Synthesis of Linear Nearest Neighbor Quantum Circuits

Md. Mazder Rahman & Gerhard W. Dueck
University of New Brunswick
Canada
Mazder.Rahman@unb.ca & gdueck@unb.ca

Presented at the 10th International Workshop on Boolean Problems (2012), Freiberg, Germany.

Abstract

This paper presents models for transforming standard reversible circuits into Linear Nearest Neighbor (LNN) architecture without inserting SWAP gates. Templates to optimize the transformed LNN circuits are proposed. All minimal LNN circuits for all 3-qubit functions have been generated to serve as benchmarks to evaluate heuristic optimization algorithms. The minimal results generated are compared with optimized LNN circuits obtained from the post synthesis algorithm — template matching with LNN templates. Experiments show that the suggested synthesis flow significantly improves the quantum cost of circuits.

1 Introduction

For the last decades, significant research on synthesizing quantum circuits has been done. Most synthesis approaches ignore physical constrains, i.e. operation may be applied to qubits that are distant in physical space [1]. However, some technologies such as one dimensional Ion Trap only support the Linear Nearest Neighbor (LNN) architecture of circuits in which the control and target of a gate must be adjacent. Therefore, the synthesis of LNN circuits is of interest. Toffoli networks can be transformed into LNN quantum circuits by using the standard decomposition of multiple-control Toffoli (MCT) circuits [2] and further inserting SWAP gates [3] or appropriate SWAP sequence [4] whenever a gate with non-adjacent control and target occurs. The obtained circuits are optimized by post synthesis methods. One such method is template matching with SWAP templates proposed in [3]. In this paper, we identify efficient ways of transforming standard MCT circuits into LNN architecture.

2 Background

A Boolean logic function $f : B^n \rightarrow B^n$ is said to be reversible if there is a one-to-one and onto mapping between input vectors and output vectors. A reversible function can be embedded into a reversible circuit by cascading the reversible gates without allowing feedback and fanout to preserve the reversibility. A generalized multiple-control Toffoli gate is defined as $T_n(C,t)$ based on number of lines $0 < n$, which maps the pattern $(x_{i_1}, x_{i_2}, ..., x_{i_k})$ to $(x_{i_1}, x_{i_2}, ..., x_{j-1}, x_j \oplus x_{i_1} x_{i_2} ... x_{j-1} x_{j+1} ... x_{i_k}, x_{i_1}, x_{i_2} x_{i_3} x_{i_4} ... x_{i_k})$, where $C = \{x_{i_1}, x_{i_2}, ..., x_{i_k}\}$, $t = \{x_j\}$ and $C \cap t = \phi$. $C$ is referred to as the control set and $t$ is referred to as the target. $T_1$ and $T_2$ are referred to as NOT and CNOT respectively. A picture of a $T_3$ gate is shown in Figure 2(d).

The Controlled-V gate has two lines (control and target), the target line changes using the transformation defined by the matrix $V = \frac{i+1}{2} \left( \begin{array}{cc} 1 & -i \\ 1 & 1 \end{array} \right)$ if the control line has the value 1. Similarly, the Controlled-V$^\dagger$ gate has two lines (control and target), the target line changes using...
the transformation defined by the matrix \( V^\dagger = V^{-1} = \frac{i-1}{2} \left( \begin{array}{c} 1 \\ i \\ i \\ 1 \end{array} \right) \) if the control line has the value 1. The SWAP\((x, y)\) gate maps the input \((x, y)\) to \((y, x)\).

