J Comput Electron (2017) 16:442–458
DOI 10.1007/s10825-017-0960-4

Towards designing efficient reversible binary code converters and a dual-rail checker for emerging nanocircuits

Neeraj Kumar Misra[1] · **Bibhash Sen**[2] · **Subodh Wairya**[1]

Published online: 25 February 2017

Abstract Reversible logic has attracted interest from many researchers in the area of quantum information science. Since there is no information loss in reversible logic, energy consumption is greatly reduced. However, realization of quantum equivalent circuits using cascading reversible gates is complex. Predominantly, this work targets implementation of quantum equivalent circuits using cascading reversible gates. In this work, novel code converters and a dual-rail checker with lower cost metrics such as gate count, garbage output, ancilla input, unit delay, logical calculation, and quantum cost are constructed. Several new reversible gates, namely BE (binary excess), BG-2 (binary Gray), GB-2 (Gray binary), and NG-R1 and NG-R2 (N = new, R = reversible), are designed and used to construct efficient code converter and dual-rail checker circuits. The main contribution of these novel circuits is the consideration of the gate-level schematics in the respective quantum equivalent circuit using our proposed algorithm. The performance results establish that the novel binary-coded decimal (BCD)-to-excess-3, binary-to-Gray, and dual-rail checker achieve improvement of 25 and 66.6 % in gate count and 44.4 % in quantum cost, respectively, compared with counterpart designs.

✉ Neeraj Kumar Misra
neeraj.mishra@ietlucknow.ac.in

Bibhash Sen
bibhash.sen@cse.nitdgp.ac.in

Subodh Wairya
swairya@ietlucknow.edu

[1] Department of Electronics Engineering, Institute of Engineering and Technology, Lucknow, India

[2] Department of Computer Science and Engineering, National Institute of Technology, Durgapur, India

Keywords Reversible computation · Quantum computation · Code converter · Dual-rail checker · Quantum circuit · Nanocircuits

1 Introduction

The very large-scale integration (VLSI) industry is currently moving at high speed towards miniaturization, resulting in many challenges in terms of energy dissipation and quantum effects at the nanoscale. These problems are exacerbated when further downscaling the feature size of transistors [1]. To overcome this limitation, rigorous research is being carried out to identify prominent alternatives. One possible solution would be to develop a robust computational paradigm based on quantum technologies [2]. On the other hand, reversible logic is gaining in popularity due to its capability to implement quantum logic circuits [3]. Elementary quantum logic circuits can be designed using reversible logic gates due to their unitary behavior, in turn favoring more effective implementation of quantum computing [4]. Another prominent feature of reversible logic is zero information loss, which helps realization of energy-efficient circuits [4]. Reversible logic is popular for design of digital logic circuits with negligible power consumption, as proved by the famous researchers Landauer and Bennett [5,6]. This concept provides strong motivation to choose reversible logic for digital logic circuit synthesis. The scope of reversible implementations extends to include thermodynamics and adiabatic complementary metal–oxide–semiconductor (CMOS) technology [7]. Thus, application of reversible logic is advocated in various fields, including quantum computing [8], nanoelectromechanical systems (NEMS) [9], low-power CMOS [10], optical computing [11], and nanotechnology [12].

A reversible logic gate provides a bijective mapping between its input and output vectors. Based on this bijective property, when knowing the output states, the input states should be recoverable, and the system shows no loss of information, resulting in reduced heat dissipation [13]. It is believed that reversible computing will become the dominant technology in the near future. Furthermore, increasing the computational speed of quantum circuits also increases their quantum cost, hindering synthesis of reversible logic circuits [14]. Hence, the quantum cost must be minimized by reducing the elemental quantum gates in a reversible circuit, providing strong motivation to synthesize reversible circuits with lower quantum cost. An important step in reversible circuit synthesis is the estimation of other reversible metrics such as the gate count (GC), garbage output (GO), ancilla inputs (CI), logical calculation (LC), and unit delay (UD), which we also calculate in this study. By achieving optimal reversible metrics, the performance of the reversible circuit can be enhanced.

Code converters are very common in electronic systems, being very important in electronic circuits to change information from one format to another. Furthermore, the importance of testing to enhance the reliability of VLSI circuits is well known in the digital world. The dual-rail checker (DRC) is a modest device for such testing. The operation of a DRC is verified when the outputs are complementary to each other.

Several reversible logic circuits have already been presented in literature, including an adder [15,16], multiplier [17], arithmetic logic unit (ALU) [18], and sequential circuits [19]. However, few reversible circuits such as code converters and dual-rail checkers (DRCs) have been presented in literature [20–27]. Due to this lack of compact reversible code converters and DRCs in literature, we targeted these circuits in this work. An attempt was made to transform a cascaded gate-level circuit into a respective quantum equivalent (QE) circuit using our proposed algorithm, being especially useful to help beginners in the area of quantum computing to build quantum circuits from cascaded gate-level schematics. Elemental quantum gates (EQGs) such as CNOT, C-V, C-V$^+$, and NOT are adopted for the design of the QE circuits [28]. The quantum cost (QC) was successfully obtained using the RCviewer+ tool with some additional coding. The proposed code converters and DRC were very effective in terms of the reported cost metrics, namely gate count (GC), garbage output (GO), constant input (CI), unit delay (UD), logical calculation (LC), and quantum cost (QC). The quantum cost of our code converter and dual-rail checker circuits was calculated using the standard NCV library. The optimal quantum cost value of our proposed circuits was better than those of existing counterpart designs. Furthermore, use of the proposed gates strongly changes the cost metric values of the proposed circuits. The novelty of this work lies in the design of high-level blocks such as code converters and a DRC using quantum equivalent circuits in the quantum computing paradigm. The four major pillars of this work can be summarized as follows:

- We demonstrate synthesis of reversible circuits such as BCD-to-excess-3, BCD-to-Gray, binary-to-Gray, Gray-to-binary, and dual-rail checker in a quantum computing framework. In addition, we construct the quantum equivalent of these gate-level schematics using elemental quantum gates with the standard NCV library.
- We discuss elemental quantum gate-based qubit transitions in the quantum circuit of our three proposed reversible gates.
- We demonstrate and illustrate an algorithm for converting a cascaded gate-level schematic into the respective quantum equivalent circuit, which could be applied using criteria other than the quantum cost.
- We use several novel reversible gates to achieve circuit models for code converter and dual-rail checker circuits with low cost metrics, achieving optimal results compared with their best counterpart designs.

1.1 Organization and acronyms

The rest of this paper is organized as follows: Section 1.1 presents acronyms and a short explanation. Section 2 recalls the principles of reversible logic and quantum computing. Section 3 presents preliminary work on synthesis of code converters and a dual-rail checker, including details of the parameters and synthesis approach. Section 4 proposes synthesized circuits based on three reversible gates with low quantum cost. Section 4.5 includes a description of the elementary quantum gate-based qubit transitions in the quantum circuits. Section 5 presents the proposed circuits, gate level schematics, and quantum equivalent circuits. Section 5.1 contains a discussion of the proposed algorithm for transformation of a gate-level schematic into the corresponding quantum equivalent circuit. Section 6 presents our cost metric results as well as results for existing circuits identified based on a review of the state of the art. Section 7 summarizes the work, with conclusions and future prospects for research.

This work contains many acronyms, which are identified in Table 1 for better understanding of the paper. Quantum reversible acronyms have been surveyed in [1–4,14,28,29].

2 Principles of reversible logic and quantum computing

We list below the basic principles that are essential to understand reversible circuits. The principles of reversible logic and quantum computing have been surveyed in [1–3,15,16,28,29].

Table 1 Acronyms with short explanation

Acronym	Denotes	Short explanation
CI	Constant input	Inputs that are not needed in the structure of a reversible circuit
CNOT	Controlled-NOT (Feynman gate)	If the control qubit = 1, then the target qubit is inverted; if the control qubit = 0, then the target qubit is non-inverted. The CNOT has a QC of 1
C-V	Square-root-of-NOT	Property: V × V = Inversion. The C-V has a QC of 1
C-V^+	Hermitian of NOT gate\Inverse of C-V gate	Property: $V^+ \times V^+$ = Inversion. The C-V^+ has a QC of 1
DRC	Dual-rail checker	The two outputs are complementary to each other
EQG	Elemental quantum gate	Gates such as CNOT, NOT, C-V^+, and C-V
GC	Gate count	The total number of gates in a reversible circuit
GO or g	Garbage/unwanted/auxiliary output	Outputs that are not needed in the structure of a reversible circuit
I\O	Input\output	Input and output pin
IQG	Integrated qubit gate	One NOT or CNOT with the C-V and C-V^+ gate contributing identical input and output lines in series
LC	Logical calculation	A number of gates such as Ex-OR, NOT, and AND used to construct a logical function
MCT	Multiple-control Toffoli gates	A reversible circuit which utilizes the MCT library with optimization rules in an integrated library
NCT	NOT, CNOT, and Toffoli	A reversible circuit which utilizes the NCT library with optimization rules in an integrated library
NCV	NOT, Controlled-NOT, and square-root-of-NOT	A reversible circuit which utilizes the NCV library with optimization rules in an integrated library
NOT	The most widely used EQG	A single qubit is inverted
QC	Quantum cost	The number of EQGs needed to synthesize a quantum circuit
QE	Quantum equivalent of reversible circuit	Information regarding quantum primitive gates such as NOT, CNOT, C-V, and C-V^+
Qubit	Qubit	A quantum bit
T2\T3	Two-input Toffoli\three-input Toffoli	Part of *.tfc code
v\i\o	Variable\input\output	Part of *.tfc code

2.1 Quantum computing and basic elemental quantum gates

Quantum computing In quantum computing, a bit is specified by a qubit. A quantum circuit is realized using EQGs. In fact, a quantum circuit is reversible and works with qubits. The state of a qubit can be explicitly stated as $|\Psi >= \alpha|0> +\beta|1>$, where $|1>$ and $|0>$, denote logic states 1 and 0, respectively. The complex numbers α and β have the property $|\alpha|^2 + |\beta|^2 = 1$.

Elemental quantum gate EQGs are the basic elements for constructing a quantum circuit. The quantum cost is a measure of the EQG count in a quantum circuit. In quantum computing, the state of a particle is depicted by a qubit. The computations using a qubit in a quantum circuit are just matrix computations. Elemental quantum gates can compute with matrices. The popular elemental quantum gates are controlled-V (C-V), controlled-V^+ (C-V^+), NOT, and CNOT (FG). A more interesting feature of controlled gates (V and V^+) arises if one uses V and V^+ on the same line, i.e., $V \times V^+ = I$. This tells us that, if one uses V and V^+ on the same line, one forms a quantum wire with zero quantum cost. A short discussion of all four EQGs is presented below:

- *Controlled-V gate* This gate implements the square-root-of-NOT, its basic feature being that the square of V is equivalent to an inversion.
- *Controlled*-V^+ *gate* This gate is the inverse of the controlled-V, known as the Hermitian of the NOT gate. Its basic feature is that the square of V^+ is equivalent to an inversion.
- *CNOT* The basic feature of this gate is that, if the control qubit = 1, then the target qubit is inverted, whereas if the control qubit = 0, then the target qubit is non-inverted. The CNOT is also known as the Feynman gate (FG).
- *NOT* A single qubit is inverted.

