Compact Quantum Circuits for Dimension Reducible Functions

Anna Bernasconi
Department of Computer Science
Università di Pisa, Italy
anna.bernasconi@unipi.it

Valentina Ciriani
Department of Computer Science
Università degli Studi di Milano, Italy
valentina.ciriani@unimi.it

Asma Taheri Monfared
Department of Computer Science
Università degli Studi di Milano, Italy
asma.taheri@unimi.it

Stefano Zanoni
Department of Computer Science
Università di Pisa, Italy
s.zanoni@studenti.unipi.it

***Abstract*—The classical synthesis method for quantum oracles generally requires a reversible logic synthesis and a quantum compilation step. In the reversible logic synthesis it is important to obtain a compact reversible circuit in order to minimize the quantum cost of the final quantum circuit. In this paper, we exploit function regularities for enabling efficient reversible synthesis. In particular, we propose and implement a new method for the quantum synthesis of Dimension reducible Boolean functions. The experimental results validate the proposed approach showing relevant gains in area.**

***Index Terms*—D-reducibile functions, reversible logic, quantum circuits**

I. Introduction

In the last few years, the technological enhancement in quantum architectures has lead to a renewed and growing interest in quantum computing, as well as to the design of new secure cryptographic protocols. As a consequence, the research in quantum logic synthesis has also attracted considerable attention. In fact, many quantum algorithms, including Grover's search algorithm, usually require to compute *oracles* [1], i.e., subroutines given as classical logic functions.

Standard approaches to synthesize quantum oracles generally consist of two steps: reversible logic synthesis and quantum compilation. Indeed, since the evolution of quantum systems is described by reversible unitary operators, logic functions must be first realized in terms of classical reversible circuits. Then, each reversible gate must be decomposed into a sequence of elementary unitary quantum gates, according to a given quantum gate library. These two steps should be performed with the goal of minimizing the overall gate count of the quantum circuits to be executed. Recently, new methods for reversible circuit synthesis and quantum compilation have been proposed in the literature. Among these, a method exploiting a structural regularity called *autosymmetry* to synthesize compact quantum circuits is presented in [2].

In this paper we propose and implement a strategy for the quantum synthesis of Boolean functions exhibiting a different structural regularity called *Dimension reducibility* (i.e., *D-reducible* functions). These functions can still be formalized through the XOR operators. Intuitively, all the minterms of a *D-reducible* function are entirely contained within an *affine space* that is strictly smaller than the whole Boolean cube $\{0,1\}^n$. D-reducible functions are sufficiently common among classical benchmarks to make the case interesting: experimental results in [3] show that about 70% of the functions in the classical ESPRESSO benchmark suite [4] have at least one D-reducible output. Moreover, the D-reducible decomposition can be computed in polynomial time.

The proposed method is an ad-hoc strategy for the quantum synthesis of D-reducible functions. The synthesis method is based on the structural decomposition of these functions and enables the standard quantum compilation in finding more compact quantum circuits. The theoretical part of the paper shows that we can obtain a reversible circuit, for the given Boolean function, without adding any new input line. Moreover, the final reversible circuit exhibits an uncomputing part implemented with CNOT gates only, i.e., no T-gates are required. We can notice that this aspect is particularly important, since T-gates are more expensive then CNOTs.

The quantum compilation phase produces compact quantum circuits as validated by the experimental results. The experiments show that the proposed strategy allows to compute compact quantum circuits for D-reducible functions, showing gains in area (measured in terms of the number of elementary quantum gates) of about 38%, starting from standard ESOP forms. Moreover, considering the XAG-based quantum compilation [5], the experimental results show a gain in area (measured in terms of T gates) of about 21%. Notice that XAG quantum based compilation usually provides very compact implementations. Therefore, even for this setting, the gain in area is not negligible.

