Universality of 2-qudit ternary reversible gates

2006 J. Phys. A: Math. Gen. 39 7763

(http://iopscience.iop.org/0305-4470/39/24/013)

INSTITUTE OF PHYSICS PUBLISHING JOURNAL OF PHYSICS A: MATHEMATICAL AND GENERAL

J. Phys. A: Math. Gen. **39** (2006) 7763–7773 doi:10.1088/0305-4470/39/24/013

Universality of 2-qudit ternary reversible gates

Guowu Yang[1,2], Fei Xie[1], Xiaoyu Song[3] and Marek A Perkowski[3]

[1] Department of Computer Science, Portland State University, Portland, OR 97207, USA
[2] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 610054, People's Republic of China
[3] Department of Electrical and Computer Engineering, Portland State University, Portland, OR 97207, USA

E-mail: guowu@cs.pdx.edu, xie@cs.pdx.edu, song@ece.pdx.edu and mperkows@ece.pdx.edu

Received 8 February 2006, in final form 5 April 2006
Published 31 May 2006
Online at stacks.iop.org/JPhysA/39/7763

Abstract

In this paper, we investigate the synthesis of ternary reversible circuits in the absence of ancilla bits. We demonstrated that 2-qudit ternary Swap, NOT and 1-controlled-NOT gates are universal for realization of arbitrary ternary n-qudit reversible circuits without ancilla qudits, and all even ternary n-qudit reversible circuits can be constructed by ternary NOT and ternary 1-controlled-NOT gates without ancilla qudits. The realization approach is constructive. This result is significantly different from the binary case.

PACS numbers: 03.67.Lx, 03.65.Fd

1. Introduction

Quantum computation connects ideas from computer science and physics [1–3]. Reversible circuits are a necessary subclass whose realization is required for any quantum computer to be universal. Binary logic circuits play an important role in classic quantum computation as they are used to construct oracles, for instance in Grover algorithm [4]. We consider the extension of unit of memory to the ternary-valued domain, where the unit of memory is the qudit [5–7] with the three basic states $|0\rangle$, $|1\rangle$ and $|2\rangle$. Three state quantum systems have recently been discussed in the framework of cryptography [5], and the concept of a qudit cluster state has been proposed [6]. Qudit systems received further study in [8, 9] wherein quantum hybrid gates acting on tensor products of qudits of different dimensions were discussed. The ternary-valued reversible circuits are experimentally feasible in the context of the linear ion trap scheme for quantum computing [10]. Recently synthesis of d-level systems showing asymptotic optimality was also proposed [7]. The study in [8] found hybrid quantum gates that, when considered to be controlled by and act on three level quantum systems define the hybrid Toffoli, Swap and Not gates used in this paper. The physical realization of these hybrid gates might be accomplished via spin systems [8, 11] or quantum harmonic oscillators

[8, 11]. A universal set of ternary quantum gates enables the realization of any tristate switching network using candidate qudit implementation.

The computer science community has experienced interest in the universal sets of gates required for quantum computing systems; the main results of which appear in [12–19]. We observe that the universality discussed in the literature has an assumption that the inputs of gates can be set to constant values, thus ancilla bits are used [18, 19]. These synthesis approaches can be applied to large functions but their disadvantage is that they create m ancilla bits (one for each output) and use multi-input gates that may be expensive. Although [8] discussed entanglement generation with and without ancilla qudits, [20] constructively proved that all n-qudit ternary reversible circuits can be constructed by ternary Swap, ternary Not and ternary Toffoli ($(n-1)$ qudits control another qudit, so it is an n-qudit gate) gates without ancilla qudits, or by ternary NOT, controlled-NOT, multiply-two and Toffoli gates. [19] dealt with the universality of general reversible multiple-valued logic gates under the assumption that constant signals can be applied to arbitrary number of inputs. No one discussed yet the universality of two-qudit ternary gates without ancilla qudits.

Group theory [21] has found particular use to generate reversible logic circuits [13, 14, 16, 17]. The motivation of this paper is to find the universality of a gate family [14, 16, 19, 20, 22] to be used in synthesis of ternary reversible circuits without ancilla bits. We prove that 2-qudit ternary Swap, NOT and 1-controlled-Not gates [8] are universal for realization of arbitrary ternary $n \times n$ reversible circuits without ancilla bits. The realization approach is constructive, not search based. This result is significantly different from the binary case. In binary reversible circuits, only a small per cent of reversible circuits can be constructed by 2-qubit gates without ancilla qubits. In 3×3 reversible circuits, 33.3% can be constructed; in 4×4 reversible circuits, only 3×10^{-6}% can be constructed. Even using 3-qubit gates, only 50% of $n \times n$ binary reversible circuits can be constructed without ancilla qubits [23].

The rest of this paper is organized as follows. In section 2, we introduce some basic definition of ternary reversible circuit and the needed group theory notations, and terms. In section 3, we show the universality of 2-qudit ternary reversible gates. In section 4, we give two conjectures about the universality of d-level reversible gates. We conclude in section 5.

