry twice as lots as information a binary variable (B), we have the following relation:

$$|Q| = 2 \times |B|. \quad (1)$$

Table 1: Binary Equivalents for Quaternary Logic

Quaternary logic	Binary logic	
A	X	Y
0_4	0_2	0_2
1_4	0_2	1_2
2_4	1_2	0_2
3_4	1_2	1_2

Further to the reduction in interconnections, Quaternary logic also offers a possibility of increasing the functional complexity per unit silicon area, producing circuits that have performance similar to the equivalent binary circuits.

In a novel quaternary logic has been proposed that's close to Boolean algebra. It extends binary functions in quaternary, on the identical time contains some new functions of its very own. Extending Boolean algebra makes this logic system ideal for replacing binary logic as its logic blocks are compatible with their binary counterparts.

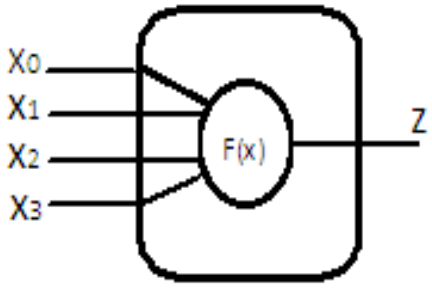
This paper is organized as follows. In Section II, we review the basic concepts and an overview of binary and quaternary LUTs. Then, in Section III Literature review. In section IV, the quaternary LUT (QLUT) topologies with the circuits proposed are discussed. In Section V, an adder prototype based on the proposed QLUT is developed. Section VI presents the simulation results of the proposed design. Finally, Section VII summarizes and concludes this paper.

II. Overview of Binary and Quaternary Luts

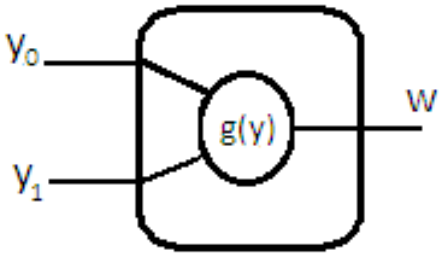
Standard Lookup Tables are essentially reminiscences, which implement a given logic function. A LUT is an array indexing operator, in which the output is mapped by the input based on the configuration memory. The configuration values are initially stored in the LUT configuration memory, and according to the input, the logic value in the addressed position is assigned to the output. By means of properly programming the LUT configuration memory, the LUT can put into effect any logic function with the given quantity of inputs and outputs, making it very sensible to implement reconfigurable hardware, including FPGAs. Values are initially stored in the lookup table structure, and once inputs are applied, the logic value in the addressed position is assigned to the output. The capacity of a LUT $|c|$ is given by way of

$$|c| = nb^k \quad (2)$$

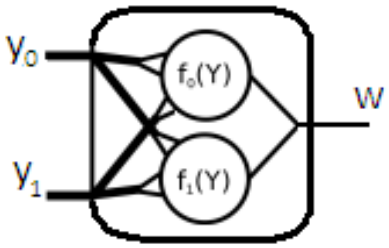
Wherein n is the number of outputs, k is the number of inputs and b is the number of logic values. For example, a 4-input binary lookup table with one output is able to store $1\times 2^4 = 16$ Boolean values. For the cause of this work, only 1-output LUTs (n = 1) are mentioned on this paper.



(a) 4-input BLUT.



(b) 2-input QLUT.



(C) QLUT Function

Fig 1: Binary and quaternary lookup tables

A binary function implemented by a Binary Lookup Table (BLUT) is defined as f:$B^k \rightarrow B$. The total number of different functions $|F|$ that can be implemented in a BLUT with k input variables is given by

$$|F| = b^{|c|} \quad (3)$$

where b = $|B|$ (i.e. b = 2 in the binary case).

