Dynamic Template Matching with Mixed-polarity Toffoli Gates

Md Mazder Rahman[1], Mathias Soeken[2,3], and Gerhard W. Dueck[1]
[1] Faculty of Computer Science, University of New Brunswick, Canada
[2] Department of Mathematics and Computer Science, University of Bremen, Germany
[3] Cyber-Physical Systems, DFKI GmbH, Bremen, Germany
email: Md._Mazder.Rahman@unb.ca, msoeken@informatik.uni-bremen.de, gdueck@unb.ca

***Abstract*—The Toffoli gate, as originally proposed, had only positive controls. It has been shown that mixed polarity controlled Toffoli gates can be efficiently implemented. In fact, their quantum cost is the same as for positive controlled gates in most cases. Thus it is advantageous to consider circuits with mixed polarity Toffoli gates. Template matching has been successfully used to reduce the number of Toffoli gates in reversible circuits. Little work on templates with mixed polarity gates has been reported. Unfortunately, the number of potential templates increases dramatically, if mixed polarity is introduced. Here we propose a dynamic template matching algorithm that takes templates with few lines and dynamically extends the lines to find matches. Experimental results show that the proposed approach has a significant impact on reducing the total number of gates (57% in the best case) in circuits.**

I. Introduction

Applications of reversible logic are found in quantum computation [1] where exponential speed up can be achieved for some algorithms [2]. For classical reversible functions (a special case of Boolean functions), a series of synthesis and post-synthesis methods are employed to obtain quantum circuits. The most common method to obtain quantum circuits from reversible functions consists of several steps. First, a circuit of Toffoli gates [3], [4] is constructed. The Toffoli circuit is then optimized with such methods as described in [5], [6]. Next, the Toffoli circuit is transformed into a quantum circuit by decomposing [7] Toffoli gates. Template matching can be used to reduce the number of gates in Toffoli circuits [8] as well as quantum circuits [9]. When the Toffoli gate was first proposed [3], only positive controls were considered. However, it has been shown that using both positive and negative controls [10], [11] will result in circuits with lower quantum cost. This is the motivation for considering templates that traditionally only considered positive controls, with mixed polarity (positive and negative controls).

The objective of our work is to investigate the use and effect of mixed polarities in templated-based optimization. We first find a complete set of 2-line templates for NOT and CNOT gates with mixed controls. It turns out that the number of templates is very large. This has two disadvantages: first, a large number of templates must be stored; second, for each template the circuit must be scanned for a potential match. We develop a dynamic template matching that takes a set of templates for a few lines as input. For a successful match on a subset of lines in the circuit to be optimized it is checked whether the template can be extended to match the remaining lines. Note that the size of the set of matched gates must be more than half of the size of the template. Experiments show very good results in reducing the number of gates in circuits in comparison to the best known results of MCT benchmark circuits [12]. It is observed that the runtime of the proposed heuristic is high since many possibilities are checked by adding controls to the gates in a template.

Previous work has considered using negative control lines for reversible circuit optimization. Both [13] and [14] present approaches, also called template matching, to reduce the number of gates in reversible circuits. However, they neither present an algorithm to create identities nor classify circuit identities. Hence, the algorithms can better be described as rule-based approaches. This is further justified by the fact that such rules are presented in terms of a subcircuit and its optimized form, rather than in terms of an identity circuit. In fact, the technique described in [13] considers sequences of gates that share the same target line and can therefore be effeciently reduced using ESOP minimzation as demonstrated in [15]. In fact, all optimization approaches that take negative controls into account are

0195-623X/15 $31.00 © 2015 IEEE
DOI 10.1109/ISMVL.2015.44

IEEE computer society

instances of rewriting according to the elementary rules presented in [16]. This also explains the huge number of templates when considering negative controls and justifies a dynamic matching approach as presented in this paper.