II. Preliminaries

A. Dimension Reducible Functions

A special type of regularity can be observed in dimension reducible (*D-reducible*) functions [3], [6], [7], which is based on *affine spaces* [8], [9]. Recall that a vector subspace V of the classical Boolean vector space $(\{0,1\}^n, \oplus)$ is a subset of $\{0,1\}^n$ containing the zero vector $\mathbf{0} = (00\ldots0)$, such that for each v_1 and v_2 in V we have that $v_1 \oplus v_2$ is still in V. Recall that if $\alpha \in \{0,1\}^n$ is a Boolean point (or vector), the set $A = \alpha \oplus V = \{\alpha \oplus v \mid v \in V\}$ is an *affine space* over V with *translation point* α. The dimension of A is the dimension of the corresponding vector space V. An *affine space* can be algebraically represented by a *pseudoproduct* consisting of an AND of XORs or literals [8]. There are several ways to express *affine space* characteristic functions as a pseudoproduct, out of them we use the *canonical expression* (CEX) [10]. Consider the points of V and A sorted in binary ordering. In the vector space V, rows are indexed from 0 to $2^k - 1$ and the points with indices $2^0, 2^1, 2^2, ..., 2^{k-1}$ form the *canonical basis* B_A of V. The *canonical translation point* α_A is the point of A with index 0. Generally, the canonical representation of an *affine space* is given by its canonical translation point and its canonical basis. In each canonical basis vector, the variable corresponding to the first 1-component from left is called *canonical variable*. The variables that are not canonical in the canonical basis are called *non-canonical variables*.

Starting from these definitions, we can now describe the regular functions that are *D-reducible*. Intuitively, the points (i.e., minterms) of *D-reducible* functions are entirely contained within an *affine space* that is strictly smaller than the whole Boolean cube $\{0,1\}^n$. When a function f is *D-reducible*, it can be written as $f = \chi_A \cdot f_A$, where A is the smallest affine space that contains f and it is called the associated *affine space* of f. Moreover, χ_A is the characteristic function of A and f_A is the projection of f onto A.

Example 1: The Karnaugh map on the left side of Figure 1 illustrates a D-reducible function. According to the Karnaugh map on the right side of the figure, the new function f_A depends on two variables. It should be noted that although the number of the onset minterms is the same for both f and f_A, they are compacted in a smaller space (map) in the f_A function. If we synthesize f and f_A in the classical SOP framework, we obtain f and f_A equal to $x_1x_2\overline{x}_3x_4 + x_1\overline{x}_2x_3$ and $\overline{x}_2 + x_2x_4$, respectively. As a result, the overall number of products is unchanged, and the overall number of literals is reduced from 7 to 3. Moreover, the canonical basis can be derived in polynomial time exploiting the Gauss-Jordan elimination as described in [11]. Finally, a more compact form for the function f can be derived as $x_1(x_2 \oplus x_3)(\overline{x}_2 + x_2x_4)$, where $x_1(x_2 \oplus x_3)$ is the CEX representing χ_A.

B. ESOP forms

In the Exclusive-or Sum-of-Products (ESOP) form, a Boolean function is represented by multi-input AND gates on one level followed by one multi-input XOR gate on the second level. Nowadays, ESOP forms are deeply studied due to their applications in emerging technologies. In comparison with standard SOP, the ESOP form offers several advantages, including the need for fewer products to realize randomly generated functions [12], more compactness for arithmetic or communication circuits [13], higher testability properties [14] and security [15], [16]. Moreover, since the XOR operation is reversible inherently [17], ESOP forms are essential for the synthesis of reversible logic circuits and quantum computing [18]. ESOP minimization is a computationally very hard problem that has long attracted the attention of algorithm designers. Several heuristic and exact methods have been designed for this purpose [17], [19].

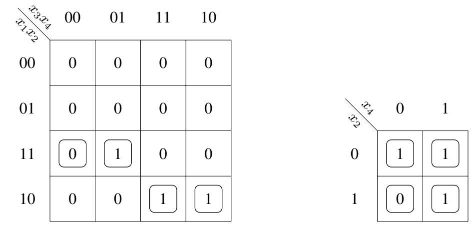

Figure 1. Karnaugh maps of a D-reducible function f (left) and its corresponding projection f_A (right).

