Publikacja / Publication	Synthesis of Reversible Circuits Based on Products of Exclusive Or Sums, Schaeffer Ben, Tran Linh, Gronquist Addison, Perkowski Marek, Kerntopf Paweł
Adres publikacji w Repozytorium URL / Publication address in Repository	http://repo.pw.edu.pl/info/article/WUT287612/
Cytuj tę wersję / Cite this version	Schaeffer Ben, Tran Linh, Gronquist Addison, Perkowski Marek, Kerntopf Paweł: Synthesis of Reversible Circuits Based on Products of Exclusive Or Sums, In: Proceedings 2013 IEEE 43rd International Symposium on Multiple-Valued Logic / Guerrero Juan (eds.), 2013, IEEE Computer Society, ISBN 978-0-7695-4976-7, pp. 35-40

Synthesis of Reversible Circuits Based on Products of Exclusive Or Sums

Ben Schaeffer, Linh Tran, Addison Gronquist, Marek Perkowski
Department of Electrical and Computer Engineering
Portland State University
Portland OR, USA
bms@pdx.edu, linht@pdx.edu, addyg@addyg.net, mperkows@cecs.pdx.edu

Pawel Kerntopf
Institute of Computer Science
Warsaw University of Technology
Warsaw, Poland
Department of Theoretical Physics and Computer Science, University of Lodz,
Lodz, Poland
p.kemtopf@ii.pw.edu.pl

Abstract— **This paper introduces a new synthesis approach called Products of EXOR-sums (POE) and demonstrates its utility when employed in an Exclusive-Or-Sum-of-Products (ESOP) based algorithm for reversible and permutative quantum circuits restricted to single output functions. Unlike previous ESOP approaches such as EXORCISM-4 which realize functions as an EXOR-ed sum of product literals, this new algorithm realizes functions as an EXOR-ed sum of POEs (EPOE). A comparison of EXORCISM-4 and EPOE circuits shows that the latter approach often produces circuits with significantly lower quantum costs.**

Keywords- Reversible, ESOP, Product of EXOR Sums, Synthesis, Factorization, Minimization, Quantum Permutative Circuits, Quantum Cost

I. INTRODUCTION

There are several types of algorithms for synthesis of reversible circuits: (1) cycle-based methods [12], (2) group-theory methods [14], (3) transformative methods like MMD [2, 7], (4) BDD-based methods [15], and (5) Exclusive-Or-Sum-of-Products (ESOP) based methods [3, 5, 6, 10, 11]. The ESOP-based methods have two variants: (a) for arbitrary reversible functions, (b) for a subset of arbitrary reversible functions in which only ancilla lines used specifically for output become modified. The latter variant has utility in mapping irreversible functions to reversible functions and has been studied in [11] using a quantum cost metric. Here we present a study of this latter ESOP variant comparing quantum cost metric of the synthesis program EXORCISM-4 [9] and a new approach which uses EXORs of Products-of-EXOR-Sums (EPOE).

The main benefit of Products-of-EXOR-Sums (POE) based synthesis is that it produces lower quantum cost circuits than EXORCISM-4 in a majority of cases. It achieves this by simplifying expressions of multiple high Hamming Distance minterms into a new form that EXORCISM-4 is incapable of producing. In general terms POE synthesis employs an iterative search for satisfactory product terms, each of which is an EXOR sum of terms, and ANDs them together. For instance, Fig. 1 shows how a POE synthesis of a function of four minterms compactly expresses an AND of two relatively simple terms. Comparing POE circuits which synthesize functions of the same number of minterms, quantum costs are generally lower for functions with simpler product terms. This can be observed on a Karnaugh map as the degree to which a set of minterms creates a symmetrical or self-similar pattern.

Most functions cannot be synthesized with a single POE circuit, so a heuristic, divide-and-conquer program named EPOE-1 was developed which can synthesize any single-output function. While EXORCISM-4 employs a two-level AND-EXOR form, EPOE employs a three-level EXOR-AND-EXOR form. This additional level permits EPOE to search for more types of function minimizations. The EPOE algorithm strategy is to search for inexpensive POE functions first, and then more expensive POE functions later if necessary, and ensure all POE functions produce a covering in which over two-thirds of the covered minterms achieve their final output state. This strategy was developed through analysis which indicated that coverings with less than two-thirds of the minterms achieving their final output state usually lead to higher quantum cost circuits.

