Optimizing the Mapping of Reversible Circuits to Four-Valued Quantum Gate Circuits

Mathias Soeken* Zahra Sasanian† Robert Wille* D. Michael Miller† Rolf Drechsler*‡

*Institute of Computer Science
University of Bremen, Germany
{msoeken,rwille}@uni-bremen.de

†Department of Computer Science
University of Victoria, Canada
{sasanian,mmiller}@uvic.ca

‡Cyber-Physical Systems, DFKI GmbH
28359 Bremen, Germany
rolf.drechsler@dfki.de

***Abstract*—This paper considers the optimization of reversible circuits composed of multiple-control Toffoli gates to quantum circuits using the well-known NCV-$|1\rangle$ (NCV) library and the recently introduced NCV-$|v_1\rangle$ library which both use a four-valued model for the quantum gates. The techniques introduced handle positive and negative controls which are central to many reversible circuit synthesis procedures. Experimental results confirm the methods are applicable to circuits obtained by diverse synthesis methods. The results also show the significant advantage of the NCV-$|v_1\rangle$ library.**

I. Introduction

Quantum computation [1] has received significant attention in recent years. Using quantum circuits, many important problems, e.g. factorization or data-based search, can be solved exponentially faster in comparison to conventional technologies. Most of the underlying circuits include a significant reversible Boolean function component, e.g. the oracle in Grover's search algorithm [2] or the modulo exponentiation in Shor's algorithm [3]. As a consequence, synthesis and optimization of quantum circuits realizing Boolean functions have become important research areas [4]–[8].

The synthesis of a quantum circuit for a reversible Boolean function is typically done in two stages. First, the desired function is realized as a Boolean reversible circuit. The circuit is then mapped to a quantum circuit. Several synthesis approaches leading to reversible circuits (see [9] for a survey of methods) and respective mapping algorithms leading to quantum circuits (e.g. [10]) have been proposed. Since the number of circuit lines — so called qubits — usually is a limited resource in quantum circuits, the methods usually try to keep the number of circuit lines as small as possible.

Optimized mapping of a synthesized reversible Boolean circuit to a quantum circuit is clearly crucial. First attempts led to quantum circuit realizations with very high numbers of gates even for relatively small functions [11]. Recently, more efficient mappings have been introduced [12]–[14].

However, existing mapping methodologies typically do not fully exploit the characteristics of reversible circuits obtained by established synthesis methods. For example, many well-known synthesis approaches yield circuits containing cascades of reversible gates with similar control lines and a common target line. Furthermore, new technology mapping possibilities have been introduced (e.g. [15]). Better circuits arise if these factors are taken into account as will be shown below.

In this paper, we propose new optimization and mapping methods aiming for the efficient realization of quantum circuits for reversible Boolean functions. In particular, we introduce

- an improved optimization method for reversible Boolean circuits exploiting the characteristics of the circuits obtained by established synthesis methods,
- an improved mapping method considering two different quantum gate libraries, and
- an improved optimization of the resulting quantum circuits.

Combining the proposed mapping and optimization methods leads to quantum circuits which are significantly cheaper compared to previously presented realizations. In fact, improvements in the average of 79% are possible when using the well-known NCV-$|1\rangle$ (often just NCV in the literature) quantum gate library and up to 92% when using the recently introduced NCV-$|v_1\rangle$ gate library [15].

The necessary background for this paper is reviewed in Section II. Section III discusses common synthesis methods as well as the characteristics of reversible circuits they produce. The proposed optimization and mapping methods are described in Section IV. Finally, experimental results are presented in Section V before the paper is concluded in Section VI.

II. Background

This section outlines the background on reversible and quantum circuits necessary for this paper.

A. Reversible Logic

A Boolean function is *reversible* if, and only if, it has the same number of inputs and outputs, 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. Fan-out and feedback are not allowed [1].

Several reversible gates have been introduced. A *multiple-control Toffoli (MCT) gate*, a direct generalization of the basic Toffoli gate [16], has a *target line* x_j and *control lines* $\{x_{i_1}, x_{i_2}, \ldots, x_{i_k}\}$. This gate maps $(x_1 x_2 \ldots x_j \ldots x_n)$ to $(x_1 x_2 \ldots (x_{i_1} x_{i_2} \ldots x_{i_k}) \oplus x_j \ldots x_n)$, i.e. the target line is inverted if all the control lines 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.

In this paper, we consider an extension to the MCT gate model that allows each control to be activated by either value 1 (a *positive control*) or value 0 (a *negative control*). We term this a *mixed-polarity MCT gate* (MPMCT). An MCT gate is a case of an MPMCT gate with all controls positive.

A cascade of MPMCT gates with a common target is an implementation of a an *exclusive-or sum-of-products* (ESOP) expression [17]. If all gates are MCT gates, it is a *positive-polarity* Reed-Muller expression. As shown below, techniques for reducing such expressions can be used to optimize the corresponding gate cascades.