2.2 Reversible logic

In a reversible gate, the defined Boolean function maps each input vector to a unique output vector. A reversible gate has the following characteristics: (1) there are equal numbers of

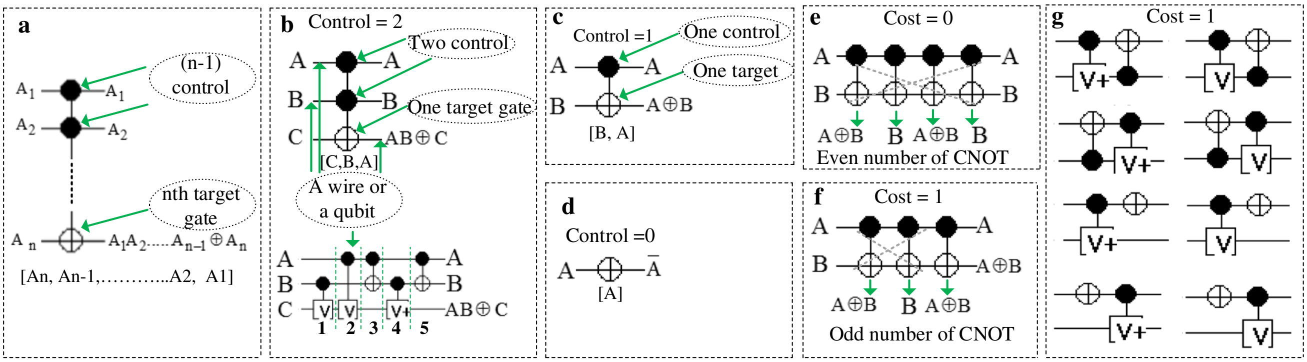


Fig. 1 Reversible circuits: **a** MCT, **b** CCNOT gate and QE, **c** CNOT gate, **d** NOT gate, **e** even number of CNOT, **f** odd number of CNOT, **g** IQG

input and output lines, (2) various operations (loops, feedback, and fanout) are not allowed, and (3) each input vector maps to a unique output vector.

Definition 1 An n-input n-output Toffoli gate has the first $n - 1$ inputs equal to the control and the remaining input as a target (Fig. 1a). The target gate output is an Ex-OR with the control. A gate with two controls and one target gate is known as a CNOT gate (Fig. 1c). A gate with no control gate is known as an invert gate (Fig. 1d).

Example 1 A three-input three-output Toffoli gate (CCNOT) reversibly executes the operation $(a, b, c) \rightarrow (a, b, a \oplus b)$, where a and b are the controls, the notation ⊕ indicates the Ex-OR operation, and c is the target. If A = B= 1, the state of the target C is inverted (Fig. 1b).

Definition 2 A common technique used to reduce the QC of a quantum circuit is as follows: If there are an even number of CNOT gates with identical I/O stage, the QC is zero (In Fig. 1e). Similarly, for an odd number of CNOT gates with identical I/O stage, first consider the even number of CNOT gates with zero QC, then find the effective QC (Fig. 1f).

Definition 3 Another elemental gate that is helpful for reducing the QC is the IQG. Figure 1g shows the general arrangement of an IQG.

Definition 4 Quantum gate libraries such as MCT, NCV, NCT, and EQ are utilized for quantum circuit design and optimization. In the NCV library, four elements are included, viz. C-V, C-V$^+$, CNOT, and NOT. The MCT library includes NOT instead of CNOT, whereas the NCT includes NOT, CNOT, and square-root-of-NOT gates. In the EQ library, the basic elemental quantum gates are defined.

The performance of a reversible circuit is often estimated using cost metrics such as the gate count (GC), garbage output (GO), ancilla input (CI), logical calculation (LC), and unit delay (UD). To enhance circuit performance, these cost metrics must be minimized. A short discussion of the important cost metrics is presented below:

- *Quantum cost* In the quantum computing paradigm, the quantum cost (QC) is the sum of the EQGs.
- *Garbage/unused or unwanted output* The garbage output (GO) specifies the unused outputs.
- *Constant/ancilla input* Constant input (CI) is not needed in the structure of a reversible function in the quantum computing paradigm.
- *Logical calculation/hardware complexity* The total number of gates, e.g., AND, NOT, and Ex-OR, utilized to synthesize the logical function. In general, a logical calculation (LC) is written according to the following equation: LC = number of gates (Ex-OR + AND + NOT) of the logical function = number of gates $(\alpha + \beta + \delta)$ of the logical function, where α is the two-input Ex-OR, β is the two-input AND, and δ is the one-input NOT.
- *Unit delay/critical path* The unit delay (UD) is defined as the maximum number of reversible gates in a path from the input line to any output line.

2.3 Reversible circuit mapping library

The standard NCV library is very popular in current research [28]. It is complete; i.e., it enables a quantum logic circuit to be built for any reversible function. The NCV library comprises the NOT, CNOT, C-V, and C-V$^+$ gates. The goal of the NCV library is to synthesize a quantum circuit with set outputs and inputs [29]. To achieve this, only three elemental quantum gates, i.e., NOT, C-V, and C-V$^+$, are used in the quantum circuit. In this library, binding and optimization models are applied to construct the quantum circuit. Such structures are well known in quantum computing modeling. Most of the quantum circuit construction described herein is based on this library. The basic properties V × V = V$^+$ × V$^+$ = Inversion and V × V$^+$ = I are applied to propagate the qubit (Fig. 2a). In the case of a Toffoli gate (CCNOT), the synthesis flow using the NCV library is shown in Fig. 2b.

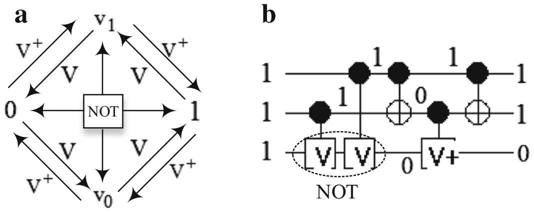


Fig. 2 Basics of NCV library: **a** state transition of a qubit, **b** TG quantum circuit with qubit transition using NCV library

3 Related work

Previous syntheses of reversible code converters and dual-rail checker circuits assumed that the quantum cost depended only on the reversible gates used in the design. Existing designs, such as those proposed in [20–27], were therefore constructed from gate-level schematics, considering only the individual gates as the criteria for calculation of the quantum cost. However, when using a quantum computation model, the whole quantum equivalent circuit model must be used to determine the quantum cost. However, integration of circuits based on cascaded gates into the respective quantum circuit is an intricate process. In [20], the authors proposed a variety of reversible code converter circuits. The cost metrics of the BCD-to-excess-3 code converter were GC of 4, CI of 6, and QC of 24. The construction of a reversible dual-rail checker was described in [25], where new reversible gates were used for circuit construction. The circuit implementation had GC of 6, GO of 14, CI of 12, and QC of 54. However, none of these existing designs establish a framework for transformation of a gate-level schematic into a respective quantum equivalent circuit.

By a cost-efficient circuit, we mean the circuit implementation showing the *best synthesis*, i.e., with optimal metrics. The criteria determining these two concepts (best synthesis and optimal metrics) depend on the gate used and the algorithm applied to obtain the quantum equivalent circuit. This work proposes three reversible gates for construction of code converters and a dual-rail checker, along with a few existing reversible gates. We also demonstrate how elemental quantum gates can process the qubit transition in a reversible circuit and thereby produce the outputs. We used the NCV library to construct the quantum circuits. Important metrics are also evaluated for the synthesized reversible circuits, including GO, UD, LC, QC, and CI. The proposed approach for circuit synthesis leads to reduced cost metrics compared with counterpart designs.

4 Proposed reversible gates

For the construction and optimization of reversible circuits, we propose several reversible gates. The QE circuit for a single reversible gate is first illustrated in Sect. 4.1, for readability.

4.1 Illustration of the transformation from a single reversible gate-level circuit to a respective quantum equivalent circuit

In this subsection, the quantum circuit for a single reversible gate is illustrated. In the first step, we create a *.pla file for the reversible gate, based solely on its truth table. During the second step, a Toffoli block (in *.tfc code format) is created from the *.pla file. The third step includes decomposition of the Toffoli block into EQGs (in *.real code format) using the RCviewer+ tool. The EQGs are then used to construct the quantum equivalent circuit. This methodology was used in this work to construct a quantum equivalent circuit for a single reversible gate.

4.2 Reversible BE gate

A new 4 × 4 schematic of the BE gate is presented in Fig. 3a. In the quantum equivalent of the BE gate, six CNOT, two C-V, and one C-V$^+$ gate are utilized to implement the logical function. Therefore, the QC is 9 (counting the primitive gates). The truth table of the BE gate is presented in Fig. 3b. The QE and BE.tfc code are presented in Fig. 3c. The BE.tfc code is then used to construct the QE circuit. The BE.tfc code is also utilized to evaluate the QC, as shown by the snapshot in Fig. 3d.

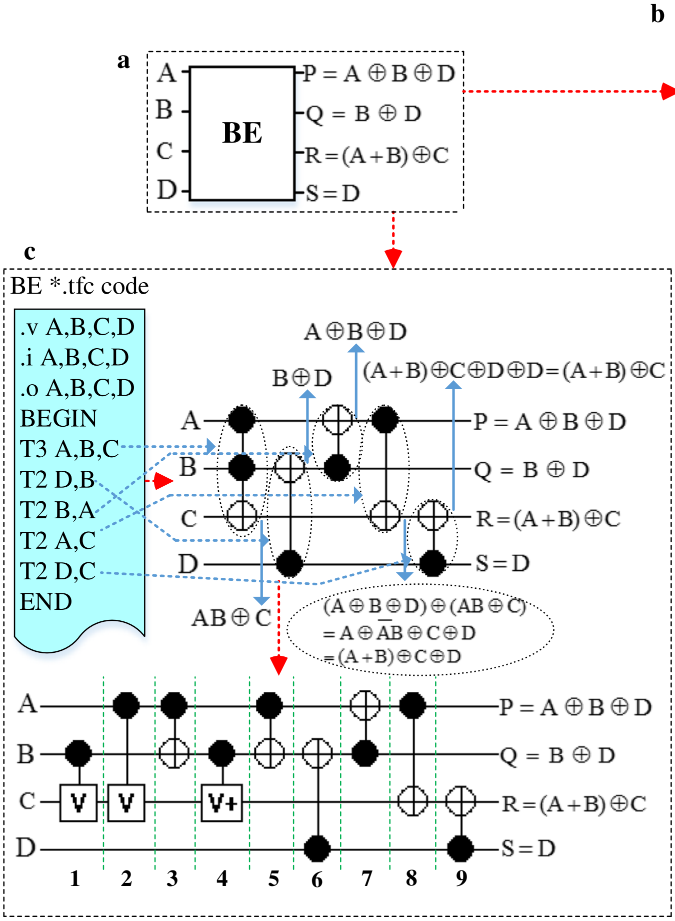
The synthesis expression was constructed from Fig. 3a. When setting the third and fourth input of the BE gate to low and high, respectively, the outputs are generated as $P = A \oplus B \oplus 1 = \overline{A \oplus B}$, $Q = \overline{B} \oplus 1 = B$, and $S = (A + B) \oplus 0 = A + B$. For the aims of this study, we minimized the gate count in the BCD-to-excess-3 code converter, utilizing one BE, one PG, and one NG-R2 to construct the BCD-to-excess-3 converter (Sect. 5.2).

The LC of the BE gate was calculated as follows: The expression for S has LC = 0. The expression for Q $(B \oplus D)$ has LC $= 1\alpha$. Then, we use signal duplication to obtain $A \oplus B$ for the expression for P. The final LC value can then be computed as: LC (BE) $= 1\alpha(P) + 1\alpha$ (for the signal duplication operation in $P) + 1\alpha(Q) + 1\alpha(R) + 0\alpha(S) = 4\alpha$.

