

Decision Diagrams for the Design of Reversible and Quantum Circuits

(Invited Overview Paper)

Robert Wille^{1,2} Philipp Niemann^{2,3} Alwin Zulehner¹ Rolf Drechsler^{2,3}

¹Institute for Integrated Circuits, Johannes Kepler University, 4040 Linz, Austria

²Cyber-Physical Systems, DFKI GmbH, 28359 Bremen, Germany

³Department of Computer Science, University of Bremen, 28359 Bremen, Germany

{robert.wille, alwin.zulehner}@jku.at

{pniemann,drechsle}@informatik.uni-bremen.de

Abstract—Reversible circuits found great interest in the past as an alternative computation paradigm which can be beneficial e.g. for encoder circuits, low power design, adiabatic circuits, verification, and much more. Besides that, reversible circuits provide the basis for many components of quantum circuits, which by themselves emerged as a very promising computing technology that, particularly these days, gains more and more relevance. All that led to a steadily increasing demand for methods that efficiently and correctly design such circuits. Decision diagrams play an important role in the design of conventional circuitry. In the meantime, also their benefits for the design of the newly emerging reversible and quantum circuits become evident. In this overview paper, we review and illustrate past work on decision diagrams for such circuits and sketch corresponding design methods relying on them. By this, we demonstrate how broadly decision diagrams can be employed in this area and what benefits they yield for these emerging technologies.

I. INTRODUCTION

While the vast majority of circuits and systems rely on a conventional computation paradigm, alternative schemes of computations may provide potential for further technologies and/or applications. Reversible circuits are a corresponding example as they have been shown to be beneficial e.g. for coding/encoding [42], low-power design [4], [5], [14], adiabatic circuits [3], [25], verification [2], or on-chip interconnects [36], [38]. In all these applications, the main property of reversible logic, namely that corresponding circuits only realize bijections which map a given input pattern to a unique output pattern and, by this, allow for computations in both directions is exploited.

Besides that and related to reversible logic, the domain of quantum circuits [19] received steadily increasing attention. Here, qubits rather than conventional bits are utilized which allow for exploiting quantum-physical phenomena such as superposition and entanglement. This provides the basis for quantum parallelism, i.e. the ability to conduct operations on an exponential number of basis states concurrently, and, by this, plenty of applications in domains such as quantum chemistry, machine learning, cryptography, search, or simulation

exist where conventional systems reach their limits [12], [18], [28]. Particular in the recent past, this domain gained more and more relevance with companies such as Google, IBM, Microsoft, etc. getting more and more involved. Since all quantum computations are inherently reversible in nature, reversible and quantum circuits employ many similarities which is why many accomplishments in the domain of reversible circuits also can be employed in the domain of quantum circuits.

This broad variety of applications eventually led to a steadily increasing demand for methods that efficiently and correctly design such circuits. Accordingly, this area has been intensely considered by researchers worldwide in the past years (for overviews of the respective work we refer to [10], [26]).

In this overview paper, we put a particular emphasis on corresponding methods which rely on decision diagrams. In fact, decision diagrams such as *Binary Decision Diagrams* (BDDs, [6]), *Kronecker Functional Decision Diagrams* (KFDDs, [8]), or *Binary Moment Diagrams* (BMDs, [7]) played an important role in the design of conventional circuitry and have been applied for numerous design tasks in this domain (see e.g. [9], [15]). And although the concepts of reversible and particularly quantum circuits are fundamentally different to that, some decision diagram solutions have already been proposed and proven useful for design automation in these emerging domains.

More precisely, the most prominent proposals comprise the *Quantum Decision Diagram* (QDD, [1]), the *Quantum Information Decision Diagram* (QuIDD, [31]), the *X-decomposition Quantum Decision Diagram* (XQDD, [34]) as well as the *Quantum Multiple-Valued Decision Diagram* (QMDD, [24]). These decision diagrams have been used in various applications such as synthesis (see e.g. [1], [30]), simulation (see e.g. [11], [33], [39]), and verification (see e.g. [32], [34], [37]) of reversible and quantum circuits.¹

¹For a comprehensive overview of these diagrams we refer to [20, Chap. 3].

