Optimizing DD-based Synthesis of Reversible Circuits using Negative Control Lines

Eleonora Schönborn[1], Kamalika Datta[2], Robert Wille[1,3], Indranil Sengupta[4], Hafizur Rahaman[2], Rolf Drechsler[1,3]

[1]Institute of Computer Science, University of Bremen, 28359 Bremen, Germany
[2]Department of Information Technology, Bengal Engineering & Science University, Shibpur, Howrah 711103, India
[3]Cyber-Physical Systems, DFKI GmbH, 28359 Bremen, Germany
[4]Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur 721301, India
{eleonora,rwille,drechsle}@informatik.uni-bremen.de, kdatta.iitkgp@gmail.com, isg@iitkgp.ac.in, rahaman_h@it.becs.ac.in

***Abstract*—Synthesis of reversible circuits has attracted the attention of many researchers. In particular, approaches based on *Decision Diagrams* (DDs) have been shown beneficial since they enable the realization of corresponding circuits for large functions. However, all existing approaches rely on a gate library composed of positive control lines only. Recently, it has been shown that the additional use of negative control lines enables significant reductions of the respective circuit costs. In this paper, we aim for exploiting this potential. To this end, two complementary schemes are investigated. First, a post-synthesis optimization that exploits the power of negative control lines is utilized to optimize the circuits generated by previously proposed DD-based methods. Second, negative control lines are explicitly considered during synthesis. Experimental results demonstrate that the proposed approaches result in a significant reduction with respect to gate count as well as quantum costs.**

I. Introduction

Recent interest in the synthesis and optimization of reversible logic circuits has been motivated by the emphasis on low-power design alternatives and developments in quantum computing. It is believed that reversible circuits may offer a design alternative with promising low-power properties supported by the well-known principles investigated by Landauer [1] and Bennett [2] which recently have been experimentally confirmed in [3]. In the design of low-power interconnect encoders, reversible logic also found application for today's technologies [4]. Applications in the domain of quantum computation are motivated by the fact that each quantum operation is inherently reversible. Quantum circuits in turn are interesting as they enable to solve important tasks such as factorization in significantly less run-time than the currently best known conventional solutions [5].

Reversible circuits are constructed by creating a cascade of basic reversible gates, like NOT, controlled NOT [6], or Toffoli gates [7], with additional constraints like no direct support of fanout and feedback. Because of these constraints as well as the new gate library, synthesis of reversible circuits significantly differs from the design of conventional circuits.

Consequently, new approaches for the synthesis of reversible circuits have been explored by researchers. These include

- exact methods [8] for obtaining optimal circuits, which, due to their computational complexity, work for very small functions only,
- constructive approaches [9], [10] which are able to synthesize relatively large functions (with up to 30 inputs), and
- methods based on *Decision Diagrams* (DDs, [11]) or *Exclusive Sum of Products* (ESOPs, [12]) which enable synthesis for very large functions.

In these methods, the given function to be synthesized is represented using different function descriptions such as truth tables, DDs, or ESOPs. In the following, we focus on DD-based synthesis. Here, a hierarchical approach is applied which uses a DD to represent the function to be synthesized and transforms each node into a corresponding sub-circuit (this approach is reviewed in more detail later in Section III). Thus far, all existing approaches following this scheme (such as [11], [13], [14]) rely on a gate library composed of Toffoli gates with positive control lines only. Recently, an extension of these gates with mixed control lines, i.e. with both positive and negative control lines, received attention. It has been shown that additionally considering negative control lines enables the synthesis of reversible circuits with significantly less costs [15]–[18]. However, these recent findings have not yet been exploited for DD-based synthesis.

In this work, we investigate the potential of utilizing negative control lines for DD-based synthesis. To this end, we consider

- how the application of an existing (post-synthesis) optimization approach utilizing negative control lines improves the circuit realizations obtained by DD-based synthesis, and
- how negative control lines can explicitly be exploited during the synthesis.

Both schemes have been evaluated. The results clearly show that the utilization of negative control lines significantly reduces the costs of the respective circuits. In the best cases, up to 43% of the gates and 15% of the quantum costs can be saved.

978-1-4799-4558-0/14/$31.00 © 2014 IEEE

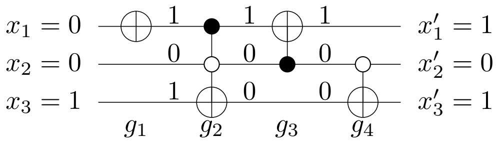


Fig. 1: Reversible circuit

The remainder of this paper is organized as follows. Section II provides the background on reversible circuits while Section III briefly reviews DD-based synthesis and motivates this work. Afterwards, the post-synthesis optimization scheme and the the explicit consideration of negative control lines in DD-based synthesis are discussed in Section IV and Section V, respectively. Finally, Section VI summarizes the experimental evaluation and Section VII concludes the paper.