Logic operations in quantum computation are quite different from those in classical logic. The fundamental unit of information in quantum computation is a qubit represented by a state vector. A qubit has a state either \( |0\rangle \) or \( |1\rangle \); these are known as computational basis states. An arbitrary qubit is described by the following state vector

\[
|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \left( \begin{array}{c} \alpha \\ \beta \end{array} \right)
\]

(1)

where \( \alpha \) and \( \beta \) are complex numbers that satisfy the constraint \( |\alpha|^2 + |\beta|^2 = 1 \). The measurement of a qubit results either 0 with probability \( |\alpha|^2 \), that is, the state \( |0\rangle = \left( \begin{array}{c} 1 \\ 0 \end{array} \right) \) or 1 with probability \( |\beta|^2 \), that is, the state \( |1\rangle = \left( \begin{array}{c} 0 \\ 1 \end{array} \right) \). On the other hand, a classical bit has a state either 0 or 1 which is analogous to the measurement of a qubit state either \( |0\rangle \) or \( |1\rangle \) respectively. The fundamental difference between bits and qubits is that a bit can be either state 0 or 1 whereas a qubit can be a state rather than \( |0\rangle \) or \( |1\rangle \). A two qubit system has four computational basis states \( |00\rangle, |01\rangle, |10\rangle \) and \( |11\rangle \) can be represented by the state vector

\[
|\phi\rangle = \lambda_1|00\rangle + \lambda_2|01\rangle + \lambda_3|10\rangle + \lambda_4|11\rangle = \left( \begin{array}{c} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{array} \right)
\]

(2)

where \( \lambda_1 \lambda_4 = \lambda_2 \lambda_3 \). If \( \lambda_1 \lambda_4 \neq \lambda_2 \lambda_3 \) then the state \( |\psi\rangle \) is referred to as an entangled state which is not separable as the tensor product of two single qubits. The elementary quantum gates \( \text{NOT}, \text{CNOT}, \text{Controlled-V} \) and \( \text{Controlled-V}^\dagger \) are also known as quantum primitives have been widely used to synthesis of binary reversible functions. A quantum circuit is realized by the cascades of quantum primitives. The quantum cost of a reversible circuit is defined by the number of quantum gates required to realized the circuit. To perform the logic operations in quantum circuits, two more qubit states \( |v_0\rangle \) and \( |v_1\rangle \) rather than \( |0\rangle, |1\rangle \), are possible at the intermediate position in the circuits where \( |v_0\rangle = \frac{1+i}{2} \left( \begin{array}{c} 1 \\ i \end{array} \right) \) and \( |v_1\rangle = \frac{1-i}{2} \left( \begin{array}{c} 1 \\ -i \end{array} \right) \). However, if the state vector \( |v_0\rangle \) or \( |v_1\rangle \) is applied to the control of a two-qubit gate, then the resulting output vector results in an entangled state \( |5\rangle \). If a quantum circuit is obtained from the quantum decomposition of a MCT circuit, the entangled state does not arise.

**Definition 1** If a quantum circuit generates an entangled state for any given binary input state is said to be an entangled circuit.

**Example 1** The cascades of quantum primitives shown in Figure 1 is an entangled circuit because the circuit generates an entangled state for input vector \( |1,1,1\rangle \) and the resulting outputs are not separable into 3 single-qubit states.

A quantum circuit that contains gates which are not necessarily acting on the adjacent qubits, is referred to as a standard quantum circuit. A Linear Nearest Neighbor (LNN) quantum circuit is defined as follows:

**Definition 2** A quantum circuit \( C \) is said to be a LNN circuit if all gates are acting on adjacent qubits.

**Definition 3** The cost of a circuit \( C \) is defined as the number of its gates and denoted by \( |C| \). For a given function \( f \), a circuit \( C \) is said to be optimal if there is no realization of \( f \) with lower cost.
Figure 2: (a) symbol of SWAP gate, (b) and (c) quantum realization of SWAP gate, (d) $T_3$ and (e) LNN implementation of $T_3$.

Figure 3: (a) $T_3$ with non-adjacent controls, (b) optimal quantum realization of (a) and (c) LNN implementation of (b) with cost 13.

