Design of Reversible/Quantum Ternary Comparator Circuits

Mozammel H. A. Khan

***Abstract*—Multiple-valued quantum circuits are promising choices for future quantum computing technology, since the multiple-valued quantum system is more compact than the corresponding binary quantum system. Grover's quantum search algorithm requires a sub-circuit called oracle, which takes a set of inputs and gives an output stating whether a given search condition on the inputs is satisfied or not. Equality, less-than, and greater-than comparisons are widely used as search conditions. In this paper, we show design of quantum ternary equality, less-than, and greater-than comparators on the top of ion-trap realizable 1-qutrit gates and 2-qutrit Muthukrishnan-Stroud gates.**

***Index Terms*—Multiple-valued logic, quantum logic, reversible logic, ternary comparators, ternary logic**

I. Introduction

Landauer [1] proved that binary logic circuits built using traditional irreversible gates lead inevitably to energy dissipation, regardless of the technology used to realize the gates. Zhirnov *et al* [2] showed that power dissipation in any future CMOS will lead to impossible heat removal and thus the speeding-up of CMOS devices will be impossible at some point of time, which will be reached soon. Bennett [3] proved that for power not to be dissipated in a binary logic circuit, it is necessary that the circuit be built from reversible gates. *A gate (or circuit) is reversible if it is a one-to-one mapping between the input values and the output values*. Thus all output patterns are just permutations of input patterns. Such circuit can be described by a permutation matrix. Bennett's theorem suggests that every future binary technology will have to use some kind of reversible gates in order to reduce power dissipation. This is also true for multiple-valued logic, which demonstrates several potential advantages over binary technology. Quantum technology is inherently reversible and is one of the most promising technologies for future computing systems [4]. Quantum computing allows solving problems much more efficiently than in classical computing. For instance, while classical algorithm needs N steps to search an unstructured database, a quantum Grover algorithm [5] needs only $\sqrt{N}$ steps and it can be proved that there is no classical algorithm that would require less steps than $O(N)$ [6, 7]. Although only few quantum algorithms are now known, researchers believe that many problems may be reduced to some of these algorithms, for instance to Quantum Fourier Transform or to Grover Algorithm. Thus, many NP-hard problems may be reduced to Grover search to give a practically useful and substantial reduction of complexity for large values of N.

An *oracle* is a logic circuit that takes a set of inputs and gives an output stating whether a given search condition on the input is satisfied or not. An oracle is an integral part of Grover's quantum search algorithm [4, 5]. Quantum oracles are built from quantum gates and many oracles include arithmetic, logic, and mixed blocks. If one only knows how to build a respective oracle, Grover quantum search algorithm and its modifications would be immediately useful to solve many problems when the physical quantum computers will be available. It is, therefore, important to study methods and algorithms to design various types of oracles. The problem of designing various classes of oracles or their blocks (components) is well known for binary quantum circuits [4, 8].

Multiple-valued quantum systems are physically realizable. Klimov *et al* [9] and Mc Hugh and Twamley [10] showed realization of ternary quantum gates using trapped ions. Muthukrishnan and Stroud [11] showed realization of d-valued ($d > 2$) quantum gates using liquid ion-trap. Bartlett *et al* [12] showed d-valued quantum encoding in spin systems and harmonic oscillators. Realizations of these multiple-valued quantum systems enable us to develop quantum algorithm and related oracles on these systems. In this paper, we show design of ternary equality, less-than, and greater-than comparator circuits, which are very important components in constructing oracles for ternary quantum search algorithm.

Ternary reversible/quantum logic synthesis is a new and immature research area [13-26]. Papers [17, 24-26] presented ternary quantum logic synthesis methods using 2-qutrit (quantum ternary digit) controlled gates. Papers [14-16, 22, 23] presented ternary quantum logic synthesis methods using multi-qutrit macro-level gates, which are needed to be realized using primitive quantum gates. Papers [14-17, 22-26,] presented generalized synthesis techniques but did not give synthesis examples of practically important ternary circuits like adder, subtractor, encoder, decoder, multiplexer, demultiplexer, comparators, etc. These circuits are major sub-circuits needed for ternary system design as well as constructing ternary quantum oracles. Design of ternary reversible/quantum adder is given in [17]. Synthesis of ternary reversible/quantum adder/subtractor is given in [19, 20]. Realization of ternary reversible/quantum encoder and decoder is given in [18]. Synthesis of ternary reversible/quantum multiplexer/demultiplexer is given in [21]. In this paper, we present design of reversible/quantum realization of ternary equality, less-than, and greater-than comparator circuits. For this purpose, we introduce 1-qutrit gates and 2-qutrit Muthukrishnan-Stroud (M-S) gates [11].

Manuscript received August 29, 2007.

Mozammel H. A. Khan is with the Department of Computer Science and Engineering, East West University, 43 Mohakhali, Dhaka 1212, Bangladesh phone: 880-2-9882308; fax: 880-2-8812336; e-mail: mhakhan@ewubd.edu

We then show the realization of macro-level 3-qutrit controlled-controlled gates built on the top of 1-qutrit and 2-qutrit M-S gates. Then we show the design of *n*-qutrit reversible/quantum equality, less-than, and greater-than comparator circuits using 1-qutrit and 3-qutrit controlled-controlled gates.