The remainder of the paper is structured as follows: Section II briefly describes the fundamentals of reversible logic and template matching. In Section III we discuss the basis for constructing templates with positive and negative controls. Section IV describes the dynamic template matching method. The significance of the proposed approach is shown with experiments. All 3-line MCT circuits and 43 benchmarks are optimized and the results are reported in Section V. The paper concludes with some observations and directions for future research in Section VI.

II. Background

To keep this paper self-contained, this section briefly describes the essentials for reversible logic, reversible circuits, and circuit optimization using template matching.

A Boolean logic function $f : \mathbb{B}^n \to \mathbb{B}^n$ over a set of variables $X = \{x_1, \ldots, x_n\}$ is reversible if it is bijective. Classical reversible functions [1] are the special case of multiple-output Boolean functions. A mixed-polarity multiple-controlled Toffoli (MPMCT) gate $g(C, t)$ is a reversible gate consisting of a set of control lines $C \subset \{x, \bar{x} \mid x \in X\}$ that are literals of X and a target line $t \in X$ such that $\{t\} \cup C = \emptyset$. A literal can occur with at most one polarity in C. A control line is called positive if it occurs as positive literal in C and negative if it occurs as negative literal in C. The semantics of such a gate are as follows: The value at the target line is inverted if all literals evaluate to true. All remaining values are passed through the gate unchanged. Gates for which $|C| = 0$ and $|C| = 1$ are known as NOT and CNOT, respectively. In the literature, often only the case of gates which have solely positive control lines has been considered.

Reversible circuits are realized by cascading reversible gates. An MPMCT circuit is realized by cascading MPMCT gates. A circuit for a function is **minimal**, if no circuit with fewer gates realizes the same function. The size of a circuit G, denoted by $|G|$, is the number of gates in G.

Two adjacent gates $g_1(C_1, t_1)$ and $g_2(C_2, t_2)$ can be interchanged if $C_1 \cap \{t_2\} = \emptyset$ and $C_2 \cap \{t_1\} = \emptyset$. That is, a gate may move within the circuit by commuting gates; this is known as **moving rule** [10]. If gates in a circuit G can be brought together by the moving rule, then they form a subcircuit of G. Let two gates $g_1(C_1, t_1)$ and $g_2(C_2, t_2)$ in a circuit act on the same line(s) and they realize the functions f and f^{-1} respectively. If they can be made adjacent by using the moving rule, then they both can be deleted; this is known as the **deletion rule**. These rules origin from previous work in which only postive control lines were considered. By allowing negative control lines for the circuit description, one gains more flexibility [16].

A. Templates and Template Matching

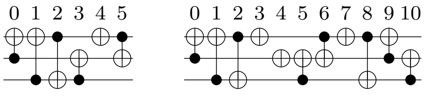
A **template** T is a circuit that realizes the identity function. If a sequence of gates in a circuit matches with a sequence of gates with size $s_1 > \lfloor |T|/2 \rfloor$ in a template T, then the matched sequence of gates in the circuit can be replaced with the inverse of the remaining sequence of size $s_2 < \lfloor |T|/2 \rfloor$. This is called **template matching**.

Gates in a reversible circuit can often be reordered without affecting the function of the circuit. The order of gates in a template are considered to be circular and the moving rule can also be applied. The process of matching gates from a template to a circuit can start from anywhere, and matching can be extended either in a forward or backward direction of the template. Hence, if a template can be applied to a circuit, by rearranging the gates in the circuit as well as in the template, then such match will be found in **exact template matching**. In this context, in exact template matching, a gate $g_1(C_1, t_1)$ of a template matches with a gate $g_2(C_2, t_2)$ of a circuit such that g_1 and g_2 realize the same function. An example of matching of the gates of a template to the gates of a circuit is shown in Fig. 1. However, a graph-based algorithm for exact template matching can be found in [17].

