Quantum-Fourier-transform-based quantum arithmetic with qudits

Archimedes Pavlidis [*]
Department of Informatics, University of Piraeus, 80, Karaoli & Dimitriou Str., GR 185 34 Piraeus, Greece

Emmanuel Floratos [†]
Department of Physics, National and Kapodistrian University of Athens, Panepistimiopolis, Ilissia, GR 157 84 Athens, Greece and Institute of Nuclear and Particle Physics, N.C.S.R. Demokritos, 27, Neapoleos Str., Agia Paraskevi, GR 153 41 Athens, Greece

(Received 6 June 2020; accepted 29 January 2021; published 16 March 2021)

We present some basic integer arithmetic quantum circuits, such as adders and multiplier-accumulators of various forms, which operate on multilevel qudits. The integers to be processed are represented in an alternative basis after they have been Fourier transformed. Several arithmetic circuits operating on Fourier-transformed integers have appeared in the literature for two-level qubits. Here we extend these techniques to multilevel qudits, as they may offer some advantages relative to qubit implementations. The arithmetic circuits presented here can be used as basic building blocks for higher level algorithms such as quantum phase estimation, quantum simulation, quantum optimization, etc. Detailed decomposition is given down to elementary two-level single- and two-qudit gates as such gates are the most appropriate for physical implementation. A complexity analysis is given after this decomposition step and it is shown that the depth of the circuits is linear in the number of qudits employed and quadratic in the dimension of each qudit while their quantum cost is quadratic in the number of the qudits and quadratic in the dimension.

DOI: 10.1103/PhysRevA.103.032417

I. INTRODUCTION

A quantum computer is a finite-dimensional quantum system composed of qubits, performing various unitary operations on the qubits (quantum gates) as well as quantum measurements. Alternatively, d-dimensional *qudits* can be used instead of the two-dimensional qubits. *Qutrit* is a special name for the case $d = 3$, while *ququart* corresponds to $d = 4$. In many cases, the employment of a multivalued quantum logic is more natural, e.g., in ion traps we could exploit more than two energy levels. Multiple laser beams could be used to manipulate the transitions between these levels [1].

Working with qudits instead of qubits may offer some advantages. The required number of qudits is smaller by a factor $\log_2 d$ compared to the number of qubits for the same dimension a quantum computer has to explore, e.g., the dimension of a composite system of n qubits is 2^n, while the same dimension can be reached with only $\log_d 2^n = \log_2 2^n / \log_2 d = n/\log_2 d$ qudits. Such a reduction of quantum information physical carriers is advantageous, considering the difficulty of reliably controlling a large number of them. When fewer quantum information carriers are used, a decrease in the overall decoherence is expected, helping to alleviate scalability issues [1,2].

Another advantage, which is also related to the adverse effect of decoherence, is that fewer multilevel qudit gates are required to construct a quantum circuit implementing a given unitary operation compared to the case of using two-level gates [1,3,4]. Fewer gates reduce the number of steps needed to complete the circuit operation (depth) and, consequently, less errors are accumulated during the overall operation of the circuit. Even so, protection of quantum information against environmental interaction is inevitable. Quantum error correcting codes and fault-tolerant gate constructions to combat decoherence on multilevel qudits have been reported and they are similar to the ones used for the qubit case [5–7]. Moreover, it is shown that working with qudits instead of qubits offers substantial advantages in terms of the fault-tolerance-induced overhead bottleneck [8,9]. An assortment of single- and two-qudit quantum gates have been proposed or experimentally realized on various technologies, e.g., ion traps [1,10], superconducting [11,12], and optical [13,14] systems.

At a higher level, generalizations of known quantum algorithms and circuits using d-level qudits may offer improvements with respect to their qubits' implementation counterparts, e.g., quantum phase estimation, which is both a core part of Shor's algorithm [15] and used in quantum simulation [16], is improved in terms of success probability when multilevel qudits are incorporated [17]. A qudit version of the Deutsch-Josza algorithm has been reported in Ref. [18] and may find applications in image processing, while an implementation proposal for five-level superconducting qudits appeared in Ref. [19]. A qudit version of Grover's algorithm [20] has been reported in Ref. [21] and offers a trade-off between space and time.

In this paper, we present some quantum arithmetic circuits operating on d-level qudits by extending results given in prior works [22–24]. These circuits exploit the quantum Fourier

[*]adp@unipi.gr
[†]mflorato@phys.uoa.gr

transform (QFT) and various single-qudit and two-qudit rotation gates to perform the desired calculations. Processing in the Fourier domain may offer some advantages related to speed [24] and robustness to decoherence [25,26]. Among the proposed circuits are various versions of adders (adder with constant, generic adder, adder with constant controlled by single qudit) and multipliers [multiplier with constant and accumulator (MAC), multiplier with constant]. Such circuits are useful in many quantum algorithms, e.g., quantum phase estimation and quantum simulation. The extension of the QFT method to process integers in the multilevel qudit case demands the definition of suitable generalized rotation gates for qudits. These gates are derived from the corresponding qudit QFT circuits and they are decomposed down to more elementary two-level qudit rotation gates. Approximation of such gates by a sequence of qudit gates belonging to a discrete set can be done using techniques similar to the qubit case. The depth of the presented circuits is linear in the length of integers processed and quadratic with respect to the dimensionality of the qudits (when decomposition to two-level qudit gates is required). Their quantum cost is quadratic in the length of integers and quadratic in the qudit dimensionality.

The rest of the paper is organized as follows: A short background about design and synthesis of qudits quantum circuits is given in Sec. II. The QFT definition and its circuit for q qudits of d levels is presented in Sec. III along with the definition of the basic rotation gates used in the proposed circuits. Section IV introduces an adder of two integers of q qudits [depth $O(q)$ including the QFTs, width $2q$], an adder of q qudits with a constant [depth $O(q)$ including the QFTs, width q], a single state controlled adder of a q qudits integer with a constant [depth $O(q)$, width $q+1$], and a generalized controlled adder of an integer of q qudits with a constant [depth $O(q)$, width $q+1$]. In Sec. V, a MAC [depth $O(q)$, width $2q$], and a multiplier with constant [depth $O(q)$, width $2q$ of which q qudits are ancilla] are presented. All these integer processing units accept one of their operands after it has been Fourier transformed. Detailed complexity analysis in terms of quantum cost, depth, and width is reported in Sec. VI. In the Appendix, we present a library of elementary qudit gates operating on two levels and which can be used to synthesize the required gates of the proposed arithmetic circuits. Finally, we conclude in Sec. VII.

II. BACKGROUND AND RELATED WORK

A d-level single-qudit gate is represented by a unitary matrix of dimensions $d \times d$, while a two-qudit gate is represented by a unitary matrix of dimensions $d^2 \times d^2$. Two single-qudit gates V_1 and V_2 operating on two different qudits can be seen as a two-qudit gate which is the tensor product $U = V_1 \otimes V_2$. However, not every two-qudit gate can be decomposed as a tensor product of two single-qudit gates, in which case we have an *entangling* gate. Thus, quantum computing in a qudit-based system works analogously to that of a qubit-based system.

It is known that single-qudit gates and a two-qudit gate alone are adequate to form a *universal* set of gates, provided that the two-qudit gate is an entangling gate [27]. A universal set of gates can be used to approximate any target quantum circuit with arbitrary precision. Various gate libraries and methods to synthesize a circuit have been introduced in the literature, like spectral decomposition [1], Cosine-Sine decomposition [28], QR decomposition [29], and Shannon decomposition [10]. The previous methods and results are similar to the two-level qubits synthesis cases. It is proven in Ref. [30] that the cost of the resulting circuit in terms of two-qudit gates is upper bounded by $O(d^{2n})$, where n is the qudits number. Thus, these automated methods are suitable only for small quantum circuits due to their exponential cost.

When the target circuit is an arithmetic or logic block which is prepresented by a permutation matrix consisting of elements 0 and 1, multiple-valued classical reversible synthesis methods can be applied. These methods are an extension of the binary reversible logic case and may be applied to a specific value of d, e.g., [31] ($d = 3$), or applied to any value of d [32,33]. Similar to the quantum synthesis case, these algorithms are not suitable for large circuits.

As many algorithms widely use quantum arithmetic blocks like adders or multipliers recurrently, it is crucial to have available efficient arithmetic and logic blocks. *Ad hoc* design of such blocks usually offers better results compared to the automated synthesis methods. One can exploit the iterative and regular structure of these arithmetic blocks or extend known classical designs to the quantum case. A diversity of *ad hoc* designed quantum arithmetic and logic circuits for two-level qubits can be found in the literature [22,24,34–39] but few (usually adders) are known for multilevel qudits. A few examples of such designs are a ternary ($d = 3$) adder for three inputs [32] as a byproduct of a synthesis method, ternary adders-subtractors *ad hoc* designed for any number of inputs [40] and *quaternary* ($d = 4$) comparators [41]. A ternary extension of the well-known Vedral-Barenco-Ekert [34] ripple-carry adder is reported in Ref. [42]. In Ref. [43], two ternary extensions of the Cuccaro-Draper-Kutin-Moulton [35] ripple carry adder and the Draper-Kutin-Rains-Svore [37] carry look-ahead adder are given.