2. Ternary reversible circuit and permutation group

In this section, we first present some basic definitions of ternary reversible circuits and the needed group theory notation and terms.

Definition 1 (ternary reversible circuit)**.** *Let $A = \{0, 1, 2\}$. A ternary logic circuit f with n input variables, $A_1, \ldots, A_n$, and n output variables, $P_1, \ldots, P_n$, is denoted by $f : A^n \to A^n$, where $\langle A_1, \ldots, A_n \rangle \in A^n$ is the input vector and $\langle P_1, \ldots, P_n \rangle \in A^n$ is the output vector. There are 3^n different assignments for the input vectors. A ternary logic circuit f is reversible if it is a one-to-one and onto function (bijection). A ternary reversible logic circuit with n inputs and n outputs is also called an n-qudit ternary reversible gate. It realizes a ternary reversible circuit. There are a total of $(3^n)!$ different n-qudit ternary reversible circuits.*

We now introduce the concept of a permutation group and its relationship with reversible circuits.

Definition 2 (permutation)**.** *Let $M = \{d_1, d_2, \ldots, d_k\}$. A bijection (one-to-one, and onto mapping) of M onto itself is called a permutation on M. The set of all permutations on M*

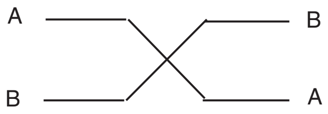

Figure 1. Swap gate.

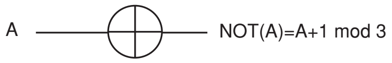

Figure 2. Ternary NOT gate.

forms a group under composition of mappings, called a symmetric group on M. It is denoted by S_k [21]. A permutation group is simply a subgroup [21] of a symmetric group.

A mapping $s : M \to M$ can be written as

$$s = \begin{pmatrix} d_1, d_2, \ldots, d_k \\ d_{i_1}, d_{i_2}, \ldots, d_{i_1} \end{pmatrix}. \tag{1}$$

Here we use a product of disjoint cycles as an alternative notation for a mapping [21]. For example,

$$\begin{pmatrix} d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9 \\ d_1, d_4, d_7, d_2, d_5, d_8, d_3, d_6, d_9 \end{pmatrix} \tag{2}$$

can be written as $(d_2, d_4)(d_3, d_7)(d_6, d_8)$. Denote '( )' as the identity mappings direct wiring and call this the unity element in a permutation group. The inverse mapping of mapping s is denoted as s^{-1}. As per convention, a product $s * t$ of two permutations applies mapping s before t.

We order the 3^n different n-input assignment vectors as

$$\langle 0, 0, \ldots, 0\rangle, \langle 1, 0, \ldots, 0\rangle, \langle 2, 0, \ldots, 0\rangle, \langle 0, 1, \ldots, 0\rangle, \ldots, \langle 2, 2, \ldots, 2\rangle,$$

and denote them by $a_1, a_2, a_3, \ldots, a_m$, where $m = 3^n$. Thus an $n \times n$ ternary reversible circuit is just a permutation in S_m (*where* $m = 3^n$), and vice versa. A ternary reversible circuit is called even (odd) reversible circuit if its corresponding permutation is even (odd) permutation. Cascading two gates is equivalent to multiplying two permutations. In what follows, no distinction between an $n \times n$ reversible gate and a permutation in S_m (where $m = 3^n$) will be made.

Definition 3 (Swap gate) (see figure 1)**.** *A Swap gate $E_{i,j}$ exchanges the ith bit A_i and the jth bit A_j, i.e. $P_i = A_j$, $P_j = A_i$; $P_r = A_r$, if $r \neq i, j$.*

Definition 4 (ternary NOT gate) (see figure 2)**.** *A Ternary NOT Gate N_j is defined as $P_j = A_j \oplus_3 1$, where $\oplus_3$ stands for addition modulo 3; $P_i = A_i$, if $i \neq j$. $1 \leqslant j \leqslant n$.*

Definition 5 (ternary k-controlled-NOT gate)**.** *A ternary k-controlled-NOT gate $C_{i_1,i_2,\ldots,i_k;j}$ is defined as*

- *If $m \neq j$, then $P_m = C_{i_1,i_2,\ldots,i_k;j}(A_m) = A_m$.*

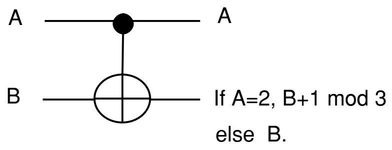

Figure 3. 1-Controlled-NOT gate.

- *If* $m = j$ *and* $A_{i_1} = \cdots = A_{i_k} = 2$*, then* $P_j = C_{i_1,i_2,\ldots,i_k;j}(A_j) = A_j \oplus_3 1$; *else,* $P_j = A_j$*.*

Remark 1. A ternary k-controlled-NOT gate $C_{i_1,i_2,\ldots,i_k;j}$ means the jth qudit A_j is controlled by the qudits $A_{i_1}, \ldots, A_{i_k}$; only the qudit A_j may change its value after the action of the ternary k-controlled gate.