II. One-qutrit permutative gates

Any transformation of the qutrit state represented by a 3×3 unitary matrix specifies a valid 1-qutrit quantum gate. There are many such non-trivial 1-qutrit gates. However, in this work, we use only the unitary permutative transforms as shown by the permutative matrices of Figure 1. Transforms $Z(+1)$ and $Z(+2)$ shift the qutrit states by 1 and 2, respectively. Transform $Z(12)$ permutes the qutrit states 1 and 2, $Z(01)$ permutes the qutrit states 0 and 1, and $Z(02)$ permutes the qutrit states 0 and 2 without affecting the other qutrit state. The input-output relationships of these 1-qutrit gates are shown in truth table form in Table 1. The reversible 1-qutrit gates are elementary gates and can be realized using liquid ion trap quantum technology [11]. Therefore, we assign them a cost of 1. In reversible/quantum circuits, we will represent a 1-qutrit gate by the symbol of Figure 2, where *x* is the input, the output *y* is the *Z* transform of the input *x*, and the transform *Z* is any of the transforms shown in Figure 1 and Table 1.

$$Z(+1)=\begin{bmatrix}0&0&1\\1&0&0\\0&1&0\end{bmatrix}\quad Z(+2)=\begin{bmatrix}0&1&0\\0&0&1\\1&0&0\end{bmatrix}$$

$$Z(12)=\begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}\quad Z(01)=\begin{bmatrix}0&1&0\\1&0&0\\0&0&1\end{bmatrix}$$

$$Z(02)=\begin{bmatrix}0&0&1\\0&1&0\\1&0&0\end{bmatrix}$$

Figure 1. One-qutrit unitary permutative transforms.

Table 1. Truth table of 1-qutrit reversible gates.

Input	Transformed Output				
	$Z(+1)$	$Z(+2)$	$Z(12)$	$Z(01)$	$Z(02)$
0	1	2	0	1	2
1	2	0	2	0	1
2	0	1	1	2	0

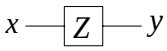


Figure 2. Symbol of 1-qutrit reversible gates [cost = 1].

Cascade of two 1-qutrit gates behaves like another 1-qutrit gate. The resultant 1-qutrit gates for all possible pairs of 1-qutrit cascades are shown in Table 2. If a cascade pair of two 1-qutrit gates has the resultant effect that the input signal to the first gate is restored at the output of the second gate (that is, the resultant gate is +0), then the second gate is said to be the *inverse gate* of the first gate. From Table 2, we can identify inverse gates for all five 1-qutrit gates.

Table 2. Resultant 1-qutrit gate for two cascaded 1-qutrit gates.

	Second gate				
First gate	+1	+2	12	01	02
+1	+2	+0	02	12	01
+2	+0	+1	01	02	12
12	01	02	+0	+1	+2
01	02	12	+2	+0	+1
02	12	01	+1	+2	+0

III. Ternary Muthukrishnan-Stroud gate family

Muthukrishnan and Stroud [11] proposed a family of 2-qudit (quantum digit) *d*-valued gates, which applies a 1-qudit unitary transform on the second qudit conditional on the first qudit being $(d-1)$. Figure 1 and Table 1 show five such ternary ($d = 3$) unitary permutative transforms. Figure 3 gives the symbolic representation of ternary Muthukrishnan-Stroud (M-S) gates. The ternary M-S gate is a controlled gate. The input *A* is the controlling input and the input *B* is the controlled input. The output *P* is equal to the input *A*. The output *Q* is the *Z* transform of the controlled input *B* if the controlling input $A = 2$, *Q* is equal to *B* otherwise. The ternary M-S gates can be realized using liquid ion trap quantum technology as an elementary gate [11]. Therefore, we assign these gates a cost of 1.

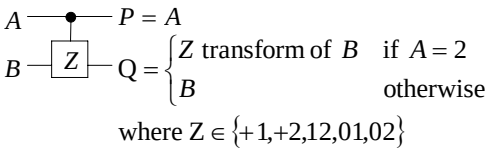


Figure 3. Symbol of 2-qutrit Muthukrishnan-Stroud gate family [cost = 1].

IV. Ternary controlled-controlled gate family

We propose a ternary controlled-controlled gate family using the symbol of Figure 4. In Figure 4, *A* and *B* are the controlling inputs and *C* is the controlled input. The output *P* is equal to the controlling input *A* and the output *Q* is equal to the controlling input *B*. The output *R* is the *Z* transform of the controlled input C if the controlling inputs $A = 2$ and $B = 2$, *R* is equal to *C* otherwise.