The previous qudit arithmetic circuits are designed for a specific value of d. In contrast, the proposed designs are parametrized for any value of d. The majority of qudit rotation gates used in our proposed designs are synthesized using elementary gates introduced in Ref. [10], where physical implementation directions are also given. As the rotation angles of these gates vary with the size of the circuit and small angles are required, implementation and fault tolerance issues are briefly discussed in Sec. VII.

III. QUANTUM FOURIER TRANSFORM AND BASIC GATES

The QFT definition is

$$|j\rangle \xrightarrow{\mathrm{QFT}_N} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{i2\pi}{N} jk} |k\rangle \tag{1}$$

on the N-dimensional computational basis $\{|0\rangle, |1\rangle, \ldots, |N-1\rangle\}$.

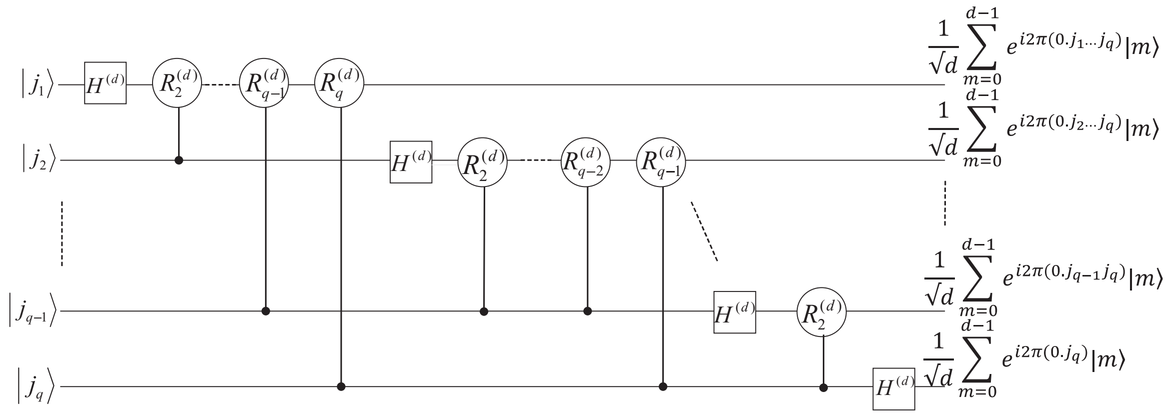

FIG. 1. QFT circuit implemented on d-level qudits. The order of the qudits must be reversed at the end.

Using q qudits of d levels [2,44], and setting $N = d^q$, the q qudits basis consists of $|j\rangle = |j_1 \dots j_q\rangle = |j_1\rangle \dots |j_q\rangle$, where for the lth qudit it holds $|j_l\rangle \in \{|0\rangle, \dots, |d-1\rangle\}$. Then, the QFT action on a basis state $|j\rangle$ ($j = 0 \dots d^q - 1$) is

$$
\begin{aligned}
|j\rangle = |j_1 \dots j_q\rangle &\xrightarrow{\mathrm{QFT}_N} \frac{1}{\sqrt{d^q}} \\
&\times \sum_{k_1=0}^{d-1} \cdots \sum_{k_q=0}^{d-1} e^{\frac{i2\pi}{d^q} j \sum_{l=1}^{q} k_l d^{q-l}} |k_1 \dots k_q\rangle \\
&= \frac{1}{\sqrt{d^q}} \sum_{k_1=0}^{d-1} \cdots \sum_{k_q=0}^{d-1} \bigotimes_{l=1}^{q} e^{i2\pi j k_l d^{-l}} |k_l\rangle \\
&= \frac{1}{\sqrt{d^q}} \bigotimes_{l=1}^{q} \sum_{k_l=0}^{d-1} e^{i2\pi j k_l d^{-l}} |k_l\rangle \\
&= \frac{1}{\sqrt{d^q}} \left(\sum_{m=0}^{d-1} e^{i2\pi(0.j_q)m} |m\rangle \right) \left(\sum_{m=0}^{d-1} e^{i2\pi(0.j_{q-1}j_q)m} |m\rangle \right) \\
&\times \cdots \left(\sum_{m=0}^{d-1} e^{i2\pi(0.j_1 j_2 \dots j_{q-1} j_q)m} |m\rangle \right). \qquad (2)
\end{aligned}
$$

The d-ary representation $(j_1 j_2 \dots j_q)$ of the integer $j = j_1 d^q + j_2 d^{q-1} + \cdots + j_q$ as well as the fractional d-ary representation $(0.j_1 j_{2\dots} j_q) = j_1/d + j_2/d^2 + \dots + j_q/d^q$ are used in the above definition. This tensor product form is similar to the form of the order 2^n QFT which is implemented on n qubits of two levels. Thus, the structure of an order d^q QFT circuit implemented with qudits is similar to the binary QFT case as depicted in Fig. 1, where in place of the the usual Hadamard and rotation qubit gates, their generalization in d-dimensional qudits is used.

The definition of the generalized d-dimensional Hadamard gate is

$$
H^{(d)} = \frac{1}{\sqrt{d}} \sum_{j=0}^{d-1} \sum_{m=0}^{d-1} e^{i\frac{2\pi}{d} jm} |j\rangle\langle m|, \qquad (3)
$$

while the definition of the generalized d-dimensional controlled rotation two-qudit gates $R_k^{(d)}$ is

$$
\begin{aligned}
R_k^{(d)} &= \sum_{j=0}^{d-1} \sum_{m=0}^{d-1} e^{i\frac{2\pi}{d^k} jm} |j\rangle\langle j| \otimes |m\rangle\langle m| \\
&= \sum_{j=0}^{d-1} \sum_{m=0}^{d-1} e^{i2\pi(0.\underbrace{00\dots0}_{k-1} j)m} |j\rangle\langle j| \otimes |m\rangle\langle m|. \qquad (4)
\end{aligned}
$$

Taking into account the above two definitions, the effect of $H^{(d)}$ on a computational basis state as well as the effect of $R_k^{(d)}$ to an arbitrary superposition state, as analyzed in the Appendix, we can confirm that the network of Fig. 1 corresponds to Eq. (2). Indeed, comparing the state $\sum_{m=0}^{d-1} e^{i2\pi(0.j_l j_{l+1} \dots j_{q-1} j_q)m} |m\rangle$ of the lth qudit after the transformation of Eq. (2) with Eqs. (A5) and (A16), we can conclude that this state can be generated by applying at the basis state $|j_l\rangle$ of the lth qudit a Hadamard gate $H^{(d)}$ and a sequence of $q-l$ generalized rotation gates $R_k^{(d)}$, with $k = 2 \dots q-l+1$, controlled by the qudits $l+1 \dots q$, respectively. At the end, the order of the qudits must be reversed with swap gates as in the case of the QFT operated on qubits. This swapping of the qudits is not shown in Fig. 1. An inverse QFT circuit is derived by reversing horizontally the direct QFT circuit of Fig. 1 (including the SWAP gates not shown) with opposite signs in the angles of the rotation gates.

IV. ADDERS

The integer arithmetic circuits presented in this section are developed in a bottom-up succession, starting from the simpler ones and gradually proceeding to more complex ones. The arithmetic operations are assumed to be modulo d^q, where d is the qudit dimension and q is the number of qudits used to represent the integers. All the adders can be easily converted to subtractors using opposite signs in the angles of the rotation gates while retaining the same circuit structure.

A. Adder of two integers

A basic arithmetic operation block is an adder of two integers of q d-ary digits each, e.g., $a = (a_1 a_2 \dots a_q)$ and $b = (b_1 b_2 \dots b_q)$ or two superpositions of integers. Following the

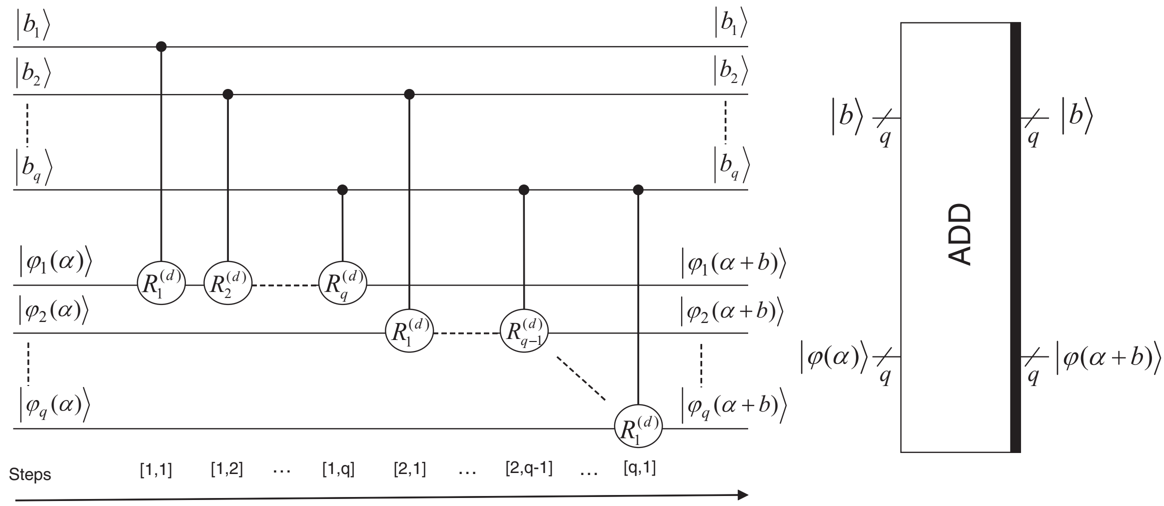

FIG. 2. Adder of two integers (ADD) and the respective symbol. The direct QFT at the lower register which precedes the rotations block as well as the inverse QFT following it are not shown.

