ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 10, October 2013

Design and Synthesis of 16-bit ALU using Reversible Logic Gates

Premananda B S[1], Y M Ravindranath[2]

Department of Telecommunication, R.V. College of Engineering, Bangalore, India[1]

Department of Electronics and Communication, M.E.I. Polytechnic, Bangalore, India[2]

Abstract - The paper describes the design of programmable reversible logic gate structures, targeted for the ALU implementation and their use in the realization of an efficient reversible ALU. Using reversible logic gates instead of traditional logic AND/OR gates, a reversible ALU whose function is the same as traditional ALU is constructed. Comparing with the number of input bits and the discarded bits of the traditional ALU, the reversible ALU significantly reduce the use and loss of information bits. The proposed reversible 16-bit ALU reduces the information bits use and loss by reusing the logic information bits logically and realizes the goal of lowering power consumption of logic circuits. Programmable reversible logic gates are realized in Verilog HDL, simulated and synthesized using Cadence NCSIM and RTL complier.

Keywords - Reversible logic gate, garbage output, toffoli gate, reversible ALU.

I. INTRODUCTION

Recent advances in reversible logic allow for improved quantum computer algorithms and schemes for corresponding computer architectures. Reversible logic is widely being considered as the potential logic design style for implementation in modern nanotechnology and quantum computing with minimal impact on physical entropy. Significant contributions have been made in the literature towards the design of reversible logic gate structures and arithmetic units. However, there are not many efforts directed towards the design of reversible ALUs. The Binary logic circuits built using traditional irreversible gates inevitably lead to energy dissipation, regardless of the technology used to realize the gates.

The power dissipation in any future CMOS will lead to an impossible heat removal problem and thus the speeding-up of CMOS devices will be impossible at some point of time in near future. According to the theorem of Landauer [1], if we do not consider the factors of technology and material in the manufacture of computer, the energy consumption in computer is mainly produced by the logical irreversible operations. The commonly used gate-AND gate, which has two inputs and one output, one bit lost when the information bits go through this gate. For every bit of information loss, there will generate kT ln2 joules, where k is the Boltzmann's constant and T is the absolute temperature. The generation of heat is inevitable on logic because of the uses of traditional logic gates in computer.

The loss of energy can be minimized or even prevented by applying the principle of reversibility to the operation of digital circuits. It can be shown that for power not to be dissipated, it is necessary to build a circuit from reversible gates. This solution based on the reversible logic promises a circuit operation with arbitrarily small fraction of signal energy. The key point of reversible computing is that the electric charge on the storage cell consisting of transistors is not permitted to flow away when the transistor is switched. Then it can be reused through reversible computing, which can decrease the energy consumption. When there is no loss of information bits, then the system is reversible. In VLSI circuits, it means that the circuits consisting of AND and OR gate, the bit information presented by charge can be saved when it is not used, which leads to the reversibility of the system. Therefore, reversible computing is an appealing solution in many emerging fields such as nanotechnology, as well as quantum and optical computing.

II. THE REVERSIBLE LOGIC GATE

A circuit is said to be reversible if the input vector can be uniquely recovered from the output vector and there is a one to one correspondence between its input and output assignments, i.e. not only the outputs can be uniquely determined from the inputs, but also the inputs can be recovered from the outputs. Thus the number of inputs and outputs in reversible gates are equal. Any arithmetic logic unit must be able to produce a variety of logic outputs based on inputs determined by the programmer for implementation in an instruction set architecture. Therefore, reversible logic devices used in an environment must have both fixed select input lines that receive opcode signals manipulated by the programmer and permanent output lines where the result of the logical output is produced.

For an *n* input/output logic gate, if there is a one-to-one correspondence between its inputs and outputs, then this

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 10, October 2013

logic gate is reversible. The corresponding expresses are as follows:

$$I_v = (I_1, I_2, \ldots\ldots, I_n) \quad (1)$$
$$O_v = (o_1, o_2, \ldots\ldots, o_n) \quad (2)$$

where the I_v is input vector and O_v is the output vector. That is to say, a reversible gate has the same number of inputs and outputs.