Quaternary functions are basically generalizations of binary functions. A quaternary function implemented by a quaternary lookup table (QLUT) is defined as g: $Q^k \rightarrow Q$, over a set of quaternary variables Y = (Y_0; _ _ _; Y_i; _ _ _ ; Y_{k-1}), where the values of a variable Y_i, as the values of the function g(Y), can be in Q= {0, 1, 2 3}. As in the binary case, the number of possible function in QLUTs is given by (3), where b = 4. Figure 1b illustrates a 2-input quaternary function implemented in a QLUT. Note that the function g(Y) performs exactly the same function as the two binary BLUTs, f0(Y) and f1(Y), as depicted in Figure 1c, where f0 represents the least significant Boolean values and f1 represents the most significant ones. Since a quaternary variable y is capable of representing twice as much information as a binary variable x, we consider the cardinality of $|Q| = 2 \times |B|$ in our experiments. In other words, we count on that two binary variables may be grouped which will represent a quaternary variable. Such that technique pursuits at reducing both the whole number of connections and the number of gates.

III. Literature Review

The proposed work focused on reducing the interconnection in circuits. Diogo Brito, Taimur G. Rabuske, Jorge R. Fernandes says the change of logic can help in reducing the interconnections and also the power consumption. The use of look up table in a new concept to reduce interconnection [2]. The power optimization is the main need of the world. Multiple valued logics are long forgotten method but today's technologies have made it a possible thing says Shweta Hajare, Dr.P.K.Dakhole this increases the data density, increased computational ability, reduced dynamic power dissipation[3]. Quaternary logic has shown to be a promising alternative for

implementing FPGAs, since the quaternary circuits can reduce the circuits' cost and at the same time reduce its power consumption said by Marcus Ritt, Carlos Arthur Lang Lisboa, and Luigi Carro[4]. Field Programmable Gate Arrays (FPGAs) are very convenient and flexible hardware platforms for the implementation of these systems, helping designers to cope with one of the more demanding requirements currently imposed on the industry: time to market. Their flexibility comes at a price: in order to allow the many different interconnection schemes of the increasing number of devices in a single FPGA, an enormous area of the circuit must be used to implement all the required switches and wires. As technology evolves according to Moore's Law, providing ever smaller, faster, and lower voltage devices, the amount of interconnections inside a single chip is increasing significantly by W. J. Dally [5]. When considering FPGAs, this problem becomes even more critical, since the huge amount of interconnections impacts not only the circuit delay, but also the power consumption and area by A. Singh and M. Marek-Sadowska [6]. Multiple-valued logic (MVL) has received increased attention in the last decades because of the possibility to represent the information with more than two discrete levels. As a consequence, there is a possibility to increase the amount of information transferred using a single wire by P. M. Kelly [7]. In a VLSI circuit, interconnection plays the dominant role in every part of the circuit nearly 70 percent of the area depends on interconnection, 20 percent of the area depends on insulation, and remaining 10 percent to devices. Ilanthendral, Sathyavathi says that the binary logic is limited due to interconnect which occupies a large area on a VLSI chip. The quaternary-valued logic circuits have been explored over multi-valued logic due to the following reasoning. An approach to mitigate the impact of interconnections is to use multiple-valued logic (MVL), hence, more information can be carried in each wire, reducing the routing network. Therefore, a single wire carrying a signal with N logic levels can replace [log2N] wires carrying binary signals. Reducing interconnect be the way to a direct reduction of line capacitances and the area. Therefore, this results in increasing the maximum operation frequency and also reducing the power consumption [8].

IV. Design of Proposed Qlut

The proposed 2-input and 1-output QLUT is shown in Fig. 2. For this given QLUT complexity, 16 quaternary configuration inputs are necessary, one for each feasible aggregate of the two quaternary inputs. The configuration word defines the reconfigurable quaternary function. In practice, the input signals are used to select which one of the configuration inputs is attached to the output. The principle concept of the paper proposed is to reduce the interconnection present in the existing circuit.

