IJETAE
EXPLORING RESEARCH AND INNOVATIONS

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 7, Issue 8, August 2017)

Design of 32-bit Reversible Booth Multiplier and Area Optimization using Genetic Algorithm

Nasreen[1], Kamal Prakash Pandey[2], D K Tripathi[3]

[1]M.Tech Student, [2,3]Associate Professor, SIET Allahabad

***Abstract* – Multipliers circuits always present issues on the hardware complexity and implementation. The Hardware being used presents the challenges in the speed and further complexities. The area of implementation and speed in these circuit presents the trade-offs in the final design. If speed of the circuit is to be increased, the area will be more and similarly vice versa. Thus, the ideal design for the multipliers should be faster and with lesser area consumption. Also, the power dissipation is the area of concern for these designs. Thus, reversible logic gates are to be used for the design and implementation of the circuit. In this proposed work, we have presented the design and implementation of the reversible booth multiplier. Also, the floor planning technique has been done for optimized area of implementation. The Booth multiplier design has been presented. The simulation results and the RTL structures have been presented. Thus, the optimality of the genetic algorithm has been shown.**

***Keywords*: Booth Multiplier, Reversible logic gates, Booth algorithm**

I. INTRODUCTION

The reversible logic circuits are very important in design and development for the VLSI due to low power consumption as compared to the conventional reversible designs. The reversible circuits do not erase the loss of the information bits. Thus, there is no loss of the information after all. An n-input n-output function (gate) is called reversible if and only if it maps each input instance to a unique output instance. The only possible structure for a reversible network is the cascade of reversible gates [1][2]. Several reversible gates have been proposed in recent times, where the synthesis of reversible circuits varies from the design of conventional irreversible circuits. Two restrictions are added for reversible networks, namely fan-outs and back-feeds [1].

Booth‘s Algorithm is an intelligent algorithm for multiplication both signed numbers and unsigned numbers. It presents the feature to both add and subtract there are multiple ways to compute a product [5]. Booth‘s algorithm is a multiplication algorithm that utilizes two‘s complement notation of signed binary numbers for multiplication [9].

Earlier multiplication was in general implemented via sequence of addition then subtraction, and multiple shifting operations. Multiplication can be well thought-out as a series of repeated additions. The number which is to be added is known as the multiplicand, and the number of times it is added is known as the multiplier, and the result we get is the multiplication result. After Each step of addition a partial product is generated.

When the operands are integers, the product in general is twice the length of operands in order to protect the information content. This repetitive addition method that is recommended by the arithmetic definition is slow as it is always replaced by an algorithm that makes use of positional depiction.

Floor planning can be seen as one of the most important stages in the designing or fabrication of VLSI circuits is. It is computationally quite difficult. The process of determining the circuit modules and their position with the objective of area optimization is referred to as floor planning. Genetic algorithm is a stochastic optimization technique inspired by the theory of evolution. According to the theory of evolution, living organisms change their characteristics in response to change in environmental conditions. The characteristics of individuals are based on the makeup of cell structure called chromosome [4].

In this proposed work, we have presented the design and implementation of the reversible booth multiplier. Also, the floor planning technique has been done for optimized area of implementation. The 32-bit reversible Booth multiplier design has been presented.

II. REVERSIBLE LOGIC GATES

Now days, energy dissipation is the major issue in many applications. In every logical operation heat is dissipated from the circuit. This means loss of information happening in the circuit. In all high speed designs heat dissipation occurs. Recently several researchers have focused to reduce the loss of information from the circuit by using reversible logic gates design. It is one of the best designs to control the loss of information and it takes less power. The purpose of designing a reversible logic is to decrease the cost, reduce the loss of information and power and reduce the garbage output. Reversible logic function is used to determine the input from the output. Power dissipation is also very less, because of using reversible logic gates. It controls the overall power dissipation. Reversible logic is used in quantum computation, nanotechnology and other low power digital circuits [3]. The essential gates used for reversible logic synthesis are New Gate, Feynman Gate and Toffoli gate. One bit of information loss dissipates energy.

