Quantum Inf Process
DOI 10.1007/s11128-015-1011-3

Radix-independent, efficient arrays for multi-level n-qudit quantum and reversible computation

Majid Mohammadi[1]

Received: 5 November 2014 / Accepted: 22 April 2015

Abstract Multiple-valued quantum logic allows the designers to reduce the number of cells while obtaining more functionality in the quantum circuits. Large r-valued reversible or quantum gates (r stands for radix and is more than 2) cannot be directly realized in the current quantum technology. Therefore, we are interested in designing the large reversible and quantum controlled gates using the arrays of one-quantum digit (qudit) or two-qudit gates. In our previous work, we proposed quantum arrays to implement the r-valued quantum circuits. In this paper, we propose novel efficient structures and arrays, for r-valued quantum logic gates. The quantum costs of the developed quantum arrays are independent of the radix of calculations in the quantum circuit.

Keywords Quantum computing · Reversible logic · Multiple-valued logic · Controlled gate · Quantum cost · Optimization

1 Introduction

Multiple-valued quantum and reversible logic (MVQRL) circuits are important in the emerging quantum technologies. Using r-valued quantum circuits, designers can reduce the number of quantum cells to obtain a desired functionality for the quantum circuit [1]. A large amount of work has been done in the field of synthesis and optimization of binary quantum circuits. Methods for automated synthesis of binary and r-valued quantum circuits, based on genetic algorithms (GA) and evolutionary algorithms (EA), have also been developed [2–8]. An important group of r-valued

✉ Majid Mohammadi
M_mohamadi@sbu.ac.ir

[1] Faculty of Computer Engineering, Shahid Bahonar University of Kerman, Kerman, Iran

Published online: 14 May 2015

quantum gates used in the synthesis and design of quantum circuits is controlled gates [1]. Khan et al. [3] proposed a 2 × 2 generalized ternary gate (GTG) and realized the ternary Toffoli gate using GTG gates. Miller et al. [9] used quantum CNOT, cycle, and shift gates in their synthesis algorithm. Designing of ternary multiplexers has also been proposed in [10].

The controlled gates with many controls are not directly realizable in current quantum technology; therefore, efficient implementing of these gates using smaller gates is an important problem for r-valued quantum computing [11]. The quantum array which is proposed in [12] for binary quantum circuits is extended by Brennen et al. [12] for r-valued quantum circuits. Rosenbaum et al. [13] proposed a structure for implementing large quantum gates using two-qudit gates; however, the transposition gates which are used in their arrays are not proved to be realizable in quantum technology.

In the previous work in [1], we proposed quantum arrays to implement the r-valued quantum gates. In this paper, we propose new arrays of r-valued quantum logic circuits. We consider various realizable controlled quantum logic gates (including CNOT, controlled cycle, and controlled self-shift gates) to implement efficient multiple-valued logic (MVL) arrays. We also develop efficient structures for implementing r-valued controlled cycle gates. These structures are optimized circuits which we proposed in [1]. The main advantage of the proposed circuits in [1] and this paper is that they use pure states$|0\rangle$, $|1\rangle$, ..., $|r-1\rangle$ to implement large gates. This advantage allows us to directly use the theorems and circuits in reversible logic, too. Also, the circuits are implementable in other technologies such as complementary metal–oxide–semiconductor (CMOS), quantum cellular automata (QCA), single-electron transistor (SET),. In [16], the implementable gates in our current technology are introduced. Many researchers also tried to use these implementable gates in their works such as [1–3,8–10,12–16].

The basic material on multiple-valued quantum gates is presented in Sect. 2. In the Sect. 3, we introduce new arrays for quantum r-valued three-qudit controlled U gates. In Sects. 4 and 5, we extend the designs to the gates with different control inputs and different threshold values. Comparison, conclusions and references are presented in Sects. 5, 6, and 7, respectively.

2 Multiple-valued quantum gates

A quantum digit or qudit is a cell of an r-valued quantum logic circuit which can accept r distinct quantum states. In r-valued logic, r quantum states $|0\rangle$, $|1\rangle$, ..., $|r-1\rangle$ are assigned to r levels of energy in the system as the base vectors, as shown in Eq. 1. For ternary logic, base vectors are named quantum ternary digits or qutrits. As the binary qubits, a qudit has also the superposition property which allows it to accept a linear combination of the base states. That is, it is possible to have the state of an r-valued quantum system as $\psi = a_0|0\rangle + a_1|1\rangle + \cdots + a_{r-1}|r-1\rangle$ where $a_0, a_1, \ldots, a_{r-1}$ are complex numbers. Measuring of ψ results in getting the base states $|0\rangle, |1\rangle$, ..., $|r-1\rangle$, with probabilities of $P_0 = |a_0|^2$, $P_1 = |a_1|^2$, ..., $P_{r-1} = |a_{r-1}|^2$, respectively. The $a_0, a_1, \ldots, a_{r-1}$ coefficients must be constant values such that the vector ψ has a norm of one, i.e., $\|\psi\| = 1$. An r-valued quantum gate manipulates the qudits and produces a new state, named the output state of the quantum gate. A one-qudit quantum gate U which manipulates an input qudit is expressed by an r by r unitary matrix as shown in Eq. 2. For example, unitary matrix of a ternary NOT gate is shown in Eq. 3. To calculate the output state of a quantum gate, we have to multiply the gate's unitary matrix by its input state vector. In an r-valued quantum logic system, an n by n gate which operates on n qudit is specified as a unitary matrix of r^n by r^n. A resultant unitary matrix of arbitrary quantum circuit is created by matrix multiplications of composing subcircuits. Operation of a quantum gate is described by matrix multiplication.