previous section, the most significant d-ary digit of an integer is indexed with 1 while the least significant digit is indexed with q. The circuit in Fig. 2 operates on $2q$ qudits, the state $|b_1 \dots b_q\rangle$ of the q upper qudits (upper register) represents integer b while the state of the lower q qudits (lower register) represents the Fourier-transformed state (see Ref. [45]) of the other integer a, that is, $|\varphi_1(a)\rangle|\varphi_2(a)\rangle\cdots|\varphi_q(a)\rangle$, where $|\varphi_l(a)\rangle = \frac{1}{\sqrt{d}}\sum_{m=0}^{d-1} e^{i2\pi(0.a_l a_{l+1}\dots a_q)m}|m\rangle$ [see Eq. (2)]. It is a generalization on qudits of the adder proposed in Ref. [22].

The first qudit of the lower register is initially in the state $|\varphi_1(a)\rangle$. The effect of the first rotation gate $R_1^{(d)}$ controlled by state $|b_1\rangle$ to this qudit (step[1,1]), taking into account Eq. (A16), is to evolve it in the state

$$|\varphi_1(a)\rangle \xrightarrow{R_1^{(d)}} \frac{1}{\sqrt{d}}\sum_{m=0}^{d-1} e^{i2\pi[(0.a_1a_2\dots a_q)+(0.b_1)]m}|m\rangle. \tag{5}$$

The effect of the second gate $R_2^{(d)}$ controlled by $|b_2\rangle$ (step[1,2]) is to further evolve the state to

$$|\varphi_1(a)\rangle_1 \xrightarrow{R_2^{(d)}} \frac{1}{\sqrt{d}}\sum_{m=0}^{d-1} e^{i2\pi[(0.a_1a_2\dots a_q)+(0.b_1)+(0.0b_2)]m}|m\rangle \tag{6}$$

Proceeding in a similar way up to gate $R_q^{(d)}$ controlled by $|b_q\rangle$, we derive the final state (step[1,q]) of the first qudit

$$|\varphi_1(a)\rangle_{q-1} \xrightarrow{R_q^{(d)}} \frac{1}{\sqrt{d}}\sum_{m=0}^{d-1} e^{i2\pi[(0.a_1a_2\dots a_q)+(0.b_1b_2\dots b_q)]m}|m\rangle. \tag{7}$$

In general, the final state of the lth qudit of the lower register is found to be

$$|\varphi_l(a)\rangle_{q-l+1} = \frac{1}{\sqrt{d}}\sum_{m=0}^{d-1} e^{i2\pi[(0.a_la_{l+1}\dots a_q)+(0.b_lb_{l+1}\dots b_q)]m}|m\rangle. \tag{8}$$

Consequently, the total effect of all the rotation gates to the lower register is the evolution

$$|\varphi_1(a)\rangle|\varphi_2(a)\rangle\cdots|\varphi_q(a)\rangle \xrightarrow{\text{ADD}} |\varphi(a+b)\rangle$$
$$= \bigotimes_{l=1}^{q} \frac{1}{\sqrt{d}}\sum_{m=0}^{d-1} e^{i2\pi[(0.a_la_{l+1}\dots a_q)+(0.b_lb_{l+1}\dots b_q)]m}|m\rangle, \tag{9}$$

which is the QFT of the sum state $|a+b \pmod{d^q}\rangle$. Thus, by applying the inverse QFT at the lower register, we can get the desired sum back in the computational basis, while the upper register remains in the initial state $|b\rangle$. The required direct and inverse QFT blocks are not shown in Fig. 2.

B. Adder with constant

Whenever one of the integers is constant, e.g., $b = (b_1b_2\dots b_q)$, the upper register in Fig. 2 is not necessary as all the controlled rotation gates become classically controlled single-qudit rotation gates with their angles defined by the constant integer b. In such a case, a sequence of $q-l+1$ rotation gates $(\Phi_k^{(d)})^{b_{k+l-1}} = \sum_{m=0}^{d-1} e^{i\frac{2\pi}{d^k}mb_{k+l-1}}|m\rangle\langle m|$, for $k = 1\dots q-l+1$ are applied on the lth qudit of the lower register [see Eqs. (A12)–(A14)]. This product of gates can be merged into one gate of the form

$$\begin{aligned} B_l(b) &= \prod_{k=1}^{q-l+1} \left(\Phi_k^{(d)}\right)^{b_{k+l-1}} \\ &= \prod_{k=1}^{q-l+1}\left(\sum_{m=0}^{d-1} e^{\frac{i2\pi m}{d^k}}|m\rangle\langle m|\right)^{b_{k+l-1}} \\ &= \sum_{m=0}^{d-1}\left(\prod_{k=1}^{q-l+1} e^{\frac{i2\pi m}{d^k}b_{k+l-1}}\right)|m\rangle\langle m|. \end{aligned} \tag{10}$$

These are diagonal gates of the form of Eq. (A8), whose angles depend on the constant b by the relation

$$\varphi_{l,m}(b) = \sum_{k=1}^{q-l+1} \frac{2\pi}{d^k} m b_{k+l-1}, \tag{11}$$

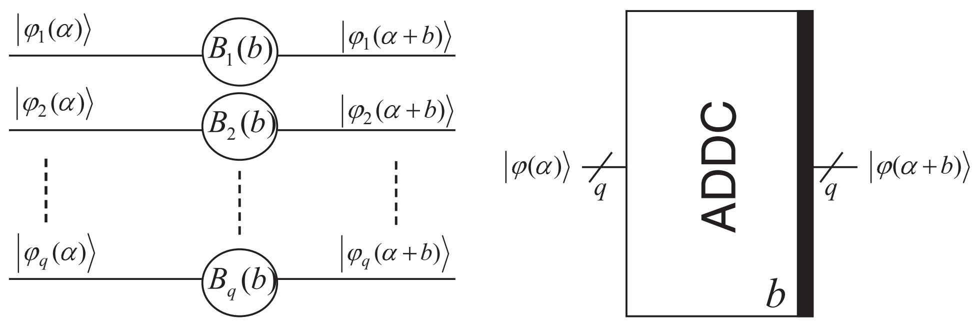

FIG. 3. Adder of an integer with constant b (ADDC_b) and the respective symbol. Direct and inverse QFT blocks not included in this diagram.

so they can be constructed with elementary $R_z^{(jk)}(\theta)$ two-level qudit gates using the procedure described in the Appendix. Figure 3 shows the constant b adder. Like the general adder ADD, this adder performs the addition modulo d^q.

C. Single state controlled adder with constant

The constant adder ADDC_b can be easily converted to a constant adder controlled by the state of an additional control qudit so as to perform the transformation

$$C_c\mathrm{ADDC}_b(|e\rangle|a\rangle) = |e\rangle|a + b\delta_{ce}\rangle, \tag{12}$$

where δ_{ce} is the Kronecker delta function. Consequently, the addition is performed iff the control state equals $|c\rangle$, otherwise the target state $|a\rangle$ remains unaltered. The single-state controlled constant adder $\mathrm{C}_c\mathrm{ADDC}_b$ can be constructed as shown in Fig. 4 where the single-qudit rotation gates $B_l(b)$ of Fig. 3 have been converted to the respective two-qudit diagonal gates controlled by state $|c\rangle$. Such controlled gates affect the phases of the target qudit iff the control qudit has the state $|c\rangle$, thus they are represented by a block diagonal matrix whose cth block is given by Eq. (10) and the other blocks are identities. They have the form of the $\mathrm{CD}_{(c)}$ gates of Eq. (A11) in the Appendix, where decomposition in elementary two-level qudit gates is shown.

D. Generalized controlled adder of an integer with constant

A useful generalization of the previous $\mathrm{C}_c\mathrm{ADDC}_b$ circuit can be achieved if we permit all the basis states of the control qudit to influence the result of the addition. We define this generalized controlled adder with constant b by the relation

$$\mathrm{GCADDC}_b(|e\rangle|a\rangle) = |e\rangle|a + be\rangle. \tag{13}$$

The above equation can be rewritten as

$$\mathrm{GCADDC}_b(|e\rangle|a\rangle) = |e\rangle|a + b\delta_{1e} + \cdots + (d-1)b\delta_{(d-1)e}\rangle. \tag{14}$$

Equation (14) directly leads to the implementation of Fig. 5, where $d-1$ consecutive applications of $\mathrm{C}_c\mathrm{ADDC}_{cb}$ ($c = 1\dots d-1$) adders are employed.

