

# Low-Quantum Cost Circuit Constructions for Adder and Symmetric Boolean Functions

Anupam Chattopadhyay and Anubhab Baksi

School of Computer Engineering, Nanyang Technological University, Singapore

**Abstract**—Quantum computing necessitates the design of circuits via reversible logic gates. Efficient reversible circuit can be constructed by achieving low ancilla count, reducing logical depth and lowering Quantum costs. Generalized Peres gates have recently been realized with very low Quantum Cost (QC) by utilizing Quantum rotation gates. This is utilized in recent literature for efficient reversible circuit constructions for symmetric Boolean functions. In this paper, we extend this line of construction further by demonstrating efficient realization of adder circuits. In particular, we revisit the adder construction of Vedral, Barenco and Eckert to show that improvement of gate count and QC is achievable by exploiting a construction based only on Peres gates. We also report improved constructions of symmetric Boolean functions by following an approach recently proposed in the context of Boolean function complexity analysis.

## 1. INTRODUCTION

Due to the application of arithmetic circuits in various Quantum algorithms, their efficient implementation using reversible logic gates is a well-studied problem. For example, Shor's factorization [1], has remained a key attraction for experimental Quantum computing groups for obvious reasons [2] and involves modular arithmetic operations [3]. Naturally,  $n$ -variable binary adder circuits have received significant research attention. Several adder realizations for Quantum circuits have been proposed in the literature [4], [5], [6], [7], [8], [9]. In this paper, we investigate potential improvements in the adder circuit realization. Considering a more general form, symmetric Boolean functions [10] represent an important subclass of Boolean functions, with specific applications in circuits dominant in arithmetic operations, and therefore, deserves closer attention as well. In the second part of this paper, we revisit a recently proposed construction of symmetric Boolean functions [11] and show that further improvements on this construction is possible.

### A. Preliminaries

A Boolean function  $f$  is of the form  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  (or equivalently  $f : \mathbb{V}_2^n \rightarrow \mathbb{V}_2$ ). The output of the Boolean function  $f$  can be represented as a string  $s$  of ones and zeros. It can also be represented as a multivariate polynomial over  $GF(2)$ . This polynomial can be expressed as an exclusive disjunction (EXOR) of a constant  $a_0$  and one or more conjunctions of the function argument. This is called the Exclusive Sum-Of-Product (ESOP) representation. An alternative representation of the ESOP form is known as the Algebraic Normal Form (ANF). The general ANF for a

function  $f(x_1, \dots, x_n)$  over  $n$ -variables can be written as,

$$f(x_1, \dots, x_n) = a_0 \oplus a_1 x_1 \oplus \dots \oplus a_i x_i \oplus \dots \oplus a_n x_n \\ \oplus \dots \oplus a_{1,2,\dots,n} x_1 x_2 \dots x_n$$

The *Hamming weight* (or simply *weight*) of a binary string  $S$  is the number of 1's present in it, which we denote as  $wt(S)$ .

Symmetric Boolean functions [10] form an important subclass of Boolean functions. A Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  is called *symmetric* if its output is invariant under any permutation of its input bits. Equivalently, we can say that the value of  $f(x)$  is constant for all  $x$ 's having the same weight.

*Reversible and Irreversible Boolean functions:* An  $n$ -variable vectorial Boolean function is termed *reversible* if all its output patterns map uniquely to an input pattern and vice-versa. It can be expressed as an  $n$ -input,  $n$ -output bijection or alternatively, as a Boolean permutation function over the truth value set  $\{0, 1, \dots, 2^{n-1}\}$ . An *irreversible* Boolean function  $f_{irr} : \{0, 1\}^n \rightarrow \{0, 1\}^m$  with  $n \neq m$  can also be made reversible with the help of additional output lines, which adds distinguishing patterns to the irreversible output. Correspondingly, additional inputs are added. If an input line is constant-initialized and the constant is recovered after the circuit execution then, it is termed as *ancilla*. Otherwise, it is termed as *garbage*.