Table I
THE COST OF k-CONTROLLED TOFFOLI GATES IN NUMBER OF T, H AND *CNOT* GATES

k	T	H	*CNOT*	ancillary qubits
2	7	2	6	0
3	16	6	14	1
≥ 4	8k-8	8k-12	4k-6	$\lceil \frac{k-2}{2} \rceil$

C. Reversible Circuits and Quantum Compilation

In a *reversible circuit*, there is a one-to-one mapping between the input and output vectors, as a result the number of outputs and inputs are the same. In this kind of circuit, any constant input is known as an *ancilla input*, and any output that is generated to preserve one-to-one mappings, but it is not a useful output is known as a *garbage output*. In general, reversible circuits consist of a sequence of *Mixed-polarity Multiple Control (MPMC) Toffoli gates*.

Given a set of circuit lines $X = x_1, x_2, ..., x_n$, an *MPMC Toffoli gate* $T(C, t)$ has *control lines* $C = \{x_{j1}, x_{j2}, ..., x_{jc}\} \subset X$ and a *target line* $t \in X \setminus C$. The gate maps $t \rightarrow t \oplus (x_{j1}^{p1} \wedge x_{j2}^{p2} \wedge \ \ldots \ \wedge x_{jl}^{pl})$, where each literal x_{ji}^{pi} is either a propositional variable $x_{ji}^1 = x$ or its negation $x_{ji}^0 = \bar{x}$. All remaining other lines are passed through unaltered. Note that, whenever t is equal to 0, the MPMC Toffoli gate computes the AND of all control lines on the target line. Two MPMC Toffoli gates are depicted in Figure 5: the conventional notation $\oplus$ indicates the target line; control lines are denoted by • to indicate positive control connections, and by ◦ to indicate negative control connections.

An MPMC Toffoli gate without any control connection is known as a *NOT gate*, with a single positive control connection is called a *Controlled-NOT (C-NOT) gate*, and with only positive controls is called *Multiple-control Toffoli gate* [1].

The natural correspondence between product terms in ESOP forms and MPMC Toffoli gates makes ESOP-based synthesis widely utilized in reversible logic synthesis [18], [20], [21]: a sequence of MPMC Toffoli gates can be extracted from an ESOP expression, where literals in each product term become control lines in the corresponding MPMC Toffoli gate. Since all these gates act on the same target line, the overall circuit computes the XOR of all product terms on the target line.

Reversible circuit synthesis plays an important role in quantum computing, as it represents a first step towards the construction of quantum circuits. To convert a reversible circuit containing MPMC Toffoli gates into a functionally equivalent quantum circuit, an additional *quantum compilation* step is required: each reversible gate must be decomposed into a sequence of elementary quantum gates, according to a given quantum gate library [22], [23]. In this work, we will use the *Clifford+T* library for this mapping. This library is composed of the Pauli, Hadamard, *CNOT* gates and of the T gate, which is considered the most expensive one (we refer the reader to [1] for more details on elementary quantum gates).

Table I reports the classical cost in terms of Hadamard, *CNOT*s, T gates, and ancillary qubits of the realization of k-controlled MPMC Toffoli gates with the algorithm described in [24]. New quantum compilation heuristics have been proposed, able to synthesize compact circuits [5], [25], [26].

III. Reversible circuits for D-reducible functions

As already reviewed in the previous section, there is a natural correspondence between ESOP expressions and reversible circuits, based on the fact that a product terms in an ESOP expression can be easily represented with a reversible MPMC Toffoli gate. Thus, given an ESOP expression, one can extract a sequence of MPMC Toffoli gates whose control lines correspond to the literals in the product terms of the ESOP form, and whose target line corresponds to the output of the function. Notice that all gates act on the same target line, thus realizing the exclusive OR sum of all product terms. The main problem of this approach is due to the fact that MPMC Toffoli gates are only an intermediate representation. They need to be mapped into elementary quantum gates using an additional synthesis step, whose goal is to transform the reversible circuit of MPMC Toffoli gates into a functionally equivalent quantum circuit implementation. Unfortunately, this mapping step might be very onerous, especially for MPMC Toffoli gates controlled by many variables, thus leading to quantum circuits of high size and depth.

