ICCAD: G: Decision Diagrams for Quantum Computing

Stefan Hillmich (Advisor: Robert Wille)
Institute for Integrated Circuits, Johannes Kepler University Linz, Austria
stefan.hillmich@jku.at ACM Member No. 0544476

I. Problem and Motivation

In the 1970s, researchers started to utilize quantum mechanics to address questions in computer science—establishing (among others) the field of quantum computing [10]. Since then, several important and notoriously difficult problems have been tackled with the power of this new computing paradigm. Famously, Shor's algorithm [11] promises to efficiently factorize integers on a quantum computer—allowing to break widely employed encryption schemes once quantum computers with enough qubits are available. Besides that, there are applications within reach of the current generation of quantum computers, including quantum chemistry [12], quantum finance [13], and machine learning [14].

The tremendous impact quantum computing may have hinges on *Computer-Aided Design* (CAD) keeping up with the increasing power of physical realizations. The complexity of quantum computing has to be tackled with dedicated methods and data structures as well as a close cooperation between the CAD community and physicists. In the classical realm, decision diagrams have proven to be an excellent tool in many cases—a perfectly suitable data structure for solving numerous problems. However, in the quantum realm, decision diagrams have been utilized only for a couple of years with promising first results but, nonetheless, *huge* untapped potential.

In my PhD studies, I strive to unleash this potential. Since starting one and a half years ago, I made the following contributions (among others): I (1) implemented decision diagrams for quantum computing in a fashion that is orders of magnitudes faster than previous implementations, (2) developed an approach utilizing decision diagrams to closely mimic the output of physical quantum computers, (3) exploited the inherent error-resistance of quantum algorithms to enable a precise trade-off between fidelity and compactness, and (4) researched the applicability of concurrency for operations on decision diagrams to exploit current many-core CPUs—a feature taken for granted in the classical world. In the following, I describe these milestones in more detail.

II. Background and Related Work

A. Quantum Computing

In the realm of quantum computing, conventional bits are generalized to *quantum bits* (*qubits*) [10]. While the former can only be in exactly one of the states 0 and 1, qubits may assume two basis states (denoted $|0\rangle$ and $|1\rangle$) as well, but also any linear combination of them. This is described by $|\psi\rangle = \alpha_0 \cdot |0\rangle + \alpha_1 \cdot |1\rangle$ with *amplitudes* $\alpha_0, \alpha_1 \in \mathbb{C}$ and the normalization constraint $|\alpha_0|^2 + |\alpha_1|^2 = 1$. The amplitudes of a quantum state are commonly written as a vector, i.e., $|\psi\rangle = \left[\begin{smallmatrix}\alpha_0\\ \alpha_1\end{smallmatrix}\right]$. Another more visual representation is the Bloch Sphere depicted in Fig. 1, where each point on the surface is a possible state of a single-qubit quantum system.

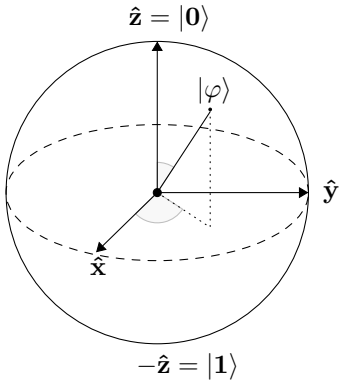


Fig. 1. Bloch Sphere

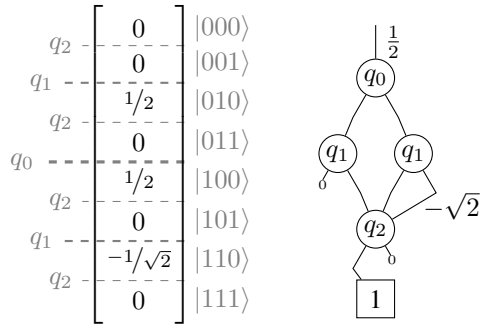
Qubits with non-zero α_0 and α_1 are referred to as being in *superposition*, i.e., they are in "both states" at the same time. Another important phenomenon is *entanglement*, where manipulating a single qubit can affect the state of another qubit. Superposition and entanglement are the main characteristics of quantum computing that allows for substantial speed-ups in certain applications due to the resulting potential for considering multiple states at the same time.