Realization of the ternary controlled-controlled gate family using ternary M-S gates is shown in Figure 5. From Figure 5, we see that if and only if the controlling inputs $A = 2$ and $B = 2$, then $x = 2$ and the *Z* transform is applied on the controlled input *C*. For all other combinations of the controlling inputs *A* and *B*, *x* will never be 2 and *Z* transform will not be applied on the controlled input *C*. The right most two M-S gates are inverse gates of the left most two M-S gates and are used to restore the input constant 0 for reuse in the circuit. *An auxiliary constant input used in the design of a circuit is called ancilla qutrit.* This realization needs 5 elementary gates and 1 ancilla qutrit.

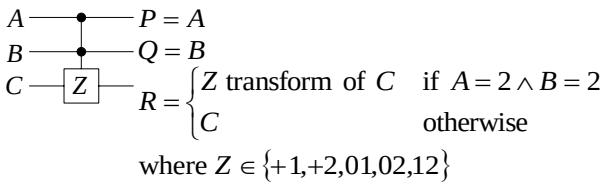


Figure 4. Symbol of ternary controlled-controlled gate family.

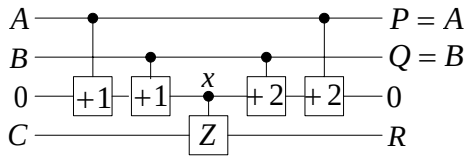


Figure 5. Realization of ternary controlled-controlled gate family [cost = 5 and ancilla qutrits = 1].

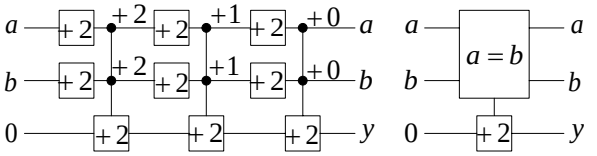
V. Ternary Equality Comparator

We will first realize 1-qutrit equality comparator and then we will realize *n*-qutrit equality comparator using 1-qutrit equality comparators.

A 1-qutrit equality comparator compares two 1-qutrit numbers *a* and *b* and sets *y* = 2 if and only if *a* = *b*; otherwise sets *y* = 0. The truth table of a 1-qutrit equality comparator is shown in the first two columns of Table 3. From Table 3, we see that for three input combinations the value of *y* is 2 and for the remaining input combinations it is 0. In reversible/quantum circuits, the fan-out of the input signals is only one. The 1-qutrit equality comparator circuit will be used as a building block of *n*-qutrit equality comparator, so the inputs will have to be reused in the other parts of the circuit. Therefore, we will keep the inputs unchanged and the *y* output will be generated along a constant input 0. This sort of circuit design is especially suitable for designing oracles. For the input combinations producing a 0 output, no transform will be applied on the input constant 0 and the output will be automatically 0. For the input combinations producing output 2, we will change both the *a* and *b* inputs to 2 by applying 1-qutrit transform (+1 or +2) and then using these two changed inputs as controlling signal, we will apply a +2 transform on the input constant 0 to produce the output 2 by using ternary controlled-controlled gates. The transforms needed on the inputs *a* and *b* and the input constant 0 are shown in Table 3. The realization of the 1-qutrit equality comparator is shown in Figure 6(a). In Table 3, the transforms needed for the input combination 00 are +2+2. So a +2 gate is placed along the *a* input and another +2 gate is placed along the *b* input. Then a ternary controlled-controlled gate is placed with transform +2 along the output *y* line to generate it. So, when the input values are *a* = 0 and *b* = 0, then a +2 transform will be applied on the input constant 0 and the output *y* will be 2. The other transforms needed from the Table 3 are placed in the similar manner. Along the lines *a* and *b*, 1-qutrit gates are in cascade. Therefore, the effective transforms at the controlling points are explicitly shown to verify the transforms needed from Table 3. The resultant effect of two cascaded 1-qutrit gates is shown in Table 2. The output *y* may be set to +1 by simply changing the transform of the ternary controlled-controlled gates. The circuit for the 1-qutrit equality comparator needs 6 + 3 × 5 = 21 elementary gates and 2 ancilla qutrits (the input constants 0 along the *y* output and another input constant 0 needed in the ternary controlled-controlled gates). We will use the symbol of Figure 6(b) to represent a 1-qutrit equality comparator in a larger circuit.

Table 3. Truth table and transformation table for 1-qutrit equality comparator.

Input *ab*	Output *y*	Transformation needed on inputs *ab* to make *ab* = 22	Transformation needed on input constant 0 to make *y* = 2
00	2	+2+2	+2
01	0		
02	0		
10	0		
11	2	+1+1	+2
12	0		
20	0		
21	0		
22	2	+0+0	+2



(a) Realization (b) Symbol

Figure 6. One-qutrit equality comparator [cost = 21, ancilla qutrits = 2].