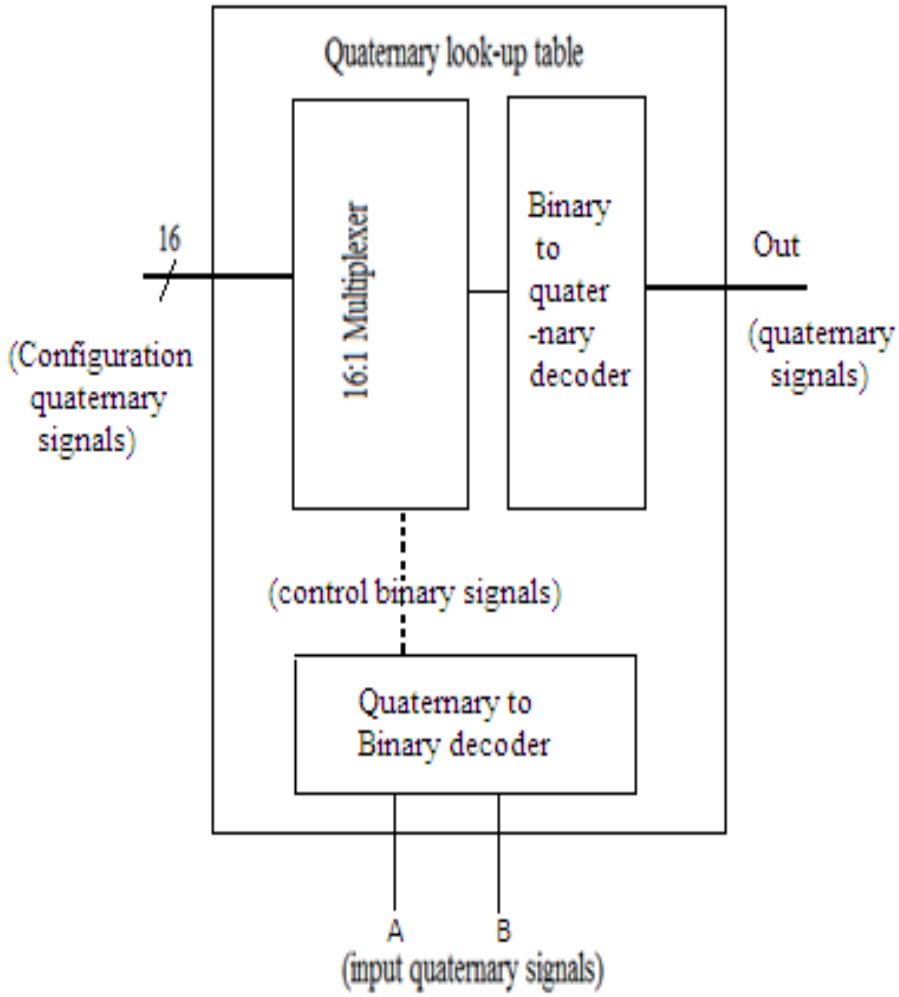


Fig 2: Proposed quaternary LUT overview

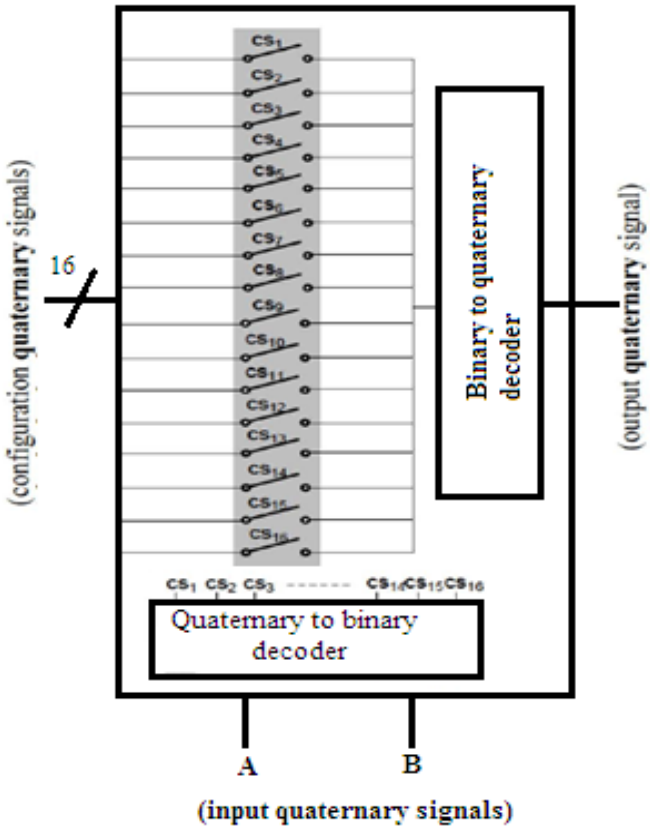


Fig 3: Proposed quaternary LUT Expanded

The combinatorial block logic capabilities will assist in understanding the function and switch operation inside the circuit proposed. The subsequent table.2 gives the better readability of the signal fetching and switches operations. The proposed QLUT consists of two main blocks: a 16-1 multiplexer using an array of switches, that establishes a low-resistance path between one configuration input and the output according to the input values; and a quaternary-to-binary decoder, including of a 2-bit analog-to-digital (ADC) frontend followed by combinational logic used to generate the control signals feeding the multiplexer. These blocks are described within the following sections.

