Templates for Positive and Negative Control Toffoli Networks

Md Zamilur Rahman and Jacqueline E. Rice

Department of Mathematics and Computer Science,
University of Lethbridge, Lethbridge, AB, Canada
{mdzamilur.rahman,j.rice}@uleth.ca

Abstract. This paper proposes templates for positive and negative control Toffoli gates for post synthesis optimization of reversible circuits. Templates $1-5$ can be applied to two adjacent Toffoli gates $T_1(C_1, t_1)$ and $T_2(C_2, t_2)$ where C_i is the set of controls, $|C_1| = |C_2|$, and $|t_1| = |t_2|$. Templates $6-7$ can be applied to two different size Toffoli gates $T_1(C_1, t_1)$ and $T_2(C_2, t_2)$ where C_i is the set of controls, $|C_1| = |C_2|$ and t_i is the target, $|t_1| = |t_2|$. When applying our templates to circuits generated by the improved shared cube synthesis approach [14] a reduction in quantum cost was achieved for 98 of the 122 circuits. On average a 16.82% reduction in quantum cost was achieved, and in some cases up to 49.60% reduction was obtained.

Keywords: reversible circuit, Toffoli gate, quantum cost, gate count

1 Introduction

Power dissipation and heat generation are serious problems in today's traditional circuit technologies. According to R. Landauer's observation in 1961, the amount of energy dissipated for each lost bit of information is $KTln2$ where K is the Boltzmann's constant $(1.3807 \times 10^{-23} JK^{-1})$ and T is the Temperature [7]. This is a significant amount of energy for millions of operations. In [1], Bennett said that in order to not dissipate energy the system must be logically reversible. Reversible circuits do not erase any information when operations are performed. In reversible circuits, all operations are performed in a bijective manner. Thus fan-out and feedback operations are not allowed in reversible circuits. Reversible circuits have applications in fields such as quantum computing [15] and optical computing [3]. As a result, reversible logic is being considered as an alternative to conventional logic. Instead of conventional logic gates reversible gates like Toffoli gates, Fredkin gates, and Peres gates are used in reversible circuits.

Several synthesis approaches for reversible logic have been proposed, including transformation based synthesis [11], Exclusive-OR Sum-of-Products (ESOP) based synthesis [5,13] and binary decision diagram (BDD) based synthesis [18]. In this paper we describe a template-based post-processing approach that is based on mixed-polarity Toffoli gates.

The remainder of this paper is organized as follows. The following section briefly introduces basic concepts in reversible logic. It offers an overview of the Toffoli gate and the cost metrics of a reversible circuit. Section 3 gives the motivations of this work, and the proposed templates are discussed in section 4. Section 6 summarizes the experimental results followed by conclusions in section 7.

2 Background

In this section we provide some brief background and notation to orient the reader.

2.1 Reversible Gates and Reversible Circuits

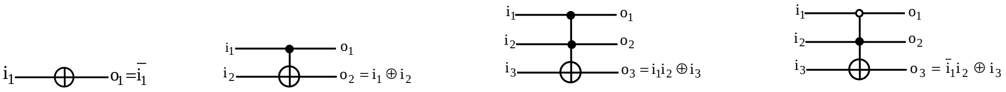
In this work we focus solely on the Toffoli gate. Other reversible gates are described in *e.g.* [16], which is recommended as a useful introductory article on the topic of reversible logic.

An n-bit Toffoli gate or Multiple Control Toffoli (MCT) gate is a reversible gate that has n inputs and n outputs where $(i_1, i_2, ..., i_n)$ is the input vector, $(o_1, o_2, ..., o_n)$ is the output vector, and $o_j = i_j$ where $j = 1, 2, ..., n-1$ and $o_n = i_1 i_2 ... i_{n-1} \oplus i_n$. The first $n-1$ bits are known as controls and the last n^{th} bit is known as the target. This gate passes all the inputs to the outputs and inverts the target bit when all control bits are 1. When $n = 1$, this gate is known as the NOT gate. When $n = 2$, it is referred to as a controlled-NOT (CNOT) gate or Feynman gate. We note that for the sake of simplicity we assume that the n^{th} bit is the target; however the target bit could be any of the n bits with which the gate interacts.