In an *n*-qutrit equality comparator, we will compare whether two *n*-qutrit numbers $a = a_{n-1}\cdots a_1 a_0$ and $b = b_{n-1}\cdots b_1 b_0$ are equal or not. For designing an *n*-qutrit equality comparator, we define the following qutrit-wise function for $i = 0,1,\cdots,(n-1)$:

$$E_i = \begin{cases} 2 & \text{if } a_i = b_i \\ 0 & \text{otherwise} \end{cases}$$

We also define a recursive function F_i for $i = 0,1,\cdots,(n-1)$, whose value will be 2 if the partial numbers $a_i \cdots a_1 a_0$ and $b_i \cdots b_i b_0$ are equal, otherwise 0. The recursive function F_i can be defined as

$$F_i = \begin{cases} 2 & \text{if } (E_i = 2) \wedge (F_{i-1} = 2) \\ 0 & \text{otherwise} \end{cases}$$

If $F_{n-1} = 2$, then the *n*-qutrit numbers *a* and *b* are equal. From the recursive function F_i, we have that

$$F_0 = \begin{cases} 2 & \text{if } E_0 = 2 \\ 0 & \text{otherwise} \end{cases}$$

$$F_1 = \begin{cases} 2 & \text{if } (E_1 = 2) \wedge (F_0 = 2) \\ 0 & \text{otherwise} \end{cases}$$

$$F_{n-1} = \begin{cases} 2 & \text{if } (E_{n-1} = 2) \wedge (F_{n-2} = 2) \\ 0 & \text{otherwise} \end{cases}$$

Realization of *n*-qutrit equality comparator is shown in Figure 7. First F_0 is realized using a 1-qutrit equality comparator. Then E_1 is realized using a 1-qutrit equality comparator. Then we applied a controlled-controlled gate with E_1 and F_0 as controlling inputs and +2 as the transform. If $(E_1 = 2) \wedge (F_0 = 2)$, then F_1 will be 2, otherwise F_1 will be 0. In this way we complete the whole circuit. We know that +2 +1 = 0. Using this fact, we restore the ancilla constants in the constant restoring part of the circuit. For example, the input constant 0 along E_{n-1} is restored using a similar 1-qutrit equality comparator with the same set of inputs and a +1 transform. From Figure 7, we can compute that the number of elementary gates or the cost of the realization is $52n-15$ and the number of ancilla qutrits is $3n-1$.

VI. Ternary Less-than Comparator

We will first realize 1-qutrit less-than comparator, then we will realize *n*-qutrit less-than comparator using 1-qutrit less-than comparators and 1-qutrit equality comparators.

A 1-qutrit less-than comparator compares two 1-qutrit numbers *a* and *b* and sets *y* = 2 if and only if *a* < *b*; otherwise sets *y* = 0. The truth table of 1-qutrit less-than comparator is shown in the first two columns of Table 4. The transformations needed on the inputs *ab* and the constant input 0 are calculated as is done for the 1-qutrit equality comparator. The realization of the 1-qutrit less-than comparator is shown in Figure 8(a). The circuit for the 1-qutrit less-than comparator needs 5 + 3 × 5 = 20 elementary gates and 2 ancilla qutrits (the input constants 0 along the *y* output and another input constant 0 needed in the ternary controlled-controlled gates). We will use the symbol of Figure 8(b) to represent a 1-qutrit less-than comparator in a larger circuit.

Table 4. Truth table and transformation table for 1-qutrit less-than comparator.

Input *ab*	Output *y*	Transformation needed on inputs *ab* to make *ab* = 22	Transformation needed on input constant 0 to make *y* = 2
00	0		
01	2	+2+1	+2
02	2	+2+0	+2
10	0		
11	0		
12	2	+1+0	+2
20	0		
21	0		
22	0		

In an *n*-qutrit less-than comparator, we will compare whether an *n*-qutrit number $a = a_{n-1} \cdots a_1 a_0$ is less than another *n*-qutrit number $b = b_{n-1} \cdots b_1 b_0$ or not. For designing *n*-qutrit less-than comparator, we define the following qutrit-wise functions for $i = 0,1,\cdots,(n-1)$:

$$E_i = \begin{cases} 2 & \text{if } a_i = b_i \\ 0 & \text{otherwise} \end{cases}$$

$$L_i = \begin{cases} 2 & \text{if } a_i < b_i \\ 0 & \text{otherwise} \end{cases}$$

We also define a recursive function F_i for $i = 0,1,\cdots,(n-1)$, whose value will be 2 if the partial number $a_i \cdots a_1 a_0$ is less than the partial number $b_i \cdots b_i b_0$, otherwise 0. The recursive function F_i can be defined as

