Quantum Circuit Simplification Using Templates

D. Maslov, C. Young, D. M. Miller
Department of Computer Science
University of Victoria
Victoria, BC, V8W 3P6, Canada
dmaslov, youngce, mmiller@uvic.ca

G. W. Dueck
Faculty of Computer Science
University of New Brunswick
Fredericton, NB, E3B 5A3, Canada
gdueck@unb.ca

Abstract

Optimal synthesis of quantum circuits is intractable and heuristic methods must be employed. Templates are a general approach to reversible and quantum circuit simplification. In this paper, we consider the use of templates to simplify a quantum circuit initially found by other means. We present and analyze templates in the general case, and then provide particular details for circuits composed of NOT, CNOT and controlled-$sqrt$-of-NOT gates. We introduce templates for this set of gates and apply them to simplify both known quantum realizations of Toffoli gates and circuits found by earlier heuristic Fredkin and Toffoli gate synthesis algorithms. While the number of templates is quite small, the reduction in quantum cost is often significant.

1. Introduction

Research in quantum circuit synthesis is motivated by the growing interest in quantum computation [16] and advances in quantum circuit technology [1, 2]. Even for circuits with only a few variables, it is at present intractable to find an optimal circuit. Thus a number of heuristic synthesis methods have emerged. Application of these methods usually results in a non-optimal circuit specification, which can be later simplified using local optimization techniques.

Local optimization has only recently been considered as a possible tool for the simplification of quantum [9, 21] and reversible circuits [7]. While those works provide several rewriting rules, there is clearly a benefit to systemizing the approach through a method such as the templates discussed in this paper. A somewhat different approach for local optimization of reversible NOT-CNOT-Toffoli circuits was applied for the simplification of random reversible circuits in [18]. That approach and our template method are very different in that they have different metrics for the circuit cost, use different types of gates, and are applied to different types of circuits. This makes it difficult to compare the methods. However, we believe that our method is more general and will be more scalable.

Templates have been considered for Toffoli [13] and Toffoli-Fredkin [12] reversible network simplification. In this paper, we revisit the definition of templates and show how they can be applied in the quantum case.

2. Background

We present a short review of the basic concepts of quantum computation necessary for this paper. For a more detailed introduction, please see [16].

The state of a single qubit is a linear combination $\alpha|0\rangle + \beta|1\rangle$ (also written as a vector (α, β)) in the basis $\{|0\rangle,\ |1\rangle\}$, where α and β are complex numbers called the amplitudes, and $|\alpha|^2 + |\beta|^2 = 1$. Real numbers $|\alpha|^2$ and $|\beta|^2$ represent the probabilities p and q of reading the 'pure' logic states $|0\rangle$ and $|1\rangle$ upon measurement. The state of a quantum system with $n > 1$ qubits is given by an element of the tensor product of the single state spaces and can be represented as a normalized vector of length 2^n, called the state vector. Quantum system evolution allows changes of the state vector through multiplication by the appropriate $2^n \times 2^n$ unitary matrices.

The above models how a transformation can be performed, but does not indicate how to identify the unitary matrices that compose the transformation or how to implement them. Typically, certain primitive gates are used as elementary building blocks with an assumed unit cost [4, 6, 16]. Among these are:

- NOT ($x \mapsto \bar{x}$) and CNOT ($(x, y) \mapsto (x, x \oplus y)$) gates;
- Hadamard gate defined by $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$;
- controlled-V gate that depending on the value on its control line changes the value on the target line using the transformation given by the matrix $\mathbf{V} = \frac{i+1}{2}\begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}$;
- controlled-V^+ that depending on the value of its control line changes the value on the target line using the transformation $\mathbf{V}^+ = \mathbf{V}^{-1}$;
- rotation gates $R(\gamma)$, $\gamma \in [0, 2\pi]$.

The controlled V and V^+ gates can be seen to be controlled-$sqrt$-of-NOT gates since $V^2 = (V^+)^2 = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ which is the transformation matrix for NOT. Note that once a gate is available, its inverse is also available with the same cost. This observation will be useful when considering the templates and their application to quantum circuit simplification.