4.3 Reversible NG-R1 gate

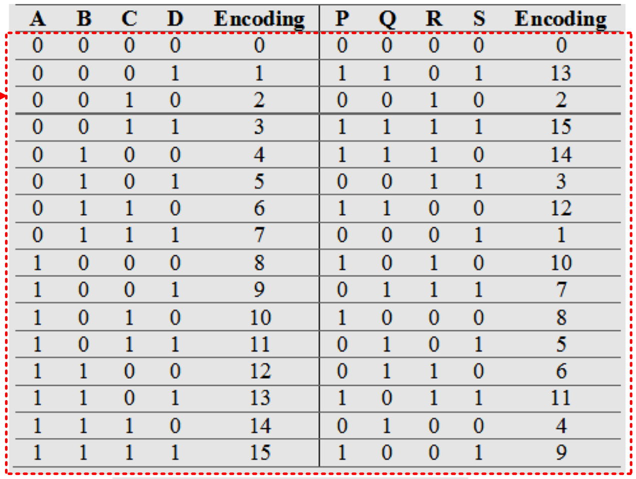
The schematic diagram and truth table of the novel NG-R1 are shown in Fig. 4a, b. The QE and NG-R1.tfc code are shown in Fig. 3c. The QC and two-qubit gate result are both 10, as shown in Fig. 4d. The LC for the NG-R1 was calculated as $2\alpha + 2\beta$.

The synthesis expression was constructed from Fig. 4a. When setting the fourth and fifth input of the NG-R1 gate to



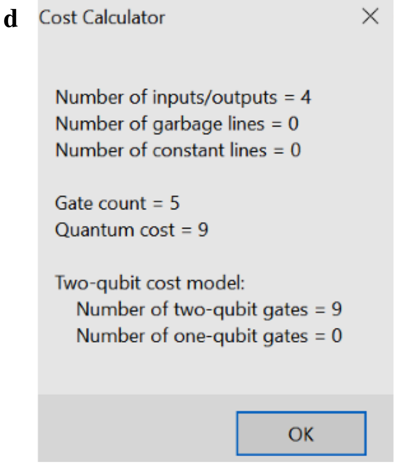
```
.v A,B,C,D
.i A,B,C,D
.o A,B,C,D
BEGIN
T3 A,B,C
T2 D,B
T2 B,A
T2 A,C
T2 D,C
END
```

b



A	B	C	D	Encoding	P	Q	R	S	Encoding
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	1	13
0	0	1	0	2	0	0	1	0	2
0	0	1	1	3	1	1	1	1	15
0	1	0	0	4	1	1	1	0	14
0	1	0	1	5	0	0	1	1	3
0	1	1	0	6	1	1	0	0	12
0	1	1	1	7	0	0	0	1	1
1	0	0	0	8	1	0	1	0	10
1	0	0	1	9	0	1	1	1	7
1	0	1	0	10	1	0	0	0	8
1	0	1	1	11	0	1	0	1	5
1	1	0	0	12	0	1	1	0	6
1	1	0	1	13	1	0	1	1	11
1	1	1	0	14	0	1	0	0	4
1	1	1	1	15	1	0	0	1	9

d



Number of inputs/outputs = 4
Number of garbage lines = 0
Number of constant lines = 0

Gate count = 5
Quantum cost = 9

Two-qubit cost model:
Number of two-qubit gates = 9
Number of one-qubit gates = 0

Fig. 3 Reversible BE gate: **a** schematic diagram, **b** truth table, **c** QE and BE.tfc code, and **d** snapshot of results

low, the required outputs of the NG-R1 gate are then $P = A$, $Q = B$, $R = C$, $S = A \cdot B \oplus 0 = A \cdot B$, and $T = A \cdot C \oplus 0 = A \cdot C$. For the aims of this study, we minimized the garbage output in the dual-rail checker circuit. In this work, we utilized two NG-R1 gates and two NG-R2 gates to construct the dual-rail checker circuit (Sect. 5.7).

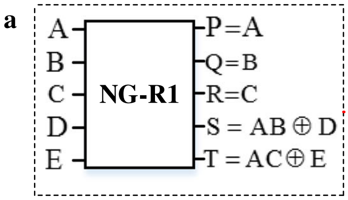
4.4 Reversible NG-R2 gate

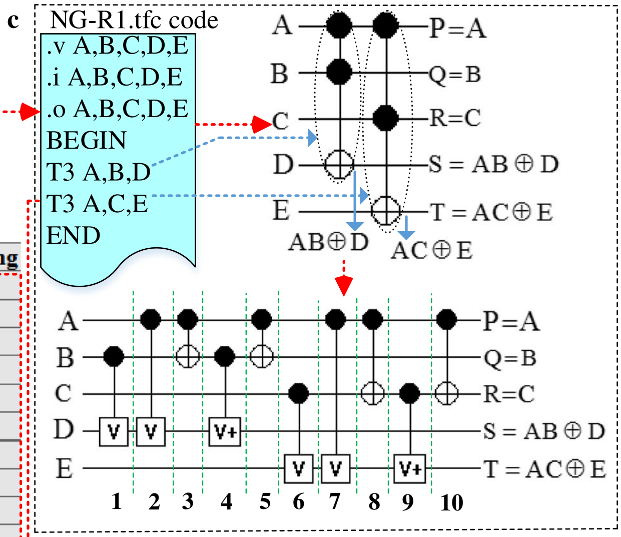
The NG-R2 structure has five inputs and five outputs. Its schematic diagram and truth table are shown in Fig. 5a, b. The QE and NG-R2.tfc code are shown in Fig. 5c. Five two-qubit gates are considered to be present in the QE circuit, with QC of 5; we evaluated the LC as 2α. To check the QC of the NG-R2, the result is presented in Fig. 5d.

The synthesis expression was constructed from Fig. 5a, when setting the third input of the NG-R2 gate to be low. The required outputs of the NG-R2 are obtained as $P = A$, $Q = A \oplus B$, and $R = (A + B) \oplus 0 = A + B$. For the aims of this study, we minimized the gate counts and garbage outputs in the code converters and dual-rail checker circuit (Sect. 5.7).

4.5 EQG-based synthesis flow with proposed reversible gates

The synthesis flow was investigated based on the outputs obtained with set inputs using the NCV library [29]. To achieve this, only four EQGs were used in the quantum circuit, such structures being well known in quantum computing. The basic properties $\mathrm{V} \times \mathrm{V} = \mathrm{V}^{+} \times \mathrm{V}^{+} =$ Inversion and $\mathrm{V} \times \mathrm{V}^{+} = \mathrm{I}$ were used to propagate the qubit in the quantum circuit [30]. The inputs were all on the left side of the QE, with the corresponding outputs on the right side, and the quantum gate outputs in the middle. In the case of the BE synthesis flow, the input “1111” was taken for computing using the heuristic: a square of V and square of V^{+} gives inversion, whereas if V and V^{+} are combined, the identity results. These basic properties yield the output shown in Fig. 6a. The synthesis flow for the other proposed gates, such as NG-R1 and NG-R2, were obtained using the same procedure, as depicted in Fig. 6b, c. The target output was verified by studying the truth table of these gates (Figs. 3b, 4b, 5b). This test was conducted using only a few qubit

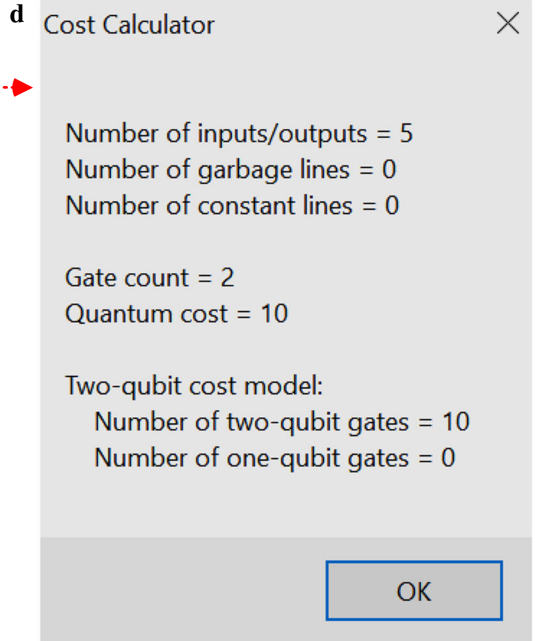




```
NG-R1.tfc code
.v A,B,C,D,E
.i A,B,C,D,E
.o A,B,C,D,E
BEGIN
T3 A,B,D
T3 A,C,E
END
```

b

A	B	C	D	E	Encoding	P	Q	R	S	T	Encoding
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0	1	1
0	0	0	1	0	2	0	0	0	1	0	2
0	0	0	1	1	3	0	0	0	1	1	3
0	0	1	0	0	4	0	0	1	0	0	4
0	0	1	0	1	5	0	0	1	0	1	5
0	0	1	1	0	6	0	0	1	1	0	6
0	0	1	1	1	7	0	0	1	1	1	7
0	1	0	0	0	8	0	1	0	0	0	8
0	1	0	0	1	9	0	1	0	0	1	9
0	1	0	1	0	10	0	1	0	1	0	10
0	1	0	1	1	11	0	1	0	1	1	11
0	1	1	0	0	12	0	1	1	0	0	12
0	1	1	0	1	13	0	1	1	0	1	13
0	1	1	1	0	14	0	1	1	1	0	14
0	1	1	1	1	15	0	1	1	1	1	15
1	0	0	0	0	16	1	0	0	0	0	16
1	0	0	0	1	17	1	0	0	0	1	17
1	0	0	1	0	18	1	0	0	1	0	18
1	0	0	1	1	19	1	0	0	1	1	19
1	0	1	0	0	20	1	0	1	0	1	21
1	0	1	0	1	21	1	0	1	0	0	20
1	0	1	1	0	22	1	0	1	1	1	23
1	0	1	1	1	23	1	0	1	1	0	22
1	1	0	0	0	24	1	1	0	1	0	26
1	1	0	0	1	25	1	1	0	1	1	27
1	1	0	1	0	26	1	1	0	0	0	24
1	1	0	1	1	27	1	1	0	0	1	25
1	1	1	0	0	28	1	1	1	1	1	31
1	1	1	0	1	29	1	1	1	1	0	30
1	1	1	1	0	30	1	1	1	0	1	29
1	1	1	1	1	31	1	1	1	0	0	28



Number of inputs/outputs = 5
Number of garbage lines = 0
Number of constant lines = 0

Gate count = 2
Quantum cost = 10

Two-qubit cost model:
Number of two-qubit gates = 10
Number of one-qubit gates = 0

Fig. 4 Reversible NG-R1 gate: **a** schematic diagram, **b** truth table, **c** QE and NG-R1.tfc code, and **d** snapshot of results

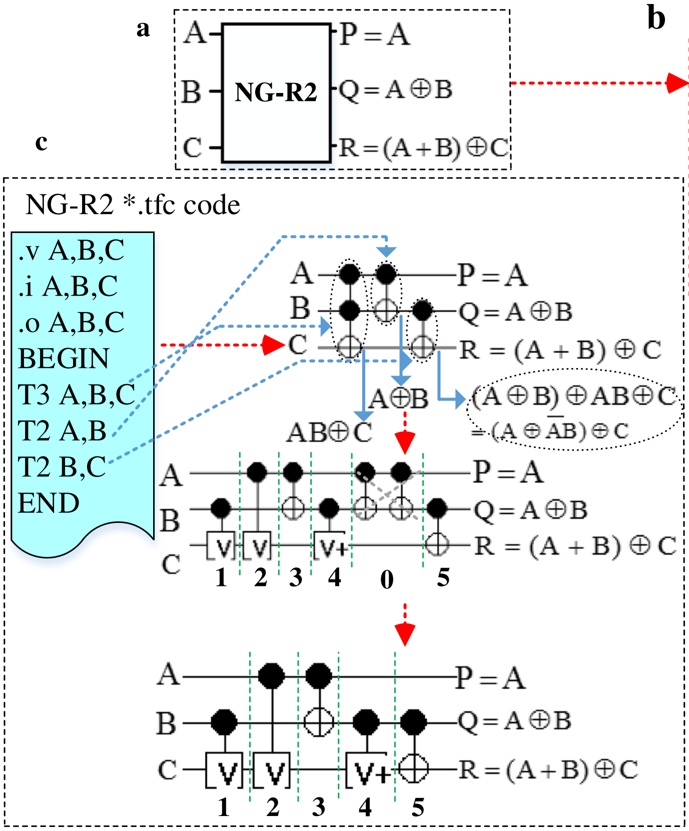
states; a similar method could be applied for any other qubit states.