The main function of reversible logic is the number of inputs is equal to number of outputs. A device is said to be deterministic if its inputs and outputs be individually retrievable from each other. In other words such a device can also be called logically reversible.

If the whole device has the capacity to run backward it can be called physical reversible. These are the two basic conditions for reversible logic. Sometimes we get garbage outputs for attaining number of inputs equal to number of outputs. So garbage outputs are additional to make inputs and outputs equal whenever necessary.

Input + constant input = output + garbage

Feynman gate, Newmann gate and Toffoli gate from these three gates, we can design a new full adder circuit. These different reversible full adder designs are used in many arithmetic calculations like multiplication

III. Booth Algorithm

Signed multiplication is a vigilant process. Through unsigned multiplication there is no need to take the sign of the number into consideration. Even though in signed multiplication the same procedure cannot be applied for the reason that the signed number is in a 2‘s compliment form which would give in an inaccurate result if multiplied in an analogous manner to unsigned multiplication [10].

Thus here Booth‘s algorithm comes in. Booth‘s algorithm conserves the sign of the end result. While doing multiplication, strings of 0s in the multiplier call for only shifting. While doing multiplication, strings of 1s in the multiplier need an operation only at each end. We require to add or subtract merely at positions in the multiplier where there is a switch from 0 to 1 or from 1 to 0.

In the following flow chart we have, b=Multiplier, a=Multiplicand, m= Product [3].

Now here we will require twice as many bits in our product as we already have in our two operands. The leftmost bit of our operands of both the multiplicand and the multiplier is always a sign bit, and can‘t be used as part of the value. Then choose which operand will be multiplier and which will be multiplicand. If one operand and both are negative then they are represented in two's complement form. Start in on with a product that consists of the multiplier in the company of an additional X leading zero bits. Now check the LSB and the previous LSB of product to find out the arithmetic action. Add ‘0’ as the previous LSB if it is the FIRST pass [3]

Probable arithmetic actions are if:

00:- no arithmetic operation is performed only shifting is done.
01:- add multiplicand to left half part of product and then shifting is done.
10:- subtract multiplicand from left half part of product and then shifting is performed
11:- no arithmetic operation is performed only shifting is performed

IV. Genetic Algorithm

Genetic algorithm has the ability to simultaneously examine a set of possible solutions, manipulate them to achieve an optimized solution of the problem. The genetic algorithm starts with determining individuals in a given population. The individuals are encoded as binary strings called as chromosomal strings.

The Genetic algorithm operates on these encodings during the optimization process The Genetic algorithm then selects individuals from the population (selection process can involve probabilistic select function or specific techniques like roulette wheel selection, tournament selection etc.) And evaluates them using a predefine fitness function. [2]. A fitness function is a complex mathematical function needed to evaluate each chromosome encountered by the GA. The fitness measure of the chromosome represents the quality of the solution being examined that ultimately decides the optimality of the solution.

The fitness value of an individual determines whether or not it will survive through the generations. Hence it decides if an individual is fit to be chosen to participate in the operations. The Genetic algorithm then uses these individuals to produce a new generation thereby moving upwards on the evolutionary chart. The algorithm then uses two operators namely crossover and mutation. The crossover performs an exchange of chromosomal information between two individuals to produce an offspring. It combines the good qualities from the parents to produce fitter offspring. Thus the offspring inherit the best qualities from both of its parents. But the crossover operator leads to sameness in the genetic characteristics of individual generation by generation. The offspring resemble the parents to a great effect. Then mutation operator plays an important role in restoring lost genetic information by providing diversity. [1] Another advantage of a hardware implementation of a Genetic algorithm is the elimination of the need for complex time and resource consuming communication protocols needed by an equivalent software implementation to interface with the main application [5]

This is particularly advantageous to real-time applications such as reconfiguration of evolvable hardware. Another notable strength of genetic algorithms is that they perform well in problems for which the fitness landscape is complex - ones where the fitness function is discontinuous, noisy, changes over time, or has many local optima. Genetic algorithm has proven to be effective at escaping local optima and discovering the global optimum in even a very rugged and complex fitness landscape. However, even if a Genetic algorithm does not always deliver a provably perfect solution to a problem, it can almost always deliver at least a very good solution.

