From Biochips to Quantum Circuits: Computer-Aided Design for Emerging Technologies

(Invited Paper)

Robert Wille[1] Bing Li[2] Ulf Schlichtmann[2] Rolf Drechsler[3]
[1]Institute for Integrated Circuits, Johannes Kepler University, Linz, Austria
[2]Institute for Electronic Design Automation, Technical University of Munich, Munich, Germany
[3]Institute of Computer Science, University of Bremen, Bremen, Germany
Cyber Physical Systems, DFKI GmbH, Bremen, Germany
robert.wille@jku.at, b.li@tum.de, ulf.schlichtmann@tum.de, drechsler@uni-bremen.de

ABSTRACT

While previous decades have witnessed impressive accomplishments in the design and realization of conventional computing devices, physical boundaries and cost restrictions led to an increasing interest in alternative technologies (often referred to as *Beyond CMOS* or *More than Moore* technologies). In addition, these accomplishments also triggered many "complementary" applications and led to technologies providing an additional value to the conventional logic (often referred to as *More than Moore*). This led to a variety of emerging technologies such as Quantum Computation, Optical Circuits, or Microfluidic Biochips out of which many are considered very promising and some even entered the market recently. This poses new challenges to researchers and engineers working in computer-aided design. In this tutorial paper, we provide an overview on the main concepts of selected emerging technologies as well as the resulting design methods. To this end, we review the respective technological background and introduce the correspondingly used circuit models. Based on that, we show how computer-aided design has to adapt the common design tasks and review recently proposed solutions.

1. INTRODUCTION

While previous decades have witnessed impressive developments in the design and realization of conventional computing devices, physical boundaries and cost restrictions led to an increasing interest in alternatives. Several promising technologies are currently discussed including:

- Quantum Computation, in which quantum-mechanical effects (e.g. superposition or entanglement) are exploited in order to represent multiple states at the same time and, thus, allow for massive parallelism.
- Optical Circuits, which rely on optical rather than electrical signals and allow for ultra-high-speed networks while having beneficial low-power properties.
- Microfluidic Biochips, in which laboratory procedures in biochemistry and molecular biology are automated by providing a platform in which samples and corresponding operations can be controlled.

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

Although most of these technologies are still in a rather "academic" state, first physical realizations have already been presented. In case of microfluidic biochips, first solutions even entered the market recently.

Besides the underlying physical, biological, or chemical issues to be addressed, also the question how to efficiently design corresponding circuits gains more and more relevance. Here, entirely different concepts and paradigms have to be employed which significantly differ from conventional logic and circuitry. For example, quantum computation works with qubits rather than conventional bit which do not only assume Boolean values 0 and 1 but also superpositions of them. Optical circuits distinguish between electrical and optical signals. Biochips rely on entirely different definitions e.g. of placement and routing.

All that obviously affects the way how circuits and systems for these emerging technologies have to be designed. Methods and data-structures which emerged over the past decades for conventional computer-aided design have to be significantly extended or cannot be applied at all anymore. Alternative solutions capable of addressing the new challenges are currently under development. Once circuits and systems based on these emerging technologies will establish themselves in the near future, expert knowledge on these CAD-methods will be needed.

This tutorial aims to provide an overview on the main concepts of selected emerging technologies, namely quantum computation, optical circuits, and microfluidic biochips, as well as the resulting design methods. For each technology, we review the respective paradigms and introduce the correspondingly used circuit models. Based on that, we show how computer-aided design has to change for common tasks and review recently proposed solutions.

2. QUANTUM CIRCUITS

The first representative is probably one of the best-known emerging technologies: quantum circuits [28] which received significant attention within both, the scientific community but also within the public at large. By exploiting quantum mechanical phenomena, they allow to solve important computation problems significantly faster than on conventional computers (prominent examples include integer factorization [41, 46] or database search [12]). However, this also affects the way computations are conducted: instead of conventional bits, quantum circuits deal with so-called qubits – a new entity which has to be supported by CAD.

2.1 Background

A *qubit* can represent 0 or 1 as well as superpositions of the two. More formally, a *qubit* is a two-level quantum system, described by a two-dimensional complex Hilbert space. Two orthogonal quantum states $|0\rangle \equiv \binom{1}{0}$ and $|1\rangle \equiv \binom{0}{1}$ are used to represent Boolean values 0 and 1. The state of a qubit may be written as $|x\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers and $|\alpha|^2 + |\beta|^2 = 1$.