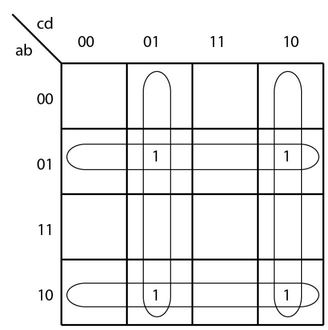

Figure 1. Example of a POE covering four minterms: *$(a \oplus b)(c \oplus d)$*.

The paper is organized as follows. Section II introduces basic concepts. Section III compares quantum costs of different synthesis approaches for functions of two minterms. Section IV introduces an EPOE synthesis algorithm for single output functions. Section V illustrates EPOE synthesis with examples. Section VI presents experimental results and Section VII concludes the paper.

II. PRELIMINARIES

An arbitrary reversible Boolean function is a bijective mapping of N Boolean inputs to N Boolean outputs. In classical reversible logic physical implementation, each input/output pair is typically called a line or wire, whereas in quantum logic it is called a qubit. A reversible circuit schematic representation is shown in Fig. 2. In the figure input signals propagate from left to right through horizontal lines and can be modified as they pass through a cascade of reversible gates.

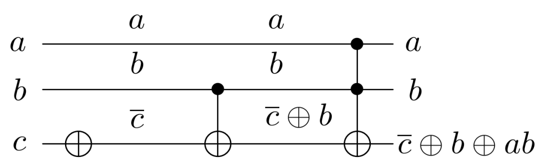

Figure 2. A reversible circuit with a NOT, CNOT, and Toffoli gate.

The fundamental or classical reversible gates correspond to the following Boolean functions.

$$\text{NOT: } (x_1) \rightarrow (x_1 \oplus 1) \quad (1)$$

$$\text{CNOT: } (x_1, x_2) \rightarrow (x_1, x_2 \oplus x_1) \quad (2)$$

$$\text{Tofolli: } (x_1, x_2, x_3) \rightarrow (x_1, x_2, x_3 \oplus x_1 x_2) \quad (3)$$

Multiple-control $N \times N$ Tofolli:

$$(x_1, x_2, \ldots, x_{n-1}, x_n) \rightarrow (x_1, x_2, \ldots, x_{n-1}, x_n \oplus x_1 x_2 \ldots x_{n-1}) \quad (4)$$

For simplicity the ESOP-based reversible circuit synthesis methods considered here will be restricted to a subset of arbitrary reversible Boolean functions in which N-1 lines are treated as inputs which remain unchanged at the circuit's output and one line is treated as an output. This single output function can be expressed in the following form:

$$(x_1, x_2, \ldots, x_{n-1}, x_n) \rightarrow$$
$$(x_1, x_2, \ldots, x_{n-1}, x_n \oplus f(x_1, x_2, x_3, \ldots, x_{n-1})) \quad (5)$$

Consequently for the output line to produce function f an initial output value of $x_n=0$ must be set in advance.

The subset of arbitrary reversible Boolean functions which employs only CNOT and NOT gates is known as affine-linear reversible circuits [14]. These types of circuits can be represented compactly in the form $Y = MX \oplus B$ where X is a vector of N Boolean inputs representing reversible circuit lines, M is an N by N Boolean coefficient matrix which is invertible under GF(2), and B is a vector of N Boolean constants. In this treatment the product of matrix M with vector X uses AND for multiplication and EXOR for addition. Consequently elements of the vector Y are linear functions of X with a constant term from vector B of the following form:

$$y_k = a_{k1}x_1 \oplus a_{k2}x_2 \ldots \oplus a_{kn}x_n \oplus b_k \quad (6)$$

If the above equation is reduced by one input to an N–$1 \times N$–1 affine-linear reversible circuit then it can be used to describe POE functions in the following form:

$$(x_1, x_2, \ldots, x_{n-1}, x_n) \oplus (x_1, x_2, \ldots, x_{n-1}, x_n \oplus y_k y_l y_m \ldots) \quad (7)$$

In the current implementation synthesized POEs are realized using three components: an N–$1 \times N$–1 affine-linear reversible circuit which modifies only input lines, a single fundamental reversible gate targeting the output line which can have a size ranging from a multiple-control $N \times N$ Tofolli to the NOT gate, and a mirror of the N–$1 \times N$–1 affine-linear reversible circuit.

