An Extension of Transformation-based Reversible and Quantum Circuit Synthesis

Mathias Soeken
Integrated Systems Laboratory
EPFL, Switzerland
mathias.soeken@epfl.ch

Gerhard W. Dueck, Md. Mazder Rahman
Faculty of Computer Science
University of New Brunswick, Canada
{gdueck,md._mazder.rahman}@unb.ca

D. Michael Miller
Department of Computer Science
University of Victoria, Canada
mmiller@uvic.ca

***Abstract*—Transformation-based synthesis is a well established systematic approach to determine a circuit implementation from a reversible function specification. Due to the inherent bidirectionality of reversible circuits the basic method can be applied in a bidirectional manner. In the approaches to date, gates are added either to the input side or the output side of the circuit on each iteration. In this paper, we introduce a new variation where gates may be added at both ends during a single iteration when this is advantageous to reducing the cost of the circuit. Experimental results show the advantage of the new approach over previous transformation-based synthesis methods and that the additional computation is justified by the possibility of improved circuit costs.**

I. Introduction

Reversible circuits have been widely studied because of their potential for reducing power consumption [1] and because of their connection to quantum computation [2]. A key issue is the synthesis of reversible circuits. This is a difficult problem that is significantly different from the synthesis of circuits employing traditional irreversible gates. Exact synthesis is only possible for very small circuits. The methods discussed here are heuristic and greedy relying on local optimization.

Transformation-based reversible circuit synthesis was introduced in [3] where two algorithms, a basic (unidirectional) approach and a bidirectional extension, were given. The bidirectional approach exploits the reversibility of both gates and circuits and while the basic algorithm builds a circuit in only one direction, typically from output to input, the bidirectional approach builds the circuit from both the output and the input sides. The bidirectional approach outperforms the basic algorithm.

In this paper, we introduce a new generalized approach which as experimental results will show can yield significantly smaller circuits in some cases. On each iteration, the bidirectional algorithm adds gates to either the input or the output side of the circuit, but not both. The new algorithm better exploits the structure of the reversible function being synthesized and can add gates at both ends on each iteration when that is of advantage.

II. Background

A multiple-output Boolean function is ***reversible*** if it maps each input assignment to a unique output assignment, i.e. it is a bijection; otherwise the function is ***irreversible***. A reversible function can be realized by a cascade of reversible gates with no fan-out or feedback [2]. A completely or incompletely-specified irreversible function can be embedded into a reversible function, usually with more inputs (constants) and outputs (garbage), and then realized by a reversible circuit [4].

A ***multiple-control Toffoli*** (MCT) gate with ***target line*** x_j and ***control lines*** $\{x_{i_1}, x_{i_2}, \ldots, x_{i_k}\}$, maps $(x_1 \ldots x_j \ldots x_n)$ to $(x_1 \ldots (x_{i_1} x_{i_2} \cdots x_{i_k}) \oplus x_j \ldots x_n)$. Note that all controls must be 1 (positive) for the target to be inverted. An MCT gate with no control is a **NOT** gate. An MCT gate with a single control line is called a **controlled-NOT** (CNOT) gate. Note that while we do not consider the option of negative controls (i.e. controls sensitive to 0 rather than 1), the methods discussed here can be extended to allow them. Also, while in this paper we concentrate on MCT gates, the approach can be modified to allow other reversible gates such as Fredkin [5], [6], Peres and inverse-Peres gates [7].

Many quantum gates have been defined and studied in the literature [2]. In this paper, we concentrate on the following gates which we assume have unit cost: NOT and CNOT (as defined above); the 2-line **controlled-**V gate which changes the target line using the transformation defined by the matrix $\mathbf{V} = \frac{1+i}{2}\begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}$ if the single control line has the value 1; and the 2-line **controlled-**V^+ gate which changes the target line using the transformation $\mathbf{V}^+ = \mathbf{V}^{-1} = \frac{1-i}{2}\begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix}$ if the single control line has the value 1. Gates V and V^+ are referred to as ***controlled-square-root-of-NOT*** gates since $\mathbf{V}^2 = (\mathbf{V}^+)^2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

A circuit line not used as the target or as a control of a gate is an *ancillary line* for that gate. Such lines are used in realizing an MCT gate using quantum gates, see [8], [9]

III. Transformation-based Synthesis Algorithms

In this paper for ease of description, we present transformation-based synthesis in terms of the truth table representation of a reversible function. Note that transformation-based methods can be implemented using more efficient representations such as decision diagrams [10].