The best reported LNN realization of the $T_3$ gate has quantum cost 9. However, different LNN realizations of $T_3$ with cost 9 are possible by not only replacing Controlled-V (Controlled-$V^\dagger$) with Controlled-$V^\dagger$ (Controlled-V) but also by using the two different realizations of the SWAP gate as shown in Figure 2(b) and (c). The synthesis flow for the generation of LNN circuits is done in 3 steps: i) decomposition of a MCT circuit into a quantum circuit, ii) transformation of the resulting gates into LNN architecture by inserting SWAP gates or appropriate SWAP sequences and iii) optimization of the LNN circuits with post-synthesis methods [3, 1, 4]. In this straightforward implementation, the resulting LNN circuits might be entangled realizations and suboptimal. For example, the circuit shown in Figure 3(b) is an optimal standard quantum realization of the circuit shown in Figure 3(a). By inserting SWAP gates to move the control of both CNOT towards the target results in a LNN circuit with quantum cost 17. The insertion of appropriate SWAP sequences results in a circuit with quantum cost 13 as shown in Figure 3(c). However, the circuit is an entangled circuit and we ignore such type of realization. Moreover, for the MCT circuit as shown in Figure 4(a), the optimization method proposed in [3] results in a LNN circuit with quantum cost 24 (Figure 4(b)). By replacing the SWAP gates with appropriate SWAP sequences as proposed in [4] the circuit with cost 18 as shown in Figure 4(c) is obtained. This circuit is not minimal.

3 Transformation of MCT Circuits into LNN Circuits

In this section, we propose methods for transforming MCT circuits into LNN architecture by using three different models to move the control (target) of a 2-qubit quantum gate towards the target (control) until they become adjacent. This approach always results in non-entangled LNN circuits with considerably lower quantum cost than previously proposed methods.

Figure 4: (a) A MCT circuit, (b) its LNN implementation according to [3] and (c) as proposed in [4].
3.1 LNN Transformation of 2-qubit Quantum Gates

For a standard quantum circuit, the Nearest Neighbor Cost (NNC) of a 2-qubit quantum gate \( g \), where its control and target are placed at the \( c^{th} \) and \( t^{th} \) line respectively, is defined as \( |c - t| - 1 \), i.e. the distance between control and target lines [3]. The \( CNOT \) gate with \( NNC = 1 \) as shown in Figure 5(a) has three different LNN implementations as shown in Figure 5(b), (c) and (d) that we refer to as Model-1, Model-2, and Model-3 respectively. Clearly, fewer gates are needed in each LNN implementation than with SWAP gates. This model can be generalized for \( NNC = k \) as follows:

**Model-1 (Control moves towards target):** A \( CNOT \) gate with \( NNC = k \) in \( n \)-qubit circuit \( 1 \leq k < n - 1 \) can be transformed into a LNN architecture with quantum cost \( 4k \) by using this model whereas it requires \( 6(k + 1) \) quantum gates if SWAP gates are used. For instance, the \( CNOT \) with \( NNC = 4 \) and its LNN transformation to move the control towards the target as shown in Figure 5(b) has three different LNN implementations as shown in Figure 5(b), (c) and (d). The 2\(^{nd} \) and 4\(^{th} \) \( CNOT \) gates in Figure 5(b) are replaced with their reverse implementation of each other by using Model-1. The resulting circuit is shown in Figure 5(d). This process is iterated until no \( CNOT \) gates with \( NCC > 0 \) remain. The final circuit is shown Figure 5(d).

**Model-2 (Control moves towards target):** A \( CNOT \) with \( NNC = k \) in \( n \)-qubit circuit \( 1 \leq k < n - 1 \) can be transformed into a LNN architecture with quantum cost \( 4(k + 1) \) by using Model-2. For instance, the \( CNOT \) with \( NNC = 4 \) can be transformed to a LNN circuit by iteratively moving the control towards the target as shown in Figure 6(e) and (f). This model can also be used for transforming Controlled-V and Controlled-V\(^\dagger\) gates with non-adjacent control and target lines.