Example 1: Let the circuit shown in Fig. 1(a) be optimized with the template shown in Fig. 1(b). The gate sequence $0, 1, 2, 3, 4, 5$ of the template matches the gate sequence $0, 1, 2, 5, 3, 6$ in the circuit in exact template matching. Therefore, the gate sequence of optimized circuit would be the gate sequence $10, 9, 8, 7, 4$ of the template.

III. Base Templates and Rules for Template Construction

Templates can be constructed in many different ways by using rewriting rules. However, some empirical results



(a) Circuit G (b) Template T

Fig. 1: (a) A circuit and (b) A template

show that the set of templates of more than 2 lines can be very large. Moreover, applications of templates depend on how circuits to be optimized are constructed, and many templates may not be useful. For example, if a 3-line function is synthesized by using the transformation-based algorithm, then it is unlikely that the template in Fig. 2(i) is applied to reduced the obtained circuit. In this section, we first find a set of templates of 2 lines which are referred to as base templates. From these base templates, a set of rules is defined to reduce the template set. We also present rules of deriving templates with a large number of lines dynamically in template matching. A heuristic for dynamic template matching is described in the next section. For the NOT and CNOT gates with both positive and negative controls, an exhaustive search algorithm [18] finds a complete set of 2-line templates which are shown in Fig. 2.

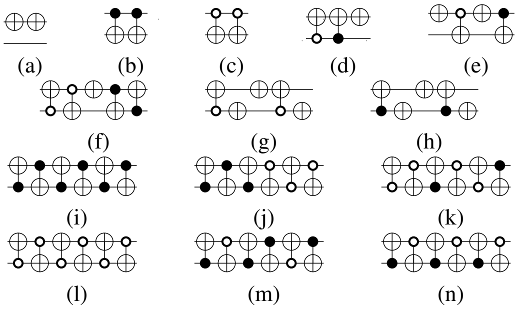


Fig. 2: The set of all 2-line templates

In contrast to dynamic template matching, in exact template matching, a gate $g_1(C_1, t_1)$ in a template can only match a gate $g_2(C_2, t_2)$ in a circuit if $|C_1| = |C_2|$ and both C_1 and C_2 have same number of positive and negative controls. That is, both g_1 and g_2 have to realize the same function up to variable names.

This has a consequence that a large set of templates must be known in advance for exact template matching. However, if templates with MPMCT gates are considered, then a reduced set of templates obtained by the rules in subsection III-A can be used in an exact template matching. Moreover, templates with gates of a large number of controls still require for optimizing circuits with gates of a large number of controls. On the other hand, the dynamic template matching takes a template with few lines as an input and if the lines of gates in the template partially match with the lines of gates in the circuit then it will extend the lines of the template. That is, a gate $g_1(C_1, t_1)$ in a template matches a gate $g_2(C_2, t_2)$ in a circuit if $|C_1| \leq |C_2|$ as well as both g_1 and g_2 do not necessarily have to realize the same function.

A. Equivalence of Templates

Let T be a l-line template of size n, then the following rules can be applied to reduce the set of templates.

1) All polarities of the controls of gates in a template can be flipped. For instance, the templates in Fig. 2(g) and (h) can be considered as equivalent in template matching.
2) On a line where no gate has a target, all polarities of the controls on that line can be flipped. For instance, the templates in Fig. 2(b) and (c) can be considered as equivalent.

According to these equivalence rules, the set of templates in Fig. 2 can be reduced to the set of templates which are shown in Fig. 2(a), (b), (d), (e), (f), (g), (i), (k), and (n).

B. Extension of Lines in Templates

Let T is an l-line template of size n, then the following rules can be applied to derived new templates.

1) Adding a line to T results in a template of $l + 1$ lines.
2) If the removal of a subset S of m ($2 \leq m \leq n$) gates in T results in an identity, then a new control of each gate of S in T can be added and placed into a new line. All new controls of gates in S must either all be positive or all be negative. This results in a new template.

According to these rules, it follows that the templates in Fig. 2(a), (b), and (c) can be considered as equivalent in dynamic template matching. In summary, we obtain a reduced set of templates as shown in Fig. 3.