$$|0\rangle = \begin{bmatrix}1\ 0\ 0\ \ldots\ 0\end{bmatrix}^{\mathrm{T}}, |1\rangle = \begin{bmatrix}0\ 1\ 0\ \ldots\ 0\end{bmatrix}^{\mathrm{T}}, \ldots, |r-1\rangle = \begin{bmatrix}0\ 0\ 0\ \ldots\ 1\end{bmatrix}^{\mathrm{T}} \quad (1)$$

$$\mathrm{U} = \begin{bmatrix} u_{11} & \cdots & u_{1r} \\ \vdots & \ddots & \vdots \\ u_{r1} & \cdots & u_{rr} \end{bmatrix} \quad (2)$$

$$\mathrm{NOT} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \mathrm{NOT}(|0\rangle) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = |2\rangle \quad (3)$$

For a two-partite qudit quantum system, there are r^2 pure quantum states $|00\rangle$ $|01\rangle$, $|02\rangle$, and $|r-1, r-1\rangle$, and all combinations of these pure states. If the state of each qudit is not separable, the state of the quantum system is said to be entangled. Superposition and entanglement properties allow qudit states to grow much faster in dimension than classical r-valued digits and qubits [7]. The state of an n-digit quantum register (n-partite quantum system) with r-valued qudits can be at superposition state of all r^n basic states. This allows us to apply an operator (or a gate) to all possible states, simultaneously. This is a type of parallel processing which exists inherently in quantum computers. Any quantum circuit is a composition of parallel and serial connections of gates. Serial connection of gates corresponds to multiplication of their unitary matrices. A parallel connection of gates corresponds to Kronecker multiplication of unitary matrices [1]. These all contribute to make difficulties in understanding the concepts of quantum computing and creating efficient analysis and synthesis algorithms for quantum computing. However, all of these become much easier when we consider only permutative gates, which are equivalent to n-inputs/n-outputs truth tables of quantum functions. We work with such special class of r-valued quantum logic gates in this paper.

The binary controlled gates are extended to the MVL ones [12]. For example, the Feynman gate is extended to MV-CNOT such that it has one control input and one target input/output. In this way, a controlled U gate can be defined as a gate which has two control and target inputs and two control and target outputs. If control input is more than a threshold, the U function is applied to the input to make the target output. In [14,15] a radix-r number system as a Galois Field (GFr) is defined, and primary operations of the Galois field are defined as the primary gates. Modulo-r sum is defined as an operator in GFr. In the definition of r-valued Feynman gate in [14], the control input does not have the controlling role as its binary counterpart. Some basic r-valued quantum gates, such as ternary Feynman gate are realizable in quantum technology [16,17]. In this paper, we consider the controlled gates whose control inputs control the operation of the gate. These types of gates are named *Toffoli-like gates* or *controlled gates* in some papers [1,3,9].

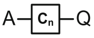
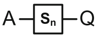
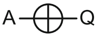
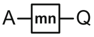
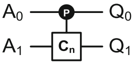
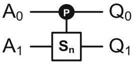
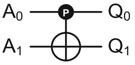
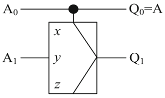
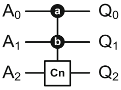
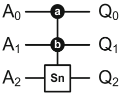
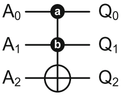
In the category of controlled quantum logic gates, two set of gates are used in synthesis of r-valued quantum circuits. The first category, generalized ternary gates (GTG), are ternary controlled 2×2 gates. They are used to synthesize the ternary quantum circuits in [3]. The second category, controlled NOT, cycle, and self-shift gates, used in [8,9], have a simpler structure than GTG gates and are extended to higher radixes with more control inputs. Table 1 shows the symbols and operations of the controlled gates. In this table, r is radix and $n(0 \leq n < r)$ is type of the gate (for example, $\mathrm{C}_1,\ \mathrm{C}_2,\ \ldots,\ \mathrm{C}_{n-1}$). A cycle gate adds a value (n) to its input. Add operation is a summation in GFr field or “mod r” addition. Self-shift gate multiplies the input by $r - 1$ and adds the result to a number n (again in mod r).

Table 1 also shows the symbol of the two-partite r-valued controlled gates. The A_0 input is control input and Q_1 is the target output. The Pvalue in the circle of control input (black-filled circle) represents the threshold value of the input. If the control input (A_0) is less than the threshold (P), then the gate is inactive and target Q_1 is equal to its input (A_1); otherwise, the gate’s function is applied to the input (A_1) to generate a value for the target output (Q_1). The Q_0 is always equal to A_0. As the binary quantum Toffoli gates [11], the r-valued quantum gates are also generalized to the controlled $n \times n$ gates [12]. A generalized controlled gate has k control inputs/outputs and one main or target input/output. A threshold is also assigned to each of control inputs. Each gate applies its function to the main input if all controls are equal to or greater than their corresponding threshold values; otherwise, the target output is equal to the main input. Control outputs are always equal to their corresponding inputs.

