Realizing Reversible Circuits Using a New Class of Quantum Gates

Zahra Sasanian  
University of Victoria  
Victoria, BC, Canada  
sasanian@uvic.ca

Robert Wille  
University of Bremen  
Bremen, Germany  
rwille@uni-bremen.de

D. Michael Miller  
University of Victoria  
Victoria, BC, Canada  
mmiller@uvic.ca

ABSTRACT
Quantum computing offers a promising alternative to conventional computation due to the theoretical capacity to solve many important problems with exponentially less complexity. Since every quantum operation is inherently reversible, the desired function is often realized in reversible logic and then mapped to quantum gates. We consider the realization of reversible circuits using a new class of quantum gates. Our method uses a mapping that grows at a very low linear rate with respect to the number of controls. Results show that, particularly for medium to large circuits, our method yields substantially smaller quantum gate counts than do prior approaches.

Categories and Subject Descriptors
B.6.3 [Design Aids]: Optimization

General Terms
Design

Keywords
Reversible Logic, Quantum Gates, Mapping, Optimization

1. INTRODUCTION
Quantum computation [1] offers the promise of efficient computing for problems that are of exponential difficulty for classical computing paradigms. Here, information is stored in terms of qubits which provide the probabilistic superposition of the Boolean states 0 and 1. This enables solutions for many important problems (e.g. database search, factorization, graph problems) significantly faster than with classical approaches (see e.g. [2, 3, 4]). The states of the qubits are modified by quantum operations which are inherently reversible and can be represented by unitary matrices.

Considering that many of the established quantum algorithms include a significant Boolean component (e.g. the oracle transformation in the Deutsch-Jozsa algorithm, the database in Grover’s search algorithm, and the modulo exponentiation in Shor’s algorithm), it is crucial to have efficient methods to synthesize quantum gate realizations of Boolean functions. The problem is often approached by a two-stage procedure: First, a reversible circuit is designed using a reversible gate library. Then, the resulting reversible circuit is mapped into an equivalent quantum circuit.

The synthesis of reversible circuits has been extensively addressed e.g. in [5, 6, 7]. In this work, we focus on the mapping of reversible circuits to efficient quantum circuits. So far, the well known NCV quantum gate library (NOT, controlled-NOT and square-root-of-NOT gates) introduced in [8] has been applied to the mapping problem [9, 10]. Different optimization techniques have been introduced e.g. in [11]. However, mappings based on the NCV-library become very expensive particularly if large reversible gates are considered which often require so called ancillaries, i.e. circuit lines being utilized as temporary work lines only.

In this paper, we consider a modified NCV library motivated by the approach introduced in [12]. We introduce a new methodology for mapping reversible circuits into quantum circuits using this library. We demonstrate that the new library leads to realizations for multiple-control Toffoli gates with far fewer quantum gates than have been found using the NCV library. Our approach uses a structure similar to one recently introduced in [13]. However, while that work addressed a particular application, we here consider the realization of general reversible circuits.

Experiments demonstrate the benefits of the proposed mapping methodology. Compared to the best previously introduced methods, we show that our mapping yields substantially smaller circuits particularly for medium to large scale problems. More precisely, improvements of around 70% can be achieved on average. In the best case, the size of the circuits can even be reduced by approximately 90%. The proposed mapping has other advantages like the direct handling of Toffoli gates with mixed-polarity controls, as well as a better consideration of additional technology-based constraints like the nearest-neighbor constraint.

The remainder of this paper is structured as follows. The next section briefly reviews the basics of reversible and quantum circuits. Section 3 introduces the new gate library motivated by the work in [12] which forms the basis for our mapping methodology. Afterwards, the proposed mappings are presented in Section 4 and evaluated in Section 5. Finally, further benefits of the proposed mappings are discussed in Section 6 and conclusions are drawn in Section 7.

2. PRELIMINARIES
This section presents the background necessary for this paper. Readers interested in more detail should consult the literature, e.g. [1].

2.1 Reversible Functions, Gates and Circuits
A Boolean function \( f : \mathbb{B}^n \rightarrow \mathbb{B}^m \), \( \mathbb{B} = \{0, 1\} \) with inputs \( X = \{x_1, \ldots, x_n\} \) is reversible if it has the same number of inputs and outputs i.e. \( n = m \), and it maps each input
pattern to a unique output pattern. Otherwise, the function is termed irreversible. A reversible function can be realized by a circuit comprised of a cascade of reversible gates with no fan-out and feedback [1].

