FPGA IMPLEMENTATION OF 4-BIT AND 8-BIT SQUARE CIRCUIT USING REVERSIBLE LOGIC

Shwetha. S Patil[1], Mahesh Patil[2], Venkateshappa[3]
Assistant Professor[1],PG Student[2], Professor[3]
[1,2,3]Dept. of ECE,
[1]MVJ College of Engineering, [2]M. S. Engineering College Bangalore,
[3]M. S. Engineering College Bangalore

Abstract

With the increase in the demand for the low power Systems that are Digital, dissipation of heat or energy is one the major critical and limiting factors. To reduce the heat or energy dissipation in the digital circuits Reversible logic can be used and it plays a very important role in bioinformatics, optical information processing, Digital Signal processing etc., Squaring is most commonly used in divisions, roots, or reciprocals and hence it is most widely used in reversible DSPs and Reversible ALUs. In this paper a dedicated reversible logic circuit to compute square of an operand is done. The proposed dedicated circuit for square computation has less constant inputs, less garbage outputs, less gate count and less quantum cost when compared to the traditional existing reversible multipliers.

Keywords: Reversible logic, ALU, DSP, Toftoll gates, Peres gates and Double peres gates.

I. A. INTRODUCTION

Today electronic industries are rapidly growing that requires the digital system which are efficient i.e. systems with low power. In the conventional all the logic operations performed using the millions of gates available are irreversible. That is, whenever a particular logical operation is computed some of the information about the input is erased or lost and is dissipated as heat/energy and this energy loss is a very important parameter in the digital design. Reversible quantum logic is one of the solution or an alternate that reduces the power dissipation.

According to Rolf Landauer, for an irreversible logic operation kTln2 [1]Joules of heat is dissipated when a bit is erased or when a single bit of information is lost. Here k is Boltzmann's constant and T is absolute temperature at which computation will be performed. For room temperature T, one bit information dissipates 2.9 x 10^{-21}. J of heat energy.

B. INTRODUCTION TO REVERSIBLE LOGIC

Reversible logic gates /circuit generates unique output pattern from each input pattern and vice versa, i.e., there is always a one to one correspondence between the inputs and outputs. That there will be equal number of inputs and outputs. To be more formal, the Reversible logic gate has m-input, m-output (denoted m x m) where each possible unique input vector is mapped into a unique output vector as shown in fig.1. While designing the reversible logic gates some important parameters must be considered such as gate count (N), constant inputs (CI) / ancilla inputs, garbage outputs (GO) and quantum cost (QC). The use of reversible logic was introduced by C.H.Bennett [2].

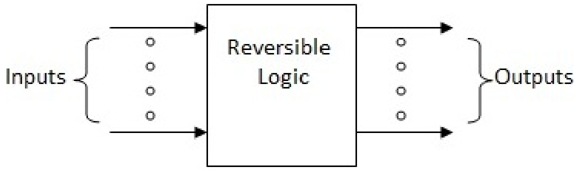


Fig.1 A general reversible gate

The quantum cost is a number of 1 x 1 or 2 x 2 reversible gates that are used to construct a circuit. Constant inputs also called as ancilla inputs or bits are used to store the intermediate values during any computation. Garbage outputs are the unused bits present in circuit. Garbage outputs cannot be avoided as they are require to achieve one to one mapping. While constructing a quantum reversible circuit the designer always tries to optimize the above parameters. In reversible logic circuits fan-out and loops are not permitted.

Arithmetic units are the important components of any processors. Recently the designers are working more on the designs of reversible quantum adders [3,4,5,18,19], barrel shifters [13,8], multipliers [16,9], floating point units [11,17]. Among all of them multiplier circuit plays an important role in improving the data processing performance in a processor and improves the speed of computation. Squaring the most frequently used function in division (Newton Raphson division and Taylor series expansion), roots, or reciprocals [6,7]. Squaring also has its application in digital signal processing like Euclidean distance computation and exponential calculation in cryptography. To perform the square of an operand the normal reversible multiplies circuit is not efficient as it results in redundant partial products and requires an extra additional circuitary which creates once head in terms of ancilla inputs, garbage output and quantum cost.

II. REVERSIBLE LOGIC GATES USED IN THE DEIGN

Toffoli gate (TG): Figure 2a and 2b shows the Toffoli gate and graphical representation. It is a 3x3 gate with 3 inputs (A,B,C) and 3 outputs (P,Q,R) where P=A, Q=B, $R=AB\oplus C$. It is one of the universal reversible gate with a quantum cost 5 [14], since it requries 5 2 x 2 quantum gates for its implementation.