III. COMPARATIVE QUANTUM COST ANALYSIS

In this section the quantum cost of POE synthesis will be compared to two other approaches for functions of two minterms. The first approach uses Disjoint-Sum-of-Products (DSOP) [13] in which cubes cannot overlap, allowing the result to be treated as an EXOR-sum. The second approach uses ESOP with Exorlink operations [9], which for functions of two minterms of Hamming Distance greater than one produces a group of overlapping cubes.

A. Example 1

The first example is the best-case scenario for the POE method, showing how powerful it can be in certain cases. Fig. 3 and Fig. 4 illustrate one group of POE product terms which realize the function $F_1 = ab'c'd' \oplus a'bcd$ and how it compares with the other methods. A quantum cost comparison of POE with the other synthesis methods using the values in Table I follows.

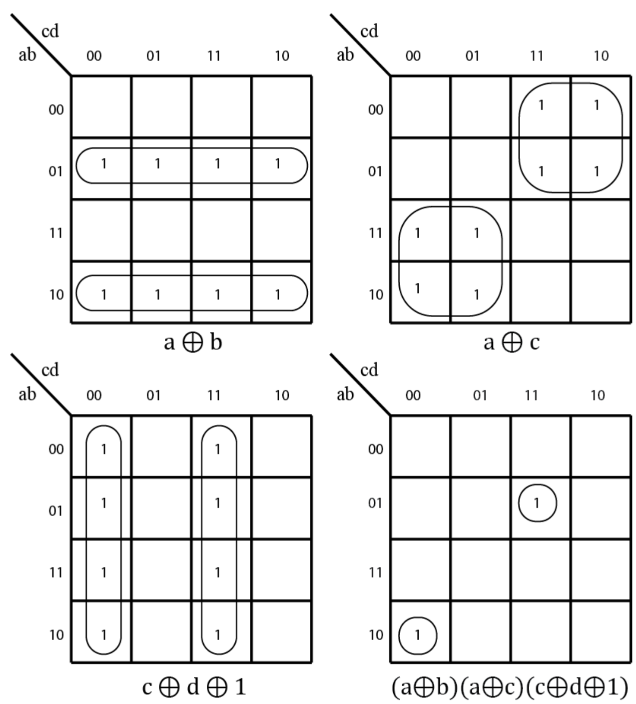

Figure 3. Example POE realization of $F_1 = ab'c'd' \oplus a'bcd$.

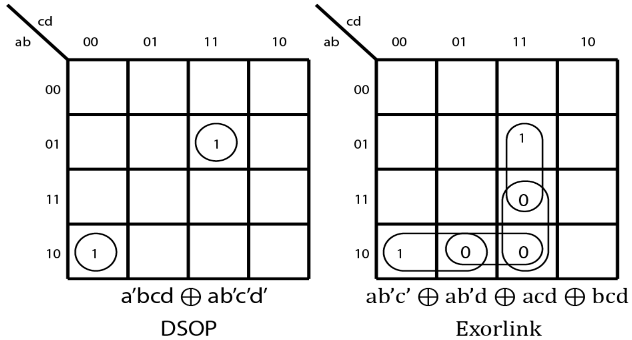

Figure 4. Comparison of DSOP and Exorlink methods for function $F_1 = ab'c'd' \oplus a'bcd$.

TABLE I. QUANTUM COSTS OF FUNDAMENTAL REVERSIBLE GATES.

Gate	Cost
NOT	1
CNOT	1
Toffoli	5
N×N Toffoli	$2^N - 3$

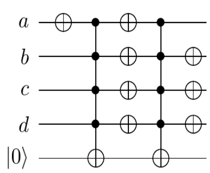

Figure 5. DSOP reversible circuit realization for function $F_1 = ab'c'd' \oplus a'bcd$.

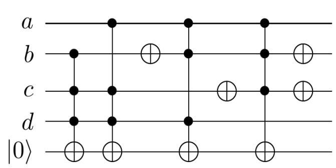

Figure 6. Exorlink reversible circuit realization for function $F_1 = ab'c'd' \oplus a'bcd$.