Several reversible gates have been introduced including the Toffoli gate [14], the Fredkin gate [15], and the Peres gate [16]. A multiple-control Toffoli (MCT) gate, a direct generalization of the basic Toffoli gate, has a target line \( x_t \) and control lines \( \{x_1, x_2, \ldots, x_n\} \). This gate maps \( (x_1, x_2, \ldots, x_n) \) to \( (x_1, x_2, \ldots, x_n, x_t) \), i.e., the target line is inverted if all the controls have value 1; otherwise the value on the target line is passed through unchanged. The values on the control and unconnected lines always pass through the gate unchanged. An MCT gate with no controls always inverts the target and is a NOT gate. An MCT gate with one control line is called a controlled-NOT (CNOT) gate (also known as the Feynman gate). The case of two control lines is the original gate defined by Toffoli.

MCT gates are universal in that all reversible functions can be realized using this gate type alone [14]. Fredkin and Peres gates can, for example, be realized using MCT gates. This paper considers reversible circuits composed of MCT gates. An MCT gate is denoted by \( T(C; t) \) where \( C \subseteq X \) is the possibly empty set of control lines and \( t \in X \setminus C \) is the target line. For drawing circuits, we follow the established convention of using the symbol \( \oplus \) to denote the target line and solid black circles to indicate control connections.

**Example 1.** Figure 1(a) shows an MCT circuit with three circuit lines and two gates that emulate a Peres gate [16]. As shown, this circuit maps the input pattern 111 to the output pattern 100. Note that the gate operations can be applied in any order. Positive (from input to output) and negative (from output to input) values are realized in a particular reversible function and from the outputs towards the inputs realizing the inverse of that function. This is because every MCT gate is its own inverse.

### 2.2 Quantum Gates & Circuits

The basic unit of quantum information is the qubit whose state is written as \( | \psi \rangle = \alpha |0 \rangle + \beta |1 \rangle \), where \( \alpha \) and \( \beta \) are complex numbers such that \( |\alpha|^2 + |\beta|^2 = 1 \). \( |0 \rangle \) and \( |1 \rangle \) are basis states corresponding to the classical 0 and 1 states.

The state of a single qubit can be expressed as a vector \( | \psi \rangle \). The state of a quantum system with \( n > 1 \) qubits can be represented as a normalized (length 1) vector with \( 2^n \) elements, called the state vector. A quantum circuit is a cascade of quantum gates and the operation of the circuit on the state vector corresponds to the multiplication of appropriate \( 2^n \times 2^n \) unitary matrices, one for each of the quantum gates [1].

A qubit has a potentially infinite number of values and there is also a potentially infinite number of distinct quantum gates. However, in practice researchers consider circuits composed of a small number of gate types.

The **NCV gate library** was introduced by Barenco et al. [8] and contains the following set of quantum gates:

- **NOT gate** \( T(\theta; t) \): A single qubit \( t \) is inverted which is described by the unitary matrix \( \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \).
- **Controlled NOT (CNOT) gate** \( T(c; t) \): The target qubit \( t \) is inverted if the control qubit \( c \) is 1.

- **Controlled V gate** \( V(\{c\}; t) \): The operation described by the unitary matrix \( V = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \) is performed on the target qubit \( t \) if the control qubit \( c \) is 1.
- **Controlled \( V^+ \) gate** \( V^+(\{c\}; t) \): The operation described by the unitary matrix \( V^+ = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \) is performed on the target qubit \( t \) if the control qubit \( c \) is 1. The \( V^+ \) gate performs the inverse operation of the \( V \) gate since \( V^+ = V^{-1} \).

The \( V \) and \( V^+ \) gates are referred to as controlled square-root-of-NOT gates since two adjacent identical \( V \), or \( V^+ \), gates are equivalent to a CNOT gate.

**Example 2.** Figure 1(b) shows an NCV circuit which is functionally equivalent to the circuit in Figure 1(a). Note the quantum value output of the first gate. To apply the circuit in reverse, i.e. from output to input, \( V \) and \( V^+ \) gates must be interchanged as they are the inverse of each other.

### 3. A NEW CLASS OF QUANTUM GATES

Although the NCV gate library is universal in the sense that every reversible Boolean function can be realized by a circuit composed of NCV gates [8], other libraries are of interest as they can lead to better circuits, e.g. fewer gates. In this work, we focus on a modification to the NCV gate library based on concepts introduced in [12].

If circuits with Boolean inputs use NCV gates only, the value of each qubit at each stage of the circuit is restricted to one of \( \{0, v_0, 1, v_1\} \) where \( v_0 = \frac{1}{\sqrt{2}} (|+\rangle + |\downarrow \rangle) \) and \( v_1 = \frac{1}{\sqrt{2}} (|+\rangle - |\downarrow \rangle) \).