II. Reversible Circuits

A Boolean function $f : \mathbb{B}^n \to \mathbb{B}^n$ is said to be *reversible* if it represents a one to one mapping, i.e. is bijective. The problem of *synthesis* is to determine a reversible circuit that realizes a given function f.

Reversible circuits are digital circuits with the same number of input signals and output signals. They realize reversible functions and differ from conventional circuits, since e.g. fanout and feedback are not directly allowed [5]. Consequently, reversible circuits are composed as a cascade of reversible gates. In this work, we consider the established *Toffoli gate* library.

A Toffoli gate over the inputs $X = \{x_1, \ldots, x_n\}$ consists of a (possibly empty) set of *control lines* $C = \{x_{i_1}, \ldots, x_{i_k}\} \subset X$ and a single *target line* $x_j \in X \setminus C$. The Toffoli gate inverts the value on the target line iff all values on the control lines are assigned 1 or if $C = \emptyset$, respectively. All remaining values are passed through unaltered.

Recently, an extension of Toffoli gates with *mixed control lines* received attention. In addition to the (positive) control lines as defined above, they may also include *negative control lines*. The functionality of such Toffoli gates is the same as defined above, except that the value on the target line is inverted iff all values on positive control lines are assigned 1 and all values on negative control lines are assigned 0.

As an example, Fig. 1 shows a reversible circuit with three lines and composed of four gates. The target lines are denoted by $\bigoplus$, while a ● represents a positive control line and a ○ represents a negative control line. For example, assigning the input pattern 001 to the circuit results in the output pattern 101. Due to the reversibility, this computation can be performed in both directions.

To estimate the cost of an implementation, several metrics are used. Most common is counting the number of gates and the number of equivalent basic quantum operations (so called *quantum costs*). The latter is derived by mapping a reversible circuit into an equivalent cascade of quantum gates. For this purpose, several schemes have been introduced which have their origin in the initial work by Barenco et al. [19]. The mappings proposed in [20] provide the best known mappings thus far. While these mappings mainly considered Toffoli gates with positive control lines only, respective schemes for negative control lines have been discussed in [21]. Here, it has been shown that the same calculation as for positive controls can be applied except for the case where *all* the controls are negative. In this case, the determined quantum costs have to be increased by 1 (in case of a Toffoli gate with up to two control lines) or 2 (in all other cases).

III. DD-based Synthesis

DD-based synthesis is a hierarchical synthesis approach which enables the automatic generation of a reversible circuit realizing a given function f. To this end, the function f to be synthesized is decomposed into smaller sub-functions. This decomposition is repeatedly applied until the sub-functions evaluate to a constant 0 or 1. By this, the (possibly very large) function f is represented by a logical combination of co-factors. While the overall function f is usually hard to synthesize in one step, the respective co-factors as well as logical combinations resulting from the decomposition are rather small and, hence, can easily be realized as sub-circuits. Composing all these sub-circuits eventually results in a circuit realizing the desired function f.

This scheme has originally been applied in [11] and further refined e.g. in [13], [14]. The decompositions have been conducted by the use of data structures like *Binary Decision Diagrams* (BDDs, [22]) or *Kronecker Functional Decision Diagrams* (KFDDs, [23], [24]). Both are directed, acyclic graphs $G = (V, E)$ with a root that represents the function f. Each inner node $v \in V$ has two child nodes low(v) and high(v) representing the sub-functions obtained by the decomposition. Possible decompositions are defined by:

$$f = \overline{x}_i \cdot f_{x_i=0} + x_i \cdot f_{x_i=1} \qquad \text{(Shannon)}$$
$$f = f_{x_i=0} \oplus x_i \cdot (f_{x_i=0} \oplus f_{x_i=1}) \qquad \text{(positive Davio)}$$
$$f = f_{x_i=1} \oplus \overline{x}_i \cdot (f_{x_i=0} \oplus f_{x_i=1}) \qquad \text{(negative Davio)}$$

Each inner node is labeled with a variable of f and each variable is assigned a decomposition type. For example, if a node representing the function f is labeled with the variable x_i which is assigned Shannon decomposition, its child nodes represent $f_{x_i=0}$ (low(v)) and $f_{x_i=1}$ (high(v)), where $f_{x_i=0}$ ($f_{x_i=1}$) is the negative (positive) co-factor of f obtained by assigning x_i to 0 (1). Co-factors evaluating to the constant 0 or 1 are represented by terminal nodes.