**Model-3 (Target moves towards control):** This model can be used to move the target to the control of a \( CNOT \) with \( NNC = k \). This transformation requires \( 4(k + 1) \) gates.

In summary, Controlled-V or Controlled-V\(^\dagger\) with non-adjacent control and target can only
be transformed by using Model-2. Model-2 and Model-3 can be used to move controls (target) towards the target (controls) of a MCT gate. Model-1 enables the move of the control towards the target in CNOT gates.

### 3.2 LNN Transformation of Toffoli Gates

$T_3$ gates with non adjacent controls and target can be transformed into MCT circuits where all gates have adjacent controls. Two different cases can be considered.

Let $p, q$ be the number of free lines in between the controls $C = \{c_1, c_2\}$ ($c_1 < c_2$) and the target $t$ of $T_3(C, t)$, then the following 2 cases are possible.

**Case 1:** If $c_1 = i, c_2 = i + q + 1$ and $t = c_1 - p - 1$ or $t = c_2 + p + 1$ and $0 \leq p, q$ then the control $c_1$ can move towards the $c_2$ and the target $t$ can move towards the control $c_2$ by using $4(p + q)$ gates or the control $c_2$ can move towards the $c_1$ and the target $t$ can move towards the control $c_1$ by using $4(p + q)$ gates results in a LNN circuit with $4(p + q) + 9$ gates. When $q = 0$, the controls are adjacent, for instance the [7(b)] shows the form of transformation $T_3$ with 6 lines when $q = 0$ and $p = 3$. The replacement of $T_3$ with its LNN circuit results in a LNN architecture of [7(a)].

**Case 2:** If $c_1 = t - p - 1, c_2 = t + q + 1$, $0 \leq p, q$ then the the controls can move towards the target by using $4(p + q)$ gates. When $p = q = 1$, $T_3$ is the form as shown in Figure 7(e). Two controls can move towards the target as shown in Figure 7(d) and (e) successively. When $p = 0$ and $q = 0$ the $T_3$ as the form shown in Figure 7(f). Further, the two the controls can be adjacent as the form shown in Figure 7(g) or (h) by using 4 gates. Therefore, the final LNN circuit requires $4(p + q + 1) + 9$ gates when $0 < p, q$. By replacing $T_3$ in circuits 7(g) and (h) with its LNN implementation results in LNN architectures with 13 gates. Moreover, the resulting LNN circuit of 7(f) would be non-entangled whereas the previously published approach of LNN transformation gives entangled circuit in this case. However, if the $T_3$ in MCT circuits is either one of the form $T_3(c_1, c_2, t)$ or $T_3(t, c_1, c_2)$ before quantum decomposition of circuits then the synthesis flow of LNN circuits ensures the non-entangled LNN circuit as a result.

### 4 Optimization of LNN Circuits with LNN Templates

LNN circuits obtained from the proposed transformation of MCT circuits are most likely not minimal even if an optimal standard quantum circuit is transformed into an LNN architecture. For instance, by using the models proposed in Section 3.1, the three different LNN implementations shown in Figure 8(b), (c), and (d) of the optimal standard quantum circuit shown in Figure 8(a). However, none of these implementations are minimal.

The idea of post synthesis optimization – template matching – for simplifying standard MCT circuits originated in [7] and later on extensive studies have been done by introducing reconfigured templates [7], developing an algorithm to find templates [8] as well as modifying the definition of template and analyzing their properties [9]. Template matching has been extended to optimize LNN circuits based on templates that are comprised of SWAP gates [3]. In this
In this section, we propose LNN templates that can be used in template matching to optimize LNN circuits. This new approach outperforms the previously proposed approaches. With LNN, it is necessary to have quantum templates that ensure the resulting optimized circuit does not violate the constraint of LNN quantum circuits when a template is applied. Therefore, we first present the formal definition of LNN templates. The properties of templates proposed in [9] hold for these templates as well.