An MPMCT gate will be denoted by $T(C,t)$ where C is the possibly empty set of control lines and t is the target line. For drawing circuits, we follow the established conventions of using the symbol $\oplus$ to denote the target line, solid black circles to indicate positive controls and white circles to indicate negative controls.

B. Quantum Logic

The basic unit of quantum information is the *qubit* whose *state* is written as $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers such that $|\alpha|^2 + |\beta|^2 = 1$. $|0\rangle$ and $|1\rangle$ are basis states corresponding to the conventional 0 and 1 logic states.

The quantum state of a single qubit can be expressed as a vector $\binom{\alpha}{\beta}$. 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].

The very frequently considered *NCV-$|1\rangle$ gate library* was introduced by Barenco et al. [11] and contains the following universal set of quantum gates:

- NOT gate $T(\emptyset, t)$: A single qubit t is inverted which is described by the unitary matrix $\mathbf{X} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$.
- Controlled NOT (CNOT) gate $T(\{c\}, t)$: The target qubit t is inverted if the control qubit c is $|1\rangle$ (this fact also leads to the name NCV-$|1\rangle$).
- Controlled V gate $V(\{c\}, t)$: The operation described by the unitary matrix $\mathbf{V} = \frac{1+i}{2}\left(\begin{smallmatrix} 1 & -i \\ -i & 1 \end{smallmatrix}\right)$ is performed on the target qubit t if the control qubit c is $|1\rangle$.
- Controlled $V^\dagger$ gate $V^\dagger(\{c\}, t)$: The operation described by the unitary matrix $\mathbf{V}^\dagger = \frac{1-i}{2}\left(\begin{smallmatrix} 1 & i \\ i & 1 \end{smallmatrix}\right)$ is performed on the target qubit t if the control qubit c is $|1\rangle$. The $V^\dagger$ gate performs the inverse operation of the V gate since $\mathbf{V}^\dagger = \mathbf{V}^{-1}$.

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

If circuits with Boolean inputs use NCV-$|1\rangle$ gates only, the value of each qubit at each stage of the circuit is restricted to one of $\{|0\rangle, |v_0\rangle, |1\rangle, |v_1\rangle\}$ where $|v_0\rangle = \frac{1+i}{2}\binom{1}{-i}$ and $|v_1\rangle = \frac{1+i}{2}\binom{-i}{1}$. The $\mathbf{X}$, $\mathbf{V}$, and $\mathbf{V}^\dagger$ operations over these four logic values are given in Table II-B.

TABLE I
FOUR-VALUED GATE OPERATIONS

$\|\varphi\rangle$	$\mathbf{X}\|\varphi\rangle$	$\mathbf{V}\|\varphi\rangle$	$\mathbf{V}^\dagger\|\varphi\rangle$
$\|0\rangle$	$\|1\rangle$	$\|v_0\rangle$	$\|v_1\rangle$
$\|v_0\rangle$	$\|v_1\rangle$	$\|1\rangle$	$\|0\rangle$
$\|1\rangle$	$\|0\rangle$	$\|v_1\rangle$	$\|v_0\rangle$
$\|v_1\rangle$	$\|v_0\rangle$	$\|0\rangle$	$\|1\rangle$

As shown, $\mathbf{X}$ is a complement operation, $\mathbf{V}$ is the cycle $(|0\rangle, |v_0\rangle, |1\rangle, |v_1\rangle)$, and $\mathbf{V}^\dagger$ is the inverse cycle.

In this paper, we also consider the *NCV-$|v_1\rangle$ gate library*, a four-valued quantum gate library introduced in [15]. This library is composed of

- the three single-qubit gates (i.e. gates without a control line) performing the $\mathbf{X}$, $\mathbf{V}$, and $\mathbf{V}^\dagger$ operations
- single-control versions of these gates. In contrast to the NCV-$|1\rangle$-library, and in keeping with the work in [18], the controlled gates perform the respective operation not when the control line is set to $|1\rangle$, but rather when the control line is set to $|v_1\rangle$. We label control connections for NCV-$|v_1\rangle$ gates with a $|v_1\rangle$ to emphasize this point.

In addition to the benefits in the physical implementation, as discussed in [18], the NCV-$|v_1\rangle$ gate library also enables a much more efficient mapping of an MPMCT gate circuit as we show below. However, since the NCV-$|1\rangle$-library is more frequently used, it is often just referred to as NCV.

III. MPMCT Circuit Synthesis

MCT and MPMCT circuit synthesis has been widely studied and is not a major focus of this paper. But, as would be expected the synthesis approach used can significantly affect the possible optimizations.