V. MULTIPLIERS

The generalized controlled constant adder adder can be used as building block for the multiplier with constant-accumulator and the multiplier with constant presented in this section. Both kind of multipliers have linear depth with respect to the size of integers processed.

A. Multiplier with constant and accumulator

MAC_b multiplies a q qudits integer x with a constant b of q d-ary digits, and accumulates the product bx to a q qudits integer a (modulo d^q). Namely, the MAC_b circuit consists of two q qudit registers initially holding the states $|x\rangle$ and $|a\rangle$ and transforms them as

$$\mathrm{MAC}_b(|x\rangle|a\rangle) = |x\rangle|a + bx\rangle. \tag{15}$$

Taking into account that x can be expressed as $(x_1x_2\dots x_q) = \sum_{l=1}^{q} x_l d^{q-l}$, then Eq. (15) can be

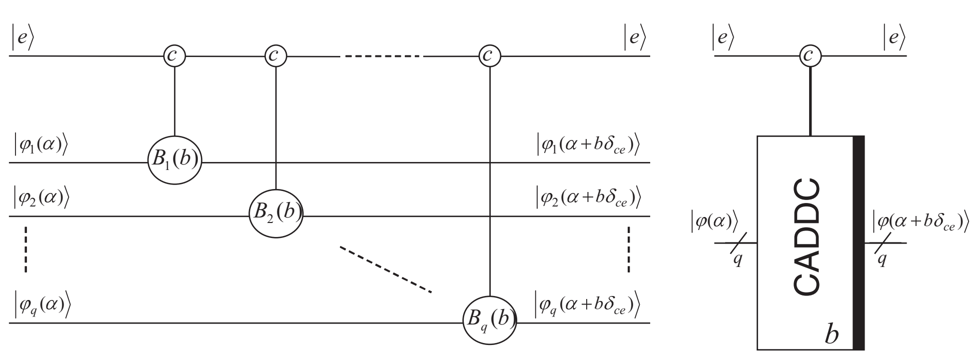

FIG. 4. State $|c\rangle$ controlled adder with constant b ($\mathrm{C}_c\mathrm{ADDC}_b$) and the respective symbol.

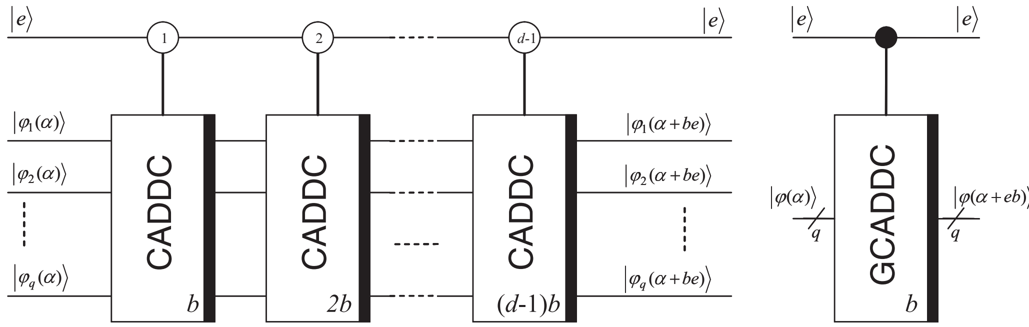

FIG. 5. Generalized controlled adder with constant b (GCADDC_b) and the respective symbol.

written as

$$MAC_b(|x\rangle|a\rangle) = |x\rangle\left|a + b\sum_{l=1}^{q} x_l d^{q-l}\right\rangle$$
$$= |x\rangle|a + x_q b + x_{q-1}db + \cdots + x_1(d^{q-1}b)\rangle. \quad (16)$$

This means that the above transformation can be implemented by applying q GCADDC circuits, where the control is done consecutively by the qudits $x_q, x_{q-1}, \ldots, x_1$ and the constant parameter for each one GCADDC block is $b, db, \ldots, d^{q-1}b$ (modulo d^q), respectively, as shown in Fig. 6.

B. Multiplier with constant

A multiplier (modulo d^q) with constant b implements the function $f : \{0 \ldots d^q - 1\} \rightarrow \{0 \ldots d^q - 1\}$ with $y = f(x) = bx \pmod{q^q}$. When constant b is relative prime to d^q, then there exists the inverse $b^{-1} \pmod{d^q}$ and, consequently, there exists the inverse function $f^{-1}(y) = b^{-1}y \pmod{d^q} = b^{-1}bx \pmod{d^q} = x$. Figure 7 shows how to construct a Multiplier with constant b using two MAC_b blocks and the necessary direct and inverse QFT blocks. It requires a q qudits register initially holding the integer x and another q qubits ancilla register initially in the zero state. The result is that one of the registers is set to the state $|bx \pmod{d^q}\rangle$ while the ancilla register is set to state zero, so effectively the ancilla register can be reused.

In the diagram of Fig. 7, the boxes with the black strip on their right side are the direct blocks while these with the black strip at their left side are the respective inverses. The operation of an inverse MAC with parameter b^{-1} is to perform subtraction instead of accumulation, that is, referring to Fig. 7, we have the operation $\mathrm{MAC}_{b^{-1}}^{-1}|bx\rangle|\varphi(x)\rangle = |bx\rangle|\varphi(x - b^{-1}(bx))\rangle = |bx\rangle|\varphi(0)\rangle$. The inverse MAC^{-1} has the same internal topology as the direct MAC of Fig. 6 (of course, with parameter b^{-1} instead of b) with the only difference that the angles of its rotation gates have a minus sign. The labels at the qudit buses of Fig. 7 which describe the respective states show that the circuit implements the multiplication

$$\mathrm{MULC}_b(|x\rangle|0\rangle) = |bx\rangle|0\rangle. \quad (17)$$

Excluding the ancilla register, which is in the zero state before and after the operation and thus remains unentangled, we conclude that this circuit performs the desired multiplication operation.

VI. COMPLEXITY ANALYSIS

The arithmetic quantum circuits proposed in the previous sections are broken down to the level of elementary gates $H^{(d)}$, as well as elementary $\mathrm{GCX}_m^{(jk)}$, $R_z^{(jk)}(\theta)$, and $R_x^{(jk)}(\theta)$ gates introduced in the Appendix. This decomposition is depicted

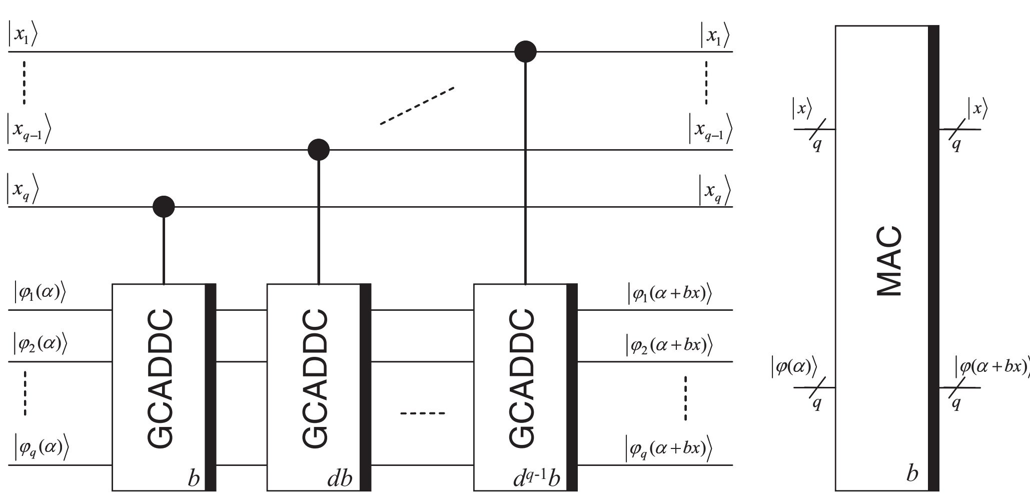

FIG. 6. Multiplier with constant b accumulator (MAC_b) and the respective symbol.

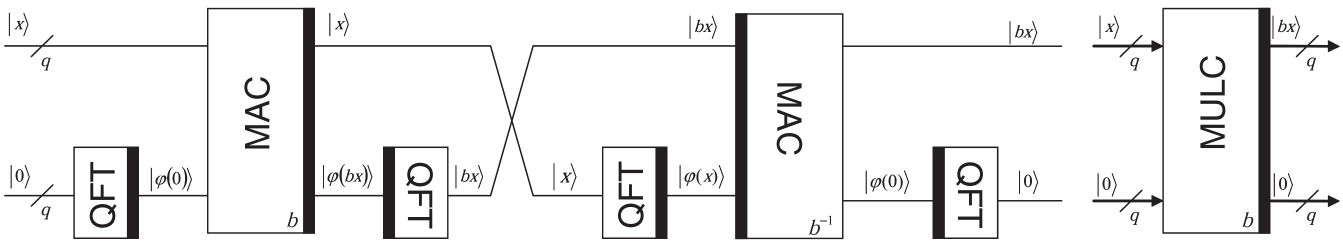

FIG. 7. Multiplier with constant b (MULC_b) and the respective symbol.

in Fig. 8 as a tree structure, where the root of each tree is any of the complete circuits proposed and the leaves of each tree (trapezoids) represent the elementary gates. The edges of each tree are labeled with the number of components needed by one level below (no label stands for 1).