**Definition 4** A LNN quantum template is an LNN identity circuit with $d$ gates, such that at least one sequence of \( \lceil \frac{d}{2} \rceil + 1 \) gates in the circuit cannot be reduced by any other LNN template.

Clearly, all two-qubit templates as well as all templates proposed in [9] for which the LNN constraint holds, must be the LNN templates. The significance of proposed LNN templates shown in Figure 9 is illustrated with the subsequent examples.

**Example 2** The gate sequence in the LNN circuits shown in Figure 8(b) and (c) match with the templates in Figure 9(d) and (b). Template matching results in an optimized circuit as shown in Figure 8(e). These small circuits cannot be optimized by previously proposed methods.

**Example 3** Consider the circuit in Figure 10(a) reported in [3]. According to our proposed approach, the LNN transformation and optimization are done by the steps: 1) move targets towards the controls by using Model-3, 2) replace $T_3$ with its LNN circuit, 3) apply gate deletion rules, 4) apply template 9(d), and 5) apply gate merge rules [8]. The resulting optimized circuit is shown in Figure 10(b). The number of quantum gates in the optimized circuit is 13. The cost of the solution proposed in [3] is almost 50% higher (see Figure 4(b)). However, the proposed templates in [3] are derived from SWAP gates, therefore, the resulting LNN circuit is still contains SWAP gates. The optimization by choosing appropriate SWAP sequence proposed in [3] results in a circuit with cost 18 as shown in Figure 4(c). However, the gate sequence from index 4 (starting at 0) to

Figure 8: LNN transformations of (a) using model-1, (c) using model-2, (d) using model-3, and (e) optimized circuit.

Figure 9: LNN quantum templates.

Figure 10: (a) MCT circuit, (b) Optimized LNN circuit of (a) and (c) Optimized circuit in Figure 4(c).
and further reconfiguring $16^{th}$ of the template as shown in Figure 9(e) matches with the gate sequence $0, 1, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13$ in circuit in Figure 4(c). Therefore, template matching results in a circuit with cost 13 as shown in Figure 10(c).

5 3-Qubit Optimal LNN Circuits

In general, the direct synthesis of quantum circuits for a given reversible function specification is intractable. However, for 3-qubit functions, all optimal standard quantum circuits have been obtained by directly cascading the quantum primitives [10]. Therefore, a similar method can be used to find all optimal LNN circuits of 3 qubits.

**Definition 5** Given a library of gates $L$, a LNN circuit $c$ with $n$ gates that realizes the function $f$, is said to be optimal with respect to $L$, if no LNN realization of $f$ exists that has fewer than $n$ gates.

Let $C_n$ be the set of all optimal circuits with $n$ gates. In constructing LNN circuits, we use the 15 permuted quantum gates with 3 qubits whose control and target are acting on the adjacent qubits. An exhaustive search method has been used to find all LNN quantum circuits $C_n$ by cascading the optimal LNN quantum circuits from the sets $C_{n-1}$ and $C_1$. For all 3-qubit binary functions, the results of optimal LNN quantum circuits are shown in column II in Table 1.

6 Synthesis Flow of LNN circuit

LNN decomposition of Higher-Order Toffoli gates has been studied in [4] in which the minimized standard quantum circuit of Higher-Order Toffoli is transformed into a LNN circuit by inserting appropriate SWAP gates. However, it is evident that the insertion of SWAP gates into optimal standard quantum circuit results LNN circuits that can still be optimized. We investigate the minimal way of transforming Higher-Order Toffoli gate into LNN architecture in which optimization is to be done at the end of the process. The synthesis flow of LNN circuit is shown in Algorithm 1.

**Algorithm 1** Synthesis flow LNN circuit

1) Decompose Higher-Order Toffoli in a MCT circuit into $T_3$ gates according to [2].
2) Transform all $T_3$ gates with non-adjacent controls and target by using Model-2 and Model-3 results in a circuit of all Toffoli-3 with adjacent controls and target.
3) Replace all $T_3$ with its LNN architecture results in a non-minimal LNN circuit.
4) Optimize the circuit obtained in step 3 by using LNN quantum templates.