Many well-known synthesis approaches apply a similar paradigm. That is, the function to be synthesized is traversed and gates are added to the circuit and the equivalent transformation applied to the function being synthesized until the function has been transformed to the identity function. Different data-structures have been used to store the function to be synthesized. For example, truth tables are used in the *transformation-based synthesis* (TBS) method introduced in [19], Reed-Muller spectra (RMS) are used for the method introduced in [20], and *Quantum Multiple-Valued Decision Diagrams* (QMDD) [21] are used for the synthesis approach recently introduced in [22]. While these approaches can be efficient, they frequently lead to circuits with unacceptably high costs. This is mainly caused by the fact that the approaches are greedy and choose gates such that the already traversed parts of the circuit are not affected. The amount of information used in choosing gates varies. TBS traverses the truth table from the top to the bottom using only local information. RMS uses a similar traversal scheme but the more global information associated with each Reed-Muller coefficient drives the gate selection. The QMDD based synthesis method traverses nodes in a prescribed order but the QMDD structure captures important structural information about the function being synthesized that affects the choice of gates.

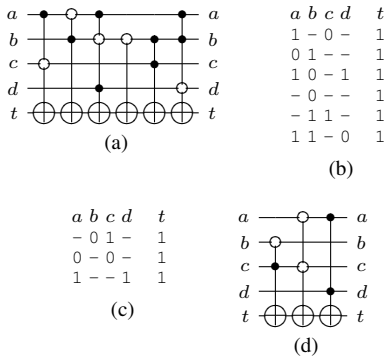

Fig. 1. Circuit optimization based on ESOP minimization

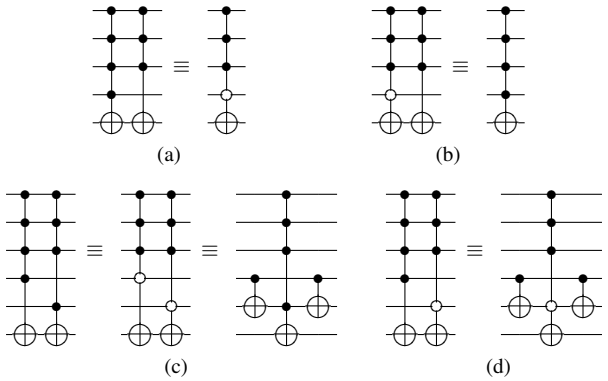

Fig. 2. MPMCT Reduction Rules

Note: In the illustrations above, one can change all controls on any of the top three lines to negative.

One characteristic feature that the three approaches outlined have in common is that, in order to not modify already traversed parts of the function description, gates with a significant number of control lines are typically added to the circuit. In addition, the synthesized circuits often include cascades of Toffoli gates with similar control lines. In particular, cascades of the form $T(C_1, t) \dots T(C_k, t)$ frequently occur, i.e. cascades of Toffoli gates which work on the same target line.

IV. Mapping and Optimization Techniques

In this section, we introduce advance mapping and optimization techniques that realize an efficient quantum circuit realization for a given reversible circuit obtained e.g. by the previously reviewed synthesis approaches. The proposed approach involves three stages: (1) MPMCT circuit optimization, (2) MPMCT to quantum gate mapping, and (3) quantum gate optimization.

A. MPMCT Optimization

Before a given reversible circuit is mapped to a quantum circuit, two MPMCT optimization techniques are applied. Both apply to gates with a common target. Note that this includes cascades directly generated by the synthesis approach as well as situations where the gates in the circuit can be rearranged to create such a cascade.

ESOP-based Optimization. The idea is illustrated by means of the cascade in Fig. 1(a). The six MPMCT gates act on the same target t. This can be expressed as an ESOP as shown in cube notation in Fig. 1(b). Note that each positive (negative) control leads to a positive (negative) literal in the respective cube. Each line which does not contain a control line is represented by a don't-care in the cube. This ESOP can be reduced to the one shown in Fig. 1(c) using a program such as EXORCISM-4 [23]. The circuit from Fig. 1(a) is thus optimized to the circuit in Fig. 1(d). In this example, the number of quantum gates found using standard NCV-$|1\rangle$ gate mapping [24] is reduced from 46 to 16, an improvement of almost two thirds.

Rule-based Optimization. The second MPMCT optimization procedure applies a number of reduction rules:

1) $T(C, x_t)T(C, x_t) = I$
2) $T(C, x_t)T(C \cup \{x_i\}, x_t) = T(C \cup \{\overline{x_i}\}, x_t)$
3) $T(C \cup \{x_i\}, x_t)T(C \cup \{x_j\}, x_t)$
   $= T(\{x_i\}, x_j)T(C \cup \{x_j\}, x_t)T(\{x_i\}, x_j)$
4) $T(C \cup \{\overline{x_i}\}, x_t)T(C \cup \{\overline{x_j}\}, x_t)$
   $= T(\{x_i\}, x_j)T(C \cup \{x_j\}, x_t)T(\{x_i\}, x_j)$
5) $T(C \cup \{x_i\}, x_t)T(C \cup \{\overline{x_j}\}, x_t)$
   $= T(\{x_i\}, x_j)T(C \cup \{\overline{x_j}\}, x_t)T(\{x_i\}, x_j)$

The first rule comes from the fact that MPMCT gates are self-inverse. The second rule is for two adjacent MPMCT gates that are different only in one control connection. Fig. 2(a) and Fig. 2(b) illustrate the possible reductions in this case. Reduction rules 3 to 5 are illustrated in Fig. 2(c) and Fig. 2(d).