### B. Reversible Logic Gates

Reversible logic circuit is implemented with the elementary reversible logic gates. The gates are characterized by their implementation cost in quantum technologies, which is denoted as the Quantum Cost (QC). A reversible gate library is a complete set of reversible gates which can be used to implement any reversible circuit. For example the set of NOT, CNOT, controlled- $V$  and Controlled- $V^+$ , known as NCV, is a reversible gate library widely used in the literature. Recently, there has been a significant research activity towards the realization of quantum circuits using Clifford+T gates, considering the importance of fault tolerance in quantum computing. Efficient synthesis for NCV circuits [12], [13], [14], Clifford+T circuits [15] and mapping of NCV circuits to Clifford+T gates [16] have been proposed in the literature. Few gates from these libraries are outlined below. For detailed discussion on primitive quantum gates and their universality, readers may refer to [17], [18].

- **NOT gate:** This is represented using the matrix  $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ .
- **CNOT gate:**  $\text{CNOT}(a, b) = (a, a \oplus b)$ . This gate can be generalized with  $Tof_n$  gate, where first  $n - 1$  variables are used as control lines. For 2 control lines and 1 target line,

it is denoted as Toffoli gate. Generalized Toffoli gates are also referred to as Multiple Control Toffoli (MCT) gate. When both positive control and negative control lines are permitted, the gate is denoted as Mixed-Polarity Multiple Control Toffoli (MPMCT) gate.

- **Peres gate:**  $Per(a, b, c) = (a, a \oplus b, ab \oplus c)$ . This gate can be generalized with  $Per_n$  gate ( $n > 2$ ) [19], where first  $n - 1$  variables are used as control lines.

### C. Cost Models

For evaluating the performance of the synthesis tools and benchmark circuit implementations, different cost models have been proposed in the literature. The most basic cost model is the number of reversible logic gates needed for the implementation. However, the actual implementation cost of these logic gates could be, with varying number of control lines, can be very different. The QC value is computed for each of these gates, which is nothing but the number of 2-qubit gates [20] needed to implement these circuits. In recent fault-tolerant Quantum circuit implementations, the cost is estimated in terms of T gates, corresponding to the realization of the circuits using Clifford+T library. Specifically, the logical depth in terms of the depth of T gates serve as a performance metric. Another important performance indicator is the total number of lines or ancilla/garbage count.

In the following complexity analysis, we provide the complexity results in terms of the count of MPMCT gates and the total line count. Further, we assume a conservative per-gate estimate for T-depth, which is  $3 \times |C|$ , where  $|C|$  is the number of control lines [16].

## 2. REVERSIBLE CIRCUIT FOR ADDER

For the first Quantum ripple-carry adder circuit proposed in [4], in fact,  $(2n - 1)$  Peres gates are deployed in the carry computation blocks, though it uses  $n$  ancilla lines. Among the ancilla-free  $n$ -bit binary adder circuits reported so far in the literature, [9] has the least gate cost with  $(7n - 6)$  gates, out of which, there are  $(2n - 1)$   $Tof_3$  (CCNOT) gates.

In the following, we propose a simple ripple-carry adder construction that is fully based on Peres gates. Our design is motivated by two recent results. QC of  $Per_n$  is  $n^2$  and QC of  $Tof_{n+1}$  is  $2n^2 - 2n + 1$  [19]. An exemplary structure of cascaded  $Per_n$  gates is shown graphically in Fig. 1. We report a lemma from [11] and show how that can be used for designing adder circuits with low QC.

*Lemma 2.1:* 2 cascaded  $Per_n$  gates require  $n^2 + n$  elementary 1-QC gates.

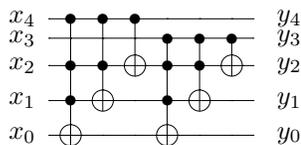


Fig. 1. Cascaded Peres Gates

*Lemma 2.2:* An  $n$ -bit adder circuit with  $n$  ancilla lines can be realized with  $2n$   $Per_2$  gates, where all the  $Per_2$  gates are in pairwise cascade form.

*Proof:* For performing the addition, we repeatedly apply the following equations.

$$sum_i = x_i \oplus y_i \oplus c_{in} \quad (1)$$

$$\begin{aligned} c_{out} &= Majority(x_i, y_i, c_{in}) \\ &= x_i \cdot c_{in} \oplus (y_i \cdot (x_i \oplus c_{in})) \end{aligned} \quad (2)$$

The  $c_{out}$  is treated as  $c_{in}$  for the next level in a simple ripple-carry adder formation. The realization of these Boolean functions can be arranged in form of pairwise cascaded  $Per_2$  gates as shown in the Figure 2. For an  $n$ -bit adder, clearly  $2n$   $Per_2$  gates are required. ■