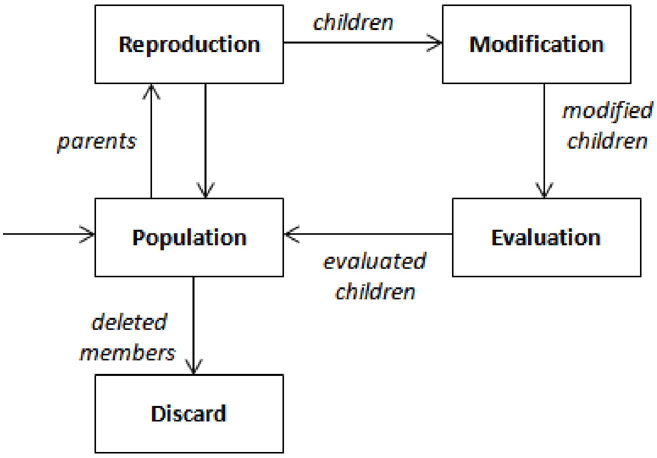


Fig.1 Genetic Algorithm cycle of reproduction

V. RESULTS & DISCUSSION

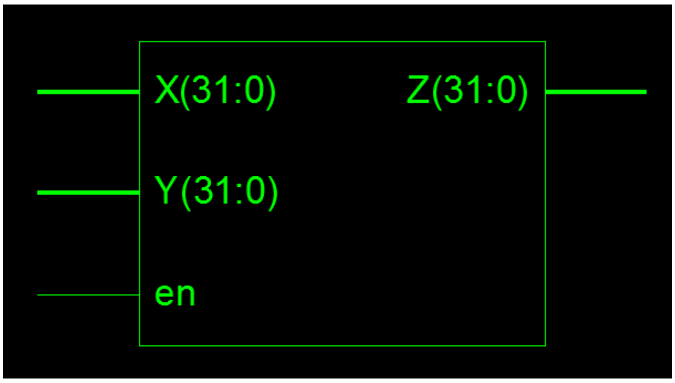


Fig. 2 Block structure of 32- bit booth multiplier

In fig.2, the block structure of 32-bit booth multiplier has been presented. Here, the enable bit has been added to enable the chip.

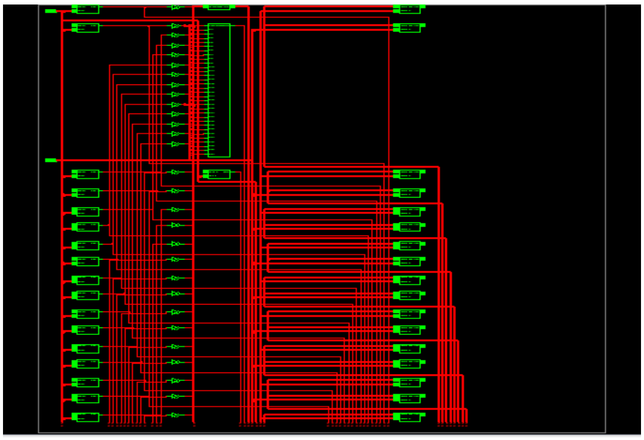
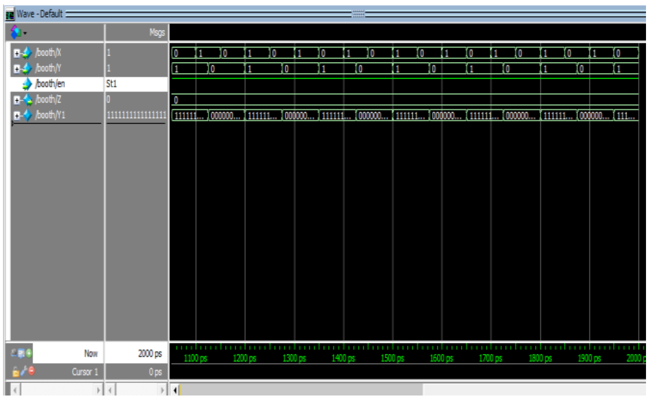


Fig. 3 Detailed RTL structure of the block structure of 32- bit booth multiplier

In fig.3, the detailed block structure of 32-bit booth multiplier has been presented. The booth algorithm implementation has been done for RTL synthesis and design. Here, the structure represents the multiplier structure for the final implementation.



