Abstract—Two unique, reversible Arithmetic Logic Units (ALU) have been designed in this paper. The number of functionalities of the ALU has been maximized and the cost metrics, minimized. Each functionality is realized using the least possible number of gates and incorporated together to get the optimal design. The ALU designs so obtained have AND, XOR, NOR, NOT and SUM functionalities with reduced optimized values of line costs, gate costs and transistor costs, which translates into lesser chip area for fabrication. The inability of the software to simulate complex V and V+ gates, increases the quantum cost marginally. However this tradeoff can be negated during actual fabrication. The platform used for the simulation of all reversible gates and reversible ALUs, is the ‘Revkit’ which is a package of iPython.

Index Terms—Arithmetic Logic Unit, Low Power VLSI, Quantum Cost, Reversible Logic.

I. INTRODUCTION

Recent advances in designing chips with high performance have shown that power dissipation is a crucial issue that needs to be tackled. In the recent past, especially the last decade with the advent of more and higher performance chips, the heat dissipation to the surrounding environment of a chip has increased considerably. Now, given that excessive heat will reduce the reliability of a chip (in some cases goes as far as to destroy it), power dissipation thus becomes a crucial impediment to the progress of development of faster and smaller chips. Thus, in order to satisfy the need of increased power for computational purposes, we require alternatives which transcend the scope of regular and traditional technologies such as CMOS. It was found by Landauer [1] that with regard to irreversible logic computations, each bit of information lost generates kTln (2) joules of energy, where k is the Boltzmann’s constant and T is the absolute temperature at which computation is performed. Bennett [2] demonstrated the reverse that, kTln(2) energy dissipation would not occur if the computation were carried out in a reversible manner [3]. Hence, reversible logic marks a new direction where all operations are performed in an invertible manner.

II. MOTIVATION

The greatest concern while designing high-performance chips today, is the controllability of the power dissipation, which increases with chip complexity. Thus in order to design chips of greater complexity, it is an absolute necessity to go beyond traditional technologies. Landauer’s Principle led to the unravelling of a new area of research in the domain of low power digital systems. The advent of reversible logic has created new possibilities and has shown a new direction to future researchers. Reversible logic technology today, has improved the design of quantum computer schemes and the necessary algorithms the corresponding computer architectures. Significant contributions have been made towards the design of arithmetic units and reversible logic gate structures. However, there still hasn’t been much effort which is directed towards the design of reversible arithmetic logic units (ALU). This paper aims to design and simulate an optimized reversible logic ALU with reduced cost metrics and compare its performance with the existing reversible logic ALUs. As the ALU is reversible in nature, it will also reduce the power dissipated.

III. OBJECTIVES

Any arithmetic logic unit should be able to produce a variety of logical outputs, such as OR, AND and XOR, based on inputs which are determined by the programmer for implementation in an instruction set architecture. Therefore, a reversible gate which is used for this purpose must be able to:

- Maximize the types of logical operations it can carry out—namely AND, XOR, NOR, NOT, SUM/SUB etc.
- Minimize the number of logical output lines and select lines—This directly translates to lower line cost and hence lower spatial requirements
- Minimize delay and quantum cost—These two parameters dictate the performance of the ALU and hence require greater focus.
- The ancillary inputs and garbage outputs -inputs and outputs which are not implemented in the design of the gate and only serve to maintain the reversibility of the device – ought to be reduced and ideally eliminated.

The designer of any programmable logic device must also consider which values to be propagated to the output. In certain instances, it may be useful to produce a copy of the input data values, whereas other designers can choose to propagate the input signals to the output signals.