A negative-control Toffoli gate is a gate that may have one or more negative controls. The gate maps the n inputs $(i_1, i_2, ..., i_n)$ to the n outputs $(o_1, o_2, ..., o_n)$ where $o_j = i_j$, $j = 1, 2, ..., n-1$ and $o_n = \bar{i_1} i_2 ... i_{n-1} \oplus i_n$ and $\bar{i_1}$ is a negative control. This gate passes all the inputs to the outputs and inverts the target bit when all the positive controls have value 1 and negative controls have value 0.

In this paper, $\oplus$ represents the target line, $\bullet$ indicates a positive control, and $\circ$ is used to indicate a negative control line. A Toffoli gate can also be written as TOF$(C; t)$ where C is the set of controls and t is the target line. The size of a Toffoli gate refers to the number of controls plus target. Figure 1 illustrates several versions of the n-bit Toffoli gate.

A reversible circuit is a cascade of reversible gates without fan-out and feedback. If a reversible circuit is built using only NOT, CNOT, and Toffoli gates (NCT) or Multiple Control Toffoli gates (MCT) it is referred to as a Toffoli circuit.



(a) NOT gate.

(b) CNOT gate.

(c) 3-bit Toffoli gate.

(d) Negative-control Toffoli gate.

Fig. 1. Toffoli gates

2.2 Cost Metrics

A reversible function may be realized in different ways, resulting in different circuits. We briefly summarize two common cost metrics used in evaluating reversible circuits.

Gate Count Gate count is the simplest way to evaluate different reversible circuits. This refers to a simple count of the number of gates in a circuit. It does not, however, consider the complexity of the circuit. Consider two circuits where the first circuit consists of three 2-input Toffoli gates and the second circuit consists of two 6-input Toffoli gates. In this case a gate count might indicate that the second circuit is preferable, as it has fewer gates. However, it contains significantly more complex gates.

Quantum Cost Quantum cost is an important measure for comparison of reversible circuits. The quantum cost of a gate is defined as the number of basic quantum operations needed to realize the gate [8]. Any reversible gate can be decomposed into basic quantum (1×1 and 2×2) gates. The number of basic quantum gates required to implement a circuit is referred to as the quantum cost of the circuit. The quantum cost of the NOT, CNOT, and 3-bit Toffoli gate is 1, 1, and 5, respectively. In general, as the number of controls for a gate increases so does the quantum cost.

The quantum cost of an n-bit negative control Toffoli gate with at least one control is exactly the same as the cost of an n-bit Toffoli gate. When all the controls are negative, an extra cost of 2 is required if zero or $(n - 3)$ garbage lines are used. An additional cost of 4 is required when only one garbage line is used [10].

3 Motivation and Related Work

If a circuit is non-optimal then it may be possible to decrease the size and quantum cost of the circuit by replacing sequences of gates with another equivalent sequence of gates; this is known as a template-driven reduction method, or template matching [11]. Template matching is an approach to reduce the number of gates and quantum cost by removing unnecessary gates from the network and has no effect on the functionality of the circuit. Templates for synthesis of positive control Toffoli networks have been classified based on the number of variables and proposed in [9] as well as [6]. For positive and negative control Toffoli gates new merging, moving, and splitting rules are proposed and an algorithm utilizing these rules is proposed in [2]. Templates and rules using both positive and negative control Toffoli gates are also proposed in [4].

4 Proposed Approach

In developing our templates we considered the various ways in which two Toffoli gates with the same target line can appear in a circuit:

1. Two same size gates with controls on the same or different lines, as shown in Figure 2(a), or
2. Two different size gates with controls on the same or different lines, as shown in Figure 2(b).