A. Basic Algorithm

Given a truth table representing a reversible function f, the basic transformation-based synthesis algorithm [3] proceeds through the truth table rows in order $0 \leq i < 2^n - 1$. At each row i, if $f(i) \neq i$ MCT gates are selected to map $f(i)$ to i. These gates are chosen such that they do not affect any row j for $j < i$, i.e. those that have already been considered. The gates are added to the circuit being constructed from the output towards the input and the reversible specification is updated by applying the gates to the output side of the specification. When

all rows $0 \leq i < 2^n - 1$ have been considered the resulting truth table is the identity function and the gates chosen represent an implementation of the original reversible function. Note that row $2^n - 1$ does not have to be considered as $f(2^n - 1) = 2^n - 1$ after all previous rows are made to match.

B. Bidirectional Algorithm

The bidirectional transformation-based synthesis which is an extension of the basic algorithm was also introduced in [3]. For each row i, the gates G_0 required to transform the output pattern $f(i)$ to i are determined as in the basic algorithm. In addition, there must be a row j later in the table where $f(j) = i$. MCT gates G_1 that transform j to i are determined. The less expensive of G_0 and G_1 is determined and those gates are added to the circuit and used to update f. Note that if G_1 is chosen the gates apply from the input toward the output of the circuit and are used to update the input side of the specification. The cost of a set of gates can be simply the MCT gate count or can be based on the quantum cost of implementing the MCT gates. We use the latter in this work.

C. A New Algorithm

Our new method is shown in Algorithm 1. For each row i, the new generalized algorithm tries every row from i to $2^n - 1$ mapping the input and the output patterns to i thereby potentially adding gates to both the input and the output side of the circuit. The algorithm chooses the first such mapping that has lowest cost. To see the new algorithm subsumes the previous two, note that the basic algorithm is the case of only considering $j = i$, while the bidirectional algorithm is the case of only considering two cases: $j = i$ and j such that $f(j) = i$.

Algorithm 1 employs three functions: Cost(G) which returns the quantum cost of a list of MCT gates G; RMdist(f) which determines the Hamming distance between the Reed-Muller spectra [11] of a reversible function f and the identity function; and Map(y, x) which returns a set of MCT gates that map the pattern y to x where $y > x$ such that the gates to do not affect any $z < x$.

In this work, we use a simple version of Cost(G) that sums the quantum cost of each MCT gate in G using the MCT quantum gates costs from [12]. It does not look for simplifications between MCT gates.

RMdist(f) computes the Hamming distance between the Reed-Muller spectrum of f and the Reed-Muller spectrum of the identity function with the same number of inputs and outputs as f. The identity function is the reversible function that maps each input pattern to itself. The Hamming distance between two spectra is the sum of the Hamming distances between the corresponding output patterns. The Reed-Muller spectrum is used since it measures global rather than local function information. The Hamming distance is used to measure how close two spectra are to each other.

Map(y, x) is described in Algorithm 2. It identifies a sequence of MCT gates to map the bit pattern y to x where $y > x$ which is a consequence of going through the rows of the truth table in ascending order. As noted, the gates are selected so that they have no effect on any bit pattern $z < x$. Map uses a function MakeGate($target, controls$) that creates the representation of an MCT gate with specified $target$ and $controls$.

Algorithm 1 Generalized Transformation-Based Synthesis

```
procedure SYNTHESIZE(f, n)
    C_in = C_out = empty
    for 0 ≤ i < 2^n − 1 do
        bestCost = ∞
        if f(i) ≠ i then
            for i ≤ j < 2^n do
                G_in = Map(j, i) G_out = Map(f(j), i)
                if Cost(G_in) + Cost(G_out) < bestCost then
                    B_in = G_in B_out = G_out
                    bestCost = Cost(G_in) + Cost(G_out)
                    f_B = f
                    apply gates in G_out to the output side of f_B
                    apply gates in G_in to the input side of f_B
                end if
                if Cost(G_in) + Cost(G_out) = bestCost then
                    f_t = f
                    apply gates in G_out to the output side of f_t
                    apply gates in G_in to the input side of f_t
                    if RMdist(f_t) < RMdist(f_B) then
                        B_in = G_in B_out = G_out
                        f_B = f_t
                    end if
                end if
            end for
            apply gates in B_out to the output side of f
            apply gates in B_in to the input side of f
            C_in = concatenate(C_in, B_in)
            C_out = concatenate(reverse(B_out), C_out)
        end if
    end for
    return concatenate(C_in, C_out)
end procedure
```