In this section, we therefore propose to exploit the regularity of D-reducible functions to ease their reversible synthesis, in order to obtain a final quantum circuit of reduced size and depth. Given a D-reducible function $f = \chi_A \cdot f_A$, the idea is to concatenate two reversible circuits: a circuit computing the characteristic function χ_A of the affine space A and a reversible circuit implementing the projection f_A. Since the characteristic function χ_A may depend on all input variables, the inputs of the circuit for χ_A are all variables: non-canonical and canonical ones. The circuit for f_A, instead, depends only on the canonical variables. To compute f, we then need a final Toffoli gate computing the AND between the two subfunctions

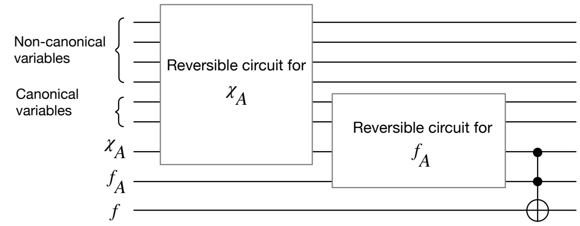

Figure 2. A reversible circuit for a D-reducible function based on the decomposition $f = \chi_A \cdot f_A$.

χ_A and f_A, with f as target line. Note that this approach requires three additional lines (and therefore 3 new qubits in the quantum implementation of the reversible circuit for f): one line for χ_A, one for f_A, and finally one for f. The overall circuit structure is shown in Figure 2.

Let us now discuss how to implement the circuits for χ_A and f_A. The circuit for f_A can be derived from an ESOP representation of the projection f_A. Since f_A depends only on the canonical variables, that are a subset of the input variables, we can reasonably expect that its ESOP representation will contain product terms with less literals. Therefore, the corresponding MPMC Toffoli gates will be controlled by less input lines, thus leading to a quantum circuit of smaller size and depth after the quantum compilation with respect to the *Clifford+T* library. A reversible circuit for χ_A can be implemented directly from its canonical CEX expression. Indeed, recall from Section II-A that a CEX consists of an AND of XOR factors, and each XOR factor can be implemented using CNOT gates. In particular, a XOR factor of k literals can be implemented using $k-1$ CNOTs. Finally, an MPMC Toffoli gate is required to implement the AND of all XOR factors. The number of control lines in input to this gate corresponds to the number of XOR factors, and therefore to the number of non-canonical variables of the affine space A. Interestingly, we do not need to introduce new input lines to represent each XOR factor, as proved in the following proposition.

Proposition 1: Let $\chi_A : \{0,1\}^n \to \{0,1\}$ be the canonical CEX expression representing an affine subspace A of $\{0,1\}^n$. Then, a reversible circuit for χ_A can be implemented without adding new input lines.

Observe that the modification of the non-canonical variables has no effect on the successive calculation of the projection f_A, as f_A depends only on the canonical variables. Once the overall function f has been computed on the output line (the last line in Figure 2), we can restore the non-canonical variables to their initial values applying the so-called *uncomputing procedure* [1]: we uncompute the XOR factors stored in the non-canonical variables by re-applying the CNOTs in reverse order. This disentangles the variables, reverting them to their initial values. The overall methodology is summarized in the algorithm in Figure 3.

Example 2: Consider the function $f = x_1 x_2 \overline{x}_3 x_4 + x_1 \overline{x}_2 x_3$ described in Example 1. This function is D-reducible, with canonical variables x_2 and x_4, and can be decomposed as

INPUT
f /* D-reducible Boolean function with affine space A */
f_A /* projection of f onto A */
χ_A /* characteristic function of A */
OUTPUT
Q /* Quantum circuit for f */

Q_A = ReversibleSynthesis(f_A);
Q_{χ_A} = ReversibleSynthesis(χ_A);
Q = Toffoli(Q_A, Q_{χ_A});
Q = UncomputingProcedure(Q);
return Q

Figure 3. Reversible synthesis of D-reducible functions.

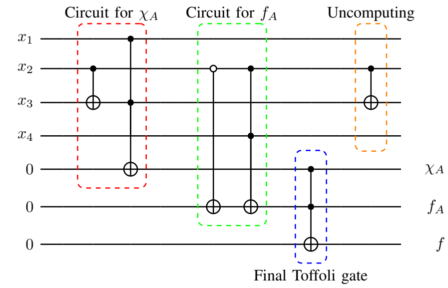

Figure 4. Reversible circuit, with the uncomputing procedure, for the D-reducible function f of Example 2, derived from its decomposition as $f = \chi_A \cdot f_A$. After quantum compilation with the *Clifford+T* library, the size of the circuit becomes equal to 48 elementary quantum gates.