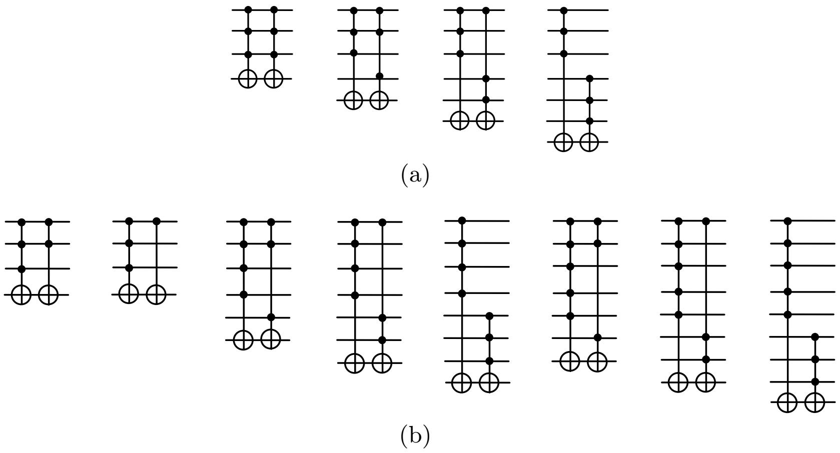


Fig. 2. Possible ways for two gates with the same target line to appear in a circuit.

We have proposed 7 templates that may be applied in various situations. Templates $1-5$ can be applied to two adjacent Toffoli gates $T_1(C_1, t_1)$ and $T_2(C_2, t_2)$ where C_i is the set of controls, $|C_1| = |C_2|$ and t_i is the target, $|t_1| = |t_2|$. In templates $1-4$, two gates share the same control line but in template 5 one of the controls of one gate is on a different line. Templates $6-7$ can be applied to two different size Toffoli gates $T_1(C_1, t_1)$ and $T_2(C_2, t_2)$ where C_i is the set of controls, $|C_1| > |C_2|$ or $|C_2| > |C_1|$ and t_i is the target, $|t_1| = |t_2|$. In template 6 the two gates may differ, but only by at most 1 line. In template 7, the difference in the size of two Toffoli gates is ≥ 1. In all cases we are interested in Toffoli gates that have the same target line. Details of each type of template are as follows.

Template 1:

Template 1 can be applied to two adjacent CNOT gates in the case where one CNOT gate has a positive control and the other has a negative control. In this case the two CNOT gates can be replaced by a single NOT gate [4].

$$T(C; x_t)T(\overline{C}; x_t) \equiv T(; x_t) \tag{1}$$

Template 2:

If two Toffoli gates have the same controls, then the two gates negate each other. This property is known as self-reversibility [9].

$$T(C; x_t)T(C; x_t) \equiv I \tag{2}$$

Template 3:

If two Toffoli gates have same controls but one of the controls is the inverse, then these two gates can be replaced by one Toffoli gate with all the common controls [2]. An example is shown in Figure 3.

$$T(C \cup x_i; x_t)T(C \cup \overline{x_i}; x_t) \equiv T(C; x_t) \tag{3}$$

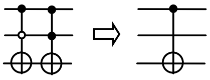


Fig. 3. Template 3

Template 4:

If two n-bit $(n \geq 3)$ Toffoli gates have controls on the the same lines but two (i.e. x_i, x_j) of the controls have different polarity, then the two n-bit $(n \geq 3)$ gates can be replaced by two CNOT gates and one $(n-1)$-bit $(n \geq 2)$ Toffoli gate. Equations(4a) and (4b) formalize this, while Figure 4 illustrates two possible ways to apply this template.

$$T(C \cup x_i \cup \overline{x_j}; x_t)T(C \cup \overline{x_i} \cup x_j; x_t) \equiv T(x_i; x_j)T(C \cup x_j; x_t)T(x_i; x_j) \tag{4a}$$

$$T(C \cup \overline{x_i} \cup \overline{x_j}; x_t)T(C \cup x_i \cup x_j; x_t) \equiv T(x_i; x_j)T(C \cup \overline{x_j}; x_t)T(x_i; x_j) \tag{4b}$$

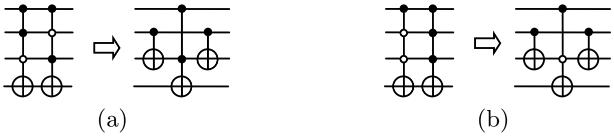


Fig. 4. Template 4

Template 5:

This template can be applied to two Toffoli gates of the same size where one of the controls is on a different line. In this case the two Toffoli gates can be replaced by two CNOT gates and one Toffoli gate [17]. The three situations in which this may occur are formally described in Equations (5a), (5b), and (5c) and illustrated in Figure 5.

$$T(C \cup x_i; x_t)T(C \cup x_j; x_t) \equiv T(x_i; x_j)T(C \cup x_j; x_t)T(x_i; x_j) \tag{5a}$$

$$T(C \cup \overline{x_i}; x_t)T(C \cup \overline{x_j}; x_t) \equiv T(x_i; x_j)T(C \cup x_j; x_t)T(x_i; x_j) \tag{5b}$$

$$T(C \cup \overline{x_i}; x_t)T(C \cup x_j; x_t) \equiv T(x_i; x_j)T(C \cup \overline{x_j}; x_t)T(x_i; x_j) \tag{5c}$$

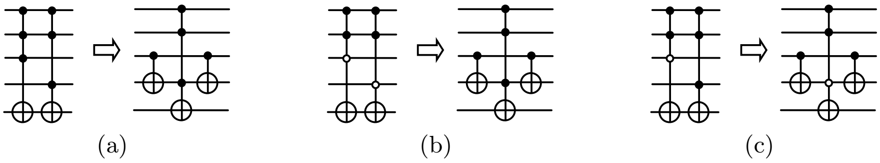


Fig. 5. Template 5

Template 6:

If the size of two Toffoli gates differs by 1 and all the controls except the additional control in the larger gate are on the same lines, then this sequence of gates can be replaced by a Toffoli gate of the same size as the larger gate [17]. The two situations are described in Equations (6a) and (6b) and illustrated in Figure 6.

$$T(C; x_t)T(C \cup x_i; x_t) \equiv T(C \cup \overline{x_i}; x_t) \tag{6a}$$

$$T(C; x_t)T(C \cup \overline{x_i}; x_t) \equiv T(C \cup x_i; x_t) \tag{6b}$$

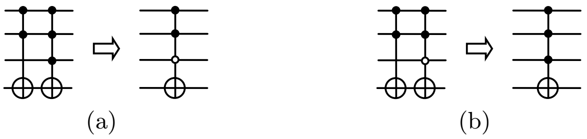


Fig. 6. Template 6

Template 7:

This template can be applied to two different sized n-bit ($n \geq 3$) Toffoli gates as described in Equations (7aa)-(7db)and illustrated in Figure 7.

$$T(C \cup x_i \cup x_j; x_t)T(C \cup x_k; x_t) \equiv T(x_i \cup x_j; x_k)T(C \cup x_k; x_t)T(x_i \cup x_j; x_k) \tag{7aa}$$

$$T(C \cup x_i \cup x_j; x_t)T(C \cup \overline{x_k}; x_t) \equiv T(x_i \cup x_j; x_k)T(C \cup \overline{x_k}; x_t)T(x_i \cup x_j; x_k) \tag{7ab}$$

