Complexity of Reversible Toffoli Cascades and EXOR PLAs

Dmitri Maslov
Faculty of Computer Science
University of New Brunswick
Fredericton, N.B. E3B 5A3 CANADA
Email: dmaslov@unb.ca

Gerhard W. Dueck
Faculty of Computer Science
University of New Brunswick
Fredericton, N.B. E3B 5A3 CANADA
Email: gdueck@unb.ca

***Abstract*— Reversible logic is an emerging research area. Interest in reversible logic is sparked by its applications in quantum computing, low-power CMOS, nanotechnology, and optical computing. Little work has been done on reversible logic synthesis. The basic implementation structure for a reversible network is a cascade of reversible gates. Several gates have been proposed in the literature. In this paper we restrict the analysis to Toffoli gates were inputs may be inverted. A fundamental question is whether the complexity of the new reversible structures is comparable to traditional PLA like implementations. We compare a cascade of Toffoli gates, for which a synthesis method exists, with an EXOR PLA implementation. We show that in the worst case, the reversible synthesis model produces a result which is only a constant times higher than the one for a conventional EXOR PLA. We also show that there are functions whose complexity is linear in terms of variables for Toffoli cascades, but exponential for EXOR PLAs.**

First we give a detailed description of Tofolli cascades. Next we show that any EXOR PLA can always be converted to a Toffoli cascade with a constant increase in the complexity. It is worth noting, that this conversion may not be minimal. Finally we propose a class of Boolean functions that can be implemented with $\frac{n}{2}+1$ Toffoli gates, and show that the minimal ESOP contains an exponential number of terms.

I. Definitions

A variety of reversible gates have been proposed [2], [3], [10]. Here we use a generalized Toffoli gate defined as follows:

Definition 1: For the set of domain variables $\{x_1, x_2, ..., x_n\}$ **generalized Toffoli gate** is a gate of a form $TOF(C;T)$, where $C = \{x_{i_1}^{\sigma_1}, x_{i_2}^{\sigma_2}, ..., x_{i_k}^{\sigma_k}\}$, where $x_{i_j}^{\sigma_j} = x_{i_j}$ if $\sigma_j = 1$ or $x_{i_j}^{\sigma_j} = \bar{x}_{i_j}$ if $\sigma_j = 0$, $T = \{x_j\}$ and $C \cap T = \emptyset$ which maps a Boolean pattern $(x_1, x_2, ..., x_n)$ to $(x_1, x_2, ..., x_{j-1}, x_j \oplus x_{i_1}^{\sigma_1} x_{i_2}^{\sigma_2} ... x_{i_k}^{\sigma_k}, x_{j+1}, ..., x_n)$.

Such gates were considered in [5]. The cascades of these gates are capable of realizing any multiple output Boolean function [5]. We call this model reversible cascades with minimal garbage (RCMG). This reflects the reason of their introduction. The complexity of a network in this model is the number of gates in it.

We will treat EXOR PLA as a standard model for network realizing EXOR sum-of-products (ESOP). For more information on ESOPs see [8]. The complexity of an ESOP is the number of products it contains. This model was chosen for the comparison, since it is very similar to the one defined above.

II. Comparison of the model to EXOR PLA

To exploit similarity between the two chosen models note that each of the terms in ESOP can be treated as a separate gate when all of them are arranged in a form of cascade, a string: EXOR of terms builds the ESOP polynomial. The following summarizes the similarities and differences of models.

- Both models use the same operations: AND, EXOR, and negation.
- The gates are similar. Each gate acts as EXOR of the term built from the input variables. The difference is that in ESOP the set of input variables is not changing while passing through the gates, where for RCMG this is not true.
- The number of distinct gates is comparable: $n * 3^{n-1}$ for RCMG and 3^n for the ESOP.
- The gates are in a linear order: terms being "exored" form a string, and generalized Toffoli gates form a cascade. The difference is in the question whether the order matters. The order of terms in a polynomial does not matter, where the order of reversible gates of RCMG model does.

Lemma 1: By adding a constant input it is possible to use the results of an ESOP minimization to build a reversible network.