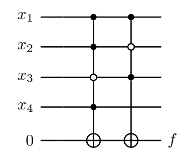

Figure 5. Reversible circuit for the D-reducible function f of Example 2, derived without exploiting the D-reducible decomposition. After quantum compilation with the *Clifford+T* library, the size of the circuit becomes equal to 90 elementary quantum gates.

follows $f = x_1(x_2 \oplus x_3)(\overline{x}_2 + x_2 x_4)$, where $\chi_A = x_1(x_2 \oplus x_3)$ and $f_A = \overline{x}_2 + x_2 x_4$. To derive a reversible circuit of f exploiting this decomposition, we first need to represent f_A in ESOP form. For this example, since the two products $\overline{x}_2$ and $x_2 x_4$ are disjoint, we can immediately derive an ESOP form simply replacing the OR operator with a XOR: $f_A = \overline{x}_2 \oplus x_2 x_4$. To implement the reversible computation of χ_A we only need a CNOT for computing $x_2 \oplus x_3$ onto the line corresponding to the non-canonical variable x_3, and a Toffoli gate for computing the AND between the first factor x_1 and the factor $x_2 \oplus x_3$ (first and second gate in Figure 4). The projection f_A can be computed with two gates corresponding to the two terms in its ESOP form (third and forth gate in Figure 4). Then, the next Toffoli gate computes the AND between χ_A and f_A. Finally, the last CNOT gate implements the uncomputing procedure. Figure 5 shows a reversible circuit derived from the ESOP representation $f = x_1 x_2 \overline{x}_3 x_4 \oplus x_1 \overline{x}_2 x_3$ of the same function. This circuit contains only two MPMC Toffoli gates, and may appear more convenient than the one exploting the D-reducibility properties. However, the first circuit contains Toffoli gates with at most two control variables, while the gates in the second circuit are controlled by up to 4 variables. This has a very important impact on their quantum implementations. Indeed, if we compute the cost in terms of T, H, and CNOT gates of these two implementations (using the costs reported in Table I) we can easily verify that the first circuits requires 21 T gates, 6 H gates and 21 CNOTs, leading to an overall size of **48 gates**. On the other hand, the second circuits has a cost of 40 T gates, 26 H gates and 24 CNOTs, and an overall size of **90 gates**.

Finally, observe that the same strategy can be applied also when the two parts of the decompostion, i.e., f_A and χ_A, are represented in XAG form, as discussed in Section IV.

IV. Experimental results

In this section we evaluate the proposed decomposition in the context of reversible circuit synthesis and quantum compilation. In particular, we conducted two different experimental evaluations. The first one, discussed in Section IV-A, aims at measuring to what extent the D-reducibility property can be exploited to implement compact circuits, in the context of standard reversible synthesis starting from ESOP forms. The aim of the second experimental evaluation, discussed in Section IV-B, is to establish whether more advanced quantum compilation methods could also benefit from the decomposition based on D-reducibility.

These two experimental evaluations have been conducted on D-reducible functions taken from the LGSynth'89 benchmark suite [4]. Since D-reducibility is a property of single outputs, we consider single outputs of the benchmark functions.

A. Standard reversible synthesis

Following the strategy depicted in Figure 2, we implement a reversible circuit for a D-reducible function $f = \chi_A \cdot f_A$. To better evaluate the quality of the reversible circuits derived for D-redicibile funtions, and to compare them with those derived without exploiting the D-reducibility property, we have measured their size in terms of elementary quantum gates. More precisely, instead of considering the overall number of MPMC Toffoli gates, we have mapped each Toffoli gate into elementary quantum gates, considering the Clifford+T low-level quantum gate library and the algorithm described in [24].