According to [2], to transform a Higher-Order Toffoli gate into a circuit with $T_3$ requires at least one extra line, however, the decomposition by using more lines results in a circuit with the less number of $T_3$. We observed that the above synthesis flow gives better results if more working lines are used in decomposition. We achieve the LNN circuit as shown in Figure 11(b) for $T_4$ with one working line.
7 Experimental Results

The proposed synthesis flow of LNN circuits has been implemented in C/C++ on top of RevKit-1.2.1 [11]. To evaluate the effectiveness of the new approach, we have taken all 3-qubit minimal MCT circuits and transformed them into LNN circuits by using different approaches as shown in columns III, IV, V, and VI of Table I. The proposed transformation approach results the average number of gates 27.1 compared to the optimal of 15.9. It can be seen that the new transformation method results in smaller circuits. The results of optimized LNN circuits in column Optz(M) are obtained by template matching using 21 LNN templates. The results show that 41% gate reduction is required on average to reach the optimal LNN circuits shown in column M. However, we gain an approximate 19% reduction and a further 27% reduction is needed for the optimal result.

8 Conclusion

We propose a new synthesis flow for LNN quantum circuits in which the transformation models result in circuits with considerable lower quantum cost compared to others methods. Moreover, the template matching with new LNN templates significantly reduces the number of gates in circuits. In some cases, the reduction is more than 50%. The effectiveness of our approach is evident in the examples.

References

[1] M. Perkowski, M. Lukac, D. Shah, and M. Kameyama, “Synthesis of quantum circuits in linear nearest neighbor model using positive Davio lattices,” Facta universitatis - series: Electronics and Energetics, vol. 4, no. 1, pp. 73–89, 2011.
[2] A. Barenco, C. H. Bennett, R. Cleve, D. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter, “Elementary gates for quantum computation,” The American Physical Society, vol. 52, pp. 3457–3467, 1995.

[3] M. Saeedi, R. Wille, and R. Drechsler, “Synthesis of quantum circuits for linear nearest neighbor architectures,” Quantum Information Processing, vol. 10, no. 3, pp. 73–89, 2011.

[4] D. M. Miller, R. Wille, and Z. Sasanian, “Elementary quantum gate realizations for multiple-control Toffoli gates,” in Proceedings of the International Symposium on Multiple-Valued Logic, 2011, pp. 288–293.

[5] W. Hung, X. Song, G. Yang, J. Yang, and M. Perkowski, “Optimal synthesis of multiple output Boolean functions using a set of quantum gates by symbolic reachability analysis,” Transactions on Computer Aided Design, vol. 25, no. 9, pp. 1652–1663, 2006.

[6] D. M. Miller, D. Maslov, and G. W. Dueck, “A transformation based algorithm for reversible logic synthesis,” in Design Automation Conference, June 2003.

[7] M. M. Rahman, G. W. Dueck, and A. Banerjee, “Optimization of reversible circuits using reconfigured templates,” in 3rd Workshop on Reversible Computation, July 2011, pp. 143–154.

[8] M. M. Rahman and G. W. Dueck, “An algorithm to find quantum templates,” in IEEE Congress on Evolutionary Computation, 2012, pp. 623–629.

[9] ——, “Properties of quantum templates,” in 4th Workshop on Reversible Computation, July 2012, p. Accepted.

[10] ——, “Optimal quantum circuits of 3-qubits,” in Proceedings of the International Symposium on Multiple-Valued Logic, 2012, pp. 161–166.

[11] M. Soeken, S. Frehse, R. Wille, and R. Drechsler, “RevKit: a toolkit for reversible circuit design,” in Proceedings of the International Symposium on Multiple-Valued Logic, 2008, RevKit is available at http://www.revkit.org.