5 Proposed circuit models of reversible code converters and a dual-rail checker

In the field of computation, a circuit that converts information from one format to another can be regarded as a code converter. The most popular types of code converter are BCD-to-excess-3, BCD-to-Gray, binary-to-Gray, and Gray-to-binary. The operation of the BCD-to-excess-3 converter requires a BCD number (0 to 9). The conversion is done by adding the decimal number 3 to each BCD number. The working process of the BCD-to-Gray converter requires a four-bit BCD number (0 to 9). With increasing demands for circuit testing, the DRC has emerged as a new popular circuit. The QE circuit for the cascaded gate is illustrated in Sect. 5.1, in advance for readability.

5.1 Illustration of transforming cascaded reversible gate-based circuits into respective quantum equivalent circuits

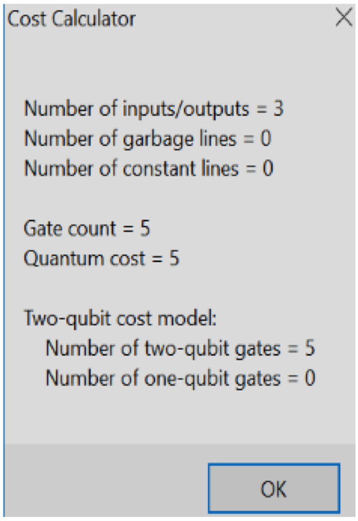
In this subsection, we present a technique for converting a cascaded reversible gate circuit into a respective QE cir-



b

A	B	C	Encoding	P	Q	R	Encoding
0	0	0	0	0	0	0	0
0	0	1	1	0	0	1	1
0	1	0	2	0	1	1	3
0	1	1	3	0	1	0	2
1	0	0	4	1	1	1	7
1	0	1	5	1	1	0	6
1	1	0	6	1	0	1	5
1	1	1	7	1	0	0	4

d



Number of inputs/outputs = 3
Number of garbage lines = 0
Number of constant lines = 0

Gate count = 5
Quantum cost = 5

Two-qubit cost model:
Number of two-qubit gates = 5
Number of one-qubit gates = 0

Fig. 5 Reversible NG-R2 gate: **a** schematic diagram, **b** truth table, **c** QE and NG-R2.tfc code, and **d** snapshot of results

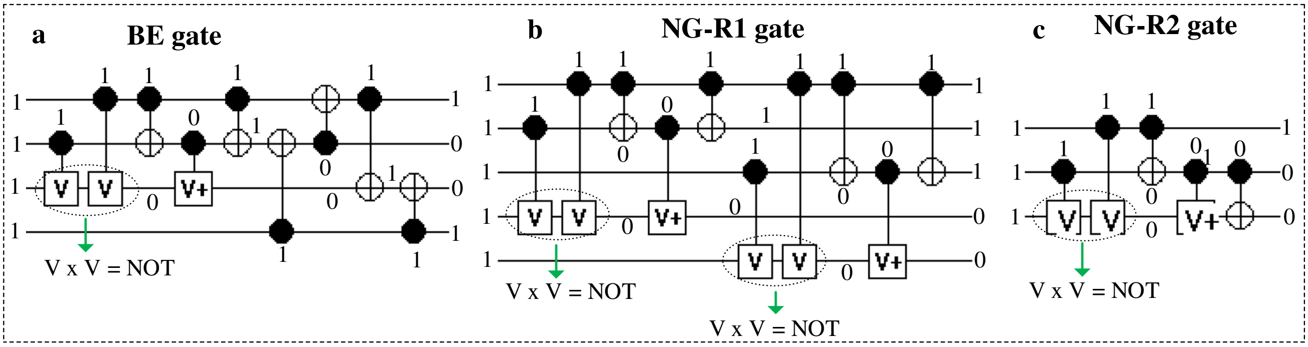


Fig. 6 EQG-based synthesis flow in **a** BE, **b** NG-R1, and **c** NG-R2

cuit based on EQGs. To the best of the authors' knowledge, such QE circuits for code converter and DRC circuits are not discussed in literature. The steps to convert a cascaded reversible gate into the respective QE circuit is presented in Algorithm 1.

Algorithm 1: Algorithm to convert cascaded reversible gates into the respective QE circuit.

Step 1 Construct *.tfc code for the reversible gates involved in the cascaded gate-level circuit.

Step 2 Assign input labels in alphabetical order to all the gates. The assigned alphabetical order just indicates the input pins. All the output pins also have the same alphabetical order that was assigned to the input pins. Then, check that the assigned input alphabets are the same as the output alphabets.

Step 3 The *.tfc code is modified after combining each gate in the cascaded gate-level circuit.

Step 4 The final combined.tfc code for the cascaded gate-level circuit is obtained using the RCviewer+ tool, ready for decomposition, optimization, and QC calculation.

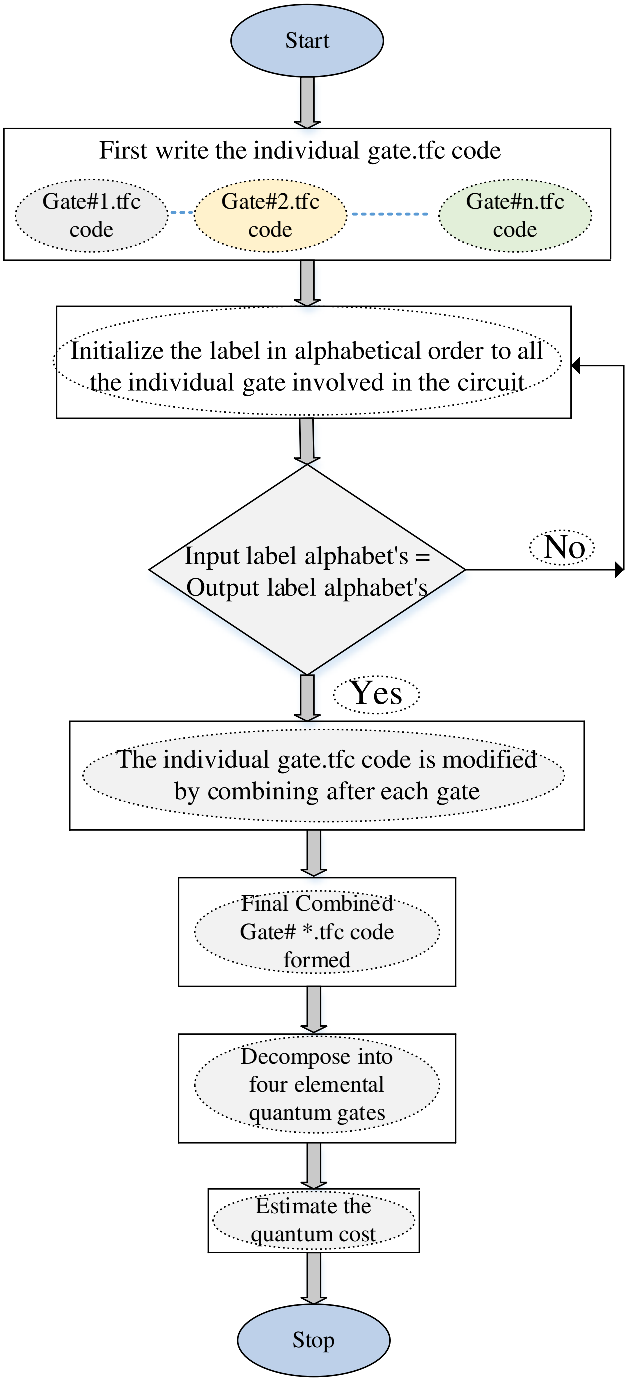

Start
First write the individual gate.tfc code
Gate#1.tfc code
Gate#2.tfc code
Gate#n.tfc code
Initialize the label in alphabetical order to all the individual gate involved in the circuit
Input label alphabet's = Output label alphabet's
No
Yes
The individual gate.tfc code is modified by combining after each gate
Final Combined Gate# *.tfc code formed
Decompose into four elemental quantum gates
Estimate the quantum cost
Stop

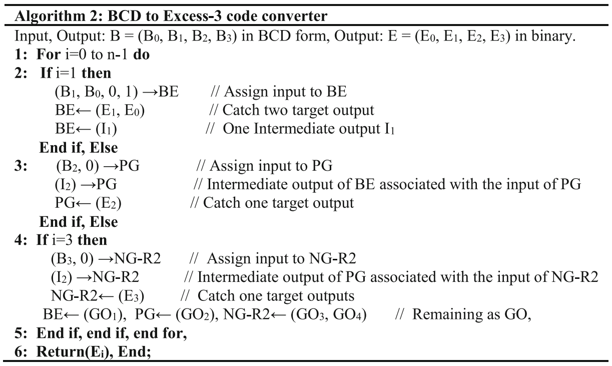
Fig. 7 Flowchart for conversion of a cascaded gate-level circuit to a respective QE

The flow chart for constructing a quantum circuit is depicted in Fig. 7. The QE circuits proposed herein were constructed using this methodology.

5.2 The proposed reversible BCD-to-excess-3 code converter

A four-bit BCD-to-excess-3 circuit was synthesized using BE, PG, and NG-R2 gates. A schematic diagram of the proposed circuit is shown in Fig. 8a. We describe the construction of this circuit in Algorithm 2. The intermediate bits such as I_1 and I_2 are determined by the input bits B_i ($i = 0$–3). The result is achieved after a particular connection performed on the circuit. The construction of the Toffoli gate block and BCD-to-excess 3.tfc code is shown in Fig. 8b. The QE circuit was synthesized using the BCD-to-excess-3.tfc code by decomposition into EQGs as shown in Fig. 8c. The QC and two-qubit gates are presented in Fig. 8d.

Proposition 1 *The LC of the BCD-to-excess-3 converter is* $LC(\textit{BCD-to-excess-3}) = 4\alpha(BE) + 2\alpha + 1\beta(PG) + 2\alpha(NG-R2) = 8\alpha + 1\beta$.