The procedure employs the Circuit Line Labeling procedure described in [25]. The Line Labeling Procedure (Procedure 1 of [25]) traverses a circuit assigning labels to line segments such that two segments on the same line that are assigned the same label have identical functionality.

The MPMCT optimization procedure finds possible reductions in the circuit by moving gates across the circuit and making them adjacent to every gate in their movement domain. Gates are moved based on the new moving rule introduced in [25] according to which a gate can be moved to places in a circuit that have the same labels for its control lines as long as its target does not pass over a control connection. The new moving rule outperforms the commonly used old moving rule [19], [26] as it provides more freedom to move gates. The procedure starts from one end of the circuit and labels one MPMCT gate at a time using the Circuit Line Labeling Procedure. Then, it moves that gate back through the circuit as far as possible to find the best reduction. The gate may either be canceled using Rule 1 or may be reduced to a less expensive cascade using Rules 2 to 5. After a reduction is applied, the optimization restarts from the position of the earliest gate in the substituted cascade.

B. MPMCT to Quantum Gate Mapping

After MPMCT optimization, the resulting MPMCT circuit is mapped to a quantum circuit. We first describe the mapping of a single gate. For the NCV-$|1\rangle$ library, we use the catalogue of circuits described in [13]. The cost depends on the number

TABLE II
NCV-$|1\rangle$ COST OF MPMCT GATES FOR $n = 1 \ldots 8$ WITH $n-3$ ANCILLARY LINES.

	Number of Negative Controls							
Controls $c = n-1$	0	1	2	3	4	5	6	7
0	1							
1	1							
2	5							
3	14	14	16	18				
4	20	20	20	22	24			
5	32	32	32	34	36	38		
6	44	44	44	44	46	48	50	
7	56	56	56	56	58	60	62	64

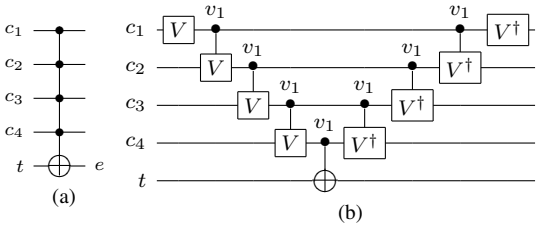

Fig. 3. Mapping of a 4-control MCT gate (a) to NCV-$|v_1\rangle$ gates (b).

of controls, the number of those controls that are negative and the number of ancillary lines available. An ancillary line is a circuit line which is not a control nor the target for a gate and, thus, available as a temporary line in the NCV-$|1\rangle$ realization. Table II is indicative of the costs.

For the NCV-$|v_1\rangle$ library we use the structure introduced in [15] and illustrated in Fig. 3 for an MCT gate with 4 controls. This structure is based on a circuit suggested in [27].

Recall that NCV-$|v_1\rangle$ gate controls are activated by the logic value $|v_1\rangle$. In the illustrated structure, the operation of an MCT gate $T(\{x_1, \ldots, x_m\}, x_t)$ is performed by a CNOT gate $T(\{x_m\}, x_t)$ controlled by the value $|v_1\rangle$. A V gate is applied to the first control x_1 and $m-1$ controlled-V gates (controlled by the $|v_1\rangle$ value) are applied to $x_2 \ldots x_m$ respectively to ensure that the CNOT operates only when all of the controls of the MCT gate are set to 1. Finally, the corresponding set of inverse gates are applied to the controls to restore their values. Note that a negative control can be incorporated by exchanging the V and $V^\dagger$ gates on the line for that control. Consequently, MPMCT gates are realizable in this library with no additional cost. Note also that no ancillary lines are ever required.

Mapping to NCV-$|1\rangle$ Gates. The NCV-$|1\rangle$ mapping procedure is similar to the Procedure 2 in [25] with extensions to incorporate MPMCT gates rather than just MCT gates. For each MPMCT gate G, all of the following gates that can be made adjacent to G are examined to find a pair whose combined quantum cost when implemented as a single unit is minimum. The selected pair is substituted by its NCV-$|1\rangle$ realization and the procedure proceeds to the next MCT gate. This procedure is greedy and it does not examine different realizations caused by control permutation in the MPMCT gates.

Mapping to NCV-$|v_1\rangle$ Gates. For NCV-$|v_1\rangle$ gates, we have found that a direct mapping converting each MPMCT gate to its NCV-$|v_1\rangle$ realization independently using the controls in the order given is the most effective. In particular, the method described above for NCV-$|1\rangle$ gates does not give better results for the NCV-$|v_1\rangle$ case. We also tried an alternative procedure for the NCV-$|v_1\rangle$ case that finds a set of common controls among as many consecutive MPMCT gates as possible and for those gates starts the mapping from the controls in common. This approach heuristically considers local constraints to determine the mapping structure. Compared to the direct mapping method, that uses a fixed order among all gates in a circuit, while the new approach yields more reduction in some circuits, it worsens the mapping results for others. We have found that on average the results are worse for this method. We are exploring other heuristics for ordering the controls but to date direct mapping is the best.