A rough complexity analysis in terms of quantum cost (number of elementary gates used) and depth (execution time) can be done with the help of Fig. 8. The analysis assumes that single- and two-qudit gates are equivalent in terms of costs and execution time. Exact costs and depths depend on the particular implementations. The total gate counts for each block can be found by traversing the tree emerging from the inspected block down to each leaf of the subtree. The labels of the edges for each path are multiplied and then the products of each path used are summed together, e.g. the QFT circuit needs q Hadamard gates, $(q^2 - q/2)(d-1)(2d-1)$ $\mathrm{GCX}_m^{(jk)}$gates, and $(q^2 - q/2)(d-1)2d$ $R_z^{(jk)}(\theta)$ gates. Similar calculations provide us with the quantum costs shown in Table I, which shows only the highest order terms.

For the depth calculation, the following remarks apply:

QFT At first glance, Fig. 1 exhibits a quadratic depth $O(q^2)$, but it can be easily shown [46] that we can parallelize the execution with an appropriate reordering of the gates and thus achieve a linear depth, namely, $depth(\mathrm{QFT}) = 8d^2q$.

ADD Similarly as in the QFT case, a reordering of gates in Fig. 2 offers a linear depth too, that is, $depth(\mathrm{ADD}) = 4d^2q$.

MAC Concurrent execution of gates is possible in this case, too. It can be seen that by flattening the hierarchy MAC-GCADDC-CADDC, q different controlled gates $B_l(b)$ [Eq. (10)] belonging in different GCADDC blocks can be executed concurrently [24]. Thus, the MAC depth is of the order $O(4d^2q)$ instead of $O(4d^2q^2)$ as directly calculated by the number of elementary gates.

MULC Observing Fig. 7, we easily calculate that $depth(\mathrm{MULC}) = 3depth(\mathrm{QFT}) + 2depth(\mathrm{MAC})$, as the two middle QFT blocks (direct and inverse) can be executed simultaneously. Thus, we derive $depth(\mathrm{MULC}) = 32d^2q$.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we presented an assortment of quantum circuits for multilevel qudits. They perform basic integer arithmetic operations (like addition, multiplication-accumulation and multiplication). Additional extensions can be applied, e.g., the MAC and MULC circuits can be converted to single-qudit controlled versions. Such controlled versions could be useful for qudit-based quantum phase estimation algorithms and quantum simulations.

The designs are based on the alternative representation of an integer after being QFT-transformed instead of the usual computational basis representation, a method which has already been exploited in the binary qubit case. QFT-based design is a versatile method to develop many arithmetic

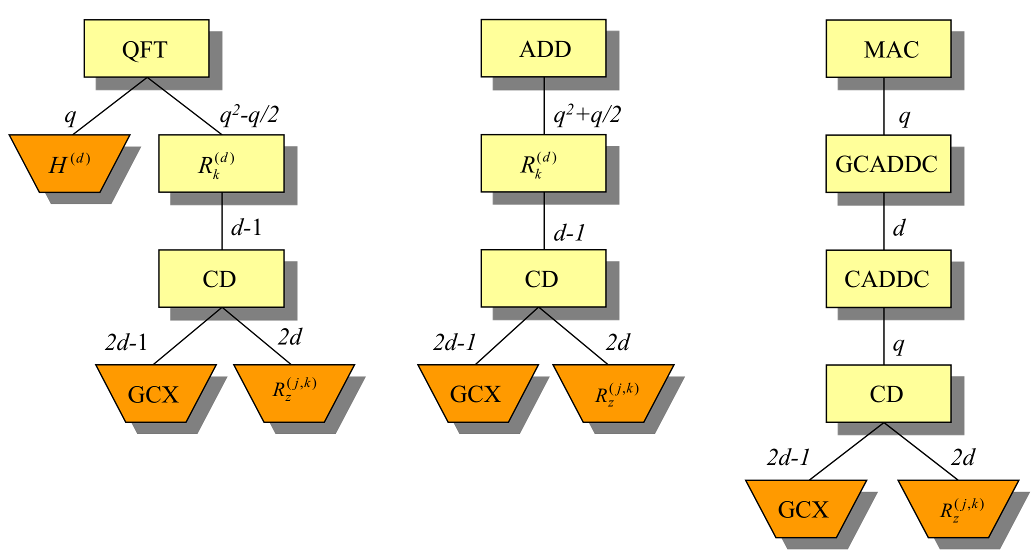

FIG. 8. Hierarchy breakdown of various arithmetic quantum circuits proposed. Elementary two-level gates are denoted in trapezoids as the leaves of the tree. Parameter q is the integer argument size, while d is the dimensionality of the qudits.

TABLE I. Quantum cost, depth and width of the proposed arithmetic circuits.

Circuit	Cost	Depth	Width
QFT	$4d^2q^2$	$8d^2q$	q
ADD	$4d^2q^2$	$4d^2q$	$2q$
MAC	$4d^2q^2$	$4d^2q$	$2q$
MULC	$24d^2q^2$	$32d^2q$	$2q$

circuits, e.g., there is no need to handle carries, which leads to space reduction. Moreover, if it is suitably used, it can offer advantages in terms of speed. This is possible when similar blocks are iterated to act on a data path whose state follows the QFT representation. The extensive usage of rotation gates (which mutually commute) on such a data path permits their rearrangement to execute concurrently [24]. This capability is observed in the MAC block, where the application of a suitable reordering of gates led to depth reduction from $O(q^2)$ to $O(q)$.

Another advantage that has been observed in designs adopting the QFT method is their robustness to various kinds of deviations from the ideal operation, e.g. approximate QFT [47] or QFT banding is the design procedure of eliminating small angle rotation gates. Studies of Shor's algorithm which uses the QFT showed that the algorithm still works sufficiently even when a large proportion of the QFT rotation gates are eliminated [25,48,49]. Recent studies extended to circuits beyond QFT. In Refs. [50,51] the simultaneous gate pruning of rotation gates of the QFT circuit and the QFT based modular exponentiator of Beauregard's circuit [23] were simulated. The simulation results showed similar robustness of Shor's algorithm to these gates eliminations. This robustness is sustained even if the parameters of the remaining rotation gates are randomly selected [26]. The above results suggest that a similar robustness is expected in the multidimensional qudits case and further investigation needs to be carried.

On the other hand, there is a drawback related to the requirement of implementing high accuracy small angle rotation gates. Moreover, these gates must belong to a set of fault-tolerant gates if large-scale quantum computation is considered. The fault-tolerant universal set of gates which is usually adopted in the qubit case is the Clifford+T. Efficient techniques to synthesize (approximately or exactly) an arbitrary quantum gate to a sequence of gates belonging to this fault-tolerant set have been established. The first Solovay-Kitaev algorithms [52–54] generate a sequence of such gates of length $O[\log^{3.97}(1/\epsilon)]$ and synthesis time in order of $O[\log^{2.71}(1/\epsilon)]$, where ϵ is the approximation error. In the last decade, extensive research resulted in great improvements both in terms of the sequence length and synthesis time. They used a diverse set of techniques (usage of ancilla or not, different libraries, approximate or exact synthesis, etc). Some of the best results in terms of the generated sequence length can be found in Refs. [55–58]. These works offer a length of less than $10\log_2(1/\epsilon)$ T gates (T gates are considered more costly if they are built fault tolerantly).

While universal fault-tolerant gates set analogous to qubit Clifford+T have been established for qudits of prime d dimensions [9,59], unfortunately the best synthesis method to synthesize an *arbitrary* qudit gate like the rotation gates is the original Solovay-Kitaev algorithm [54], although optimization techniques for synthesizing qutrit gates belonging to Clifford $+T$ group have appeared in Refs. [60,61] and in Ref. [62] (topological model). Thus, an overhead is expected in the proposed circuits if fault tolerance is entered in the analysis but the exact overhead cannot be easily estimated because the suspected robustness of QFT-based circuits significantly influences the chosen approximation error ϵ.

For all the above reasons and also because the exact cost depends on the technology used, which for qudits is at an early stage, the complexity analysis of Sec. VI is to be considered as a crude indicator of performance. Despite that, we think that the proposed designs enrich the toolkit of future quantum computing.

APPENDIX: GATE DECOMPOSITION

Decomposition of basic gates, like the rotation gates $R_k^{(d)}$, used in the presented circuits to more elementary gates which adopt physical implementation is given. At the lowest level, the elementary gates operate in a two-dimensional subspace of the d-dimensional qudit space. By combining such elementary gates, we can derive more complex gate operating in the whole d-dimensional space. The elementary gates used here are reported in Ref. [10], although alternative library of elementary gates could be used as well.