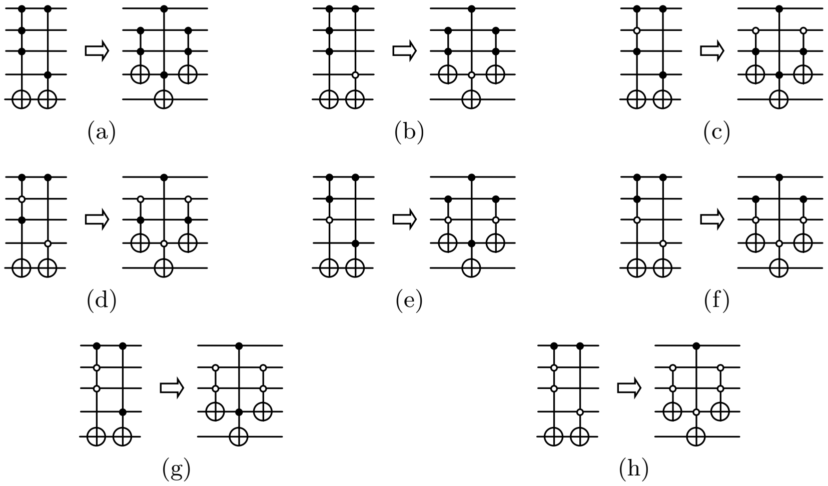


Fig. 7. Template 7

$$T(C \cup \overline{x_i} \cup x_j; x_t)T(C \cup x_k; x_t) \equiv T(\overline{x_i} \cup x_j; x_k)T(C \cup x_k; x_t)T(\overline{x_i} \cup x_j; x_k) \tag{7ba}$$

$$T(C \cup \overline{x_i} \cup x_j; x_t)T(C \cup \overline{x_k}; x_t) \equiv T(\overline{x_i} \cup x_j; x_k)T(C \cup \overline{x_k}; x_t)T(\overline{x_i} \cup x_j; x_k) \tag{7bb}$$

$$T(C \cup x_i \cup \overline{x_j}; x_t)T(C \cup x_k; x_t) \equiv T(x_i \cup \overline{x_j}; x_k)T(C \cup x_k; x_t)T(x_i \cup \overline{x_j}; x_k) \tag{7ca}$$

$$T(C \cup x_i \cup \overline{x_j}; x_t)T(C \cup \overline{x_k}; x_t) \equiv T(x_i \cup \overline{x_j}; x_k)T(C \cup \overline{x_k}; x_t)T(x_i \cup \overline{x_j}; x_k) \tag{7cb}$$

$$T(C \cup \overline{x_i} \cup \overline{x_j}; x_t)T(C \cup x_k; x_t) \equiv T(\overline{x_i} \cup \overline{x_j}; x_k)T(C \cup x_k; x_t)T(\overline{x_i} \cup \overline{x_j}; x_k) \tag{7da}$$

$$T(C \cup \overline{x_i} \cup \overline{x_j}; x_t)T(C \cup \overline{x_k}; x_t) \equiv T(\overline{x_i} \cup \overline{x_j}; x_k)T(C \cup \overline{x_k}; x_t)T(\overline{x_i} \cup \overline{x_j}; x_k) \tag{7db}$$

Moving Rule

Two adjacent gates $g(C_1, t_1)$ and $g(C_2, t_2)$ in a reversible circuit can be interchanged iff $C_1 \cap t_2 = \emptyset$ and $C_2 \cap t_1 = \emptyset$, i.e. the target of each gate is not a control of the other gate [20]. Applying a moving rule increases the possibilities for matching more templates and can lead to further optimization.

5 Steps/Algorithm

The template matching process is performed as follows: Consider two gates $g1$ and $g2$ from the gate list of a circuit.

1. if two gates have the same target line then we begin searching for templates
   (a) if $g1$ and $g2$ match any of the templates then replace $g1$ and $g2$ with the equivalent gates from that template (i.e. $g1'$, $g2'$...) and add the new gates to the new gate list
   (b) move on to consider the next two gates in the circuit (i.e. $g3$ and $g4$); go to step 1
   (c) if no match is found for any template then apply moving rule:
      i. if $g1$ can pass $g2$ then interchange $g1$ and $g2$; add $g2$ into the new gate list, $g1$ and $g3$ become the gates under consideration; go to step 1
      ii. else $g1$ and $g2$ add into the new gate list and consider the next two gates (i.e. $g3$ and $g4$); go to step 1
2. else apply moving rule to $g1$ and $g2$
   (a) if $g1$ can pass $g2$ then interchange $g1$ and $g2$; add $g2$ into the new gate list, $g1$ and $g3$ become the gates under consideration; go to step 1
   (b) else add $g1$ and $g2$ to the new gate list and consider the next two gates in the circuit (i.e. $g3$ and $g4$); go to step 1

The algorithm is iterated until no further reduction is possible.