C. Quantum Circuit Optimization

The optimization approach introduced in [28] is employed in this work with extensions to handle the NCV-$|v_1\rangle$ library. Considering the gates in order from the circuit input side, the current gate is marked (using the Line Labeling Procedure of [25]) and then each quantum gate earlier in the circuit that can be made adjacent to the selected gate (using the new moving rule in [25]) and has the same control and target is added to a list. After forming the list, those gates are removed from the circuit and an equivalent minimal cascade is substituted with the leftmost gate in the list. The procedure then proceeds from the first gate following the substituted cascade. Note that the optimized equivalent sequence may be empty which indicates that the corresponding set of gates realizes the identity function.

V. EXPERIMENTAL RESULTS

We have implemented the mapping and optimization techniques discussed above using C++ and Python. Table III shows the results for a selection of benchmark functions with 10 to 15 lines as considered in [22]. For each function, we consider three synthesis methods: TBS [29], RMS [20], and QMDD-based [22]. TBS and RMS produce circuits with only MCT gates. QMDD-based synthesis produces circuits with MPMCT gates.

For each function and synthesis method we show the quantum gate count (GC) for four cases:

Case (a) The MPMCT gates in the circuit produced by the synthesis method are individually mapped to quantum gates using the approach first introduced in [11] and further refined in [24]. The quantum gate library used contains the NCV-$|1\rangle$ gates as well as higher order roots-of-NOT gates. The number of quantum gates required for each MPMCT gate depends on the number of controls and the number of ancillary lines available as shown in Table IV.

Case (b) In this case, the MPMCT gate optimizations presented in Section IV-A are applied to the circuit produced by the synthesis method and the quantum gate count is then determined as in Case (a).

Case (c) In this case, the optimized circuit from Case (b) is mapped to the NCV-$|1\rangle$ library using the approach described in Section IV-B. The mapped circuit is then reduced using