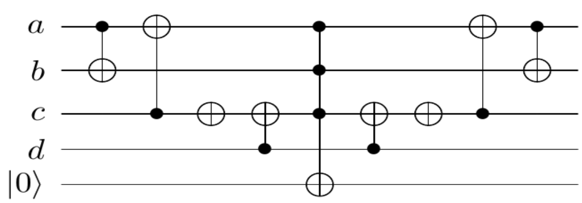

Figure 7. POE reversible circuit realization for function $F_1 = ab'c'd' \oplus a'bcd$.

To realize the DSOP approach, the reversible circuit shown in Fig. 5 was used. The final line '0' is the output line, which is the model that will be used for all synthesis comparisons. All input lines will be restored to their original values to give fair comparison results between the methods. The cost for this realization is two 5×5 Toffoli gates at a cost of 29 each plus 8 NOT gates at a cost of 1 each, hence $2\cdot29 + 8\cdot1 = 66$.

Considering functions of N input variables, the quantum cost equation for two minterms at maximum Hamming Distance can be generalized as: two $N+1\times N+1$ Toffoli gates plus $2N$ NOT gates, costing $2(2^{N+1} - 3) + (2N) = 2^{N+2} + 2N - 6$. This is an exact cost because each input line will need to be inverted twice for any pair of maximum Hamming Distance minterms in order to return all input lines to their original values.

The reversible circuit shown in Fig. 6 illustrates an Exorlink synthesis which realizes the function as $bcd \oplus acd \oplus ab'd \oplus ab'c'$. The quantum cost for this circuit is: four 4×4 Toffoli gates at a cost of 13 each and four NOT gates at a cost of 1 each, hence $4\cdot13 + 4\cdot1 = 56$, a slight improvement upon the DSOP approach. This case is more difficult to quantify for any maximum Hamming Distance because there can be different numbers of NOT gates needed depending upon both the particular minterms in the function and the cubes selected to cover them.

To simplify the quantum cost equation for Exorlink the average case was used in which only half the input lines will have to be inverted. This leads to a realization that requires inverting half of the input lines two times each. For the Toffoli gates, the generalization is that to Exorlink a pair of maximum Hamming Distance minterms in an N-variable function, N Toffoli gates are needed, each of size $N\times N$ because in this situation Exorlink cubes will always cover two minterms. The quantum cost is $N(2^N - 3) + N = N2^N - 2N$. It can be seen that as N grows this method will become more costly than the DSOP approach at $N=5$.

The POE realization is shown in Fig. 7. The cost of this circuit is: one 4×4 Toffoli gate, six CNOT gates, and two NOT gates, for a cost of $1\cdot13 + 6\cdot1 + 2\cdot1 = 21$, which is a cost reduction of at least 37% compared to the previous solutions. The improvement is due mainly to the use of fewer and smaller Toffoli gates.

Considering functions of N input variables, a POE expression for a pair of maximum Hamming Distance minterms employs $N - 1$ product terms. This occurs because for each product term AND-ed the number of minterms covered in the resultant POE is halved, effectively reducing the number of minterms covered from 2^{N-1} to 2. Consequently the reversible circuit requires only one $N\times N$ Toffoli gate at a cost of $2^N - 3$. The affine-linear reversible circuit used in creating product terms uses $N - 1$ CNOT gates, as also is the case for the mirror affine-linear reversible circuit, bringing the CNOT cost to $2N - 2$. Considering the NOT gate quantum cost in the affine-linear reversible circuit and its mirror, taking the average case when half of the inputs require inverting yields an average cost of N NOT gates ($N/2$ literals inverted and then inverted again to revert the inputs). Hence the total average quantum cost of the POE maximum Hamming Distance realization is $(2^N - 3) + (2N - 2) + N = 2^N + 3N - 5$. This is a significant improvement over the other two methods, and in the limit it is 4 times cheaper than the DSOP method, and N times cheaper than the Exorlink method.

B. *Example 2*

Continuing the comparison in Example 1, this second example will generalize the quantum cost equations using other Hamming distances. Again the quantum cost treatment is for functions of N input variables of two minterms at a Hamming Distance of $D \leq N$.