6 Experimental Results

We have implemented the proposed templates along with the described moving rule in Java. The implemented programs have been run on an Intel Core 2 Duo CPU T6670 @ 2.20GHz×2 systems running Ubuntu 13.04 with 2GiB main memory for 122 benchmark circuits. These benchmarks were obtained from RevLib [19] and preprocessed by applying the improved shared cube synthesis approach from [14]. All the resulting circuits are QMDD (Quantum Multiple-valued Decision Diagrams) verified [12]. Using QMDD, we compare the resulting circuits (after applying templates) with the original circuits, in order to ensure that the behaviour of the circuit has not been modified. The running time is negligible for the program we developed to implement the algorithm discussed in section 5 and the results are listed in Table 1. Table 1 compares the outputs obtained in the current experiment to the results from the improved shared cube synthesis approach in terms of quantum cost and gate count. In this table PrevGC/PrevQC refers to the gate count/quantum cost obtained from the circuit generated by the improved shared cube synthesis approach, while NewGC/NewQC refers to the new gate count/quantum cost as computed from the circuits generated from our template matching post-processing. The proposed templates reduce the quantum cost of circuits 16.82% on average. col4_135 is the best reported circuit in terms of reduction in quantum cost. apex4_103 exhibited the greatest reduction in gate count, at 86%. fredkin_3, x2_223, miller_5, and pcler8_190 showed no changes in gate count but significant reductions in quantum cost.

Table 1: Applying templates with moving rule

Circuit	PrevGC[14]	NewGC	GCImp.(%)	PrevQC[14]	NewQC	QCImp.(%)
co14_135	14	21	-50.00	3472	1750	49.60
cm85a_127	48	54	-12.50	2206	1232	44.15
decod24-enable_32	9	5	44.44	29	17	41.38
bw_116	287	94	67.25	637	387	39.25
4mod5_8	4	4	0.00	21	13	38.10
decod24_10	9	4	55.56	16	10	37.50
C7552_119	89	32	64.04	399	250	37.34
decod_137	89	32	64.04	399	250	37.34
ham15_30	114	46	59.65	263	183	30.42
ham7_29	37	17	54.05	67	47	29.85
rd73_69	43	52	-20.93	856	619	27.69
add6_92	153	159	-3.92	5135	3714	27.67
hwb5_13	49	32	34.69	372	270	27.42
clip_124	78	80	-2.56	3824	2803	26.70
fredkin_3	7	7	0.00	15	11	26.67
mod5d2_17	15	12	20.00	38	28	26.32
mod5d1_16	11	12	-9.09	27	20	25.93
0410184_85	218	256	-17.43	7636	5662	25.85
plus127mod8192_78	36	31	13.89	803	602	25.03
z4_224	34	38	-11.76	489	370	24.34
z4ml_225	34	38	-11.76	489	370	24.34
adr4_93	41	41	0.00	645	489	24.19
apla_107	72	40	44.44	1683	1277	24.12
radd_193	41	43	-4.88	645	490	24.03
dc1_142	31	18	41.94	127	97	23.62
mod5mils_18	11	12	-9.09	30	23	23.33
max46_177	42	52	-23.81	4524	3540	21.75
3_17_6	11	9	18.18	28	22	21.43
cycle10_2_61	42	46	-9.52	1273	1004	21.13
apex4_103	5622	760	86.48	35840	28268	21.13