TABLE III
EXPERIMENTAL RESULTS

Function	n	TBS [29]				RMS [20]				QMDD [22]			
		Case (a) GC	Case (b) GC (%)	Case (c) GC (%)	Case (d) GC (%)	Case (a) GC	Case (b) GC (%)	Case (c) GC (%)	Case (d) GC (%)	Case (a) GC	Case (b) GC (%)	Case (c) GC (%)	Case (d) GC (%)
max46	10	12018	5231 (56.5)	3144 (73.8)	741 (93.8)	11176	3878 (65.3)	1987 (82.2)	483 (95.7)	7496	3878 (48.3)	1987 (73.5)	483 (93.6)
rd73	10	5923	4729 (20.2)	3233 (45.4)	1015 (82.9)	1027	452 (56.0)	343 (66.6)	194 (81.1)	13863	935 (93.3)	667 (95.2)	285 (97.9)
sqn	10	7516	5482 (27.1)	3614 (51.9)	1036 (86.2)	1872	1279 (31.7)	1084 (42.1)	354 (81.1)	3515	1306 (62.8)	862 (75.5)	284 (91.9)
sym9	10	4694	5086 (-8.4)	2873 (38.8)	704 (85.0)	4458	5379 (-20.7)	2871 (35.6)	643 (85.6)	10513	5379 (48.8)	2871 (72.7)	643 (93.9)
dc1	11	12152	9566 (21.3)	6274 (48.4)	1018 (91.6)	157	143 (8.9)	137 (12.7)	79 (49.7)	455	243 (46.6)	196 (56.9)	91 (80.0)
wim	11	32646	20726 (36.5)	14088 (56.8)	3395 (89.6)	181	171 (5.5)	165 (8.8)	100 (44.8)	259	169 (34.7)	157 (39.4)	73 (71.8)
z4	11	1093	709 (35.1)	583 (46.7)	244 (77.7)	461	162 (64.9)	140 (69.6)	83 (82.0)	3630	464 (87.2)	384 (89.4)	180 (95.0)
cm152a	12	421	244 (42.0)	170 (59.6)	99 (76.5)	410	211 (48.5)	139 (66.1)	75 (81.7)	211	211 (0.0)	139 (34.1)	75 (64.5)
cycle10	12	1204	1204 (0.0)	724 (39.9)	91 (92.4)	1204	1204 (0.0)	724 (39.9)	91 (92.4)	3003	1220 (59.4)	814 (72.9)	220 (92.7)
plus63mod4096	12	23584	978 (95.9)	621 (97.4)	59 (99.7)	2830	2830 (0.0)	1715 (39.4)	108 (96.2)	1585	986 (37.8)	632 (60.1)	59 (96.3)
rd84	12	12145	8493 (30.1)	6304 (48.1)	1752 (85.6)	2101	976 (53.5)	688 (67.3)	327 (84.4)	33908	1838 (94.6)	1401 (95.9)	487 (98.6)
sqrt8	12	55342	40964 (26.0)	25554 (53.8)	5709 (89.7)	2597	913 (64.8)	716 (72.4)	249 (90.4)	3943	521 (86.8)	376 (90.5)	152 (96.1)
adr4	13	5245	3258 (37.9)	2527 (51.8)	727 (86.1)	631	215 (65.9)	187 (70.4)	103 (83.7)	5134	179 (96.5)	157 (96.9)	91 (98.2)
dist	13	47660	34527 (27.6)	21761 (54.3)	4704 (90.1)	6672	5698 (14.6)	4700 (29.6)	944 (85.9)	20631	4728 (77.1)	3162 (84.7)	937 (95.5)
plus127mod8192	13	51302	1164 (97.7)	758 (98.5)	61 (99.9)	3717	3717 (0.0)	2290 (38.4)	133 (96.4)	1252	1174 (6.2)	766 (38.8)	61 (95.1)
plus63mod8192	13	30814	1214 (96.1)	785 (97.5)	74 (99.8)	3582	3582 (0.0)	2180 (39.1)	120 (96.6)	2451	1222 (50.1)	792 (67.7)	74 (97.0)
radd	13	17168	11142 (35.1)	8058 (53.1)	1793 (89.6)	632	174 (72.5)	150 (76.3)	78 (87.7)	5135	180 (96.5)	159 (96.9)	116 (97.7)
root	13	59599	36526 (38.7)	22919 (61.5)	5175 (91.3)	6211	3953 (36.4)	3098 (50.1)	797 (87.2)	18513	2521 (86.4)	1686 (90.9)	494 (97.3)
squar5	13	1990	1809 (9.1)	1489 (25.2)	341 (82.9)	262	234 (10.7)	209 (20.2)	97 (63.0)	718	322 (55.2)	263 (63.4)	104 (85.5)
clip	14	113910	75481 (33.7)	43327 (62.0)	8698 (92.4)	8359	4435 (46.9)	3806 (54.5)	898 (89.3)	22501	3378 (85.0)	2461 (89.1)	724 (96.8)
cm42a	14	182407	109307 (40.1)	60840 (66.6)	11454 (93.7)	225	181 (19.6)	175 (22.2)	108 (52.0)	276	276 (0.0)	194 (29.7)	94 (65.9)
cm85a	14	29724	14672 (50.6)	9963 (66.5)	2841 (90.4)	9905	3652 (63.1)	2922 (70.5)	641 (93.5)	13745	546 (96.0)	482 (96.5)	151 (98.9)
sao2	14	103092	74979 (27.3)	44912 (56.4)	8954 (91.3)	26563	12977 (51.1)	9203 (65.4)	1611 (93.9)	9837	5268 (46.4)	2735 (72.2)	666 (93.2)
co14	15	634359	5112 (99.2)	3515 (99.4)	296 (100.0)	674558	1762 (99.7)	1320 (99.8)	164 (100.0)	3820	1762 (53.9)	1320 (65.4)	164 (95.7)
dc2	15	43950	30836 (29.8)	21688 (50.7)	4987 (88.7)	2498	2214 (11.4)	1825 (26.9)	463 (81.5)	2980	1387 (53.5)	1116 (62.6)	308 (89.7)
misex1	15	115663	68605 (40.7)	41933 (63.7)	8346 (92.8)	835	670 (19.8)	580 (30.5)	233 (72.1)	1015	667 (34.3)	527 (48.1)	163 (83.9)
Average			(40.2)	(60.3)	(90.0)		(34.2)	(49.9)	(82.6)		(59.1)	(71.5)	(90.9)

The percentages are the improvements relative to case (a).

TABLE IV
NCV-$|1\rangle$ GATE COUNT FOR A MPMCT GATE WITH c CONTROLS AND a ANCILLARY LINES AS USED IN REVLIB [30].

c	$a = 0$	$1 \leq a < c-2$	$a \geq c-2$
0	1	1	1
1	1	1	1
2	5	5	5
3	13	14	14
4	29	29	26
5	61	52	38
6	125	80	50
7	253	100	62
8	509	128	74
9	1021	152	86
≥ 10	$2^{c+1}-3$	$24c-64$	$12c-22$

the quantum gate optimization described in Section IV-C. The gate count is the number of gates following that optimization.

Case (d) This is the same as Case (c) except the mapping is to the NCV-$|v_1\rangle$ library.

Gate count is, of course, only a rough estimate of the complexity of the circuit particularly as two different quantum gate libraries are used: NCV-$|1\rangle$ for cases (a)-(c) and NCV-$|v_1\rangle$ for case (d). The comparison is a good indicator for the purposes of this paper. A more accurate quantum cost model would be highly technology dependent.

Note that we assume there is always at least one ancillary line available at every gate. Hence if a circuit contains a gate that uses every line as a control or the target, we assume one extra line is added to the circuit. This is required to ensure a circuit can be built from NCV-$|1\rangle$ gates. As a result, higher order roots-of-NOT gates are not used in cases (a) and (b). Only one extra line needs to be added as ancillaries can be reused.