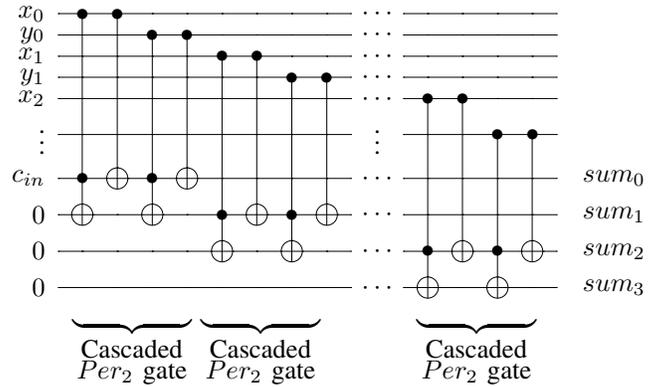


Fig. 2. Reversible Adder based on  $Per_2$

Considering the state-of-the-art adder constructions, the proposed circuit offers lower QC at the expense of higher ancilla lines (Table I). For the Toffoli ( $Tof_3$ ) and Peres ( $Per_2$ ) gates, QC values of 5 and 4 are used respectively [19]. Note that, we accounted for the  $Per_2$  gates in the adder structures of [9] and [4]. Furthermore, the input bits for the adder circuit remain unaltered at the output and therefore, can be used for subsequent circuit blocks. The total gate count is reported in MCT equivalence, where each Peres gate accounts for 2 MCT gates.

Circuit	Ancilla	Gate Count				QC
		CNOT	Toffoli	Peres	Total	
Vedral <i>et al.</i> [4]	$n$	$2n + 1$	$2n - 1$	$2n - 1$	$8n - 2$	$20n - 8$
Takahashi <i>et al.</i> [9]	0	$4n - 4$	$n$	$n - 1$	$7n - 6$	$13n - 8$
Cuccaro <i>et al.</i> [7]	1	$4n + 1$	$2n$	0	$6n + 1$	$14n + 1$
This work	$n$	0	0	$2n$	$4n$	$8n$

TABLE I. PERFORMANCE OF ADDER CIRCUITS

The proposed realization of adder circuit with only Peres gates reduces QC compared to [4], [9], [7] when  $n \geq 2$ . A similar adder structure has been recently proposed in [21, Fig. 14a], where, however, the cascading of generalized Peres gates have not been explored. Thus, the QC is higher in that case.

## 3. REVERSIBLE CIRCUITS FOR SYMMETRIC BOOLEAN FUNCTIONS

Implementation of symmetric Boolean functions follows two steps. First, the computation of Hamming weight and second, the comparison of the Hamming weight with a specific

outcome of 0 or 1. It has been shown in [11] that, by using a ripple carry adder circuit, as described in the previous section 2, circuit with more efficient construction than state-of-the-art implementations can be obtained. We improve this result with another implementation of the Hamming weight computation circuit. First, we present a general result for an  $n$ -input 1-output symmetric Boolean function.

*Lemma 3.1:* The reversible circuit corresponding to an  $n$ -input 1-output symmetric Boolean function requires at most  $n + \lceil \log_2^n \rceil$  garbage outputs.

*Proof:* Maximum Hamming weight of an  $n$ -variable Boolean function can be  $n$ , which can be stored at  $\lceil \log_2^n \rceil$  output lines. Considering these lines to be different from the  $n$  input lines, no more than  $n + \lceil \log_2^n \rceil + 1$  output is produced, out of which 1 output contains the result. ■

In the following we study, two different circuit constructions for symmetric Boolean functions.

### A. Construction I

The first construction, proposed in [11], computes the Hamming weights of the input variables in additional lines. The construction is presented in the Figure 3. The Hamming weight computation is followed by a series of Toffoli gates for deriving the output corresponding to each Hamming weight.

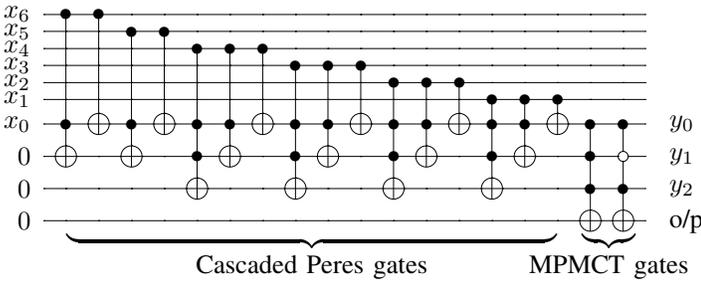


Fig. 3. Symmetric Boolean Function: Construction I

A Quantum cost derivation for this circuit design approach is already presented in [11]. For the ease of future benchmarking, we also present the gate count, T-depth and line computations here.