Table II reports a significant subset of benchmarks as representative indicators of our experiments. The first column reports the name and the number of the considered D-reducible output of each benchmark. The following group of 3 columns reports the costs, in terms of elementary quantum gates, of the reversible circuits derived from minimal ESOP expressions of the benchmarks, without exploiting the D-reducibility structural regularity. The last group of columns reports the costs of the reversible circuits derived exploiting the D-reducibily decomposition, as explained in Section III. ESOP minimization of the benchmarks and of their projections onto the associated affine spaces is performed using the

Table II
COMPARISON BETWEEN QUANTUM CIRCUITS COMPUTED WITHOUT AND WITH THE D-REDUCIBLE DECOMPOSITION

		without decomposition				with decomposition				
Benchmark_output	**input**	**T**	**H**	**CNOT**	**ancillae**	**T**	**H**	**CNOT**	**ancillae**	**T gain**
b10_3	15	488	448	133	27	406	354	129	23	17%
dk48_2	15	512	492	200	30	110	92	45	5	79%
dk48_4	15	520	500	204	31	80	76	38	5	85%
gary_2	15	488	448	133	27	406	354	129	23	17%
gary_4	15	744	686	246	44	646	588	198	36	13%
in0_3	15	232	216	71	13	181	154	65	9	22%
in0_5	15	856	812	306	53	615	556	198	33	28%
in2_5	19	1520	1410	491	89	1118	1002	307	63	26%
in2_9	19	2144	2002	715	127	1878	1736	581	107	12%
in5_9	24	1568	1492	574	94	1423	1342	504	83	9%
m181_1	15	135	114	26	7	103	70	46	5	24%
newtpla_0	15	200	188	66	12	134	112	38	7	33%
newtpla_2	15	704	652	208	42	382	322	94	20	45%
rckl_6	32	1928	1862	817	120	1814	1748	759	106	6%
spla_21	16	752	724	298	45	142	120	56	7	81%
spla_32	16	1560	1492	592	93	1094	1024	387	64	30%
t2_6	17	494	452	165	29	287	262	91	17	42%
vg2_2	25	1152	1096	421	70	527	462	111	30	54%
vg2_6	25	520	492	182	30	231	182	64	11	56%
vtx1_5	27	4096	3984	1739	244	1287	1154	387	78	69%
Average Benchmark Suite		**379**	**352**	**126**	**22**	**236**	**204**	**80**	**13**	**38%**

Table III
COMPARISON OF THE NUMBER OF T GATES IN QUANTUM CIRCUITS COMPILED WITHOUT AND WITH THE D-REDUCIBLE DECOMPOSITION

Benchmark_output	**input**	**T gates without decomposition**	**T gates with decomposition**	**T gain**
b10_3	15	76	96	-26%
dk48_2	15	84	52	38%
dk48_4	15	76	40	47%
gary_2	15	84	96	-14%
gary_4	15	124	96	23%
in0_3	15	64	52	19%
in0_5	15	156	136	13%
in2_5	19	240	184	23%
in2_9	19	228	292	-28%
in5_9	24	152	168	-11%
m181_1	15	24	24	0%
newtpla_0	15	56	48	14%
newtpla_2	15	80	68	15%
rckl_6	32	120	120	0%
spla_21	16	96	56	42%
spla_32	16	324	88	73%
t2_6	17	68	44	35%
vg2_2	25	120	80	33%
vg2_6	25	68	64	6%
vtx1_5	27	184	116	37%
Average Benchmark Suite		**63**	**49**	**21%**

EXORCISM-4 heuristic [19]. Due to the heuristic nature of this ESOP minimizer, the synthesis times for the functions and their projections are similar and very short, leading to negligible gain in synthesis time. Finally, the last column reports the gain in the number of T gates, and the last row reports the average costs for all the benchmarks considered in our experiments. The gain obtained synthesizing a reversible circuit exploiting this structural regularity is quite interesting. Indeed, the cost gain for T gates is about 38%, the cost gain for H gates is about 42%, the cost gain for CNOTs is about 37% (including the cost of the uncomputing procedure), and the gain in ancillary qubits is about 42%. Notice that the uncomputing procedure does not introduce new T gates.

B. *XAG-based quantum compilation*

We now discuss the experimental results obtained by applying the quantum compilation heuristic proposed in [5]. In particular, we are interested in evaluating experimentally whether this recent technique could benefit from the decomposition of the target function based on the D-reducibility property.

The considered compilation heuristic starts from a XAG representation of a Boolean function f and produces quantum circuits containing elementary quantum gates taken from the Clifford+T gate set. Since, as already observed, the T gate is particularly expensive to be applied, the overall number of T gates is considered a good measure for the cost of the quantum implementation. Therefore, in this experimental evaluation, we only consider the number of T gates in the circuits obtained applying the XAG-based quantum compilation heuristic with and without exploiting the decomposition based on D-reducibility.