Example 1. *A single-qubit system with equal probabilities (50 % each) to measure either zero or one could have the state* $|\psi\rangle = \frac{1}{\sqrt{2}}\left[\begin{smallmatrix}1\\ 1\end{smallmatrix}\right]$.

Multi-qubit systems work in the same way by assigning an amplitude to every basis state $|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x \cdot |x\rangle$ with normalizing constraint $\sum_{x\in\{0,1\}^n} |\alpha_x|^2 = 1$, where n is the number of qubits. The corresponding vector is 2^n-dimensional—requiring an exponential amount of memory.

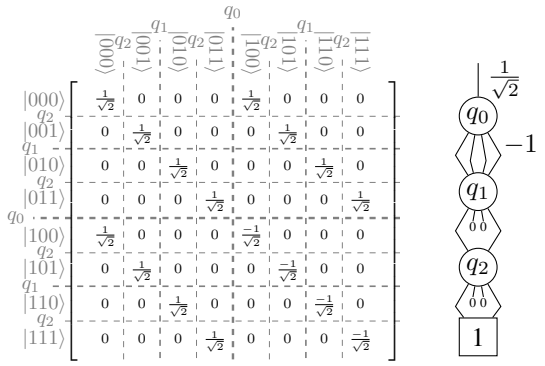
Example 2. *The two-qubit state* $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ *exhibits superposition similar to the previous example. In addition, the qubits are also entangled. In this case, measuring the the first qubit allows us to deduce the state of the second qubit.*

Quantum states can be manipulated by quantum operations, which are represented by $2^n \times 2^n$-dimensional matrices for operations on an n-qubit state—resulting in a double-exponential representation.



(a) Vector (b) DD

Fig. 2. Representation of a quantum state as vector and decision diagram



(a) Matrix (b) DD

Fig. 3. Representation of a quantum operation as matrix and decision diagram

Example 3. *Common operations applied on one and two qubits are the Hadamard operation* H *(setting a qubit in superposition), the* NOT *operation (negating the state of a qubit), and the* CNOT *operation (inverting the second qubit iff the first qubit is in state* $|1\rangle$*). The respective matrices are as follows:*

$$\mathrm{H} = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \quad \mathrm{NOT} = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} \quad \mathrm{CNOT} = \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}$$

In the formalism of vectors and matrices, applying an operation to a state is carried out by matrix-vector multiplication, i.e., $|\psi'\rangle = U \times |\psi\rangle$ for a given operation U and a quantum state $|\psi\rangle$.

Example 4. *Applying the* NOT *operation to the quantum state* $|0\rangle$ *yields*

$$\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} \times \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}0\\1\end{bmatrix} \equiv |1\rangle\,.$$

B. Decision Diagrams

The main idea of decision diagrams in quantum computing is to utilize redundancies in the state vector and to provide compaction by sharing structures [15]–[17]. More precisely, the vector is split in half into two sub-vectors. This process is repeated until the sub-vector contains a single element only, i.e., one split for every qubit. If identical sub-vectors occur in the process, this redundancy is exploited by re-using (sharing) the same structure in the resulting decision diagram.

Example 5. *Fig. 2a and 2b illustrate the quantum state of a three qubit system* $|\psi\rangle = \frac{1}{2}(|010\rangle + |100\rangle) + \frac{1}{\sqrt{2}}|110\rangle$ *as vector and as decision diagram, respectively.*

Similarly to quantum states themselves, operations on quantum states, defined by matrices, can be efficiently represented through decision diagrams.