As mentioned, the GTG gate which is used in [3] for synthesis of ternary quantum circuits is a controlled 2×2 gate whose functionality is more complex than the above-mentioned gates. Each GTG gate is composed of a ternary multiplexer and three one-qutrit gates (Table 1, row 8). Three $x,\ y$, and z gates are elementary one-qutrit gates. Each of these gates can be a buffer, C_1, $\mathrm{C}_2,\ \mathrm{S}_0,\ \mathrm{S}_1$ or NOT gate. If we consider 0, 1, …, five constants for the buffer, C_1, $\mathrm{C}_2,\ \mathrm{S}_0,\ \mathrm{S}_1$, and NOT gates, respectively; each of x, y, or z parameters can be a constant value of 0–5. According to value of the control input (A_0), one of $x,\ y$, or z gates is applied to A_1 input. If A_0 is 0, the x gate is applied to A_1 input and results the Q_1 value. If A_0 is 1 or 2, the y or z gates are applied to A_1 input (Table 1, row 8).

Some of the one-qudit and controlled two-qudit quantum gates are realizable in current quantum technology [16]. Implementation of larger quantum controlled gates using the one-qudit or two-qubit gates is important because bigger gates are not directly realizable in the quantum technology. Therefore, we try to propose implementations for these bigger gates in terms of one- and two-qudit gates. The quantum cost of an r-valued quantum logic gate is the number of one- and two-qudit gates (1×1 and 2×2 gates) which are needed to implement the gate [1]. Table 1 also shows the QC of the r-valued quantum gates calculated based on the lemmas defined in the original paper [1].

Table 1 The r-valued reversible and quantum gates and their specifications

Gate	Name	Function	Symbol	QC
C_n	Cycle	$C_n(A) = (A+n) \bmod r$		1
S_n	Self-shift	$S_n(A) = ((r-1)A+n) \bmod r$		1
NOT	NOT	$\text{NOT}\,(A) = r-1-A$		1
(m, n)	Transposition	$Q = \begin{cases} m & \text{if } A = n \\ n & \text{if } A = m \\ A & \text{otherwise} \end{cases}$		1
CC_n	Controlled cycle	$Q_1 = \begin{cases} A_1 & \text{if } A_0 < P \\ C_n(A_1) & \text{if } A_0 \geq P \end{cases}$		1
CS_n	Controlled self-shift	$Q_1 = \begin{cases} A_1 & \text{if } A_0 < P \\ S_n(A_1) & \text{if } A_0 \geq P \end{cases}$		1
CNOT	Controlled NOT	$Q_1 = \begin{cases} A_1 & \text{if } A_0 < P \\ \text{NOT}\, A_1 & \text{if } A_0 \geq P \end{cases}$		1
GTG	Generalized ternary gate	$Q_1 = \begin{cases} x\,(A_1) & \text{if } A_0 = 0 \\ y\,(A_1) & \text{if } A_0 = 1 \\ z\,(A_1) & \text{if } A_0 = 2 \end{cases}$		1
CCC_n	Controlled controlled cycle	If $A_0 \geq a \,\&\, A_1 \geq b$: $Q_2 = Cn\,.\,A_2$ Else: $Q_2 = A_2$		4
CCS_n	Controlled controlled self-shift	If $A_0 \geq a \,\&\, A_1 \geq b$: $Q_2 = Sn(A_2)$ Else: $Q_2 = A_2$		5
CCNOT	Controlled controlled NOT	If $A_0 \geq a \,\&\, A_1 \geq b$: $Q_2 = \text{NOT}(A_2)$ Else: $Q_2 = A_2$		5

The QC parameter is calculated based on definitions and calculations in [1]

3 New arrays for quantum r-valued three-qudit controlled U gates

Definition 1 The MVL gate is said to be Hermitian or self-inverse if its inverse is itself. For example, ternary S_0, S_1, and NOT gates are Hermitian. In r-valued circuits, we can show that all S_n and NOT gates are Hermitian [1].

Lemma 1 *Any r-valued three-qudit Hermitian controlled U gate with input thresholds of $r-1$ can be implemented using five two-qudit controlled gates as shown in* Fig. 1a*, in which r = radix (odd value), $P = r-1$, $n = r-2$, and $m = r/2$.*

Proof To verify the operation of the circuit, we consider the values of Q_2, Q_1 and Q_0 for all possible values of inputs. The Q_1 is A_1 for all values of A_1. Since Sn is Hermitian, Q_2 is also equal to A_2 for all values of A_1 and A_2. If the value of A_1 is less than $r-1$, two Sn gates and G3 are inactive. Depending on the A_2 value, both