Algorithm 2 MCT Gate Selection to map y to x

```
procedure MAP(y, x)
    glist = empty
    if x = y then
        return glist
    end if
    c = y
    remove 1 bits from the right of c while c ≥ x
    p = (x ⊕ y)&(∼ c)
    for each bit position p_j = 1 do
        g = MakeGate(j, c)
        glist = concatenate(glist, g)
    end for
    q = c&(∼ x)
    c = x
    for each bit position q_j = 1 do
        g=MakeGate(j, c)
        glist = concatenate(glist, g)
    end for
    return glist
end procedure
```

Algorithm 2 begins by setting the control specification c to have as few 1's as possible from y such that $c \geq x$. The latter condition is required to be sure the gates will not affect earlier rows in the truth table. The first **for** loop generates MCT gates with controls c with one gate for each variable outside c that has to be flipped to make y match x. The second **for** loop then uses x as the control and generates one gate for each variable in c that has to be made 0 to match x. In both loops each gate generated has a variable whose value is to change as its target.

IV. Experimental Results

We have implemented the basic, bidirectional and new synthesis algorithms in Python which is convenient as it has efficient built-in list handling and reversible and quantum circuits are lists of gates. Table I shows the results of applying each

TABLE I. 3 VARIABLE FUNCTIONS

	Basic Algorithm	Bidirectional Algorithm	New Algorithm
Max MCT Gates	17	15	14
Avg. MCT Gates	8.67	7.06	6.85
Max NCV Gates	29	29	29
Avg. NCV Gates	17.87	16.16	15.72

of the algorithms to all 40,320 3-variable reversible functions. The bidirectional method reduces the average numbers of MCT and NCV gates required and the new algorithm reduces the averages further. The differences are not particularly large because there is little scope for optimization with 3 variables.

Table II shows the results of applying transformation-based synthesis for a selection of benchmarks of varying size taken from RevLib [13]. The first section of the table gives the name and the number of circuit lines for each benchmark The next three sections show the number of MCT gates in the circuits produced by the basic, the bidirectional and the new transformation-based synthesis methods respectively (labeled Synth.), and the number of gates after application of MCT template reduction [14] (labeled Opt.). Note that in this paper we restrict our attention to circuits with positive controls. Further reduction can be achieved if negative controls are used, assuming they are permitted in the target technology.

The lowest synthesized and optimized gate counts for each benchmark are indicated in bold. The final section of the table shows the percentage improvements (a few are negative) for the new algorithm compared to the better of the results from the basic and the bidirectional algorithms. Percentages are shown for the synthesized and the optimized gate counts.

The techniques presented including the new generalized transformation-based synthesis algorithm employ heuristic and greedy techniques. Hence no approach is best in all cases and is the reason for the negative improvements in Tables II. However these results do illustrate the potential benefit of the new algorithm and show that our new approach is generally as good or better than the unidirectional and bidirectional transformation approaches. Several of the benchmarks show that the advantage can be very significant so that the new algorithm can be considered an important addition to previous transformation based approaches.

Table III shows results for mapping the MCT circuits summarized in Table II to NCV circuits. The mapping method developed by Sasanian [15] was used first with the NCV gate counts shown as Map. A * beside the gate count means an extra helper line had to be added to the circuit during the MCT to NCV mapping. This occurs when there is at least one MCT gate that uses every line in the circuit.

The NCV circuits were then optimized using NCV template matching [14] yielding smaller circuits with the gate counts shown as Opt. Minimums are shown in bold and the percentage improvements are shown as before. Once again, because the methods are heuristic and because the optimization depends on the actual circuit structure, some unexpected results arise. For example, for benchmark hwb5_21 the NCV circuit is largest for the basic algorithm, but after optimization it becomes the smallest of the three options. However, in general, except for the hwb (hidden weight bit) functions and ckt3_cycle_68 mapping and optimizing the circuits from the new algorithm gives the best results and in some cases they are significantly better. We are studying the structure of the hwb functions and ckt3 in an effort to improve the new algorithm.

V. CONCLUSION

This paper has presented a new approach to transformation-based synthesis of reversible and quantum circuits. The experimental results show this approach can in some cases give significantly better circuits than do the basic and the bidirectional transformation-based algorithms.

It is clear from Algorithms 1 and 2 that the overall procedure has complexity $O(n2^{2n})$ whereas the basic transformation-based synthesis procedure has complexity $O(n2^n)$. The space requirement is the more restrictive parameter for truth-table based approaches and is the same, $O(2^n)$, for all three synthesis approaches considered. As noted earlier, this can be improved by using decision diagrams.