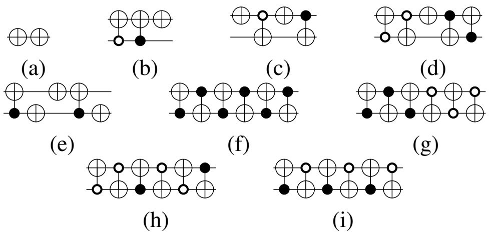


Fig. 3: The reduced set of 2-line templates

IV. Dynamic Template Matching

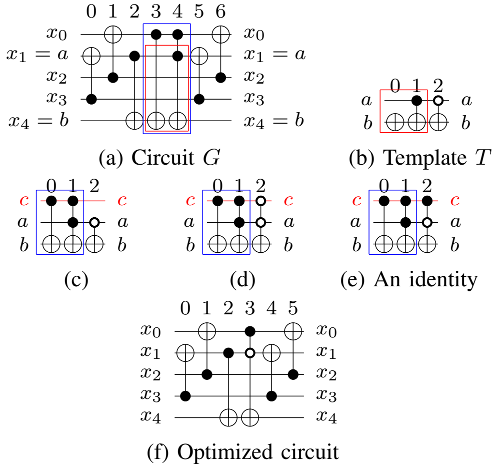
In this section, we present an algorithm for dynamic template matching. To illustrate the algorithm in Fig. 4, we consider the circuits in Fig. 5. Let the procedure **DynamicTemplateMatching()** take the circuit G in Fig. 5(a) and the template T in Fig. 5(b) as input; $n = 5$ and $l = 2$. The procedure **match()** returns true and sets the values of m, m_l, u_l, and u if more than half of the gates of I, can be mapped into a sub-circuit of G. For instance, for the gates $\{0, 1\}$ in the template T, lines a and b are mapped into the circuit lines x_1 and x_4. The gates in the inner rectangle in C are the candidates for matching with the gates $\{0, 1\}$ in T. Therefore, $m = \{3, 4\}$, $m_l = \{1, 4\}$, $u_l = \{0, 2, 3\}$, and $u = \{2\}$. Now for the line x_0, there are controls of gates in the set $m = \{3, 4\}$ in G. Therefore, the procedure **addConInNewLine**() will add a new line c to the circuit I and also add controls with respect to the gates set $m = \{3, 4\}$ and $i = 0$. The resulting circuit I is shown in Fig. 5(c) in which the unmatched gate 2 of T remains unchanged. Now for the gate $u = \{2\}$ in the circuit shown in Fig. 5(c), the procedure **addConcheckID**() checks whether any identity can be found by adding controls to the gate $u = \{2\}$ in I in all possible ways. In this case two possibly generated circuits are shown in Fig. 5(d) and (e). Since the circuit in Fig. 5(e) is an identity, the procedure **addConcheckID**() returns true. Since lines $\{2, 3\}$ in the circuit have no controls for the gates set $m = \{3, 4\}$, the inner for loop in the algorithm will terminate without changing $succ = true$. Now $succ = true$, hence, the procedure **doReplacement**() replaces the gates set $m = \{3, 4\}$ with the gate 2 of the identity circuit of Fig. 5(e). The resulting optimized circuit is shown in Fig. 5(f).