All circuits generated in our experiments have been verified using the QMDD-based verification approach described in [31]. Extensions were required to accommodate the NCV-$|v_1\rangle$ gates and to handle negative controls. However, since the verification is based on a decision diagram approach to representing the required matrices the required changes were easily implemented by simple extensions to the matrix construction procedure.

The results clearly show the advances of the proposed approaches. Using the MPMCT optimization presented in Section IV-A, average reductions in the gate count by approximately 40%, 34%, and 59% can be achieved (see columns labeled with Case (b)). This can be further improved to approximately 60%, 50%, and 72% if the methods from Section IV-B and Section IV-C are applied (see columns labeled with Case (c)). And, if the NCV-$|v_1\rangle$ gate library is considered, the gate count can be decreased by very significant amounts in almost all cases (see columns labeled with Case (d)) where the average improvements are approximately 90%, 83%, and 91%. Thus, we can conclude that particularly the use of NCV-$|v_1\rangle$ gates with the mapping and optimization techniques described in this paper is a viable approach to produce more efficient quantum circuit realizations for reversible functions than can be found using earlier techniques.

Further, although the comparison of the different synthesis methods is not a focus for this paper, some observations can be made. RMS and QMDD-based synthesis generally significantly outperform TBS. Generally QMDD-based synthesis outperforms RMS but the reverse is true for some circuits. Surprisingly, TBS yields the minimum for four cases: cycle10, plus63mod4096, plus127mod8192 and plus63mod8192. Clearly, there is considerable scope for improving synthesis methods. The results show that targeting the NCV-$|v_1\rangle$ library

within the synthesis process might be very profitable. It is important to note that the QMDD-based synthesis method uses the variable order of the function specification whereas the TBS and RMS methods include heuristics that try to make gate choices somewhat independent of the variable order. We anticipate that the QMDD-based results will improve if variable reordering is applied before synthesis. We are exploring that in our ongoing work.

VI. Conclusions

This paper has further explored the use of the NCV-$|v_1\rangle$ gates introduced in [15]. The approach has been extended to negative controls and an ESOP-based optimization for mixed control circuits has been introduced. Experimental results were presented to show the effectiveness of the mapping and optimization approaches presented. It was demonstrated that the advantages apply to circuits generated by different synthesis methods.

We have not used template matching [19] in this work as we have focused on demonstrating the effectiveness of the approaches introduced in this paper. We plan to both compare the effectiveness of template matching to the new approaches and to also how the techniques can be used together to further optimize circuits.

The structure illustrated in Fig. 3 satisfies the linear *nearest-neighbor property* [32], [33], i.e. the control and target for each controlled gate are on adjacent lines. This is not the case for NCV-$|1\rangle$ realizations of MPMCT gates. We thus anticipate it will be easier to produce fully nearest-neighbor circuits for reversible functions using the NCV-$|v_1\rangle$ library. This is a major focus of our ongoing work.

Acknowledgement

This work was supported in part by grants from the Natural Sciences and Engineering Research Council of Canada (NSERC), the German Research Foundation (DFG) (DR 287/20-1), and the German Academic Exchange Service (DAAD).

References

[1] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*. Cambridge Univ. Press, 2000.

[2] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Phys. Rev. Lett.*, vol. 79, pp. 325–328, Jul 1997. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.79.325

[3] P. W. Shor, "Algorithms for Quantum Computation: Discrete Logarithms and Factoring," in *Symp. on Foundations of Computer Science*, Nov. 1994, pp. 124–134.

[4] M. Saeedi, N. MohammadZadeh, M. Sedighi, and M. S. Zamani, "Evaluation and improvement of quantum synthesis algorithms based on a thorough set of metrics," in *Proc. European Conf. on Digital Systems Design*, 2008, pp. 490–493.

[5] M. Perkowski, N. Alhagi, M. Lukac, N. Saxena, and S. Blakely, "Synthesis of small reversible and pseudo-reversible circuits using y-gates and inverse y-gates," in *Proc. Int'l Symp. on Multiple-valued Logic*, 2010, pp. 245–251.

[6] N. Alhagi, M. Hawash, and M. Perkowski, "Synthesis of reversible circuits with no ancilla bits for large reversible functions specified with bit equations," in *Proc. Int'l Symp. on Multiple-valued Logic*, 2010, pp. 39–45.

[7] M. Saeedi and I. L. Markov, "Synthesis and optimization of reversible circuits - a survey," *arXiv*, vol. abs/1110.2574, 2011.

[8] M. Lucas, M. Perkowski, and M. Kameyama, "Evolutionary quantum logic synthesis of boolean reversible logic circuits embedded in ternary quantum space using structural restrictions," in *Proc. Int'l Conf. on Evolutionary Computation*, 2010, pp. 1–8.

[9] R. Wille and R. Drechsler. Morgan and Claypool, 2010, ch. Synthesis of Boolean Functions in Reversible Logic.