Commonly used reversible gates are NOT gate, CNOT gate, CCNOT gate (Toffoli gate), Peres gate, Fredkin gate, Feynman gate, etc... Figure 2.1a and 2.1b shows the classical gate (irreversible gate) and general N*N reversible gate.

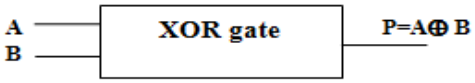

Fig. 2.1a: Classical (Irreversible) gate

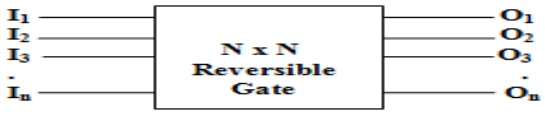

Fig. 2.1b: N x N Reversible gate

In the reversible XOR gate there is no loss of information bit signals. Since it maps the input vector with output vector which gives the equal number of inputs and output and it is shown in Figure 2.1c.

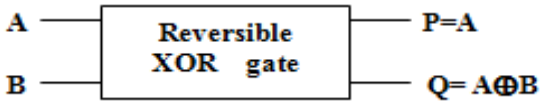

Fig. 2.1c: Reversible XOR (CNOT) gate

Peres gate is represented as 3×3 vector in Figure 2.1d. In the proposed design, Peres gate is used because of its lowest quantum cost. The input vector is I (A, B, C) and the output vector is O (P, Q, R). The output is defined by P = A, Q=A⊕B and R=AB⊕C. Quantum cost of a Peres gate is 4.

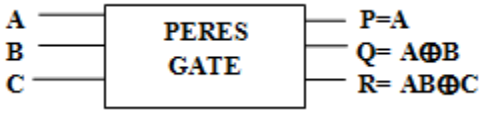

Fig. 2.1d: 3x3 Peres gate**.**

The input vector of Feynman gate is I (A, B) and the output vector is O (P, Q). The outputs are defined by P=A, Q=A ⊕ B and it is shown in Figure 2.1e. Quantum cost of a Feynman gate is 1.

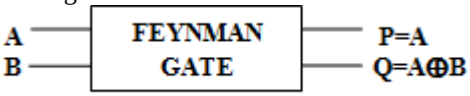

Fig. 2.1e: 2x2 Feynman gate.

Figure 2.1f shows a 3×3 Fredkin gate. The input vector is I (A, B, C) and the output vector is O (P, Q, R). The output is defined by P=A, Q=A′B⊕AC and R=A′C⊕AB. Quantum cost of a Fredkin gate is 5.

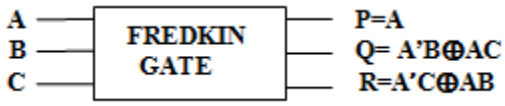

Fig. 2.1f. 3×3 Fredkin gate.

The 3*3 Toffoli gate is represent in Figure 2.1g. The input vector is I(A, B, C) and the output vector is O(P,Q,R).The outputs are defined by P=A, Q=B, R=AB⊕C. Quantum cost of a Toffoli gate is 5.

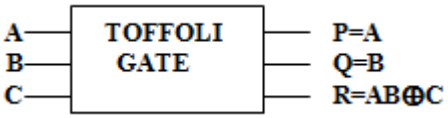

Fig. 2.1g. 3×3 Toffoli (CCNOT) gate.

III. DESIGN OF ARITHMETIC LOGIC UNIT

Arithmetic and logic unit (ALU) is a data processing unit, which is an important part of CPU. There are various types of CPUs are available but every CPU contains an ALU.

3.1 Conventional ALU:

The DM74LS181 is reference logic for proposed design, which is a 4-bit ALU and can perform all the possible 16 logic operations on two variables and a variety of arithmetic operations. ALU provides 16 arithmetic operations: add, subtract, compare, double, plus twelve other arithmetic operations. Provides 16 logic operations of two variables: EXOR, compare, AND, NAND, OR, NOR, plus ten other logic operations.