Table II: Combinatorial Block Logic Functions

A	B	X_A	Y_A	X_B	Y_B	C_s (activebit)
0_4	0_4	0_2	0_2	0_2	0_2	Cs(1)
1_4	0_4	0_2	0_2	0_2	1_2	Cs(2)
2_4	0_4	0_2	0_2	1_2	0_2	Cs(3)
3_4	0_4	0_2	0_2	1_2	1_2	Cs(4)
0_4	1_4	0_2	1_2	0_2	0_2	Cs(5)
1_4	1_4	0_2	1_2	0_2	1_2	Cs(6)
2_4	1_4	0_2	1_2	1_2	0_2	Cs(7)
3_4	1_4	0_2	1_2	1_2	1_2	Cs(8)
0_4	2_4	1_2	0_2	0_2	0_2	Cs(9)
1_4	2_4	1_2	0_2	0_2	1_2	Cs(10)
2_4	2_4	1_2	0_2	1_2	0_2	Cs(11)
3_4	2_4	1_2	0_2	1_2	1_2	Cs(12)
0_4	3_4	1_2	1_2	0_2	0_2	Cs(13)
1_4	3_4	1_2	1_2	0_2	1_2	Cs(14)
2_4	3_4	1_2	1_2	1_2	0_2	Cs(15)
3_4	3_4	1_2	1_2	1_2	1_2	Cs(16)

A) 16X1 Multiplexer

The mux used here's a 16x1, so it has 16 input pins and 1 output pin. This mux has usually four control lines in binary logic. Through the usage of the quaternary logic, the control lines are reduced to two. This is done with the brand new quaternary to binary decoder part introduced in the lookup table. The decoder reduces the fetch line of the mux from four to two. Why do we choose mux for this purpose is we can design a mux into an adder, subtracter, and divider or can use it for any other arithmetic or logical functions. The look up table will guide the mux to do various operations. The mux used has 16 pins, so we have switches that need to be operated very speedy to get the correct output. In the proposed method, here we use the direct technique to do the work in a single clock cycle. In the present technique four clock cycles are used.

B) Quaternary-To-Binary Decoder

The 2-bit quaternary-to-binary decoder allows the use of a single row of switches to drive the input configuration signals to the output of the QLUT. To do so, it is necessary to generate 16 control signals, to be applied in the clk inputs of each switch. These switches are employed to connect one quaternary configuration input to the output. To generate the required control signals, the quaternary variables are decoded into binary, allowing the use of binary logic gates. Thus, an ADC frontend is necessary, considering the analog nature of the quaternary signals. The decoder is the essential part to reduce the control pins from four to two. It will fetch the quaternary inputs and decode it into binary so as to perform the regular operation of the system designed. The designed system can be an adder or subtracted or divider etc. Here in the proposed

design, an adder circuit is used to understand the quaternary logic and its other related benefits in the system.

C) Binary-To-Quaternary Decoder

The binary-to-quaternary decoder is used at the output side. The decoder is used because the internal operations are done in binary and the actual input/output are quaternary. So we need a convertor, that is, a decoder to convert the output generated from binary to quaternary. The internal functionality are similar and just the reverse of quaternary to binary decoder.

V. Introducing Adders Prototype for Qlut

The adder is just a prototype to explain the logical function of quaternary logic. The adder designed using mux is used. The half and full adder circuit are presented here. The truth table and results obtained are discussed further.

In quaternary logic, addition can be performed in many ways. Numbers in quaternary logic can be directly added or numbers in quaternary logic can be converted to binary logic and addition can be performed in binary logic. Binary results of addition can be displayed in quaternary logic after conversion. Hence quaternary to binary converter is required in the beginning. Binary to quaternary converter is used to display the result in quaternary logic. Here fig.4 and fig.5 are the quaternary half and full adder design respectively. Both the design shows how the Quaternary logic is useful in multiplexer. In future we can implement this logic in anywhere. It is easy to understand how it will be converted.