Note that BDDs only allow Shannon decomposition, while KFDDs support all decompositions mentioned above. In this sense, BDDs are a special case of KFDDs. Due to the reduced diagram complexity, algorithms for BDDs are often more efficient than those for KFDDs. On the other hand, KFDDs allow for a more compact representation of certain types of Boolean functions [25]. In the following, we generically denote these data structures by *Decision Diagrams* (DDs).

For an even more compact representation of functions in a DD, *complement edges* have been introduced. If a complement edge is pointing to a node v, the function $\overline{f_v}$ rather than the function f_v is used. These edges are denoted by a • in the following figures.

Taking all that into consideration, synthesis for a given function f represented by a DD $G = (V, E)$ can be performed by conducting the following steps:

1) Traverse the DD in a depth-first manner.

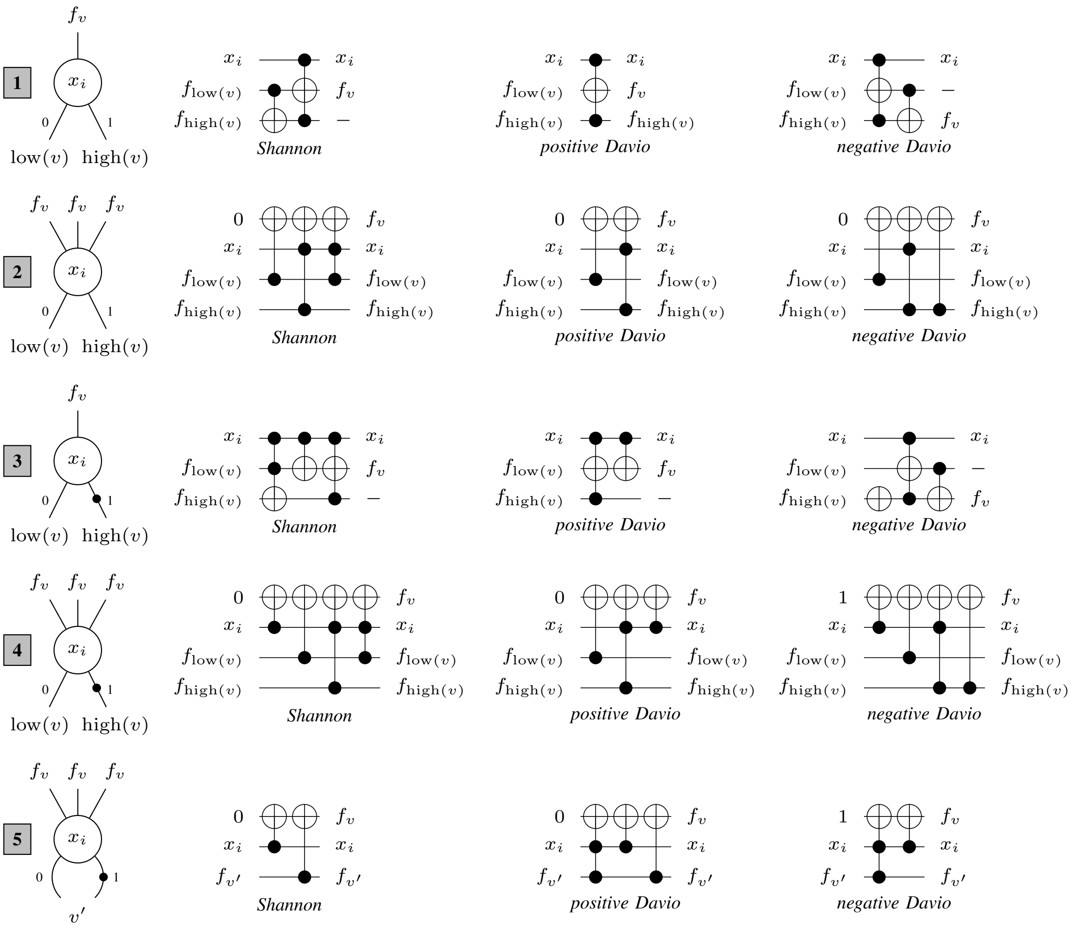


Fig. 2: Reversible cascades representing the different decompositions

2) For each inner node $v \in V$, generate a cascade of reversible gates which computes the (sub-)function represented by v. Output values of the previously traversed child nodes of v are utilized for this purpose.
3) Cascade all generated sub-circuits which eventually leads to a circuit realizing f.

The sub-circuits generated in Step 2 vary depending on decomposition type, types of child nodes, use of complement edges, etc. Fig. 2 provides a selection of cases that may occur in a DD together with their corresponding circuit patterns.

