ISBN 978-981-11-3671-9
Proceedings of 2017 the 7th International Workshop on Computer Science and Engineering (WCSE 2017)
Beijing, *25-27 June*, *2017*, *pp. 1028-1032*
doi: 10.18178/wcse.2017.06.178

Efficient Reversible Arithmetic Logic Units Designs and Evaluation

Omkar S. Murkumbi, Tanmayee Tankasali and Nan Wang[+]

California State University, Fresno, USA

Abstract. Every day, the brilliant minds of the semi-conductor industry thrive in bettering computing ICs. As a result of advances made by the silicon industry world, computing devices today process faster, occupy less area, and minimize overall power consumption. By Moore's Law, the transistors on a die of silicon doubles in number every two years, which causes the leakage currents' rates to increase rapidly. As of today, the feature size in transistor fabrication has reached 8 nm. In the coming years, the fabrication industry will approach to a saturation point in the feature size reduction. The electrical circuit's energy consumption can be reduced if the operation performed is reversible in nature. This reversible computing opens up ways for quantum computing, which has been considered the best design approach for over five decades. In this paper, we will begin by discussing existing Arithmetic Logic Unit designs based on reversible structures. We then propose a new reversible gate named the OSG gate based on which two new ALU designs are presented. The designs and simulations were carried out on Xilinx ISE 13.2 design suite and the results are compared and evaluated.

Keywords: ALU design, OSG gate, quantum switching, reversible gates

1. Introduction

A quantum computer with a specified number of qubits is totally different from a traditional computer which employs the same number of binary bits. For example, expressing an n-qubit system on a traditional computer needs the storage of 2^n complex coefficients, while to describe a traditional n-bit system the quantum computer is sufficient to represent the values of the n bits, that is, only n numbers[1]. The AND, NOT and OR gates are considered to be universal gates. Using these three gates, any logical statement can be realized through a set of Boolean equations, otherwise known as the Boolean realization of a problem statement. However, the OR and AND gates aren't reversible in nature. The output can be calculated from the input but not vice versa. Consider an OR gate with 2 inputs and 1 output. When the output is 0, we can easily deduce that the inputs to the gate is 00. On the other hand, when the output is 1, there are three possible combinations of the inputs (01,10,11) which makes the gate irreversible and cause a loss of information in the computation. In digital logic execution, every computational step requires a minimum amount of energy. In Boolean realization, one or more bits at the output are overwritten, which projects an information loss analogous to energy loss. According to Von Neumann Landauer principle [2], this energy loss is given by the equation: ***E= kT ln 2 Joules.*** In the equation, E is the energy loss with respect to 1 bit of information loss. This loss of energy on a large scale adds up to the heat loss in the IC component due to the billions of transistors. This heat loss can be reduced or ideally stopped if the computation takes place in a reversible way. For a computation to be reversible, we need to apply a quantum computing approach in the design's primitive cells. This type of computing employs switching logic for realization of complex equations.

The organization of this paper is as follows. In Section 2, we provide a brief theory background about different reversible gates available and their respective working mechanisms. Section 3 gives the designs of some existing reversible ALU (Arithmetic Logic Unit) architectures. Section 4 proposes a new reversible OSG gate and discusses two new approaches to construct reversible ALUs. Section 5 shows the simulation results, and finally, Section 6 concludes the paper.

[+] Nan Wang. Tel: + (001) 304-610-7731; fax: +(001) 559-278-6297
Email address: nwang@csufresno.edu

2. Background

For a gate or any Boolean function to be realized in a reversible way, there are conditions that have to be satisfied. The conditions [2] are:

- The number of inputs is equal to the number of outputs.
- Every output vector has a unique pre-image of the input vector.

Thus, a reversible gate is an N input and N output device. A programmable reversible logic gate with m select inputs may produce at maximum n*2m logical calculations on the n logical outputs [3]. There has to be a one to one mapping between the output vector and the input vector. In this way, output can be determined from the input vector and vice versa. This ability of the circuit can make a function reversible in nature, resulting in reduction of power dissipation. There are some cost metrics on which reversible gates can be analysed [4]. The basic principles of reversible gates and their respective design metrics used for this project are as follows. Garbage signals: Number of additional output signals generated to attain reversibility. They are not used in the synthesis of the design.

- Quantum cost: The number of primitive reversible logic gates required to realize an operation.
- Gate Count: The number of reversible gates used in order to realize a given function from ground up.
- Transistor Cost: Number of transistors if CMOS technology is adapted for the design.