Example 6. *Fig. 3a and 3b exemplify the quantum operation of setting the first qubit in a three-qubit system into superposition as matrix and as decision diagram, respectively. Every quantum operation acting on a single qubit can be represented by a decision diagram whose number of nodes equals the number of qubits in the system.*

Applying operations to quantum states with decision diagrams is conducted similarly to matrices and vectors since decision diagrams can be multiplied and added just like their counterparts. However, the complexity scales with the number of nodes in the decision diagram which can be kept low for many problems in quantum computing.

Because of that, decision diagrams have been successfully utilized to reduce the memory requirements for representing state vectors and operation matrices in quantum computing [16]–[19]. Moreover, simulation approaches based on decision diagrams have recently moved into the spotlight since they significantly outperform array-based simulators in cases where redundancies can be exploited—in extreme cases leading to an improvement in runtime from 30 days to 2 minutes [20].

III. Approach and Uniqueness

Decision diagrams have shown promising results in regards to design automation for quantum computing but a lot more research has to be done to exploit their full potential. Guided by the impact decision diagrams had on classical design automation, in the first one and a half years of my PhD studies, I made the following contributions towards exploiting this potential:

A. Efficient Representation

The research on decision diagrams has spawned a multitude of different types of decision diagrams (DDs) such as *Quantum Information Decision Diagrams* (QuIDDs [16]), *X-decomposition Quantum Decision Diagrams* (XQDDs [18]), and *Quantum Multiple-valued Decision Diagrams* (QMDDs [19]).

However, no efficient and scaleable implementation has been publicly available thus far. Hence, the first step was to implement a DD package that could be used as a basis for design automation tools. To this end, a versatile package has been created that incorporates the existing expertise from implementing classical decision diagrams as well as dedicated handling of quantum specific requirements. The most challenging new problem was the efficient handling of complex numbers. While mathematically sound, the inherently limited precision on (classical) computers means dedicated solutions are required to ensure scalability. This led to an implementation which focused on the efficient handling of complex numbers while, at the same time, dealing with the limited

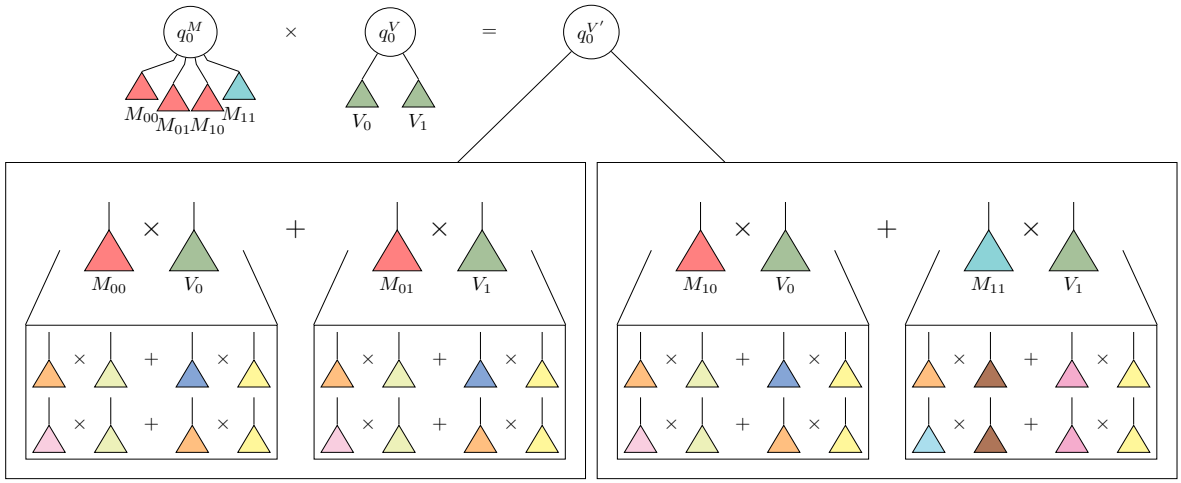


Fig. 4. Sketch of DD-based matrix-vector multiplication with operation M and state V. The triangles represent sub-diagrams, where same colors indicate equal sub-diagrams.

precision provided by (classical) computers [1]. In addition to adapting established techniques from classical DD packages, the numerical stability was ensured and knowledge about the conducted operations (such as multiplication) was exploited to provide a fully-fletched quantum DD package.