Proof: Take an n-input Boolean function and create a zero constants for input line x_{n+1}. This may result in non-optimality of the garbage (at most one higher than the optimal). Transform each term $x_{i_1}^{\sigma_1} x_{i_2}^{\sigma_2} ... x_{i_k}^{\sigma_k}$ to the gate $TOF(x_{i_1}^{\sigma_1}, x_{i_2}^{\sigma_2}, ..., x_{i_k}^{\sigma_k}; x_{n+1})$. Such a transformation of each of the terms in ESOP results in the set of gates of RCMG, which arranged in a cascade form a reversible network for the function. ■

The more interesting question is if the RCMG model is sufficiently beneficial in comparison to ESOP. The answer for this question is "yes" and the following set of the Boolean functions belongs to the class of polynomial complexity for the RCMG and exponential for the ESOP.

Definition 2: For every even integer n a Boolean function $exphard_n(x_1, x_2, ..., x_n)$ is defined as $(x_1 \oplus x_2)(x_3 \oplus x_4)...(x_{n-1} \oplus x_n)$.

Lemma 2: Function $exphard_n$ can be realized with cost $1 + \frac{n}{2}$ in terms of the RCMG model.

Proof: The cascade of gates $TOF(x_1; x_2)$ $TOF(x_3; x_4)$ ... $TOF(x_{n-1}; x_n)$ $TOF(x_2, x_4, ..., x_{n-1}; x_n)$ defines the structure of the network (Figure 1A). ■

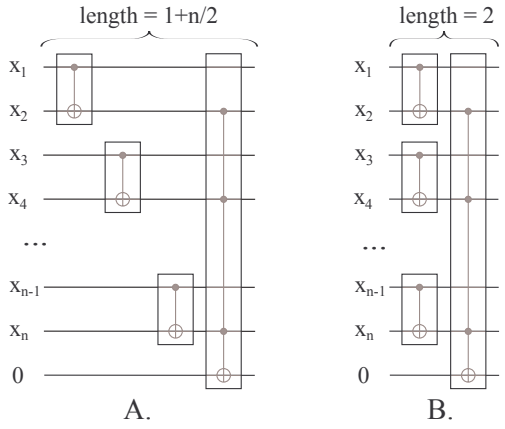

Fig. 1. Reversible design structure.

Note, that in the actual implementation the first $n/2$ gates $TOF(x_1; x_2)$, $TOF(x_3; x_4)$, ..., $TOF(x_{n-1}; x_n)$ form a single layer. The remaining gate, TOF($x_2, x_4, ..., x_n$) forms the second layer. Thus, the total length, or the spatial complexity of the network becomes a constant, namely 2 (Figure 1B).

To show that no better ESOP, other than an ESOP with exponential length can represent function $exphard_n$ we need the following Lemmas:

Lemma 3: Every term in an optimal ESOP for $g(x_1, x_2, ..., x_n, y) = yf(x_1, x_2, ..., x_n)$, where $y \notin \{x_1, x_2, ..., x_n\}$ contains variable y and contains it without negation.

Proof: Let M be an optimal ESOP for the function $g(x_1, x_2, ..., x_n, y)$. Write it as

$$M = yM_1' \oplus M_2' \oplus \bar{y}M_3' = y(M_1' \oplus M_3') \oplus (M_2' \oplus M_3'), \quad (1)$$

where M_1', M_2' and M_3' do not contain y. The total cost of this ESOP is sum of numbers of terms in M_1', M_2' and M_3', which is $|M_1'| + |M_2'| + |M_3'|$. Let N be an ESOP for f. Then yN forms an ESOP for yf. Add yN and M:

$$0 = yf \oplus yf = yN \oplus M = y(M_1' \oplus M_3' \oplus N) \oplus (M_2' \oplus M_3').$$

If we write it by components, we have:

$$M_1' \oplus M_3' \oplus N = 0, M_2' \oplus M_3' = 0. \quad (2)$$

Use the last equality to continue (1):

$$\begin{aligned} yf = M &= y(M_1' \oplus M_3') \oplus (M_2' \oplus M_3') \\ &= y(M_1' \oplus M_3') \oplus 0 \\ &= y(M_1' \oplus M_3') \\ &= yM_1' \oplus yM_3'. \end{aligned}$$

This ESOP has $|M_1'| + |M_3'|$ terms. Since M was minimal, the number of terms of M_2' was zero. Therefore, the number of terms of M_3' is also zero, which can be seen from the second equation in (2). In other words, $M = yM_1'$. ■

Lemma 4: Any optimal ESOP for $g(x_1, x_2, ..., x_n, y) = yf(x_1, x_2, ..., x_n)$, where $y \notin \{x_1, x_2, ..., x_n\}$ has the same complexity as an optimal ESOP for $f(x_1, x_2, ..., x_n)$.