*Proposition 3.2:* Reversible circuit realization of an  $n$ -variable 1-output symmetric Boolean function requires  $n + \lceil \log_2^n \rceil - 1$  garbage lines if construction I is used.

The proof follows directly from the Fig. 3, which shows that the least significant bit of the Hamming weight is stored on one of the input lines. Note that, this upper bound is 1 less than the one derived at Lemma 3.1 since, one input line is reused for the output.

*Proposition 3.3:* As per construction I, reversible circuit realization of an  $n$ -variable 1-output symmetric Boolean function needs at most  $n + (n - 1) \times (\lceil \log_2 n \rceil + 1)$  MCT gates and has a T-depth of at most  $1.5(n - 1)(\lceil \log_2 n \rceil + 1)(\lceil \log_2 n \rceil + 2) + 3(n)(\lceil \log_2 n \rceil + 1)$ .

*Proof:* For an  $n$ -variable Boolean function, the Hamming weight computation requires  $(n - 1)$  cascaded  $Per_k$  gates, where the maximum value of  $k$  can be  $\lceil \log_2 n \rceil + 1$ . Considering the worst case scenario, total  $(n - 1)$  cascaded

$Per_{\lceil \log_2 n \rceil + 1}$  gates are needed. Since one  $Per_k$  gate is equivalent to  $k$  MCT gates, we obtain a total  $(n - 1) \times (\lceil \log_2 n \rceil + 1)$  MCT gates. The computation of Symmetric function is composed of Hamming weight calculation followed by a set of comparators. Each comparator is due to one Hamming weight value. There can be at most  $n$  different Hamming weights for a Symmetric function realization, as otherwise, it will become a constant function. Each comparator for a specific Hamming weight value require a mixed-polarity  $Tof_k$  gate, where  $k$  is at most  $(\lceil \log_2 n \rceil + 1)$ . Hence, another  $n$  MPMCT gates are required. In total  $n + (n - 1) \times (\lceil \log_2 n \rceil + 1)$  MPMCT gates are used, at most.

For one  $Per_k$  gate,  $k$  Toffoli gates are required, with the number of control lines ranging from  $k$  to 1 in decreasing order. Hence, the T-depth of a  $Per_k$  gate is  $3 \times \frac{k(k+1)}{2} = 1.5 \times k(k + 1)$ . Considering the worst-case value of  $k$  to be  $\lceil \log_2 n \rceil + 1$  and total  $(n - 1)$   $Per_k$  gates, the T-depth for the Hamming weight computation is  $1.5(n - 1)(\lceil \log_2 n \rceil + 1)(\lceil \log_2 n \rceil + 2)$ . For the comparator part, total T-depth is  $3(n)(\lceil \log_2 n \rceil + 1)$ . Summing up, we get the result. ■

### B. Construction II

Recently, Demenkov et al [22] presented a circuit of size  $4.5n$  using which, the binary representation of the sum of  $n$  input bits can be computed. Note that, there, the size refers to 2-input Boolean functions from the set  $\{\vee, \wedge, \oplus\}$ . The overall complexity for a symmetric Boolean function is presented as  $4.5n + O(n)$ , where the  $O(n)$  is contributed by the comparator circuit, as discussed in the previous construction. Hence, we concentrate on the Hamming weight construction part and assume the comparator part to be implemented with exactly the same complexity as in construction I.