V. Experimental Results

We implemented exact template matching and the dynamic template matching heuristic using C++. Along with 2-line templates, we have also generated a set of 3-line templates of size up to 6. In our first experiment, we apply a total of 228 templates to optimize all 3-line circuits (gates in circuits have only positive controls) obtained from the transformation based synthesis algorithm. Column 3 in Table I shows the average number of gates for all 40320 functions. The average number of gates in circuits obtained from exact template matching are show in column 4 in Table I. The total average reduction is 24.18%. It is observed that in exact template matching,

```
circuit& DynamicTemplateMatching($G$, $T$)
    $G$ is a circuit of $n$ lines
    $T$ is a template $l$ lines
    /* Let m be an array for index of matched gates in the circuit */
    /* Let $u$ be an array for index of unmatched gates in the template */
    /* Let $m_l$ and $u_l$ be arrays of matched lines in the circuit where $|m_l| = l$ and $|u_l| = n - l$ */
    Let $I = T$ is a circuit
    while match($G$, $I$, &$m$, &$m_l$, &$u_l$, $u$)
        if $|m| = |I|$
            if isIdentity($G$, $m$) /* whether the whole sub-circuit for $m$ in $G$ is identity */
                $G$ = doReplacement($G$, $I$, $m$, $m_l$);
        else
            bool $succ = true$;
            for each $i \in u_l$
                if line $i$ has controls of the gates in $m$
                    $I$ = addConInNewLine($I, m, i$);
                    /* number of lines of $I$ is increased by only 1 */
                    $m_l = m_l \cup i$
                    if !addConcheckID($I, u$)
                        $succ = false$;
                        $break$;
            if succ
                $G$ = doReplacement($G$, $I$, $m$, $m_l$);
        $I = T$;
    return $G$;
```

Fig. 4: Algorithm for dynamic template matching



(a) Circuit G

(b) Template T

(c)

(d)

(e) An identity

(f) Optimized circuit

Fig. 5: Dynamic template matching

out of 228 templates, only 31 templates are applied.

In our second experiment, we have taken benchmark MCT circuits from RevLib [12] as shown in column 1 in Table II. Column 2 shows the number of gates in the original circuits. By using the reduced set of 2-line templates in Fig. 3 and 214 templates of 3 lines, the obtained results from dynamic template matching are shown in column 3 in Table II. According to column 5 in Table II, a significant percentage of reduction of gates can be noticed. More than 20% reductions are highlighted. For 43 benchmarks, the average reduction is 23.46%. The best result is 57% reduction for the benchmark $rd53\_131$. It can be noticed that for the benchmark $urf3\_155$ with $26,468$ gates, the number of gate reductions is $10,732$. Among 223 templates, those templates which are frequently used for benchmarks optimization is shown in Fig. 6. It is very clear that we only use templates of lines up to 3 in which targets of gates in templates are placed into at most 3 lines, and only controls of gates in templates are extended. However, templates in which gate target acts on more than 3 lines can be more beneficial for the optimization of circuits with a large number of lines. Even templates of size more than 6 can be investigated.

We observe that reduction of gates in Toffoli circuits may not lead to circuits with reduced quantum costs when naive gate count of quantum circuits is considered as quantum cost. Toffoli circuits with fewer gates may have higher quantum costs than the costs of original circuit. For instance, if the circuit shown in Figure 7(a) is replaced with the circuit shown in Figure 7(b), then the quantum cost of the resulting circuit increases where quantum costs of a Toffoli gate with 2 controls and a Toffoli gate with 3 controls are 5 and 14, respectively. The question is that whether Barenco's [7] or library-based Toffoli [19] decomposition would be beneficial to obtain possibly low quantum cost circuits. However, we conjecture that for a template T, if the gate sequence of size $\lfloor |T|/2 \rfloor + 2$ in T matches with the gate sequence of a circuit, then the resulting Toffoli circuit always leads to the low quantum cost circuit. In the literature, different quantum cost models are proposed [20], [21], therefore, in this work, we leave the measurement of quantum costs of circuits for future research.