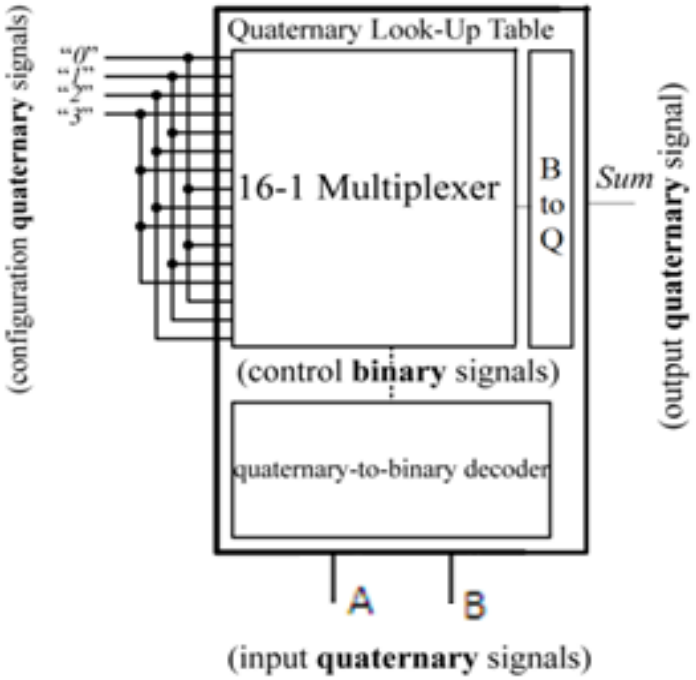


Fig 4: Quaternary half adder

Table II: Truth Table of Quaternary Half Adder

A	B	Sum	In Binary	
			X_{sum}	Y_{sum}
0_4	0_4	0_4	0_2	0_2
1_4	0_4	1_4	0_2	1_2
2_4	0_4	2_4	1_2	0_2
3_4	0_4	3_4	1_2	1_2
0_4	1_4	1_4	0_2	1_2
1_4	1_4	2_4	1_2	0_2
2_4	1_4	3_4	1_2	1_2
3_4	1_4	0_4	0_2	0_2
0_4	2_4	2_4	1_2	0_2
1_4	2_4	3_4	1_2	1_2
2_4	2_4	0_4	0_2	0_2
3_4	2_4	1_4	0_2	1_2
0_4	3_4	3_4	1_2	1_2
1_4	3_4	0_4	0_2	0_2
2_4	3_4	1_4	0_2	1_2
3_4	3_4	2_4	1_2	0_2

In this half adder, two inputs of the Quaternary logic added and produced the same quaternary values but interiorly binary operation happened in it. Refer table 1 to convert binary values into quaternary values and without giving any carry in values to

the binary side operation. Similar operation present in the full adder but the one difference is giving carry values which is 0 and 1. Changes are shown in the fig. 5 in half adder we need only one block but ion full adder we need two QLUT blocks which means two multiplexer greatly having only one carry in line.