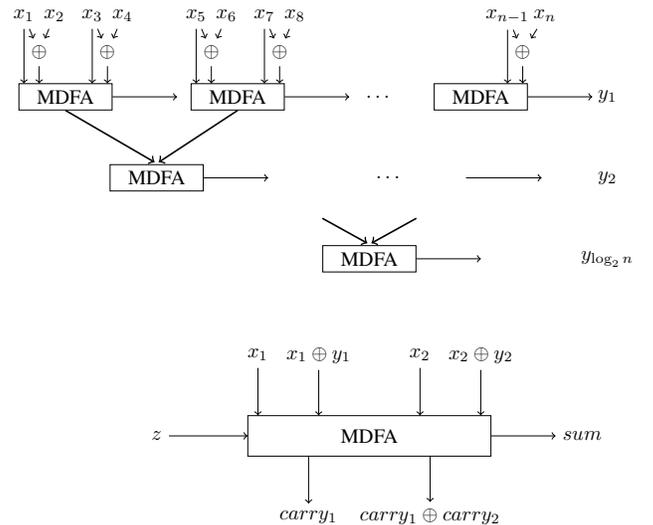


Fig. 4. Symmetric Boolean Function: Construction II

The key circuit idea is presented in the Figure 4. An  $n$ -variable Hamming weight computing circuit uses  $n/2$  Modified Double Full Adders (MDFAs), which is shown in detail at the bottom of the figure. Each MDFA is a 5-input, 3-output circuit, which is presented in form of a 8-gate circuit in [22, Figure 3]. In the Figure 5, we present the corresponding

reversible circuit realization using MPMCT gates. As it can be observed, for each MDFA, 5 garbage lines are generated. From this construction, we can state the following.

**Proposition 3.4:** Reversible circuit realization of an  $n$ -variable 1-output symmetric Boolean function requires  $2.5n + \lceil \log_2^n \rceil - 1$  garbage lines if construction II is used.

*Proof:* Construction II uses total  $n/2$  MDFA blocks, resulting in  $2.5n$  garbage lines. From the  $\lceil \log_2^n \rceil$  lines storing the Hamming weight, except 1 output line, the rest lines are not useful. Hence, the result. ■

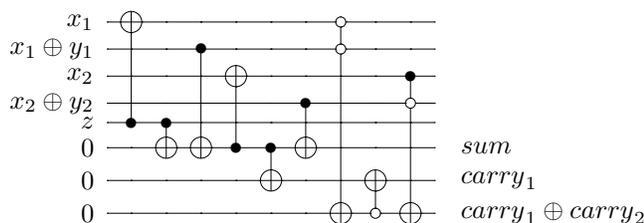


Fig. 5. Reversible Circuit Implementation of MDFA

**Proposition 3.5:** As per construction II, reversible circuit realization of an  $n$ -variable 1-output symmetric Boolean function needs at most  $6n$  MPMCT gates and has a T-depth of at most  $19.5n + 3n(\lceil \log_2 n \rceil + 1)$ .

*Proof:* For an  $n$ -variable Boolean function, the Hamming weight computation requires  $n/2$  MDFA blocks, each of which has 9 Toffoli gates. Hence,  $4.5n$  Toffoli gates are needed. For the initial  $\oplus$  operations,  $n/2$  Toffoli gates are used. Altogether, for the Hamming weight circuit,  $5n$  MPMCT gates are needed, at most. Adding with the previous results of comparator circuit complexity, we obtain the result.

For each MDFA, including the initial  $\oplus$  operations, a T-depth of  $3 \times 13 = 39$  is obtained. Hence, the total T-depth for the Hamming weight computation circuit is  $39 \times n/2 = 19.5n$ . Summing up with the previously established T-depth for the comparator circuit, we get the result. ■

In short, construction I provides an implementation with less garbage lines and construction II, provides an implementation with less gate count and T-depth.

#### 4. CONCLUSION

In this paper, we propose a new reversible circuit construction for binary adder, which improves state-of-the-art designs in terms of gate count and QC, while admitting ancilla lines. We also performed a detailed analysis of symmetric Boolean function implementations in reversible circuits, and proposed an improved circuit construction. In future, we will follow similar implementation techniques for realizing further complex designs of relevance in Quantum algorithms.

#### REFERENCES

[1] 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, Oct. 1997. [Online]. Available: <http://dx.doi.org/10.1137/S0097539795293172>

[2] M. K. L. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood and I. L. Chuang, "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance," *Nature*, vol. 414, no. 6866, 2001. [Online]. Available: <http://dx.doi.org/10.1038/414883a>

[3] I. L. Markov and M. Saeedi, "Faster quantum number factoring via circuit synthesis," *Phys. Rev. A*, vol. 87, p. 012310, Jan 2013. [Online]. Available: <http://link.aps.org/doi/10.1103/PhysRevA.87.012310>

[4] V. Vedral, A. Barenco, and A. Ekert, "Quantum networks for elementary arithmetic operations," *Physical Review A*, vol. 54, no. 1, pp. 147–153, 1996.