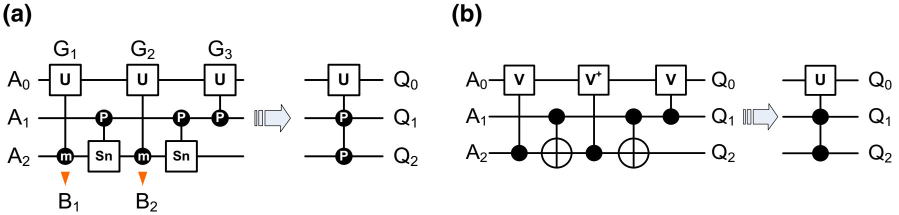


Fig. 1 **a** Combination of gates in Lemma 1, **b** binary counterpart [11], The r is radix (odd value) and $P = r - 1$, $n = r - 2$, and $m = r/2$

G1 and G2 gates are either active or inactive. In both cases, the Q_0 is equal to A_0. For $A_1 = r - 1$ and $A_2 = 0, 1, 2, \ldots, r - 2$, two controlled U gates are active (G3 and one of G1 or G2) which results in $Q_0 = A_0$. Finally, for $A_1 = r - 1$ and $A_2 = r - 1$, all three controlled U gates are active which results in $Q_0 = \mathrm{U}(A_0)$ □

The circuit in Fig. 1a is similar to the quantum decomposition of binary controlled U gates proposed in [11]. For simpler comparison, we show this circuit in Fig. 1b. In binary implementation, V is square root of U gate and V^+ is complex conjugate transpose of V. The circuit in Fig. 1a is not a generalization of Fig. 1b because in Fig. 1a, there exist a situation that all gates are active. In this case, the equivalent circuit in Fig. 1b is a V gate which is not correct. This new structure uses only five gates for implementation of any Hermitian controlled U gate with two control inputs and radix r. Comparing to previous design proposed in [1], this design needs only five gates for any radixes, whereas the previous one needs $r - 1$ repetition of cycle and U gates. The U gate must be Hermitian because when two U gates are active the output Q_0 must be equal to the input A_0. In the final situation three U gates are active and the output Q_0 must be equal to the $\mathrm{U}(\mathrm{A}_0)$.

Lemma 2 *Any r-valued three-qudit Hermitian controlled U gate can be implemented using five two-qudit controlled gates as shown in* Fig. 2 *in which r = radix (odd value),* $P = r - 1$, $m = r/2$, $n = m + 1$.

Proof We can prove this lemma by verifying outputs for all values of inputs as is done in Lemma 1. The Q_1 is A_1 for all values of A_1. Since the cascade of C_n and $\mathrm{C}_{-\mathrm{n}}$ is a buffer gate, Q_2 is also equal to A_2 for all values of A_1 and A_2. If the value of A_1 is less than $r - 1$, C_n, $\mathrm{C}_{-\mathrm{n}}$, and G3 are inactive. Depending on the A_2 value, both G1 and G2 gates are either active or inactive. In both cases, the Q_0 is equal to A_0. For $A_1 = \quad r - 1$ and $A_2 = 0, 1, 2, \ldots, r - 2$, two controlled U gates are active (G3 and one of G1 or G2) which results in $Q_0 = A_0$. Finally, for $A_1 = r - 1$ and $A_2 = r - 1$, we have:

$$r{-}1{=}P \rightarrow G3 \text{ is Active}$$
$$r{-}1 > m \rightarrow G1 \text{ is Active}$$
$$(r - 1 + r/2 + 1) \bmod r = (r + r/2) \bmod r = r/2 = m \quad \rightarrow G2 \text{ is Active}$$

Therefore, all three controlled U gates are active which results in $Q_0 = \mathrm{U}(A_0)$ □

To be more specific and clear, consider a radix 5 NOT gate, for example (Fig. 3). If A_1 is 4, the G3 and two cycle gates are active. In this situation, we have the values

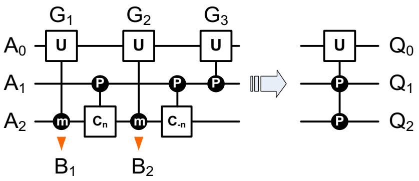


Fig. 2 Combination of gates in Lemma 2,The r is radix (odd value) and $P = r - 1,\ m \quad = r/2$ and $n = m + 1$

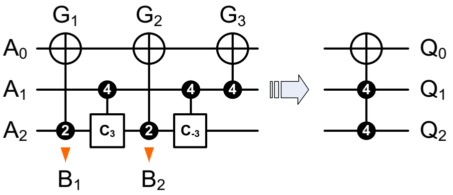


Fig. 3 A 5-valued quantum controlled NOT gate

Table 2 The values of $A_2,\ B_1$, and B_2 signals when $A_1 = 4$ (in Fig. 3)

A2	B1	B2
0	0	3
1	1	4
2	2	0
3	3	1
4	4	2

G1 is active
G2 is inactive

G1 is inactive
G2 is active

Both G1 and G2 are
active

of $A_2,\ B_1$, and B_2 signals as shown in Table 2. The m parameter which is the input threshold of G1 and G2 gates is 2. Therefore, as is shown in Table 2, just one of G1 and G2 gates are active for four first values of A_0, result in two active U gates in the A_0 to Q_0 path. As a result $Q_0 = A_0$ for all values of $A_2 = 0$, 1, 2, and 3. For the last value, $A_0 = 4$, both G1 and G2 are active which results in three active U gates in the A_0 to Q_0 path and $Q_0 = \mathrm{U}(A_0)$. The U gate must be Hermitian because when two U gates are active, the output Q$_0$ must be equal to the input A$_0$. In the last situation, three U gates are active and the output Q$_0$ has to be equal to the U(A$_0$). This gate is extension of the binary Toffoli gate to the 5-valued case.

