A Novel Design of Balanced Quaternary Reversible Logic Circuits Using Verilog HDL

Pasupulety Sudheer  
M.Tech,  
Dr.K.V. Subba Reddy Institute of Technology.

P.Jayarami Reddy, M.Tech  
Assistant Professor,  
Dr.K.V. Subba Reddy Institute of Technology.

Abstract:  
Routing has become the main contributor in many areas of design such as area, delay and power. Multiple Valued Logic (MVL) offers a means to reduce the routing since each wire in MVL can carry the twice as much information as single binary wire. Reducing this routing directly leads to the reduction of overall circuit area and power consumption. Multi–valued ternary logic under GF (3) and quaternary logic under GF (4) are available in the literature, but circuit design based on these logic systems is very few. As traditional computing devices based on irreversible logic are approaching their limit in terms of heat dissipation, power and speed requirement. Reversible computing is emerging as an alternative technology. Usage of multi–valued logic for irreversible computing is also growing.

Even today also, the latest computing systems are designed and developed using binary logic only. Then why there is a need of inventing multi valued logic over binary logic??? With the advancing technology, interconnections are the main contributor for delay, area and power consumption. One of the solution for the problem of interconnection is that: what if we will develop the idea of transmitting the number of logic levels through a single wire. Since, multi valued logic scheme allows more data to be grouped in single digit, researchers seems the use of multivalued logic as a solution for the above problem.so, researchers have been working on the idea of using multivalued logic for many years. Existing VLSI technology has put some limitations on the selection of number of logic states, therefore researchers seems the use of Quaternary logic systems to be best in this regards.

In literature, the multivalued ternary (in 3-dimension Hilbert space under GF (3)) [1], quaternary (GF (4)) [2] logics have been discussed by the researchers. To realize the ternary logic circuit three digits (trit) 0, 1 and 2 are used with balanced ternary logic states -1, 0 and 1 [3]. Circuit balancing is necessary for realizing the ternary reversible logic and above. This paper introduces the balancing rules and their implementation mechanism for quaternary reversible logic gates. Quaternary logic circuits realize 4-digits 0, 1, 2 and 3 (quaternary standard states) [2]. The balanced quaternary is a non-standard number system which is useful for comparison with quaternary numbers. In a quaternary system, the balanced quaternary logic contains 4-states -2, -1, +1 and +2. For more precise representation, some conventions are also assumed for balanced quaternary logic states. -2 and -1 states are represented as low state (L) and high

Keywords:  
Balanced quaternary logic, reversible logic gate, m-s gate, half-adder, full-adder, multiplier.

I. INTRODUCTION:  
For many years digital devices have been designed using binary logic.
state (H), respectively. +1 and +2 states are the inverse states of H and L, respectively. All integers can be represented by the balanced quaternary. Any unbalanced quaternary can be converted into balanced quaternary notation by subtracting +2 and +1 from 0, 1 and 2, 3, respectively. For example, \((2301)_{\text{unbal}_4} = (12LH)_{\text{bal}_4} = (177)_{10}\). From the balancing rules of quaternary reversible logic, the realization of balanced quaternary reversible half-adder, full-adder and multiplier is also proposed in this paper. To the best of our knowledge, it is the first attempt for balanced quaternary circuit realization.

The next sections of this paper are arranged as follows: The basics of quaternary algebra are described in section II. Quaternary reversible logic gates and their literature survey are addressed in section III. Section IV has our proposed balanced quaternary operation rules on balanced quaternary reversible gates. Section V has the design of balanced quaternary reversible circuits, half-adder, full-adder and multiplier. The synthesis and simulation results are presented in section VI. The conclusion of the paper and future work is discussed in section VII.

II. THE BASICS OF QUATERNARY ALGEBRA

The set \(Q_{gf}\) has the elements \((0, 1, 2 \text{ and } 3)\) exhibits an algebraic structure of quaternary Galois Field (GF (4)). The two binary operations addition and multiplication are defined in Table 1 (a) and Table 1 (b).