1. Generalized X gates

The $X^{(jk)}$ gates [10] operate on a two-dimensional subspace of a d-level qudit by exchanging the basis states $|j\rangle$ and $|k\rangle$, while leaving intact the other basis states, thus they are a generalization of the well-known X gate for qubits which exchange the basis states $|0\rangle$ and $|1\rangle$. They are defined by the $d\times d$ matrix:

$$X^{(jk)} = |j\rangle\langle k| + |k\rangle\langle j| + \sum_{\substack{n=0\\ n\neq j\\ n\neq k}}^{d-1} |n\rangle\langle n|, \qquad j,k = 0\ldots d-1. \tag{A1}$$

It holds that $X^{(jk)} = X^{(kj)}$, so there are $d(d-1)/2$ different such gates in this family.

2. Rotation gates of two levels

These gates perform a rotation on a two-dimensional subspace [10] of a d-level qudit and are defined as

$$R_a^{(jk)}(\theta) = exp\big(-i\theta\sigma_a^{(jk)}/2\big), \quad 0 \leqslant j,k \leqslant d-1, \tag{A2}$$

where $\sigma_x^{(jk)} = |j\rangle\langle k| + |k\rangle\langle j|$, $\sigma_y^{(jk)} = -i|j\rangle\langle k| + i|k\rangle\langle j|$ and $\sigma_z^{(jk)} = |j\rangle\langle j| - |k\rangle\langle k|$ for $j,k = 0\ldots d-1$ are matrices of dimensions $d\times d$. Parameter θ is the rotation angle and label $a\in\{x,y,z\}$,

3. Generalized controlled X gates

The $\mathrm{GCX}_{(m)}^{(jk)}$ gates are generalizations in the qudits of the CNOT gates acting on qubits [10]. Thus, they are gates which operate on a control and a target qudit. A GCX gate has three

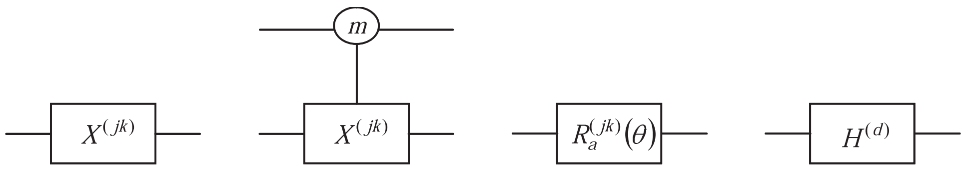

FIG. 9. Symbols of $X^{(jk)}$, $\mathrm{GCX}_{(m)}^{(jk)}$, $R_a^{jk}(\theta)$, and $H^{(d)}$ elementary gates (a is x, y or z).

parameters, m, j, and k, which define its operation. A $\mathrm{GCX}_{(m)}^{(jk)}$ acts like a $X^{(jk)}$ on the target qudit iff the control qudit is on the basis state $|m\rangle$. Consequently, the definition matrix of such a gate is block diagonal with dimension $d^2 \times d^2$ consisting of d blocks of $d \times d$ dimensions each and is given by

$$\mathrm{GCX}_{(m)}^{(jk)} = |m\rangle\langle m| \otimes \left(|j\rangle\langle k| + |k\rangle\langle j| + \sum_{\substack{n=0\\ n\neq j\\ n\neq k}}^{d-1} |n\rangle\langle n| \right) + \sum_{\substack{n=0\\ n\neq m}}^{d-1} |n\rangle\langle n| \otimes I_d, \tag{A3}$$

where $j, k, m = 0 \dots d-1$ and I_d is the identity matrix of dimensions $d \times d$. Equation (A3) can be equivalently written as

$$GCX_m^{(jk)} = \mathrm{diag}(I_d, I_d, \dots, \underbrace{X^{(jk)}}_{m\text{th block}}, \dots, I_d). \tag{A4}$$

4. Hadamard gate

The Hadamard gate $H^{(d)}$ on d-level qudits has already been defined in Eq. (3). It is easy to show that application of a $H^{(d)}$ gate to a basis state $|j\rangle$ derives the superposition

$$\begin{aligned} H^{(d)}|j\rangle &= \frac{1}{\sqrt{d}}\begin{bmatrix}1 & e^{i2\pi(0.j)} & \dots & e^{i2\pi(d-1)(0.j)}\end{bmatrix}^T \\ &= \frac{1}{\sqrt{d}}(|0\rangle + e^{i2\pi(0.j)}|1\rangle + \dots + e^{i2\pi(d-1)(0.j)}|d-1\rangle) \end{aligned} \tag{A5}$$

in d-ary fractional representation. The Hadamard gate for qudits essentially performs the order-d Fourier transform; likewise, the Hadamard gate for qubits performs the order-2 Fourier transform. Methods for implementation of the $H^{(d)}$ gate are proposed in Refs. [2,44].

The symbols used throughout the text for the three families of elementary gates defined and the $H^{(d)}$ gate are shown in Fig. 9.

5. Diagonal gates of one and two qudits

The qudit elementary gates $X^{(jk)}$, $R_a^{(jk)}$ and $\mathrm{GCX}_m^{(jk)}$ affect a two-dimensional subspace of the whole d-dimensional Hilbert space of a single qudit. They can be used to derive single- and two-qudit (controlled) diagonal basic gates affecting the whole d-dimensional space of one of the qudits as follows.

The diagonal $D'(a_1, a_2, \dots, a_{d-1})$ gate [10] is defined by the equation

$$\begin{aligned} &D'(a_1, a_2, \dots, a_{d-1}) \\ &\quad = e^{i\varphi}\mathrm{diag}(e^{-i(a_1+a_2+\dots+a_{d-1})}, e^{ia_1}, e^{ia_2}, \dots, e^{ia_{d-1}}). \end{aligned} \tag{A6}$$

It can easily be proved that such a gate can be constructed by sequentially applying $d-1$ $R_z^{(jk)}(\theta)$ gates as shown in the following equation:

$$\begin{aligned} &D'(a_1, a_2, \dots, a_{d-1}) \\ &\quad = e^{i\varphi}R_z^{(01)}(a_1)R_z^{(02)}(a_2)\cdots R_z^{(0(d-1))}(a_{d-1}). \end{aligned} \tag{A7}$$

A related gate is the $D(\varphi_1, \varphi_2, \dots, \varphi_{d-1})$ defined as

$$D(\varphi_1, \varphi_2, \dots, \varphi_{d-1}) = \mathrm{diag}(1, e^{i\varphi_1}, e^{i\varphi_2}, \dots, e^{i\varphi_{d-1}}). \tag{A8}$$

The $D(\varphi_1, \varphi_2, \dots, \varphi_{d-1})$ gate is identical with the $D'(a_1, a_2, \dots, a_{d-1})$ gate if we set

$$a_j = \varphi_j - \frac{1}{d}\sum_{k=1}^{d-1}\varphi_k, \quad j = 1 \dots d-1 \tag{A9}$$

and add a global phase of angle $\varphi = \frac{1}{d}\sum_{k=1}^{d-1}\varphi_k$ to every diagonal element of $D'(a_1, a_2, \dots, a_{d-1})$.

The previously defined diagonal gates can be extended to operate on two qudits, where the first is the control qudit and the second is the target qudit, in the following manner: A diagonal gate $D'(a_1, a_2, \dots, a_{d-1})$ or $D(\varphi_1, \varphi_2, \dots, \varphi_{d-1})$ is applied on the target qudit iff the control qudit is in state $|m\rangle$, otherwise no operation is effective on the target. Thus, the $d^2 \times d^2$ matrices representing such gates have the following block diagonal form:

$$\begin{aligned} &\mathrm{CD}'_m(a_1, a_2, \dots, a_{d-1}) \\ &\quad = \mathrm{diag}(I_d, \dots, I_d, \underbrace{D'(a_1, a_2, \dots, a_{d-1})}_{m\text{th block}}, I_d, \dots, I_d) \end{aligned} \tag{A10}$$

and

$$\begin{aligned} &\mathrm{CD}_m(\varphi_1, \varphi_2, \dots, \varphi_{d-1}) \\ &\quad = \mathrm{diag}(I_d, \dots, I_d, \underbrace{D(\varphi_1, \varphi_2, \dots, \varphi_{d-1})}_{m\text{th block}}, I_d, \dots, I_d) \end{aligned} \tag{A11}$$

A construction of a $\mathrm{CD}'_m(a_1, a_2, \dots, a_{d-1})$ gate using $4(d-1)$ elementary $\mathrm{GCX}_{(m)}^{(jk)}$ and $R_z^{(jk)}(\theta)$ gates is shown in Fig. 10. Single qudit gate $S_m = \mathrm{diag}(1, \dots, 1, \underbrace{e^{i\varphi}}_{m\text{th pos}}, 1, \dots, 1)$ is a phase gate which is identical to a D' gate up to a global phase.