Example 1: Fig. 3a shows a DD representing the function $f = \overline{x}_1\overline{x}_2\overline{x}_3x_4 + \overline{x}_1x_2x_3\overline{x}_4 + x_1\overline{x}_2x_3\overline{x}_4 + x_1x_2\overline{x}_3x_4$ as well as the respective co-factors resulting from the application of the Shannon decomposition. The co-factor f_1 can easily be represented by the primary input x_4. Having the value of f_1 available, the co-factor f_2 can be realized by the first two gates depicted in Fig. 3b[1]. In this fashion, respective sub-circuits can be added for all remaining co-factors until a circuit representing the overall function f results. The remaining steps are shown in Fig. 3b.

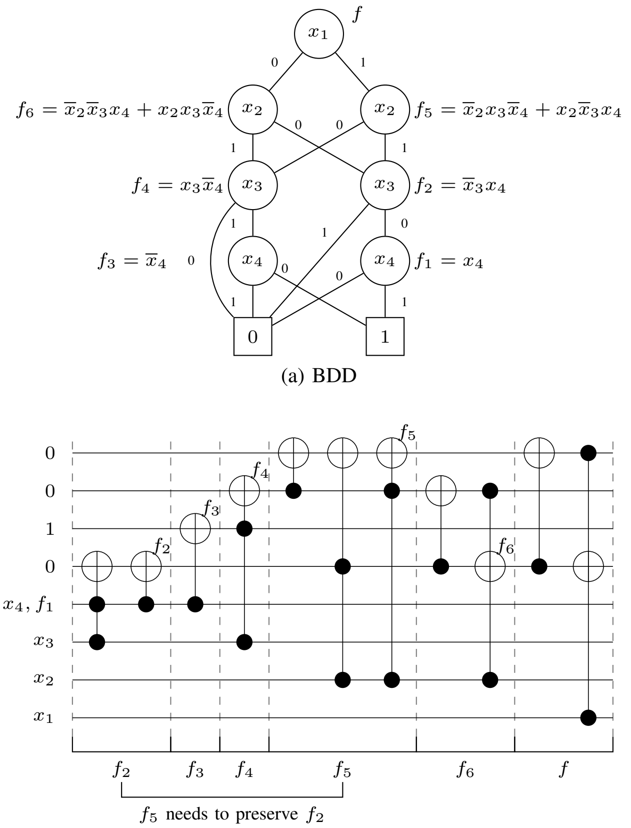
Thus far, only positive control lines have been considered in the DD-based synthesis. But as shown in previous work such as [15]–[18], additionally utilizing negative control lines may significantly reduce the number of gates as well as the resulting quantum costs of a reversible circuit. However, the utilization of negative control lines during DD-based synthesis has not been investigated yet. Because of this, significant potential for the improvement of DD-based synthesis has not been exploited. In particular, the realization of complement edges or negative Davio decomposition may significantly profit from negative control lines.

In this work, these missing investigations and evaluations are performed. To this end, two complementary schemes are considered. First, it is evaluated whether corresponding post-synthesis approaches presented in the past can be applied in order to improve circuits obtained by DD-based synthesis. Second, negative control lines are explicitly considered during synthesis, i.e. an extended DD-based synthesis approach is proposed which directly applies negative control lines when mapping from nodes to sub-circuits. Experimental evaluations summarized in Section VI confirm that both schemes lead to significant improvements.

IV. Post-Synthesis Optimization

Synthesis and optimization of reversible logic circuits have gained lots of attention in the past. As circuits generated from certain synthesis approaches require a large number of gates, there is a huge scope for post-synthesis optimization. In the literature, most of the optimization techniques relied on a gate library composed of positive control Toffoli gates only.

[1] Note that an additional circuit line is added to preserve the values of x_4 and x_3 which are still needed by the co-factors f_3 and f_4, respectively.


f
x_1
$f_6 = \overline{x}_2\overline{x}_3x_4 + x_2x_3\overline{x}_4$
$f_5 = \overline{x}_2x_3\overline{x}_4 + x_2\overline{x}_3x_4$
$f_4 = x_3\overline{x}_4$
$f_2 = \overline{x}_3x_4$
$f_3 = \overline{x}_4$
$f_1 = x_4$
x_2
x_3
x_4
0
1
x_4, f_1
x_3
x_2
x_1
f_2
f_3
f_4
f_5
f_6
f
f_5 needs to preserve f_2

(a) BDD

(b) Resulting circuit

Fig. 3: Illustration of BDD-based synthesis

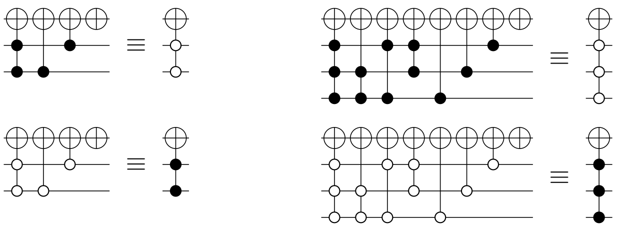


Fig. 4: Optimization rules (taken from [17])

However, in a recent work [17] it was shown that the power of negative control lines in Toffoli gates can be used quite elegantly to frame a set of template matching rules. Then, these rules can be applied in order to optimize a given reversible circuit. Some of these rules that may be used are illustrated in Fig. 4. Detailed experimental evaluations on circuits derived from various synthesis approaches demonstrated that significant reductions in the gate count and the quantum costs are possible when applying these rules.