In the following, we will review and illustrate selected previous work on decision diagrams for reversible and quantum circuits as well as corresponding design methods relying on them. To this end, we will focus on decision diagrams identical or similar to the QMDD whose concepts are briefly motivated and reviewed in the following section. Afterwards, Section III illustrates how QMDDs can be utilized for typical design tasks such as synthesis, verification, and simulation. During all that, we will not provide a comprehensive description, but aim to sketch the main ideas while referring to the respective original work for a more detailed treatment. By this, we exemplarily demonstrate how broadly decision diagrams are already employed in this area and what benefits they yield for these emerging technologies. References are provided to equip the interested reader with more comprehensive descriptions and implementations.

II. DECISION DIAGRAMS FOR REVERSIBLE AND QUANTUM LOGIC

We start this work by providing a motivation for decision diagrams in reversible and quantum logic. Similar to conventional logic, reversible and quantum function representations suffer from an exponential complexity which is aimed to be coped by decision diagrams. This is discussed in more detail next. After that, an intuition of the main concepts of the decision diagrams considered in this paper is provided.

A. Motivation

Reversible and quantum circuits obviously realize reversible and quantum functions. For details on the background for both, we refer to the respective literature such as [19] and focus on their function representation in the following. In fact, reversible and quantum functions can be represented by matrices defined as follows:

Definition 1. A reversible Boolean function $f : \mathbb{B}^n \rightarrow \mathbb{B}^n$ defines an input/output mapping where the number of inputs is equal to the number of outputs and where each input pattern is mapped to a unique output pattern. This can be described using a permutation matrix describing a permutation π of the set $\{0, \dots, 2^n - 1\}$, i.e. a $2^n \times 2^n$ matrix $P = [p_{i,j}]_{2^n \times 2^n}$ with $p_{i,j} = 1$ if $i = \pi(j)$ and 0 otherwise, for all $0 \leq i, j < 2^n$. Each column (row) of the matrix represents one possible input pattern (output pattern) of the function. If $p_{i,j} = 1$, then the input pattern corresponding to column j maps to the output pattern corresponding to row i .

Example 1. Fig. 1a shows a permutation matrix describing a reversible function that maps e.g. input pattern 00 to the output pattern 10 (denoted by the 1-entry in the first column of the matrix).

Quantum functions are similar, but work on so-called *qubits* rather than conventional (i.e. Boolean) bits. A qubit can

Inputs

x_1	x_2	00	01	10	11
		00	01	10	11
00	01	0	1	0	0
01	10	0	0	1	0
10	11	1	0	0	0
11		0	0	0	1

Outputs

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

(a) Permutation matrix
(b) Unitary matrix

Fig. 1: Representation for reversible and quantum Functions

represent two basis states 0 and 1 as well as superpositions of the two. More formally:

Definition 2. A qubit is a two-level quantum system, described by a two-dimensional complex Hilbert space. The two orthogonal basis states $|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ are used to represent the Boolean values 0 and 1. The state of a qubit may be written as $|x\rangle = \alpha|0\rangle + \beta|1\rangle$, where the amplitudes α and β are complex numbers with $|\alpha|^2 + |\beta|^2 = 1$.

The quantum state of a single qubit is denoted by the vector $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$. The state of a quantum system with $n > 1$ qubits can be represented by a complex-valued vector of length 2^n , called the state vector. According to the postulates of quantum mechanics, the evolution of a quantum system can be described by a series of transformation operations satisfying the following:

Definition 3. A quantum operation over n qubits can be represented by a unitary transformation matrix, i.e. a $2^n \times 2^n$ matrix $U = [u_{i,j}]_{2^n \times 2^n}$ with

- each entry $u_{i,j}$ assuming a complex value and
- the inverse U^{-1} of U being the conjugate transpose matrix (adjoint matrix) U^\dagger of U (i.e. $U^{-1} = U^\dagger$).

Every quantum operation is reversible since the matrix defining any quantum operation is invertible. At the end of the computation, a qubit can be measured causing it to collapse to a basis state. Then, depending on the current state of the qubit, either a 0 (with probability of $|\alpha|^2$) or a 1 (with probability of $|\beta|^2$) results. The state of the qubit is destroyed by the act of measuring it.

Example 2. Consider the quantum operation H defined by the unitary matrix shown in Fig. 1b which is the well-known Hadamard operation [19]. Applying H to the input state $|x\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, i.e. computing $H \times |x\rangle$, yields a new quantum state $|x'\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$. For $|x'\rangle$, $\alpha = \beta = \frac{1}{\sqrt{2}}$. Measuring this qubit would either lead to a Boolean 0 or a Boolean 1 with a probability of $|\frac{1}{\sqrt{2}}|^2 = 0.5$ each. This computation represents one of the simplest quantum circuits—a single-qubit random number generator.