Some of the commonly used reversible gates to realize Boolean operations are shown in Fig.1.

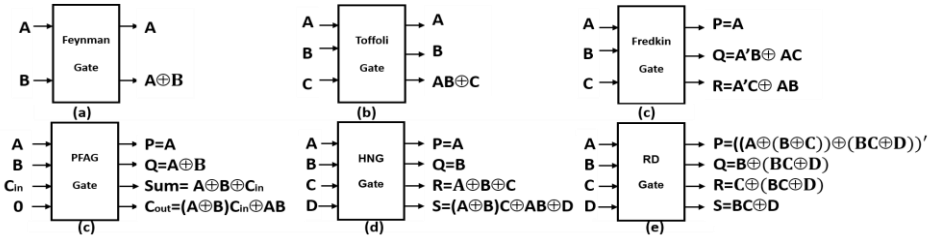


Fig.1: Basic reversible logic gates: (a) Feynman Gate; (b) Toffoli Gate; (c) Fredkin Gate; (d) PFAG Gate; (e) HNG Gate and (f) RD Gate [4-6].

As seen in the figure above, every gate has equal number of inputs and outputs. Some reversible gates are parity preserving reversible gates, which have equal parity on the input and output vector. These are examples of good quantum design. In these gates, problem solving and debugging become easier due to the one to one mapping of inputs and outputs. By forcing the inputs to constant values, a number of logical operations can be realized using these basic reversible gates.

3. Existing Reversible ALU Architectures

3.1. ALU1 and ALU2 Designs

The block diagram and control circuit of ALU1 and ALU2 are shown in Fig.2: (a) and (b) respectively.

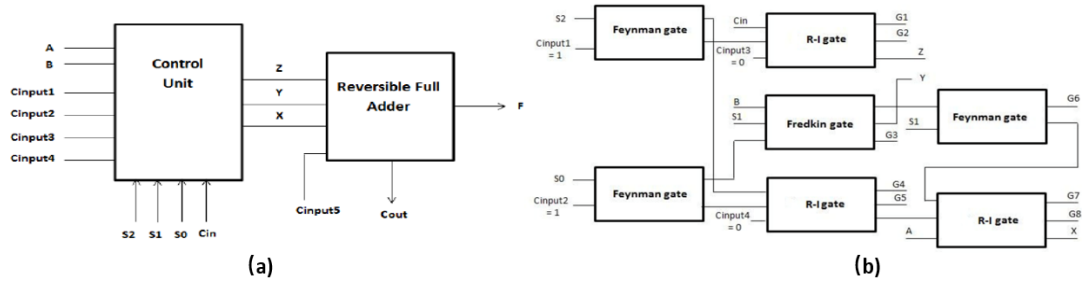


Fig.2: (a) Block diagram (b) control circuit for ALU1 and ALU2 designs [6].

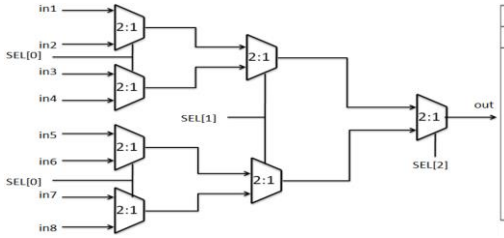
The ALU1 uses the PFAG (Fig.1-d) as the reversible full adder and ALU2 uses the HNG (Fig.1-e) gate.

3.2. ALU3 Design

In this ALU design, a 2:1 MUX by Boolean operations using reversible gates is proposed. This 2:1 MUX is used to form an 8:1 MUX. To realize the operations of the ALU, a specific value for B is provided as the input to the full adder. The 8:1 MUX and truth table for the B input to be generated are shown in Fig.3.

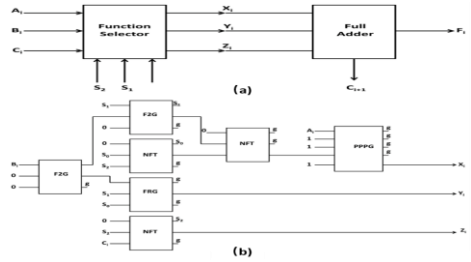
3.3. ALU4 Design

The block diagram and control circuit of the ALU4 design and the truth table for ALU operations are shown in Fig.4.