B. Simulation

The simulation of quantum computations plays a vitally important role in design automation, since physical quantum computers are prohibitively expensive and the fundamental restrictions at the quantum level do not allow to directly access the amplitudes of a quantum state. To mitigate this issue, simulations on classical computers are conducted for the development and testing of quantum algorithms.

On the conceptual level, the simulation of a quantum computation is simple as it "just" requires matrix-vector multiplication. However, due to the exponential memory needed to store the matrices and vectors, all but the most trivial examples require dedicated approaches to successfully perform the simulation. With decision diagrams' drastically decreased memory consumption and runtime for many quantum algorithms [20], they promise substantially faster simulation times.

Example 7. *Fig. 4 sketches how decision diagrams are multiplied to conduct simulation: Each sub-diagram is recursively traversed until the terminal is reached. The colors indicate different sub-diagrams which, through caching results of previous computations, are reused.*

However, since decision diagrams exploit redundancies in quantum operations and states, they do not perform equally well for the simulation of different algorithms. It is important develop an thorough understanding of advantages and weaknesses of decision diagrams. Hence, the performance of decision diagrams and matrix-vector-style simulation for different quantum algorithms was evaluated [2]. To this end, a compute server with $1.5\,\mathrm{TiB}$ of main memory was used, so both approaches could be applied to circuits with more than 30 qubits.

Besides that, I investigated how to use this as a basis to simulate the *real* output of quantum computers. In fact, the current approach to simulation deviates from physical quantum computers in an important property: The simulation provides the individual amplitudes of the quantum state, which are fundamentally unobservable in physical quantum computers as discussed in Sec. II-A. A simulation approach more closely mimicking physical quantum computers is referred to as *weak simulation* (as opposed to *strong simulation* discussed above) and provides only measurement results instead of amplitudes. I developed fast methods for weak simulation based on decision diagrams [3], which feature a compact representation and a novel algorithm allowing for measurement in linear time with respect to the number of qubits. Compared to the state of the art, this allowed for the first time to faithfully mimic the output of a quantum computer at scales infeasible for array-based approaches.

C. Approximation

Calculations on a physical quantum computer are always tainted by noise and errors—increasingly so with the number of operations performed. Luckily, at the same time quantum algorithms are resistant to errors to some extent even without error correction since small errors only have a small effect on the resulting measurements.

In [4], this resistance to small inaccuracies that hardly affect the results of measurements was exploited for a much more compact state representation. The key idea is to identify quantum states with a low probability of being measured and prune them from the decision diagram. The proposed method allows the user to configure the required *fidelity* (in this sense the exactness) of the quantum state and to get the best approximation that does not drop below the required fidelity. For each node in the decision diagram the "contribution" to the quantum state is calculated and, afterwards, the maximum number of nodes are removed while guaranteeing the required fidelity.

Example 8. *With exaggeration for the sake of this example, assume every amplitude with less than $30\,\%$ chance of being measured is considered neglectable. In Fig. 2b, the nodes labeled q_0 and q_2 have a contribution of $100\,\%$, since they are*