The \( NOT, V, \) and \( V^+ \) operations over these four values are:

<table>
<thead>
<tr>
<th>( x )</th>
<th>NOT( (x) )</th>
<th>( V(x) )</th>
<th>( V^+(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>( v_0 )</td>
<td>( v_0 )</td>
<td>( v_1 )</td>
<td>0</td>
</tr>
<tr>
<td>( v_1 )</td>
<td>( v_1 )</td>
<td>( v_0 )</td>
<td>1</td>
</tr>
</tbody>
</table>

As shown, \( NOT \) is a complement operation, \( V \) is the cycle \( 0 \rightarrow v_0 \rightarrow 1 \rightarrow v_1 \rightarrow 0 \), and \( V^+ \) is the inverse cycle.

In this work, we adopt a new quantum gate library which we call the NCV-\( \{v_1\} \) library. The NCV-\( \{v_1\} \) gate library is composed of (1) the three unitary gates (i.e. gates without a control line) performing the \( NOT, V, \) and \( V^+ \) operation as well as (2) single-control versions of these gates. In contrast to the NCV-library, and in keeping with the work in [12], the controlled gates perform the respective operation not when the control line is 1, but rather when the control line is set to the value \( v_1 \). We label control connections for NCV-\( \{v_1\} \) gates with \( v_1 \) to emphasize this fact.

**Example 3.** Figure 1(c) shows an NCV-\( \{v_1\} \) circuit functionally equivalent to the circuits in Figures 1(a) and 1(b).

Besides the benefits in the physical implementation, as discussed in [12], this gate library enables a much more efficient mapping from an MCT gate circuit as we introduce below.

The implementation cost of a quantum gate is heavily technology dependent. Here we assume that all quantum gates, NCV and NCV-\( \{v_1\} \) in particular, have unit cost. Under that assumption the cost of a quantum circuit is the gate count. This is clearly an approximation but a suitable one when considering technology independent optimization.
4. PROPOSED MAPPING METHOD

The common approach to synthesize a quantum circuit implementing a reversible Boolean function has two steps. First, a circuit composed of reversible gates implementing the desired function is synthesized. That circuit is then mapped to a cascade of gates from the target quantum gate library. Optimizations can be applied at various stages of the mapping process (see e.g. [11]).

In this section, we briefly review the established mapping methodology based on the NCV library. We then introduce a new mapping approach based on the NCV-[e1] library.

We show that, except for very small circuits, the proposed mapping leads to circuits with significantly fewer gates than the circuits determined using the established NCV mapping.

4.1 Mapping Individual MCT Gates

The well-known [8] optimal mapping of a Toffoli gate \( T(\{c_1, c_2\}; t) \) to a cascade of NCV gates is shown in Figure 2(b). As shown, five NCV-gates are required. For MCT gates with more control lines, the number of required NCV-gates increases rapidly.

Table 1 shows the number of NCV gates required to realize MCT gates with up to 15 controls using the approach described in [10]. Besides the number of control lines, the number of ancillary lines available also affects the size of the quantum gate circuit. An ancillary line is a circuit line which is neither used as the target line nor as a control line of a Toffoli gate, and is thus available to be used as a temporary work line in the quantum realization. For each number of controls, the rightmost gate count is the lowest possible. Blank entries indicate when the availability of more ancillary lines is not advantageous. The NCV gate counts given in Table 1 are the best known for NCV gate realizations of MCT gates. It is important to note (1) that at least one ancillary line is required for three or more controls and (2) that the gate counts grow quite quickly with the number of controls. In Table 1 for \( c \geq 4 \), assuming the maximum number of ancillaries required is available, the number of gates is \( 12c - 28 \). If only one ancillary is available, for \( c \geq 10 \) the number of gates required is \( 24c - 132 \). These formulas have been verified for up to 20 controls and it is believed they will continue to hold for larger numbers of controls.

In contrast, better mappings with much slower linear growth are possible using the NCV-[e1] library. Therefore, the structure illustrated in Figure 3 for a Toffoli gate with 4 control lines is proposed. Here, the actual operation of the Toffoli gate (i.e., the inversion of the target line) is performed by a single CNOT gate controlled by \( v_1 \). The \( V \)-gates ensure that the control line of the CNOT gate is set to \( v_1 \) if, and only if, all control lines \( c_1 \) to \( c_4 \) are equal to 1. The last \( V^+ \) gates are needed to undo the corresponding \( V \) operations on the control lines in order to restore their values.