6. Generalized controlled rotation gate $R_k^{(d)}$

The controlled diagonal gates CD'_m and CD_m are activated whenever the control state is equal to one of the d possible

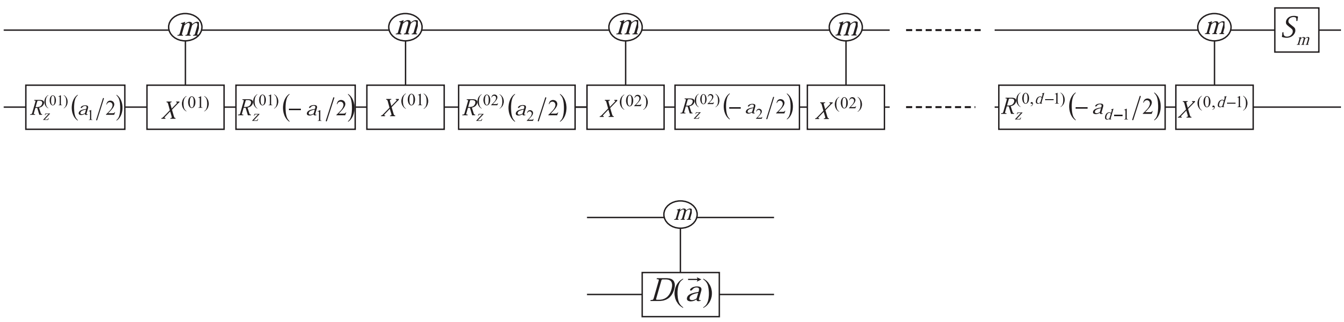

FIG. 10. Controlled diagonal $\mathrm{CD}'_m(a_1, a_2, \ldots, a_{d-1})$ gate construction and its symbol. The parameter $\vec{a}$ inside the symbol represents the angles $(a_1, a_2, \ldots, a_{d-1})$.

basis states, e.g., $|m\rangle$. They can be used to synthesize the generalized controlled diagonal gate, $R_k^{(d)}$, of Eq. (4) such that each one of the d possible control states have a different effect on the target qudit.

The $R_k^{(d)}$ gate is parametrized by the integer k. The matrix defining this gate is block diagonal of the form

$$R_k^{(d)} = \mathrm{diag}\Big(\big(\Phi_k^{(d)}\big)^0, \big(\Phi_k^{(d)}\big)^1, \ldots, \big(\Phi_k^{(d)}\big)^{d-1}\Big), \qquad \text{(A12)}$$

where the matrix $\Phi_k^{(d)}$ is diagonal too, and defined with

$$\Phi_k^{(d)} = \mathrm{diag}(1, e^{i\varphi_1}, e^{i\varphi_2}, \ldots, e^{i\varphi_{(d-1)}}). \qquad \text{(A13)}$$

The angles $\varphi_1, \varphi_2, \ldots, \varphi_{(d-1)}$ depend on the parameter k as follows:

$$\varphi_m = \frac{2\pi}{d^k} m, \quad m = 1, \ldots, d-1. \qquad \text{(A14)}$$

The implementation of an $R_k^{(d)}$ can be achieved by sequentially combining $d-1$ controlled diagonal gates $\mathrm{CD}_m(\varphi_1, \varphi_2, \ldots, \varphi_{d-1})$ for $m = 1 \ldots d-1$ and different angles for each case of m as shown below [see also Eqs. (A11)–(A13)]:

$$\begin{aligned} R_k^{(d)} = {} & \mathrm{CD}_{(1)}\left(\frac{2\pi}{d^k}, \frac{2\pi}{d^k}2, \ldots, \frac{2\pi}{d^k}(d-1)\right)\mathrm{CD}_{(2)}\left(\frac{2\pi}{d^k}2, \frac{2\pi}{d^k}4, \ldots, \frac{2\pi}{d^k}2(d-1)\right) \\ & \times \mathrm{CD}_{(d-1)}\left(\frac{2\pi}{d^k}(d-1), \frac{2\pi}{d^k}(d-1)2, \ldots, \frac{2\pi}{d^k}(d-1)(d-1)\right). \end{aligned} \qquad \text{(A15)}$$

Taking into account that a $\mathrm{CD}_{(m)}(\varphi_1, \varphi_2, \ldots, \varphi_{d-1})$ gate is composed by $4(d-1)$ elementary $\mathrm{GCX}_{(m)}^{(jk)}$ and $R_z^{(jk)}(\theta)$ gates, we conclude that an $R_k^{(d)}$ gate requires $4(d-1)^2$ elementary gates.

We can see by inspecting Eq. (4) that an $R_k^{(d)}$ gate is a generalization on qudits of the controlled rotation gates $R_k = R_z(2\pi/2^k) = \mathrm{diag}(1, 1, 1, e^{i2\pi/2^k})$ for the qubit case (where $d = 2$) and this generalization is exploited in the construction of the QFT and the various arithmetic circuits based on the QFT. To understand this, it is useful to see what the effect is of an $R_k^{(d)}$ gate when the control qudit is on a basis state $|j_1\rangle$ $(j_1 = 0, 1, \ldots, d-1)$ and the target qudit is in a superposition of equal amplitudes but with different phases, such as $|b\rangle = \frac{1}{\sqrt{d}}\sum_{l=0}^{d-1} e^{i\varphi_l}|l\rangle$. The joint state of the two qudits after the application of an $R_k^{(d)}$ gate is

$$\begin{aligned} R_k^{(d)}(|j_1\rangle|b\rangle) &= \frac{1}{\sqrt{d}}\sum_{j=0}^{d-1}\sum_{m=0}^{d-1} e^{i2\pi(0.\underbrace{00\ldots0}_{k-1}j)m}|j\rangle\underbrace{\langle j|j_1\rangle}_{=\delta_{jj_1}} \otimes \underbrace{|m\rangle\langle m|\sum_{l=0}^{d-1}e^{i\varphi_l}|l\rangle}_{=e^{i\varphi_m}|m\rangle} \\ &= \frac{1}{\sqrt{d}}\sum_{m=0}^{d-1} e^{i2\pi(0.\underbrace{00\ldots0}_{k-1}j_1)m}|j_1\rangle e^{i\varphi_m}|m\rangle = \frac{1}{\sqrt{d}}|j_1\rangle\sum_{m=0}^{d-1} e^{i2\pi\left[(0.\underbrace{00\ldots0}_{k-1}j_1)m+\varphi_m\right]}|m\rangle. \end{aligned} \qquad \text{(A16)}$$

Thus, an angle $2\pi(0.\underbrace{00\ldots0}_{k-1}j_1)m = \frac{2\pi}{d^k}j_1 m$ is added to every component $|m\rangle$ of the target qudit superposition and this angle is proportional to the value $|j_1\rangle$ of the control qudit and also proportional to index m in the $|m\rangle$ component of target qudit superposition.

[1] A. Muthukrishnan and C. R.Stroud, Multivalued logic gates for quantum computation, Phys. Rev. A **62**, 052309 (2000).

[2] A. Muthukrishnan and C. R. Stroud, Quantum fast Fourier transform using multilevel atoms, J. Mod. Opt. **49**, 2115 (2002).

[3] B. P. Lanyon, M. Barbieri, M. P. Almeida, T. Jennewein, T. C. Ralph, K. J. Resch, G. J. Pryde, J. L. O'Brien, A. Gilchrist, and A. G. White, Simplifying quantum logic using higher-dimensional Hilbert spaces, Nat. Phys. **5**, 134 (2008).

[4] P. Gokhale, J. M. Baker, C. Duckering, N. C. Brown, K. R. Brown, and F. T. Chong, Asymptotic improvements to quantum circuits via qutrits, in *Proceedings of the 46th International ACM Symposium on Computer Architecture, ISCA '19* (ACM, Phoenix, AZ, USA, 2019), pp. 554–566.

[5] D. Gottesman, Fault tolerant quantum computation with higher dimensional systems, 1st NASA Conference on Quantum Computing and Quantum Communications Palm Springs, California, February 17-20, 1998, Chaos Solitons Fractals **10**, 1749 (1999).

[6] A. Ketkar, A. Klappenecker, S. Kumar, and P. K. Sarvepalli, Nonbinary stabilizer codes over finite fields, IEEE Trans. Inf. Theor. **52**, 4892 (2006).

[7] E. T. Campbell, Enhanced Fault-Tolerant Quantum Computing in d-Level Systems, Phys. Rev. Lett. **113**, 230501 (2014).

[8] A. Krishna and J.-P. Tillich, Towards Low Overhead Magic State Distillation, Phys. Rev. Lett. **123**, 070507 (2019).

[9] E. T. Campbell, H. Anwar, and D. E. Browne, Magic-State Distillation in All Prime Dimensions Using Quantum Reed-Muller Codes, Phys. Rev. X **2**, 041021 (2012).

[10] Y.-M. Di and H.-R. Wei, Synthesis of multivalued quantum logic circuits by elementary gates, Phys. Rev. A **87**, 012325 (2013).

[11] M. Neeley, M. Ansmann, R. C. Bialczak, M. Hofheinz, E. Lucero, A. D. O'Connell, D. Sank, H. Wang, J. Wenner, A. N. Cleland, M. R. Geller, and J. M. Martinis, Emulation of a quantum spin with a superconducting phase qudit, Science **325**, 722 (2009).

[12] F. W. Strauch, Quantum logic gates for superconducting resonator qudits, Phys. Rev. A **84**, 052313 (2011).

[13] A. Babazadeh, M. Erhard, F. Wang, M. Malik, R. Nouroozi, M. Krenn, and A. Zeilinger, High-Dimensional Single-Photon Quantum Gates: Concepts and Experiments, Phys. Rev. Lett. **119**, 180510 (2017).