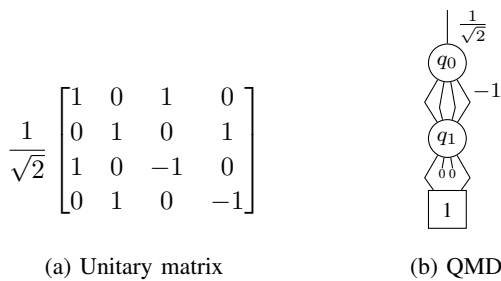


Fig. 2: Representations for $U = H \otimes I_2$.

B. Compact Representation of Matrices

The core idea for obtaining compact representations of the permutation and transformation matrices occurring in the study of reversible and quantum functions is to identify redundancies in the matrices and to represent recurring patterns, i.e. identical or similar sub-matrices, by shared structures.

To this end, a matrix of dimension $2^n \times 2^n$ is partitioned into four sub-matrices of dimension $2^{n-1} \times 2^{n-1}$ as follows:

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{00} & \mathbf{M}_{01} \\ \mathbf{M}_{10} & \mathbf{M}_{11} \end{bmatrix}$$

The partitioning process can recursively be applied to each of the sub-matrices and to each of the subsequent levels of sub-matrices until one reaches the terminal case where each sub-matrix is a single matrix entry.

Now, the core idea of QMDDs [24] is to represent this matrix decomposition in terms of a *Directed Acyclic Graph (DAG)* and to represent identical sub-matrices by shared nodes. As QMDDs additionally allow to annotate edge weights, this also allows to use shared nodes for structurally equivalent matrices that only differ by a scalar factor.

Example 3. Fig. 2b shows the QMDD for the transformation matrix from Fig. 2a. Here, the single root node (labeled q_0) represents the whole matrix and has four outgoing edges to nodes representing the top-left, top-right, bottom-left, and bottom-right sub-matrix (from left to right). This decomposition is repeated at each partitioning level until the terminal node (representing a single matrix entry) is reached. Note that a single node at the q_1 level is sufficient in our case, since the first three 2×2 sub-matrices are identical and the bottom-right matrix differs only by the scalar factor -1 . During the construction of the QMDD, this scalar factor is identified and annotated to the corresponding edge. Similarly, the common multiplier $\frac{1}{\sqrt{2}}$ is extracted and annotated to the root edge.

Moreover, efficient algorithms have been presented for applying operations like matrix addition or multiplication directly on the QMDD data-structure. Overall, QMDDs allow for both, a compact representation as well as an efficient manipulation of permutation/transformation matrices. As a

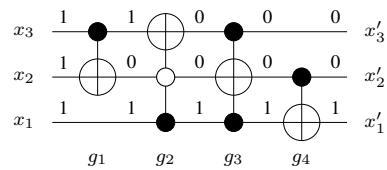


Fig. 3: Reversible circuit

consequence, they have been used in a broad variety of applications in the design of reversible and quantum circuits. This will be discussed in more detail in the following.

III. APPLICATION IN THE DESIGN OF REVERSIBLE AND QUANTUM CIRCUITS

This section illustrates how decision diagrams such as the one reviewed above can be utilized for typical design tasks such as synthesis, verification, and simulation. To this end, we first sketch the premise of the respective design task followed by an illustration of how decision diagrams help in this regard.

A. Synthesis

Synthesis constitutes the task of realizing a reversible or quantum circuit for a given function. This obviously is one of the most important steps in the design of circuits and system as it provides the user with first realizations of the desired function. To this end, a circuit model including a gate library (realizing the respective reversible or quantum operations) is used. In terms of reversible circuits, the commonly used gate library is formed by generalized Toffoli gates.

Definition 4. A Toffoli gate $g_i = \text{TOF}(C_i, t_i)$ is composed of a set $C_i \subseteq \{x_j^+ | x_j \in X\} \cup \{x_j^- | x_j \in X\}$ of positive and negative control lines (where X denotes the set of all circuit lines) and a target line $t_i \in X$ with $\{t_i^-, t_i^+\} \cap C_i = \emptyset$. Furthermore, a line must not occur both as positive and as negative control line in a gate, i.e. $\{x_j^+, x_j^-\} \cap C_i \neq \{x_j^+, x_j^-\}$. Then, the value of the target line t_i is inverted by gate g_i iff all positive (negative) control lines are assigned one (zero). All other lines are passed through the gate unaltered. A cascade of such gates $G = g_1 g_2 \dots g_l$ forms a reversible circuit.