IV. THEORY AND FUNDAMENTALS

A. Basic Quantum Gates

During quantum computation, the state of a qubit (quantum bit) can take not only two states |0⟩ and |1⟩, but also any linear combinations of these states, and this is called superposition, and this results in exponentially larger state space. Quantum gates perform operations on qubits and form the foundation of reversible logic. There are four fundamental
quantum gates:
  - Inverter (NOT): Inverts a single qubit.
  - Controlled inverter (CNOT): If the control qubit is 1, the target qubit is inverted.
  - Controlled V gate: The V operation is also known as the square root of NOT, because two consecutive V operations are equivalent to an inversion.
  - Controlled V+ gate: The V+ gate performs the inverse operation of the V gate, i.e. \( V^+ \equiv V^{-1} \)

B. Cost Metrics

The parameters that are used to calculate the efficiency of a quantum circuit are:
  - Quantum Cost: Controlled V, Controlled V+ and CNOT gates that are used in implementing it are counted to calculate the quantum cost.
  - Transistor Cost: This is denoted by the effort needed, to realize reversible circuits in CMOS.
  - Line Cost: It denotes the total number of Input or output lines used.
  - Gate Cost: Denotes the number of gates the circuit consists of.

V. UNIVERSAL REVERSIBLE QUANTUM GATES

The universal reversible logic gates are including but not limited to the Feynman gate, Toffoli gate, Fredkin gate, Peres gate, etc. The following gates have been used in the design of the ALUs simulated in this paper.

A. Feynman Gate

The Feynman gate also known as a Controlled NOT (CNOT) gate is one of the popular examples of a 2 x 2 reversible gate. As shown in Figure 1, the first input in this gate is passed to the output without any change and the second output is the XOR of the first and second inputs.

B. Toffoli Gate

The Toffoli gate, or a doubly controlled NOT gate, is a (3x3) universal reversible gate. As shown in Figure 2, the first two inputs are directly passed to the corresponding outputs and the third output is the logical XOR of the third input with the logical AND of all the first two inputs. A, B and C are the three inputs to the gate and the corresponding Output lines are X, Y and Z respectively. The corresponding functions computed are as following:

C. Fredkin Gate

The Fredkin gate functions as Controlled Swap Gate where the control signal is the first input. As can be observed in Figure 3, if the control is 1, the other two inputs are swapped. Else, they pass through directly. The functions computed by the outputs of Fredkin’s Gate can be interpreted as follows:

D. Peres Gate

Peres gate is another important gate which has a low quantum cost as compared to other gates. As illustrated in Figure 4, the input vector is \( I = (A, B, C) \) and the output vector is \( O = (P, Q, R) \). It is a single gate that performs the same functions that the Feynman and the Toffoli gates perform individually. The output is defined by:

VI. COMPARISON OF EXISTING REVERSIBLE ALUS

Recent Advances in reversible logic, today allow for
improved quantum computer algorithms and schemes for corresponding computer architectures. Significant contributions have been made in the literature towards the design of arithmetic units and reversible logic gate structures, however, there are not many efforts directed towards the design of reversible arithmetic logic units (ALU). Morrison proposed a reversible ALU which was based on the V-shaped design of the Van Rentergem adder and it produced the following logical outputs: ADD, SUB, NSUB, XOR and NOR. Thomsen’s ALU [9] has a lower quantum cost than the ones proposed by Morrison [4]. However, it has less number of logical operations and a greater worst case delay. In 2011, Morrison proposed two 4x4 programmable reversible gates, MRG and Peres And-Or Gate (PAOG), to be used specifically in the design of reversible ALUs. The other 4x4 gates reversible gates in literature are TSG gate, MKG gate, HNG gate and PFAG gate, which are all full adders that can be used in the design of a reversible ALU. These gates have costs varying from 6-10 and worst case delays of 4-6 units. Two 1-bit ALUs were also proposed in [4]. Both ALUs have 8 inputs and 8 outputs with 3 data lines. The cost of the ALUs is 24 and the worst case delay is 16. The first ALU performs the logical functions: ADD, SUB, XOR, XNOR, OR, and NOR. The second one performs ADD, SUB, AND, NAND, OR and NOR. Different universal reversible gates are used to make the two ALUs. A reversible logic ALU based on Reversible Quantum-dot Cellular Automata gates (RQCA) has been proposed in [7]. This will also be simulated and analyzed along with the other three ALUs. This ALU performs ADD, NOR, NAND, OR, AND, and SUB. It can be observed from these pre-existing ALUs that the gate cost, transistor costs and quantum cost increase as the number of lines increase. Hence, an optimal number of lines must be selected while designing the gate. As the number of operations increases, the costs also increase. The cost depends on the complexity of the operations performed. In the Thomsen ALU, operations such as AND, NAND, OR and NOR have not been performed. Thus the cost is lesser for that ALU. The MRG, PAOG and RQCA based ALUs perform these logical functions and hence their cost is more. As the complexity of the gates used increases, so does the cost. In RQCA gate, the quantum cost, although low, when used in a circuit to make an ALU, increases the total cost of the ALU circuit. Whereas, the MRG and PAOG gates were modified versions of the pre-existing gates and hence the cost was comparatively lower. The traditional ALU design approach used will not yield lesser cost, even if it is functionally correct. The design approach followed takes into consideration all the above results; For each function, the best method for its design is analyzed from the pre-existing ALUs. All such designs are consolidated into an ALU in an optimal manner to make to new ALU. The simplest gates will be used to minimize the costs.