3. Universality of 2-qudit ternary gates

In this section, we prove that ternary Swap, NOT and 1-controlled-NOT (see figure 3) gates are universal for realization of arbitrary ternary n-qudit reversible circuits without ancilla qudits, and all even ternary n-qudit reversible circuits can be constructed by ternary NOT and ternary 1-controlled-NOT gates without ancilla qudits.

For concise representation, we denote C_j the $(n-1)$-controlled-NOT gate whose jth qudit is controlled by the other $n-1$ qudits.

Lemma 1. *All even n-qudit* $(n \geqslant 2)$ *ternary reversible circuits can be generated by 1-qudit ternary NOT gate and n-qudit ternary* $(n-1)$*-controlled-NOT gate without ancilla qudits.*

The proof of lemma 1 is given in the appendix. The proof is similar to the proof of lemma 4 in [20]. We do not need to use Swap gate because here the controlled qudit in the $(n-1)$-controlled-NOT gate can be connected to any qudit.

Lemma 2. *All n-qudit* $(n \geqslant 2)$ *ternary reversible circuits can be generated by 2-qudit ternary NOT, Swap gates and n-qudit ternary* $(n-1)$*-controlled-NOT gate without ancilla qudits.*

Proof. A Swap gate is an odd permutation [20]. Combining with lemma 1, we have: all n-qudit $(n \geqslant 3)$ ternary reversible circuits can be generated by 2-qudit ternary NOT, Swap gates and n-qudit ternary $(n-1)$-controlled-NOT gate without ancilla qudits. □

Lemma 3. *If* $n \geqslant 5$*, and* $3 \leqslant k \leqslant n-2$*, then any k-controlled-NOT gate can be constructed by 2-controlled-NOT gates.*

Proof. We prove this lemma by an induction on k.

Case 1. Any 3-controlled-NOT gate can be constructed by 2-controlled gates.

Let $C_{i_1,i_2,i_3;j}$ be a 3-controlled-NOT gate.

Let $h \neq i_1, i_2, i_3, j$. (Because $n \geqslant 5$, so this is possible.)

Consider $C_{i_1,i_2;h}$ and $C_{h,i_3;j}$.

We will prove

$$\left(C_{i_1,i_2;h} * C_{h,i_3;j}\right)^3 = C_{i_1,i_2,i_3;j}.$$

- *Subcase* 1. $m \neq j, h$.
  Obviously,
  $$C_{i_1,i_2,i_3;j}(A_m) = A_m, \qquad C_{i_1,i_2;h}(A_m) = A_m, \qquad C_{h,i_3;j}(A_m) = A_m.$$
  Thus
  $$\big(C_{i_1,i_2;h} * C_{h,i_3;j}\big)^3(A_m) = A_m = C_{i_1,i_2,i_3;j}(A_m).$$
- *Subcase* 2. $m = h$.
  Obviously,
  $$C_{i_1,i_2,i_3;j}(A_h) = A_h.$$
  (1) If $A_{i_1} = A_{i_2} = 2$, then $C_{i_1,i_2;h}(A_h) = A_h \oplus_3 1$.
  $$\big(C_{i_1,i_2;h} * C_{h,i_3;j}\big)(A_h) = C_{h,i_3;j}(A_h \oplus_3 1) = A_h \oplus_3 1. \tag{3}$$
  So
  $$\big(C_{i_1,i_2;h} * C_{h,i_3;j}\big)^3(A_h) = A_h \oplus_3 1 \oplus_3 1 \oplus_3 1 = A_h.$$
  (2) Else,
  $$\big(C_{i_1,i_2;h} * C_{h,i_3;j}\big)^3(A_h) = A_h.$$
  Therefore, we have
  $$\big(C_{i_1,i_2;h} * C_{h,i_3;j}\big)^3(A_h) = A_h = C_{i_1,i_2,i_3;j}(A_h).$$
- Subcase 3. $m = j$
  (1) If $A_{i_1} = A_{i_2} = A_{i_3} = 2$, then
  $$C_{i_1,i_2,i_3;j}(A_j) = A_j \oplus_3 1.$$
  And
  $$\big(C_{i_1,i_2;h} * C_{h,i_3;j}\big)(A_j) = C_{h,i_3;j}(A_j) = \begin{cases} A_j \oplus_3 1, & \text{if } A_h = 2 \\ A_j, \text{else} \end{cases} \tag{4}$$
  From (3), we know that in A_h, $\big(C_{i_1,i_2;h} * C_{h,i_3;j}\big)(A_h)$, $\big(C_{i_1,i_2;h} * C_{h,i_3;j}\big)^2(A_h)$, there is only one situation such that the value is 2. Therefore, by using (4), we have
  $$\big(C_{i_1,i_2;h} * C_{h,i_3;j}\big)^3(A_j) = A_j \oplus_3 1 = C_{i_1,i_2,i_3;j}(A_j).$$
  (2) If $A_{i_1} = A_{i_2} = 2$, but $A_{i_3} \neq 2$. Obviously,
  $$C_{i_1,i_2,i_3;j}(A_j) = A_j.$$
  And,
  $$\big(C_{i_1,i_2;h} * C_{h,i_3;j}\big)(A_j) = C_{h,i_3;j}(A_j) = A_j.$$
  So,
  $$\big(C_{i_1,i_2;h} * C_{h,i_3;j}\big)^3(A_j) = A_j = C_{i_1,i_2,i_3;j}(A_j).$$
  (3) If $A_{i_3} = 2$, but $A_{i_1} \neq 2$, or $A_{i_2} \neq 2$.
  Obviously, $\big(C_{i_1,i_2;h} * C_{h,i_3;j}\big)(A_h) = C_{h,i_3;j}(A_h) = A_h$. Namely, the *h*th qudit A_h will not change. So,
  - if $A_h = 2$, then $\big(C_{i_1,i_2;h} * C_{h,i_3;j}\big)(A_j) = A_j \oplus_3 1$, thus
    $\big(C_{i_1,i_2;h} * C_{h,i_3;j}\big)^3(A_j) = A_j \oplus_3 1 \oplus_3 1 \oplus_3 1 = A_j = C_{i_1,i_2,i_3;j}(A_j)$.
  - else, then $\big(C_{i_1,i_2;h} * C_{h,i_3;j}\big)(A_j) = A_j$, thus
    $$\big(C_{i_1,i_2;h} * C_{h,i_3;j}\big)^3(A_j) = A_j = C_{i_1,i_2,i_3;j}(A_j).$$
  (4) If no value of A_{i_1}, A_{i_2}, A_{i_2} is 2, obviously,
  $$\big(C_{i_1,i_2;h} * C_{h,i_3;j}\big)^3(A_j) = A_j = C_{i_1,i_2,i_3;j}(A_j).$$