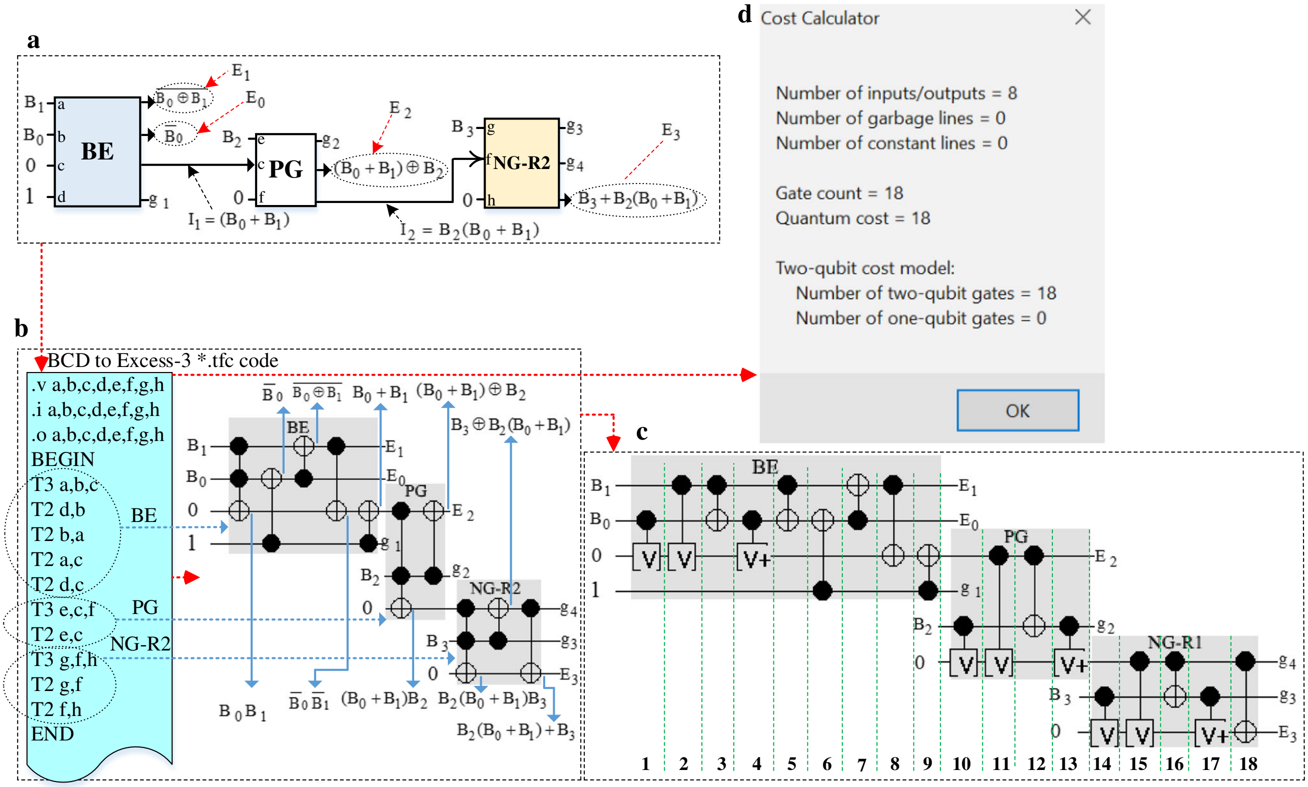
Algorithm 2: BCD to Excess-3 code converter

```
Input, Output: B = (B0, B1, B2, B3) in BCD form, Output: E = (E0, E1, E2, E3) in binary.
1: For i=0 to n-1 do
2:  If i=1 then
       (B1, B0, 0, 1) →BE       // Assign input to BE
       BE← (E1, E0)             // Catch two target output
       BE← (I1)                 //  One Intermediate output I1
    End if, Else
3:     (B2, 0) →PG              // Assign input to PG
       (I2) →PG                 // Intermediate output of BE associated with the input of PG
       PG← (E2)                 // Catch one target output
    End if, Else
4: If i=3 then
      (B3, 0) →NG-R2      //  Assign input to NG-R2
      (I2) →NG-R2         // Intermediate output of PG associated with the input of NG-R2
      NG-R2← (E3)         //  Catch one target outputs
    BE← (GO1),  PG← (GO2), NG-R2← (GO3, GO4)      //  Remaining as GO,
5: End if, end if, end for,
6: Return(Ei), End;
```

5.3 The proposed reversible BCD-to-Gray code converter

The BCD-to-Gray circuit is composed of three gates, namely NG-R2, FG, and F2G (Fig. 9a). In this circuit, the intermediate bits such as I_1, I_2, and I_3 are determined by the input bits. The Toffoli gate block and BCD-to-Gray.tfc code are shown in Fig. 9b. To construct the BCD-to-Gray converter, we propose Algorithm 3, which is named the BCD-to-Gray code converter construction algorithm. The BCD-to-Gray QE circuit is presented in Fig. 9c. The construction of this circuit required QC of 8 (Fig. 9d). To the best of the authors' knowledge, this is the first BCD-to-Gray circuit to be proposed using state-of-the-art technology.

Proposition 2 *The LC of the BCD-to-Gray converter is* $LC(\textit{BCD-to-Gray}) = 2\alpha(NG-R2) + 1\alpha(FG) + 2\alpha(F2G) = 5\alpha$.



BCD to Excess-3 *.tfc code

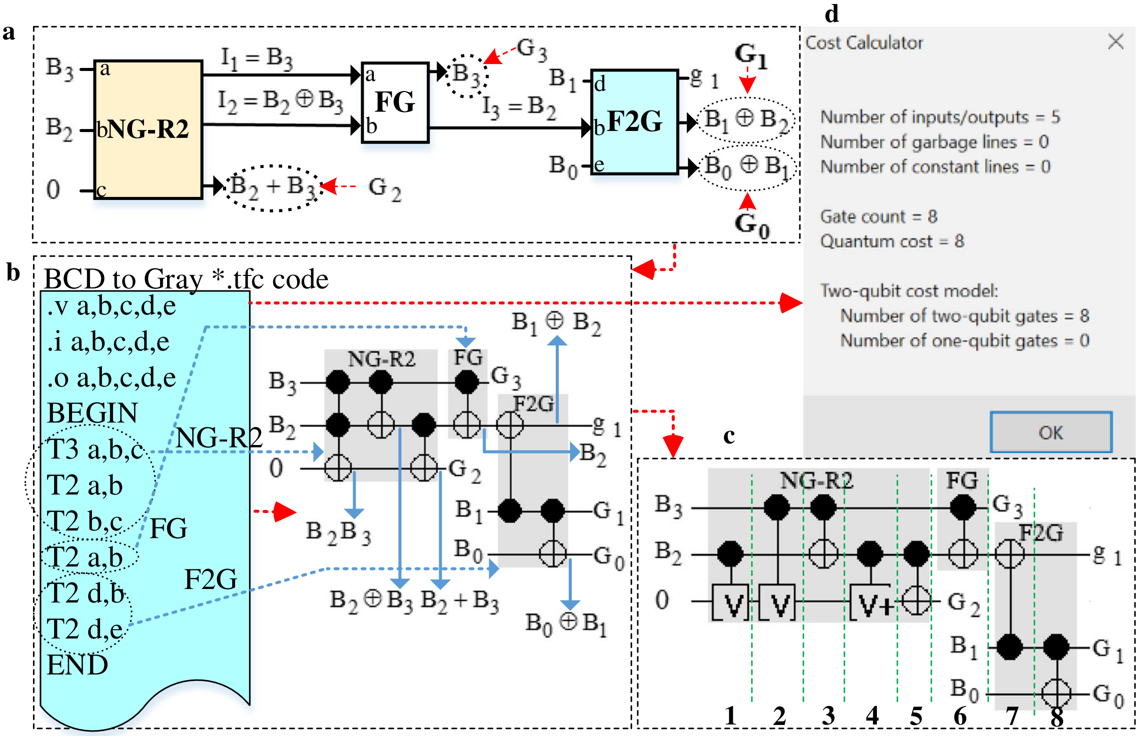
```
.v a,b,c,d,e,f,g,h
.i a,b,c,d,e,f,g,h
.o a,b,c,d,e,f,g,h
BEGIN
T3 a,b,c
T2 d,b
T2 b,a
T2 a,c
T2 d,c
T3 e,c,f
T2 e,c
T3 g,f,h
T2 g,f
T2 f,h
END
```

Number of inputs/outputs = 8
Number of garbage lines = 0
Number of constant lines = 0

Gate count = 18
Quantum cost = 18

Two-qubit cost model:
Number of two-qubit gates = 18
Number of one-qubit gates = 0

Fig. 8 BCD-to-excess-3 converter: **a** schematic diagram, **b** Toffoli gate block and BCD to excess-3.tfc code, **c** QE, and **d** snapshot of result



BCD to Gray *.tfc code

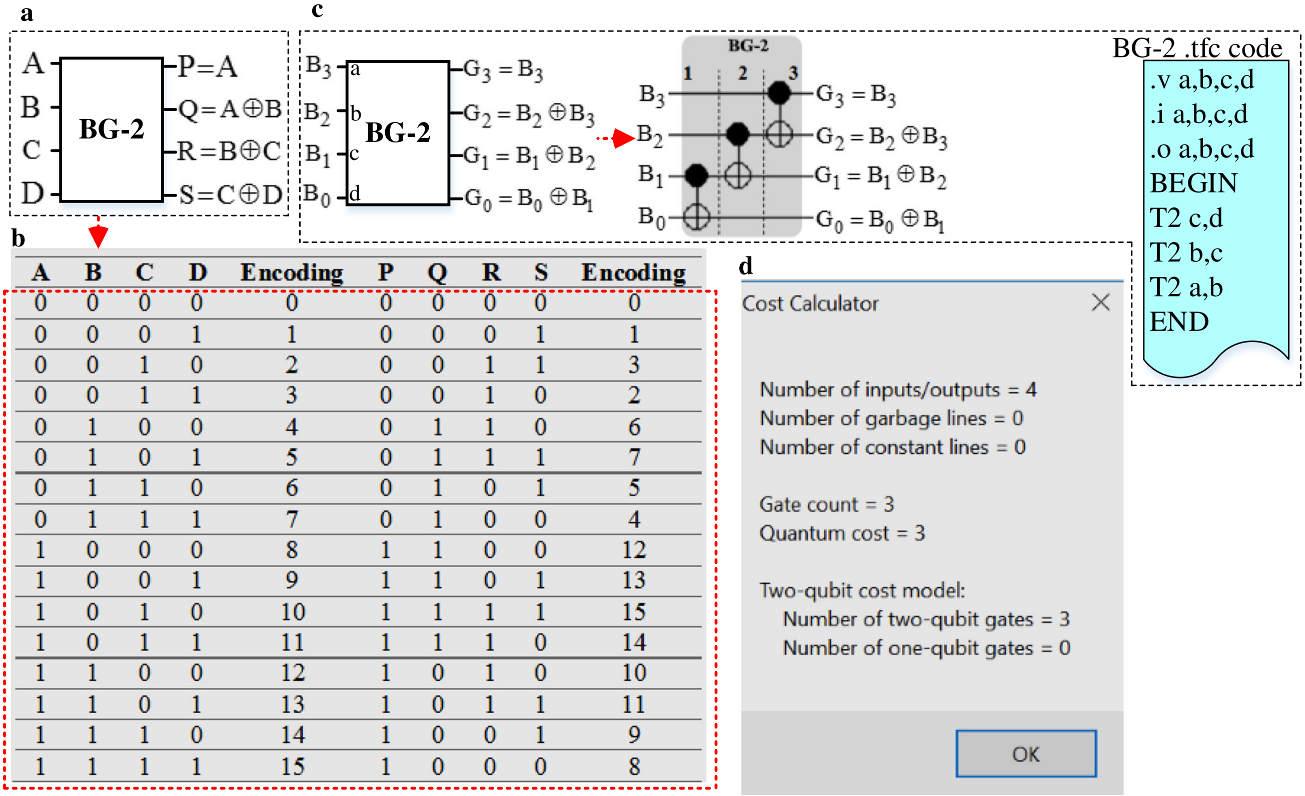
```
.v a,b,c,d,e
.i a,b,c,d,e
.o a,b,c,d,e
BEGIN
T3 a,b,c
T2 a,b
T2 b,c
T2 a,b
T2 d,b
T2 d,e
END
```