The proposed circuits in Lemmas 1 and 2 cannot be applied to r-valued quantum gates with even values of radix r. For even radixes, we cannot find the combination of B_1 and B_2 values such that just one of G1 or G2 is active, for $r - 2$ first values of A_2. For example, we consider $r = 10$ and $A_1 = 9$. Table 3 shows the values of $A_2,\ B_1$ and B_2 for this case. The m (threshold) parameter is 5 and n is 6. As is shown in Table 3, for $A_2 = 4$, both G1 and G2 gates are inactive which results in activation of one U gate in the A_0 to Q_0 path. However, this is not our desired operation. We leave this problem to be open for future works (finding proper values for $m,\ P$, and n parameters when r is even).

The proposed circuits can be used for a variety of controlled quantum gates which are Hermitian. However, one of the most important gates, the cycle gate, is not Hermitian. For this gate, we propose a new circuit which is introduced in Lemma 3.

Table 3 The values of A_2, B_1, and B_2 signals for $r = 10$ when A_1 is 9

A0	B1	B2
0	0	6
1	1	7
2	2	8
3	3	9
4	4	0
5	5	1
6	6	2
7	7	3
8	8	4
9	9	5

G1 is active
G2 is inactive

G1 is inactive
G2 is active

Both G1 and G2 are
inactive

Both G1 and G2 are
active

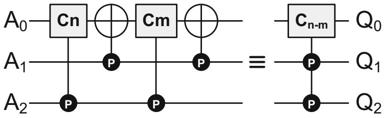


Fig. 4 Implementation of a three-qudit, r-valued controlled cycle gate: $n + m$ is equal to r and P is $r - 1$

Lemma 3 *A three-qudit, r-valued controlled cycle gate can be implemented using four two-qudit quantum gates, as shown in* Fig. 4 *in which $n + m$ is equal to r and P is $r - 1$.*

Proof We verify the operation of the circuit for all possible values of inputs. For all values of A_1 and A_2, we have: $Q_1 = A_1$ and $Q_2 = A_2$. When the inputs A_1, and A_2 are less than P, all controlled gates are inactive and we have: $Q_0 = A_0$. For $A_1 = P$ and $A_2 < P$ two NOT gates are active and $Q_0 = A_0$. For $A_1 < P$ and $A_2 = P$ two Cn and Cm gates are active. Since $n + m$ is r, the Cn+m gate is equal to a Cr = C0 gate which results in $Q_0 = A_0$. Finally, when A_1 and A_2 are equal to or greater than P and A_0 is A, the Q_0 is (Eq. 4):

$$Q_0 = (r - 1 - (r - 1 - (A + n) + m)) \bmod r = (A + n - m) \bmod r \tag{4}$$

This value of Q_0 is equivalent to a C$n - m$ gate □

Therefore, we can use this structure to implement the cycle gates. If our desired cycle gate is Cd, then we have these constraints: $m – n = d$ and $m + n = r$. By adding two equalities we have: $2m = d + r$. This equation implies that $d + r$ must be an even number. If radix is even, only even cycle gates can be implemented. For example for $r = 6$, we can implement C2 and C4 gates. For $r = 7$, we can construct C1, C3, and C5 gates.

Note that for odd values of r, we can implement the even cycle gates using this fact that the gate $\mathrm{C_{-d}}$ is equivalent to $\mathrm{C_{r-d}}$. For example $\mathrm{C_{-3}}$ for $r = 7$ is equal to $\mathrm{C_4}$. Therefore, using this trick, cycle gates with even values of n can be implemented. The only restriction of circuit is when d is odd and $r$is even.

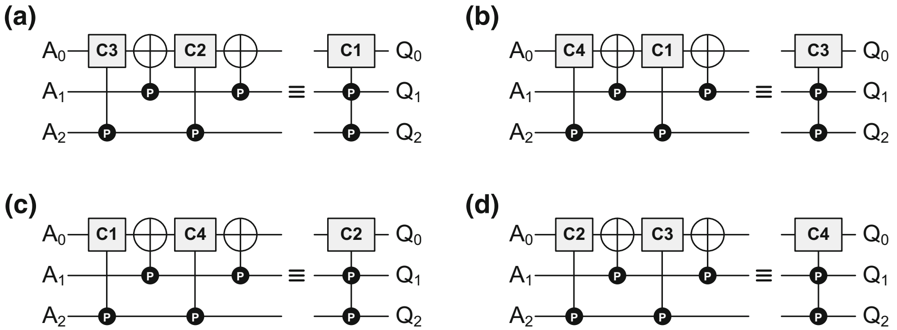


Fig. 5 Four instances of Lemma 3 for 5-valued quantum cycle gates and different values of d:, **a** 5-valued C1gate, **b** 5-valued C3 gate, **c** 5-valued C2 gate, **d** 5-valued C4 gate