The Toffoli gate [20] and its generalization with more than two controls serve as a good basis for synthesis purposes as shown by several authors [3, 8, 14]. The generalized Toffoli gate, which for simplicity we refer to simply as a Toffoli gate, is defined as follows:

Definition 1. For the set of Boolean variables $\{x_1, x_2, ..., x_n\}$ the **Toffoli gate** has the form $TOF(C;T)$, where $C = \{x_{i_1}, x_{i_2}, ..., x_{i_k}\}$, $T = \{x_j\}$ and $C \cap T = \emptyset$. It maps each pattern $(x_1^+, x_2^+, ..., x_n^+)$ to $(x_1^+, x_2^+, ..., x_{j-1}^+, x_j^+ \oplus x_{i_1}^+ x_{i_2}^+ ... x_{i_k}^+, x_{j+1}^+, ..., x_n^+)$. C is called the **control** set and T is called the **target**.

Unfortunately, Toffoli gates are not simple transformations in quantum technology. Rather they require a number of quantum gates and Toffoli gates with a large number of controls can be quite costly [4, 10]. In this paper, we apply templates to the best known quantum circuits implementing large Toffoli gates resulting in significant simplification. This is very useful since it allows simpler quantum realization of the circuits reported by authors who have used Toffoli gates with large control sets.

Definition 2. An n-input, n-output Boolean function is **reversible** if it maps each of the 2^n input patterns to a unique output pattern.

Clearly, a reversible function f has an inverse (f^{-1}). Each of the gates noted above implements a reversible function and thus there is a corresponding inverse gate. NOT, CNOT, and Toffoli gates are self-inverse. V and V^+ are the inverses of each other. The situation for rotation gates is more complex and is not considered here as we have not employed rotation gates in our work so far. Quantum circuits of necessity implement reversible functions.

We shall write G^{-1} to denote the gate implementing the inverse function of the function realized by gate G. In context, we will use G to mean a gate or the transformation matrix for that gate. The symbol $\oplus$ will be used in diagrams to denote a NOT gate.

3. Templates: definition

A **Rewriting rule** is a procedure that takes two equivalent (computing the same function) circuits and replaces one with the other. If the cost of the replacement circuit is less than the cost of the replaced circuit, this leads to a circuit cost reduction. Templates are a generalization of this idea.

Often, the cost of a circuit is defined simply as a, possibly weighted, gate count. We term this a **linear cost metric**. In the case of a **non-linear cost metric** (such as those used in [6, 19]), it is necessary to consider the cost of the entire circuit as the overall cost can be affected by a local optimization. An example of such a situation arises when considering Peres gates [17] (defined by the transformation $(a, b, c) \mapsto (a, b \oplus a, c \oplus ab)$) which, when implemented by a quantum Toffoli gate $TOF(a, b; c)$ followed by a CNOT gate $TOF(a; b)$ would have a cost of $5 + 1 = 6$, whereas a Peres gate can be directly constructed with 4 elementary quantum blocks.

We call a rewriting rule **regular** if the replacement circuit has smaller cost than the replaced circuit, otherwise we call it **irregular**. The idea of applying regular rewriting rules to simplify sub-circuits of a given circuit has two phases. First, find as many regular rewriting rules as possible, and second, apply them to reduce the cost of a given circuit. Direct application of such an approach to quantum circuit cost reduction can be found in [9]. However, we see the following potential problems with that particular approach:

- The number of regular rewriting rules is very large even for small parameters. For instance, in the case of the quantum gates listed above, the number of regular rewriting rules is infinite if no proper classification is presented and rotation gates with any parameter γ are allowed.
- Often, rewriting rules are redundant in the sense that a $G_1G_2G_3 \rightarrow G_4G_5$ rewriting rule is a derivative of $G_2G_3 \rightarrow G_5$ rewriting rule if $G_1 = G_4$.
- It often happens that interchanging the order of the gates, which sometimes is possible and itself does not change the cost of a circuit, may result in allowing a rewriting rule application that decreases the cost of the circuit.

We employ templates as a means to address the above problems while maintaining the advantages of rewriting rules. The following observations are beneficial to the understanding of templates:

Observation 1. For any circuit $G_0G_1... \ G_{m-1}$ realizing a reversible function f, circuit $G_{m-1}^{-1}G_{m-2}^{-1}... \ G_0^{-1}$ is a realization for f^{-1}.

Observation 2. For any rewriting rule $G_1G_2... \ G_k \rightarrow G_{k+1}G_{k+2}... \ G_{k+s}$ its gates satisfy the following: $G_1G_2... \ G_kG_{k+s}^{-1}G_{k+s-1}^{-1}... \ G_{k+1}^{-1} = I$.