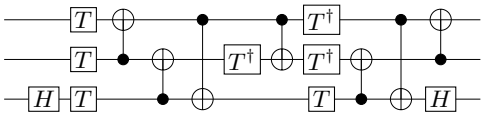


Figure 1: A quantum circuit

The quantum state of a single qubit is denoted by the vector $\binom{\alpha}{\beta}$. The state of a quantum system with $n > 1$ qubits is given by the tensor product of the respective state spaces and can be represented as a normalized 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 *quantum operations*. A quantum operation over n qubits can be represented by a *unitary matrix*, i.e. a $2^n \times 2^n$ matrix $\mathbf{U} = [u_{i,j}]_{2^n \times 2^n}$ with

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

Every quantum operation is reversible since the matrix that defines 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 $|\alpha|^2$) or a 1 (with probability $|\beta|^2$) results. The state of the qubit is destroyed by the act of measuring it.

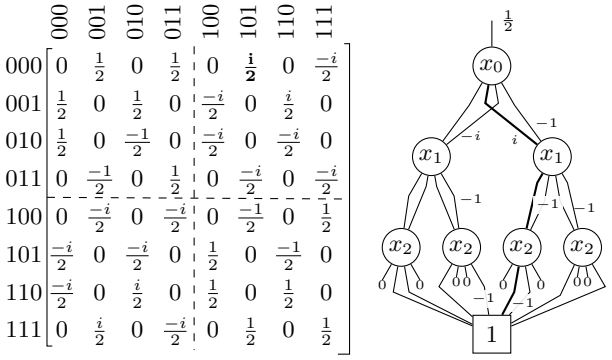
EXAMPLE 1. *Consider the quantum operation H defined by the unitary matrix* $\mathbf{H} = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$*, which is known as the Hadamard operation [28]. Applying H to the input state* $|x\rangle = \binom{1}{0}$*, i.e. computing* $\mathbf{H} \times |x\rangle$*, yields a new quantum state* $|x'\rangle = \frac{1}{\sqrt{2}}\binom{1}{1}$*. In* $|x'\rangle$*,* $\alpha = \beta = \frac{1}{\sqrt{2}}$*. Measuring this qubit would either lead to a Boolean 0 or Boolean 1, each with probability* $|\frac{1}{\sqrt{2}}|^2 = 0.5$*. This computation represents one of the simplest quantum computers – a single-qubit random number generator.*

Complex quantum operations are usually realized by a *quantum circuit*, which executes a series of elementary quantum operations using quantum *gates*. Such a composition of gates can be expressed by a direct matrix multiplication of the corresponding gate matrices. Alternatively, this process can be viewed as the implementation of a quantum algorithm in which a series of low-level quantum operations or quantum computational instructions is represented by a sequence of individual transformation (i.e. gate) matrices.

EXAMPLE 2. *Consider the 3-qubit quantum circuit shown in Fig. 1. It realizes a 2-controlled NOT operation known as the* Toffoli gate*. More precisely, the basis states of the third qubit are swapped if and only if the first and second qubits are in the* $|1\rangle$*-state. Conventionally, horizontal lines represent qubits. Operations* $\boxed{H}$ *(as in Example 1),* $\boxed{T}$ *with* $\mathbf{T} = \left(\begin{smallmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{smallmatrix}\right)$*,* $\bullet\!\!-\!\!\oplus$ *(CNOT), etc. are applied successively from left to right.*

2.2 Design Challenges

The most obvious challenge for CAD of quantum circuits is that computations are not necessarily performed on Boolean values anymore. Instead, qubits may assume an infinite number of superposition states which additionally can even entangled to each other. As a consequence, many established function representation such as Boolean algebra, binary decision diagrams, etc. cannot be applied anymore and have to be revised. In the past, corresponding languages [11, 33] or dedicated decision diagrams [47, 40, 25, 32] have been introduced for this purpose.



(a) Matrix (b) QMDD

Figure 2: Matrix and QMDD of a 3-qubit quantum circuit

In particular, *Quantum Multiple-valued Decision Diagrams (QMDDs, [25, 32])* are suitable for the efficient representation and manipulation of quantum gates and circuits. The fundamental idea is a recursive partitioning of the respective transformation matrix and the use of edge and vertex weights to represent various complex-valued matrix entries. More precisely, a transformation matrix of dimension $2^n \times 2^n$ is successively partitioned into four sub-matrices of dimension $2^{n-1} \times 2^{n-1}$. This partitioning is represented by a directed acyclic graph – the QMDD. The following example illustrates main aspects.