Number of inputs/outputs = 5
Number of garbage lines = 0
Number of constant lines = 0

Gate count = 8
Quantum cost = 8

Two-qubit cost model:
Number of two-qubit gates = 8
Number of one-qubit gates = 0

Fig. 9 BCD-to-Gray converter: **a** schematic diagram, **b** Toffoli gate block and BCD to Gray.tfc code, **c** QE, and **d** snapshot of results



A	B	C	D	Encoding	P	Q	R	S	Encoding
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	1	1
0	0	1	0	2	0	0	1	1	3
0	0	1	1	3	0	0	1	0	2
0	1	0	0	4	0	1	1	0	6
0	1	0	1	5	0	1	1	1	7
0	1	1	0	6	0	1	0	1	5
0	1	1	1	7	0	1	0	0	4
1	0	0	0	8	1	1	0	0	12
1	0	0	1	9	1	1	0	1	13
1	0	1	0	10	1	1	1	1	15
1	0	1	1	11	1	1	1	0	14
1	1	0	0	12	1	0	1	0	10
1	1	0	1	13	1	0	1	1	11
1	1	1	0	14	1	0	0	1	9
1	1	1	1	15	1	0	0	0	8

```
.v a,b,c,d
.i a,b,c,d
.o a,b,c,d
BEGIN
T2 c,d
T2 b,c
T2 a,b
END
```

```
Number of inputs/outputs = 4
Number of garbage lines = 0
Number of constant lines = 0

Gate count = 3
Quantum cost = 3

Two-qubit cost model:
    Number of two-qubit gates = 3
    Number of one-qubit gates = 0
```

Fig. 10 Binary-to-Gray converter: **a** the proposed BG-2 gate, **b** truth table, **c** schematic diagram, quantum equivalent, and BG-2.tfc code, and **d** snapshot of results

Algorithm 3: BCD-to-Gray code converter

```
Input, output: B = ($B_0$, $B_1$, $B_2$, $B_3$) in BCD form, Output: G = ($G_0$, $G_1$, $G_2$, $G_3$) in binary
1:  For i= 0 to n-1 do
2:  If i=1 then
        ($B_3$, $B_2$, 0) →NG-R2          // Assign input to NG-R2
        NG-R2← ($G_2$)                   // Catch one target output
        NG-R2← ($I_1$, $I_2$)               //  Intermediate output $I_1$ and $I_2$
      End if, Else
3:     ($I_1$, $I_2$) →FG                  // Intermediate output of NG-R2 associated with the input of FG
        FG← ($G_3$)                     // Catch one target output
        FG← ($I_3$)                     //  Intermediate output $I_3$
     End if, Else
4:  If i=3 then
       ($I_3$) →F2G    // Intermediate output of FG-associated with the input of F2G
       ($B_1$, $B_0$) →F2G        //  Assign input to F2G
       F2G← ($G_0$, $G_1$)      //  Catch one target output
       F2G← ($GO_1$)        //  Remaining as garbage output
End if, end if, end for,
Return($G_i$), End:
```

5.4 The proposed binary-to-Gray and Gray-to-binary code converters

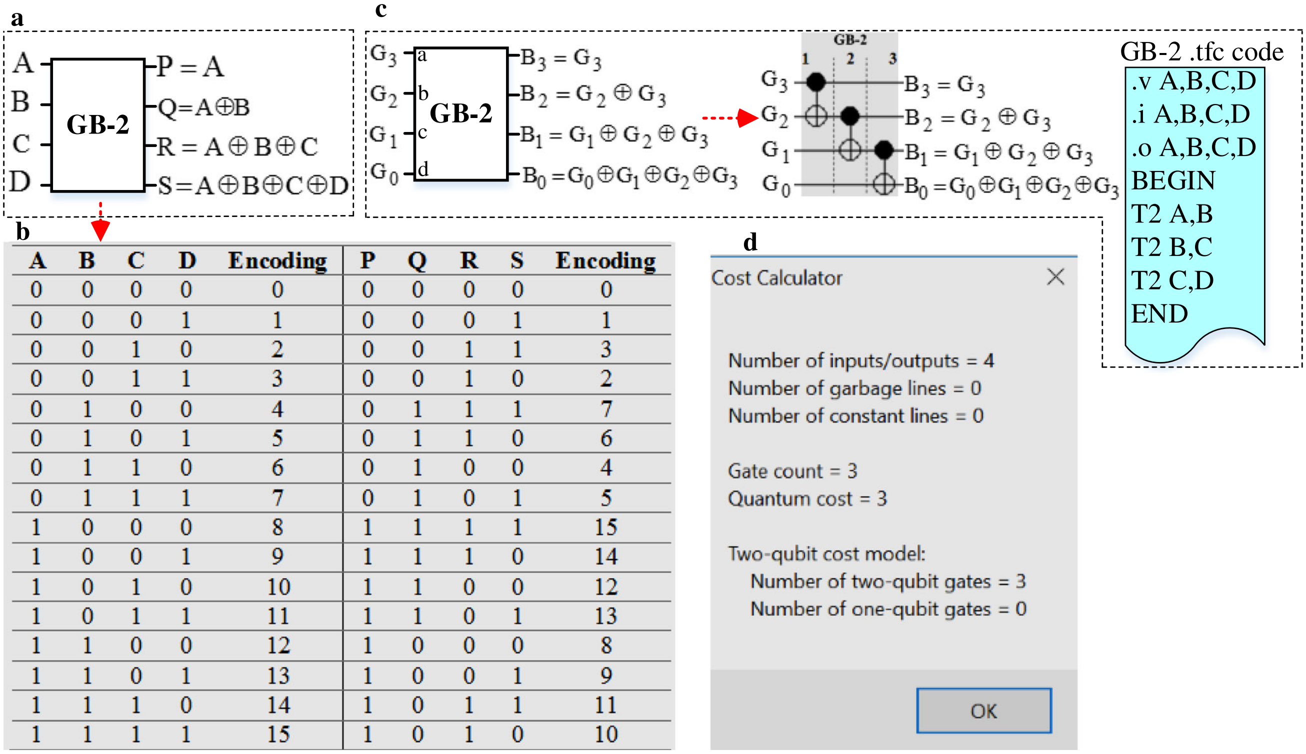
Binary-to-Gray code conversion can be considered to be a one-bit change between two successive numbers in the converter output. Existing circuits for binary-to-Gray and Gray-to-binary conversion usually have high GC, UD, LC, GO, and QC values, as studied in [20,23,24,27]. This work is quite different in that one novel reversible gate can be used to construct this circuit. To reduce the cost metrics of these code converter circuits, new reversible gates without GO or CI were constructed.

5.5 The proposed BG-2 gate and binary-to-Gray code converter

The proposed BG-2 (binary-to-Gray) gate and its truth table are shown in Fig. 10a, b. The binary-to-Gray code converter was constructed with GC of only 1 and no GO or CI to synthesize the outputs. Figure 10c, d presents the schematic, QE, BG-2.tfc code, and snapshot of results. The LC for the BG-2 was calculated as 3α.

5.6 The proposed GB-2 gate and Gray-to-binary code converter

The GB-2 (Gray-to-binary) gate proposed herein and its truth table are shown in Fig. 11a, b. The presented Gray-to-binary code converter circuit is free from GO and CI, leading to a good fit with reversible circuit synthesis, as depicted in Fig. 11c. The circuit inputs are placed on the left side, and the outputs on the right. Figure 10c, d shows the schematic, QE, GB-2.tfc code, and snapshot of results.



```
.v A,B,C,D
.i A,B,C,D
.o A,B,C,D
BEGIN
T2 A,B
T2 B,C
T2 C,D
END
```

A	B	C	D	Encoding	P	Q	R	S	Encoding
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	1	1
0	0	1	0	2	0	0	1	1	3
0	0	1	1	3	0	0	1	0	2
0	1	0	0	4	0	1	1	1	7
0	1	0	1	5	0	1	1	0	6
0	1	1	0	6	0	1	0	0	4
0	1	1	1	7	0	1	0	1	5
1	0	0	0	8	1	1	1	1	15
1	0	0	1	9	1	1	1	0	14
1	0	1	0	10	1	1	0	0	12
1	0	1	1	11	1	1	0	1	13
1	1	0	0	12	1	0	0	0	8
1	1	0	1	13	1	0	0	1	9
1	1	1	0	14	1	0	1	1	11
1	1	1	1	15	1	0	1	0	10

Cost Calculator ×

Number of inputs/outputs = 4
Number of garbage lines = 0
Number of constant lines = 0

Gate count = 3
Quantum cost = 3

Two-qubit cost model:
Number of two-qubit gates = 3
Number of one-qubit gates = 0

OK

Fig. 11 Gray-to-binary converter: **a** the proposed GB-2 gate, **b** truth table, **c** schematic diagram, quantum equivalent, and GB-2.tfc code, and **d** snapshot of results

The LC of the GB-2 gate was calculated as follows: The expression for P has LC $= 0$. The expression for Q $(A \oplus B)$ has LC $= 1\alpha$. We then apply signal duplication twice to obtain $(A \oplus B)$ for use in the expressions for both R and S. Note that we utilize FG for reversible signal duplication. The LC of the FG is 1α. Then, the LC can be computed as: LC $= 0\alpha\ (P) + 1\alpha\ (Q) + 2\alpha$ (for the two signal duplications for the expressions for R and S) $+ 1\alpha\ (R) + 2\alpha (S) = 6\alpha$.

5.7 The proposed reversible DRC

This subsection presents the construction and algorithm for the reversible dual-rail checker circuit. Such a DRC could be used for testing, although no parity bit checking is required. The DRC produces two outputs: $Z_1 = X_0Y_1 + X_1Y_0$ and $Z_2 = X_0X_1 + Y_0Y_1$, with complementary values. In this circuit, on the left, two NG-R1 gates are utilized to AND four-bit numbers, then the OR function is generated on the right using two NG-R2 gates. This circuit contains four gates (2× NG-R1, and 2× NG-R2), requiring 10 inputs and 10 outputs. The input includes four-bit numbers (X_0, X_1, Y_0, and Y_1) and six ancilla inputs, as depicted in Fig. 12a. Its reversible quantum equivalent circuit is presented in Fig. 12b, c. This circuit produces two outputs: Z_1, and Z_2, while the other eight outputs are garbage outputs. We use the DRC.tfc code to construct the Toffoli gate block, as shown in Fig. 12b. Meanwhile, Fig. 12d shows a snapshot of the results. Testing of the DRC circuit is possible using the two outputs $Z_1 = X_0Y_1 + X_1Y_0$ and $Z_2 = X_0X_1 + Y_0Y_1$. The DRC circuit was constructed herein using Algorithm 4.