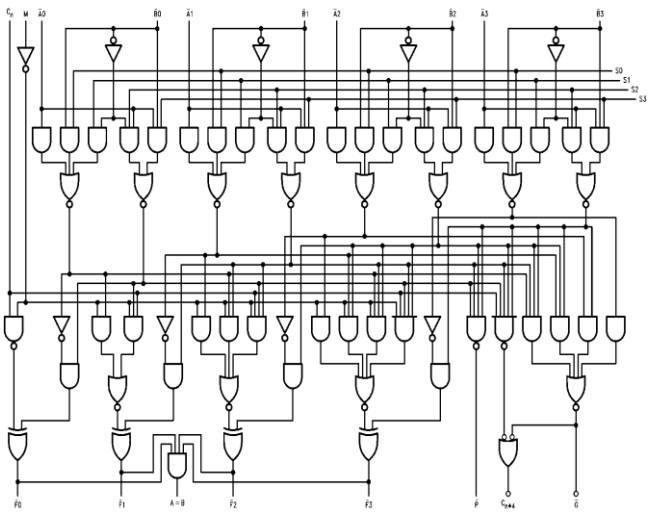

Fig. 3.1: Logic Diagram of conventional ALU

3.2 The ALU Based on Reversible logic:

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 10, October 2013

The multi-function ALU based on reversible logic gates mainly contains the reversible function generator (FUNC) and the reversible controlled unit (DXOR). The reversible function generator and the reversible controlled unit are cascaded by some *n*-Toffoli gates and NOT gates, and arbitrary bit reversible ALU modules can be realized by this way. In the procedure of cascading the reversible function generator and the reversible controlled unit, we reuse the output signals to reduce the cost of circuit design as mush as possible.

3.3 The Reversible Function Generator:

The function generator's feature is to process the input information A_i and B_i under the control of the parameters S_0, S_1, S_2 and S_3, and then we will get the combined functions X_i and Y_i at the output side, where X_i is the combined function on A_i and B_i controlled by the parameters S_3 and S_2 and Y_i is the combined function on A_i and B_i controlled by the parameters S_1 and S_0. From the fig.4.1, we can easily get the function generator logic expressions:

$$X_i = \overline{S_3 A_i B_i + S_2 A_i \overline{B_i}} \quad (3)$$

$$Y_i = \overline{A_i + S_0 B_i + S_1 \overline{B_i}} \quad (4)$$

According to the above logical expression, the reversible function generator is shown in Figure 3.2, and its corresponding package diagram is shown in Figure 3.3.

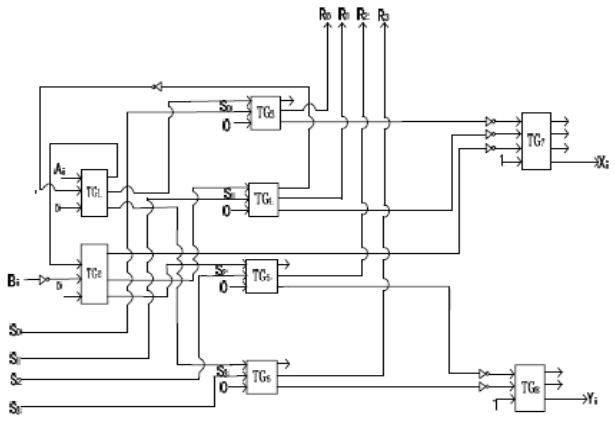

Fig. 3.2: The proposed reversible function generator

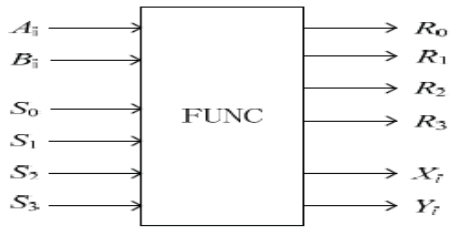

Fig. 3.3: The package diagram of reversible function generator

3.4 The Reversible Controlled Unit DXOR:

The reversible controlled unit DXOR shown in Figure 4.4 is to complete the sum of the three inputs P,Q and C_i, where $P = X_i$, $Q=Y_i$. That is to say, let the combined functions X_i and Y_i from the reversible function generator add the carry signal C_i to get the final result F_i .