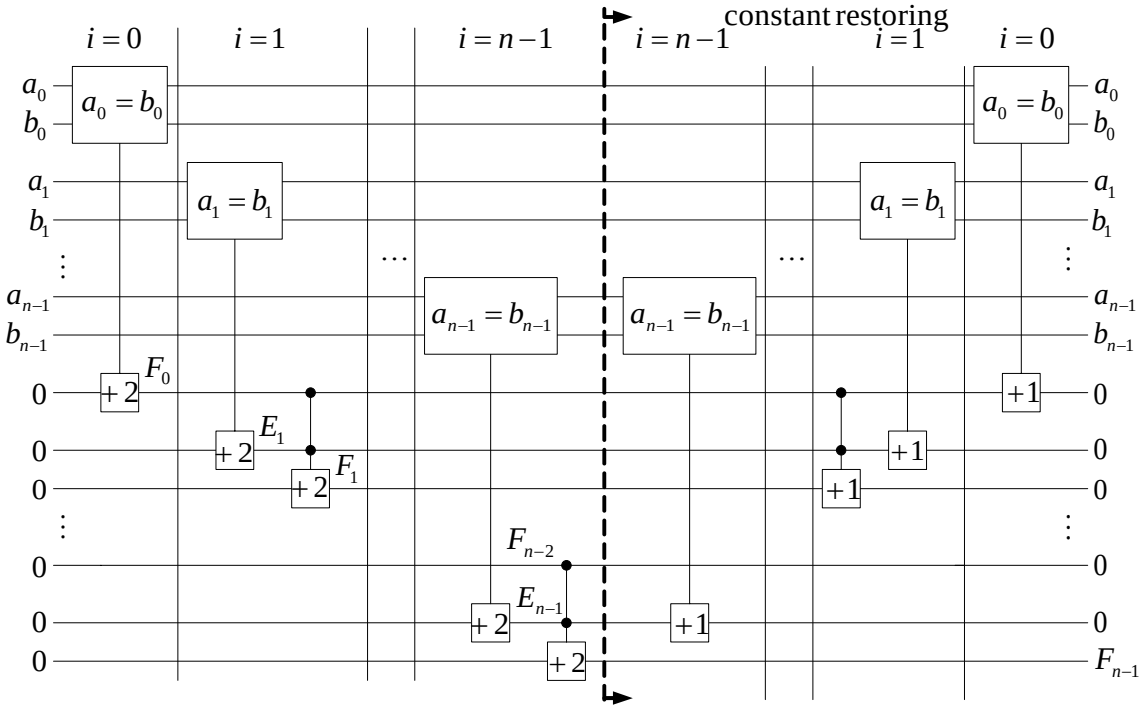
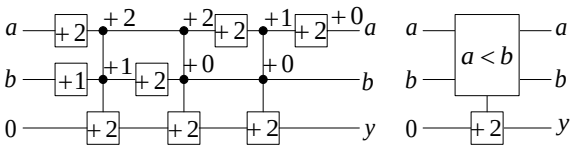


Figure 7. Realization of an *n*-qutrit equality comparator [cost = $52n-15$, ancilla qutrits = $3n-1$].



(a) Realization (b) Symbol

Figure 8. One-qutrit less-than comparator [cost = 20, ancilla qutrits = 2].

$$F_i = \begin{cases} 2 & \text{if } L_i = 2 \text{ for } i = 0 \\ 2 & \text{if } (L_i = 2) \vee (E_i = 2) \wedge (F_{i-1} = 2) \text{ for } i > 0 \\ 0 & \text{otherwise} \end{cases}$$

If $F_{n-1} = 2$, then the n-qutrit number a is less than the n-qutrit number b. From the recursive function F_i, we have that

$$F_0 = \begin{cases} 2 & \text{if } L_0 = 2 \\ 0 & \text{otherwise} \end{cases}$$

$$F_1 = \begin{cases} 2 & \text{if } (L_1 = 2) \vee (E_1 = 2) \wedge (F_0 = 2) \\ 0 & \text{otherwise} \end{cases}$$

$$F_{n-1} = \begin{cases} 2 & \text{if } (L_{n-1} = 2) \vee (E_{n-1} = 2) \wedge (F_{n-2} = 2) \\ 0 & \text{otherwise} \end{cases}$$

Realization of n-qutrit less-than comparator is shown in Figure 9. First F_0 is realized using a 1-qutrit less-than comparator. Then L_1 is realized using a 1-qutrit less-than comparator and E_1 is realized using a 1-qutrit equality comparator. Then we applied a controlled-controlled gate with E_1 and F_0 as controlling inputs and with a +2 transform. Any one of the two +2 transforms will be applied along F_1. If $L_1 = 2$, then F_1 will be 2. Or if $(E_1 = 2) \wedge (F_0 = 2)$, then F_1 will be 2. Otherwise F_1 will be 0. In this way we complete the whole circuit. We restore the ancilla constants using the constant restoring part of the circuit. From Figure 9, we can compute that the number of elementary gates or the cost of the realization is $92n - 77$ and the number of ancilla qutrits is $5n - 3$.

VII. TERNARY GREATER-THAN COMPARATOR

As in the case of ternary less-than comparator, we will first realize 1-qutrit greater-than comparator, then we will realize n-qutrit greater-than comparator using 1-qutrit greater-than comparators and 1-qutrit equality comparators.

A 1-qutrit greater-than comparator compares two 1-qutrit numbers a and b and sets $y = 2$ if and only if $a > b$; otherwise sets $y = 0$. The truth table of 1-qutrit greater-than comparator is shown in the first two columns of Table 5. The transformations needed on the inputs ab and the constant input 0 are calculated as is done for the 1-qutrit less-than comparator. The realization of the 1-qutrit greater-than comparator is shown in Figure 10(a). The circuit for the 1-qutrit greater-than comparator needs 5 + 3 × 5 = 20 elementary gates and 2 ancilla qutrits (the input constants 0 along the y output and another input constant 0 needed in the ternary controlled-controlled gates). We will use the symbol of Figure 10(b) to represent a 1-qutrit greater-than comparator in a larger circuit.