For DSOP the number of cubes remains two, except in the degenerate case of Hamming Distance of one for any given *N*. Therefore the DSOP solution remains two $N+1 \times N+1$ Toffoli gates plus $2N$ NOT gates costing $2^{N+2} + 2N - 6$.

For Exorlink using Hamming Distance *D* results in *D* cubes. Staying with previous assessment for the number of NOT gates required for the average case the cost becomes $D(2^N - 3) + N$.

The POE approach is simple to generalize from Example 1 as well. Reducing the Hamming Distance by one simplifies one product term, reducing the number of CNOT gates by two. This can be seen more concretely by an example.

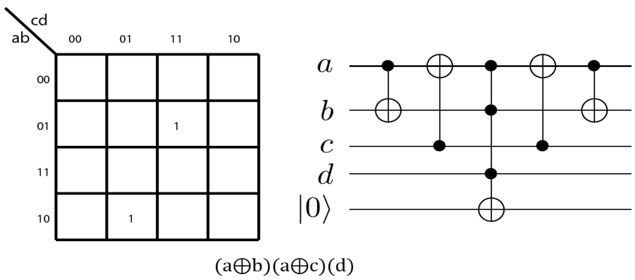

Figure 8. Karnaugh map and POE reversible circuit realization for function $F_2 = ab'c'd \oplus a'bcd$.

TABLE II. QUANTUM COST COMPARISON OF DIFFERENT ESOP METHODS SYNTHESIZING FUNCTIONS OF TWO MINTERMS

Max Hamming Distance	**Number of Variables**					
	3	***4***	***5***	***6***	***7***	***N***
DSOP	32	66	132	262	520	(2^(N+2)) + 2*N - 6
Exorlink	18	56	150	372	882	N*(2^N) - 2*N
POE	12	23	42	77	144	(2^N) + 3*N - 5
Max Hamming Distance-1						
DSOP	32	66	132	262	520	(2^(N+2)) + 2*N - 6
Exorlink	13	43	121	311	757	(N-1)*((2^N)-3) + N
POE	10	21	40	75	142	(2^N) + 3*N - 7
Max Hamming Distance-2						
DSOP	5[a]	66	132	262	520	(2^(N+2)) + 2*N - 6
Exorlink	8	30	92	250	632	(N-2)*((2^N)-3) + N
POE	5	15	33	67	133	(2^N) + 2*N - 9

a. Special case, cubes have merged for 3 variables.

Fig. 8 shows a function similar to Example 1 but with the Hamming Distance reduced by 1. The new function is $F_2 = a'bcd \oplus ab'c'd$, and the product term $(c \oplus d \oplus 1)$ from Example 1 is replaced by *(d)*.

For POE *D–1* CNOT gates are needed for the affine-linear reversible circuit as well as its mirror. The Toffoli gate will remain the same, and the same assumptions are kept about the average number of inversions needed, although this might also need to be adjusted down as there are fewer terms involved in the entire function. This yields a POE realization cost $(2^N - 3) + (2D - 2) + N$. The POE cost marginally improves as the Hamming Distance decreases.

To summarize Examples 1 and 2, Table II lists the costs for three values of *D* for functions of sizes 3 to 7.

IV. THE EPOE-1 ALGORITHM

The EPOE-1 algorithm converts a Boolean function of *N* input variables *(a,b,c,d, ...)* into EPOE form. It uses a strategy of iteratively searching for POE templates which intersect over *2/3* of the covered minterms. The algorithm requires a library of POE templates, calculated in advance, to be grouped together by the number of product terms employed. To keep the library small some template groups can be left out. When functions contain more than $(2/3)2^N$ minterms the output initially becomes $f = 1$ which is synthesized using a NOT gate. Functions of one or two minterms can be synthesized through analysis, although for EPOE-1 functions of two minterms were also included in the POE template library. The variable *level* has the relationship such that the number of minterms covered by a POE template equals 2^{level}. As *level* decreases the number of POE product terms increases. For example, some of the POE templates for functions of three variables are shown in Table III. The Algorithm for EPOE-1 follows.