TABLE I
EFFICIENCY OF THE TECHNIQUES PROPOSED FOR THE DD PACKAGE

name	q	$\#op$	$size$	$\#complex$	$t_{original}$	$t_{proposed}$
supremacy_4x4	16	50	80	238	0.00	**0.00**
supremacy_4x4	16	70	6278	18 803	4.04	**0.06**
supremacy_4x4	16	75	36 161	225 560	1 147.14	**0.57**
supremacy_4x4	16	80	1 195 979	–	>3 600.00	**65.18**
supremacy_5x4	20	70	455	3534	0.06	**0.01**
supremacy_5x4	20	80	899	8010	0.26	**0.02**
supremacy_5x4	20	89	71 105	97 942	1 453.63	**2.52**
supremacy_5x4	20	95	1 742 795	–	>3 600.00	**912.11**
supremacy_5x5	25	100	912	6151	0.15	**0.02**
supremacy_5x5	25	110	1751	12 851	0.46	**0.03**
supremacy_5x5	25	119	58 939	96 647	1 282.28	**2.63**
supremacy_5x5	25	120	365 643	–	>3 600.00	**3.51**
qft_15	15	120	32 767	32 785	3.98	**0.39**
qft_16	16	136	65 535	65 554	17.48	**0.86**
qft_17	17	153	131 071	131 091	67.02	**1.64**
qft_18	18	171	262 143	262 164	295.48	**3.50**
qft_19	19	190	524 287	524 309	1 269.35	**8.38**
qft_20	20	210	1 048 575	–	>3 600.00	**22.96**

q: number of qubits *#op*: number of operations
size: size of the DD *#complex*: number of occurring complex values
$t_{original}$: time when using the original QMDD package [19]
$t_{proposed}$: time when incorporating the techniques proposed in this paper

the only nodes representing their respective qubit. For q_1, the left (right) node has a contribution of $25\,\%$ ($75\,\%$). Therefore the left node would be removed from the decision diagram, eliminating one fourth of the nodes.

The fidelity or "exactness" is an example for a possible parameter which heavily influences the performance of simulation. In many quantum algorithms, the correct solution is one of multiple possible measurements—albeit the other outcomes have a very small (but non-zero) probability and are effectively irrelevant. Given such an algorithm, approximation can greatly speed-up the simulation by discarding the unlikely outcomes, i.e., setting their probability to zero.

D. Concurrency

Applications in the classical design automation realm heavily utilize concurrency to take advantage of today's many-core systems. For decision diagrams, concurrency has not yet been taken into account on a comparable level—forgoing a huge potential in modern computers. In [5], I investigated possible concurrency schemes, similar to ones in matrix-vector multiplication, for decision diagrams.

Example 9. *Consider again Fig. 4. In addition to the concept of multiplication of decision diagrams, it shows how the multiplication may be split into sub-tasks, which can be performed concurrently.*

Notably, concurrency schemes for multiplication are not only useful in the context of simulation, but also in verification tasks such as equivalence checking.

IV. RESULTS AND CONTRIBUTIONS

Each contribution was evaluated and showed a significant speed-up compared to previous methods and in some cases it even was the first feasible solution altogether. In the following, I describe a subset of those results.

A. Efficient Representation

The new implementation has been compared against state-of-the-art solutions [19] by building the functional description of Quantum Fourier Transformation (QFT [21]) and the quantum supremacy experiments conducted by Google [22].

TABLE II
RUNTIME AND MEMORY FOR ERROR-FREE SAMPLING OF 1M BITSTRINGS

benchmarks		vector-based		DD-based	
name	qubits	size	t [s]	size	t [s]
qft_16	16	2^{16}	0.12	$16 \approx 2^{4.0}$	**0.22**
qft_32	32	2^{32}	*MO*	$32 \approx 2^{5.0}$	**0.43**
qft_48	48	2^{48}	*MO*	$48 \approx 2^{5.5}$	**0.63**
grover_20	21	2^{21}	0.70	$40 \approx 2^{5.3}$	**0.23**
grover_25	26	2^{26}	17.91	$50 \approx 2^{5.6}$	**0.27**
grover_30	31	2^{31}	993.99	$60 \approx 2^{5.9}$	**0.29**
grover_35	36	2^{36}	*MO*	$70 \approx 2^{6.1}$	**0.43**
shor_33_2	18	2^{18}	0.15	$48\,793 \approx 2^{15.5}$	**0.20**
shor_55_2	18	2^{18}	0.16	$93\,478 \approx 2^{16.5}$	**0.21**
shor_69_4	21	2^{21}	0.62	$196\,382 \approx 2^{17.5}$	**0.26**
shor_221_4	24	2^{24}	3.72	$1\,048\,574 \approx 2^{20.0}$	**0.27**
shor_247_4	24	2^{24}	3.81	$1\,376\,221 \approx 2^{20.3}$	**0.31**
qua_chem_2x2	8	2^{8}	0.04	$117 \approx 2^{6.8}$	**0.09**
qua_chem_3x3	18	2^{18}	0.17	$59\,475 \approx 2^{15.8}$	**0.22**
supremacy_4x4_10	16	2^{16}	0.11	$65\,070 \approx 2^{15.9}$	**0.39**
supremacy_5x4_10	20	2^{20}	0.66	$486\,503 \approx 2^{18.8}$	**0.82**
supremacy_5x5_10	25	2^{25}	12.04	$16\,779\,617 \approx 2^{24.0}$	**4.28**