A very interesting result shown in [5] is that the number of T gates in a quantum circuit for a Boolean function f can be expressed in terms of the number of AND gates in its XAG representation. This implies that it is possible to provide an upper bound on the number of T gates in a circuit for a function f in terms of its *multiplicative complexity*, i.e., the minimum number of AND gates required to realize f in XAG form. Computing the multiplicative complexity for an arbitrary Boolean function is intractable, however as shown in [27], it is sometimes possible to derive good estimates of this complexity measure exploiting structural regularities of the functions, as for instance the D-reducibility property. We can derive a quantum circuit for a D-reducible function $f = \chi_A \cdot f_A$ combining the quantum circuits obtained applying this heuristic to χ_A and f_A separately. This approach should be convenient since 1) we are able to compute the exact multiplicative complexity of χ_A, and build a XAG with the minimum number of AND gates required to realize χ_A (see Theorem 2 in [27]); 2) f_A is a function that depends on fewer variables, so its XAG representation might contain a reduced number of AND gates (as already experimentally verified in [27]); 3)the two circuit components can be combined by adding a single AND gate, and therefore only 4 T gates in the final quantum circuit. Therefore, this strategy should lead to a reduced number of T gates in the final quantum implementation of the function f. The experimental results confirm this expectation, showing a significant reduction in the number of T gates: compiling a quantum circuit exploiting the decomposition of the target function f as $\chi_A \cdot f_A$, we can obtain a cost gain in T gates of about 21%.

We report in Table III a subset of all the benchmarks that are considered for our experiments. The first column contains the name and the number of the output of the considered benchmark. The second column reports the number of inputs. The following column reports the cost, in terms of T gates, of the quantum circuit compiled without exploiting the D-reducibility regularity. Finally, the last two columns report the cost of the quantum circuits derived exploiting the D-reducibily decomposition, and the gain in the number of T gates. The last row reports the average results for all the benchmarks considered in our experiments.

From the results reported in Table III, we can notice how some benchmarks highly benefit from the proposed strategy. For example, the benchmark *spla_32* shows a gain of 73%, in T gates. For other benchmarks the gain is much less significant, for example *m181_1* and *vg2_6*. There are also cases where the strategy based on D-reducibility gives circuits with a higher number of T gates (see for instance, *b10_3* and *in2_9*). This fact is due to the heuristic nature of both the XAG minimizer, used to derive the initial representation of the function, and of the quantum compiler itself. In general, we can observe how the overall T cost of the XAG based quantum compiler is much lower than the cost of the circuits derived from standard ESOP forms, both for decomposed and non-decomposed benchmarks.

V. Conclusion

In this paper we have considered the class of D-reducible functions and have shown how this regularity can be exploited to implement compact reversible circuits. Experimental results, conducted starting with both standard ESOP or XAG representations of the decomposed expressions, have validated the proposed approach. Future work can consider new regularities for enhancing quantum compilation. Moreover, we plan to investigate whether other decomposition techniques can be exploited for deriving compact quantum circuits.

Acknowledgements

This work was supported in part by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU.

References

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2016.

[2] A. Bernasconi, A. Berti, V. Ciriani, G. D. Corso, and I. Fulginiti, "Xor-and-xor logic forms for autosymmetric functions and applications to quantum computing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–1, 2022.

[3] A. Bernasconi and V. Ciriani, "Dimension-reducible boolean functions based on affine spaces," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 16, no. 2, pp. 1–21, 2011.

[4] S. Yang, "Logic synthesis and optimization benchmarks user guide version 3.0," Microelectronic Center, User Guide, 1991.

[5] G. Meuli, M. Soeken, E. Campbell, M. Roetteler, and G. D. Micheli, "The Role of Multiplicative Complexity in Compiling Low T-count Oracle Circuits," in *Proceedings of the International Conference on Computer-Aided Design, ICCAD 2019, Westminster, CO, USA, November 4-7, 2019*, D. Z. Pan, Ed. ACM, 2019, pp. 1–8.