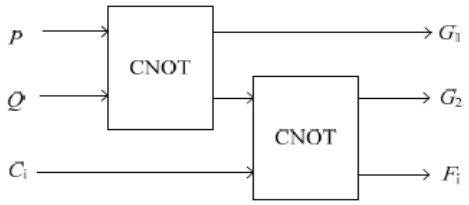

Fig. 3.4: The diagram of DXOR

The express of the reversible controlled unit DXOR is:

$$F_i = P \text{ xor } Q \text{ xor } C_i \quad (5)$$

where signals *P* and Q come from the reversible function generator's signals X_i and Y_i and C_i is the carry signal, F_i is the final output result.

The reversible controlled unit DXOR with 3 input/output bits uses two 2*2 Toffoli gates and produces 2 garbage outputs and its corresponding package diagram is shown in Figure 3.5.

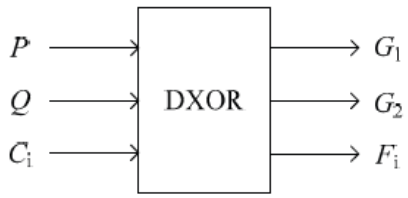

Fig. 3.5: The package diagram of DXOR

3.5 The Realization of Reversible ALU:

To design the reversible ALU with the minimum cost, we choose 3*3 Toffoli gates and NOT gates to cascade the reversible function generator and the reversible controlled unit. Further, the arbitrary bit reversible ALU modules can be realized through cascading. The reversible ALU in Figure 3.6 is cascaded by reversible function generators, reversible controlled units and 3*3 Toffoli gates. The reversible ALU performs operations on to binary numbers $A = (A_7, \ldots\ldots A_0)$ and $B = (B_7, \ldots\ldots B_0)$.

In Figure 3.6, the first operand *A*, the second operand *B*, the control signals S_0 to S_3, the low carry signal C_i and the control signal M are reversible ALU's input signal, while the result, $F = (F_7, \ldots\ldots F_0)$, the carry output signal C_{out} and the garbage outputs (no letters marked) are the output signals. In addition the outputs R_0, R_1, R_2 and R_3 of reversible function generator $FUNC_0$ are respectively seen as the inputs S_0, S_1, S_2 and S_3 of reversible function generator $FUNC_1$ and so on.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 10, October 2013

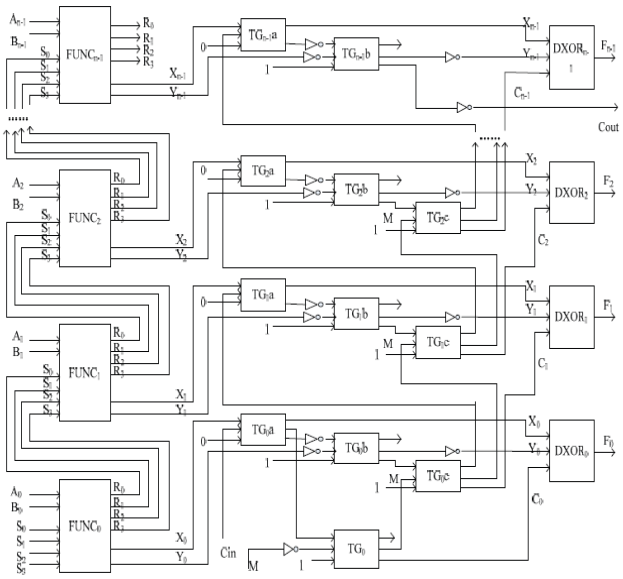

Fig. 3.6: The design of 16 bit reversible ALU

The value of the first and the second inputs of reversible controlled unit $DXOR_0$ – $DXOR_7$ are equal to the value of outputs X_i and Y_i of function generator $FUNC_0$ –$FUNC_7$. The value of the third input C_i of reversible controlled unit $DXOR_0$ – $DXOR_7$ have some connection with the control signal M.