VII. DESIGN PROCEDURE

A. AXONS ALU

As the traditional method of reversible gate design was too lengthy and ineffective, a new approach was adopted. The operations required to be performed by the ALU were taken-AND, XOR, NOR, and SUM. A circuit was made for each operation separately and it was then combined in such a manner so as to get the minimum line and gate cost. A 4:1 reversible multiplexer was used to get the output at one line. The multiplexer utilizes Fredkin gates. If modified Fredkin gates are used for the multiplexer, the quantum cost can be decreased further. The AXONS ALU, shown in Figure 5, was named after the operations it performs – And, XOR, NOR, SUM. It has two select lines and operates according to the truth table as shown in Table 1.

<table>
<thead>
<tr>
<th>S0</th>
<th>S1</th>
<th>OPERATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>SUM/SUB</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>AND</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>XOR</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>NOR</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>NOT</td>
</tr>
</tbody>
</table>

Table 1: Truth Table for AXONS ALU

Fig 5: AXONS ALU Design
The design procedure adopted was identical to the one adopted for the AXONS ALU. The operations required to be performed by the ALU -AND, XOR, NOR, and SUM- were taken and incorporated to obtain another combination. A 4:1 reversible multiplexer was used to get the output at one line. The multiplexer utilizes Fredkin gates. If modified Fredkin gates are used for the multiplexer, the quantum cost can be decreased further. Although the AXONS-II ALU, shown in Figure 6, has more cost than AXONS ALU, it requires only one line for the input ‘B’, whereas, AXONS ALU utilizes two lines for this input. AXONS-II ALU has four select lines and operates according to the truth table, illustrated in Table 2.

<table>
<thead>
<tr>
<th>S4</th>
<th>S3</th>
<th>S2</th>
<th>S1</th>
<th>S0</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>NOR</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>OR</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>AND</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>NAND</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>XOR</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>XNOR</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>SUM/ SUB</td>
</tr>
</tbody>
</table>

Table 2: Truth Table for AXONS-II ALU

VIII. RESULTS

Four existing reversible ALUs were simulated and analyzed using Revkit, namely- Thomson’s ALU, MRG ALU, PAOG-ALU, RQCA-ALU. Based on the results, a novel AXONS ALU was proposed which reduced the line cost, transistor cost and the gate cost. The traditional method of designing a reversible ALU was discarded as it increased the gate cost drastically. A completely new approach was used to design the AXONS ALU. It utilizes a mixture of parallel as well as serial architecture and a multiplexer. However, the number of operations performed is only four, as shown in Table 3, which compares operations of all simulated ALUs.

<table>
<thead>
<tr>
<th>ALU/LOGIC</th>
<th>NUMBER OF OPERATIONS</th>
<th>OPERATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>THOMSEN ALU</td>
<td>5</td>
<td>XOR, NOP, ADD, SUB, NSUB</td>
</tr>
<tr>
<td>MRG ALU</td>
<td>6</td>
<td>XOR, OR, NOR, XNOR, ADD, SUB</td>
</tr>
<tr>
<td>PAOG ALU</td>
<td>6</td>
<td>AND, OR, NOR, NAND, ADD, SUB</td>
</tr>
<tr>
<td>RQCA ALU</td>
<td>6</td>
<td>AND, OR, NOR, NAND, ADD, SUB</td>
</tr>
<tr>
<td>AXONS ALU</td>
<td>6</td>
<td>AND, XOR, ADD/SUB, NOT</td>
</tr>
<tr>
<td>AXONS-II ALU</td>
<td>6</td>
<td>AND, XOR, ADD/SUB, NOT</td>
</tr>
</tbody>
</table>