Proof. The following set of equalities constructed using the rule $GG^{-1} = I$ for a single gate G proves the statement.

$$\begin{gathered} G_1G_2... \ G_k = G_{k+1}G_{k+2}... \ G_{k+s} \\ G_1G_2... \ G_kG_{k+s}^{-1}G_{k+s-1}^{-1}... \ G_{k+1}^{-1} = \\ G_{k+1}G_{k+2}... \ G_{k+s}G_{k+s}^{-1}G_{k+s-1}^{-1}... \ G_{k+1}^{-1} \\ G_1G_2... \ G_kG_{k+s}^{-1}G_{k+s-1}^{-1}... \ G_{k+1}^{-1} = I. \end{gathered}$$

Observation 3. For an identity $G_0G_1... \ G_{m-1}$ and any parameter p, $0 \leq p \leq m-1$,

$G_0G_1...G_{p-1} \rightarrow G_{m-1}^{-1}G_{m-2}^{-1}... \; G_p^{-1}$ is a rewriting rule. In the most **trivial circuit cost metric**, when the cost of every gate is assumed to be one (that is, the simple gate count is used), the rewriting rule is regular for parameters p in the range $\frac{m}{2} < p \leq m-1$.

Proof. Proof of this statement follows from the previous one by renaming the subscripts and listing the equalities in the reverse order.

Observation 4. If $G_0G_1... \; G_{m-1} = I$, then $G_1... \; G_{m-1}G_0 = I$.

Proof. The following proves the statement.

$$\begin{aligned} G_0G_1... \; G_{m-1} &= I \\ G_0^{-1}G_0G_1... \; G_{m-1} &= G_0^{-1}I \\ G_1... \; G_{m-1} &= G_0^{-1} \\ G_1... \; G_{m-1}G_0 &= G_0^{-1}G_0 \\ G_1... \; G_{m-1}G_0 &= I. \end{aligned}$$

Definition 3. A **size** m **template** is a sequence of m gates (a circuit) that realizes the identity function. Any template of size m must be independent of all templates of smaller or equal size, *i.e.* for a given template T of size m no application of any set of templates of smaller size can decrease the number of gates in T or make it equal to another template. The template $G_0 \; G_1... \; G_{m-1}$ can be applied in two directions:

Forward application is a rewriting rule of the form $G_iG_{(i+1) \bmod m}\cdots \; G_{(i+p-1) \bmod m} \rightarrow G_{(i-1) \bmod m}^{-1} G_{(i-2) \bmod m}^{-1}\cdots G_{(i+p) \bmod m}^{-1}$, where $0 \leq i, \; p \leq m-1$.

Backward application is a rewriting rule of the form $G_i^{-1}G_{(i-1) \bmod m}^{-1}\cdots \; G_{(i-k+1) \bmod m}^{-1} \rightarrow G_{(i+1) \bmod m} G_{(i+2) \bmod m}\cdots G_{(i-k) \bmod m}$, where $0 \leq i, \; p \leq m-1$.

Note, that template application requires the inverse be available for each gate type considered. Correctness of this definition follows from the four observations above.

A clear benefit of templates is the reduction of the number of rewriting rules and the consequent storage saving. In fact, one template occupies the same storage space as a single rewriting rule but is capable of storing up to $2m^2$ non-redundant rewriting rules. Assuming the trivial circuit cost metric where each gate has a cost of one, the number of regular non-redundant rewriting rules can be as high as m^2 for odd m and $m(m-1)$ for even m. For basic gate sets where the number of non-redundant rewriting rules is infinite or grows exponentially (for instance, using Toffoli gates with multiple controls results in an exponential growth of the number of rewriting rules), further compaction through classification is required. Depending on the form of the basic gates the classification approaches may vary [12, 13].

The following Lemma is useful as it allows for a decrease in the number of gates that must be matched in order for a template to be applied as a regular rewriting rule.

Lemma 1. Assume a trivial cost metric where the cost of a circuit is calculated as a linear term with unit weights. Then, application of a template $G_0G_1... \; G_{m-1}$ for a parameter $p = p_0$ is equivalent to its application for parameter $p = p_1 := \lfloor \frac{m}{2} \rfloor + 1$ and $p_0 - p_1$ applications of an AA^{-1} template.