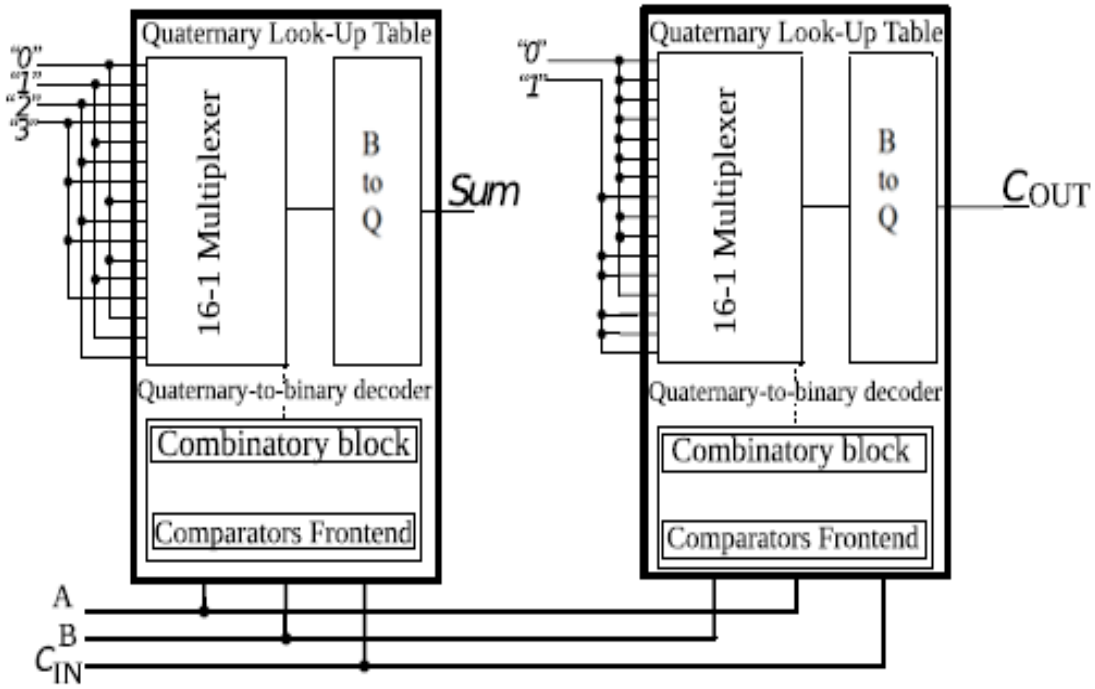
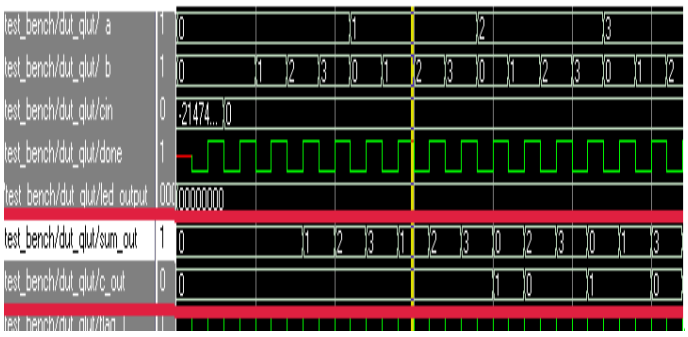


Fig 5: Quaternary full adder

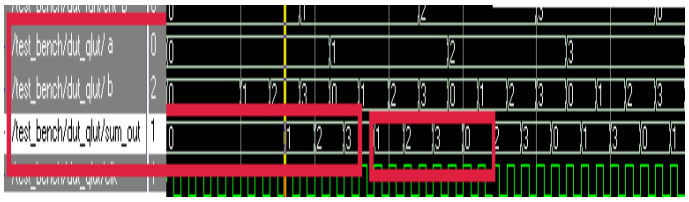
A and B lines are combination of Quaternary logic values. The two lines values producing the 16 outputs. Table 3 shows the clear view of the full adder. Carry in 0 and 1 producing Sum and Cout values.

Table III: Truth Table of Quaternary Full Adder

A	B	$C_{IN} = 0_4$		$C_{IN} = 1_4$	
		Sum	C_{out}	Sum	C_{out}
0_4	0_4	0_4	0_4	1_4	0_4
1_4	0_4	1_4	0_4	2_4	0_4
2_4	0_4	2_4	0_4	3_4	0_4
3_4	0_4	3_4	0_4	0_4	1_4
0_4	1_4	1_4	0_4	2_4	0_4
1_4	1_4	2_4	0_4	3_4	0_4
2_4	1_4	3_4	0_4	0_4	1_4
3_4	1_4	0_4	1_4	1_4	1_4
0_4	2_4	2_4	0_4	3_4	0_4
1_4	2_4	3_4	0_4	0_4	1_4
2_4	2_4	0_4	1_4	1_4	1_4
3_4	2_4	1_4	1_4	2_4	1_4
0_4	3_4	3_4	0_4	0_4	1_4
1_4	3_4	0_4	1_4	1_4	1_4
2_4	3_4	1_4	1_4	2_4	1_4
3_4	3_4	2_4	1_4	3_4	1_4



(a) Full adder with Carry 0 and 1



(b) Half adder

Fig 6: Simulation result of proposed full adder circuit

VI. Simulation Results

The simulation results obtained for half adder and full adder using quaternary logic is discussed below.