Thus,$\big(C_{i_1,i_2;h} * C_{h,i_3;j}\big)^3 = C_{i_1,i_2,i_3;j}$.

Case 2. Assume any t-controlled-NOT gate can be constructed by $(t-1)$-controlled gates. Consider a $(t+1)$-controlled-NOT gate $C_{i_1,i_2,\ldots,i_t,i_{t+1};j}$. $t+1 \leqslant n-2$. So, we can set $h \neq i_1, \ldots, i_t, i_{(t+1)}, j$, and $h \leqslant n$ (this is to say that we can find a qudit different from the qudits wired this $(t+1)$-controlled-NOT gate).

Similarly to case 1, we have

$$\left(C_{i_1,i_2,\ldots,i_t;h} * C_{h,i_{(t+1)};j}\right)^3 = C_{i_1,\ldots,i_t,i_{(t+1)};j}.$$

This equation means that a $(t+1)$-controlled-NOT gate can be constructed by a t-controlled-NOT gate and a 2-controlled-NOT gate.

Combining cases 1 and 2, lemma 3 holds. □

Remark 2. This result is similar to binary reversible control gates.

Lemma 4. *Any 2-controlled gate can be constructed by 1-controlled-NOT gates.*

Proof. Consider a 2-controlled-NOT gate $C_{i_1,i_2;j}$. We will use an algorithm minimum-length-representation (MLR) [24] implemented in a package GAP [25] to construct this 2-controlled-NOT gate. GAP (groups, algorithms and programming) is a system for computational discrete algebra, which is especially efficient for solving group theory problems. MLR is basically a breadth-first algorithm to realize any given reversible circuit by using minimal number of gates in a library. MLR includes some optimal approach to improve computation performance-based the property of reversible circuits, such as removing circuits which have been searched in the previous level and merging same circuits in the same level.

First, in order to save the memory, we set $n = 3$. Then we map an input assignment $[A_1, A_2, A_3]$ to a positive integer $1 + A_1 + 3A_2 + 9A_3$. Thus we get a one-to-one mapping from n-qudit ternary reversible circuits to a 3^n symmetry group S_{3^n}. For instance,

$$\begin{aligned}
C_{1,2;3} &= (9, 18, 27),\\
C_{1;3} &= (3, 12, 21)(6, 15, 24)(9, 18, 27),\\
C_{2;3} &= (7, 16, 25)(8, 17, 26)(9, 18, 27),\\
C_{1;2} &= (3, 6, 9)(12, 15, 18)(21, 24, 27),\\
C_{3;2} &= (19, 22, 25)(20, 23, 26)(21, 24, 27),\\
E_{1,2} &= (2, 4)(3, 7)(6, 8)(11, 13)(12, 16)(15, 17)(20, 22)(21, 25)(24, 26),\\
N_1 &= (1, 2, 3)(4, 5, 6)(7, 8, 9)(10, 11, 12)(13, 14, 15)(16, 17, 18)(19, 20, 21)\\
&\qquad (22, 23, 24)(25, 26, 27).
\end{aligned}$$