Ongoing work will consider variants of the Map function in order to further reduce the quantum cost of the synthesized circuits. We plan to consider implementation of the method using symbolic function representations rather then truth tables and to examine the extension of the method to partially-specified reversible functions. We will also consider how the concepts in [5] apply to our method.

Acknowledgements: This work was partially supported by NSERC Discovery Grants. We thank Dr. Z. Sasanian for the use of her NCV circuit optimization program and the reviewers for their helpful comments.

REFERENCES

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

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

[3] D. M. Miller, D. Maslov, and G. W. Dueck, "A transformation based algorithm for reversible logic synthesis," in *Proc. Design Automation Conf.*, 2003, pp. 318–323.

[4] M. Soeken, R. Wille, O. Keszocze, D. M. Miller, and R. Drechsler, "Embedding of large Boolean functions for reversible logic," *ACM J. on Emerging Technologies in computing Systems*, vol. 12, 2015.

[5] C. Chandak, A. Chattopadhyay, and S. M. S. Maitra, "Analysis and improvement of transformation-based reversible logic synthesis," in *Proc. Int'l Symp. on Multiple-valued Logic*, 2013, pp. 47–52.

[6] M. Soeken and A. Chattopadhyay, "Fredkin-enabled transformation-based reversible logic synthesis," in *Proc. Int'l Symp. on Multiple-valued Logic*, 2015, pp. 60–65.

[7] A. Peres, "Reversible logic and quantum computers," *Physical Review A*, vol. 32, no. 6, pp. 3266–3276, 1985.

[8] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, M. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, "Elementary gates for quantum computation," *Physical Review A*, vol. 52, no. 5, pp. 3457–3467, 1995.

[9] D. M. Miller, "Lower cost quantum gate realizations of multiple-control Toffoli gates," in *IEEE Pacific Rim Conference*, 2009, pp. 308–313.

[10] M. Soeken, L. Tague, G. W. Dueck, and R. Drechsler, "Ancilla-free synthesis of large reversible functions using binary decision diagrams," *J. Symb. Comput.*, vol. 73, pp. 1–26, 2016.

[11] D. Muller, "Application of Boolean algebra to switching circuit design and error detection," *IRE Trans.*, vol. 1, pp. 6–12, 1954.

[12] Z. Sasanian and D. M. Miller, "Mapping a multiple-control Toffoli gate cascade to an elementary quantum gate circuit," *J. of Multiple-valued Logic and Soft Computing*, vol. 18, no. 1, pp. 83–98, 2012.

[13] 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*, 2008, pp. 220–225, RevLib is available at www.revlib.org.

[14] M. M. Rahman, "Synthesis of reversible logic," Ph.D. dissertation, University of New Brunswick, 2014.

[15] Z. Sasanian, "Technology mapping and optimization for reversible and quantum circuits," Ph.D. dissertation, University of Victoria, 2012.

TABLE II. MCT EXPERIMENTAL RESULTS

		Basic Algorithm		Bidirectional Algorithm		New Algorithm			
Benchmark	**Lines**	**Synth.**	**Opt.**	**Synth.**	**Opt.**	**Synth.**	**Opt.**	**Synth. %**	**Opt. %**
3_17__6	3	14	11	**7**	**7**	**7**	**7**	0.0%	0.0%
fredkin__3	3	**3**	**3**	**3**	**3**	**3**	**3**	0.0%	0.0%
ham3__47	3	**6**	**5**	**6**	**5**	**6**	**5**	0.0%	0.0%
miller__5	3	6	6	6	6	**5**	**5**	16.7%	16.7%
peres__4	3	**2**	**2**	**2**	**2**	**2**	**2**	0.0%	0.0%
4_49__7	4	30	24	**20**	**19**	**20**	20	0.0%	-5.3%
hwb4__20	4	22	22	16	16	**13**	**12**	18.8%	25.0%
hwb5__21	5	60	51	**49**	**47**	**49**	**47**	0.0%	0.0%
mod5d1__24	5	**8**	**8**	**8**	**8**	**8**	**8**	0.0%	0.0%
mod5d2__25	5	17	14	**9**	**9**	**9**	**9**	0.0%	0.0%
mod5mils__26	5	9	8	**5**	**5**	**5**	**5**	0.0%	0.0%
graycode6__19	6	**5**	**5**	**5**	**5**	**5**	**5**	0.0%	0.0%
hwb6__22	6	147	134	**102**	**97**	108	101	-5.9%	-4.1%
mod5adder__56	6	39	39	**28**	**28**	**28**	**28**	0.0%	0.0%
ham7__48	7	286	254	98	88	**59**	**53**	39.8%	39.8%
hwb7__23	7	376	345	346	307	**316**	**286**	8.7%	6.8%
ckt2_cycle__67	8	890	787	709	**640**	**685**	642	3.4%	-0.3%
hwb8__54	8	925	822	710	647	**674**	**627**	5.1%	3.1%
ckt1_cycle__66	9	1966	1746	1680	1509	**1619**	**1480**	3.6%	1.9%
ckt5_cycle__69	9	1318	1173	909	843	**845**	**774**	7.0%	8.2%
hwb9__55	9	2078	1851	1802	1623	**1763**	**1604**	2.2%	1.2%
ckt3_cycle__68	10	4038	3545	3519	3162	**3460**	**3156**	1.7%	0.2%
cycle10_2__51	12	**19**	**19**	**19**	**19**	**19**	**19**	0.0%	0.0%
plus63mod4096__72	12	394	392	394	392	**18**	**18**	95.4%	95.4%
plus127mod8192__71	13	781	779	781	779	**19**	**19**	97.6%	97.6%
plus63mod8192__73	13	457	455	457	455	**20**	**20**	95.6%	95.6%
04101+B7:C33+B7:C3384__77	14	48	48	**37**	**37**	**37**	**37**	0.0%	0.0%