[14] M. Malik, M. Erhard, M. Huber, M. Krenn, R. Fickler, and A. Zeilinger, Multi-photon entanglement in high dimensions, Nat. Photonics **10**, 248 (2016).

[15] P. W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, in *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science, (FOCS '94)* (IEEE, Santa Fe, NM, USA, 1994), pp. 124–134.

[16] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, Simulated quantum computation of molecular energies, Science **309**, 1704 (2005).

[17] V. Parasa and M. Perkowski, Quantum phase estimation using multivalued logic, in *Proceedings of the 41st IEEE International Symposium on Multiple-Valued Logic (ISMVL '11)* (IEEE, Tuusula, Finland, 2011), pp. 224–229.

[18] Y. Fan, A generalization of the deutsch-jozsa algorithm to multi-valued quantum logic, in *Proceedings of the 37th IEEE International Symposium on Multiple-Valued Logic (ISMVL '07)* (IEEE, Oslo, Norway, 2007), pp. 12–12.

[19] E. Kiktenko, A. Fedorov, A. Strakhov, and V. Man'ko, Single qudit realization of the Deutsch algorithm using superconducting many-level quantum circuits, Phys. Lett. A **379**, 1409 (2015).

[20] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC '96)* (ACM, Philadelphia, PA, USA, 1996), pp. 212–219.

[21] S. S. Ivanov, H. S. Tonchev, and N. V. Vitanov, Time-efficient implementation of quantum search with qudits, Phys. Rev. A **85**, 062321 (2012).

[22] T. G. Draper, Addition on a quantum computer, arXiv:quant-ph/0008033.

[23] S. Beauregard, Circuit for Shor's algorithm using $2n + 3$ qubits, Quantum Inf. Comput. **3**, 175 (2003).

[24] A. Pavlidis and D. Gizopoulos, Fast quantum modular exponentiation architecture for Shor's factoring algorithm, Quantum Inf. Comput. **14**, 649 (2014).

[25] A. Barenco, A. Ekert, K.-A. Suominen, and P. Törmä, Approximate quantum Fourier transform and decoherence, Phys. Rev. A **54**, 139 (1996).

[26] Y. S. Nam and R. Blümel, Analytical formulas for the performance scaling of quantum processors with a large number of defective gates, Phys. Rev. A **92**, 042301 (2015).

[27] J. L. Brylinski and R. Brylinski, Universal quantum gates, in *Mathematics of Quantum Computation* (Chapman & Hall CRC Press, Boca, 2002).

[28] F. S. Khan and M. Perkowski, Synthesis of multi-qudit hybrid and d-valued quantum logic circuits by decomposition, Theor. Comput. Sci. **367**, 336 (2006).

[29] G. K. Brennen, D. P. O'Leary, and S. S. Bullock, Criteria for exact qudit universality, Phys. Rev. A **71**, 052318 (2005).

[30] S. S. Bullock, D. P. O'Leary, and G. K. Brennen, Asymptotically Optimal Quantum Circuits for d-Level Systems, Phys. Rev. Lett. **94**, 230502 (2005).

[31] M. Perkowski and E. Curtis, A transformation based algorithm for ternary reversible logic synthesis using universally controlled ternary gates, in *Proceedings of the 13th IEEE/ACM International Workshop on Logic and Synthesis (IWLS '04)* (IEEE/ACM, Temecula, CA, USA, 2004).

[32] D. M. Miller, G. W. Dueck, and D. Maslov, A synthesis method for MVL reversible logic [multiple value logic], in *Proceedings of the 34th IEEE International Symposium on Multiple-Valued Logic (ISMVL '04)* (IEEE, Toronto, Canada, 2004), pp. 74–80.

[33] N. Denler, M. Perkowski, and P. Kerntopf, Synthesis of reversible circuits from a subset of Muthukrishnan-Stroud quantum realizable multi-valued gates, in *Proceedings of the 13th IEEE/ACM International Workshop on Logic and Synthesis (IWLS '04)* (IEEE/ACM, Temecula, CA, USA, 2004).

[34] V. Vedral and A. Barenco and A. Ekert, Quantum networks for elementary arithmetic operations, Phys. Rev. A **54**, 147 (1996).

[35] S. A. Cuccaro, T. G. Draper, S. A. Kutin, and D. Petrie Moulton, A new quantum ripple-carry addition circuit, arXiv:quant-ph/0410184.

[36] R. Van Meter and K. M. Itoh, Fast quantum modular exponentiation, Phys. Rev. A **71**, 052320 (2005).

[37] T. G. Draper, S. A. Kutin, E. M. Rains, and K. M. Svore, A logarithmic-depth quantum carry-lookahead adder, Quantum Inf. Comput. **6**, 351 (2006).

[38] A. Khosropour, H. Aghababa, and B. Forouzandeh, Quantum division circuit based on restoring division algorithm, in *Proceedings of the 8th International Conference on Information Technology: New Generations (ITNG '11)* (IEEE, Las Vegas, NV, USA, 2011), pp. 1037–1040.
[39] B.-S. Choi and R. Van Meter, A $\Theta(\sqrt{n})$-depth quantum adder on the 2D NTC quantum computer architecture, ACM J. Emerging Technol. Comput. Sys. **8**, 24 (2012).
[40] M. H. Khan and M. A. Perkowski, Quantum ternary parallel adder/subtractor with partially-look-ahead carry, J. Sys. Architecture **53**, 453 (2007).
[41] M. H. Khan, Synthesis of quaternary reversible/quantum comparators, J. Sys. Architecture **54**, 977 (2008).
[42] S. N. Takakiko Satoh and R. V. Meter, A reversible ternary adder for quantum computation, in *Asian Conference on Quantum Information Science (AQIS '07)* (Kyoto, Japan, 2007).
[43] A. Bocharov, S. X. Cui, M. Roetteler, and K. M. Svore, Improved quantum ternary arithmetic, Quantum Inf. Comput. **16**, 862 (2016).
[44] A. S. Ermilov and V. E. Zobov, Representation of the quantum Fourier transform on multilevel basic elements by a sequence of selective rotation operators, Opt. Spectrosc. **103**, 969 (2007).
[45] Notice that the order of the lower register qudits is the same as the order appearing in Fig. 1 before the application of the swap gates.
[46] A. G. Fowler, S. J. Devitt, and L. C. L. Hollenberg, Implementation of Shor's algorithm on a linear nearest neighbour qubit array, Quantum Inf. Comput. **4**, 237 (2004).
[47] D. Coppersmith, An approximate Fourier transform useful in quantum factoring, Technical Report No. RC 19642 (IBM Research Division, T. J. Watson Research Center, 1994), arXiv:quant-ph/0201067.
[48] A. G. Fowler and L. C. L. Hollenberg, Scalability of Shor's algorithm with a limited set of rotation gates, Phys. Rev. A **70**, 032329 (2004).
[49] Y. S. Nam and R. Blümel, Performance scaling of Shor's algorithm with a banded quantum Fourier transform, Phys. Rev. A **86**, 044303 (2012).
[50] Y. S. Nam and R. Blümel, Robustness and performance scaling of a quantum computer with respect to a class of static defects, Phys. Rev. A **88**, 062310 (2013).
[51] Y. S. Nam and R. Blümel, Streamlining Shor's algorithm for potential hardware savings, Phys. Rev. A **87**, 060304(R) (2013).
[52] A. Y. Kitaev, Quantum computations: Algorithms and error correction, Russ. Math. Surveys **52**, 1191 (1997).
[53] A. Y. Kitaev, A. H. Shen, and M. N. Vyalyi, *Classical and Quantum Computation* (American Mathematical Society, Providence, RI, 2002).
[54] C. M. Dawson and M. A. Nielsen, The Solovay-Kitaev Algorithm, Quantum Inf. Comput. **6**, 81 (2006).
[55] A. Bocharov, Y. Gurevich, and K. M. Svore, Efficient decomposition of single-qubit gates into V basis circuits, Phys. Rev. A **88**, 012313 (2013).
[56] P. Selinger, Efficient Clifford + T approximation of single-qubit operators, Quantum Inf. Comput. **15**, 159 (2015).
[57] N. J. Ross and P. Selinger, Optimal Ancilla-free Clifford + T approximation of Z-rotations, Quantum Inf. Comput. **16**, 901 (2016).
[58] V. Kliuchnikov, D. Maslov, and M. Mosca, Practical approximation of single-qubit unitaries by single-qubit quantum Clifford and T circuits, IEEE Trans. Comput. **65**, 161 (2016).
[59] M. Howard and J. Vala, Qudit versions of the qubit $\pi/8$ gate, Phys. Rev. A **86**, 022316 (2012).
[60] S. Prakash, A. Jain, B. Kapur, and S. Seth, Normal form for single-qutrit Clifford + T operators and synthesis of single-qutrit gates, Phys. Rev. A **98**, 032304 (2018).
[61] A. N. Glaudell, N. J. Ross, and J. M. Taylor, Canonical forms for single-qutrit Clifford + T operators, Ann. Phys. **406**, 54 (2019).
[62] A. Bocharov, X. Cui, V. Kliuchnikov, and Z. Wang, Efficient topological compilation for a weakly integral anyonic model, Phys. Rev. A **93**, 012313 (2016).