Using the algorithm MLR and GAP, we get

$$\begin{aligned}
C_{1,2;3} = {} & C_{1;2} * C_{3;2} * C_{2;3} * C_{1;3} * C_{3;2} * C_{2;3} * C_{3;2} * C_{1;3}\\
& * C_{1;2} * C_{1;3} * C_{3;2} * C_{2;3} * C_{3;2} * C_{1;3} * C_{2;3} * C_{3;2} * C_{1;2}
\end{aligned}$$

Consider that the symmetry of the controlled gates and the other $n-3$ qudits beyond the qudits A_{i_1}, A_{i_2}, A_j will not change after 1-controlled-NOT gates action in these three qudits and will not affect these 1-controlled-NOT gates. We give the construction of 2-controlled-NOT gate $C_{i_1,i_2;j}$ below:

$$\begin{aligned}
C_{i_1,i_2;j} = {} & C_{i_1;i_2} * C_{j;i_2} * C_{i_2;j} * C_{i_1;j} * C_{j;i_2} * C_{i_2;j} * C_{j;i_2} * C_{i_1;j}\\
& * C_{i_1;i_2} * C_{i_1;j} * C_{j;i_2} * C_{i_2;j} * C_{j;i_2} * C_{i_1;j} * C_{i_2;j} * C_{j;i_2} * C_{i_1;i_2}.
\end{aligned} \tag{5}$$

Thus, lemma 4 holds. □

Remark 3. This result is different from binary reversible controlled gates. In binary reversible circuits, 2-controlled-NOT gate (2-qubit Toffoli gate) cannot be constructed by 1-controlled-NOT gate (Feynman gate) even using extra qubit. Interestingly, 2-qubit Toffoli gate can be constructed from 1-controlled-NOT, square-root-NOT and square-root-NOT adjoint gates [3]. In the binary case, one has to go outside the realm of reversible logic, to quantum operations, while in the ternary case this is not necessary and it is sufficient to stay within multiple-valued reversible domain. This may have application to future non-quantum reversible technologies that would rely on 2-bit gates only! Ternary logic will have a clear advantage over the binary logic. However, it also has a disadvantage with respect to binary logic: in ternary logic a Swap gate cannot be synthesized with the help of a finite number of controlled-NOT gates, whereas in binary logic a Swap gate can be replaced by a cascade of merely three 1-controlled-NOT gates.

Lemma 5. *Any $(n-1)$-controlled-NOT gate can be constructed by $(n-2)$-controlled-NOT gates when $n \geqslant 3$.*

Proof. Consider a $(n-1)$-controlled-NOT gate $C_{i_1,i_2,\ldots,i_{n-1};j}$.

We denote $i_x = i_3, \ldots, i_{n-1}$ for simple expression if $n > 3$. For instance, $C_{i_1,i_2,\ldots,i_{n-1};j} = C_{i_1,i_2,i_x;j}$. We will prove that

$$\begin{aligned} C_{i_1,i_2,i_x;j} = C_{i_1,i_x;i_2} * C_{j,i_x;i_2} * C_{i_2,i_x;j} * C_{i_1,i_x;j} * C_{j,i_x;i_2} \\ * C_{i_2,i_x;j} * C_{j,i_x;i_2} * C_{i_1,i_x;j} * C_{i_1,i_x;i_2} * C_{i_1,i_x;j} * C_{j,i_x;i_2} \\ * C_{i_2,i_x;j} * C_{j,i_x;i_2} * C_{i_1,i_x;j} * C_{i_2,i_x;j} * C_{j,i_x;i_2} * C_{i_1,i_x;i_2}. \end{aligned} \tag{6}$$

If $n = 3$, equation (6) becomes equation (5). So it is true.

If $n > 3$,

- if there is a qudit A_m in A_{i_x} whose value is not 2. According to definition 5 of controlled gate, for any input qudit A_h, the output value P_h, no matter under the action of the left of equation (6) or the right, will not change.
- if all qudits in A_{i_x} whose values are 2, then all qudits in A_{i_x} will keep the value 2 under the action of these 2-controlled-NOT gates on the right-hand side of equation (6). Therefore, only three qudits A_{i_1}, A_{i_2}, A_j will change their values in the action of the controlled gates in equation (6). The process is the same as equation (5).

Therefore, equation (6) holds. That is to say, lemma 5 is true. □

Theorem 1. *All n-qudit ($n \geqslant 2$) ternary reversible circuits can be generated by 2-qudit ternary gates: Swap, NOT and 1-controlled-NOT gates without ancilla qudits.*

Proof. We will prove this theorem by the value case of n.

Case 1. $n = 2$. It is lemma 2.

Case 2. $n = 3$. From lemmas 2 and 4 this theorem is true.

Case 3. $n = 4$. Lemma 5 tells us that 3-controlled-NOT gate can be constructed by 2-controlled-NOT gate, and lemma 4 tells us that 2-controlled-NOT gate can be constructed by 1-controlled-NOT gate. Invoking lemma 2, this theorem is true.

Case 4. $n \geqslant 5$. Lemma 5 tells us that $(n-1)$-controlled-NOT gate can be constructed by $(n-2)$-controlled-NOT gate, and lemma 3 tells us that $(n-2)$-controlled-NOT gate can be constructed by 2-controlled-NOT gate. Invoking lemmas 4 and 2, this theorem is true.