Proof. For simplicity, assume that the template $G_0G_1... \; G_{m-1}$ is applied forward for parameter $i = 0$, that is, starting with the gate G_0. Then, for the parameter $p = p_1$, the sequence of gates $G_0G_1... \; G_{p_1-1}$ in the circuit to be simplified can be moved together and will be replaced with $G_{m-1}^{-1}G_{m-2}^{-1}... \; G_{p_1}^{-1}$. For $j = 0, 1, 2, ...$ and while $p_0 - p_1 - j > 0$ gate G_{p_1+j} can be moved to $G_{p_1}^{-1}$ and the gate-inverse rule applies to delete them both. It can be easily seen that after $p_0 - p_1$ such applications of gate-inverse rules the circuit will take the form equivalent to the one achieved by application of $G_0G_1... \; G_{m-1}$ for parameter $p = p_0$.

4. Templates: application

To identify templates, we first find all templates of the form $ABAB$. Such templates applied for parameter $p = 2$ result in construction of the rewriting rule $AB \rightarrow BA$. That is, they define when the two gates can be moved past each other. We call such templates **moving rules** and apply them to move gates to form matches leading to reduction via other templates.

Second, we find all templates of the form AA^{-1}, which we call **gate-inverse rules**. This is straightforward as well, since every self-inverse gate A forms the template AA and every pair of gates A and B, where $B = A^{-1}$ forms one template of the form AB.

Subsequent templates are found by identifying increasingly longer sequences of gates that realize the identity function and which can not be reduced by other templates.

As a general approach, we provide a complete classification of the templates of small size and then supplement those by a set of templates that appear to be useful when a specific synthesis procedure is applied. For example, if a synthesis procedure tends to use a specific type of subcircuit of cost μ which is neither optimal (assume an optimal cost of ν) nor can be simplified by a small size complete set of templates, a template with total cost $\mu + \nu$ can be created using Observation 2 (followed by a generalization process when and if needed). In this paper, we do not construct any of these supplementary type templates, since we focus on using templates when any synthesis algorithm is used to find the initial circuit.

We now illustrate a set of quantum templates using NOT, CNOT and controlled-$sqrt$-of-NOT gates. We first describe the *moving rule*. Assuming gate A has control set C_A (C_A is an empty set in the case of an uncontrolled gate) and tar-

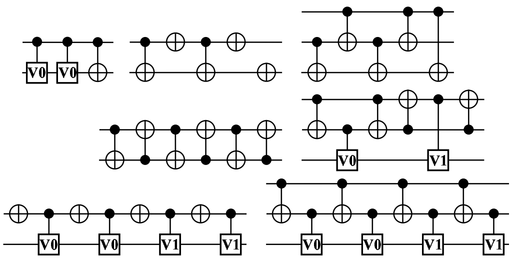

Figure 1. Quantum templates.

get T_A and gate B has control set C_B and target T_B, these two gates form a moving rule if, and only if, $T_A \notin C_B$ and $T_B \notin C_A$. Recall that NOT and CNOT are self-inverses and controlled-V and controlled-V^+ are the inverses of each other. These observations can be used to construct the *gate-inverse* template. All other templates that we have identified are shown in Figure 1, where V (or V^+) is substituted for all occurrences of V_0 and V^+ (or V) is substituted for all occurrences of V_1, i.e. the substitutions is consistent and distinct for V_0 and V_1. The templates reported here were found by inspection. We are currently developing a program to find larger templates and to verify completeness.

When applying templates, the smaller ones are tried first. This means that more general (smaller) rewriting rules are used first after which more specific rules are considered.

To illustrate how templates are applied, consider the example of a non-optimal quantum circuit for the 3-bit full adder with 10 gates which in a non-linear metric yields a final cost of 9 [6]. The original circuit presented in [6] does not appear to be correct since for the input pattern 0100 it gives 1111 as output instead of the expected 1011. The circuit shown in Figure 2A corrects this.

In the circuit in Figure 2A, gates 5 and 7 (counting from the left) can be moved together and form a gate-inverse pair. We move them together and delete them by applying the gate-inverse template. This results in the circuit in Figure 2B. Next, we notice that gates 4, 6 and 8 in this circuit can also be brought together (gates 4 and 8 should be moved towards gate 6). They match gates 5,1 and 2 in the rightmost upper template in Figure 1. Figure 2C shows the three gates brought together, and Figure 2D illustrates the resulting circuit after the template is applied.