Signal	Msgs
/booth/X	1
/booth/Y	1
/booth/en	St1
/booth/Z	0
/booth/Y1	1111111111111111

Fig. 4 Simulation results for 32-bit booth multiplier

In fig.4, the simulation result of 32-bit booth multiplier has been presented. Both the bits have been included for the same. The result has been finally presented here.

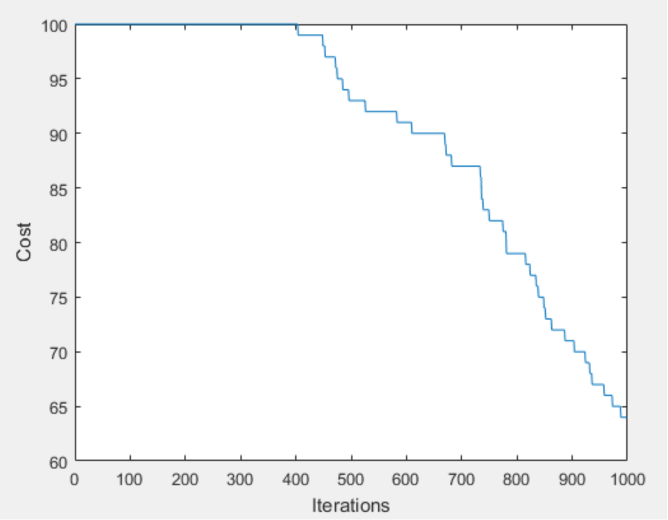


Fig.4. Quantum costing optimization using genetic algorithm

The cost is evaluated for each and every population in every iteration in fig.4 and it minimizes as the number of iterations rise. There is provision to reset the whole process if the results are not headed towards the idea outcome and start over again. Also, if the minimum cost is achieved in less than the given number of iterations, the algorithm halts and presents the current population as the final optimized result.

VI. Conclusion

Thus, in this work the final design of the 32-bit booth multiplier has been implemented. We have also presented the floor planning technique for optimized area of implementation. The Booth multiplier design has been presented. The simulation results and the RTL structures have been presented. Thus, the optimality of the genetic algorithm has been shown.

REFERENCES

[1] Deepali Chandel, Gagan Kumawat, "Booth Multiplier: Ease of multiplication", International Journal of Emerging Technology and Advanced Engineering, 2013

[2] Jakia Sultana, Sajib Kumar Mitra and Ahsan Raja Chowdhury, "On the Analysis of Reversible Booth's Multiplier", International Conference on VLSI Design

[3] CD Rawat1, Anmol Shahani, "A GENETIC ALGORITHM FOR VLSI FLOOR PLANNING", IJESAT, 2012

[4] K. Bhardwaj and B. M. Deshpande, "K-algorithm: An improved booth's recoding for optimal fault-tolerant reversible multiplier," in VLSI Design, 2013, pp. 362–367.

[5] S. Mitra and A. Chowdhury, "Minimum cost fault tolerant adder circuits in reversible logic synthesis," in International Conference on VLSI Design (VLSID), 2012.

[6] Soojin Kim and Kyeongsoon Cho. 2010. Design of High-speed Modified Booth Multipliers Operating at GHz Ranges. World academy of Science, Engineering and Technology. 61

[7] Shiann-Rong Kuang, Jiun-Ping Wang and Cang-Yuan Guo. 2009. Modified Booth multipliers with a Regular Partial Product Array. IEEE Transactions on circuits and systems-II. 56(5).

[8] Li-Rong Wang, Shyh-Jye Jou and Chung-Len Lee. 2008. A well-tructured Modified Booth Multiplier Design. 978-1-4244-1617-2/08/$25.00 ©2008 IEEE.

[9] A. K. Biswas, M. M. Hasan, A. R. Chowdhury, and H. M. H. Babu, "Efficient approaches for designing reversible binary coded decimal adders," Microelectronics Journal, vol. 39, no. 12, pp. 1693–1703, 2008.