Combining these four cases, we finish the proof of theorem 1. □

Remark 4. In binary reversible logic, the result is very different. Without adding ancilla bits, using 2-qubit gates (Not, Swap, 1-CNOT gates, etc) as the synthesis library, only 33.3% of all 3-qubit reversible functions can be synthesized, and only $3 \times 10^{-6}\%$ of all 4-qubit reversible functions can be synthesized.

Corollary 1. *If a set of k-qudit ternary reversible gates can generate all k-qudit ternary reversible circuits without ancilla qudit, then all n-qudit ($n > k$) ternary reversible circuits can be generated by these k-qudit ternary reversible gates without ancilla qudits.*

Proof. A set of k-qudit ternary reversible gates can generate all k-qudit ternary reversible circuits without ancilla qudit. Thus, Swap, NOT, 1-controlled-NOT gates can be generated by these gates. Invoking theorem 1, this corollary holds. □

Definition 6 (ternary multiply-two gate)**.** *A ternary multiply-two gate MT_i is defined as: $P_i = B_i \otimes_3 2$; $P_m = B_m$, if $m \neq i$, where $\otimes_3$ is the operation of multiplication by modulo 3. $1 \leqslant i \leqslant n$.*

Corollary 2. *All n-qudit ternary reversible circuits can be generated by Not, 1-controlled-Not and multiply-two gates without ancilla qudits.*

Proof. Using algorithm MLR in [24], we obtain

$$E_{i;j} = MT_i * C_{j;i} * C_{i;j} * C_{i;j} * MT_j * C_{i;j} * C_{j;i} * C_{j;i} * MT_i * C_{j;i} * C_{i;j} * C_{i;j}.$$

Therefore, all $n \times n$ ternary reversible circuits can be generated by Not, 1-controlled-Not, multiply-two gates without ancilla qudits. □

Theorem 2. *All even n-qudit ($n \geqslant 2$) ternary reversible circuits can be generated by 2-qudit ternary gates: NOT and 1-controlled-NOT gates without ancilla qudits.*

Proof. This is a direct result in terms of lemmas 1, 3, 4 and 5. □

4. Conjecture

In this section, we give some conjectures about the universality of general multiple-valued logic gates without ancilla qudits.

Definition 7 (d-level reversible circuit)**.** *Let $A = \{0, 1, \ldots, d\}$. A d-level logic circuit f with n input variables, $A_1, \ldots, A_n$, and n output variables, $P_1, \ldots, P_n$, is denoted by $f : A^n \to A^n$, where $\langle A_1, \ldots, A_n\rangle \in A^n$ is the input vector and $\langle P_1, \ldots, P_n\rangle \in A^n$ is the output vector. There are d^n different assignments for the input vectors. A d-level logic circuit f is reversible if it is a one-to-one and onto function (bijection). A d-level reversible logic circuit with n inputs and n outputs is also called an n-qudit d-level reversible gate. There are a total of $(d^n)!$ different n-qudit d-level reversible circuits.*

Definition 8 (d-level NOT gate)**.** *A d-level NOT gate N_j is defined as $P_j = A_j \oplus_d 1$, where $\oplus_d$ stands for addition modulo d; $P_i = A_i$, if $i \neq j$. $1 \leqslant j \leqslant n$.*

Definition 9 (d-level k-controlled gate)**.** *A d-level k-controlled gate $C_{i_1,i_2,\ldots,i_k;j}$ is defined as*

- *If $m \neq j$, then $P_m = C_{i_1,i_2,\ldots,i_k;j}(A_m) = A_m$.*
- *If $m = j$ and $A_{i_1} = \cdots = A_{i_k} = 2$, then $P_j = C_{i_1,i_2,\ldots,i_k;j}(A_j) = A_j \oplus_d 1$, else, $P_j = A_j$.*

Table 1. Encoding input vectors of 2-qudit reversible circuit.

A_1	A_2	Encode
0	0	x_1
1	0	x_2
2	0	x_3
2	1	x_4
1	1	x_5
0	1	x_6
0	2	x_7
1	2	x_8
2	2	x_9

Conjecture 1. *If d is an odd prime number, then all n-qudit* $(n \geqslant 2)$ *d-level reversible circuits can be generated by 2-qudit d-level gates: Swap, d-level NOT and d-level 1-controlled-NOT gates without ancilla qudits.*

Conjecture 2. *If d is an odd prime number, then all even n-qudit* $(n \geqslant 2)$ *d-level reversible circuits can be generated by 2-qudit d-level gates: d-level NOT and d-level 1-controlled-NOT gates without ancilla qudits.*

5. Conclusions

We demonstrated that ternary Swap, NOT and 1-controlled-NOT gates are universal for realization of arbitrary ternary n-qudit reversible circuits without ancilla qudits, and all even ternary n-qudit reversible circuits can be constructed by ternary NOT and ternary 1-controlled-NOT gates without ancilla qudits. The realization approach is constructive and can be further used to develop software for synthesis of arbitrary d-level circuits without ancilla qudits. Our results demonstrated that ternary reversible circuits have very different properties than binary reversible circuits. In binary reversible circuits, only a small per cent of reversible circuits can be constructed by 2-qubit gates without ancilla qubits. In 3×3 reversible circuits, 33.3% can be constructed; in 4×4 reversible circuits, only $3 \times 10^{-6}\%$ can be constructed. Even using 3-qubit gates, only 50% of $n \times n$ binary reversible circuits can be constructed without ancilla qubits. We gave two conjectures about the universality of general d-level reversible gates.