Generalizing this structure, every Toffoli gate with \( c \) control lines can be realized using a \( V \) gate and a \( V^+ \) gate for each control line as well as a single \( CNOT \) gate which operates on the target line. This leads to a total of \( 2c + 1 \) NCV-[e1] gates. While for a Toffoli gate with 1 control line only, this results to a more expensive mapping (3 gates instead of 1), MCT gates with more than 2 controls can be realized with significant reductions in the number of gates compared to the established NCV mappings. Furthermore, the NCV-[e1] mappings do not require any ancillary lines.

Table 1 lists the number of NCV-[e1] gates required for up to 15 controls. The gate counts are significantly lower than the NCV costs. The relative size of the NCV-[e1] circuits to the NCV circuits drops to about 30% at 6 controls and approaches 20% at 15 controls. We again emphasize the NCV-[e1] gate circuits require no ancillary lines.

4.2 Mapping a Reversible Circuit

In the last section, the mapping of single MCT gates was considered. We now address how to map circuits composed of MCT gates to a quantum circuit. This can be done by a direct substitution of each single MCT gate by its corresponding quantum gate cascade. Even for this simple approach, the mapping illustrated in Figure 3 leads to significant reductions in comparison to the established NCV mappings. However, even better results can be obtained if further optimizations are applied.

We employ an optimization method based on the ideas presented in [11] extended to handle the new NCV-[e1] gates. The optimization process involves a Line Labeling Procedure (LLP) and a Gate Reduction Procedure (GRP). Both are applicable to both MCT and quantum gates.

The LLP is used to assign labels to line segments such that two segments of a line have the same label only if they have the same functionality. The LLP is Procedure 1 of [11] extended to handle NCV-[e1] gates. The procedure involves a single pass through the circuit from the inputs toward the outputs. A stack of gate operations is kept for each line. As a gate \( g \) is processed, the stack for its target line is checked for a sequence of gates realizing the identity operation. When one is found the output line for gate \( g \) is assigned the same label that appears on the target line input to the first gate in the sequence. While this procedure only assigns the same label to two segments of the same line that have the same functionality, it is not guaranteed to find all equivalences. We have identified a few such cases, but in general the LLP
finds most equivalences in circuits we have considered.

The GRP is Procedure 2 of [11] extended to handle NCV-
| gates. The procedure starts from the input side of the
circuit and processes the gates in order. The LLP is used to
label the circuit up to the current gate \( \bar{g} \) of interest. Then,
gate \( g \) is moved back through the circuit to each of the places
that have the same labels on its control lines. Gate \( g \) can
not be moved past a point where there is a control on the
target line. As a gate is moved, a list is made that
contains gates that can be adjacent to it and have the same
target, controls, control labels, and control types. Then,
the gates in this list are removed from the circuit and an
optimized equivalent sequence, which can be empty when
the gates implement the identity function, is inserted in the
position of the removed gate closest to the circuit’s inputs.
The procedure then proceeds for subsequent gates until the
end of the circuit is reached.

In our overall approach, the MCT gate circuit is first
optimized using the GRP. Then, the appropriate NCV-
| realization is substituted for each gate in the optimized
MCT circuit. Finally, the resulting NCV-| realization is
optimized using the GRP. Because of the regular structure of
the new realizations, significant improvements are typical in
the NCV-| optimization step.

Example 4. Consider the circuit depicted in Figure 4(a).
No optimization is possible for the MCT gates. Replacing
each Toffoli gate with the appropriate version of the realiza-
tion from Figure 2(b) yields a circuit with 15 NCV gates.
Applying the GRP reduces this to the 12 gate circuit in Fig-
ure 4(b). In contrast, while using the NCV-| realization
from Figure 2(c) also results in a 15 gate circuit but applying
the GRP yields the 9 gate circuit shown in Figure 4(c).

5. EXPERIMENTAL RESULTS

In this section, we present results obtained by the pro-
posed approach. We first consider the improvements achieved
by the new mapping methodology. Then, we also briefly dis-
cuss how the resulting circuits have been verified.

5.1 Evaluation

The procedures described in the previous section have
been implemented using Python 2.7.1. Our experiments
were run on a computer with a Core 2 Duo 2.66 GHz CPU
and 4.0 GB RAM. We used a test suite of 138 circuits from
Position 4 in Table 2. Due to space limitations, the table shows only those circuits for which the
improvement was greater than 30\% (50 of the 138 circuits).