Example 4. Fig. 3 shows a reversible circuit composed of three circuit lines and four Toffoli gates. Furthermore, the circuit lines are annotated with their respective value when applying input combination $x_3 x_2 x_1 = 111$. The first gate g_1 inverts the value of target line x_2 , because the positive control line x_3^+ is assigned 1. Gate g_2 inverts the value of target line x_3 , because the control lines x_2^- and x_1^+ are assigned 0 and 1, respectively. The remaining two gates do not alter the value on any circuit lines, because the control lines are not assigned accordingly.

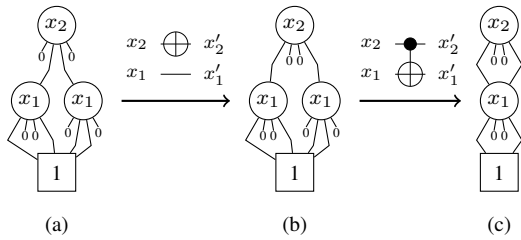


Fig. 4: Effects of applying Toffoli gates to QMDDs

Several approaches have been proposed which conduct synthesis by applying such gates until the represented functionality evaluates to the identity function (typical representatives of such a scheme are transformation-based synthesis [17], [27] as well as approaches based on Reed-Muller expansion [13] or Reed-Muller Spectra [16]). More precisely, assume the function to be synthesized is described by a matrix M . Then, since all reversible and quantum operations are inherently reversible, an inverse M^{-1} exists and their product $M \cdot M^{-1} = I$ yields the identity matrix. Consequently, if a cascade G of gates is determined which transforms M to the identity, a circuit realizing M^{-1} results. Reversing G (easily possible by reversing the gate order and replacing each gate with its inverse) yields a circuit that realizes M . However, methods relying on such a scheme (such as [13], [16], [17], [27]) suffer from the exponential complexity of the function description.

Since decision diagrams often allow for a compact representation of the function to be synthesized, they provide a suitable solution to this problem. Moreover, since they additionally offer efficient capabilities for function manipulation, corresponding transformations can easily be conducted.

Example 5. Consider the root node of the QMDD shown in Fig. 4a. To establish the identity for this node (the top-right and the bottom-left sub-matrix are zero matrices), we apply the gate $\text{TOF}(\emptyset, x_2)$. This simply exchanges the first (third) and the second (fourth) edge of the root node of the QMDD. The resulting QMDD is shown in Fig. 4b. To establish the identity for the right-most x_1 -node, we again need to apply a Toffoli gate with target line x_1 . To ensure that the other node labeled x_1 is not modified either (this node already represents the identity), we add a positive control line x_2^+ to the gate (i.e. $\text{TOF}(\{x_2^+\}, x_1)$). This way, only the nodes that can be reached through the fourth edge of the root node (i.e. the node labeled x_2)—eventually resulting in the QMDD representing the identity shown in Fig. 4c.

Approaches such as proposed in [30] and further improved in [40] successfully realize these concepts and allow for an efficient (and scalable) synthesis of reversible circuits using decision diagrams.

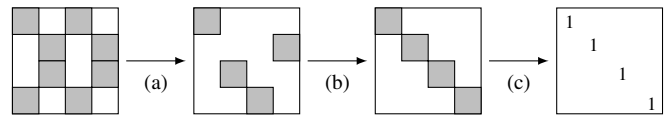


Fig. 5: Proposed global synthesis scheme.

However, in terms of quantum circuits, additionally quantum-mechanical effects (represented by the unitary matrix) have to be considered when determining a gate sequence yielding the identity matrix. Although significantly more complicated, similar approaches can be employed here as well. More precisely, the matrix is transformed in three steps (as also illustrated in Fig. 5):

- (a) *Eliminate superposition*, i.e. apply quantum gates so that all multiple non-zero matrix entries in each column are combined to a single non-zero entry.
- (b) *Move to diagonal*, i.e. apply quantum gates which move the remaining non-zero entries to the diagonal of the matrix.
- (c) *Remove phase shifts*, i.e. apply quantum gates which transform the diagonal entries to 1—eventually yielding the identity matrix.