TABLE III. EXAMPLES OF POE TEMPLATES FOR FUNCTIONS OF THREE VARIABLES

Level	**POE Template**	**Minterms Covered**
3	1	{000,001,010,011,100,101,110,111}
2	a	{100,101,110,111}
2	a ⊕ b	{010,011,100,101}
2	a ⊕ b ⊕ c	{001,010,100,111}
1	a (a ⊕ b)	{100,101}
1	a (a ⊕ b ⊕ c)	{100,111}

Algorithm 1: EPOE-1 Algorithm

```
k := 2/3
remainder := conversion of input
       specification to list of ON-set
       minterms
level := number of input variables
result := NIL
If remainder.length() > k·power(2, level)
  result.append("1") //NOT gate
  remainder := remainder.complement()
level := level - 1

While remainder.length() > 1
  If remainder.length() < k·power(2, level)
    level := level - 1
  Else
```

```
    temp := POETemplateLib.findfirst(level)
    besttemp := NIL
    besttempcovered := 0
    matchfound := FALSE
    do
      covered := Intersection(remainder,
          temp).length()
      If covered = power(2, level)
        matchfound = TRUE
      Else
        If covered > k·power(2, level)
            AND covered > besttempcovered
          besttemp := temp
          besttempcovered := covered
        temp := temp.next()
    while NOT matchfound AND temp is not NIL

    If matchfound
      result.append(temp)
      remainder := remainder.XOR(temp)
    ElseIf besttempcovered > 0
      result.append(besttemp)
      remainder := remainder.XOR(besttemp)
    Else
      level := level - 1

If remainder.length() > 0
  Analyze minterms in remainder directly and
      append POE expressions to result
Return result
```

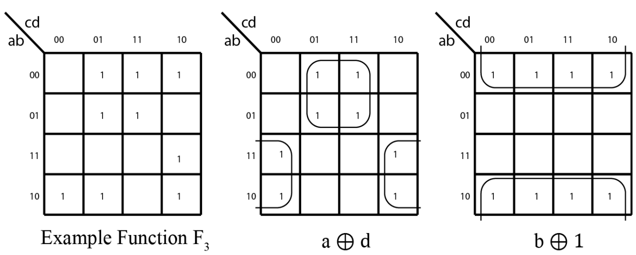

Figure 9. Template selection for EPOE-1.

Fig. 9 illustrates one iteration of the algorithm in which a partial coveing POE template is selected for the function $F_3(a, b, c, d) = \{1, 2, 3, 5, 7, 8, 9, 10, 14\}$. The function given covers seven of the eight minterms of the template $(a \oplus d)$, and the function covers six out of the eight minterms of the template $(b \oplus 1)$, so $(a \oplus d)$ is selected as the result. All the other templates have less than seven minterms covered.

V. Example of EPOE Synthesis

Here is an example of EPOE synthesis of the function $F_4(a, b, c, d) = \{1, 2, 3, 5, 7, 8, 9, 10\}$. A four-variable POE template library must be calculated in advance.

• The *remainder* becomes *{0001, 0011, 0010, 0101, 0111, 1000, 1001, 1010}*.
• At *level* $= 4$ the *remainder* has only 8 minterms and $8 < (2/3)2^4$ so a *"1"* is not appended to the *result*.
• At *level* $= 3$ a POE template library search does not find a template which covers $2^3 = 8$ minterms in *remainder*. The templates $(a \oplus d)$ and $(b \oplus 1)$ were found to cover 6 minterms in *remainder* which is shown in Fig. 10. Since $6 > (2/3)2^3$ the covering is acceptable, therefore the template found first, $(a \oplus d)$, is applied to the *remainder* and appended to the *result*.
• At *level* $= 2$ a POE template library search does not find a template which covers $2^2 = 4$ minterms in *remainder*. The first acceptable partial covering template found was $(a \oplus b \oplus 1)(d \oplus 1)$ which covered 3 minterms in *remainder* and is shown in Fig. 11. Since $3 > (2/3)2^2$ the covering is acceptable and is applied to the *remainder* and appended to the *result*.
• At *level* $= 1$ any function of two minterms can be fully covered by a specific template, here $(b \oplus 1)(c \oplus 1)(a \oplus d \oplus 1)$, leaving no remainder and completing the synthesis. The final value of *result* is as follows:

$$f(a, b, c, d) = (a \oplus d) \oplus (a \oplus b \oplus 1)(d \oplus 1) \oplus (b \oplus 1)(c \oplus 1)(a \oplus d \oplus 1)$$