Table 3: Number of ALU Operations
Table 4: Performance Analysis of ALUs

<table>
<thead>
<tr>
<th>ALU Type</th>
<th>Gate Cost</th>
<th>Transistor Cost</th>
<th>Line Cost</th>
<th>Quantum Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>THOMSEN ALU</td>
<td>5</td>
<td>56</td>
<td>7</td>
<td>17</td>
</tr>
<tr>
<td>MRG ALU</td>
<td>16</td>
<td>128</td>
<td>10</td>
<td>24</td>
</tr>
<tr>
<td>PAOG ALU</td>
<td>16</td>
<td>128</td>
<td>10</td>
<td>24</td>
</tr>
<tr>
<td>RQCA ALU</td>
<td>39</td>
<td>264</td>
<td>16</td>
<td>39</td>
</tr>
<tr>
<td>AXONS ALU</td>
<td>10</td>
<td>80</td>
<td>9</td>
<td>30</td>
</tr>
<tr>
<td>AXONS-II ALU</td>
<td>12</td>
<td>104</td>
<td>11</td>
<td>32</td>
</tr>
</tbody>
</table>

Table 3 compares all the simulated ALUs in terms of the number of operations performed. Table 4 illustrates the performance analysis of the simulated ALUs in terms of the cost metrics. The higher quantum cost of the AXONS ALU can be attributed to the inability to simulate the Modified Fredkin gate on the chosen platform, as it uses complex v-gates. The use of Modified Fredkin gates will bring the quantum cost down to 27. This is an acceptable trade-off, because the reduction of line cost, transistor cost and gate cost, translates directly into cost and space reductions, at the physical fabrication level. While incorporating arithmetic as well as logical operations, the AXONS ALU has been optimized in terms of line cost, gate cost and transistor cost. Thus it is a more efficient successor of the preceding ALUs. The number of operations performed by the ALU can be increased. However, this will come at an increase in all the cost metrics but they will still be less than the RQCA ALU. Some of the outputs of the ALU can be obtained on other output lines of the ALU. These outputs can be used for other purposes as well. Hence, a novel ALU with better cost metrics has been successfully designed. Figure 7 and Figure 8 illustrate simulation of the NOR and XOR operations respectively, carried out in AXONS ALU. Figure 9 and Figure 10 show the simulation of the NAND and NOR operations respectively, carried out in AXONS-II ALU.

IX. CONCLUSION

The functionality of the ALU can be increased by adding more number of operations. But as the number of operations increases, cost also increases. Hence, a balance of operations and cost must be obtained to get newer ALUs with better cost and efficiency. The proposed ALU’s power dissipation analysis and delay calculation can be done using Cadence in the future.
Fig 10: neither Simulation of NOR Operation in AXONS-II ALU

X. ACKNOWLEDGMENT

We would like to express our sincere thanks to our Head of Department, Dr. M. Uttarakumari for giving us this opportunity to undertake this project. We would also like to thank our Principal, Dr. B.S. Satyanarayana, for his wholehearted support. We would like to thank the faculty members and non-teaching staff of Dept. of Electronics and Communication Engineering, RVCE for their constant support.

REFERENCES


AUTHOR’S PROFILE

Chinmaye R. Assistant Professor, Department of Electronics and Communication Engineering, R.V. College of Engineering, Bengaluru
Amrutha C. B.E.-Electronics and Communication Engineering, R.V. College of Engineering, Bengaluru
Vishwanath S. B.E.-Electronics and Communication Engineering, R.V. College of Engineering, Bengaluru
Soumya S. Ghosh. B.E.-Electronics and Communication Engineering, R.V. College of Engineering, Bengaluru
Shouryadipta Sarkar. B.E.-Electronics and Communication Engineering, R.V. College of Engineering, Bengaluru