**TABLE I (A)GF(4) ADDITION; (B)GF(4) MULTIPLICATION**

<table>
<thead>
<tr>
<th>(a)</th>
<th>(b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>0 1</td>
<td>0 1</td>
</tr>
<tr>
<td>0 2</td>
<td>0 2</td>
</tr>
<tr>
<td>0 3</td>
<td>0 3</td>
</tr>
<tr>
<td>1 0</td>
<td>1 0</td>
</tr>
<tr>
<td>1 1</td>
<td>1 1</td>
</tr>
<tr>
<td>1 2</td>
<td>1 2</td>
</tr>
<tr>
<td>1 3</td>
<td>1 3</td>
</tr>
<tr>
<td>2 0</td>
<td>2 0</td>
</tr>
<tr>
<td>2 1</td>
<td>2 1</td>
</tr>
<tr>
<td>2 2</td>
<td>2 2</td>
</tr>
<tr>
<td>2 3</td>
<td>2 3</td>
</tr>
<tr>
<td>3 0</td>
<td>3 0</td>
</tr>
<tr>
<td>3 1</td>
<td>3 1</td>
</tr>
<tr>
<td>3 2</td>
<td>3 2</td>
</tr>
<tr>
<td>3 3</td>
<td>3 3</td>
</tr>
</tbody>
</table>

(A1) \(a + (b + c) = (a + b) + c\) (associative law for addition)

(A2) \(a + b = b + a\) (commutative law for addition)

(A3) There is an element 0 such that \(a + 0 = a\) for all \(a\)

(A4) For any \(a\), there is an element \((-a)\) as \(a + (-a) = 0\)

(M1) \(a \cdot (b \cdot c) = (a \cdot b) \cdot c\) (associative law for multiplication)

(M2) \(a \cdot b = b \cdot a\) (commutative law for multiplication)

(M3) There is an element 1 (not equal to 0) such that \(a \cdot 1 = a\) for all \(a\)

(M4) For any \(a \neq 0\), there is an element \(a^{-1}\) as \(a \cdot a^{-1} = 1\)

(D) \(a \cdot (b + c) = (a \cdot b) + (a \cdot c)\) (distributive law)

III. QUATERNARY REVERSIBLE LOGIC GATES AND LITERATURE SURVEY

Before describing the quaternary reversible gates a close look on reversible logic gate is necessary.

A. Reversible Logic Gate:

The permutation of a gate computes a bijective function then a gate is a reversible. There must be exist a one-one and onto correspondence between its inputs and outputs. The logic values 0 and 1 are computed in binary reversible gates whereas logic values 0, 1, 2 and 3 are accepted in quaternary reversible gates. The famous reversible gates are NOT gate Feynman gate (CNOT gate) [4], Toffoli gate(C\textsuperscript{2} NOT gate) [5] and Fredkin gate [6]. The representation of these gates is described in Fig. 1 (a) – (d).

![Fig. 1. Reversible Gates (a) NOT Gate (b) Feynman Gate (c) Toffoli Gate](image-url)
The NOT gate inverts its input to the output. In the Feynman gate, the 1st input is unchanged, and the state of the 2nd input is inverted when the 1st input is 1. Same in Toffoli gate, first two inputs unchange their states and the 3rd input is inverted when both inputs are 1. In the Fredkin gate, the 1st input is unchanged, and 2nd and 3rd inputs are interchanged when the first input is 1. Quaternary reversible gates are NOT gate, Shift gates, Feynman gate, Toffoli gate and Muthukrishnan–Stroud Gate (M-S gate) [7] etc. There are 24 (4!) shift gates in quaternary reversible logic, but these are more, so here we show only four quaternary reversible shift gates with their symbols.

![Fig. 2. Shift Gates](image)

Here addition (+) and multiplication (x or denoted by dot/absent mark) operations are followed over modulo 4 or GF (4). These all 24 shift gates are 1-qudit gates. These gates are represented by the following Fig. 2 and its simple symbolic representation is shown in Fig. 3.

![Fig. 3. Representation of quaternary reversible 1-qudit gate](image)

**TABLE II TRUTH TABLE OF QUATERNARY MUTHUKRISHNAN-STROUD GATE**

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Fig. 4. Quaternary Muthukrishnan–Stroud Gate

Here Z-transform is the process of translating the controlled-input, when the controlling-input is 3. The above Fig. 4 shows the family of M-S gate.

![Fig. 5 and Fig. 6 show the symbols of standard quaternary logic gates quaternary Feynman gate.](image)

B. Quaternary 2-qudit Muthukrishnan-Stroud Gate family:
The liquid ion trap technologies [8] are used to realize the 2-qudit multi-valued logic Muthukrishnan-Stroud gate. It is shown but not tested in the laboratory.

Thus, from a careful survey of existing literature on quaternary reversible circuits, it can be summarized that the realization of all basic reversible gates and reversible circuits (half-adder, full-adder and multiplier) are proposed using M-S primitive gate. It is observed in previous proposed quaternary circuits that the shift gate counts and M-S primitive gates are too high. Hence, the hardware complexity of these circuits will also be high.