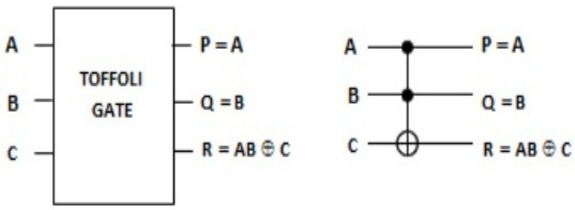


Fig.2a Toffoli gate Fig.2b Graphical representation

Peres Gate (PG):

Fig.3a and 3b shows the Peres Gate and its graphical representation. It is also a 3 x 3 reversible gate with the inputs (A,B,C) and outputs (P,Q,R) where P=A, $Q=A\oplus B$, $R=AB\oplus C$. Peres Gate has quantum cost of 4 [15], as it requires 4 2x2 reversible gate to design it.

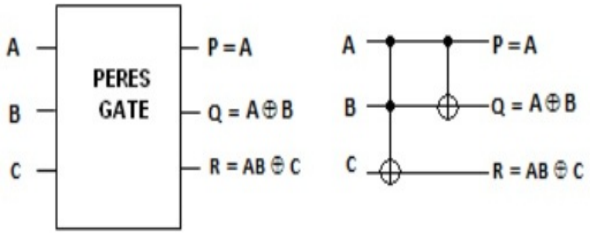


Fig.3a Peres Gate Fig.3b Graphical representation

Double Peres Gate (DPG):

Figure 4a and 4b shows the Double Peres Gate (DPG) block diagram and its graphical representation, respectively. It is a 4x4 reversible gate having inputs (A,B,C,D) and the outputs (P,Q,R,S). It has a quantum cost of 6 [16], since it needs 6 2x2 quantum gates to construct it.

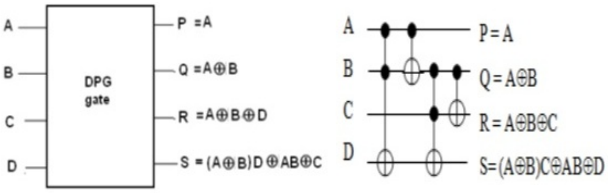


Fig.4a DPG Gate Fig.4b Graphical representation

III. PROPOSED REVERSIBLE CIRCUIT FOR SQUARE COMPUTATION

This section of paper presents a special separate quantum circuitary for squaring operation. Fig.5 shows the generation of partial products to compute the square of 4 bit operand. The middle section of the figure 5 shows the array of partial products in which an equivalence relation $a_i - a_j + a_j . a_i = 2.a_i.a_j$ is used to combine some of the product terms. Once the equivalence relation is applied to these terms, the weight of product terms increases by 2, and hence it is shown in the next column. The arrows indicate the shifting of product terms that are combined to the left by one position. Finally, we obtain the partial products in the reduced form shown in the last section of figure.5.

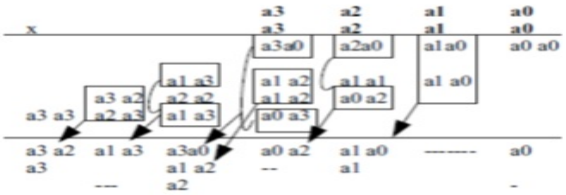


Fig.5 Partial Product Generation of 4-bit square circuit

A. *Step-1 : Partial Product Generation*

The first step in computing a square of an operand is to generate all the partial products. We use Toffoli gate to obtain the reduced partial products at the output as shown in Fig.6a and 6b. Figure 6a shows the block diagram using Toffoli gates and Fig.6b the graphical representation of partial product generation. To obtain the AND operation for the partial product terms, the input C of the Toffoli gate is kept at zero and this is an ancilla input (constant zero). The Toffoli gates are connected in series and there are no garbage outputs. Toffoli gate is used instead of Peres gate To for ANDing because the input bits have to be preserved in reversible logic, so the regenerated bits at the output will not be considered as garbage outputs.