Due to the nature of these rules, respective optimization methods usually perform better for circuits which inherit a specific structure, e.g. a clear separation between input lines and output lines. In previous work, this has successfully been shown on circuits generated by ESOP-based synthesis approaches (see [12] for a general description of ESOP-based synthesis and [17] for an evaluation on the corresponding post-synthesis optimization). However, an evaluation on circuits obtained by DD-based synthesis has not explicitly been conducted yet. Since also DD-based circuits inherit a rather regular structure, similar improvements are very likely. This is evaluated in detail later in Section VI.

V. Explicit Consideration of Negative Control Lines During Synthesis

In DD-based synthesis, negative control lines can explicitly be exploited for two purposes:

- Negative Davio decomposition can be realized in a similar fashion as positive Davio decomposition. They only differ in the polarity of the respective x_i-variable which, thanks to a negative control line, can easily be considered. This may lead to improvements since, as shown in Fig. 2, positive Davio can usually be realized with less gates and/or costs than negative Davio.
- Complemented edges can inherently be realized by negative control lines. In fact, complement edges are applied when the value of a sub-function to be considered shall be applied inversely. Again, this can easily be realized by the simple application of a negative control line, while, thus far, often additional logic has been required.

These observations are also confirmed by the realizations of the respective sub-circuits. More precisely, Fig. 5 shows the circuit realizations for all the cases previously discussed in Fig. 2 which additionally make use of negative control lines (the respective circuits have been obtained by the exact approach from [15] and represent minimal realizations with respect to the number of gates). Note that, due to page limitations, not all cases which might occur in DDs could be enlisted in a pictorial fashion. Nevertheless, Table I lists the number of gates and the quantum costs for *all* possible cases[2] and, by this, allows a comprehensive comparison. Columns denoted by d provide the number of gates, while columns denoted by *QC* provide the respective quantum costs. Both columns are additionally distinguished between values obtained if positive control lines are considered only (*pc*) and if negative control lines are considered additionally (*mc*). The last rows (*Total*) list the sum of gates and quantum costs that could be saved considering negative control lines.

As can be seen, most of the patterns could be improved with respect to gate count and quantum costs. Especially the cases with complement edges (Table Ib) unveil significant improvements. Interestingly, even some smaller realizations for the Shannon decomposition can be determined. In most of the cases, one gate – sometimes even two gates – can be saved. Quantum costs are improved by up to 4 in the best case. Considering that relatively small sub-circuits are considered which, however, are repeatedly applied during DD-based synthesis, this constitutes a significant improvement.

VI. Experimental Evaluation

The concepts and approaches discussed above have been evaluated. For this purpose, the post-synthesis optimization scheme proposed in [17] has been applied. Additionally, the *RevKit*-implementations (taken from [26]) of the BDD-based and the KFDD-based synthesis approaches [11], [13] have been extended by the new cascades which are partially sketched in Fig. 5. All these approaches have eventually

[2] Some node patterns are redundant and therefore not listed.