In this article, our goal is to reduce the hardware complexity of the circuits using balanced quaternary reversible logic. To the best of our knowledge no significant effort has been found on the realization of balanced quaternary reversible circuits. The balanced quaternary logic can represent both positive and negative numbers without complement operation.

### TABLE III  UNITARY MATRIX OF FEYNMAN GATE

<table>
<thead>
<tr>
<th>INPUT</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>1</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
</tr>
</tbody>
</table>

C. Unitary Matrix

Unitary Matrix (U) is an $n \times n$ matrix which have the same number of rows & columns. Here the row and column means the input and output. The conjugate transpose of unitary matrix is $U^*$ and $UU^* = U^*U = I$, here $I$ is the identity matrix. The Input Output relationship of the reversible gate / circuit is the unitary matrix of that gate. For example, the unitary matrix of Feynman gate is shown in table III.

### IV. BALANCED QUATERNARY REVERSIBLE LOGIC GATES

For the design of quaternary balanced reversible gates four states $-2$, $-1$, $+1$ and $+2$ are used for corresponds to $0$, $1$, $2$ and $3$. The unique column vector representations of these states are represented by Fig. 7.

![Fig. 7. Vector Representation of Balanced Quaternary States](image)

### A. BALANCED QUATERNARY NOT GATE

It inverts the input to the corresponding output of the input, e.g. $-2$ inverts into $+2$ and $-1$ inverts into $+1$. Fig. 8 shows the symbol of balanced quaternary reversible NOT gate with its Truth Table.

![Fig. 8. Balanced Quaternary NOT Gate](image)
TABLE IV TRUTH TABLE OF BALANCED QUATERNARY NOT GATE

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>+2</td>
<td>-2</td>
</tr>
<tr>
<td>+1</td>
<td>-1</td>
</tr>
<tr>
<td>-1</td>
<td>+1</td>
</tr>
<tr>
<td>-2</td>
<td>+2</td>
</tr>
</tbody>
</table>

The unitary matrix of the balanced quaternary reversible NOT gate is as-

\[
\begin{pmatrix}
0001 & 0010 & 0100 & 1000 \\
0000 & 0010 & 0100 & 1000 \\
0000 & 0000 & 0000 & 0000 \\
0000 & 0000 & 0000 & 0000
\end{pmatrix}
\]

The fig. 10 shows the balanced quaternary NOT operation e.g. Input -2 produce +2.

\[
\begin{pmatrix}
0001 \\
0010 \\
0100 \\
1000
\end{pmatrix}
\times
\begin{pmatrix}
1 \\
0 \\
0 \\
0
\end{pmatrix}
= 
\begin{pmatrix}
0 \\
0 \\
0 \\
0
\end{pmatrix}
\]

B. BALANCED QUATERNARY FEYNMAN GATE
It is a 2*2 reversible logic gate. The 1st input is unchanged for the 1st output and the 2nd input is changed by the balanced quaternary NOT gate. It is not dependent on the 1st input. Table V and VI shows the operation and Truth Table of balanced quaternary Feynman gate.
Fig. 12. Unitary Matrix of Balanced Quaternary Feynman Gate  Fig. 13. Balanced Quaternary Feynman Operation

Fig. 11, 12 and 14 shows the symbol, unitary matrix and column vector representation of balanced quaternary Feynman gate. Fig. 13 shows the Feynman operation when the Input -2-2 converted into -2+2 Output.

C. BALANCED QUATERNARY TOFFOLI GATE
It is a 3x3 reversible logic gate. The 1st and 2nd inputs are unchanged, and the 3rd input is in put changed by the balanced quaternary NOT operation on the 3rd input to convert 3rd output. It is also not dependent on the 1st and 2nd input which can have any quaternary balanced states. The balanced quaternary Toffoli operations are shown by the table VII.

TABLE VII. BALANCED QUATERNARY TOFFOLI GATE OPERATION

Table showing the balanced quaternary Toffoli gate operation.

V. BALANCED QUATERNARY REVERSIBLE CIRCUIT
QUATERNARY HALF-ADDER:
An n*n balanced reversible logic circuit is designed by the use of the balanced reversible logic gate. Here we are designing some balanced quaternary reversible logic circuit by the help of the balanced quaternary reversible logic gates. The designing of balanced quaternary reversible half adder, full adder and multiplier circuits are designed here by the specific rules of circuits.