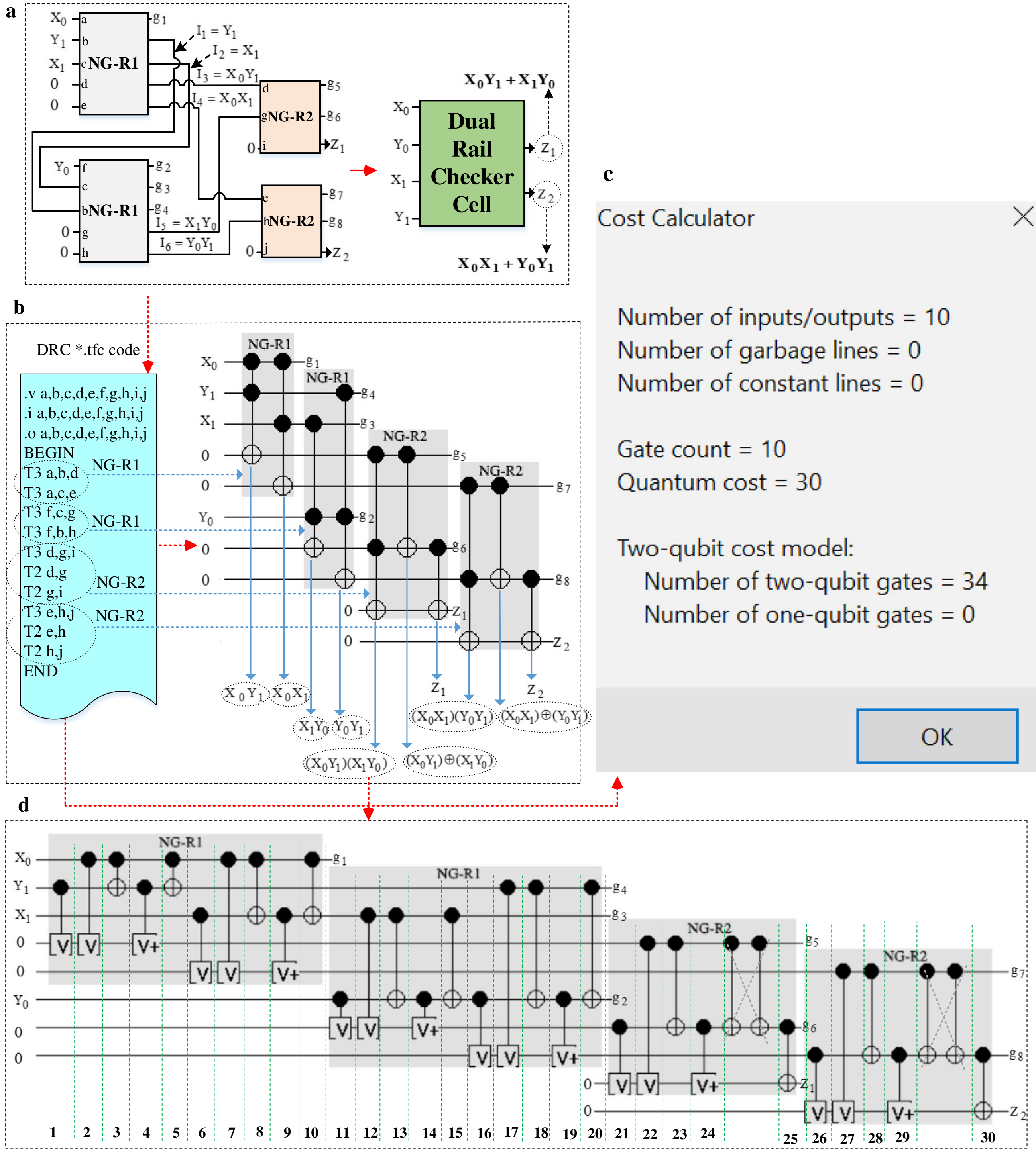
Algorithm 4: Dual-rail-checker

```
Input, output: I = (X_0, X_1, Y_0, Y_1) in binary, Output: Z = (Z_1, Z_2) in binary.
1:  For i= 0 to n-1 do
2:  If i=1 then
        (X_0, Y_1, X_1, 0, 0) →NG-R1     // Assign input to NG-R1
        NG-R1← (I_1, I_2, I_3, I_4)        // Intermediate output I_1, I_2, I_3 and I_4
    End if, Else
3:      (Y_0) →NG-R1        // Assign input to second NG-R1
        (I_2, I_1) →NG-R1     // Intermediate output of first NG-R1 associated with the input of second NG-R1
        NG-R1← (I_5, I_6)     // Intermediate output I_5, and I_6
    End if, Else
4:  If i=3 then
        (I_3, I_5) →NG-R2     // Intermediate output of first and second NG-R2 associated with input of first NG-R1
     End if, Else
5:      (I_4, I_6) →NG-R2     // Intermediate output of first and second NG-R2 associated with the input of second NG-R1
     Else
      NG-R2← (Z_1)            // Catch the one target output from first NG-R2
     NG-R2← (Z_2)             // Catch the one target output from second NG-R2
    End if, end if, end for,
    Return (Z_i), End;
```

Proposition 3 *The LC of the DRC is* $2(2\alpha + 2\beta)$ *(NG-R1)* $+ 2(2\alpha)$ *(NG-R2)* $= 8\alpha + 4\beta$.

Lemma 1 *A DRC is a circuit that performs the complement of two outputs.*

Proof Since mathematical expressions are involved in the testing process of a DRC, the testing solution can even be obtained manually for a modest DRC approach. These manual calculations for the Z_1 and Z_2 outputs of the DRC are illustrated below: □



```
.v a,b,c,d,e,f,g,h,i,j
.i a,b,c,d,e,f,g,h,i,j
.o a,b,c,d,e,f,g,h,i,j
BEGIN
T3 a,b,d
T3 a,c,e
T3 f,c,g
T3 f,b,h
T3 d,g,i
T2 d,g
T2 g,i
T3 e,h,j
T2 e,h
T2 h,j
END
```

Cost Calculator

Number of inputs/outputs = 10
Number of garbage lines = 0
Number of constant lines = 0

Gate count = 10
Quantum cost = 30

Two-qubit cost model:
Number of two-qubit gates = 34
Number of one-qubit gates = 0

Fig. 12 DRC: **a** schematic diagram, **b** Toffoli gate block and DRC.tfc code, **c** QE, and **d** snapshot of results

Outputs expressions: $Z_1 = [X_0X_1] \times \begin{bmatrix} Y_1 \\ Y_0 \end{bmatrix}$, $Z_2 = [X_0Y_0] \times \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix}$

Condition 1 Selecting inputs of $(X_0, X_1) = (1, 1)$ and $(Y_0, Y_1) = (0, 0)$, the computed outputs are $Z_1 = [X_0X_1] \times \begin{bmatrix} Y_1 \\ Y_0 \end{bmatrix} = 0$, $Z_2 = [X_0Y_0] \times \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} = 1$ (i.e., the complement of Z_1). Hence, no fault occurs.

Condition 2 Selecting inputs of $(X_0, X_1) = (1, 1)$ and $(Y_0, Y_1) = (1, 0)$, the computed outputs are $Z_1 = [X_0X_1] \times$

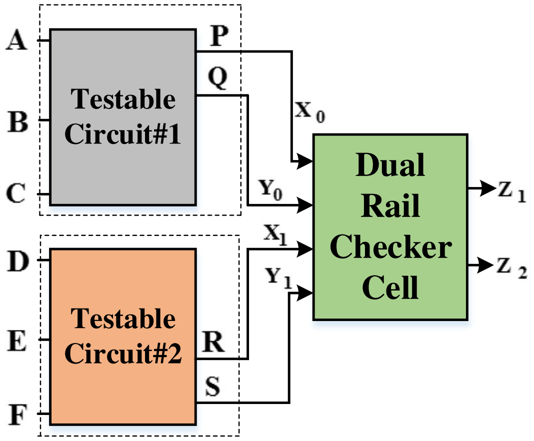


Fig. 13 Schematic presentation of testing circuits using DRC

$\begin{bmatrix} Y_1 \\ Y_0 \end{bmatrix} = 1$, $Z_2 = [X_0 Y_0] \times \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} = 1$ (not the complement of Z$_1$). Hence, fault occurs.

Therefore, a DRC can take two pairs of inputs, i.e., (X_0, X_1) and (Y_0, Y_1), and the outputs are complementary.

5.8 Testing strategy using DRC cell

Testing of the DRC cell is presented in Sect. 5.7. The testable circuit#1 and testable circuit#2 were placed such that all the outputs (P, Q, R, and S) mapped to all the inputs of the DRC cell (Fig. 13). The steps to construct a testable circuit are illustrated in Algorithm 5. The task of testing is to place the DRC cell at the output of the circuit to be tested. Since all the outputs of the tested circuit that are correctly connected to the DRC must have the same logic, the DRC cell generates the complementary outputs according to the same procedure as presented in Lemma 3.

Algorithm 5: Testing a circuit by DRC cell

```
Input, Output: I = (A, B, C, D, E, F) in binary, Output Z = (P, Q, R, S) in also binary
For i=0 to n-1 do
 If i=1 then
    (A, B, C) →Testable Circuit#1          //Assign input to Testable Circuit#1
    Testable Circuit#1← (P, Q)             // Two intermediate outputs associated to DRC
 End if, Else
   (D, E, F) →Testable Circuit#2           //Assign input to Testable Circuit#2
    Testable Circuit#1← (R, S)             // Two intermediate outputs associated to DRC
  End if, Else
   If $Z_1 = Z_2 \oplus 1$ then      //DRC checker
      Fault free checker
  Else
   Fault in the checker
 End if, end if, end for, End;
```

6 Experimental results and comparative evaluation

Quantum circuit construction was carried out using the RCviewer+ tool and results obtained on an Intel(R) core(TM) i5-6200U 2.40-GHz CPU with 4 GB RAM under Windows 10 Home (64 bit). The proposed circuits were first built using the *.tfc code, then processed using the RCviewer+ tool. The proposed circuits were compared with previously proposed circuits, in terms of GC, QC, UD, LC, and UD.