Proof: Use Lemma 3 to say that we can factor variable y out of an optimal ESOP M for the function $g(x_1, x_2, ..., x_n, y)$: $M = yM'$, where M' is an ESOP that doesn't contain y in any form. Let $y = 1$. Then, $M' = (yM')|_{y=1} = M|_{y=1} = (yf(x_1, x_2, ..., x_n))|_{y=1} = f(x_1, x_2, ..., x_n)$. In other words, M' has the complexity of a minimal ESOP for $f(x_1, x_2, ..., x_n)$, so does the ESOP M. ■

Lemma 5: Minimal ESOP for the function $g(x_1, x_2, ..., x_n, y, z) = yf(x_1, x_2, ..., x_n) \oplus zf(x_1, x_2, ..., x_n)$, where y, z are variables and $y, z \notin \{x_1, x_2, ..., x_n\}$ consists of at least $\frac{3|f|}{2}$ terms, where $|f|$ is the number of terms in minimal ESOP for f.

Proof: Take a minimal ESOP M of the function $g(x_1, x_2, ..., x_n, y, z)$ and write it as $M = yM_1 \oplus M_2 \oplus \bar{y}M_3$, where M_1, M_2 and M_3 are ESOPs that do not contain variable y in either term. Such a decomposition is unique. Notice, that the sets of terms in each of M_1, M_2 and M_3 do not intersect:

- $M_1 \cap M_2 = \emptyset$;
- $M_1 \cap M_3 = \emptyset$;
- $M_2 \cap M_3 = \emptyset$.

Otherwise, suppose $M_1 \cap M_2 \neq \emptyset$. Then, there exists a term $t : t \in M_1, t \in M_2$. Since $yt \oplus t = \bar{y}t$, by deleting these two terms from ESOPs M_1 and M_2 and adding it to M_3 we get an ESOP that has complexity one less than the optimal ESOP M. This contradicts to the optimality of M. Therefore, $M_1 \cap M_2 = \emptyset$. The other two set intersections can be proven similarly.

Let $y = 0$ in the ESOP $M = yM_1 \oplus M_2 \oplus \bar{y}M_3$ for the function $yf \oplus zf$. This results in:

$$(yf \oplus zf)|_{y=0} = (yM_1 \oplus M_2 \oplus \bar{y}M_3)|_{y=0}$$

or,

$$zf = M_2 \oplus M_3. \quad (3)$$

By analogy, assignment $y = 1$ leads to

$$\bar{z}f = M_1 \oplus M_2. \quad (4)$$

Add (3) and (4) to get

$$f = M_1 \oplus M_3. \quad (5)$$

Use Lemma 4 to say that each of the ESOPs in (3) and (4) has at least $|f|$ terms. So does the ESOP from (5).

As we proved before, the sets of terms in M_1, M_2 and M_3 do not intersect, so, based on Lemma 4, (3), (4), and (5) the following system can be written:

$$\begin{cases} |M_2 \oplus M_3| = |M_2| + |M_3| \geq |f| \\ |M_1 \oplus M_2| = |M_1| + |M_2| \geq |f| \\ |M_1 \oplus M_3| = |M_1| + |M_3| \geq |f| \end{cases}.$$

Since

$$|M| = |yM_1 \oplus M_2 \oplus \bar{y}M_3| = |M_1| + |M_2| + |M_3|,$$

the problem of finding the number of terms in a minimal ESOP for $(yf \oplus zf)$ is bound by the solution of linear optimization problem:

$$\min \begin{cases} |M_2| + |M_3| \geq |f| \\ |M_1| + |M_2| \geq |f| \\ |M_1| + |M_3| \geq |f| \end{cases} (|M_1| + |M_2| + |M_3|) = ?$$

which is given by expression $\frac{3|f|}{2}$. ■

The proof of the following statement allows to prove the exact bound on the number of terms in a minimal ESOP for $exphard_n$.

Conjecture 1: The minimal ESOP for the function $g(x_1, x_2, ..., x_n, y, z) = yf(x_1, x_2, ..., x_n) \oplus zf(x_1, x_2, ..., x_n)$, where y, z are variables and $y, z \notin \{x_1, x_2, ..., x_n\}$ consists of $2|f|$ terms.

Theorem 1: A minimal ESOP for the function $exphard_n$ has at least $\sqrt{\frac{3}{2}}^n$ terms.