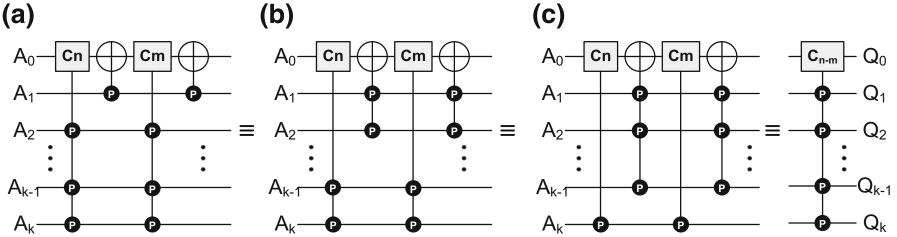


Fig. 6 Three possible extensions of cycle gate (Lemma 3) to k control input arrays

Figure 5 shows four instances of Lemma 3 for 5-valued quantum cycle gates for different values of d. The values of n and m for different values of d are calculated as follows:

For $r = 5$ and $d = 1$ we have: $n + m = 5, n - m = 1 \rightarrow n = 3, m = 2$.
For $r = 5$ and $d = 3$ we have: $n + m = 5, n - m = 3 \rightarrow n = 4, m = 1$.
For $r = 5$ and $d = 2$ we have: $\mathrm{C}_2 = \mathrm{C}_{-3} \rightarrow n + m = 5, n - m = -3 \rightarrow n = 1, m = 4$.
For $r = 5$ and $d = 4$ we have: $\mathrm{C}_4 = \mathrm{C}_{-1} \rightarrow n + m = 5, n - m = -1 \rightarrow n = 2, m = 3$.

4 Extending the arrays to the gates with *k* control inputs

The arrays which are proposed in previous section can be extended to the quantum gates with k control inputs. The extension for cycle gates (Lemma 3) is easy. In Fig. 6 three combinations are considered.

In the first circuit, Fig. 6a, the $k - 1$ controls cycle gates are used and size of the NOT gates are fixed to 2. In the second circuit, Fig. 6b, both cycle and NOT gates are extended to about $k/2$ controls (or $k/2$ if k is even). In the third circuit, Fig. 6c, the size of the cycle gates are fixed to 2, and the size of NOT gates are extended to $k - 1$ controls. We can write three recursive difference equations which describe the quantum cost of each circuit. If we show the quantum cost of a cycle and a NOT gate

Table 4 Quantum costs for three different implementations in Fig. 6 and effects on design of Fig. 7

Design	Cycle			Controlled U gate (Fig. 7)		
	Figure 6a	Figure 6b	Figure 6c	Using Fig. 6a	Using Fig. 6b	Using Fig. 6c
$k = 2$	4	4	4	5	5	5
$k = 3$	10	10	12	15	15	15
$k = 4$	22	18	32	37	37	41
$k = 5$	46	30	84	83	75	107
$k = 6$	94	50	216	177	137	277

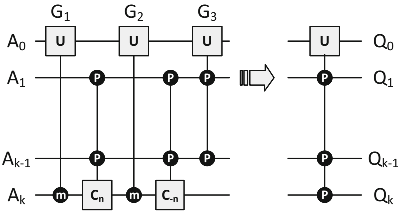


Fig. 7 Extension of a Hermitian controlled U gate to k control inputs

with k controls as $Q_{\mathrm{c}}[k]$ and $Q_{\mathrm{NOT}}[k]$, respectively; for the first design we have the Eq. 5.

$$Q_{\mathrm{c}}[k] = 2Q_{\mathrm{c}}[k-1] + 2 \tag{5}$$

For the second and third circuits, we can write Eqs. 6 and 7, respectively.

$$\begin{cases} Q_{\mathrm{c}}[k] = 2Q_{\mathrm{c}}\lceil k/2\rceil + 2Q_{\mathrm{NOT}}\lceil k/2\rceil \\ Q_{\mathrm{NOT}}[k] = 2Q_{\mathrm{c}}[k-1] + Q_{\mathrm{NOT}}[k-1] + 2 \end{cases} \tag{6}$$

$$\begin{cases} Q_{\mathrm{c}}[k] = 2Q_{\mathrm{NOT}}[k-1] + 2 \\ Q_{\mathrm{NOT}}[k] = 2Q_{\mathrm{c}}[k-1 + Q_{\mathrm{NOT}}[k-1] + 2 \end{cases} \tag{7}$$

Initial conditions for the difference equations are: $Q_{\mathrm{c}}\,[1] = 1$, $Q_{\mathrm{NOT}}\,[1] = 1$.

The Eq. 5 with mentioned initial conditions is solved using the Z-transform and the answer is Eq. 8.

$$Q_{\mathrm{c}}[k] = 3 \times 2^{k-1} - 2, \quad k = 1, 2, 3, \ldots \tag{8}$$

The Eqs. 6 and 7 can be solved numerically. Table 4 shows the quantum cost of three above designs for different values of k = 2, 3, 4, 5, and 6.

