Design of ALU using Reversible Logic and Optimization of Quantum Cost

Manpreet Kaur Walia, Balwinder Singh Dhaliwal

***Abstract*— Reversible logic has received great attention in the recent years due to their ability to reduce the power dissipation which is the main requirement in low power VLSI design. Quantum computers are constructed using reversible logic circuits. It has wide applications in low power CMOS, Optical information processing, DNA computing, quantum computation and nanotechnology. Their implementation using Quantum Gates further enhances their prospect for next generation VLSI applications.**

***Index Terms*—** Reversible logic, quantum computation, DNA Computing

I. INTRODUCTION

Today's computers erase bit of information every time they perform a logic operation. These logic operations are therefore called irreversible. They dissipate energy into the environment with every bit erased. The amount of energy dissipated into the environment is at least kT where k is a universal constant known as the Boltzmann’s constant and T is the temperature of the environment of the computer. For room temperature, this is about 2.9×10^{-21} joules. This is roughly the kinetic energy of a single air molecule at room temperature. In 1960, R. Landauer [1] demonstrated that high technology circuits and systems constructed using irreversible hardware result in energy dissipation due to information loss. Bennett[2] also showed that in order to keep a circuit away from dissipating any power, it had to be composed of reversible gates. A computation is called reversible if its inputs can always be deduced from its outputs. It’s only irreversible computations that must dissipate heat. The energy consumption in computation turns out to be deeply linked to the reversibility of the computation.

There are six important parameters for determining the complexity of the circuit. These are:

- Gate count - that defines the number of reversible gates used in the circuit.
- Garbage output - that defines the outputs that are not used for further computations.
- Quantum cost - that defines the number of 1x1 or 2x2 gates that are used in the circuit.
- Logical calculation - Related to hardware complexity representing the number of NOT and two input XOR and AND gates required to implement the logic of circuit.
- Quantum depth - that defines the quantum cost of the longest path from input to output.
- Constant inputs - that defines the inputs which are to be maintained constant at 1 or 0 throughout the circuit operation depending upon the function required from the gate or circuit.

An ALU is very important part of a computer. It is basically considered as the heart of a computer. It allows the computer to add, subtract, multiply, divide and perform many other arithmetic and logic functions. Since every computer needs to be able to do these functions, they are always included in a CPU. A simple ALU consists of two operands, one control signal to select the operation to be performed and one output signal to give the result of desired operation[3]. Reversible ALU is designed for modular arithmetic operations apart from logical operations.

The rest of the paper is structured as follows-Section II contains the overview of reversible gates. Section III decribes the proposed design of ALU. Section IV is the results and discussion part followed up by conclusion in section V.

II.OVERVIEW OF REVERSIBLE GATES

A. NOT Gate

NOT gate is the only conventional gate that is reversible. It has one input and one output which is the basic requirement of reversible logic. NOT gate is a 1*1 reversible gate. It has a zero Quantum Cost. NOT gate is shown as follows in Fig.1. In 1*1 NOT gate, there is one input i.e. A and there is one output i.e. P which is inverse of A that means if the value of A is 0 then output P is 1 and if A is 1 then P is equal to 0.

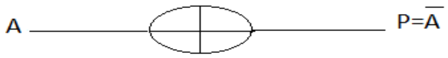

Fig. 1. NOT gate

Manuscript received Aug 21, 2014

Manpreet Kaur Walia, VLSI Design, Guru Nanak Dev Engineering College, Ludhiana, India
Balwinder Singh Dhaliwal, ECE, Guru Nanak Dev Engineering College, Ludhiana, India

B. Feynman gate

Feynman gate is a 2*2 reversible gate as shown in Fig. 2[4]. It is a reversible version of the XOR gate and also called as Controlled NOT gate. It can be defined by preserving one of the inputs. Since it is a 2x2 gate, it has a quantum cost of 1.

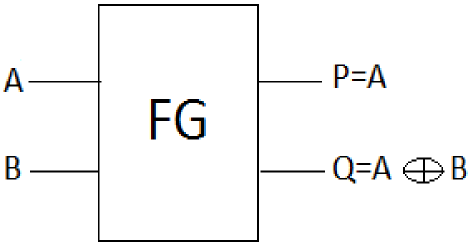

Fig. 2. Feynman or CNOT gate

C.Toffoli Gate

Toffoli gate is a 3*3 reversible gate as shown in Fig. 3[5]. Its two outputs i.e. P and Q are same as two inputs A and B and third output R is equal to A.B xor C. Toffoli gate is considered as universal gate as it is the fundamental gate to be used to realize any 3x3 gate. The quantum cost of toffoli gate is 5.