EXAMPLE 3. *Fig. 2a shows a transformation matrix for which a QMDD as shown in Fig. 2b has been built. Starting with a single* terminal vertex $\boxed{1}$ *that represents the lowest partitioning level, i.e. single matrix entries, the next upper level of* 2×2 *matrices is represented by vertices labeled* x_2*. For each entry, there is an outgoing edge to the terminal vertex with an* edge weight *corresponding to the respective complex value. For simplicity, we omit edge weights equal to 1 and indicate edges with a weight of 0 by stubs. The vertices are* normalized *by dividing the weights of all outgoing edges by a normalization factor (here: such that the "leftmost" edge with a non-zero weight has weight 1). This factor is propagated to referencing edges, e.g. the factor* $\frac{1}{2}$ *is propagated upwards from the* x_2*-level to the* x_0*-level in Fig. 2b. By this, structurally equivalent sub-matrices are compressed to a shared vertex (highlighted in grey in Fig. 2a and 2b, respectively). This procedure is repeated for each level until a single vertex labeled by* x_0 *is created for the top level. This vertex is called the* root vertex*. Finally, a possible normalization factor of this vertex is assigned to the weight of the* root edge *which points to the root vertex, but has no source.*

To obtain the value of a particular matrix entry, one has to follow the corresponding path from the root to the terminal vertex and multiply all edge weights on this path. For example, the matrix entry $\frac{i}{2}$ *from the top right sub-matrix of Fig. 2a (highlighted bold) can be determined as the product of the weights on the highlighted path of the QMDD in Fig. 2b.*

It can be shown that normalization as described above enables canonical QMDD representations. At the same time, the hierarchical structure allows for sharing as well as for a recursive application of operations – making QMDDs a proper means for compact representation as well as manipulation of quantum functionality. This has already been exploited in CAD for many design tasks including synthesis [42, 30], verification [48, 31], and simulation [10]. However, despite these initial results, much potential is still left to be exploited. This becomes particularly evident when

considering e.g. how binary decision diagrams have been utilized in numerous conventional design tasks. Compared to that, the exploitation of QMDDs (or similar function representations) are just at the beginning – many potential and open questions with respect to practical application but also theoretical understanding, respectively, remain to be explored.

Beyond that, CAD for quantum computation has to deal with the following further challenges:

- Automatic synthesis of quantum logic, i.e. how to efficiently determine a quantum circuit for a given quantum functionality, remains one of the most important topics. Similar to the representation issue from above, this becomes a crucial task because of the existence of non-Boolean values. Existing solutions (such as e.g. [39, 35]) often rely on enumeration and, hence, are not capable to automatically realize complex quantum functionality.
- For Boolean components to be used in a quantum algorithm (e.g. the database in the search algorithm of [12] or the modular exponentiation used in the factorization algorithm [41]), synthesis approaches for so-called *reversible circuits* can be utilized. In this domain, numerous solutions have been proposed in the recent past (see e.g. [8, 36] for overviews). Once a reversible circuit has been determined, it can easily be mapped into a corresponding quantum circuit. To this end, various different libraries such as NCV [2, 26, 52], NCV-$|v1\rangle$ [37], Clifford+T [1], or dedicated physical machine descriptions [21, 29] have been considered. However, what library eventually will best abstract from the physical requirements remains an open question.
- Once a circuit is realized, a technology mapping from the logic model to the physical model has to be conducted. Since physical realizations are currently subject to constant changes (depending on the recent accomplishments), this is still a very volatile task. Nevertheless, certain physical constraints are currently considered. As one recent prominent example, researchers investigated solutions to make a quantum circuit nearest neighbor-compliant (see e.g. [51, 13, 23]).

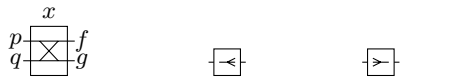
3. OPTICAL CIRCUITS

Silicon-based integrated optics (also known as *silicon photonics*), received attention particularly in communication systems [16], signal processing applications [53], or for optical interconnects as well as optical functional on-chip units [15]. While conventional solutions often require a back and forth conversion from the optical to the electrical signals e.g. at every interconnect interface, optical circuits employ both electronical but also optical signals and, hence, avoid this overhead. As a consequence, CAD has to deal with a new type of signal – an optical one.

3.1 Background

In optical circuits, Boolean functions can be realized by optical switching elements, called as *crossbar gates*, which route the optical signals between two parallel paths. The inputs of both paths can either be sourced by light (representing the logical value "1") or darkness (representing the logical value "0"). Furthermore, the routing of both paths may be switched depending on the value of an electrical signal. The output of each optical signal can be read using optical receivers. Logically, this leads to a gate realizing the function $\mathbb{B}^3 \to \mathbb{B}^2$ composed of two optical inputs p and q, one electrical input x, and two optical outputs f and g defined by

$$\overline{x} \Rightarrow (p \equiv f) \wedge (q \equiv g) \qquad \text{and} \qquad x \Rightarrow (p \equiv g) \wedge (q \equiv f).$$



(a) Crossbar gate (b) Splitter (c) Combiner

Figure 3: Optical gates

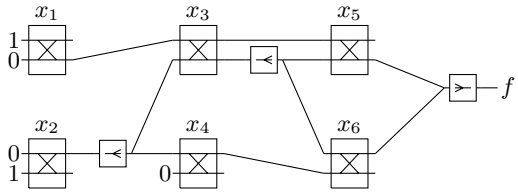


Figure 4: Optical circuit

Fig. 3a provides the graphical representation of a crossbar gate.