The A and B are the quaternary inputs given to the circuit and sum and carry are the outputs obtained. The half adder design does not have a carry as input, so for half adder we have only two inputs. In this truth table, the base 4 are quaternary logic and the base 2 are binary logic.

For a full adder, carry can be a input to the next operation. A carry can be either 0 or 1. So we have three inputs for full adder, that is, the output generated will be of two kind, say sum and carry with carry 0 and the same with carry 1. The full adder truth table is discussed above. The simulation results help to verify the correct operation of the design. The table III is the truth table of full adder with carry0 and carry1 respectively. The

truth table is based on quaternary logic and hence the inputs are mentioned as A and B. Here in the truth table, S_{um} and Cout are the results when carry is 0 and 1 respectively. These truth tables and simulation output in fig.6 are based on proposed quaternary full adder circuits.

VII. Conclusion

On this paper, we have reported an progressive QLUT layout that may be used for multiple valued logic. The proposed design is illustrated with the half and full adder design. The proposed design can be prolonged to many arithmetic and logical circuits. From the simulation results obtained, the presented design is a valid solution to reduce the interconnections impact, without increasing power consumption or dropping performance. In future this idea, which is beneficial to done in large scale, depending upon its effective implementation as it shows in the seeds to develop various real projects

Acknowledgement

We wish to thank Almighty lord and Associate Professor Mr. Solomon deva doss, Department of ECE, Infant Jesus college of Engineering, Thirunelveli, Tamilnadu for their encouragement and support on our project.

References

1. Diogo Brito, Jorge Fernandes, Paulo Flores, Jose Monteiro, Taimur Rabuske. Senior member IEEE Quaternary Logic Look up Table in standard CMOS 2015 IEEE transaction on very large scale integration system.
2. Nagamani AN, Nischai S. PES institute of Technology, Karnataka, Quaternary High Performance Arithmetic Logic Unit Design Euromica Conference, 2011.
3. Marcus Ritt. Carlos Arthur Lang Lisboa, Luigi Carro, Cristiano Lizzari, A cost effective Technique for mapping BLUTs to QLUTs in FPGAs IEEE conference, 2010.
4. Prashant Shende Y, Dr. RV Kshirsagar. Quaternary Multiplier using VHDL international journal paper, 2013.
5. Jeong Beom Kim. Dept. of Elec. Engg. KangwonNatioinal university, Chunchon, south korea, Area Efficient multiplier using current mode Quaternary logic Technique 2010, 10th IEEE conference.
6. Agarwal S, Pavankumar VK, Yokesh R. Energy-efficient high performance circuits for arithmetic units IEEE International Conference, 2008.
7. Tang AJJ, Reyes JA. Comparative analysis of low power multiplier architectures IEEE Symposium, 2011.
8. Mr. J lanthendral sathyavathi NS. A Novel Voltage-Mode LUT Using Clock Boosting Technique in Standard Cmos 2014; 2(4):(12-20).
9. DA Silva R, Lazzari C, Boudinov H, Carro L. CMOS voltage mode quaternary look-up tables for multi-valued FPGAs, Microelectron. J., 2009; 40(10):1466-1470.
10. Lazzari C, Fernandes J, Flores P, Monteiro J. An efficient low power multiple-value look-up table targeting quaternary FPGAs, in Integrated Circuit and System Design. Power and Timing Modeling, Optimization, and Simulation (Lecture Notes in Computer Science), R. van Leuken and G. Sicard, Eds., New York, NY, USA: Springer-Verlag, 2011, 84-93.
11. JH Anderson, Najm FN. Power estimation techniques for FPGAs, IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 2004; 12(10):1015-1027.
12. Uyemura J. Circuit Design for CMOS VLSI. Boston, MA, USA: Kluwer Academic Publishers, 1992.
13. Rabuske TG, Rodrigues CR, Nooshabadi S. A 5MSps 8-bit SAR ADC with single-ended or differential input, Microelectron. J., 2012; 43(10):680-686.