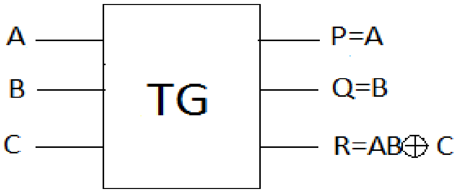

Fig. 3.Toffoli gate

D. Peres Gate

Peres gate is a 3*3 reversible gate as shown in Fig. 4[6]. One of its output P remains same as input A. Second output Q is equal to A xor B and third output R is equal to A.B xor C. Among the 3x3 reversible gates, it has the minimum quantum cost of 4.

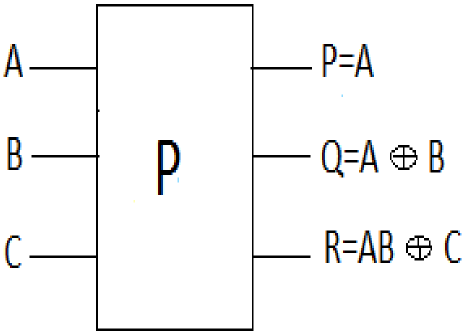

Fig. 4. Peres gate

E. Fredkin Gate

Fredkin gate is a 3*3 reversible gate as shown in Fig. 5[7]. In this the input A is obtained as first output. Inputs B and C are swapped to get the second and third outputs, which is controlled by A. If A = 0, then the outputs are simply duplicates of the inputs otherwise, if A = 1, then the two input lines (B and C) are swapped. The quantum cost of Fredkin gate is 5.

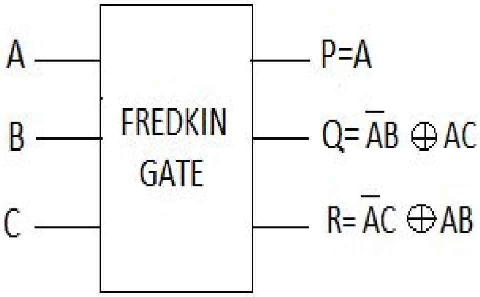

Fig. 5. Fredkin gate

III. PROPOSED WORK

This section details about the implementation of reversible 4 bit ALU using reversible gates such as toffoli gate and fredkin gate which is effective in terms of number of reversible gates, number of garbage outputs, number of constant inputs, quantum cost and hardware complexity.

Implementation of Reversible 4 Bit ALU

In the reversible 4 bit ALU, 13 inputs and 13 outputs are used in which we have four constant inputs - I_1, I_2, I_3, I_4 , nine inputs- A, B, C, D, E, F, G, H, I, eight outputs – O_1, O_2, O_3, O_4, O_5, O_6, O_7, O_8 and five garbage outputs – G_1, G_2, G_3, G_4, G_5 as shown in Fig. 6. When I is equal to 0 then AND operation is performed and when I is equal to 1 then OR operation is performed.

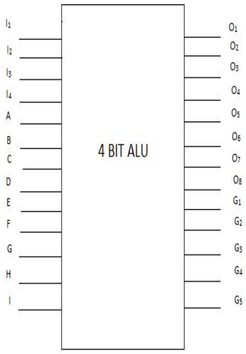

Fig. 6: Reversible 4 Bit ALU

In the implementation of an ALU, first step is to create the PPRM (Positive Polarity Reed Muller) file in the software Relos[8]. PPRM file of 4 bit ALU is shown in Fig. 7.

```
# pprm for  alu4bit
.model alu4bit
.inputs i1 i2 i3 i4 a b c d e f g h i
.z 1 1 1 1 0 0 0 0 0 0 0 0 0
.p 0000000000001 0000000000001
.p 0000000000010 0001000000010
.p 0000000000011 0001000000000
.p 0000000000100 0010000000100
.p 0000000000101 0010000000000
.p 0000000001000 0100000001000
.p 0000000001001 0100000000000
.p 0000000010000 1000000010000
.p 0000000010001 1000000000000
.p 0000000100000 0001000100000
.p 0000000100001 0001000000000
.p 0000000100010 0001000000000
.p 0000001000000 0010001000000
.p 0000001000001 0010000000000
.p 0000001000100 0010000000000
.p 0000010000000 0100010000000
.p 0000010000001 0100000000000
.p 0000010001000 0100000000000
.p 0000100000000 1000100000000
.p 0000100000001 1000000000000
.p 0000100010000 1000000000000
.end
```