[10] Magnus Sjalander and Per Larson-Edefors. 2008. The Case for HPM-Based Baugh-Wooley Multipliers. Chalmers University of Technology, Sweden.

[11] A. K. Biswas, M. M. Hasan, A. R. Chowdhury, and H. M. H. Babu, "Efficient approaches for designing reversible binary coded decimal adders," Microelectronics Journal, vol. 39, no. 12, pp. 1693–1703, 2008

[12] Pallav Gupta, Abhinav Agarwal, and Niraj K.Jha "An Algorithm for Synthesis of Reversible Logic Circuits", IEEE transactions on computer Aided Design of Integrated circuits and Systems, vol.25, 2006.

[13] H. Thapliyal and M. B. Srinivas, "A novel reversible tsg gate and its application for designing reversible carry look-ahead and other adder architectures," in Asia-Pacific Conference on Advances in Computer Systems Architecture, 2005, pp. 805–817.

[14] M. P. Frank, "Introduction to reversible computing: Motivation, progress,and challenges," in Conference on Computing Frontiers, 2005, pp. 385–390.

[15] D.Maslov and G. W. Dueck, "Reversible Cascades with Minimal Garbage", IEEE Trans on CAD, pp.1497-1509, 2004

[16] Wang G. 2004. A unified unsigned/signed binary multiplier. The Thirty-Eighth Asilomar Conference on Signals, Systems and Computers. 1: 513-516

[17] H. H. Babu, R. Islam, A. R. Chowdhury, and S. M. A. Chowdhury, "Reversible logic synthesis for minimization of full-adder circuit," in Euromicro Symposium on Digital Systems Design, 2003, pp. 50–54

[18] M. Perkowski, M. Lukac, P. Kerntopf, M. Pivtoraiko, D. Lee, H. Kim,W. Hwangbo, J.-w. Kim, and Y. W. Choi, "A hierarchical approach to computer-aided design of quantum circuits," in International Symposium on Representations and Methodology of Future Computing Technology, 2002, pp. 201–209.

[19] J.W. Bruce, M.A.Thornton, L. Shivakumaraiah, P.S. Karate and X.Li,"Efficient Adder Circuits Based on a Conservative Logic Gate", Procu.of the IEEE computer Society Annual Symposium on VLSI, pp.83-88, 2002.

[20] Hwang-Cherng Chow and I-Chyn Wey. 2002. A 3.3V 1GHz high speed pipelined Booth multiplier. Proc. of IEEE ISCAS. 1: 457-460.

[21] W. -C. Yeh and C. -W. Jen. 2000. High Speed Booth encoded Parallel Multiplier Design. IEEE transactions on computers. 49(7): 692-701

[22] Kim J. Y. 1999. Multiplier to selectively perform unsigned magnitude multiplication or signed magnitude multiplication. U Spatent. 5, 870, 322.

[23] V.Verdal A. Bareno and A Ekert, "Quantum Network for Elementary Arithmetic Operations"arXiv: quant-ph/9511018v1, nov-1995

[24] J.Fadavi-Ardekani. 1993. ªM×N Booth Encoded Multiplier Generator Using Optimized Wallace Trees,º IEEE Trans. VLSI Systems. 1(2).

[25] M. Aguirre-Hernandez and M. Linarse- Aranda Energy-efficient high -speed CMOS pipelined multiplier. Proc. of IEEE

[26] R. P. Feynman, "Quantum mechanical computers," Optics News, vol. 11,no. 2, pp. 11–20, Feb 1985.

[27] R. Landauer. 1961. Irreversibility and Heat Generation in the Computing Process. IBM Journal of Research and Development, 5(3), 183-191.

[28] C.H. Bennett. 1973. Logical reversibility of Computation, IBM Journal Research and Development. 17, 525-532.

[29] V. Kamalakannan, Dr P.V.Rao,Veeresh Patil ,"Analysis of Reversible Binary Adder/Subtractor Circuit", International Journal of Advanced and Innovative Research, ISSN: 2278-7844,Volume 1, Issue 4, pp 129-134.