In addition to crossbar gates, splitters and combiners are also utilized as optical logic elements in order to realize logic functions. A *splitter* divides an optical signal into two signals – each with only half of the incoming signal power. In contrast, a *combiner* merges two optical signals into a single one and, by this, inherently realizes the OR-function. A splitter (combiner) may have more than two outputs (inputs). Then, in case of a splitter, the strength of the signal is divided by the number of outputs. Fig. 3b and Fig. 3c provide the graphical representation of both elements.

Using these logic elements, a technology library is formed that allows to realize arbitrary Boolean functions. As an example, Fig. 4 shows an optical circuit composed of four crossbar gates, one splitter, and one combiner.

3.2 Design Challenges

The distinction between electrical and optical circuit signals significantly affects the design of optical circuits and, hence, also the corresponding CAD methods. In the following, this is illustrated by considering the synthesis of a Boolean function provided as *Sum of Products* (SOP), i.e. in terms of a disjunction (OR) of conjunctions (AND) of literals (the conjunctions are also called *products*).

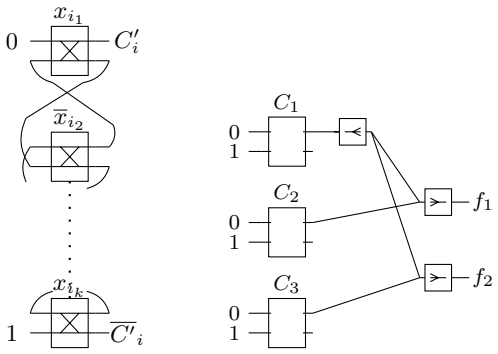
In conventional logic, an SOP is easily realized by simply applying the corresponding OR and AND gates. Considering optical circuits, this has to be mapped to the corresponding gate library presented above. While this is, in principle, solvable in a straight-forward fashion, it introduces a problem concerning the strength of certain signals in the circuit. An example illustrates the issue.

EXAMPLE 4. *Consider the Boolean functions f_1 and f_2 which are represented by the following SOP:*

	x_1	x_2	x_3	f_1	f_2
C_1	1	–	1	1	1
C_2	0	1	–	1	0
C_3	1	–	0	0	1

Here, the column on the left-hand side shows the respective products, where a "1" on the i^{th} position denotes a positive literal (i.e. x_i) and a "0" denotes a negative literal (i.e. $\overline{x}_i$), respectively. A "–" denotes that the respective variable is not included in the product. The right-hand side gives the respective primary output patterns to which the OR is applied.

All products such as w.l.o.g. $C_i' = x_{i_1}\overline{x}_{i_2}x_{i_3}\cdots x_{i_k}$ can be realized as sketched in Fig. 5a, i.e. for each literal $x_{i_j} \in C_i$, a crossbar gate is added. The respective inputs/outputs are connected in a fashion so that the input value 1 at the bottom of the circuit is passed through all gates to the output C_i iff all literals are indeed assigned 1 (for positive literals) or 0 (for negative literals). Afterwards, these blocks have to



(a) Mixed literals (b) Result of SOP-based synthesis

Figure 5: SOP-based synthesis

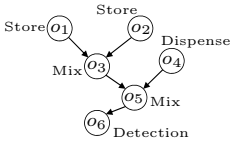
be ORed. This can easily be conducted using combiners as shown in Fig. 5b (where C_1, C_2, C_3 represent the respective blocks realizing the products).

However, for SOPs as considered in the example, products might be required more than once (in the example, this holds for the product C_1 which belongs to both outputs). Then, a splitter is added to make the intermediate results of the building blocks available to all functions relying on it (as shown in Fig. 5b). These splitters constitute a serious obstacle in the design of optical circuits, because they cause a considerable decrease in the optical signal. More precisely, each splitter with n outputs produces an output signal of strength $1/n$. Overall, this leads to a circuit's worst case fraction which can be determined by following all paths from the primary inputs to the primary outputs and multiplying all signal strengths produced by splitters visited on the current path[1]. Obviously, an important objective for CAD methods is to realize circuits with avoid too tiny fractions of signal strength.