Function Select			Y	Output	Function
S_1	S_0	C_{in}			
0	0	0	0	$F = A$	Transfer A
0	0	1	0	$F = A + 1$	Increment A
0	1	0	B	$F = A + B$	Add B to A
0	1	1	B	$F = A + B + 1$	Add B to A plus 1
1	0	0	$\overline{B}$	$F = A + \overline{B}$	Add 1's Complement of B to A
1	0	1	$\overline{B}$	$F = A + \overline{B} + 1$	Add 2's Complement of B to A
1	1	0	All 1's	$F = A - 1$	Decrement A
1	1	1	All 1's	$F = A$	Transfer A

Fig.3: (a) 8:1 MUX using 2:1 MUX (b) B input for the operations of ALU3[5].



Selection				Output	Function
S_2	S_1	S_0	C_{in}		
0	0	0	0	$F = A$	Transfer A
0	0	0	1	$F = A + 1$	Increment A
0	0	1	0	$F = A + B$	Addition
0	0	1	1	$F = A + B + 1$	Addition with Carry
0	1	0	0	$F = A - B - 1$	Subtraction with Borrow
0	1	0	1	$F = A - B$	Subtraction
0	1	1	0	$F = A - 1$	Decrement A
0	1	1	1	$F = A$	Transfer A
1	0	0	X	$F = A + B$	OR
1	0	1	X	$F = A \oplus B$	XOR
1	1	0	X	$F = AB$	AND
1	1	1	X	$F = A'$	NOT

(c)

Fig.4: (a) Block diagram and (b) control circuit for ALU4 design [5] (c) Truth table of ALU4 operations.

4. Proposed ALU Designs

4.1. Proposed OSG Reversible Gate

To implement our new ALUs, we have proposed a new reversible gate named the OSG gate. It is a 3 input 3 output gate. This gate has a parity preserving logic. The logic diagram of the OSG gate is shown below in Fig.5.

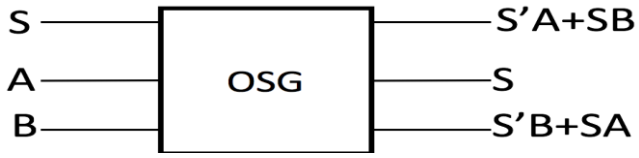


Fig.5: The proposed OSG gate.

4.2. Proposed ALU-I Design

We use a switching logic, which employs ideas of quantum computing and an OSG gate as a multiplexer to realize the whole ALU. The whole ALU design just employs one type of reversible gate. This design technique makes the ALU design less complex and more efficient compared to the existing architectures. To reduce the number of gates used, the truth table of the full adder has been simplified and reduced of redundant execution units. The Cin input to the full adder is simplified which helps in reducing the number of multiplexers needed. The circuit shown in Fig.6 (a) gives the function selector ability to generate the input B to the full adder. The Fig.6 (b) shows the execution of the sum and carry out by reduced variable methodology.

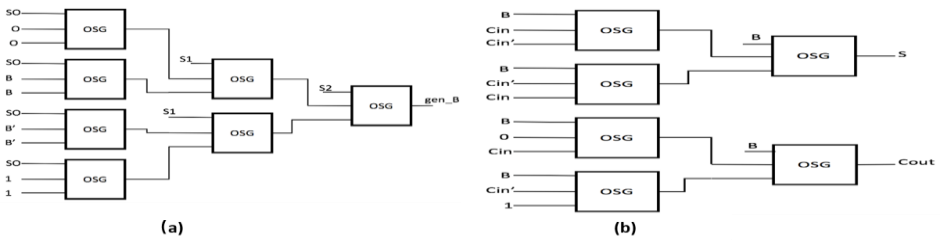


Fig.6: (a) Function selector and (b) execution unit for the proposed ALU-I.

4.3. Proposed ALU-II Design

In this ALU-II design, we use decoding logic to realize a full adder as shown in Fig.7 (a). The same Function Selector decides the operations of the ALU-II as the one ALU-I uses. In simple words, the function selector decides the B input to be fed to the full adder. The idea being that we wanted to use more and more switching logic, which used reversible gates with less number of inputs and outputs. On the other hand, the ALUs in the literature uses reversible gates with N=5, thus creating complexity and delay in the design. On the same grounds of using multiplexing logic in the proposed ALU-I design, we used decoding logic in the proposed ALU-II design.