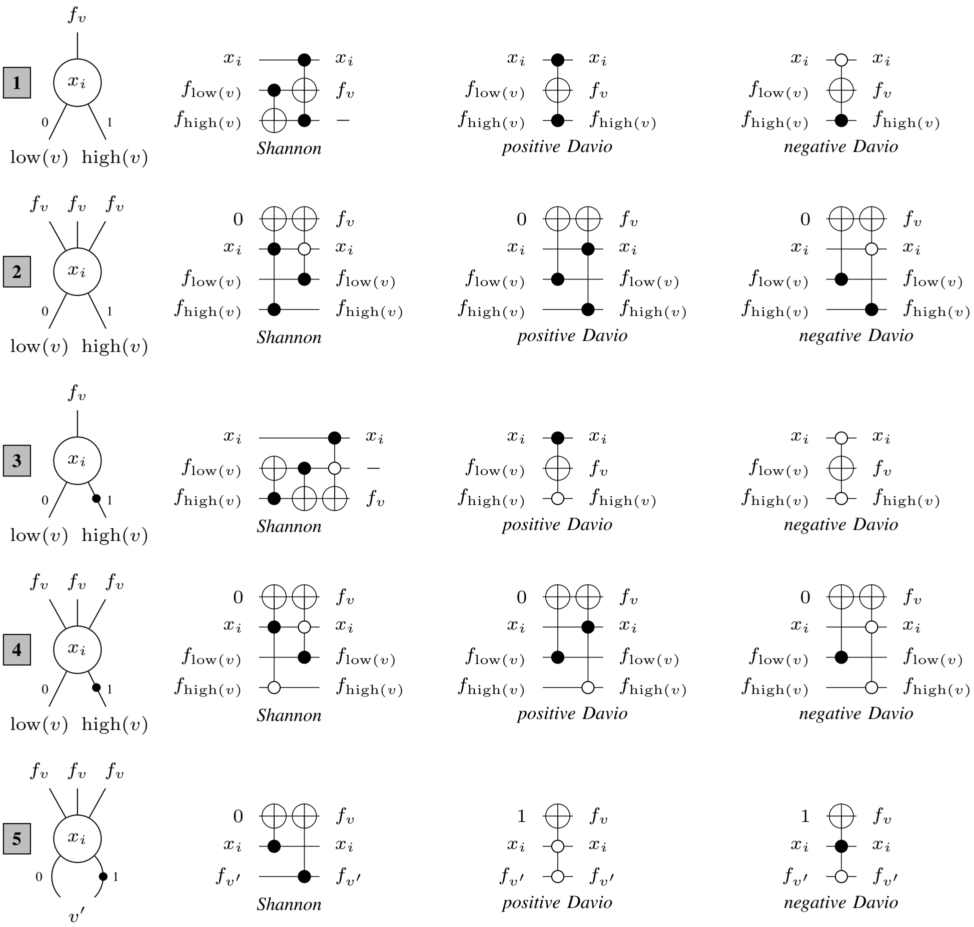


Fig. 5: Reversible cascades (w/ negative control lines) representing the different decompositions

TABLE I: Gate count/quantum costs for all decompositions

(a) W/o complement edges

	d		QC	
case	pc	mc	pc	mc
LL_pD	2	**1**	6	**5**
LL_nD	1	1	5	5
LH_S_s	3	**2**	11	**10**
LH_pD_s	2	2	6	6
LH_nD_s	3	**2**	7	**6**
LH_S	2	2	6	6
LH_pD	1	1	5	5
LH_nD	2	**1**	6	**5**
1H_S	2	**1**	6	**5**
1H_pD	1	1	5	5
1H_nD	2	**1**	6	**5**
0H_S	1	1	5	5
L1_S	3	**1**	7	**6**
L1_pD	2	2	2	2
L1_nD	2	2	2	2
L0_S	2	**1**	6	**5**
11_pD	1	1	1	1
10_S	1	1	1	1
Total		9		8

(b) W/ complement edges

	d		QC	
case	pc	mc	pc	mc
L-L_S_s	2	2	2	2
L-L_S	1	1	1	1
-LL_S_s	2	2	2	2
-LL_S	2	**1**	2	2
L-L_pD	3	**1**	7	**6**
L-L_nD	2	**1**	6	**5**
L-H_S_s	4	**2**	12	**10**
-LH_S_s	4	**2**	12	**10**
L-H_pD_s	3	**2**	7	**6**
L-H_nD_s	4	**2**	8	**7**
L-H_S	3	3	11	**7**
-LH_S	3	3	11	**8**
L-H_pD	2	**1**	6	**5**
L-H_nD	3	**1**	7	**6**
1-H_S	1	1	5	5
1-H_pD	2	**1**	6	**5**
1-H_nD	3	**1**	7	**6**
-L1_S	2	**1**	6	**5**
-L0_S	3	**1**	7	**6**
Total		20		21

been evaluated using a set of benchmark functions taken from *RevLib* [27].

Table IIa and Table IIb summarize the results for BDD-based synthesis and KFDD-based synthesis, respectively. The first two columns denote the name of the considered function as well as the number n of circuit lines generated by the respective approaches. Then, the remaining columns provide the number of gates (denoted by d) as well as the quantum costs (denoted by *QC*) of the circuits obtained by the original approach (i.e. the original BDD-based or KFDD-based synthesis) as well as the circuits obtained by applying the post-synthesis scheme (as discussed in Section IV) and the extended approach (presented in Section V). The columns *Impr.* provide the improvements with respect to the original realizations. All results have been generated in neglicable run-time, i.e. just a fraction of a second in most of the cases; the post-synthesis optimization scheme sometimes required slightly more time, but never more than 10 CPU seconds.