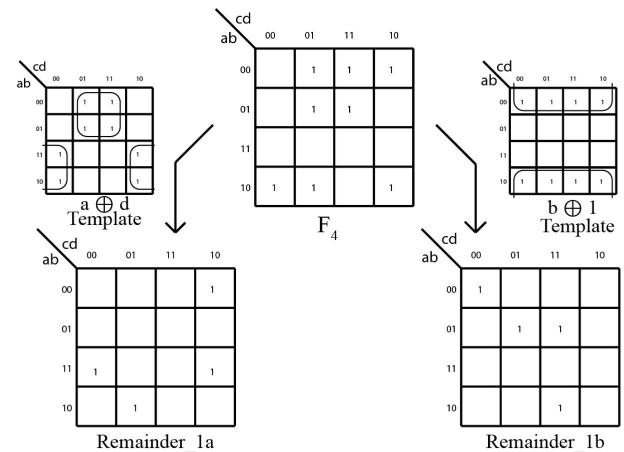

Figure 10. Remainder function comparisons for different templates in EPOE-1.

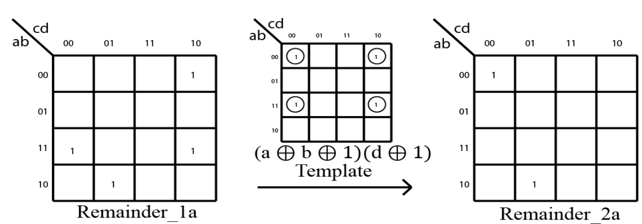

Figure 11. Remainder function evolution in EPOE-1.

VI. Experimental Results

Several tests were performed comparing EPOE-1 synthesis described above with EXORCISM-4 [9]. The results are shown in Table IV (after the References section).

The 4-variable functions tested were:

lt41(a, b, c, d) = {3, 5, 7, 8, 9, 10}
lt42(a, b, c, d) = {1, 3, 4, 8, 11, 12, 15}
lt43(a, b, c, d) = {1, 2, 3, 5, 7, 8, 9, 10}
lt44(a, b, c, d) = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
lt45(a, b, c, d) = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 15}

The 5-variable functions tested were:

lt51(a, b, c, d, e) = {3, 5, 7, 8, 9, 10, 19, 21, 23, 24, 25, 26, 29, 31}

lt52(a, b, c, d, e) = {1, 3, 5, 7, 8, 9, 10, 13, 15, 17, 19, 21, 23, 24, 25, 26, 28, 29, 31}

VII. Conclusions and Future Work

This paper presented a new approach to synthesize quantum and reversible circuits with EXOR of POE gates. There may also be applications in classical logic synthesis, especially for sparse and partially symmetric functions. Our method can potentially replace ESOP-based synthesis, demonstrated here with a cascade of multiple-control Toffoli gates, with EXOR of POE synthesis to reduce costs. Future goals are to test the new algorithm on large functions and also on several examples of practical reversible circuits that are parts of quantum algorithms such as Grover's algorithm and Shor's algorithm.

References

[1] N. Alhagi, Synthesis of Reversible Functions Using Various Gate Libraries and Design Specifications, Ph.D. Dissertation Proposal, May 2008, PSU, ECE Dept.

[2] N. Alhagi, M. Hawash, M. A. Perkowski, "Synthesis of Reversible Circuits with No Ancilla Bits for Large Reversible Functions Specified with Bit Equations" ISMVL 2010, pp. 39-45.

[3] N. Alhagi, M. Lukac, L. Tran, M. Perkowski "Two-stage approach to the minimization of quantum circuit based on ESOP minimization and addition of a single ancilla qubit", Proc. ULSI, 2012.

[4] A. Cheng, Y. Wang, A. Rajendar and M. Perkowski, Comparison of Quantum Costs for Five Realizations of k×k Toffoli Gates, Proc. ULSI, 2012.

[5] K. Fazel, M. Thornton, and J. E. Rice, "ESOP-based Toffoli gate cascade generation," Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, pp. 206-209, August 2007.