Table 1: Applying templates with moving rule

Circuit	PrevGC[14]	NewGC	GCImp.(%)	PrevQC[14]	NewQC	QCImp.(%)
sym6_63	13	16	-23.08	721	571	20.80
plus63mod8192_80	35	31	11.43	847	672	20.66
majority_176	5	6	-20.00	133	106	20.30
sym10_207	83	105	-26.51	15640	12990	16.94
cm42a_125	42	17	59.52	161	134	16.77
pm1_192	42	17	59.52	161	134	16.77
graycode6_11	12	10	16.67	12	10	16.67
cm151a_129	26	25	3.85	769	642	16.51
4_49_7	20	14	30.00	97	81	16.49
dc2_143	51	39	23.53	1084	906	16.42
sqrt8_205	22	23	-4.55	466	393	15.67
root_197	48	44	8.33	1811	1528	15.63
hwb7_15	233	118	49.36	3015	2551	15.39
hwb6_14	92	52	43.48	839	711	15.26
x2_223	23	23	0.00	433	367	15.24
sao2_199	41	33	19.51	3767	3203	14.97
aj-e11_81	18	11	38.89	74	63	14.86
urf2_73	479	254	46.97	8742	7453	14.74
wim_220	23	14	39.13	139	119	14.39
urf5_76	210	115	45.24	5364	4614	13.98
miller_5	9	9	0.00	29	25	13.79
inc_170	75	32	57.33	892	769	13.79
sqn_203	37	37	0.00	1346	1171	13.00
mlp4_184	80	66	17.50	2496	2174	12.90
hwb8_64	480	261	45.63	8195	7158	12.65
5xp1_90	58	44	24.14	786	687	12.60
cu_141	28	20	28.57	781	687	12.04
rd32_19	6	7	-16.67	25	22	12.00
cm82a_126	17	14	17.65	126	111	11.90
f51m_159	327	359	-9.79	28382	25020	11.85
ex-1_82	6	4	33.33	17	15	11.76
in0_162	245	115	53.06	7949	7014	11.76
f2_158	14	13	7.14	112	99	11.61
9symml_91	52	63	-21.15	10943	9729	11.09
sym9_71	52	63	-21.15	10943	9729	11.09
misex1_178	42	22	47.62	332	296	10.84
rd84_70	68	70	-2.94	2329	2079	10.73
table3_209	701	201	71.33	18606	16630	10.62
life_175	50	58	-16.00	6074	5429	10.62
misex3_180	854	576	32.55	49076	43865	10.62
alu3_97	72	56	22.22	1986	1780	10.37
ham3_28	6	5	16.67	10	9	10.00
mod5adder_66	28	26	7.14	353	318	9.92
hwb9_65	1011	554	45.20	23471	21173	9.79
alu2_96	78	82	-5.13	4369	3942	9.77
example2_156	78	82	-5.13	4369	3942	9.77
urf1_72	960	524	45.42	23769	21497	9.56
dist_144	94	82	12.77	3700	3348	9.51
sqr6_204	54	50	7.41	583	528	9.43
sf_232	4	5	-25.00	32	29	9.38
urf3_75	1501	941	37.31	53157	48218	9.29
ex1010_155	1675	775	53.73	52788	48110	8.86
tial_214	459	483	-5.23	43412	39731	8.48
dk17_145	34	27	20.59	1014	930	8.28
urf6_77	1862	287	84.59	39386	36314	7.80
alu_9	4	5	-25.00	40	37	7.50
misex3c_181	822	581	29.32	49720	46069	7.34
rd53_68	17	19	-11.76	220	206	6.36
squar5_206	31	29	6.45	221	207	6.33
plus63mod4096_79	32	28	12.50	676	634	6.21
4mod7_26	12	11	8.33	84	79	5.95
urf4_89	4293	2972	30.77	169830	160548	5.47
pcler8_190	18	18	0.00	323	308	4.64
alu4_98	454	456	-0.44	41127	39331	4.37
ex3_152	4	5	-25.00	76	73	3.95

Table 1: Applying templates with moving rule

Circuit	PrevGC[14]	NewGC	GCImp.(%)	PrevQC[14]	NewQC	QCImp.(%)
one-two-three_27	8	5	37.50	38	37	2.63
ex2_151	7	8	-14.29	146	143	2.05
cm163a_133	35	27	22.86	546	536	1.83
Average			**16.68**			**16.82**

7 Conclusion and Future Work

This paper proposes two new templates for positive and negative control Toffoli gates (templates 4 and 7). Template 4 can be applied to two $\geq$ 3-bit Toffoli gates with controls on the same lines while template 7 can be applied to two different size $\geq$ 3-bit Toffoli gates. 98 of the 122 circuits generated by the improved shared cube synthesis approach [14] showed improvement in quantum cost after applying our templates. Results show that the proposed templates can reduce the quantum cost up to 49% (on average, 16.82%) and the gate count up to 86% (on average, 16.68%). Future work may pursue several avenues related to this work, including identifying additional templates, particularly for Toffoli gates with different target lines, and also improving the template matching algorithm. Of course, the issue of template matching with negative controls has not yet been thoroughly studied, and as we pursue this work a broader investigation will also be required.