Fig. 15 shows the symbol of balanced quaternary Toffoli gate, which is newly proposed, without Truth Table (table must be too large, so here we are not showing its truth table).

Fig. 15. Symbol of Balanced Quaternary Toffoli Gate
A. BALANCED QUATERNARY REVERSIBLE ADDER

In balanced quaternary logic, the states -2, -1, +1 and +2 are represented by the --, -+, + and ++, respectively. By these conventional, the addition table is shown following.

TABLE VIII. BALANCED QUATERNARY ADDITION TABLE

<table>
<thead>
<tr>
<th></th>
<th>--</th>
<th>-+</th>
<th>+</th>
<th>++</th>
</tr>
</thead>
<tbody>
<tr>
<td>--</td>
<td>--</td>
<td>-+</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>-</td>
<td>-+</td>
<td>++</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>+</td>
<td>++</td>
<td>-</td>
<td>-+</td>
<td></td>
</tr>
<tr>
<td>++</td>
<td>-+</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

B. DESIGNING OF BALANCED QUATERNARY HALF ADDER

It is a 2*2 reversible logic circuit, where the inputs are simple A and B, and the outputs are sum and carry. There are some rules to find the sum and carry for the balanced quaternary reversible circuit.

Step1. Changed the both balanced quaternary input number into its standard quaternary.

Step2. Add them.

Step3. If the addition is less than or equal to 3, then the number has the Carry 0 and Sum the addition of standard state, and these are converted into balanced states.

Step4. If addition is more than 3, then number is changed into standard quaternary, the LSD is the sum and the MSD is the carry, and these standard states are converted into corresponding balanced states.

C. DESIGNING OF BALANCED QUATERNARY FULL ADDER

It is 3x3 reversible logic gate it has 4³(64) combinations of the states and the truth table must be larger, so here we show only balanced quaternary reversible full adder circuit, not the truth table of the circuit. It has two inputs with one carry_in input and one 0 (constant) input to make reversible. It produces two garbage outputs A and B, and Sum and Carry_out.
The figure shows that it is designed by the two balanced quaternary half adders, means it is the double of the balanced quaternary reversible half adder. It has the two balanced quaternary Toffoli gate and two balanced quaternary Feynman gate.

D. BALANCED QUATERNARY REVERSIBLE LOGIC MULTIPLIER
The balanced quaternary logic multiplication table is shown in table X.

The multiplication table has some rules:

- **Step1.** Firstly, change the balanced quaternary numbers into standard quaternary numbers.
- **Step2.** And multiply with each other, and find, if number is in standard quaternary then it changed in balanced quaternary else the digits multiplied with each other and then find standard quaternary.
- **Step3.** This process is followed when the result is not finding in a standard quaternary.
- **Step4.** Finally, the standard quaternary number is changed in balanced quaternary number, which shows the multiplier of the balanced states.

**TABLE X. BALANCED QUATERNARY MULTIPLICATION O PERATION**

V. SIMULATION RESULTS
The corresponding simulation results of the floating point Balanced Quaternary Reversible Logic Multiplier are shown below. All the synthesis and simulation results are performed using VHDL.
The synthesis and simulation are performed on Xilinx ISE 14.4. The simulation results are shown below figures.

**Figure-19.a** RTL schematic of Top-level of Balanced Quaternary Reversible Logic Multiplier

**Figure-19.b** RTL schematic of Internal block of Balanced Quaternary Reversible Logic Multiplier

**Figure-19.c** Technology schematic of Balanced Quaternary Reversible Logic Multiplier

**Figure-19.d** Synthesis report of Balanced Quaternary Reversible Logic Multiplier

**Figure-19.e**: simulated outputs for Balanced Quaternary Reversible Logic Multiplier

**VII. CONCLUSION AND FUTURE SCOPE**

Due to efficient realization of balanced quaternary reversible logic there would be a very high prospect to generate quaternary reversible logic synthesis. In this article, a significant realization of balanced quaternary reversible gates, circuits is proposed which will promote a standard balancing in reversible computing. We have proposed a methodology and balancing principles for the realization of balanced quaternary reversible gates, circuits. It significantly optimizes the hardware complexity of our proposed design of balanced quaternary reversible half-adder, full-adder and multiplier. The future work of the quaternary reversible logics is to realize a synthesis of reversible logic circuits. The fault tolerance can also be proposed in future based on this approach.
REFERENCES