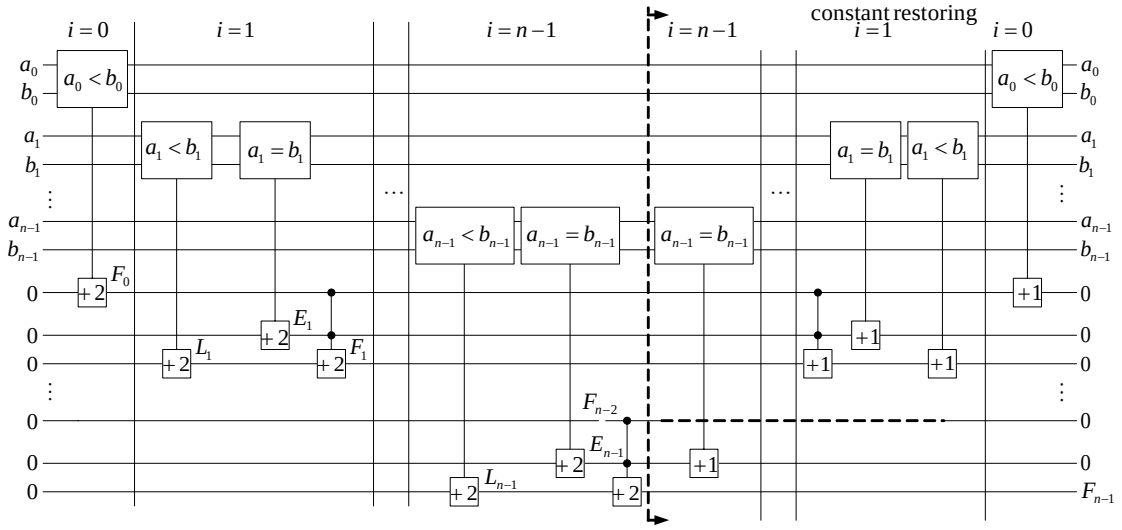
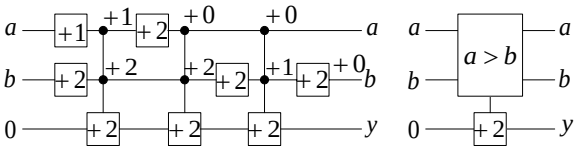


Figure 9. Realization of n-qutrit less-than comparator [cost = $92n - 77$, ancilla qutrits = $5n - 3$].

Table 5. Truth table and transformation table for 1-qutrit greater-than comparator.

Input *ab*	Output *y*	Transformation needed on inputs *ab* to make *ab* = 22	Transformation needed on input constant 0 to make *y* = 2
00	0		
01	0		
02	0		
10	2	+1+2	+2
11	0		
12	0		
20	2	+0+2	+2
21	2	+0+1	+2
22	0		



(a) Realization (b) Symbol

Figure 10. One-qutrit greater-than comparator [cost = 20, ancilla qutrits = 2].

In an *n*-qutrit greater-than comparator, we will compare whether an *n*-qutrit number $a = a_{n-1}\cdots a_1 a_0$ is greater than another *n*-qutrit number $b = b_{n-1}\cdots b_1 b_0$ or not. For designing *n*-qutrit greater-than comparator, we define the following qutrit-wise functions for $i = 0,1,\cdots,(n-1)$:

$$E_i = \begin{cases} 2 & \text{if } a_i = b_i \\ 0 & \text{otherwise} \end{cases}$$

$$G_i = \begin{cases} 2 & \text{if } a_i > b_i \\ 0 & \text{otherwise} \end{cases}$$

We also define a recursive function F_i for $i = 0,1,\cdots,(n-1)$, whose value will be 2 if the partial number $a_i \cdots a_1 a_0$ is greater than the partial number $b_i \cdots b_i b_0$, otherwise 0. The recursive function F_i can be defined as

$$F_i = \begin{cases} 2 & \text{if } G_i = 2 \text{ for } i = 0 \\ 2 & \text{if } (G_i = 2) \vee (E_i = 2) \wedge (F_{i-1} = 2) \text{ for } i > 0 \\ 0 & \text{otherwise} \end{cases}$$

If $F_{n-1} = 2$, then the *n*-qutrit number *a* is greater than the *n*-qutrit number *b*. From the recursive function F_i, we have that

$$F_0 = \begin{cases} 2 & \text{if } G_0 = 2 \\ 0 & \text{otherwise} \end{cases}$$

$$F_1 = \begin{cases} 2 & \text{if } (G_1 = 2) \vee (E_1 = 2) \wedge (F_0 = 2) \\ 0 & \text{otherwise} \end{cases}$$

$$F_{n-1} = \begin{cases} 2 & \text{if } (G_{n-1} = 2) \vee (E_{n-1} = 2) \wedge (F_{n-2} = 2) \\ 0 & \text{otherwise} \end{cases}$$

Realization of *n*-qutrit greater-than comparator is shown in Figure 11. From Figure 11, we can compute that the number of elementary gates or the cost of the realization is $92n - 77$ and the number of ancilla qutrits is $5n - 3$.