Acknowledgment

We thank the referees for their useful and sensible comments.

Appendix. Proof of lemma 1

Lemma 1. *All even n-qudit* $(n \geqslant 3)$ *ternary reversible circuits can be generated by 1-qudit ternary NOT gate and n-qudit ternary* $(n-1)$*-controlled-NOT gate without ancilla qudits.*

Proof. Similar to the binary reflective Gray code [26], we can also reflectively encode the ternary vectors in an order $x_1, x_2, \ldots, x_m$, where $m = 3^n$ such that there is only one bit different between two vectors x_i and x_{i+1}, for $1 \leqslant i \leqslant m-1$. For instance, if $n = 2$, all nine input vectors can be encoded in table 1 thorough reflecting the column $(0, 1, 2)^T$ two times for qudit A_1 and adding three 0's, three 1's and three 2's for qudit A_2. If $n = 3$, then reflect table 1 two times and add nine 0's, nine 1's and nine 2's for qudit A_3.

(x_i, x_{i+1}, x_{i+2}) where $1 \leqslant i \leqslant m-2$ are called a neighbouring 3-cycle. All even permutations can be decomposed to a product of some neighbouring 3-cycles [21]. Therefore we only need to prove that every neighbouring 3-cycles (x_i, x_{i+1}, x_{i+2}) can be constructed by NOT and $(n-1)$-controlled-NOT gates without ancilla qudits.

There are at most two different qudits among x_i, x_{i+1}, x_{i+2}. We can assume that the same qudits among x_i, x_{i+1}, x_{i+2} are valued 2. The reason is the following. If the first qudit among x_i, x_{i+1}, x_{i+2} is not 2. Let y_k be the vector whose first qudit is 2, and the other qudits are the same as those in $x_k, k = i, i+1, i+2$. If the first qudit among x_i, x_{i+1}, x_{i+2} is valued 0, then $(x_i, x_{i+1}, x_{i+2}) = N_1 * N_1 * (y_i, y_{i+1}, y_{i+2}) * N_1$. If the first qudit among x_i, x_{i+1}, x_{i+2} is valued 1, then $(x_i, x_{i+1}, x_{i+2}) = N_1 * (y_i, y_{i+1}, y_{i+2}) * N_1 * N_1$.

Case 1. There is only one different qudit A_j among x_i, x_{i+1}, x_{i+2}.

There are only two situations for the qudit A_j's values of x_i, x_{i+1}, x_{i+2}: 0, 1, 2 or 0, 2, 1.

In the first situation, $(x_i, x_{i+1}, x_{i+2}) = C_j$.

In the first situation, $(x_i, x_{i+1}, x_{i+2}) = C_j * C_j$.

Case 2. There are two different qudit A_j and A_k among x_i, x_{i+1}, x_{i+2}. Similar to subcases 2 and 1 in case 2 in the proof of lemma 4 in [20], using NOT and $(n-1)$-controlled-NOT gates, change the two different values of A_j to three different values, then change the two different values to the same value. This becomes case 1. Change the order of the vectors by C_j or C_j*C_j. Finally, using the inverses of NOT and $(n-1)$-controlled-NOT gates, we get the construction of (x_i, x_{i+1}, x_{i+2}) by NOT and $(n-1)$-controlled-NOT gates without ancilla qudits (the inverse property: $(N_i)^{-1} = N_i * N_i$, $(N_i * N_i)^{-1} = N_i$, $(C_i)^{-1} = C_i * C_i$, $(C_i * C_i)^{-1} = C_i$). □

The following example illustrates this process.

Let $n = 3$, $u = \langle 2, 0, 2\rangle$, $s = \langle 2, 1, 2\rangle$, $t = \langle 1, 1, 2\rangle$.

$$\begin{bmatrix} u \\ s \\ t \end{bmatrix} = \begin{bmatrix} 2,0,2 \\ 2,1,2 \\ 1,1,2 \end{bmatrix} \xrightarrow{C_2 * C_2} \begin{bmatrix} 2,2,2 \\ 2,0,2 \\ 1,1,2 \end{bmatrix} \begin{pmatrix} \text{Change the values} \\ \text{of} A_2 \text{to three} \\ \text{different values} \end{pmatrix}$$

$$\xrightarrow{N_2} \begin{bmatrix} 2,0,2 \\ 2,1,2 \\ 1,2,2 \end{bmatrix} \xrightarrow{C_1} \begin{bmatrix} 2,0,2 \\ 2,1,2 \\ 2,2,2 \end{bmatrix} \begin{pmatrix} \text{Change the} \\ \text{values of} A_1 \\ \text{to the same value} \end{pmatrix} \begin{pmatrix} \text{Now it} \\ \text{becomes} \\ \text{case1} \end{pmatrix}$$

$$\xrightarrow{C_2} \begin{bmatrix} 2,1,2 \\ 2,2,2 \\ 2,0,2 \end{bmatrix} \begin{pmatrix} \text{This changes} \\ \text{the order of} \\ \text{the three vectors} \end{pmatrix} \begin{pmatrix} \text{Then use the inverses of} \\ N_i \text{ and } C_i \text{ before this } C_2 \\ \text{to complete the construction.} \end{pmatrix}$$