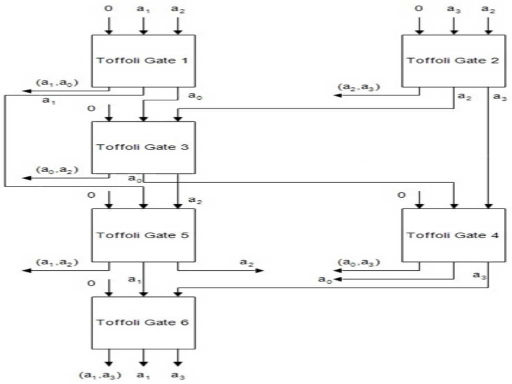


Fig.6a partial products generation block diagram

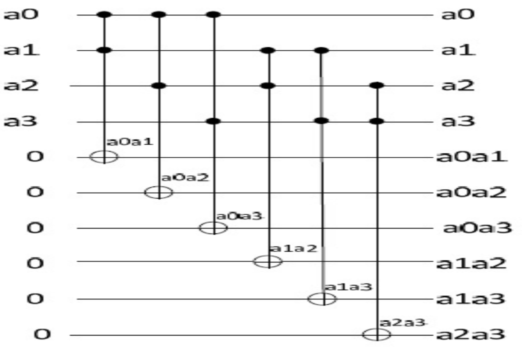


Fig. 6b Graphical Representation of Partial Product Generation

B. Step-2 Summation Circuitary

The final product term for the 4-bit square unit is obtained in the summation stage by using carry save method as shown in fig.7. It requires full and half adder in a reversible manner to obtain the final product terms. We get the reversible full adder using Double Peers gate by connecting the inputs C=0 and D=C_{in} as shown in fig.8. In the same manner we obtain half adder by making input C=0 as shown in fig.9.

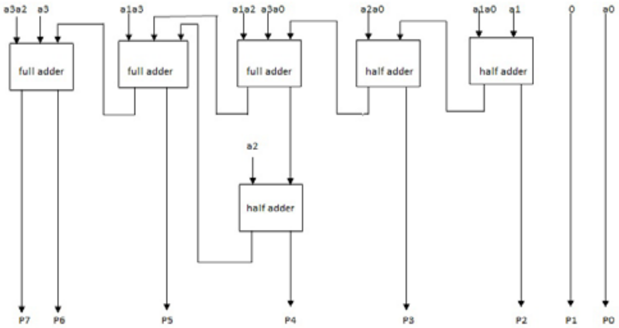


Fig.7 Block Diagram of Summation Circuitary for 4x4 square unit.

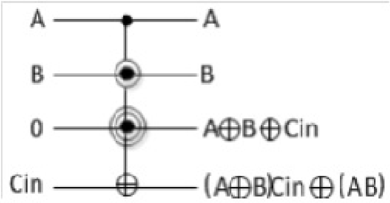


Fig.8 Full adder using DGP gate

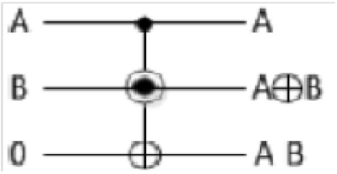


Fig.9 Half adder using PG Gate

IV. DESIGN OF AN 8 X 8 REVERSIBLE SQUARE UNIT

Design an 8-bit square architecture multiplies two binary inputs to obtain the reduced partial products by removing the redundant bits using equivalence relation.

Partial Product Generation and Summation Circuitary of 8-bit reversible square circuit

The Toffoli gates connected in series are used to obtain the final reduced partial product terms where the input C of Toffoli gate is set to zero to obtain the AND operation. The final partial product is applied to the full adders and half adders to obtain the square of number. Full adder is constructed using Double peers gate by making input C=0 and D=C_{in} and half adder using Peres gate by making C=0 as shown in Fig.10.

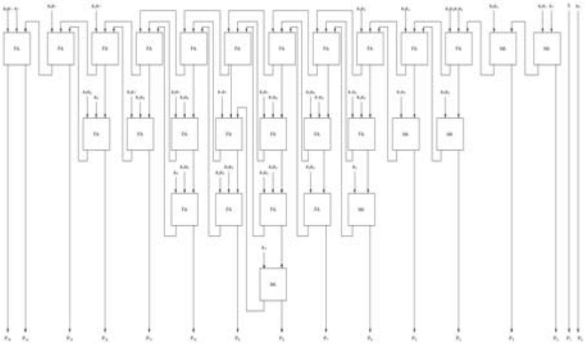
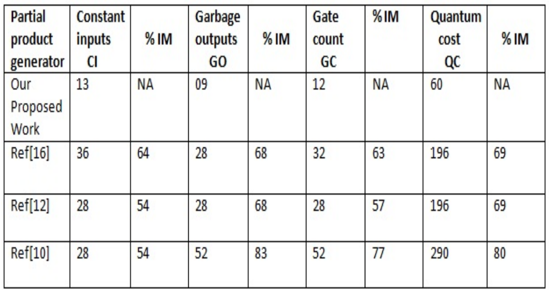


Fig. 10 A complete proposed 8x8 reversible square unit with reduced partial products.