The results confirm the discussions from Section III: The utilization of negative control lines significantly reduces the number of gates as well as the resulting quantum costs and, hence, indeed improves DD-based synthesis. In the best cases, up to 43% of the gates and 15% of the quantum costs can be saved. The improvements of circuits obtained by KFDD-based synthesis are somewhat slight. This can be explained by the fact that KFDD decomposition already leads to smaller circuits. Nevertheless, relevant improvements can also be observed here. Considering that these improvements come with no drawbacks, the application of negative control lines is a worthwhile addition to DD-based synthesis schemes.

TABLE II: Experimental results

(a) BDD

		Original [11]		Post-Synth. (Sect. IV)				Explicit (Sect. V)			
						Impr.				Impr.	
Function	n	d	QC	d	QC	d	QC	d	QC	d	QC
alu2_96	105	452	1436	358	1346	21%	6%	323	1233	29%	14%
alu4_98	541	2186	7222	1746	6784	20%	6%	1554	6476	29%	10%
apex2_101	498	1746	5922	1358	5534	22%	7%	1238	5462	29%	8%
apex5_104	1025	2909	10349	2246	9686	23%	6%	2059	9461	29%	9%
ex5p_154	206	647	1843	462	1659	29%	10%	372	1612	43%	13%
frg2_161	1219	3724	12468	2753	11497	26%	8%	2611	11404	30%	9%
hwb8_64	112	449	1461	346	1360	23%	7%	319	1289	29%	12%
hwb9_65	170	699	2275	540	2117	23%	7%	488	2001	30%	12%
seq_201	1617	5990	19362	4561	17935	24%	7%	3950	17390	34%	10%
spla_202	489	1709	5925	1321	5537	23%	7%	1217	5420	29%	9%
urf1_72	374	1848	6080	1441	5673	22%	7%	1354	5199	27%	14%
urf2_73	209	983	3187	764	2968	22%	7%	703	2720	28%	15%
urf3_75	668	3413	11357	2674	10618	22%	7%	2533	9743	26%	14%
urf5_76	216	860	2796	679	2616	21%	6%	607	2432	29%	13%
Average						23%	7%			30%	11%

(b) KFDD

		Original [13]		Post-Synth. (Sect. IV)				Explicit (Sect. V)			
						Impr.				Impr.	
Function	n	d	QC	d	QC	d	QC	d	QC	d	QC
alu2_96	107	326	894	257	836	21%	6%	212	806	35%	10%
alu4_98	452	1252	5216	1239	5206	1%	1%	1238	5243	1%	0%
apex2_101	394	949	3621	870	3555	8%	2%	823	3661	13%	-1%
apex5_104	1029	2088	9092	2011	9019	4%	1%	1984	9023	5%	1%
ex5p_154	202	419	1503	374	1459	11%	3%	367	1525	12%	-1%
frg2_161	1252	3311	9023	2587	8361	22%	7%	1920	8308	42%	8%
hwb8_64	115	337	1297	310	1270	8%	2%	334	1300	1%	0%
hwb9_65	170	513	1993	472	1952	8%	2%	510	1996	1%	0%
seq_201	828	2041	6469	1699	6200	17%	4%	1403	6214	31%	4%
spla_202	458	1116	3760	984	3644	12%	3%	854	3728	23%	1%
urf1_72	379	1614	4202	1278	3866	21%	8%	1312	3954	19%	6%
urf2_73	203	736	2420	581	2266	21%	6%	571	2290	22%	5%
urf3_75	665	2625	9149	2307	8831	12%	3%	2465	9048	6%	1%
urf5_76	207	700	1876	508	1687	27%	10%	457	1679	35%	11%
Average						14%	4%			18%	3%

VII. Conclusion

In this paper, we investigated the potential of utilizing negative control lines for DD-based synthesis. To this end, a post-synthesis scheme as well as an explicit consideration during synthesis have been inspected and evaluated. Experiments confirmed the expected improvements: Negative control lines indeed allow for the realization of reversible circuits with significantly less gate count and quantum costs. In the best cases, up to 43% of the gates and 15% of the quantum costs can be saved.

Acknowledgment

This work has been supported by the Graduate School SyDe (funded by the German Excellence Initiative within the University of Bremen's institutional strategy), the Department of Science and Technology (DST), and the German Academic Exchange Service (DAAD). We would like to thank Nils Przigoda for his support during the development of this approach.

References

[1] R. Landauer, "Irreversibility and heat generation in the computing process," *IBM Journal of Research and Development*, vol. 5, no. 3, pp. 183–191, Jul. 1961.

[2] C. H. Bennett, "Logical reversibility of computation," *IBM Journal of Research and Development*, vol. 17, no. 6, pp. 525–532, Nov. 1973.