Cost metrics and statistics for the proposed versus counterpart designs are presented in Tables 2, 3, 4, and Table 5. A comparison of the reversible metrics for the proposed BCD-to-excess-3 versus existing designs is presented in Table 2, showing that the proposed design surpasses all existing designs reviewed in [19–22]. According to Table 2, the optimal circuit in [20] was selected and its cost metrics compared, revealing GC of 25 % less than [20], CI of 42.8 % less than [20], GO of 42.8 % less than [20], and QC of 5.26 % less than [20]. Note that our BCD-to-excess-3 code converter is

Table 2 Cost metric statistics for BCD-to-excess-3

Design	Gate type	QE	LC	GC	CI	GO	UD	QC
Circuit#1 [20]	HNG	N	$20\alpha + 4\beta$	4	6	6	4	24
Circuit#2 [20]	HNG, FG	N	$19\alpha + 4\beta$	4	7	7	4	19
[21]	URG, FG	N	$13\alpha + 5\beta$	8	7	12	4	38
[22]	FG, TG	N	Not mentioned	12	8	8	4	40
[23]	TG, URG, NOT, NG3	N	$6\alpha + 5\beta + 2\delta$	6	3	10	3	24
Novel	BE, PG, NG-R2	Y	$8\alpha + 1\beta$	3	4	4	3	18
Improvement % w.r.t. circuit#1 [20]				+25	+33.3	+33.3	+25	+25
Improvement % w.r.t. circuit#1 [20]				+25	+42.85	+42.85	+25	+5.26
Improvement % w.r.t. [21]				+62.5	42.85	66.66	+25	+52.63
Improvement % w.r.t. [22]				+75	+50	+50	+25	+55
Improvement % w.r.t. [23]				+50	NI	+60	NI	+25

Table 3 Cost metric statistics for binary-to-Gray converter

Design	Gate type	QE	Inputs	QE	LC	GC	CI	GO	UD	QC
Circuit#3 [20]	FG, F2G	N	4-bit	N	5α	3	2	2	2	5
Circuit#4 [20]	FG	N	4-bit	N	3α	3	0	0	3	3
[21]	FG	N	4-bit	N	3α	3	0	3	2	3
[23]	FG, NG3	N	4-bit	N	3α	2	0	1	2	3
[24]	FG	N	3-bit	N	2α	2	0	1	2	2
[27]	BG-1	Y	3-bit	Y	2α	1	0	0	1	2
Novel	BG-2	Y	4-bit	Y	5α	1	0	0	1	3
Improvement % w.r.t. circuit#3 [20]						+66.6	+100	+100	+50	+40
Improvement % w.r.t. circuit#4 [20]						+66.6	NI	NI	+50	NI
Improvement % w.r.t. [21]						+66.6	NI	+100	+66.6	NI
Improvement % w.r.t. [23]						+50	NI	+100	+50	NI

Table 4 Cost metric statistics for Gray-to-binary converter

Design	Gate type	QE	Inputs	QE	LC	GC	CI	GO	UD	QC
[20]	FG	N	4-bit	N	3α	3	0	0	3	3
[21]	FG	N	4-bit	N	5α	5	2	3	5	5
[23]	NG1, NG2	N	4-bit	N	6α	2	0	0	2	3
[24]	FG	N	3-bit	N	2α	2	0	0	2	2
[27]	GB-1	Y	3-bit	Y	2α	1	0	0	1	2
Novel	GB-2	Y	4-bit	Y	6α	1	0	0	1	3
Improvement % w.r.t. [20]						+66.66	NI	NI	+66.66	NI
Improvement % w.r.t. [21]						+80	+100	+100	+80	+40
Improvement % w.r.t. [23]						+50	NI	NI	+50	NI

Table 5 Cost metric statistics for DRC

Design	Gate type	QE	LC	GC	CI	GO	UD	QC
Circuit#1 [25]	NPPRG	N	$24\alpha +6\beta$	6	12	14	5	54
Circuit#2 [25]	F2G, NPPRG	N	$28\alpha+ 6\beta$	8	16	18	4	58
[26]	R	N	$12\alpha+ 6\beta+ 6\delta$	6	6	8	6	36
Novel	NG-R1, NG-R2	Y	$8\alpha+ 4\beta$	4	6	8	3	30
Improvement % w.r.t. circuit#1 [25]				+33.33	+50	+42.85	+40	+44.44
Improvement % w.r.t. circuit#2 [25]				+50	+62.5	+55.55	NI	+48.27
Improvement % w.r.t. [26]				+33.33	NI	NI	+50	+16.66

NI no improvement, *Y* yes, *N* no

competitive with existing works and offers improved metrics in all cases.

The binary-to-Gray and Gray-to-binary code converter circuits proposed using the described approach offer substantial improvements over reported counterpart designs with reversible metrics. Tables 3 and 4 present the performance of our binary-to-Gray and Gray-to-binary code converters versus counterpart works. The binary-to-Gray converter in [24] has lower quantum cost, but its design was 3-bit while the binary-to-Gray code converter presented herein is 4-bit and also has lower cost metrics such as GC, CI, GO, UD, and QC. The GC for the binary-to-Gray converter is 1 (66.6 %) less than that in [20], and the GC for the Gray-to-binary converter is also 1 (66.6 %) less than that in [20]. It is obvious that these novel circuits, including the binary-to-Gray and Gray-to-binary code converters, are much better in terms of these cost metrics.

From Table 5, it is evident that the reversible metrics for the dual-rail checker circuit presented herein are significantly lower compared with those of the best counterpart design in

[25]. The constructed circuit has GC of 4 (33.33 %) less than [25], CI of 6 (50 %) less than [25], GO of 8 (42.8 %) less than [25], and QC of 30 (44.44 %) less than [25]. Hence, one can state that this novel DRC indeed has lower cost metrics. In fact, our DRC circuit is competitive with counterpart designs, with improvement in some cases.

7 Conclusions

We have constructed reversible code converters and a dual-rail checker circuit. We propose an algorithm to transform a cascaded reversible gate circuit into a respective quantum equivalent circuit. A primary advantage of this algorithm is the four-step process needed to obtain the quantum equivalent circuit. We targeted reversible designs for code converters and a dual-rail checker, aiming to optimize their cost metrics, including the gate count, ancilla inputs, garbage outputs, logical calculation, unit delay, and quantum cost. Use of particular types of reversible gate for a particular logical function is very effective for construction of the proposed circuits with minimal cost metrics. The proposed BCD-to-excess-3 circuit has 42.85 % less garbage output, 25 % less gate count, and 5.26 % less quantum cost compared with existing designs. The constructed dual-rail checker circuit shows 44.4 % improvement in quantum cost with respect to existing designs. Besides the design, the Toffoli gate block and quantum equivalent circuit of each design are also constructed by our proposed algorithm. Hence, the new code converters and dual-rail checker circuit offer improved performance, representing the best choice for low-cost quantum computing. In the future, any cascaded gate-level circuit can be quickly converted into a quantum equivalent circuit using the proposed algorithm, enabling development of quantum computing frameworks by anyone.

Acknowledgements The authors would like to thank the anonymous reviewers for their constructive criticism and effective advice that improved a preliminary version of this paper.

References

1. Dueck, G.W.: Challenges and advances in Toffoli network optimization. IET Comput. Digit. Tech. **8**(4), 172–177 (2014)
2. De Vos, A.: Alexis: 'Reversible Computing: Fundamentals, Quantum Computing, and Applications', p. 261. Wiley (2011)
3. Maslov, D., Saeedi, M.: Reversible circuit optimization via leaving the Boolean domain. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **30**(6), 806–816 (2011)
4. Gaur, H.M., Singh, A.K.: Design of reversible circuits with high testability. Electron. Lett. **52**(13), 1102–1104 (2016)
5. Landauer, R.: Irreversibility and heat generation in the computing process. IBM J. Res. Dev. **5**(3), 183–191 (1961)
6. Bennett, C.H.: Logical reversibility of computation. IBM J. Res. Dev. **17**(6), 525–532 (1973)
7. Athas, W.C., Svensson, L.J.: Reversible logic issues in adiabatic CMOS. IEEE Proceedings Workshop on Physics and Computation PhysComp'94, pp. 111–118 (1994)
8. Thomsen, M.K., Glück, R.: Reversible arithmetic logic unit for quantum arithmetic. J. Phys. A: Math. Theor. **43**(38), 382002 (2010)
9. Houri, S., Valentian, A., Fanet, H.: Comparing CMOS-based and NEMS-based adiabatic logic circuits. In: 5th International Conference on Reversible Computation, Victoria, BC, Canada, pp. 36–45. Springer, Berlin, Heidelberg (2013)
10. Shamsujjoha, M., Hossain, F., Ali, M.N.Y., Babu, H.M.H.: Optimized fault tolerant designs of the reversible barrel shifters using low power MOS transistors. J. Comput. Electron. **14**(3), 726–746 (2015)
11. Knil, E., Laflamme, R., Milburn, G.J.: A scheme for efficient quantum computation with linear optics. Nature **409**, 46–52 (2001)
12. Chabi, A.M., Roohi, A., Khademolhosseini, H., Sheikhfaal, S., Angizi, S., Navi, K., DeMara, R.F.: Towards ultra-efficient QCA reversible circuits. Microsyst. Microprocess. (2016). doi:10.1016/j.micpro.2016.09.015
13. Abdessaied, N., Amy, M., Drechsler, R., Soeken, M.: Complexity of reversible circuits and their quantum implementations. Theor. Comput. Sci. **618**, 85–106 (2016)
14. Yang, G., Song, X., Perkowski, M.A., Hung, W.N.N., Biamonte, J., Tang, Z.: Four-level realisation of 3-qubit reversible functions. IET Comput. Digit. Tech. **1**(4), 382–388 (2007)
15. Sen, B., Ganeriwal, S., Sikdar, B.K. Reversible Logic-Based Fault-Tolerant Nanocircuits in QCA, pp. 1–9. ISRN Electronics (2013)
16. Li, M.-C., Zhou, R.-G.: A novel reversible carry-selected adder with low latency. Int. J. Electron. **103**(7), 1202–1215 (2016)
17. Akbar, E.P.A., Haghparast, M., Navi, K.: Novel design of a fast reversible Wallace sign multiplier circuit in nanotechnology. Microelectron. J. **42**(8), 973–981 (2011)
18. Sen, B., Dutta, M., Some, S., Sikdar, B.K.: Realizing reversible computing in QCA framework resulting in the efficient design of testable ALU. ACM J. Emerg. Technol. Comput. Syst: JETC **11**(3), 30 (2014)
19. Thapliyal, H., Ranganathan, N.: Design of reversible sequential circuits optimizing quantum cost, delay, and garbage outputs. ACM J. Emerg. Technol. Comput. Syst: JETC **6**(4), 14 (2010)
20. Haghparast, M., Hajizadeh, M., Hajizadeh, R., Bashiri, R.: On the synthesis of different nanometric reversible converters. Middle-East J. Sci. Res. **7**(5), 715–720 (2011)
21. Saravanan, M., Suresh Manic, K.: Energy efficient code converters using reversible logic gates. In: IEEE International Conference on Green High Performance Computing (ICGHPC), pp. 1–6 (2013)
22. Gandhi, S.M., Devishree, J., Venkatesh, J., Mohan, S.S.: Design of reversible circuit for code converter and binary incrementer. Int. J. Inf. Technol. Mech. Eng. **1**(4), 24–33 (2014)
23. Kamani, K., Koneti, S., Boolampalli, U., Shankara, S.: Energy efficient reversible logic design of code converter. In: IEEE International Conference on Green High Performance Computing (ICGHPC), vol. 1, no 3, pp. 132–136 (2014)
24. Das, J.C., De, D.: Reversible binary to grey and grey to binary code converter using QCA. IETE J. Res. **61**(3), 223–229 (2015)
25. Sasamal, T.N., Singh, A.K., Mohan, A.: Design of two-rail checker using a new parity preserving reversible logic gate. Int. J. Comput. Theory Eng. **7**(4), 3–11 (2015)
26. Vasudevan, D., Lala, P.K., Di, J., Parkerson, J.P.: Reversible-logic design with online testability. IEEE Trans. Instrum. Meas. **55**(2), 406–414 (2006)
27. Misra, N.K., Wairya, S., Singh, V.K.: Optimized approach for reversible code converters using quantum dot cellular automata. In: Proceedings of the 4th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA), Springer India, pp. 367–378 (2015)

28. Große, D., Wille, R., Dueck, G.W., Drechsler, R.: Exact synthesis of elementary quantum gate circuits. Mult Valued Log. Soft Comput. **15**(4), 283–300 (2009)
29. Wille, R., Lye, A., Drechsler, R.: Considering nearest neighbor constraints of quantum circuits at the reversible circuit level. Quantum Inf. Process. **13**(2), 185–199 (2014)
30. Sasanian, Z., Wille, R., Miller, D.M.: Clarification on the Mapping of Reversible Circuits to the NCV-v1 Library. arXiv preprint, arXiv:1309.1419 (2013)