[6] Z. Hamza and G. W. Dueck, "Near-optimal ordering of ESOP cubes for Toffoli networks," Proceedings of the 2nd Annual Workshop on Reversible Computation, July 2010.

[7] D. M. Miller, D. Maslov, and G. W. Dueck, "A transformation based algorithm for reversible logic synthesis," Proceedings of the Design Automation Conference, June 2003.

[8] D. M. Miller, R. Wille, and R. Drechsler, "Reducing Reversible Circuit Cost by Adding Lines," Proc. ISMVL, 2010.

[9] A. Mishchenko and M. Perkowski, "Fast Heuristic Minimization of Exclusive Sums-of-Products," Proc. RM 2001 Workshop, August 2001.

[10] A. Mishchenko and M. Perkowski, "Logic Synthesis of Reversible Wave Cascades," Proc. IEEE/ACM International Workshop on Logic Synthesis, pp. 197 – 202, June 4-7, 2002.

[11] N. M. Nayeem, J. E. Rice, "A Shared-cube Approach to ESOP-based Synthesis of Reversible Logic," Facta Universitatis Series : Electronics and Energetics, vol. 24, no. 3, pp. 385-402, December 2011.

[12] M. Saeedi, M. S. Zamani, M. Sedighi, and Z. Sasanian. "Reversible circuit synthesis using a cycle-based approach," ACM Journal on Emerging Technologies in Computing Systems, vol. 6, no. 4, pp. 1-26, 2010.

[13] T. Sasao. Logic Synthesis and Optimization, Kluwer Academic Publishers, Norwell, MA, USA, 1993, pp. 269.

[14] A. De Vos, Reversible Computing: Fundamentals, Quantum Computing, and Applications, Weinheim: Wiley-VCH, 2010.

[15] R. Wille, R. Drechsler, "BDD-Based Synthesis of Reversible Logic," Int. J. of Applied Metaheuristic Computing 1(4), pp. 25-41, 2010.

TABLE IV. FOUR AND FIVE INPUT VARIABLE FUNCTIONS SYNTEHSIZED WITH EXORCISM-4 VERSUS EPOE-1

Function	Results		Quantum Cost	Percent Improvement
lt41	EXORCISM-4	ab'd' ⊕ b'c'd ⊕ a'd	35	37%
	EPOE-1	((a⊕d)(a⊕b)) ⊕ ((a⊕c)(b⊕1)(d))	22	
lt42	EXORCISM-4	ac⊕a'b'⊕d'⊕a'bcd'	43	2.3%
	EPOE-1	(c⊕d⊕1) ⊕((a⊕1)(b⊕c⊕1)) ⊕((a⊕1)(b)(c)(d⊕1)	42	
lt43	EXORCISM-4	b'cd'⊕a'd⊕ab'c'	35	25.71%
	EPOE-1	(b⊕1) ⊕((a⊕1)(b⊕d⊕1)) ⊕((b⊕c)(a⊕d⊕1)(c))	26	
lt44	EXORCISM-4	b'c'd'⊕a'b'c'⊕ab'd'⊕a'	44	40%
	EPOE-1	(a⊕1) ⊕((a)(b⊕1)) ⊕((a⊕c⊕1)(b⊕1)(a⊕d⊕1))	26	
lt45	EXORCISM-4	1⊕acd⊕abd⊕a'b'c'd'	60	30%
	EPOE-1	1⊕((a⊕d⊕1)(b⊕c⊕d⊕1)) ⊕((a⊕1)(b)(c)(d⊕1))	42	
lt51	EXORCISM-4	c'⊕b'e⊕c'de⊕abce⊕a'b'c'e' ⊕a'bc'd'e	142	30%
	EPOE-1	(c⊕e⊕1)⊕((a⊕1)(b⊕c⊕1)(b⊕e⊕1)) ⊕((c⊕1)(b⊕d)e) ⊕a'bc'd'e	101	
lt52	EXORCISM-4	e⊕bc'⊕bc'd'e⊕abcd'e'	99	19%
	EPOE-1	(e)⊕((c⊕1)(b))⊕((d⊕1)(c⊕e)(b)) ⊕ a'bcd'e'	86	