The circuit that we found using templates simplification (Figure 2D) is the optimal (for a given input-to-output assignment) reported in [6]. It took our program 1.16 seconds (elapsed time) to simplify the circuit in Figure 2A into the circuit in Figure 2D. The time reported in [6] to synthesize such a circuit is 7 hours. This example clearly shows that templates are useful and effective.

A likely optimal quantum circuit for the 3-bit full adder can be constructed from the reversible implementation presented in [14] and shown in Figure 3A. We first substitute quantum circuits for the Peres gates (each of which is a Toffoli-CNOT pair) see Figure 3B, and then apply the templates. In this case, gates 4 and 6 can be moved together and match the gate-inverse template. So, they are both deleted leading to the circuit in Figure 3C.

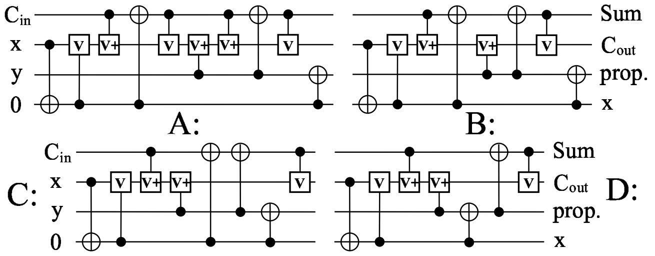

Figure 2. Simplification of a 10-gate quantum network for the 3-bit full adder.

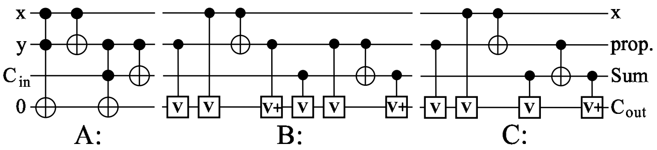

Figure 3. Simplification of an 8-gate quantum network for the 3-bit full adder.

5. Experimental results

Even though the set of gates NOT, CNOT, controlled-V and controlled-V^+ has been used by several authors [4, 6, 10, 16] as elementary building blocks, there are not many quantum circuits reported. However, there are a number of reversible circuits [2, 8, 11, 12, 14, 15] using generalized Toffoli and Fredkin [5] gates. Generalized Fredkin gates can be implemented as a circuit containing two CNOT gates and one generalized Toffoli gate [16]. The generalized Toffoli gate itself can be realized as a circuit with CNOT, controlled-V and controlled-V^+ gates [4, 10]. Thus, any Fredkin-Toffoli reversible circuit can be transformed into a circuit using elementary quantum NOT, CNOT, controlled-V and controlled-V^+ gates.

To develop our general synthesis approach, we first consider quantum circuits for the generalized Toffoli gates and apply our templates to simplify them. Having done that, to simplify a given reversible circuit we: substitute Toffoli-CNOT equivalents for any Fredkin gates; substitute simplified quantum implementations for the Toffoli gates; and apply the templates to obtain a reduced quantum specification.

Table 1 summarizes the results of applying our templates to quantum Toffoli gate implementations [4, 10]. Columns *Size*, *Garbage* and *Best reported* show the size of the Toffoli

Size	Garbage	Best reported	Optimized implementation	%
4	1	16	15	93.75%
5	2	32	26	81.25%
5	1	42	37	88.1%
6	3	48	38	79.17%
6	1	64	54	84.38%
7	4	64	50	78.13%
7	1	96	80	83.33%
8	5	80	62	77.5%
8	1	128	100	78.13%
9	6	96	74	77.08%
9	1	160	128	80%
10	7	112	86	76.79%
10	1	192	152	79.17%
11	8	128	98	76.56%
11	1	224	176	78.57%

Table 1. Toffoli gate simulations.

gate, the number of auxiliary lines associated with the particular implementation of this gate, and the gate count in the best reported quantum circuit with CNOT, controlled-V and controlled-V^+ gates. We show the gate counts for our optimized implementations and the percentage the size of the optimized circuit represents with respect to the size of the best previously reported realization.

The results in Table 1 show that the set of Toffoli gate size $(m+1)$ realizations with implementation cost of $16m-32$ and garbage lines $(m-2)$ (for $m>3$) are always simplified to size $12m-22$ circuits. We conjecture this will always be the case.