Their relationship can be expressed as follows:
When i = 0 then

$$C_0 = \overline{C_{in}} + M \quad \text{.....(5)}$$

When i = 1 then

$$C_1 = \overline{Y_0 + X_0 C_{in}} + M \quad \text{....(6)}$$

When i>2 then

$$C_i = \overline{Y_i + X_i C_{i-1}} + M \quad \text{.....(7)}$$

The outputs signals X_i and Y_i are generated after the input signals A_i, B_i and S_0 to S_3 passed through the reversible function generator X_i and Y_i as the first two input signals of reversible control unit $DXOR_i$ respectively, and also make operation with control signal M to get the third input signal C_i of reversible control unit $DXOR_i$ based on the equation 5 to 7.

IV. SIMULATION RESULTS

All the blocks of reversible ALU are designed using Verilog HDL (structural form of coding). Verilog code is simulated using Cadence NCSim simulator version 9.10. The input/ output signals are represented by variables, packaging the variety of Toffoli gates by module, it can be instantiated as the object of Toffoli gate module. The results simulated from the programs are same as the results in circuits. So the simulation results can be the base of testing the validation of circuits. Few ALU operations are as shown in the simulation results below.

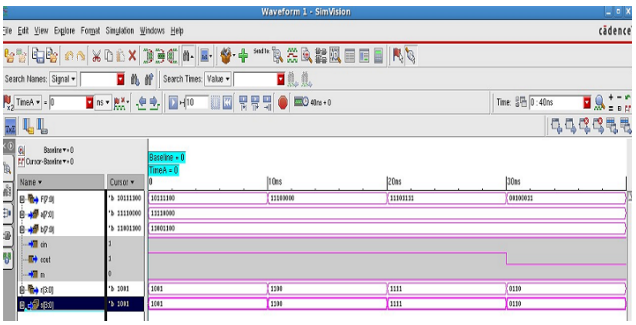

Fig. 4.1: Simulation of arithmetic operations for 8-bit ALU

The figure 4.1 shows the arithmetic operations “A plus B”, “A plus A”, “A minus 1”, and “A minus B minus 1”. The control inputs for these operations are M = ‘0’ and C_{in}= ‘1’ and output is F and C_{out}. The data streams are A = “11110000” and B = “110011001100”.

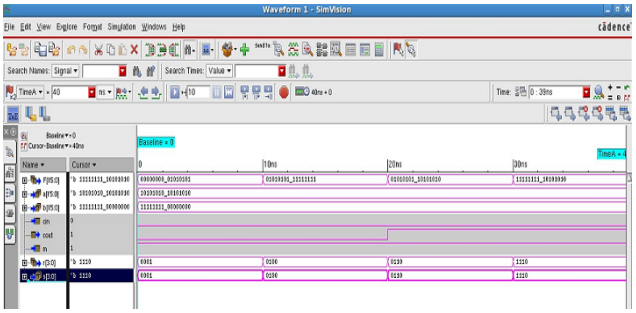

Fig. 4.2: Simulation of logical operations for 16-bit ALU

The Figure 4.2 shows logical operations XOR, NAND, XOR and OR functions for data stream A = “1010101010101010” and B = “1111111100000000”. The control inputs for these operations are M = ‘1’ and C_{in}= ‘0’ and outputs are F and C_{out}.

V. SYNTHESIS REPORTS

The Verilog code is synthesized using Cadence RTL compiler for the delay and power analysis. Table 1 shows the delay and power analysis for 8-bit & 16 –bit reversible ALUs. It is deciphered from the below table 1 that as the number of bits increases the delay and the power increases proportionately.

Table 1. Delay and Power Analysis of Reversible ALU

	Reversible ALU (8 Bit)	Reversible ALU (16 Bit)
Power (µW)	14.5	31.3
Delay (nS)	1.3	2.5

Table 2 shows power analysis of reversible and conventional 16 bit ALU. Reversible 16 bit ALU shows 11% power reduction in comparison with conventional 16 bit ALU.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 10, October 2013

Table 2. Power analysis of 16 bit conventional and reversible ALU

	Reversible ALU (16 Bit)	Conventional ALU (16 Bit)
Power (μW)	31.30	33.83

VI. CONCLUSION

In this paper we propose design and synthesis of 16-bit reversible ALU using reversible logic gates instead of conventional gates. The reversibility significantly reduces the use and loss of information bits hence optimal power consumption. The performance checks of various modules are carried out using Cadence tools. Circuits designed using reversible logic showed a reduced power and delay. The discussion has focused on logical reversibility-the inputs and outputs that can be uniquely recovered from each other.