Therefore,

$(u, s, t) = C_2 * C_2 * N_2 * C_1 * C_2 * C_1 * C_1 * N_2 * N_2 * C_2$.

In the above equation the latter part $C_1 * C_1 * N_2 * N_2 * C_2$ is the inverse of the former part $C_2 * C_2 * N_2 * C_1$.

References

[1] Fredkin E and Toffoli T 1982 Conservative logic *Int. J. Theor. Phys.* **21** 219–53

[2] Feynman R 1985 Quantum mechanical computers *Opt. News* **11** 11–20

[3] Nielsen M and Chuang I 2000 *Quantum Computation and Quantum Information* (Cambridge: Cambridge University Press)

[4] Grover L K 1996 A fast quantum mechanical algorithm for database search, online available http://arxiv.org/abs/quant-ph/9605043

[5] Pasquinucci H B and Peres A 2000 Quantum cryptography with 3-state systems *Phys. Rev. Lett.* **85** 3313

[6] Zhou D L, Zeng B, Xu Z and Sun C P 2003 Quantum computation based on d-level cluster state *Phys. Rev.* A **68** 062303

[7] Bullock S S, O'Leary D P and Brennen G K 2005 Asymptotically optimal quantum circuits for d-level systems *Phys. Rev. Lett.* **94** 230502

[8] Daboul J, Wang X and Sanders B C 2003 Quantum gates on hybrid qudits *J. Phys. A: Math. Gen.* **36** 7063–78

[9] Kunio F 2003 The controlled-U and unitary transformation in two-qudit online available http://arxiv.org/abs/quant-ph/0304078

[10] Muthukrishnan A and Stroud C R Jr 2000 Multivaluved logic gates for quantum computation *Phys. Rev.* A **62** 052309

[11] Bartlett S, de Guise D and Sanders B 2002 Quantum encodings in spin systems and harmonic oscillators *Phys. Rev.* A **65** 052316

[12] Sasao T and Kinoshita K 1979 Conservative logic elements and their universality *IEEE Trans. Comput.* **28** 682–5

[13] Storme L, De Vos A and Jacobs G 1999 Group theoretical aspects of reversible logic gates *J. Universal Comput. Sci.* **5** 307–21

[14] De Vos A, Raa B and Storme L 2002 Generating the group of reversible logic gates *J. Phys. A: Math. Gen.* **35** 7063–78

[15] Picton P 2000 A universal architecture for multiple-valued reversible logic *Multiple-Valued Log. J.* **5** 27–37

[16] Song X, Yang G, Perkowski M and Wang Y 2006 Algebraic characterization of reversible logic gates *Theory Comput. Syst.* **39** 311–9

[17] Yang G, Hung W N N, Song X and Perkowski Marek 2005 Majority-based reversible logic gates *Theor. Comput. Sci.* **334** 259–74

[18] Miller D M, Maslov D and Dueck G 2005 Synthesis of quantum multiple-valued circuits *J. Multiple-Valued Log. Soft Comput.* http://www.cs.uvic.ca/ mmiller/ at press

[19] Kerntopf P, Perkowski M A and Khan M H A 2006 On universality of general reversible multiple-valued logic gates *J. Multiple-Valued Log. Soft Comput.* January 1–13

[20] Yang G, Song X, Perkowski M A and Wu J 2005 Realizing ternary quantum switching networks without ancilla bits *J. Phys. A: Math. Gen.* **38** 9689–97

[21] Dixon J D and Mortimer B 1996 *Permutation Groups* (New York: Springer)

[22] Toffoli T 1981 Bicontinuous extensions of invertible combinatorial functions *Math. Syst. Theory* **14** 13–23

[23] Shende V V, Prasad A K, Markov I L and Hayes J P 2003 Synthesis of reversible logic gates *IEEE Trans. CAD* **22** 710–22

[24] Yang G, Song X, Hung W N N and Perkowski M 2005 Fast synthesis of exact minimal reversible circuits using group theory *Asia and South Pacific Design Automation Conference (ASP-DAC) (Shanghai, China, Jan. 2005)* (ACM/IEEE) pp 1002–5

[25] The GAP Group 2005 *GAP—Groups, Algorithms, and Programming* Version 4.4

[26] Sandige R S 2002 *Digital Design Essentials* (Englewood Cliffs, NJ: Prentice-Hall)