[5] T. G. Draper, "Addition on a quantum computer," *CoRR*, vol. abs/quant-ph/0008033, 2000.

[6] Y. Takahashi and N. Kunihiro, "A Linear-size Quantum Circuit for Addition with No Ancillary Qubits," *Quantum Info. Comput.*, vol. 5, no. 6, pp. 440–448, Sep. 2005. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2011670.2011672>

[7] S. A. Cuccaro, T. G. Draper, S. A. Kutin, and D. P. Moulton, "A new quantum ripple-carry addition circuit," *CoRR*, vol. abs/quant-ph/0410184, 2005.

[8] C. H. Bennett, "Logical Reversibility of Computation," *IBM Journal of Research and Development*, vol. 6, pp. 525–532, 1973.

[9] Y. Takahashi, S. Tani, and N. Kunihiro, "Quantum Addition Circuits and Unbounded Fan-out," *Quantum Info. Comput.*, vol. 10, no. 9, pp. 872–890, Sep. 2010. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2011464.2011476>

[10] A. Canteaut and M. Videau, "Symmetric Boolean functions," *Information Theory, IEEE Transactions on*, vol. 51, no. 8, pp. 2791–2811, Aug 2005.

[11] A. Chattopadhyay, S. Majumder, C. Chandak, and N. Chowdhury, "Constructive Reversible Logic Synthesis for Boolean Functions with Special Properties," in *Reversible Computation*, ser. Lecture Notes in Computer Science, S. Yamashita and S.-i. Minato, Eds., 2014, vol. 8507, pp. 95–110.

[12] M. Soeken and A. Chattopadhyay, "Fredkin-Enabled Transformation-Based Reversible Logic Synthesis," in *Multiple-Valued Logic (ISMVL), 2015 IEEE International Symposium on*, May 2015, pp. 60–65.

[13] R. Wille and R. Drechsler, "BDD-based Synthesis of Reversible Logic for Large Functions," in *Proceedings of the 46th Annual Design Automation Conference*, ser. DAC '09, 2009, pp. 270–275.

[14] D. Grosse, R. Wille, G. Dueck, and R. Drechsler, "Exact Multiple-Control Toffoli Network Synthesis With SAT Techniques," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 28, no. 5, pp. 703–715, May 2009.

[15] M. Amy, D. Maslov, M. Mosca, and M. Roetteler, "A Meet-in-the-Middle Algorithm for Fast Synthesis of Depth-Optimal Quantum Circuits," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 32, no. 6, pp. 818–830, June 2013.

[16] D. M. Miller, M. Soeken, and R. Drechsler, "Mapping NCV Circuits to Optimized Clifford+T Circuits," in *Reversible Computation*, ser. Lecture Notes in Computer Science, S. Yamashita and S.-i. Minato, Eds., 2014, vol. 8507, pp. 163–175.

[17] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, "Elementary gates for quantum computation," *Phys. Rev. A*, vol. 52, pp. 3457–3467, Nov 1995. [Online]. Available: <http://link.aps.org/doi/10.1103/PhysRevA.52.3457>

[18] A. Barenco, "A Universal Two-Bit Gate for Quantum Computation," *Proceedings: Mathematical and Physical Sciences*, vol. 449, no. 1937, pp. 679–683, 1995.

[19] M. Szyrowski and P. Kerntopf, "Low quantum cost realization of generalized peres and toffoli gates with multiple-control signals," in *Nanotechnology (IEEE-NANO), 2013 13th IEEE Conference on*, Aug 2013, pp. 802–807.

[20] "Reversible Logic Synthesis Benchmarks Page, howpublished = <http://webhome.cs.uvic.ca/~dmaslov/>, note = Accessed: 2015-10-19."

[21] A. Banerjee and D. K. Das, "Efficient Squaring in Reversible Logic using Reduced Garbage and Ancillary Inputs," in *Work-in-Progress, 6th Conference on Reversible Computation*, 2014.

[22] E. Demenkov, A. Kojevnikov, A. Kulikov, and G. Yaroslavtsev, "New upper bounds on the boolean circuit complexity of symmetric functions," *Information Processing Letters*, vol. 110, no. 7, pp. 264 – 267, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0020019010000256>