Table I lists the individual benchmarks along with the size of the resulting decision diagram and the number of complex values encountered during the execution (which are not persisted in the new package). The results clearly show the impact of efficiently handling complex numbers for quantum computing as the proposed package consistently improves the runtime by several orders of magnitude.

B. Simulation

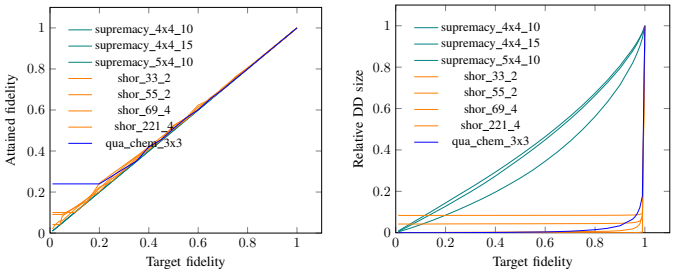
The comparison of matrix-vector- and DD-based simulation approaches [2] highlighted the trade-off between both approaches. The former allows to accurately predict runtime and memory consumption from the number of qubits at the expense of an always exponential complexity. Decision diagrams on the other are hard to predict, but generally perform well until they reach a certain critical size, from where the runtime "skyrockets".

The method for weak simulation introduced in [3] avoids the exponential memory requirement in many important cases and allows to quickly draw samples—mimicking physical quantum computers. The benchmarks include QFT [21], Grover's search [23], Shor's algorithm [24], quantum chemistry [25], and supremacy experiments by Google [22].

Table II shows the comparison of this method against drawing samples from a state vector with a prefix sum for one million samples. Small instances (less than 24 qubits) are executed faster with the vector representation, since vectors require less overhead. However, for larger instances the exponential memory requirements of vectors kick in, while decision diagrams can exploit redundancy to drastically reduce the required memory. On the used setup with $32\,\mathrm{GiB}$ of RAM, decision diagram could handle the sampling when the vector representation failed.

C. Approximation

In [4], the proposed approximation approach was evaluated by generating the resulting quantum states for Shor's algorithm [24], quantum chemistry [25], as well as Google's



(a) Attained fidelity v. target (b) Size v. target fidelity

Fig. 5. Approximation for a target fidelity

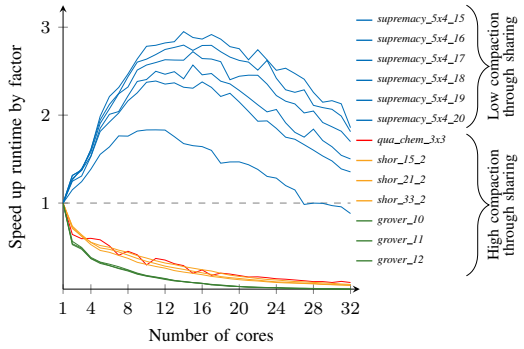


Fig. 6. Performance of concurrent DD-based simulation

supremacy experiments [22] and subsequently approximating these states with different target fidelities.