Future work one can also try the other aspects of reversible logic-physical reversibility, which is related to the key issue as to whether we can build physical gates and circuits that can actually operate backward and dissipate almost zero power. Still we are facing many challenges before actually turning reversible logic into a practical competitive technology. We are also short of efficient simulation, synthesis, testing and verifying tools for designing reversible logic.

REFERENCES

[1] R. Landauer, "Irreversibility and Heat Generation in the Computing Process," IBM Journal of Research and Development, 1961, vol: 5, DOI: 10.1147/rd.53.0183, pp.183-191.

[2] C.H. Bennett, "Logical Reversibility of Computation", IBM Journal of Research and Development, 1973, Vol. 17, DOI: 10.1147/rd.176.0525, pp. 525-532.

[3] A. Khazamipour and K. Radecka, "Adiabatic Implementation of Reversible Gates", Proceedings of IEEE Midwest Symposium on Circuits and Systems (MSCAS), 2005, vol. 1, DOI: 10.1109/ MWSCAS.2005.1594096, pp.291-294.

[4] Matthew Morrison and Nagarajan Ranganathan, "Design of Reversible ALU based on Novel Programmable Reversible Logic Gate Structures", IEEE Computer Society Annual Symposium on VLSI, 2011, vol. 3, DOI: 10.1109/ISVLSI.2011 .30, pp.126-131.

[5] A. Khazamipour and Katarzyna Radecka "A New Architecture of Adiabatic Reversible Logic Gates", Proceedings of IEEE North East Workshop on Circuits and Systems (NEWCAS), 2006, DOI:10.1109/NEWCAS.2006.250900, pp.233-236.

[6] Himanshu Thapliyal and Nagarajan Ranganathan, "Design of Efficient Reversible Binary Subtractors Based on A New Reversible Gate", IEEE Computer Society Annual Symposium on VLSI, 2009, DOI: 10.1109/ISVLSI.2009.49, pp.229-234.

[7] Zhijin Guan, Wenjuan Li, Weiping Ding, Yueqin Hang, and Lihui Ni, "An Arithmetic Logic Unit Design Based on Reversible Logic Gates", IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing, 2011, DOI: 10.1109/PACRIM.2011.6033020, pp.925-931.

[8] Bruce, J.W., M.A. Thornton, L. shivakuamaraiah, P.S. kokate and X. Li, "Efficient adder circuits based on a conservative reversible logic gate", IEEE Computer Society Annual Symposium on VLSI, Pittsburgh, Pennsylvania, 2000, DOI: 10.1109/ISVLSI.2002.1016879 pp. 83-88.

[9] Zeljko, zilic, Radecka.K and Kzamipur.A "Reversible circuit technology mapping from non reversible specifications", Design, Automation and Test in Europe Conference and Exhibition, 2007, DOI: 10.1109/DATE.2007.364652, pp.1-6.

[10] Madhusmita Mahapatro, M. Suresh, Panda S.K, Saheel M, Panda A.K, and Shukla M.K, "Design of Arithmetic Circuits Using Reversible Logic Gates and Power Dissipation Calculation", International Symposium on Electronic System Design (ISED), 2010, DOI: 10.1109/ISED.2010.25, pp.85-90.

[11] D.P. Vasudevan, P.K. Lala and J.P. Parkerson "CMOS Realization of Online Testable Reversible Logic Gates", IEEE Computer Society Annual Symposium on VLSI, 2005, DOI:10.1109/ISVLSI.2005.23, pp.309-310.

[12] Thapliyal, and Vinod, A.P. "Design of Reversible Sequential Elements with Feasibility of Transistor Implementation", IEEE International Symposium on Circuits and Systems, 2007, DOI: 10.1109/ISCAS.2007.378815, pp.625-628.

[13] Frehse, S. Wille, and R. Drechsler, R. "Efficient Simulation-Based Debugging of Reversible Logic", IEEE International Symposium on Multiple Valued Logic (ISMVL), 2010, DOI: 10.1109/ISMVL.2010.37,pp.156-161.