[3] A. Berut, A. Arakelyan, A. Petrosyan, S. Ciliberto, R. Dillenschneider, and E. Lutz, "Experimental verification of Landauer's principle linking information and thermodynamics," *Nature*, vol. 483, pp. 187–189, 2012.

[4] R. Wille, R. Drechsler, C. Oswald, and A. Garcia-Ortiz, "Automatic design of low-power encoders using reversible circuit synthesis," in *Design, Automation and Test in Europe*, 2012, pp. 1036–1041.

[5] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*. New York, NY, USA: Cambridge University Press, October 2000.

[6] R. Feynman, "Quantum mechanical computers," *Optic News*, vol. 11, pp. 11–20, 1985.

[7] T. Toffoli, "Reversible computing," in *Automata, Languages and Programming*, Jul. 1980, pp. 632–644.

[8] D. Große, R. Wille, G. W. Dueck, and R. Drechsler, "Exact multiple-control Toffoli network synthesis with SAT techniques," *IEEE Trans. on CAD*, vol. 28, no. 5, pp. 703–715, May 2009.

[9] D. Maslov, G. W. Dueck, and D. M. Miller, "Toffoli network synthesis with templates," *IEEE Trans. on CAD*, vol. 24, no. 6, pp. 807–817, 2005.

[10] P. Gupta, A. Agrawal, and N. K. Jha, "An algorithm for synthesis of reversible logic circuits," *IEEE Trans. on CAD*, vol. 25, no. 11, pp. 2317–2329, 2006.

[11] R. Wille and R. Drechsler, "BDD-based synthesis of reversible logic for large functions," in *Design Automation Conf.*, Jul. 2009, pp. 270–275.

[12] K. Fazel, M. A. Thornton, and J. Rice, "ESOP-based Toffoli gate cascade generation," in *Pacific Rim Conf. on Communications, Computers and Signal Processing*, 2007, pp. 206–209.

[13] M. Soeken, R. Wille, and R. Drechsler, "Hierarchical synthesis of reversible circuits using positive and negative Davio decomposition," in *Int'l Design and Test Workshop*, 2010, pp. 143–148.

[14] R. Wille and R. Drechsler, "Effect of BDD optimization on synthesis of reversible and quantum logic," *Electronic Notes in Theoretical Computer Science*, vol. 253, no. 6, pp. 57–70, 2010.

[15] R. Wille, M. Soeken, N. Przigoda, and R. Drechsler, "Exact synthesis of Toffoli gate circuits with negative control lines," in *Int'l Symp. on Multiple-Valued Logic*, May 2012, pp. 69–74.

[16] K. Datta, I. Sengupta, and H. Rahaman, "Particle swarm optimization based reversible circuit synthesis using mixed control Toffoli gates," *Journal of Low Power Electronics*, vol. 9, no. 3, pp. 363–372, October 2013.

[17] K. Datta, G. Rathi, R. Wille, I. Sengupta, H. Rahaman, and R. Drechsler, "Exploiting negative control lines in the optimization of reversible circuits," in *Int'l Conf. on Reversible Computation*, July 2013, pp. 209–220.

[18] M. Soeken and M. K. Thomsen, "White dots do matter: Rewriting reversible logic circuits," in *Int'l Conf. on Reversible Computation*, 2013, pp. 196–208.

[19] 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.

[20] D. M. Miller, R. Wille, and Z. Sasanian, "Elementary quantum gate realizations for multiple-control Toffoli gates," in *Int'l Symp. on Multiple-Valued Logic*, 2011, pp. 288–293.

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

[22] R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE Trans. on Computers*, vol. 35, no. 8, pp. 677–691, 1986.

[23] R. Drechsler, A. Sarabi, M. Theobald, B. Becker, and M. A. Perkowski, "Efficient representation and manipulation of switching functions based on ordered Kronecker functional decision diagrams," in *Design Automation Conf.*, 1994, pp. 415–419.

[24] R. Drechsler and B. Becker, "Ordered Kronecker functional decision diagrams-a data structure for representation and manipulation of Boolean functions," *IEEE Trans. on CAD*, vol. 17, no. 10, pp. 965–973, Nov. 2006.

[25] B. Becker, R. Drechsler, and R. Werchner, "On the relation between BDDs and FDDs," in *Information and Computation*, 1995, pp. 72–83.

[26] M. Soeken, S. Frehse, R. Wille, and R. Drechsler, "RevKit: an open source toolkit for the design of reversible circuits," in *Reversible Computation 2011*, ser. Lecture Notes in Computer Science, vol. 7165, 2012, pp. 64–76, RevKit is available at www.revkit.org.

[27] 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 Multiple-Valued Logic*, 2008, pp. 220–225, RevLib is available at www.revlib.org.