Existing solutions approach this problem from different angles:

- Initial approaches [5] applied so-called virtual gates. They completely realize the desired function by combining crossbar gates eventually resulting in modules (called virtual gates). However, how to compose several of these virtual gates so that, eventually, the desired function results remains a hard problem. Here, particularly the fact that electrical and optical signals cannot easily been converted is a problem (to this end, an opto-electrical interface would be required which, however, is considered as expensive as well as slow). Hence, how to deal with different types of signals remains an important problem to tackle.
- Other approaches such as presented in [7] solve this problem by introducing redundancy: Instead of applying a splitter, the respective functionality is simply re-computed. This however significantly increases the number of requires gates and, hence, the costs of the resulting circuit.
- Finally, solutions based on efficient data-structures such as decision diagrams have been considered. Here, complementary solutions were introduced thus far: Initial approaches in [5, 38] (again, with the problem of requiring splitters) and an improved approach in [50] (which deals with this problem at the expense of additionally required gates).

[1]Note that this is just an approximation that may not correspond perfectly to the physical implementation, but gives a rough idea of the resulting signal strength.



Resource	Area	Time
Mixer	4	7
Mixer	3	4
LED	1	10
Storage	1	N/A

Max. Area: 5x5 array
Max. Completion Time:
50 seconds

(a) Sequencing graph (b) Design library (c) Design specifications

Figure 6: Given design task

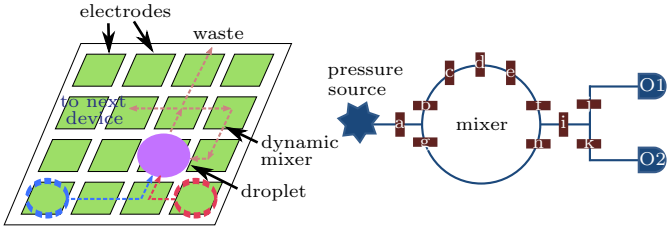
4. MICROFLUIDIC BIOCHIPS

Microfluidic biochips are another representative emerging technology that has attracted much research attention. The purpose of developing such chips is to miniaturize biochemical assays to nanoliter level, so that both chip size and cost can be reduced. In addition, the minuscule volumes of fluid samples reduce reaction time and save expensive reagents. Research on biochips is twofold. On the one hand, researchers in the microfluidics community are working on physical-level technologies to introduce new architectures [24, 9]. On the other hand, design tools are evolving from manual design to a computer-aided design flow [4, 43, 3]. This trend of design automation for microfluidic biochips is driven by the increasing integration of biochips. For example, already in 2008, a biochip array with 25K valves was manufactured [34]. To use such vast resources in a chip, it is natural that researchers in the design automation community adapt the design flow for integrated circuits to address emerging problems in biochemical processing.

4.1 Motivation & Background

Microfluidic biochips are used to execute biochemical assays. For example, in hospitals pathological samples from patients are tested with various reagents to identify diseases. Typical operations include mixing, heating, and detection. An assay is composed of many operations and their execution order is defined by a *sequencing graph*. An example of a sequencing graph is illustrated in Fig. 6a, where nodes represent operations and edges represent the execution order of these operations. The operations in the sequencing graph are executed by specific devices. For a given function such as mixing, multiple devices with different performances are provided in a library. It is the task of the design flow to choose the most suitable devices which execute the assay and meet the given specifications. This is very similar to high-level synthesis for integrated circuits, with steps such as scheduling and binding. The difference is that there is no explicit logic-level design, because the functions of the operations are not described using Boolean logic. Consequently, the results from scheduling and binding are mapped to underlying technologies directly in the physical design step to determine the final chip layout.

In recent years, two biochip technologies have been the focus of EDA research. The first type, digital biochips, uses electrowetting forces to drive tiny droplets on an electrode array (illustrated in Fig. 7a). The droplet is moved when a voltage difference between two neighboring electrodes exists. By applying a voltage pattern to different electrodes, droplets can be moved along given paths for transportation. If two droplets merge at a given location and thereafter are moved on a circular path, a dynamic mixer is formed. For this technology, the design flow should determine the transportation paths and the locations of dynamic mixers as well as dedicated devices such as heaters and detectors. The second type of biochips are the continuous-flow-based biochips, where transportation channels are etched on a substrate and microvalves are used to control the transportation of fluid segments. In flow-based biochips, mixers are also dedicated, e.g. as illustrated in Fig. 7b [14]. After two fluid segments enter the mixer, the three valves at the top of the mixer switch in a given on-off pattern resulting in a circular flow to mix the two reaction samples.



(a) Digital biochip (b) Flow-based biochip

Figure 7: Digital and flow-based biochips

4.2 Design Challenges

High-level synthesis of biochips can benefit from many existing EDA algorithms, but the underlying technologies require a significant adaptation of these algorithms. Compared to electrical signals, the physical volumes of fluid samples transported along channels make biochip design more complex. First, the volumes of fluid segments/droplets should be managed during transportation and execution. Second, transportation channels and virtual devices should be cleaned/washed after usage to avoid contamination.