Also these steps can accordingly be conducted using decision diagrams as discussed in [21], [23].

Finally, note that all above-mentioned approaches require a reversible description of the function to be synthesized in order to work properly. However, it is often the case that (Boolean) functions are to be realized which are originally described in a non-reversible way, i.e. output patterns are not unique. Then, a so-called *embedding* is to be conducted in the first place. To this end, corresponding extensions (e.g. in terms of additional primary outputs; called *garbage outputs*) are employed on the function which allow to explicitly distinguish non-unique output patterns—making the function reversible. Since also here, a function in its entirety has to be considered, decision diagrams as a means for compact representation have successfully been utilized for this purpose (see e.g. [41]). Moreover, they even have been employed to completely get rid of this extra step and, instead, do a *one-pass synthesis* scheme which combines embedding and synthesis (see e.g. [43], [44]).

B. Verification

Verification means the task of checking whether two structurally different function descriptions are functionally equivalent or not. Typical use cases include e.g. the situation in which a designer wants to confirm whether the generated circuit indeed realizes the desired function or whether an optimization conducted on a circuit did not change its functionality. Those are, in general, exponentially hard problems.

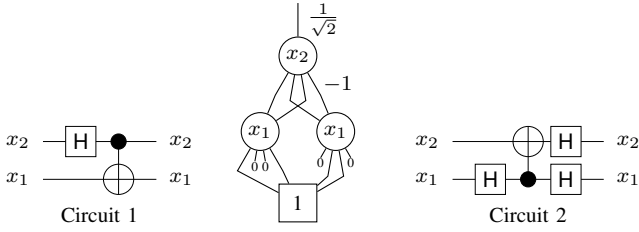


Fig. 6: Equivalence checking for quantum circuits

For tasks like these, decision diagrams have already received a well-known reputation for conventional circuits since the corresponding representations are inherently canonic (assuming a fixed variable order) [6]. Luckily, for the decision diagrams considered here, the same property exists. In fact, as proven in [24], the representation illustrated in Section II-B is canonic. Accordingly, two reversible or quantum functions can easily be verified by simply generating the decision diagram in the same fashion and comparing them (this has e.g. been evaluated in [24], [37]).

Example 6. Consider the two quantum circuits shown in Fig. 6. To determine whether these two circuits are equivalent, we construct a QMDD describing the functionality for each circuit. Since both circuits lead to the same QMDD (also shown in Fig. 6), their equivalence is proven.

Moreover, this technique can be generalized for multiple-valued reversible and quantum functionality as demonstrated in [22].

C. Simulation

Simulation constitutes the task of determining the output state for a given input state applied to a reversible or quantum circuit. For reversible circuits, simulation is rather trivial as basically only Boolean values are applied on the inputs which can easily be evaluated considering reversible gates such as Toffoli gates. Fig. 3 nicely shows this: the input pattern can easily be propagated from left to right by checking whether all control lines of a gate are set to the appropriate value (i.e. 1 for positive and 0 for negative controls) and flipping the target line accordingly, while, at the same time, all remaining values are passed through the gate unaltered.

In case of quantum circuits, however, a substantially harder problem results.² In fact, in this case, the respective input state has to be provided in terms of a state vector so that it can be evaluated with respect to a unitary matrix representing the quantum operation to be simulated. The simulation step itself can then be conducted by a matrix-vector multiplication.

²This also does not come with a surprise since, if simulating quantum circuits would be trivial on a conventional machine, there would be no need for a quantum circuit in the first place.

Example 7. Consider a quantum system composed of two qubits which is currently in state $|x\rangle = |00\rangle$. Applying an H -operation to the first qubit (as defined by the matrix shown in Fig. 2a) yields a new state vector determined by

$$|x'\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$$

Also here, decision diagrams as reviewed above in Section II-B can help as they already provide a compact representation for unitary matrices. However, additionally a representation of state vectors is required. Moreover, the quantum operations to be simulated (and, accordingly, methods to manipulate matrices *and* vectors) as well as the (non-reversible) measurement step needs to be supported. To this end, corresponding solutions using decision diagrams have recently been proposed in [39]. Eventually, this led to substantial improvements with respect to currently available simulators such as *LIQUi* [35] from Microsoft or *qHiPSTER* [29] from Intel. In fact, while e.g. *LIQUi* is capable of simulating Shor's Algorithm (a well-known quantum method for factorization [28]) for at most 31 qubits in more than 30 days, the simulation approach based on decision diagrams completes this task within a minute—showing an impressive display of the capabilities of these data-structures.