V. COMPARISON RESUTLS FOR 4-BIT SQUARE ARCHITECTURE.

The comparison results of a complete 4x4 square architecture is as shown in Table-1. In our proposed work the number of constant inputs, garbage outputs, gate count and quantum cost are reduced or optimized when compared to the previously proposed multiplier designs [16,12,10]. In the table CI indicates constant inputs, % IM is percentage of improvement which is rounded off to nearest value, GO is garbage output and Q is quantum cost.

.



Partial product generator	Constant inputs CI	% IM	Garbage outputs GO	% IM	Gate count GC	% IM	Quantum cost QC	% IM
Our Proposed Work	13	NA	09	NA	12	NA	60	NA
Ref[16]	36	64	28	68	32	63	196	69
Ref[12]	28	54	28	68	28	57	196	69
Ref[10]	28	54	52	83	52	77	290	80

Table-1: Comparison results of complete 4-bit square architecture

VI. SIMULLATION RESULTS

The complete 4x4 and 8x8 reversible square architecture is functionally verified by simulation. A library of reversible gates is created for the reversible gates used in the design to perform, in the simulation in Verilog. This library is used implement the gate level model of this design. We have used test bench in order to verify the functional correctness of the design. Xilinx 13.4 ISE simulate is used to perform the simulation. The simulation results for 4x4 and 8x8 reversible square unit are shown in Fig.11 and Fig.12 respectively.

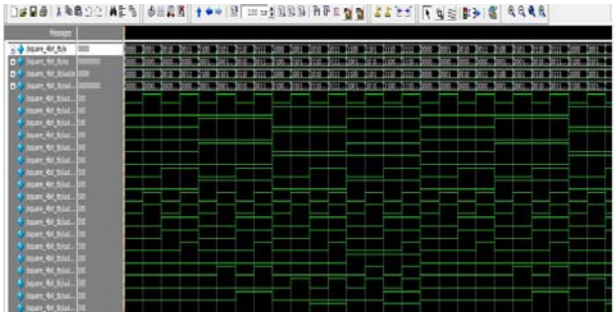
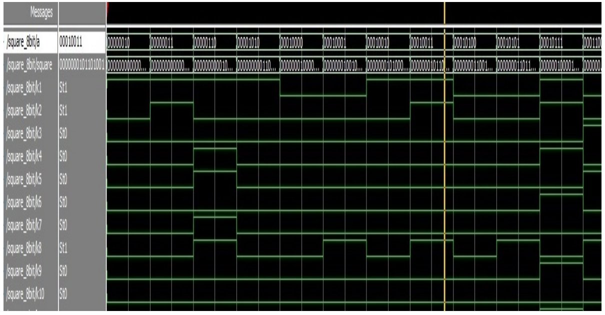


Fig. 11 Functional Correctness of 4 bit square circuit



Messages	
/square_8bit/a	00010011
/square_8bit/square	0000000101101001
/square_8bit/k1	St1
/square_8bit/k2	St1
/square_8bit/k3	St0
/square_8bit/k4	St0
/square_8bit/k5	St0
/square_8bit/k6	St0
/square_8bit/k7	St0
/square_8bit/k8	St1
/square_8bit/k9	St0
/square_8bit/k10	St0

Fig. 12 Functional Correctness of 8 bit square circuit

VII. HARDWARE IMPLEMENTATION

Once the simulation is done using Xilinx simulation tool, the modules are dumped to the hardware. Here an ASIC FPGA i.e., Field Programmable Gate Array consisting of various programmable logic blocks is used to logically verify the correctness of a design. FPGA with spartan 3 family is used and configured to get the output on the FPGA board.

Fig13. shows the complete connection of FPGA board, LED kits and the Computer. After these connections are made, switch on the power supply and the switch on FPGA board is turned ON all the LEDs light up. Once the device is configured it is ready to accept the inputs to produce the outputs.

The inputs are given using the dip switches persent on the kits where LEDs glow, for the input given the particular LED lights up to show the output.