The Hermitian controlled U gates can also be extended to the k-controlled gates using a similar method; however, there is one choice for extending the gates. Figure 7 shows the equivalent circuit of a k-controlled U gate.

Table 4 also shows the quantum cost of controlled U gate for different values of k. For large values of k, the second design (Fig. 6b) is much better than two other designs because of exponential reduction of size in Eq. 6. For small values of the k, we can also use the other designs.

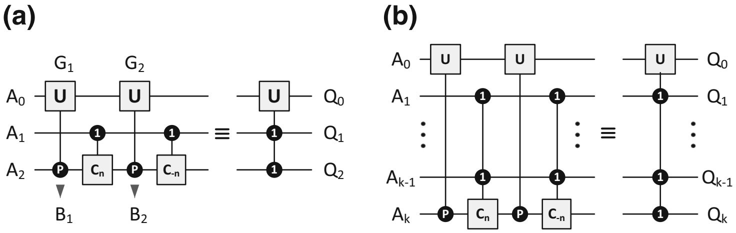


Fig. 8 Implementation of the Hermitian gates with odd radixes and thresholds of 1

Table 5 The values of B_1 and B_2 signals for all values of input A_2, when $r = 9$, $P = 5$, and $n = 4$

A_2 input	B_1	B_2	Active gates
0	0	4	None
1	1	5	G2
2	2	6	G2
3	3	7	G2
4	4	8	G2
5	5	0	G1
6	6	1	G1
7	7	2	G1
8	8	3	G1

5 Other thresholds of control inputs

In the previous sections, we have considered the same values $(r - 1)$ for all thresholds of control inputs (P). It is possible to design the circuits with other values of thresholds. The circuit in Lemma 3 can be directly used for other thresholds without any changes. Actually, in Fig. 4, the P parameter can be a desired integer value; though, the threshold of inputs can also be different for different inputs. For Lemmas 1 and 2, the circuits have to be modified. An interesting case is when all thresholds are one. In this case, we can obtain a more efficient circuit than Fig. 2. This circuit which is shown in Fig. 8a, has only four two-qudit gates (for implementing the Hermitian gates with odd radixes).

In this circuit, P is $\lceil r/2 \rceil$ and n is $P - 1$. Table 5 shows the values of B_1 and B_2 signals for all values of input A_2, when $r = 9$, $P = 5$, and $n = 4$. For $A_2 = 0$, the G1 and G2 are both inactive (for all values of A_1) results in $Q_0 = A_0$. For $A_2 = 1, 2, 3,$ and 4, the G1 is inactive and G2 is active (for $A_1 \geq 1$) results in $Q_0 = \mathrm{U}(A_0)$. Finally, For $A_2 = 5, 6, 7,$ and 8, the G1 is active and G2 is inactive (for $A_1 \geq 1$) results in $Q_0 = \mathrm{U}(A_0)$.

Figure 8b shows the extended version of the circuit to k control inputs. For this circuit, we can write the difference equation as Eq. 8.

$$Q_{\mathrm{U}}[k] = 2Q_{\mathrm{c}}[k-1] + 2 \tag{9}$$

Using Fig. 8b as equivalent circuit for cycle gate, we can obtain more efficient designs for controlled U gate. The quantum cost of this design for values of $k = 2, 3, 4, 5,$ and 6 is shown in Table 6.

Table 6 Quantum cost of circuits with threshold of 1

	Cycle	Controlled U (threshold = 1)	Controlled U (threshold = $r-1$)
$k = 2$	4	4	5
$k = 3$	10	10	15
$k = 4$	16	22	37
$k = 5$	28	34	75
$k = 6$	40	58	137

Table 7 Instances for quantum cost of proposed circuits in this paper and comparison with [1]

No.	Controlled gate	Control lines, size	QC [1]	QC (this paper)
1.	Ternary $\mathrm{C}_1, \mathrm{C}_2$	2, 3×3	4	4
2.	Quaternary Cycle	2, 3×3	5	4
3.	Ternary NOT, S	2, 3×3	5	5
4.	r-valued C_1 (odd)	2, 3×3	$2r-2$	4
5.	r-valued C_2 (r even)	2, 3×3	$2r-3$	4
6.	r-valued C_1 (r odd)	2, 3×3	$2r-2$	4
7.	r-valued C_1 (r even)	2, 3×3	$2r-3$	4
8.	r-valued NOT, S(r odd)	2, 3×3	$2r-1$	5

The second column of Table 6 shows the quantum costs for cycle gates from Table 4. Third column shows the values of quantum cost which is obtained from combination of Eqs. 5 and 8. For easier comparison of the quantum costs for two designs, we repeated the 6th column of Table 4 in Table 6. Table 6 shows that controlled gates with input thresholds of 1 can be implemented more efficient than ones with input thresholds of $r-1$.