IV. CONCLUSIONS

In this work, we provided an overview of decision diagrams for reversible and quantum circuits as well as their potential for the design of these emerging technologies. While already established in conventional circuit design for many decades, decisions diagrams for the area considered here are still not that common. However, with the approaches and the potential from the recent past as discussed in this work as well as alternative diagram types such as QDDs, QuIDDs, or XQDDs, a case can be made that decision diagrams might become similarly important for reversible and quantum circuit design as they have done for the design of conventional circuits and systems.

ACKNOWLEDGMENTS

We sincerely thank all co-authors and collaborators who worked with us in the past in this exciting area.

This work has partially been supported by the European Union through the COST Action IC1405.

REFERENCES

- [1] A. Abdollahi and M. Pedram. Analysis and synthesis of quantum circuits by using quantum decision diagrams. In *Design, Automation and Test in Europe*, pages 317–322, 2006.
- [2] L. G. Amarù, P. Gaillardon, R. Wille, and G. D. Micheli. Exploiting inherent characteristics of reversible circuits for faster combinational equivalence checking. In *Design, Automation and Test in Europe*, pages 175–180, 2016.
- [3] W. C. Athas and L. J. Svensson. Reversible logic issues in adiabatic cmos. In *Physics and Computation, 1994. PhysComp '94, Proceedings., Workshop on*, pages 111–118, Nov 1994.
- [4] C. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17(6):525–532, Nov 1973.
- [5] A. Berut, A. Arakelyan, A. Petrosyan, S. Ciliberto, R. Dillenschneider, and E. Lutz. Experimental verification of Landauer’s principle linking information and thermodynamics. *Nature*, 483:187–189, 2012.
- [6] R. E. Bryant. Graph-based algorithms for Boolean function manipulation. *IEEE Trans. on Comp.*, 35(8):677–691, 1986.
- [7] R. E. Bryant and Y. Chen. Verification of arithmetic circuits with binary moment diagrams. In *Design Automation Conf.*, pages 535–541, 1995.
- [8] R. Drechsler and B. Becker. Ordered Kronecker functional decision diagrams—a data structure for representation and manipulation of boolean functions. *IEEE Trans. on CAD*, 17(10):965–973, 1998.
- [9] R. Drechsler, J. Shi, and G. Fey. Synthesis of fully testable circuits from BDDs. *IEEE Trans. on CAD*, 23(3):440–443, 2004.
- [10] R. Drechsler and R. Wille. From truth tables to programming languages: Progress in the design of reversible circuits. In *Int’l Symp. on Multiple-Valued Logic*, pages 78–85, 2011.
- [11] D. Goodman, M. A. Thornton, D. Y. Feinstein, and D. M. Miller. Quantum logic circuit simulation based on the QMDD data structure. In *Int’l Reed-Muller Workshop*, 2007.
- [12] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Symposium on the Theory of Computing*, pages 212–219, 1996.
- [13] P. Gupta, A. Agrawal, and N. K. Jha. An algorithm for synthesis of reversible logic circuits. *IEEE Trans. on CAD*, 25(11):2317–2330, 2006.
- [14] R. Landauer. Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5(3):183–191, 1961.
- [15] S. Malik, A. Wang, R. Brayton, and A. Sangiovanni-Vincentelli. Logic verification using binary decision diagrams in a logic synthesis environment. In *Int’l Conf. on CAD*, pages 6–9, 1988.
- [16] D. Maslov, G. W. Dueck, and D. M. Miller. Techniques for the synthesis of reversible Toffoli networks. *ACM Trans. on Design Automation of Electronic Systems*, 12(4), 2007.
- [17] D. M. Miller, D. Maslov, and G. W. Dueck. A transformation based algorithm for reversible logic synthesis. In *Design Automation Conf.*, pages 318–323, 2003.
- [18] A. Montanaro. Quantum algorithms: an overview. *npj Quantum Information*, 2:15023, 2016.
- [19] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge Univ. Press, 2000.
- [20] P. Niemann and R. Wille. *Compact Representations for the Design of Quantum Logic*. Springer, 2017.