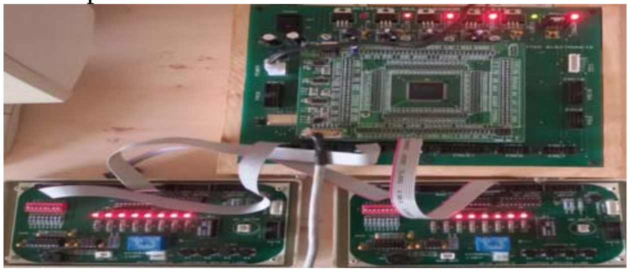


Fig.13. FPGA Implementaion of Square circuit

VIII. CONCLUSION

In this paper we have designed a unique separate circuit for square computation using reversible logic. Since square finds its applications in arithmetic units which are used in DSPs, it increase the processor speed by when the circuit is optimized interms of ancilla inputs, garbage output gate count and quantum. Here the maximum improvements are achieved in regard to these terms and hence the energy loss in the form of heat is less and can be used in low power digital systems. This design is very useful to construct reversible ALUs and DSP processors. There is an 85% improvement in garbage outputs and gate count, 83% improvement on quantum cost and ancilla inputs for 4x4 reversible square circuit.

IX. REFERENCES

1. R.Landauer, “Irreversibility and heat generation in the computational process,” IBM J.Research and Development, vol.5, pp.183-191, Dec.1961.
2. C.H.Bennett, “Logical reversibility of computation,” IBM.J. Research and Development, vol.17, pp.525-532, Nov.1973.
3. V.Vedral, A.Barenco, and A.Ekert, “Quantum networks for elementary arithmetic operations,” Phys. Rev. A, vol.54, no.1, pp.147-153, Jul. 1996.
4. Y.Takahashi, “Quantum arithmetic circuits: a survey,” IEICE Trans. Fundamentals, vol. E92-A, no.5, pp.276-1283, 2010.
5. A Peres, “Reversible logic and quantum computers,” Phys. Rev. A Gen. Phys., vol.32, no.6, pp.3266-3276, Dec. 1985.
6. S.Oberman and M.Flynn, “Division algorithms and implementations,” IEEE Transactions on Computes, 46, pp.833-854, Aug.1997.
7. P.Rabinowitz, “Multiple precision division,” Commun. ACM, vol.4, p.989, Feb.1961.
8. H.Thapliyal, A. Bhatt and N.Ranganathan, “A New CRL Gate as Super Class of Fredkin Gate to Design Reversible Quantum Circuits”, Proceedings of the 56th IEEE
9. International Midwest Symposium on Circuits and Systems (MWSCAS), Columbus, Aug 2013, pp.1067-1070.
10. S.Kotiyal,H. Thapliyal and N.Ranganathan, “Circuit for Reversible Quantum Multiplier Based on Binary Tree Optimizing Ancilla and Garbage Bits”, Proceedings 27th International Conference on VLSI Design (VLSI Design), Mumbai, India, Jan 2014.
11. M.S.Islam, M.M.Rahman, Z.Begum and M.Z. Hafiz, “Low cost quantum realization of reversible multiplier circuit,” Information Technology Journal, vol.8, no.2, pp.208-213, 2009.
12. T.D.Nguyen, and R.Van Meter, “A Space-Efficient Design for Reversible Floating Point Adder in Quantum Computing”, arXiv preprint arXiv: 1306.3760,2013.
13. M.Haghparast, M.Mohammadi, K.Navi, and M.Eshghi, “Optimized reversible multiplier circuit,” Journal of Circuits, Systems & Computers, vol.18, no.2, pp.311-323-2009.
14. S.Kotiyal, H. Thapliyal, and N.Ranganathan, “Design of a reversible bidirectional barrel shifter,” Proceedings of the 11th IEE NANO, Portland, Aug.2011, pp.463-468.
15. E.Fredkin and T.Toffoli, “Conservative logic,” International J.Theor. Physics, vol.21, pp.219-253, 1982.
16. A. Peres, “Reversible logic and quantum computers,” Phys. Rev. A.Gen.Phys, vol.32, no.6, pp.3266-3276, Dec.1985.
17. W.N.Hung, X.Song, G.Yang, J.Yang, and M.Perkowski, “Optional synthesis of multiple output Boolean functions using a set of quantum gates by symbolic reachability analysis, IEEE Trans. Sept.2006.
18. M.Nachtigal, H.Thapliyal and N.Ranganathan, “Design of a Reversible Single Precision Floating Point Multiplier Based on Operand Decomposition”, Proceedings of the 10th IEEE International Conference on Nanotechnology, Seoul, Korea, Aug 2010, pp.233-237
19. H.Thapliyal and N.Ranganathan, “Design of Efficient Reversible Logic Based Binary and BCD Adder Circuits”, ACM Journal of Emerging Technologies in Computing Systems, Vol.9, No.3, pp.17:1-17:31, Sep. 2013.