Fig. 5 shows two graphs: (a) plots the target fidelity (which was an input for the algorithm) against the attained fidelity of the approximated quantum state; (b) plots the target fidelity against the size of the resulting decision diagram relative to the original one. The graphs illustrate that the proposed approach guarantees to not drop below the target fidelity and that Shor's algorithm as well as the quantum chemistry benchmark allow for significantly smaller representations if one does not require $100\,\%$ fidelity. They also show that, even though the supremacy experiments are designed explicitly to not allow for compaction, the proposed scheme achieves effectively reduces the DD size (e.g., with a target fidelity of $50\,\%$, the resulting decision diagram has a relative size of $25\,\%$ in the case of supremacy_5x4_10).

D. Concurrency

I investigated the effects of the concurrency scheme for the simulation of quantum algorithms, based on the simulator from [20]. The benchmarks include QFT [21], Grover's search [23], Shor's algorithm [24], quantum chemistry [25], and supremacy experiments by Google [22].

Fig. 6 illustrates the results as a plot with the number of used cores against the relative speed up compared to the execution with only one core. The plot shows two different trends:

1) Quantum algorithms that *do not allow* for a compact representation, such as the supremacy experiments, benefit from concurrent execution. This is caused by the lower overhead in sharing during the execution, which affects reads and writes to the different kinds of caches used in the simulator.
2) Quantum algorithms that *do allow* for compact representation, such as QFT, quantum chemistry experiments, Grover's search, and Shor's algorithm, have an increased runtime for the simulation. As the caches in the simulator are more frequently hit, synchronization between different execution units becomes more expensive and, hence, decreases the performance.

The results clearly show that the advantage of concurrently using decision diagrams in this fashion depends on the considered functionality. Based on these findings, I am currently investigating new concurrency schemes with a focus on building the functionality, i.e., combining operators, to enable this advantage for quantum algorithms which do not profit from the first scheme.

E. Research Impact

My work conducted on the problems covered in the previous sections has led to the following results:

- An efficient and publicly available implementation of decision diagrams for quantum computing has been published [1]. The experimental evaluation showed that this package is indeed capable of handling complex numbers much more efficiently and allows constructing decision diagrams for established quantum functionality in significantly less runtime (up to several orders of magnitude). This is the basis for the integration into tools developed by big players such as IBM and Atos. Additionally, the paper was chosen as *Best Paper Award Candidate*.
- Studying strengths and weaknesses of DD-based *strong simulation* [2] as well as establishing an approach to *weak simulation* [3] (in collaboration with the University of Michigan) by efficiently providing the measurement results instead of exponentially sized state vectors.
- Further compaction with decision diagrams is possible if small errors are tolerated by approximating the quantum state [4] (in collaboration with the University of Michigan). This led to a reduction of several orders of magnitude in the size while, at the same time, controlling the attained fidelity.
- Utilizing concurrency to speed up DD-based simulation of quantum computations has been examined [5]. This unveiled that the potential of concurrent DD-based simulation heavily depends on the amount of sharing exploitable by the decision diagrams.

Overall, in less than *one and half years* since I started my PhD studies, these key results were published at premier design automation conferences. Additionally, I published in prestigious journals such as TCAD [6] on the compilation/mapping of quantum circuits (in collaboration with the Indian Institute of Technology Kharagpur). Altogether, my PhD studies up until now resulted in seven publications [1]–[7]—most of which at premier venues. Besides that, I contributed to two publications in my studies before starting my PhD [8], [9].

Moreover, my team and I placed 3$^{\text{rd}}$ out of 700 participants at last years IBM Qiskit Challenge and further joint work with IBM Tokyo is currently underway.

Publications During PhD Studies

[1] A. Zulehner, S. Hillmich, and R. Wille, "How to efficiently handle complex values? Implementing decision diagrams for quantum computing," in *Int'l Conf. on CAD*, 2019, The implementation is publicly available at http://iic.jku.at/eda/research/quantum_dd.