TABLE I: Results of optimizing 3-line MCT circuits obtained from transformation-based algorithm

$Size(min)$	$\#f$	$\#Avg(G)TB$	$\#Avg(G)Optz$
0	1	0	0
1	12	1	1
2	102	2.44	1.9
3	625	4.18	2.97
4	2780	6.01	4.33
5	8921	7.65	5.73
6	17049	9.02	6.92
7	10253	9.94	7.64
8	577	10.75	8.18
Total	40320	6.37	4.83

$\#Avg(G)TB$: Average no. of gates in original circuits;
$\#Avg(G)Optz$: Average no. of gates in optimized circuits;

TABLE II: Results of optimizing benchmarks

Benchmarks	$\#G(Org.)$	$\#G(Optz.)$	Rd	$Rd(\%)$
4gt11-v1_85	4	3	1	**25.00**
4gt13-v1_93	4	3	1	**25.00**
4mod5-v0_19	5	3	2	**40.00**
4mod5-v1_24	5	4	1	20.00
mod5mils_65l	5	3	2	**40.00**
mod5mils_71	5	3	2	**40.00**
4gt12-v1_89l	5	4	1	20.00
3_17_13l	6	4	2	**33.33**
3_17_14l	6	5	1	16.67
decod24-v0_38	6	5	1	16.67
4gt4-v0_72	6	5	1	16.67
mod5d1_63	7	5	2	**28.57**
alu-v2_33	7	6	1	14.29
alu-v2_32	7	6	1	14.29
mod10_176	7	5	2	**28.57**
4mod5-v1_23	8	4	4	**50.00**
mod8-10_178	9	8	1	11.11
aj-e11_168	10	9	1	10.00
4gt13_91	10	9	1	10.00
4gt12-v0_87	10	9	1	10.00
mod10_171	10	9	1	10.00
4_49_17	12	10	2	16.67
aj-e11_165	13	11	2	15.38
alu-v2_31	13	9	4	**30.77**
mod8-10_177	14	11	3	21.43
rd53_137	16	13	3	18.75
4_49_16	16	13	3	18.75
rd53_135	16	13	3	18.75
hwb4_49	17	14	3	17.65
4gt4-v0_73	17	13	4	23.53
alu-v2_30	18	15	3	16.67
mod5adder_127	21	16	5	**23.81**
ham7_106	25	22	3	12.00
rd53_131	28	12	16	**57.14**
rd53_130	30	22	8	**26.67**
sym6_145	36	31	5	13.89
hwb7_62	331	294	37	11.18
hwb8_116	749	638	111	14.82
hwb9_123	1959	1701	258	13.17
urf2_152	5030	3274	1756	**34.91**
urf5_158	10276	5578	4698	**45.72**
urf1_149	11554	7347	4207	**36.41**
urf3_155	26468	15736	10732	**40.55**

$\#G(Org.)$: No. of gates in original circuits;
$\#G(Optz.)$: No. of gates in optimized circuits;
Rd: No. of gates reduced;
$Rd(\%)$: Percentage of reductions;

VI. Conclusion

We present a heuristic of dynamic template matching in which templates of more than 2 quibts have been generated during the matching process. The results from optimizing benchmark functions are promising, whereas computationally, it is not feasible to obtain all templates even with three lines before template matching starts. Relating our findings in template matching with mixed-polarity Toffoli gates to the results of [16] can help to devise a better understanding of template matching and estimate the number of total templates, but also

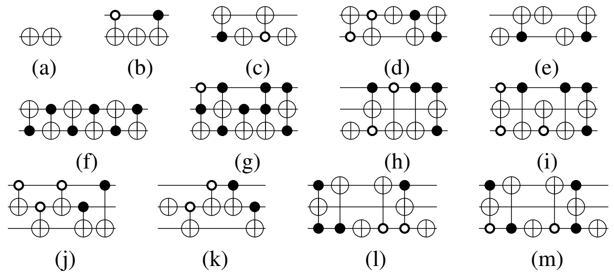


Fig. 6: The set of templates successfully used in benchmarks optimization

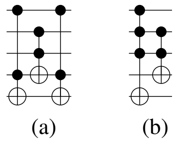


Fig. 7: Reversible circuits with (a) $\#gate = 3$ and quantum cost $15 = 5 + 5 + 5$ and (b) $\#gate = 2$ and quantum cost $19 = 14 + 5$.

to find rewriting strategies to guide circuit rewriting as an alternative to optimization. A cost-benefit analysis of circuits in quantum decomposition would be useful since the gates in the resulting circuits have both positive and negative controls. The heuristic of dynamic template matching can be improved by finding the best matches for given templates. Moreover, an in-depth analysis of the use of the 13 templates shown in Fig. 6 may lead to the development of algorithms for expansion of gates in templates, which is our future research.