We next consider the derivation of quantum circuits from Toffoli/Fredkin gate circuits. One of the main obstacles in such an approach is that given a Toffoli gate, there are multiple ways of substituting an equivalent quantum circuit. First, Toffoli gates are symmetric with regard to their control variables, while the corresponding quantum circuits are not. For m controls, this results in $m!$ substitutions. In addition, the quantum realization of a Toffoli gate $TOF(x_1, x_2, ..., x_m; x_{m+1})$ in an n-line circuit, uses k auxiliary lines (assuming $m+k+1 \leq n$). This gives $(n-m-1)$-choose-k ways of assigning auxiliary lines for the quantum implementations to the remaining lines in the circuit. Also, since a Toffoli gate is self-inverse, but its quantum realizations are not symmetric circuits, the number of possible substitutions is doubled. Finally, each size $(m+1)$ Toffoli gate has two basic quantum realizations [4, 10] one with only 1 garbage line and the other with $(m-2)$ garbage lines. This very large search space requires a heuristic be used.

We first observe that quantum circuits for Toffoli gates of size $(m+1)$ that require $(m-2)$ garbage lines are less costly and better structured. Thus, it is easier to come up with a good procedure for substituting these gates in the circuit to yield the most simplification. We use this implementation rather than using the implementation with just 1 garbage line which is more expensive. We further substitute the circuits for the Toffoli gates so that two neighboring Toffoli gates would share as many quantum gates as possible. Since this is not guaranteed to be the best way of replacing a Toffoli gate with a quantum circuit, we are now working on better ways of substituting the Toffoli gates as well as utilizing the quantum circuits for Toffoli gates which require only one auxiliary bit.

Many reversible circuits have constant input values and garbage outputs. This typically results when a non-reversible function is mapped to a reversible one prior to synthesis as a reversible circuit. In these cases, we apply two additional procedures. First, we look at the beginning of the circuit for a gate whose control is an input constant. If such a gate can be moved to the beginning of the circuit one of two transformations applies. If the input constant controlling the gate is zero, the gate can be removed. If the input constant is one, the constant control of the gate can be deleted (assuming an uncontrolled gate has a lesser cost). The second procedure looks at the end of the circuit and for every gate with the target on a garbage output tries to move the gate to be the last gate affecting that output such that the output is not used after the gate is applied. Every such gate can be safely deleted from the circuit since we do not care about the resulting values on the garbage lines.

We took several circuits from [11] and compared their quantum realization costs before and after applying the above approach. Table 2 summaries the results. Different circuits for the same function are distinguished by the numbers that come after the # sign. *Cost before* and *Garbage* list the quantum gate count and number of garbage lines in the circuit to be simplified. It can be calculated that given a reversible circuit on n rails with Toffoli gates of maximal size M ($M<n$), the garbage overhead in our Toffoli gate substitution procedure is $2M-n-3$ when this number is greater than zero and zero otherwise. *Cost after* shows the quantum gate count after reversible gates are substituted with their simplified quantum circuits and the resulting circuit is run through the template simplification process.

Finally we present a very good circuit for the 5-bit Grover oracle function $4mod5$. It leaves the first four wires unchanged and inverts the last wire if, and only if, the first four wires represent an integer divisible by 5. We first found a Toffoli gate circuit using a synthesis method we are currently developing. We then applied the techniques described above. If we don't require that the first four lines pass the inputs unchanged, the circuit is as shown in Figure 4A. If

Name	Garbage	Cost before	Cost after	%
$4mod5\#1$	4	24	15	62.5%
$4mod5\#3$	4	13	9	69.23%
$5mod5\#1$	8	184	128	69.57%
$5mod5\#2$	8	80	45	56.25%
$add3$	2	12	6	50%
$mod1024adder$	0	1975	1521	77.01%
$rd53\#1$	4	277	233	84.12%
$rd53\#2$	4	152	114	75%
$rd53\#3$	5	44	33	75%
$rd53\#4$	4	137	96	70.07%
$rd53\#5$	4	86	68	79.07%

Table 2. Simplification of the benchmarks.

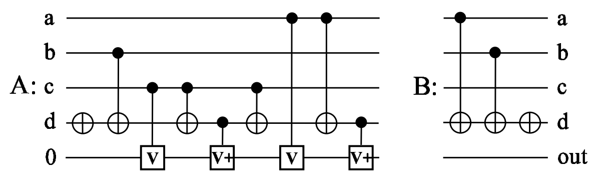

Figure 4. Circuit for the Grover oracle.

the inputs are required to be passed through unchanged, the subcircuit in Figure 4B must be appended to the right of the circuit in Figure 4A.