[2] T. Grurl, J. Fuß, S. Hillmich, L. Burgholzer, and R. Wille, "Arrays vs. decision diagrams: A case study on quantum circuit simulators," in *Int'l Symp. on Multi-Valued Logic*, 2020.

[3] S. Hillmich, I. L. Markov, and R. Wille, "Just like the real thing: Fast weak simulation of quantum computation," in *Design Automation Conf.*, 2020.

[4] A. Zulehner, S. Hillmich, I. Markov, and R. Wille, "Approximation of quantum states using decision diagrams," in *Asia and South Pacific Design Automation Conf.*, 2020, pp. 121–126.

[5] S. Hillmich, A. Zulehner, and R. Wille, "Concurrency in DD-based quantum circuit simulation," in *Asia and South Pacific Design Automation Conf.*, 2020, pp. 115–120.

[6] A. Kole, S. Hillmich, K. Datta, R. Wille, and I. Sengupta, "Improved mapping of quantum circuits to IBM QX architectures," *IEEE Trans. on CAD of Integrated Circuits and Systems*, 2020.

[7] R. Wille, S. Hillmich, and L. Burgholzer, "Efficient and correct compilation of quantum circuits," in *IEEE International Symposium on Circuits and Systems*, 2020.

Publications Before PhD Studies

[8] A. Deb, R. Wille, O. Keszocze, S. Hillmich, and R. Drechsler, "Gates vs. splitters: Contradictory optimization objectives in the synthesis of optical circuits," *J. Emerg. Technol. Comput. Syst.*, vol. 8, 2016.

[9] R. Wille, O. Keszöcze, 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*, 2016, pp. 1140–1143.

Further References

[10] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*. Cambridge Univ. Press, 2000.

[11] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Jour. of Comp.*, vol. 26, no. 5, pp. 1484–1509, 1997.

[12] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. Sawaya *et al.*, "Quantum chemistry in the age of quantum computing," *Chemical reviews*, vol. 119, no. 19, pp. 10 856–10 915, 2019.

[13] R. S. Lee, *Quantum Finance*. Springer, 2020.

[14] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, 2018.

[15] D. M. Miller and M. A. Thornton, "QMDD: A decision diagram structure for reversible and quantum circuits," in *Int'l Symp. on Multi-Valued Logic*, 2006, p. 6.

[16] G. Viamontes, I. Markov, and J. Hayes, "High-performance QuIDD-based simulation of quantum circuits," in *Design, Automation and Test in Europe*, 2004, pp. 1354–1355.

[17] A. Abdollahi and M. Pedram, "Analysis and synthesis of quantum circuits by using quantum decision diagrams," in *Design, Automation and Test in Europe*, 2006, pp. 317–322.

[18] S.-A. Wang, C.-Y. Lu, I.-M. Tsai, and S.-Y. Kuo, "An XQDD-based verification method for quantum circuits," *IEICE Trans. Fundamentals*, vol. 91-A, no. 2, pp. 584–594, 2008.

[19] P. Niemann, R. Wille, D. M. Miller, M. A. Thornton, and R. Drechsler, "QMDDs: Efficient quantum function representation and manipulation," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 35, no. 1, pp. 86–99, 2016.

[20] A. Zulehner and R. Wille, "Advanced simulation of quantum computations," *IEEE Trans. on CAD of Integrated Circuits and Systems*, 2019.

[21] A. Ekert and R. Jozsa, "Quantum computation and Shor's factoring algorithm," *Reviews of Modern Physics*, vol. 68, no. 3, p. 733, 1996.

[22] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven, "Characterizing quantum supremacy in near-term devices," *Nature Physics*, vol. 14, no. 6, p. 595, 2018.

[23] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Theory of computing*, 1996, pp. 212–219.

[24] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," *Foundations of Computer Science*, pp. 124–134, 1994.

[25] R. Babbush, N. Wiebe, J. McClean, J. McClain, H. Neven, and G. K.-L. Chan, "Low-depth quantum simulation of materials," *Phys. Rev. X*, vol. 8, p. 011044, 2018.