Fig. 7. PPRM File of Reversible 4 Bit ALU

After the PPRM file has been created, the netlist for reversible 4 bit ALU is created by using the software Relos. The netlist formed shows that 20 Toffoli gates are used for implementation of reversible 4 bit ALU. Netlist file is shown in the Fig. 8.

```
# Netlist generated by command 'write_netlist'
in ReLoS

.model alu4bit
.inputs i1 i2 i3 i4 a b c d e f g h i
.gate TOF - - - x - - - - - - - 1 -
.gate TOF x - - - - - - - 1 - - - 1
.gate TOF x - - - 1 - - - - - - - -
.gate TOF - - - x - - - - - - - 1 1
.gate TOF - - - x - - - 1 - - - - -
.gate TOF - - - x - - - 1 - - - - 1
.gate TOF x - - - 1 - - - 1 - - - -
.gate TOF x - - - 1 - - - - - - - 1
.gate TOF - - - x - - - 1 - - - 1 -
.gate TOF - - x - - - - - - - 1 - 1
.gate TOF - - x - - - - - - - 1 - -
.gate TOF - - x - - - 1 - - - - - -
.gate TOF - - x - - - 1 - - - - - 1
.gate TOF - - x - - - 1 - - - 1 - -
.gate TOF - x - - - - - - - 1 - - 1
.gate TOF - x - - - - - - - 1 - - -
.gate TOF - x - - - 1 - - - - - - -
.gate TOF - x - - - 1 - - - - - - 1
.gate TOF - x - - - 1 - - - 1 - - -
.gate TOF x - - - - - - - 1 - - - -
.end
```

Fig. 8. Netlist of Reversible 4 Bit ALU

IV. RESULTS AND DISCUSSION

The implementation of design is done on Ubuntu 12.04 (LTS) / Relos tool which simulates and synthesizes the reversible logic circuit based on positive-polarity Reed-Muller expressions[9]. After the netlist is made, then the testing of the operations is done to check whether it is functioning correctly or not. Simulation is also done in the Relos tool. Simulation results of reversible 1 bit ALU are shown as follows in Figure 9.

```
ReLoS>                              read_cost_table
./benchmarks/quantum_cost.txt

ReLoS> read_pprm alu4bit.pprm

ReLoS> syn -f
time elapsed: 180.000 seconds
9 0000000000010
9 0000000000011
9 0000000100000
9 0000000100001
9 0000000100010
12 0000100000001
12 0000100010000
12 0000100000000
12 0000000010001
10 0000000000101
10 0000000000100
10 0000001000000
10 0000001000001
10 0000001000100
11 0000000001001
11 0000000001000
11 0000010000000
11 0000010000001
11 0000010001000
12 0000000010000
best solution = 20

ReLoS> write_netlist alu4bitperes.net
ReLoS> print_stats

I/O: 13   # PPRM terms: 29   #    Gates:    20
   Cost: 68
```

CONCLUSION

Reversible 4 bit ALU is designed successfully using toffoli gates. Such circuits can be helpful not in terms of hardware complexity and power saving but also help in reducing cost. Any number of operations can be performed using these ALU’s. More complex ALU’s can be implemented using the methodology followed in the above ALU’s.

REFERENCES

[1]R. Landauer, “Irreversibility and Heat Generation in the Computing Process”, *IBM Journal of Research and Development*, vol. 5, 3, pp. 183-191, 1961.

[2]C. H. Bennett, “Logical reversibility of computation*”,* *IBM J. Research and Development*, 17, November 1973, pp:525 – 532.

[3]M. K. Thomson, Robert Gluck and Holger Bock Axelsen, “Reversible Arithmetic Logic Unit for Quantum arithmetic”, *Journal of Physics A:Mathematical and Theoritical*. 43 (2010).

[4]Feynman, R., “Quantum mechanical computers”, *Optical News*, vol. 11, pp. 11-20, 1985.

[5]Toffoli, T., “Reversible computing”, *Technical Memo MIT/LCS/TM-151, MIT Laboratory for Computer Science (February*), 1980.

[6]Peres, A.,“Reversible logic and quantum computers”, *Physical Reviews*. A, 32: 1985, pp: 3266-3276.

[7]Fredkin, E. and Toffoli, T.,"Conservative Logic," *International Journal of Theoretical Physics,* vol. 21, pp 219-53,1980.

[8]Gupta, P.; Agrawal, A. and Jha, N. K., *“An Algorithm for Synthesis of Reversible Logic Circuits,*” *IEEE Trans. CAD*, vol. 25, 11, pp. 2317–2330, 2006.

[9]A. Agrawal and N. K. Jha , *“Synthesis of Reversible Logic,*” Proceedings of Design Automation & Test, Europe, vol. 2, pp. 1384–1385, 2004.