VIII. Conclusion

Multiple-valued quantum circuits are a promising choice for future quantum computing technology, since they are more compact than the corresponding binary quantum circuits. Ternary reversible/quantum logic synthesis is a new and immature research area [13-26]. Papers [14-17, 22-26,] presented generalized synthesis techniques. Design of ternary reversible/quantum adder is given in [17]. Synthesis of ternary reversible/quantum adder/subtractor is given in [19, 20]. Realization of ternary reversible/quantum encoder and decoder is given in [18]. Synthesis of ternary reversible/quantum multiplexer/demultiplexer is given in [21]. But, as far as our knowledge is concerned, designs of ternary comparator circuits are not yet reported in the literature.

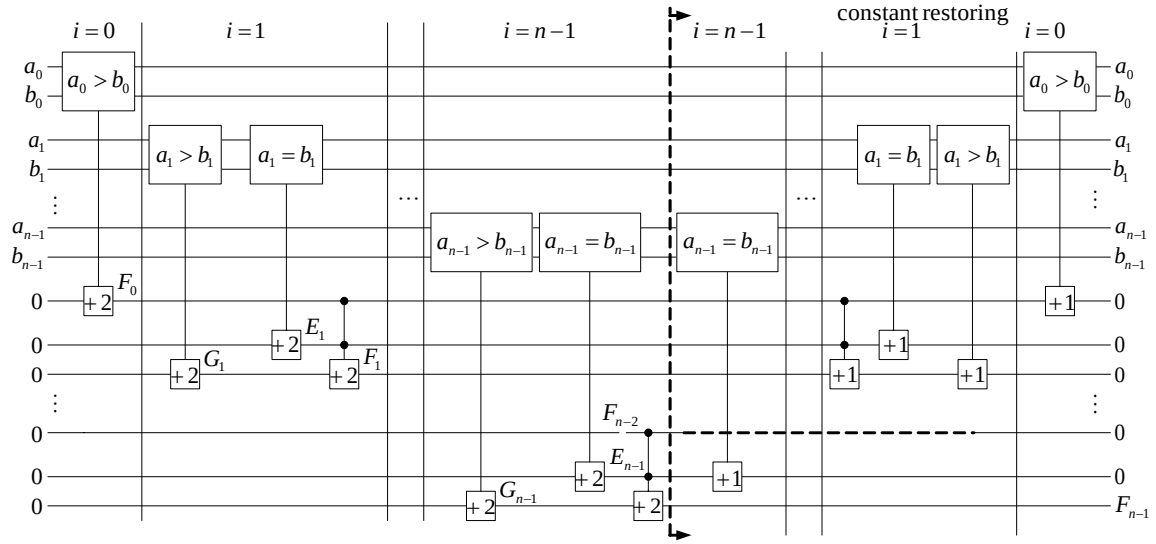


Figure 11. Realization of *n*-qutrit greater-than comparator [cost = $92n - 77$, ancilla qutrits = $5n - 3$].

The most practically important quantum algorithm is Grover's quantum search algorithm [5]. This algorithm requires a sub-circuit called 'oracle', which takes a set of inputs and generates an output indicating whether a specified search condition is satisfied or not. The oracle is designed using classical reversible design techniques without considering the superposition of the qutrit states [4]. Equality, less-than, and greater-than condition testing are very important in search problems. In this paper, we present reversible synthesis of ternary equality, less-than, and greater-than comparator circuits on the top of ion-trap realizable quaternary 1-qutrit gates and 2-qutrit Muthukrishnan-Stroud gates [11]. These gates are so far the best presented multiple-valued quantum gates in the literature.

Observe that this method of synthesis, in contrast to most synthesis methods from literature, performs a conversion of a non-reversible function to a reversible one. Moreover, comparing to the general synthesis methods from literature, this problem specific synthesis method can be used to synthesize quaternary comparators of very large size.

REFERENCES

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

[2] V. V. Zhirnov, R. K. Kavin, J. A. Hutchby, and G. I. Bourianoff, "Limits to Binary Logic Switch Scaling – A Gedanken Model", *Proc. of the IEEE*, vol. 91, no. 11, 2003, pp. 1934-1939.

[3] C. H. Bennett, "Logical Reversibility of Computation", *IBM Journal of Research and Development*, 17, 1973, pp. 525-532.

[4] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.

[5] L.K. Grover, "A fast quantum mechanical algorithm for database search", *Proceedings of 28th Annual ACM Symposium on the Theory of Computing (STOC)*, p. 212, 1996.

[6] M. Boyer, G. Brassard, P. Hoyer, and A. Tapp, "Tight bounds on quantum searching", *Proceedings of PhysComp*, 1996.

[7] Ch. Zalka, "Grover's Quantum Searching algorithm is optimal", *Phys. Rev. A* 60, pp. 2746–2751, 1999.