EXAMPLE 5. *Fig. 7a is a snapshot of executing the assay in Fig. 6a. The droplets from storage (o_1) and (o_2) are transported to the location of the dynamic mixer for the mixing operation (o_3). After this operation, the paths $o_1 \rightarrow o_3$ and $o_2 \rightarrow o_3$ as well as the location occupied by the mixer should be cleaned before they are used again by other operations. To clean these paths and locations, special washing droplets should be sent to traverse all the used electrodes. This implicit task should be considered by the synthesis tool during high-level synthesis to reduce the effort of washing, e.g. by optimizing washing path arrangement.*

In addition, this example demonstrates the volume problem in executing operations. After executing the mixing operation, the mixed result has a volume equal to two times of the normal volume of a droplet. If only one normal droplet is needed by the next mixing operation (o_5), the other half result should be discarded to a waste port, requiring a new transportation path, which, for the sake of execution efficiency, should interfere with existing execution paths as little as possible.

The droplet volume management problem and path washing problem above are implicit in biochemical applications. They should be considered not only in physical design of biochips, but also during high-level synthesis, because binding operations to different devices may lead to different complexity in dealing with these two problems. In research on volume management, the sample preparation work [27] focuses on achieving a given concentration of the reaction result by carefully arranging chained mixing ratios. In normal assays, however, the volume management problem has often been ignored. On the washing problem, existing solutions such as [55] consider this task together with physical design. Generally speaking, volume management and washing should be incorporated into all steps from scheduling to physical design in the design flow for an overall optimization. (to this end, one-pass design schemes such as proposed in [18, 49] may provide a good basis). This incorporation, however, might lead to a scalability problem, so that a good tradeoff between optimality and scalability is required.

Besides the challenges above, further problems should be considered in developing a design flow for microfluidic biochips. They include:

- Resource usage optimization of biochip design. For example, the number of electrodes or valves/channels should be minimized to reduce manufacturing cost. In addition, low-level control layers should also be considered. In digital biochips, the number of independent voltage controllers should be minimized [22, 56, 17]. In flow-based biochips, the mechanism of pressure control of valves should be optimized such as in [9]. In addition, the access efficiency of the dedicated storage should also be addressed [45].

- Test and diagnosis of biochips after manufacturing. Due to the limitation of the manufacturing process, there might be defects in manufactured chips. These chips should be identified, and, if possible, reconfigured to guarantee the proper functionality of chips. The methods [14, 54] focus on this topic using logic gates formed on digital biochips or ATPG-based vector generation for flow-based biochips.

- Design diversity in biochips. There are many different lower-level fluid driving mechanisms of biochips from the microfluidic community. Besides digital biochips (electrowetting-driven) and flow-based biochips (valve-driven), there are also dot arrays [20], paper-based biochips [19], fully programmable valve arrays (FPVAs, [9, 44]), and Networked LoCs (NLoCs, [6]). All these new architectures have different underlying driving models and existing methods still deal with them individually with dedicated flows. To maintain both, this diversity at the manufacturing level and the efficiency at the design level, a method that can isolate/unite the low-level design diversity will benefit the research on biochips significantly.

5. CONCLUSIONS

Emerging technologies are attracting intensive research nowadays, because they either represent the potential successors of the semiconductor industry which is based on Boolean logic and silicon manufacturing (e.g. quantum circuits), or they expand the capacity of the existing design and manufacturing model (e.g. photonics). Furthermore, they can be fields that traditionally belonged to other research communities, to which the accumulated knowledge of the EDA community can make a great contribution (e.g. microfluidic biochips). Due to the underlying differences from the fundamental logic representations to the physical implementations of transportation media, these technologies pose massive challenges but also provide great chances to the EDA research community. By exploring new directions and expanding new scopes of design automation beyond its traditional definition, EDA will evolve into a more general interdisciplinary research that bridges various scientific fields and improves their design quality with success-proven work flows.

6. ACKNOWLEDGMENTS

The authors would like to thank all researchers and collaborators which, in the past years, worked with us on the development of the approaches which have been reviewed in this paper. This work has partially been supported by the European Union through the COST Action IC1405 and the IGSSE Project FLUIDA of the Technical University of Munich (TUM).

7. REFERENCES

[1] M. Amy, D. Maslov, M. Mosca, and M. Roetteler. A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. *IEEE Trans. on CAD*, 32(6):818–830, 2013.

[2] A. Barenco, C. H. Bennett, R. Cleve, D. DiVinchenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter. Elementary gates for quantum computation. *The American Physical Society*, 52:3457–3467, 1995.

[3] K. Chakrabarty, R. B. Fair, and J. Zeng. Design tools for digital microfluidic biochips: Toward functional diversification and more than Moore. *IEEE Trans. on CAD*, 29(7):1001–1017, 2010.