Acknowledgments

This research was funded by a grant from the Natural Sciences and Engineering Research Council of Canada (NSERC). We also would like to acknowledge the efforts of the reviewers in providing interesting and thoughtful comments to help us improve the work; this was greatly appreciated.

References

1. C. H. Bennett. Logical reversibility of computation. *IBM J. Res. Dev.*, 17(6):525–532, November 1973.
2. Xueyun Cheng, Zhijin Guan, Wei Wang, and Lingling Zhu. A simplification algorithm for reversible logic network of positive/negative control gates. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on*, pages 2442–2446, 2012.
3. R. Cuykendall and D. R. Andersen. Reversible optical computing circuits. *Optics Letters*, 12(7):542544, 1987.
4. K. Datta, G. Rathi, R. Wille, I. Sengupta, H. Rahaman, and R. Drechsler. Exploiting negative control lines in the optimization of reversible circuits. In Gerhard W. Dueck and D.Michael Miller, editors, *Reversible Computation*, volume 7948 of *Lecture Notes in Computer Science*, pages 209–220. Springer Berlin Heidelberg, 2013.
5. K. Fazel, M.A. Thornton, and J.E. Rice. Esop-based Toffoli gate cascade generation. In *Communications, Computers and Signal Processing, 2007. PacRim 2007. IEEE Pacific Rim Conference on*, pages 206–209, 2007.

6. K. Iwama, Y. Kambayashi, and S. Yamashita. Transformation rules for designing CNOT-based quantum circuits. In *Design Automation Conference, 2003. Proceedings*, pages 419–424, 2002.
7. R. Landauer. Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 44(1.2):261–269, 2000.
8. D. Maslov and G.W. Dueck. Reversible cascades with minimal garbage. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 23(11):1497–1509, 2004.
9. D. Maslov, G.W. Dueck, and D.M. Miller. Toffoli network synthesis with templates. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 24(6):807–817, 2005.
10. D. Maslov, G.W. Dueck, D.M. Miller, and C. Negrevergne. Quantum circuit simplification and level compaction. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 27(3):436–444, March 2008.
11. D.M. Miller, D. Maslov, and G.W. Dueck. A transformation based algorithm for reversible logic synthesis. In *Design Automation Conference, 2003. Proceedings*, pages 318–323, 2003.
12. D.M. Miller and M.A. Thornton. Qmdd: A decision diagram structure for reversible and quantum circuits. In *Multiple-Valued Logic, 2006. ISMVL 2006. 36th International Symposium on*, pages 30–30, May 2006.
13. J. E. Rice N. M. Nayeem. A shared-cube approach to esop-based synthesis of reversible logic. *Facta Univ. Ser.: Elec. Energ.*, 24(3):385–402, December 2011.
14. N. M. Nayeem. Synthesis and Testing of Toffoli Circuits. Master's thesis, University of Lethbridge, 2011.
15. M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information.* Cambridge Univ. Press, 2000.
16. W. D. Pan and M. Nalasani. Reversible logic. *IEEE Potentials*, pages 38–41, February/March 2005.
17. Z. Sasanian. *Technology Mapping and Optimization for Reversible and Quantum Circuits*. PhD thesis, University of Victoria, 2012.
18. R. Wille and R. Drechsler. Bdd-based synthesis of reversible logic for large functions. In *Design Automation Conference, 2009. DAC '09. 46th ACM/IEEE*, pages 270–275, 2009.
19. 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*, pages 220–225, 2008. RevLib is available at http://www.revlib.org.
20. R. Wille, M. Soeken, C. Otterstedt, and R. Drechsler. Improving the mapping of reversible circuits to quantum circuits using multiple target lines. In *Design Automation Conference (ASP-DAC), 2013 18th Asia and South Pacific*, pages 145–150, Jan 2013.