[10] D. M. Miller and Z. Sasanian, "Lowering the quantum gate cost of reversible circuits," in *Proc. Midwest Symp. on Circuits and Systems*, 2010, pp. 260–263.

[11] A. Barenco, C. Bennett, R. Cleve, D. DiVincenzo, M. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter, "Elementary gates for quantum computation," *Physical Review A*, vol. 52, no. 5, pp. 3457–3467, Nov. 1995.

[12] D. M. Miller, R. Wille, and Z. Sasanian, "Elementary quantum gate realizations for multiple-control Toffolli gates," in *Proc. Int'l Symp. on Multiple-valued Logic*, 2011, pp. 217–222.

[13] Z. Sasanian and D. M. Miller, "NCV realization of MCT gates with mixed controls," in *Proc. Pacific Rim Conf. on Communications, Computers and Signal Processing*, 2011, pp. 567–571.

[14] D. Maslov and M. Saeedi, "Reversible circuit optimization via leaving the boolean domain," *IEEE Trans. on CAD*, vol. 30, pp. 806–816, 2011.

[15] Z. Sasanian, R. Wille, and D. M. Miller, "Realizing reversible circuits using a new class of quantum gates," in *Proc. Design Automation Conf.*, June 2012.

[16] T. Toffoli, "Reversible computing," Tech Memo LCS/TM-151, MIT Lab for Comp. Sci, 1980.

[17] T. Sasao, "AND-EXOR expressions and their optimization," in *Logic Synthesis and Optimization*, T. Sasao, Ed. Kluwer Academic Publisher, 1993, pp. 287–312.

[18] A. Muthukrishnan and C. R. Stroud, "Multivalued logic gates for quantum computation," *Physical Review A*, vol. 62, p. 052309, 2000.

[19] D. M. Miller, D. Maslov, and G. W. Dueck, "A transformation based algorithm for reversible logic synthesis," in *Proc. Design Automation Conf.*, 2003, pp. 318–323.

[20] D. Maslov, G. W. Dueck, and D. M. Miller, "Techniques for the synthesis of reversible Toffoli networks," *ACM Trans. Design Autom. Electr. Syst.*, vol. 12, no. 4, pp. 42:1–42:28, Sep. 2007.

[21] D. Miller and M. Thornton, "QMDD: A decision diagram structure for reversible and quantum circuits," in *Proc. Int'l Symp. on Multiple-valued Logic*, 2006, pp. 30–30.

[22] M. Soeken, R. Wille, C. Hilken, N. Przigoda, and R. Drechsler, "Synthesis of Reversible Circuits with Minimal Lines for Large Functions," in *Asia and South Pacific Design Automation Conference*, Jan. 2012.

[23] A. Mishchenko and M. Perkowski, "Fast heuristic minimization of exclusive sum-of-products," in *Proc. 5th Int'l Reed-Muller Workshop*, 2001, pp. 242–250.

[24] D. Maslov, G. Dueck, D. Miller, and C. Negrevergne, "Quantum circuit simplification and level compaction," *IEEE Trans. on CAD*, vol. 27, no. 3, pp. 436–444, March 2008.

[25] Z. Sasanian and D. M. Miller, "Mapping a multiple-control toffoli gate cascade to an elementary quantum gate circuit," in *Proc. Workshop on Reversible Computation*, 2010, pp. 83–90.

[26] D. Maslov, "Reversible logic synthesis," Ph.D. dissertation, University of New Brunswick, 2003.

[27] Y. Wang and M. Perkowski, "Improved complexity of quantum oracles for ternary grover algorithm for graph coloring," in *Proc. Int'l Symp. on Multiple-valued Logic*, 2011, pp. 294 – 301.

[28] Z. Sasanian and D. M. Miller, "Transforming MCT circuits to NCVW circuits," in *Proc. Workshop on Reversible Computation*, 2011, pp. 163–174.

[29] D. M. Miller, D. Maslov, and G. W. Dueck, "A transformation based algorithm for reversible logic synthesis," in *Design Automation Conference*. ACM, Jun. 2003, pp. 318–323.

[30] R. Wille, D. Große, L. Teuber, G. W. Dueck, and R. Drechsler, "RevLib: An online resource for reversible functions and reversible circuits," in *Int'l Symp. on Multi-Valued Logic*, 2008, pp. 220–225, RevLib is available at www.revlib.org.

[31] R. Wille, D. Große, D. M. Miller, and R. Drechsler, "Equivalence checking of reversible circuits," in *Proc. Int'l Symp. on Multiple-valued Logic*, 2009, pp. 324 – 330.

[32] Y. Hirata, M. Nakanishi, S. Yamashita, and Y. Nakashima, "An efficient method to convert arbitrary quantum circuits to ones on a linear nearest neighbor architecture," in *Proc. Int'l Conf. on Quantum, Nano and Micro Technologies*, 2009, pp. 26 – 33.

[33] M. Saeedi, R. Wille, and R. Drechsler, "Synthesis of quantum circuits for linear nearest neighbor architectures," *Quantum Information Processing*, 2011.