[4] K. Chakrabarty and J. Zeng. Design automation for microfluidics-based biochips. *J. Emerg. Technol. Comput. Syst.*, 1(3):186–223, 2005.

[5] C. Condrat, P. Kalla, and S. Blair. Logic Synthesis for Integrated Optics. In *Great lakes symposium on VLSI*, pages 13–18. ACM, 2011.

[6] E. De Leo, L. Galluccio, A. Lombardo, and G. Morabito. Networked labs-on-a-chip (NLoC): Introducing networking technologies in microfluidic systems. *Nano Communication Networks*, 3(4):217–228, 2012.

[7] A. Deb, R. Wille, O. Keszöcze, S. Hillmich, and R. Drechsler. Gates vs. splitters: Contradictory optimization objectives in the synthesis of optical circuits. *J. Emerg. Technol. Comput. Syst.*, 13(1):11, 2016.

[8] R. Drechsler and R. Wille. From truth tables to programming languages: Progress in the design of reversible circuits. In *Int'l Symp. on Multi-Valued Logic*, pages 78–85, 2011.

[9] L. M. Fidalgo and S. J. Maerkl. A software-programmable microfluidic device for automated biology. *Lab on a Chip*, 11:1612–1619, 2011.

[10] 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.

[11] A. S. Green, P. L. Lumsdaine, N. J. Ross, P. Selinger, and B. Valiron. Quipper: a scalable quantum programming language. *ACM SIGPLAN Notices*, 48(6):333–342, 2013.

[12] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Theory of computing*, pages 212–219, 1996.

[13] Y. Hirata, M. Nakanishi, S. Yamashita, and Y. Nakashima. An efficient method to convert arbitrary quantum circuits to ones on a linear nearest neighbor architecture. *Int'l Conf. on Quantum, Nano and Micro Technologies*, pages 26–33, 2009.

[14] K. Hu, F. Yu, T. Ho, and K. Chakrabarty. Testing of flow-based microfluidic biochips: Fault modeling, test generation, and experimental demonstration. *IEEE Trans. on CAD*, 33(10):1463–1475, 2014.

[15] Y. Kao and H. Chao. Design of a bufferless photonic clos network-on-chip architecture. *IEEE Transactions on Computers*, 63(3):764–776, 2014.

[16] G. Keiser. *Optical Fiber Communications*. McGraw-Hill, 1991.

[17] O. Keszocze, R. Wille, K. Chakrabarty, and R. Drechsler. A general and exact routing methodology for digital microfluidic biochips. In *Int'l Conf. on CAD*, pages 874–881, 2015.

[18] O. Keszocze, R. Wille, T.-Y. Ho, and R. Drechsler. Exact one-pass synthesis of digital microfluidic biochips. In *Design Automation Conf.*, 2014.

[19] H. Ko, J. Lee, Y. Kim, B. Lee, C.-H. Jung, J.-H. Choi, O.-S. Kwon, and K. Shin. Active digital microfluidic paper chips with inkjet-printed patterned electrodes. *Adv. Mater.*, 26(15):2335–2340, 2014.

[20] Z. Li, K. Y. Lai, P. Yu, T. Ho, K. Chakrabarty, and C. Lee. High-level synthesis for micro-electrode-dot-array digital microfluidic biochips. In *Design Automation Conf.*, pages 146:1–146:6, 2016.

[21] C. Lin, A. Chakrabarti, and N. K. Jha. Optimized quantum gate library for various physical machine descriptions. *Trans. on Very Large Scale Integration Systems*, 21(11):2055–2068, 2013.

[22] C. C.-Y. Lin and Y.-W. Chang. ILP-based pin-count aware design methodology for microfluidic biochips. *IEEE Trans. on CAD*, 29(9):1315–1327, 2010.

[23] A. Lye, R. Wille, and R. Drechsler. Determining the minimal number of SWAP gates for multi-dimensional nearest neighbor quantum circuits. In *ASP Design Automation Conf.*, pages 178–183, 2015.

[24] J. Melin and S. Quake. Microfluidic large-scale integration: the evolution of design rules for biological automation. *Annu. Rev. Biophys. Biomol. Struct.*, 36:213–231, 2007.

[25] 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*, page 6, 2006.

[26] D. M. Miller, R. Wille, and Z. Sasanian. Elementary quantum gate realizations for multiple-control Toffolli gates. In *Int'l Symp. on Multi-Valued Logic*, pages 288–293, 2011.

[27] D. Mitra, S. Roy, S. Bhattacharjee, K. Chakrabarty, and B. B. Bhattacharya. On-chip sample preparation for multiple targets using digital microfluidics. *IEEE Trans. on CAD*, 33(8):1131–1144, 2014.