[6] A. Bernasconi and V. Ciriani, "Dredsop: Synthesis of a new class of regular functions," in *9th EUROMICRO Conference on Digital System Design (DSD'06)*. IEEE, 2006, pp. 377–384.

[7] ——, "Logic synthesis and testability of d-reducible functions," in *2010 18th IEEE/IFIP International Conference on VLSI and System-on-Chip*. IEEE, 2010, pp. 280–285.

[8] V. Ciriani, "Synthesis of spp three-level logic networks using affine spaces," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 10, pp. 1310–1323, 2003.

[9] P. M. Cohn, *Algebra: v. 1*, 2nd ed. Chichester, England: John Wiley & Sons, Jun. 1982.

[10] F. Luccio and L. Pagli, "On a new boolean function with applications," *IEEE transactions on computers*, vol. 48, no. 3, pp. 296–310, 1999.

[11] R. A. Liebler, *Basic matrix algebra with algorithms and applications*. Chapman and Hall/CRC, 2018.

[12] T. Sasao, "Exmin2: a simplification algorithm for exclusive-or-sum-of-products expressions for multiple-valued-input two-valued-output functions," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, no. 5, pp. 621–632, 1993.

[13] ——, "Representations of logic functions using exor operators," *Representations of discrete functions*, pp. 29–54, 1996.

[14] U. Kalay, D. V. Hall, and M. A. Perkowski, "A minimal universal test set for self-test of exor-sum-of-products circuits," *IEEE Transactions on Computers*, vol. 49, no. 3, pp. 267–276, 2000.

[15] V. Kolesnikov and T. Schneider, "Improved garbled circuit: Free xor gates and applications," in *International Colloquium on Automata, Languages, and Programming*. Springer, 2008, pp. 486–498.

[16] T. Mizuki, T. Otagiri, and H. Sone, "An application of esop expressions to secure computations," *Journal of Circuits, Systems, and Computers*, vol. 16, no. 02, pp. 191–198, 2007.

[17] H. Riener, R. Ehlers, B. d. O. Schmitt, and G. D. Micheli, "Exact synthesis of esop forms," in *Advanced boolean techniques*. Springer, 2020, pp. 177–194.

[18] K. Fazel, M. A. Thornton, and J. E. Rice, "Esop-based toffoli gate cascade generation," in *2007 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*. IEEE, 2007, pp. 206–209.

[19] A. Mishchenko and M. Perkowski, "Fast heuristic minimization of exclusive-sums-of-products," *5th International Reed-Muller Workshop*, 2001.

[20] R. Drechsler, A. Finder, and R. Wille, "Improving esop-based synthesis of reversible logic using evolutionary algorithms," in *European Conference on the Applications of Evolutionary Computation*. Springer, 2011, pp. 151–161.

[21] P. Gupta, A. Agrawal, and N. K. Jha, "An algorithm for synthesis of reversible logic circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 11, pp. 2317–2330, 2006.

[22] D. Maslov, G. W. Dueck, and D. M. Miller, "Synthesis of fredkin-toffoli reversible networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 6, pp. 765–769, 2005.

[23] D. M. Miller, D. Maslov, and G. W. Dueck, "A transformation based algorithm for reversible logic synthesis," in *Proceedings 2003. design automation conference (ieee cat. no. 03ch37451)*. IEEE, 2003, pp. 318–323.

[24] D. Maslov, "Advantages of using relative-phase Toffoli gates with an application to multiple control Toffoli optimization," *Physical Review A*, vol. 93, p. 022311, 2016.

[25] M. Saeedi and I. L. Markov, "Synthesis and Optimization of Reversible Circuits - A Survey," *ACM Comput. Surv.*, vol. 45, no. 2, pp. 21:1–21:34, 2013.

[26] R. Wille, S. Hillmich, and L. Burgholzer, "Efficient and Correct Compilation of Quantum Circuits," in *IEEE International Symposium on Circuits and Systems, ISCAS 2020, Sevilla, Spain, October 10-21, 2020*. IEEE, 2020, pp. 1–5.

[27] A. Bernasconi, S. Cimato, V. Ciriani, and M. C. Molteni, "Multiplicative complexity of XOR based regular functions," *IEEE Trans. Computers*, vol. 71, no. 11, pp. 2927–2939, 2022.