TABLE III. NCV EXPERIMENTAL RESULTS

	Basic Algorithm		Bidirectional Algorithm		New Algorithm			
Benchmarl\k	**Map**	**Opt.**	**Map**	**Opt.**	**Map**	**Opt.**	**Map %**	**Opt%**
3_17__6	22	12	**13**	**11**	**13**	**11**	0.0%	0.0%
fredkin__3	**14**	**7**	**14**	**7**	**14**	**7**	0.0%	0.0%
ham3__47	**9**	**7**	**9**	**7**	**9**	**7**	0.0%	0.0%
miller__5	**9**	**8**	**9**	**8**	**9**	**8**	0.0%	0.0%
peres__4	**4**	**4**	**4**	**4**	**4**	**4**	0.0%	0.0%
4_49__7	69*	48	80*	61	**68***	**41**	1.4%	14.6%
hwb4__20	67*	52	50*	37	**22**	**20**	56.0%	45.9%
hwb5__21	308*	**207**	**284***	228	**284***	228	0.0%	-10.1%
mod5d1__24	11	**9**	**11**	**9**	**11**	**9**	0.0%	0.0%
mod5d2__25	18	15	**14**	**11**	**14**	**11**	0.0%	0.0%
mod5mils__26	10	**9**	**9**	**9**	**9**	**9**	0.0%	0.0%
graycode6__19	**5**	**5**	**5**	**5**	**5**	**5**	0.0%	0.0%
hwb6__22	957*	722	**580***	**372**	672*	472	-15.9%	-26.9%
mod5adder__56	402*	349	**237**	**235**	**237**	**235**	0.0%	0.0%
ham7__48	2241*	1703	568	554	**286**	**270**	49.6%	51.3%
hwb7__23	3132*	2517	2773*	**2087**	**2759***	2088	0.5%	0.0%
ckt2_cycle__67	8527*	7009	7222*	5696	**6895***	**5413**	4.5%	5.0%
hwb8__54	8846*	7409	**6469***	**5049**	7222	5696	-11.6%	-12.8%
ckt1_cycle__66	21261*	17948	18703*	15442	**18402**	**15132**	1.6%	2.0%
ckt5_cycle__69	16615*	13939	13170*	10716	**11685***	**9419**	11.3%	12.1%
hwb9__55	21283*	17869	19557*	**16080**	**19364***	16248	1.0%	-1.0%
ckt3_cycle__68	49063*	41935	**44114***	**37280**	46411	40065	-5.2%	-7.5%
cycle10_2__51	**724**	**724**	**724**	**724**	**724**	**724**	0.0%	0.0%
plus63mod4096__72	1556*	1283	1556*	1283	**601***	**464**	61.4%	63.8%
plus127mod8192__71	2068*	1726	2068*	1726	**726***	**607**	64.9%	64.8%
plus63mod8192__73	2012*	1716	2012*	1716	**751***	**634**	62.7%	63.1%
04101+B7:C33+B7:C3384__77	851	845	**560**	**554**	**560**	**554**	0.0%	0.0%