To design a 3-to-8 decoder, we made use of three 2-to-4 decoders. Realization of 2-to-4 decoders using reversible gate was done using the RD gate [6] which is shown in Fig.1 (f). Driving the B and C inputs of the RD gate and making A and D to logic 0, a 2-to-4 decoder can be designed as shown in Fig.7 (b).

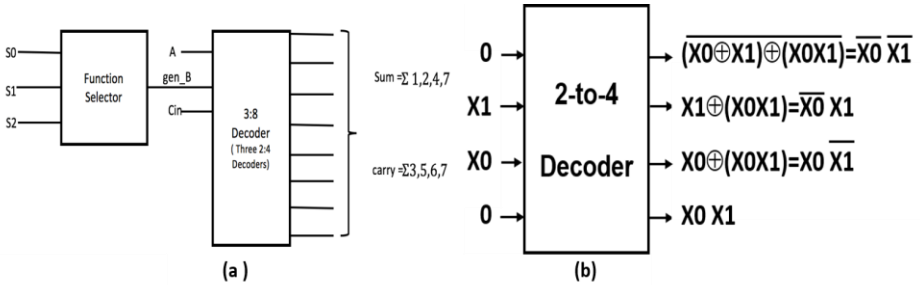


Fig.7: (a) Proposed ALU-II design (b) 2-to-4 decoder using RD gates [5]

The block inputs A, B and Cin act as the three inputs to the 3-to-8 decoder. Comparing the truth table of the full adder to the array outputs of the 3-to-8 decoder, the sum and carry out equations can be written in the form of minterm expressions as S =∑m(1, 2, 4, 7) and Cout =∑m(3, 5, 6, 7).

5. Simulation and Results

All simulations were carried out on Xilinx ISE 13.2 design suite. In all the designs, the inputs are given by A, B and Cin. The select signals to select the functionalities are given by S0, S1 and S2. S and Cout give the sum and carry out. F gives the functional outputs of the ALUs.

5.1. Simulation Results

The simulation results of the ALU1, ALU2, ALU3, and ALU4 are shown in Fig.8; the simulation results of the proposed ALU-I and ALU-II designs are shown in Fig.9.

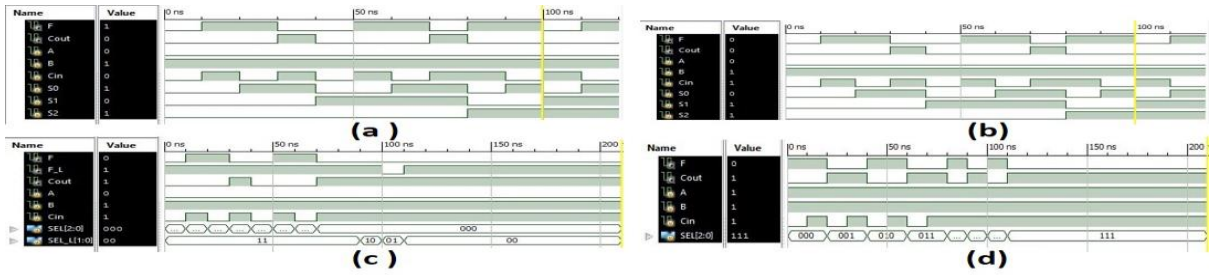


Fig.8: Simulation result for (a) ALU1; (b)ALU2; (c) ALU3 (d) ALU4 designs

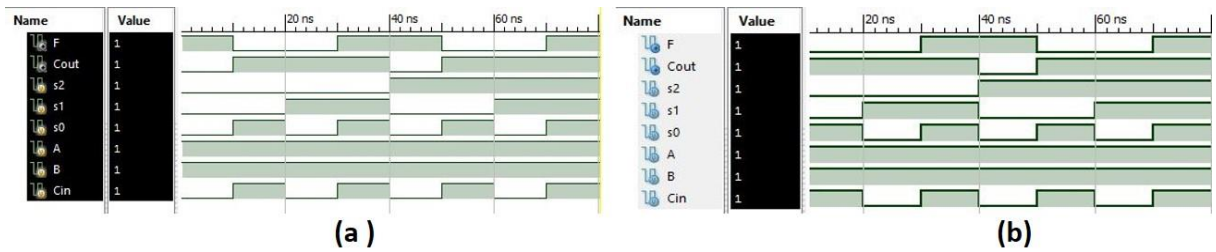
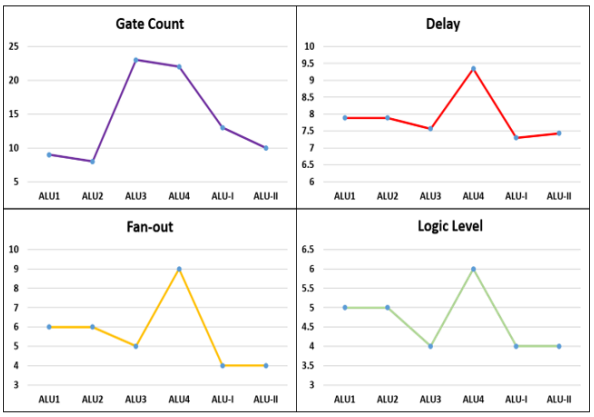