[8] M. Perkowski, **"**From Quantum Gates to Quantum Learning: recent research and open problems in quantum circuits", invited paper, *Proceedings of 6th International Workshop on Boolean Problems*, Freiberg University of Mining and Technology, Germany, September 23-24, 2004.

[9] A. C. Klimov, R. Guzman, J. C. Retamal, and C. Saavedra, "Qutrit quantum computer with trapped ions" ,*Phys. Rev. A*, 67, 062313, 2003.

[10] D. Mc Hugh, J. Twamley, "Trapped-ion qutrit spin molecule quantum computer", arXiv:quant-ph/0506031 v1 3 Jun 2005.

[11] A. Muthukrishnan, C. R. Stroud, Jr., "Multivalued logic gates for quantum computation", *Phys. Rev. A*, 62(5), 052309/1-8, 2000.

[12] S. D. Bartlett, H. de Guise, and B. C. Sanders, "Quantum encodings in spin systems and harmonic oscillators", arXiv:quant-ph/0109066 v2 6 May 2002.

[13] M. Perkowski, A. Al-Rabadi, P. Kerntopf, **"**Multiple-Valued Quantum Logic Synthesis", *Proc. of 2002 Intl Symposium on New Paradigm VLSI Computing*, Sendai, Japan, December 12-14, 2002, pp. 41 - 47.

[14] M. H. A. Khan, M. A. Perkowski, and P. Kerntopf, "Multi-Output Galois Field Sum of Products Synthesis With New Quantum Cascades", *Proc. 33rd Int. Symp. On Multiple-Valued Logic*, Tokyo, Japan, 16-19 May 2003, pp. 146-153.

[15] M. H. A. Khan, M. A. Perkowski, and M. R. Khan, "Ternary Galois Field Expansions for Reversible Logic and Kronecker Decision Diagrams for Ternary GFSOP Minimization", *Proc. 34th Int. Symp. On Multiple-Valued Logic*, Toronto, Canada, 19-22 May 2004, pp. 58-67.

[16] M. H. A. Khan, M. A. Perkowski, M. R. Khan, and P. Kerntopf, "Ternary GFSOP Minimization using Kronecker Decision Diagrams and Their Synthesis with Quantum Cascades", *Journal of Multiple-Valued Logic and Soft Computing*, vol. 11, no. 5-6, 2005, pp.567-602.

[17] M. H. A. Khan and M. A. Perkowski, "Genetic Algorithm Based Synthesis of Multi-Output Ternary Functions Using Quantum Cascade of Generalized Ternary Gates", *Proc. 2004 Congress on Evolutionary Computation*, Portland, OR, USA, 19-23 June 2004, pp. 2194-2201.

[18] M. H. A. Khan, M. A. Perkowski, "Quantum Realization of Ternary Encoder and Decoder", *Proc. 7th Int. Symp. On Representations and Methodology of Future Computing Technologies (RM2005)*, Tokyo, Japan, 5 - 6 September 2005.

[19] M. H. A. Khan, M. A. Perkowski, "Quantum Realization of Ternary Parallel Adder/Subtractor with Look-Ahead Carry", *Proc. 7th Int. Symp. On Representations and Methodology of Future Computing Technologies (RM2005)*, Tokyo, Japan, 5 - 6 September 2005.

[20] M. H. A. Khan and Marek A Perkowski, "Quantum Ternary Parallel Adder/Subtractor with Partially-Look-Ahead Carry", *Journal of Systems Architecture*, vol. 53, no. 7, 2007, pp. 453-464.

[21] M. H. A. Khan, "Design of Reversible/Quantum Ternary Multiplexer and Demultiplexer", *Engineering Letters*, vol. 13, no. 2, 2006, pp. 65-69.

[22] D. M. Miller, G. W. Dueck, and D. Maslov, "A Synthesis Method for MVL Reversible Logic", *Proc. 34th Int. Symp. On Multiple-Valued Logic*, Toronto, Canada, 19-22 May 2004, pp. 74-80.

[23] D. M. Miller, D. Maslov, and G. W. Dueck, "Synthesis of Quantum Multiple-Valued Circuits", *Journal of Multiple-Valued Logic and Soft Computing*, vol. 12, no. 5-6, 2006.

[24] E. Curtis and M. Perkowski, **"**A Transformation Based Algorithm for Ternary Reversible Logic Synthesis using Universally Controlled Ternary Gates", *Proceedings of IWLS 2004*, pp. 345-352.

[25] N. Denler, B. Yen, M. Perkowski and P. Kerntopf, "Synthesis of Reversible Circuits from a Subset of Muthukrishnan-Stroud Quantum Realizable Multi-Valued Gates", *Proceedings of IWLS 2004*, pp. 321-328.

[26] B. Yen, P. Tomson, and M. Perkowski, **"**Sum of Non-disjoint Cubes Covering Generation for Multi-Valued Systems of base 2, for use in Muthukrishnan-Stroud Quantum Realizable Gates: An Extension of the EXOR Covering Problem", *Proceedings of* IWLS 2005.