- [21] P. Niemann, R. Wille, and R. Drechsler. Efficient synthesis of quantum circuits implementing Clifford group operations. In *ASP Design Automation Conf.*, pages 483–488, 2014.
- [22] P. Niemann, R. Wille, and R. Drechsler. Equivalence checking in multi-level quantum systems. In *Conference on Reversible Computation*, pages 201–215, 2014.
- [23] P. Niemann, R. Wille, and R. Drechsler. Improved synthesis of Clifford+T quantum functionality. In *Design, Automation and Test in Europe*, 2018.
- [24] P. Niemann, R. Wille, D. M. Miller, M. A. Thornton, and R. Drechsler. QMDDs: Efficient quantum function representation and manipulation. *IEEE Trans. on CAD*, 35(1):86–99, 2016.
- [25] A. Rauchenecker, T. Ostermann, and R. Wille. Exploiting reversible logic design for implementing adiabatic circuits. In *Mixed Design of Integrated Circuits and Systems, 2017 MIXDES-24th International Conference*, pages 264–270. IEEE, 2017.
- [26] M. Saeedi and I. L. Markov. Synthesis and optimization of reversible circuits - a survey. *ACM Comput. Surv.*, 45(2):21, 2013.
- [27] V. V. Shende, A. K. Prasad, I. L. Markov, and J. P. Hayes. Reversible logic circuit synthesis. In *Int’l Conf. on CAD*, pages 353–360, 2002.
- [28] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.
- [29] M. Smelyanskiy, N. P. D. Sawaya, and A. Aspuru-Guzik. qHiPSTER: The quantum high performance software testing environment. *CoRR*, abs/1601.07195, 2016.
- [30] M. Soeken, R. Wille, C. Hilken, N. Przigoda, and R. Drechsler. Synthesis of reversible circuits with minimal lines for large functions. In *ASP Design Automation Conf.*, pages 85–92, 2012.
- [31] G. Viamontes, I. Markov, and J. Hayes. High-performance QuIDD-based simulation of quantum circuits. In *Design, Automation and Test in Europe*, pages 1354–1355, 2004.
- [32] G. F. Viamontes, I. L. Markov, and J. P. Hayes. Checking equivalence of quantum circuits and states. In *Int’l Conf. on CAD*, pages 69–74, 2007.
- [33] G. F. Viamontes, I. L. Markov, and J. P. Hayes. *Quantum Circuit Simulation*. Springer, 2009.
- [34] S.-A. Wang, C.-Y. Lu, I.-M. Tsai, and S.-Y. Kuo. An XQDD-based verification method for quantum circuits. *IEICE Transactions*, 91-A(2):584–594, 2008.
- [35] D. Wecker and K. M. Svore. LIQUi>: A software design architecture and domain-specific language for quantum computing. *CoRR*, abs/1402.4467, 2014.
- [36] R. Wille, R. Drechsler, C. Osewold, and A. G. Ortiz. Automatic design of low-power encoders using reversible circuit synthesis. In *Design, Automation and Test in Europe*, pages 1036–1041, 2012.
- [37] R. Wille, D. Große, D. M. Miller, and R. Drechsler. Equivalence checking of reversible circuits. In *Int’l Symp. on Multiple-Valued Logic*, pages 324–330, 2009.
- [38] R. Wille, O. Keszocze, S. Hillmich, M. Walter, and A. G. Ortiz. Synthesis of approximate coders for on-chip interconnects using reversible logic. In *Design, Automation and Test in Europe*, pages 1140–1143, 2016.
- [39] A. Zulehner and R. Wille. Advanced simulation of quantum computations. *CoRR*, abs/1707.00865, 2017.
- [40] A. Zulehner and R. Wille. Improving synthesis of reversible circuits: Exploiting redundancies in paths and nodes of QMDDs. In *Conf. on Reversible Computation*, pages 232–247, 2017.
- [41] A. Zulehner and R. Wille. Make it reversible: Efficient embedding of non-reversible functions. In *Design, Automation and Test in Europe*, pages 458–463, 2017.
- [42] A. Zulehner and R. Wille. Taking one-to-one mappings for granted: Advanced logic design of encoder circuits. In *Design, Automation and Test in Europe*, pages 818–823, 2017.
- [43] A. Zulehner and R. Wille. Exploiting coding techniques for logic synthesis of reversible circuits. In *ASP Design Automation Conf.*, pages 670–675, 2018.
- [44] A. Zulehner and R. Wille. One-pass design of reversible circuits: Combining embedding and synthesis for reversible logic. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018.