Each row of the Table gives: (1) The name of the circuit
including the RevLib file index number. Note that Fred-
kin and Peres gates in the RevLib circuits are substituted
by MCT gate realizations before applying our techniques.
(2) The quantum gate count given on the RevLib site (called
quantum cost in RevLib). (3) The NCV gate count for cir-
cuits determined using the mappings from [10] with the
optimization techniques described in [11]. (4) The gate count
for direct substitution of the 2+1 NCV-| gate realization
for each MCT gate in the given circuit. (5) The direct map-
ping NCV-| gate count is reported for the circuit found
by first applying the GRP to the MCT circuit. (6) The
NCV-| gate count for the circuit from (5) optimized at the
NCV-| level using the GRP.

For all 138 circuits, the gate reduction for all circuits in
total is 81.8\% with respect to the counts from RevLib and
68.7\% with respect to the NCV circuits determined using
the techniques from [10] and [11].

As the results in Table 2 show, our approach does very
well for medium to large circuits since those circuits tend
to have more MCT gates with greater than 2 controls than
do the small circuits. In addition the smaller circuits often
have a high proportion of CNOT gates which as noted above
require 3 NCV-| gates. To be precise, our methods yield
slightly more gates than the RevLib gate count for 33 of the
circuits: 22 with 1-3 extra gates; 4 with 4; 2 for each of 5, 6 and 7; and 1 circuit with 8 additional gates. The
great majority of these circuits are small and have a high
proportion of CNOT gates, i.e. the fact that CNOT is not
a primitive in the NCV-| library has a major effect. In
contrast there are 21 circuits where the NCV-| circuit
has more than 1000 fewer gates and a further 6 with from 139
to 984 fewer gates than the circuit reported in RevLib.

The total improvement comes primarily from the new
MCT to NCV-| gate mapping. Besides, the optimiza-
tions described in Section 4.2 reduce the gate count by a fur-
ther 4.5\% (MCT gate optimization) and by a further 32.5\%
(NCV-| gate optimization) on average. Note that for n-
line circuits that have an MCT gate of size \( n \), an ancillary
line must be added to map to NCV circuits. Additional lines
are never required for NCV-| circuits. Considering unit
delay for all 1-qubit and 2-qubit quantum gates as in [18],
NCV-| circuits are much faster than the NCV circuits as
they have lower logic depth.

5.2 Circuit Verification

Verification methods have been applied to confirm that
the circuits produced by our methods are functionally equiv-
alent to the original RevLib circuits. This is an interesting
problem on its own. The RevLib circuit is binary whereas
the circuit our method produces uses 4-valued logic gates.
Also our basic MCT gate substitution relies on the fact that
the circuits: 22 with 1-3 extra gates; 4 with 4; 2 for each of
5, 6 and 7; and 1 circuit with 8 additional gates.
gate mapping has a number of further important benefits.

### 6.1 MCT Gates with Negative Controls

Thus far, we have assumed that MCT gates have positive control lines, i.e. that the control lines must all have the value 1 in order to perform the corresponding operation on the target line. But a number of reversible circuit synthesis algorithms, e.g.

- [21, 22], produce circuits with MCT gates that also have negative controls, i.e. controls that are activated by the value 0. This affects the number of gates in an NCV gate circuit realization.

For example, the circuit in Figure 2(b) has five gates for a Toffoli gate with two positive control lines. If a Toffoli gate has one positive and one negative control line, the number of required gates would remain 5. However, if a Toffoli gate has two negative controls, an additional NOT gate is needed leading to a total of 6 gates.

NCV realizations of MCT gates with mixed (positive and negative) controls have been considered in [23]. To give an idea of the results, the NCV realization of a 3-controlled MCT gate has 14 gates for 0 or 1 negative controls, 16 for 2 negative controls, and 18 for 3 negative controls. For 15 controls, the number of NCV gates ranges from 228 to 246 if there is one ancillary line available and from 152 to 168 if 3 ancillary lines are available.

The situation is quite different for our new NCV-[v1] gate realizations. Consider the structure illustrated in Figure 3. To change a positive control to a negative control, the V gate on that line simply has to be swapped with the corresponding V⁺ gate. Hence, our mapping always leads to 2c+1 gates for c controls regardless of the mix of positive and negative controls. Thus, in contrast to the NCV situation, the NCV-[v1] gate circuits handle negative controls for free. And once again, no ancillary lines are required.

### 6.2 Nearest-Neighbor Constraints

Many quantum technologies, e.g. [24, 25], require that a circuit satisfies the nearest-neighbor constraint, i.e. all controlled gates in a circuit have the control line and the target line adjacent. A circuit can be modified to satisfy this constraint by adding gates which swap the values of lines so that only adjacent lines are used. How best to locate the swap gates has been studied e.g. in [26, 27]. As an example, the NCV circuit in Figure 2(b) does not satisfy the nearest-neighbor constraint and can not be made nearest-neighbor by permitting the lines. This also holds for
REFERENCES