REFERENCES

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

[2] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, 1997.

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

[4] E. Fredkin and T. Toffoli, "Conservative logic," *International Journal of Theoretical Physics*, vol. 21, pp. 219–253, 1982.

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

[6] D. Maslov and G. W. Dueck, "Reversible cascades with minimal garbage," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 23, no. 11, pp. 1497–1509, 2004.

[7] A. Barenco, C. H. Bennett, R. Cleve, D. DiVinchenzo, 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.

[8] D. Maslov, C. Young, G. W. Dueck, and D. M. Miller, "Quantum circuit simplification using templates," in *DATE - Design, Automation and Test in Europe*, 2005, pp. 1208–1213.

[9] M. M. Rahman and G. W. Dueck, "Template matching in quantum circuits optimization," in *Reed-Muller Workshop*, 2013, pp. 75–79.

[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*, vol. 27, no. 3, pp. 436–444, 2008.

[11] R. Wille, M. Soeken, N. Przigoda, and R. Drechsler, "Exact synthesis of toffoli gate circuits with negative control lines," in *Multiple-Valued Logic (ISMVL), 2012 42nd IEEE International Symposium on*. IEEE, 2012, pp. 69–74.

[12] R. Wille, D. Große, L. Teuber, G. W. Dueck, and R. Drechsler, "RevLib: An online resource for reversible functions and reversible circuits," in *Workshop on Reversible Computation*, July 2010, RevLib is available at http://www.revlib.org.

[13] K. Datta, G. Rathi, R. Wille, I. Sengupta, H. Rahaman, and R. Drechsler, "Exploiting negative control lines in the optimization of reversible circuits," in *Reversible Computation - 5th International Conference, RC 2013, Victoria, BC, Canada, July 4-5, 2013. Proceedings*, 2013, pp. 209–220.

[14] M. Z. Rahman and J. E. Rice, "Templates for positive and negative control Toffoli networks," in *Reversible Computation - 6th International Conference, RC 2014, Kyoto, Japan, July 10-11, 2014. Proceedings*, 2014, pp. 125–136.

[15] M. Soeken, Z. Sasanian, R. Wille, D. M. Miller, and R. Drechsler, "Optimizing the mapping of reversible circuits to four-valued quantum gate circuits," in *42nd IEEE International Symposium on Multiple-Valued Logic, ISMVL 2012, Victoria, BC, Canada, May 14-16, 2012*, 2012, pp. 173–178.

[16] M. Soeken and M. K. Thomsen, "White dots *do* matter: Rewriting reversible logic circuits," in *Reversible Computation - 5th International Conference, RC 2013, Victoria, BC, Canada, July 4-5, 2013. Proceedings*, 2013, pp. 196–208.

[17] M. M. Rahman, G. W. Dueck, and J. D. Horton, "An algorithm for quantum template matching," *J. Emerg. Technol. Comput. Syst.*, vol. 11, no. 3, pp. 31:1–31:20, Dec. 2014.

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

[19] 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.

[20] J. Smolin and D. DiVincenzo, "Five two-bit quantum gates are sufficient to implement the quantum Fredkin gate," *Physical review. A*, vol. 53, no. 4, pp. 2855–2856, 1996.

[21] D. Große, R. Wille, G. W. Dueck, and R. Drechsler, "Exact synthesis of elementary quantum gate circuits for reversible functions with don't cares," in *International Symposium on Multiple Valued Logic*, 2008, pp. 214–219.