Proof: Can be easily proven by induction using Lemma 5. ■

A better lower bound can be achieved for the best ESOP complexity of $exphard_n$ function by saying that at every time we apply Lemma 5, the actual ESOP lower bound is actually $\lceil \frac{3|f|}{2} \rceil$ (as a natural number, greater than $\frac{3|f|}{2}$), which brings a larger bound into the next step. The final formula for this observation will look like:

$$|M| \geq \lceil \lceil \lceil \frac{3}{2} \rceil * \frac{3}{2} \rceil * ... * \frac{3}{2} \rceil \quad (6)$$

Table I summarizes the results for the function $exphard_n$. First column, **n**, shows the number of inputs. Second column is the number of gates for needed for the model RCMG to realize the function. Third column shows the complexity for the non-reversible application of the RCMG model [5]. We used Exorcism-4 [9], [7] program to calculate the near minimal ESOP for the $exphard_n$ function. The results of this program are summarized in the 4^{th} column. Note, that this column supports the conjecture. The fifth column shows the theoretically proven lower bound on the minimal ESOP, given by formula (6).

III. Multiple Output Functions

One of the reasons why EXOR PLAs are used is ability to share terms. The described modification of RCMG model does not have such a property. But, if we unite the RCMG with mEXOR model [4], this allows the use multiple output EXOR gates, which is equivalent to the term sharing in the ESOP model. Such a hybrid gate will be a new generalization of the Toffoli gate. It can be shown that quantum realization of such hybrid gate has a complexity which differs from the complexity of the original Toffoli gate only marginally. Finally, for such a model either of the synthesis approaches suggested in [5], [1], [6] and [4] is applicable. The theoretical results of this paper will hold for the new hybrid model.

n	RCMG	NRA RCMG	Exorcism-4	ESOP min
2	2	2	2	2
4	3	2	4	3
6	4	2	8	5
8	5	2	16	8
10	6	2	32	12
12	7	2	64	18
14	8	2	128	27
16	9	2	256	41
18	10	2	512	62
20	11	2	1024	93
22	12	2	2048	140
24	13	2	4096	210

TABLE I
COMPLEXITY OF THE FUNCTION $exphard_n$.

IV. Conclusion

We have shown that reversible logic synthesis may be beneficial in comparison with traditional ESOP design. The advantage is not only from the point of zero energy dissipation, which is commonly known and accepted, but also from the point of view of the circuit complexity. We examined the RCMG reversible design model which is very similar to the EXOR PLA and conclude that it is more cost effective. This means that in the worst case the RCMG model cost of a reversible realization of a function has the same cost as the best EXOR PLA result cost when the garbage of the RCMG model might not be optimal. But there are functions for which RCMG model produces better results in practice (comparison of synthesis results for the function rd53.pla in [1]) as well as theoretically: there is a class of functions which have a polynomial complexity for the RCMG model and exponential for EXOR PLA. We do not consider any engineering technology, so the results in an actual implementations may differ.

Acknowledgement

This work was supported in part by a research grant from the Natural Sciences and Engineering Research Council of Canada.

References

[1] G. W. Dueck and D. Maslov. Reversible function synthesis with minimum garbage outputs. In *International Symposium on Representations and Methodology of Future Computing Technologies*, March 2003.

[2] R. Feynman. Quantum mechanical computers. *Optic News*, pages 11–20, 1985.

[3] E. Fredkin and T. Toffoli. Conservative logic. *International Journal of Theoretical Physics*, pages 219–253, 1982.

[4] D. Maslov and G. W. Dueck. Asymptotically optimal regular synthesis of quantum networks. In *International Workshop on Logic Sysnthesis*, May 2003.

[5] D. Maslov and G. W. Dueck. Garbage in reversible design of multiple output functions. In *6th International Symposium on Representations and Methodology of Future Computing Technologies*, March 2003.

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

[7] A. Mishchenko and M. Perkowski. Fast heuristic minimization of exclusive sum-of-products. In *5th International Reed-Muller Workshop*, pages 242–250, Aug. 2001.

[8] T. Sasao. *Switching theory for logic synthesis*. Kluwer Academic Publishers, Norwell, MA, 1999.
[9] N. Song and M. Perkowski. Minimization of exclusive sum of products expressions for multi-output multiple-valued input, incompletely specified functions. *IEEE Trans. on CAD*, 15:385–395, April 1996.
[10] T. Toffoli. Reversible computing. *Tech memo MIT/LCS/TM-151, MIT Lab for Comp. Sci*, 1980.