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

[29] P. Niemann, S. Basu, A. Chakrabarti, N. K. Jha, and R. Wille. Synthesis of quantum circuits for dedicated physical machine descriptions. In *Conf. on Reversible Computation*, pages 248–264, 2015.

[30] 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.

[31] P. Niemann, R. Wille, and R. Drechsler. Equivalence checking in multi-level quantum systems. In *Conf. on Reversible Computation*, pages 201–215, 2014.

[32] 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.

[33] B. Ömer. Classical concepts in quantum programming. *International Journal of Theoretical Physics*, 44(7):943–955, 2005.

[34] J. M. Perkel. Microfluidics: Bringing new things to life science. *Science*, 322(5903):975–977, 2008.

[35] M. Saeedi, M. Arabzadeh, M. S. Zamani, and M. Sedighi. Block-based quantum-logic synthesis. *Quantum Info. Comput.*, 11(3):262–277, 2011.

[36] M. Saeedi and I. L. Markov. Synthesis and optimization of reversible circuits - a survey. *ACM Computing Surveys*, 2011.

[37] Z. Sasanian, R. Wille, and D. M. Miller. Realizing reversible circuits using a new class of quantum gates. In *Design Automation Conf.*, pages 36–41, 2012.

[38] E. Schönborn, K. Datta, R. Wille, I. Sengupta, H. Rahaman, and R. Drechsler. BDD-based synthesis for all-optical Mach-Zehnder interferometer circuits. In *VLSI Design*, pages 435–440, 2015.

[39] V. V. Shende, S. S. Bullock, and I. L. Markov. Synthesis of quantum-logic circuits. *IEEE Trans. on CAD*, 25(6):1000–1010, 2006.

[40] W. Shiou-An, L. Chin-Yung, K. Sy-Yen, et al. An XQDD-based verification method for quantum circuits. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 91(2):584–594, 2008.

[41] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. *Foundations of Computer Science*, pages 124–134, 1994.

[42] 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.

[43] F. Su, K. Chakrabarty, and R. B. Fair. Microfluidics-based biochips: Technology issues, implementation platforms, and design-automation challenges. *IEEE Trans. on CAD*, 25(2):211–223, 2006.

[44] T. Tseng, B. Li, T. Ho, and U. Schlichtmann. Reliability-aware synthesis for flow-based microfluidic biochips by dynamic-device mapping. In *Design Automation Conf.*, pages 141:1–141:6, 2015.

[45] T. Tseng, B. Li, U. Schlichtmann, and T. Ho. Storage and caching: Synthesis of flow-based microfluidic biochips. *IEEE Design & Test*, 32(6):69–75, 2015.

[46] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414:883, 2001.

[47] G. F. Viamontes, I. L. Markov, and J. P. Hayes. High-performance QuIDD-based simulation of quantum circuits. In *Design, Automation and Test in Europe*, pages 1354–1355, 2004.

[48] R. Wille, D. Große, D. M. Miller, and R. Drechsler. Equivalence checking of reversible circuits. In *Int'l Symp. on Multi-Valued Logic*, pages 324–330, 2009.

[49] R. Wille, O. Keszocze, R. Drechsler, T. Boehnisch, and A. Kroker. Scalable one-pass synthesis for digital microfluidic biochips. *IEEE Design & Test*, 32(6):41–50, 2015.

[50] R. Wille, O. Keszocze, C. Hopfmuller, and R. Drechsler. Reverse BDD-based synthesis for splitter-free optical circuits. In *ASP Design Automation Conf.*, pages 172–177, 2015.

[51] R. Wille, A. Lye, and R. Drechsler. Exact reordering of circuit lines for nearest neighbor quantum architectures. *IEEE Trans. on CAD*, 33(12), 2014.

[52] R. Wille, M. Soeken, C. Otterstedt, and R. Drechsler. Improving the mapping of reversible circuits to quantum circuits using multiple target lines. In *ASP Design Automation Conf.*, pages 85–92, 2013.

[53] A. Willner, S. Khaleghi, M. Chitgarha, and O. Yilmaz. All-optical signal processing. *Lightwave Technology, Journal of*, 32(4):660–680, 2014.

[54] T. Xu and K. Chakrabarty. Fault modeling and functional test methods for digital microfluidic biochips. *Trans. Biomed. Circuits and Systems*, 3(4):241–253, 2009.

[55] Y. Zhao and K. Chakrabarty. Synchronization of washing operations with droplet routing for cross-contamination avoidance in digital microfluidic biochips. In *Design Automation Conf.*, pages 635–640, 2010.

[56] Y. Zhao and K. Chakrabarty. Simultaneous optimization of droplet routing and control-pin mapping to electrodes in digital microfluidic biochips. *IEEE Trans. on CAD*, 31(2):242–254, 2012.