Fig.9: Simulation result for proposed (a) ALU-I and (b) ALU-II designs.

5.2. Simulation Result Justification

The comparison of the existing ALU designs and the proposed ALU-I and ALU-II designs based on cost metrics is shown in Fig.10.



ALU DESIGN	Gate count	Delay (ns)	Macros	Logic Level	Fan-out
ALU1 [6]	9	7.883	11	5	6
ALU2 [6]	8	7.883	11	5	6
ALU3 [4]	23	7.562	42	4	5
ALU4 [4]	22	9.341	21	6	9
Proposed ALU-I	13	7.3	NA	4	4
Proposed ALU-II	10	7.43	11	4	4

Fig.10: Performance comparison of the reversible ALUs.

As shown in Fig.10, the proposed ALU-I and ALU-II have the least path delays compared to the existing ALU designs, while the gate count and number of macros are almost comparable to the existing units. The abstraction logic level is the lowest in the proposed ALU-I and ALU-II which also have less fan-out when compared to other ALUs as a result of employing switching logic. Proposed ALU-I and ALU-II uses a 3-input and 3-output gate which is less complex that the PFAG gate and HNG gate. The complexity of the gate is directly proportional to the transistor cost. For 16-bit or 32-bit ALUs, difference in the gate count is no more negligible. The gate count can be translated to the power consumption of the circuit. Thus, reducing the power consumption of the primitive block computer architecture like the ALU, power consumed during a computation is reduced on the overall system.

6. Conclusion

In this paper, the principles of reversible computing were studied. The advantages and background of quantum reversible computing were discussed. Some of the existing reversible ALUs in the literature were studied and designed using Verilog on Xilinx ISE 13.2 design suite. Subsequently, a new reversible gate was proposed called the OSG gate, which was used as a primitive block for the proposition of two new ALU designs. One used multiplexing logic and the other one used decoding logic for realization of arithmetic operations. Based on the fundamental cost metrics, simulation results suggest that performance of the proposed ALUs is better when compared to the existing reversible ALU designs. Future work on this topic will be making the use of reversible gates in more complex sequential circuits [7].

7. References

[1] https://en.wikipedia.org/wiki/Quantum_computing

[2] R., Landaurer. Irreversibility and Heat Generation in the Computational Process. *IBM Journal of Research and Development,* 5, 1961, pp.183-191.

[3] M. Morrison &N. Ranganathan. Design of a reversible ALU based on novel programmable reversible logic gate structures. *Proc. Of IEEE Computer Society Annual Symposium on VLSI,* 2011, pp. 126-131.

[4] R. Saligram, S.S.Hegde, S.A. Kulkarni, H. R. Bhagyalakshmi & M.K. Venkatesha. Design of parity preserving logic based fault tolerant reversible arithmetic logic unit. *arXiv preprint arXiv*:1307.3690, 2013.

[5] M. Morrison, M. Lewandowski, R. Meana, &N. Ranganathan. Design of static and dynamic RAM arrays using a novel reversible logic gate and decoder. Proc. of 11th *IEEE Conference on NANO,* 2011, pp. 417-420.

[6] L. Gopal, N.S.M Mahayadin, A.K. Chowdhury, A.A. Gopalai & A.K. Singh. Design and synthesis of reversible arithmetic and Logic Unit (ALU). Proc. Of IEEE *Computer, Communications, and Control Technology.* 2014, pp. 289-293.

[7] H. Thapliyal, &N. Ranganathan (2010). Design of reversible sequential circuits optimizing quantum cost, delay, and garbage outputs. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 2010, 6(4), 14.