6. Future work

There are several possibilities to improve our simplification approach. In addition to better substitution of quantum circuits for Toffoli gates and the search for new templates, we plan to adapt the templates tool for reversible circuits to minimize the quantum cost instead of the reversible gate count. This is expected to result in a better starting point for quantum cost reduction. We are also working on quantum gate level compaction, which should be possible to accomplish using the moving rule and its trivial extensions. Finally, we are interested in extending templates to other sets of quantum gates including rotation gates.

7. Conclusion

We have introduced templates for quantum circuit simplification. Templates can be developed for any type of quantum circuit, and can be applied for various cost metrics. We demonstrated the effectiveness of our approach using a variety of previously published circuits. In particular, we reduced the sizes of the best known Toffoli gate quantum realizations on average by 19.2% and the costs of a set of benchmark circuits on average by 30.2%.

References

[1] IBM's test-tube quantum computer makes history. IBM T.J. Watson Research Center, http://researchweb.watson.ibm .com/resources/news/20011219_quantum.shtml, Dec. 2001.

[2] S. Aaronson. Multilinear formulas and skepticism of quantum computing. In *ACM STOC*, 2004.

[3] A. Agrawal and N. K. Jha. Reversible logic synthesis. In *DATE*, pages 1384–1385, Paris, France, February 2004.

[4] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVinchenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter. Elementary gates for quantum computation. *Physical Review A*, 52:3457–3467, 1995.

[5] E. Fredkin and T. Toffoli. Conservative logic. *International Journal of Theoretical Physics*, 21:219–253, 1982.

[6] W. N. N. Hung, X. Song, G. Yang, J. Yang, and M. Perkowski. Quantum logic synthesis by symbolic reachability analysis. In *DAC*, pages 838–841, June 2004.

[7] K. Iwama, Y. Kambayashi, and S. Yamashita. Transformation rules for designing CNOT-based quantum circuits. In *DAC*, pages 419–424, June 2002.

[8] P. Kerntopf. A new heuristic algorithm for reversible logic synthesis. In *DAC*, pages 834–837, June 2004.

[9] C. Lomont. Quantum circuit identities. Technical Report quant-ph/0307111, Quant-Ph ArXiv, July 2003.

[10] D. Maslov and G. Dueck. Improved quantum cost for n-bit Toffoli gates. *IEE Electronics Letters*, 39(25):1790–1791, December 2003, quant-ph/0403053.

[11] D. Maslov, G. Dueck, and N. Scott. Reversible logic synthesis benchmarks page. www.cs.uvic.ca/˜dmaslov/, Aug. 2004.

[12] D. Maslov, G. W. Dueck, and D. M. Miller. Fredkin/Toffoli templates for reversible logic synthesis. In *ICCAD*, pages 256–261, November 2003.

[13] D. Maslov, G. W. Dueck, and D. M. Miller. Simplification of Toffoli networks via templates. In *Symposium on Integrated Circuits and System Design*, pages 53–58, September 2003.

[14] D. M. Miller, D. Maslov, and G. W. Dueck. A transformation based algorithm for reversible logic synthesis. In *DAC*, pages 318–323, June 2003.

[15] A. Mishchenko and M. Perkowski. Logic synthesis of reversible wave cascades. In *IWLS*, pages 197–202, June 2002.

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

[17] A. Peres. Reversible logic and quantum computers. *Physical Review A*, 32:3266–3276, 1985.

[18] V. V. Shende, A. K. Prasad, K. N. Patel, I. L. Markov, and J. P. Hayes. Scalable simplification of reversible logic circuits. In *IWLS*, May 2003.

[19] J. A. Smolin and D. P. DiVincenzo. Five two-bit quantum gates are sufficient to implement the quantum Fredkin gate. *Physical Review A*, 53:2855–2856, 1996.

[20] T. Toffoli. Reversible computing. *Tech memo MIT/LCS/TM-151, MIT Lab for Comp. Sci*, 1980.

[21] G. F. Viamontes, I. L. Markov, and J. P. Hayes. Graph-based simulation of quantum computation in the state-vector and density-matrix representation. In *SPIE*, April 2004.