6 Comparison

In the previous work [1], we proposed some quantum arrays to implement n-qudit r-valued circuits using two-qudit quantum gates. In this research, we developed quantum arrays whose quantum costs was independent of the radix of the circuit. For example, a three-qudit r-valued Hermitian quantum gate was implemented with quantum cost of $2r-1$ in [1], whereas it is designed with quantum cost of 5 in this paper (Fig. 2). Table 7 shows the quantum costs of the designs proposed in [1] and is compared with designs of this paper. As the Table 7 shows, for designs of this paper, the quantum costs of all designs are independent of r. We extended the proposed arrays to the quantum gates with k control inputs. This extension for cycle gates and Hermitian gates was introduced. We also considered circuits with different threshold values for inputs and designed efficient arrays for such circuits. In the case that all thresholds are 1, the designs are more efficient than when they have the largest possible values.

7 Conclusions

Designing of the large quantum controlled gates using the arrays of one-qudit or two-qudit gates was the issue of this research. In our prior work in [1], we proposed some quantum arrays to implement n-qudit r-valued circuits using two-qudit quantum gates. In this paper, we developed quantum arrays whose quantum costs was independent of the radix of the circuit. For example, a three-qudit $r-$valued Hermitian quantum gate was implemented with quantum cost of $2r - 1$ in [1], whereas it was designed with quantum cost of 5 in this paper. Table 7 summarizes the quantum costs of the designs proposed in [1] and designs of this paper for three-qudit quantum gates. We extended the proposed arrays to the larger quantum gates with more control inputs. This extension for cycle gates and Hermitian gates was introduced. We also considered the gates with different threshold values of inputs. When all input thresholds are 1, the proposed designs were more efficient than other cases.

References

1. Mohammadi, M., Niknafs, A., Eshghi, M.: Controlled gates for multi-level quantum computation. Quantum Inf. Process. **10**, 175–192 (2011)
2. Lukac, M., Pivtoraiko, M., Mishchenko, A., Perkowski, M.: Automated synthesis of generalized reversible cascades using genetic algorithms. In: Proceedings of the Fifth International Workshop on Boolean Problems, Freiberg, Sachsen, Germany, 19–20 Sept, pp. 33–45 (2006)
3. Kahn, M.H.A., Perkowski, M.: Evolutionary algorithm based synthesis of multi-output ternary functions using quantum cascade of generalized ternary gates. Int. J. Mult. Valued Log. Soft Comput. (2005)
4. Mohammadi, M., Eshghi, M.: Heuristic methods to use don't cares in automated design of reversible and quantum logic circuits. Quantum Inf. Process. J. **7**(4), 175–192 (2008)
5. Mohammadi, M., Eshghi, M.: On figures of merit in reversible and quantum logic designs. Quantum Inf. Process. J. **8**(4), 297–318 (2009)
6. Mohammadi, M., Eshghi, M., Dueck, G.W.: Design and optimization of single and multiple-loop reversible and quantum feedback circuits. World Sci. J. Circuits Syst. Comput. **21**(2), 301–315 (2012)
7. Mohammadi M.: Efficient genetic based methods for optimizing the reversible and quantum logic circuits. J. Adv. Comput. Res. **3**(3), 85–96 (2012)
8. Mohammadi, M., Niknafs, A.: Synthesis and optimization of multiple-valued combinational and sequential reversible circuits with don't cares. Integr. VLSI J. **46**(2), 189–196 (2013)
9. Miller, D.M., Maslov, D., Dueck, G.W.: Synthesis of quantum multiple-valued circuits. J. Mult. Valued Log. Soft. Comput. **12**(5/6), 431 (2006)
10. Khan, M.H.A.: Design of reversible/quantum ternary multiplexer and demultiplexer. Eng. Lett. **13**(2), 65–69 (2006)
11. Barenco, A., Bennett, C.H., Cleve, R., DiVincenzo, D.P., Margolus, N., Shor, P., Sleator, T., Smolin, J.A., Weinfurter, H.: Elementary gates for quantum computation. Phys. Rev. A **52**(5), 3457–3467 (1995)
12. Brennen, G.K., Bullock, S.S., O'Leary, D.P.: Efficient circuits for exact-universal computation with qudits. Quantum Inf. Comput. **6**, 436–454 (2006)
13. Rosenbaum, D., Perkowski, M.: Efficient implementation of controlled operations for multivalued quantum logic. In: ISMVL, 2009 39th International Symposium on Multiple-Valued Logic, pp. 86–91 (2009)
14. Khan, M.H.A., Perkowski, M.A., Kerntopf, P.: Multi-output Galois field sum of products (GFSOP) synthesis with new quantum cascades. In: Proceedings of the International Symposium on Multiple-Valued Logic, May, pp. 146–153 (2003)
15. Al-Rabadi, A.: New multiple-valued Galois field sum-of-product cascades and lattices for multiple-valued quantum logic synthesis. In: 6th International Symposium on Representations and Methodology of Future Computing Technologies, March, pp. 171–182 (2003)

16. Muthukrishnan, A., Stroud Jr, C.R.: Multivalued logic gates for quantum computation. Phys. Rev. A **62**(5), 052309/1–052309/8 (2000)
17. Picton, P.: A universal architecture for multiple-valued reversible logic. MVL J. **5**, 27–37 (2000)