

# Improved Mapping of Quantum Circuits to IBM QX Architectures

Abhoy Kole, *Member, IEEE*, Stefan Hillmich, *Student Member, IEEE*, Kamalika Datta, *Member, IEEE*, Robert Wille, *Senior Member, IEEE*, and Indranil Sengupta, *Senior Member, IEEE*

**Abstract**—Quantum computers are becoming a reality today due to the rapid progress made by researchers in the last years. In the process of building quantum computers, IBM has developed several versions—starting from 5-qubit architectures like IBM QX2 and IBM QX4 to larger 16- or 20-qubit architectures. These architectures support arbitrary rotations of a single qubit and a controlled negation (CNOT) involving two qubits. The two qubit operations come with added coupling-map restrictions that only allow specific physical qubits to be the control and target qubits of the operation. In order to execute a quantum circuit on the IBM QX architecture, CNOT gates must satisfy the so-called coupling constraints of the architecture.

Previous works addressed this issue with the objective of reducing the number of gates and the circuit depth. However, in this work we show that further improvements are possible. To this end, we present a general approach for further improving the number of gate operations and depth of the mapped circuit. The proposed approach encompasses the selection of physical qubits, determining initial and local permutations efficiently to obtain the final circuit mapped to the given IBM QX architecture. Through experiments improvements are observed over existing methods in terms of the number of gates and circuit depth.

**Index Terms**—IBM QX Architecture, Coupling-map, Quantum Circuit, Clifford+T library

## I. INTRODUCTION

Quantum computing has received much attention in the last few decades due to its superiority over classical computing for solving problems like factoring integers [1], searching objects in unsorted databases [2], and efficiently simulating processes in quantum chemistry [3].

The supremacy in computation comes from the exploitation of quantum mechanics like linear evolution and entanglement. A computing system in this domain comprises a number of two-state subsystems that are called qubits. The state of an individual qubit in a 2-dimensional Hilbert space can either be one of the basis states  $|0\rangle$  and  $|1\rangle$  or a linear combination

A. Kole is with the Rajendra Mishra School of Engineering Entrepreneurship, Indian Institute of Technology, Kharagpur 721302, India.  
E-mail: abhoy.kole@iitkgp.ac.in

S. Hillmich is with the Institute for Integrated Circuits, Johannes Kepler University Linz, A-4040 Linz, Austria.  
E-mail: stefan.hillmich@jku.at

K. Datta is with the Nanyang Technological University, Singapore.  
E-mail: kdatta.iitkgp@gmail.com

R. Wille is with the Institute for Integrated Circuits, Johannes Kepler University Linz, A-4040 Linz, Austria and the Cyber-Physical Systems Dept., DFKI Bremen GmbH, 28359 Bremen, Germany.  
E-mail: robert.wille@jku.at

I. Sengupta is with the Department of Computer Science & Engineering, Indian Institute of Technology, Kharagpur 721302, India.  
E-mail: isg@iitkgp.ac.in

of them, i.e.  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$  where  $\alpha$  and  $\beta$  are complex numbers and  $|\alpha|^2 + |\beta|^2 = 1$  [4]. Different technologies, such as trapped ions and superconducting qubits, provide retainable quantum states and mechanisms to interact with them [5] that are necessary requirements for the physical realization of a quantum system.

In March 2017, IBM introduced their first superconducting quantum architecture, IBM QX2 with 5 qubits [6] and made it available in the cloud for broad access. IBM also provides cloud access to their 16 qubit architecture IBM QX3. The improved version of these two architectures, IBM QX4 and IBM QX5, were made available in the cloud later on.

In order to run a quantum algorithm on one of the IBM QX architectures, the algorithm must be expressed using supported elementary quantum operations. There also exist architectural constraints on applying the elementary operations on physical qubits. The quantum circuits must be mapped to these physical architectures adhering to the constraints. In all but the trivial cases, it is not possible to physically map an entire quantum circuit at once satisfying these physical constraints. The states of physical qubits must be interchanged in performing subsequent quantum operations to satisfy such constraints. In order to do this, additional gates realizing *SWAP* operations are inserted before quantum gates operating on qubits not satisfying these physical constraints. Since quantum operations are not error free, the operational reliability of quantum circuits is compromised due to the insertion of these additional gates. Thus, it is necessary to minimize the number of *SWAP* operations.

Mapping an arbitrary quantum function to the IBM QX architecture is carried out in two steps. Initially the functionality is expressed in terms of elementary operations. Several methods exist to carry out this task [7]–[12]. Next, elementary operations on logical qubits are mapped to an IBM QX architecture satisfying the physical constraints. Early works for satisfying nearest neighbor constraints on 1-dimensional, 2-dimensional or higher dimensional physical layouts are less restricted [13]–[23] compared with physical constraints for the IBM QX architecture. There exist few works on physical mapping of these quantum circuits to IBM QX architectures [24]–[27]. The authors in [25] have addressed the physical mapping issue specifically for IBM QX2 and QX4 architectures whereas the authors in [26] have presented a generic approach for mapping to an arbitrary IBM QX architecture. In [24], the authors have suggested an architecture specific mapping scheme for quantum circuits containing  $SU(4)$  gates. Although these approaches have produced better compilation results for

quantum circuits, there is always scope for improvement. In finding an optimal solution, there are several issues that need to be addressed, like physical qubits selection, mapping logical qubits, insertion of SWAP operations and finally optimizing the mapped netlist using certain gate reduction rules. In fact, finding the optimal mapping for quantum circuits is an NP-complete problem [28], [29]. An investigation towards this remains as an active research problem (see, e.g. [30]).

In this regard, we propose a generic approach for mapping quantum circuits to IBM QX architectures satisfying physical constraints. Initially, for a given circuit, physical qubits are identified and an evolutionary algorithm is used to find a close to optimal initial ordering of the qubits. For the evolutionary algorithm, a new cost metric based on the physical mapping is introduced as the fitness function. With this cost metric we propose an approach for insertion of gates to perform *SWAP* operations. Although the run-time of our proposed approach increases due to incorporation of the evolutionary algorithm for initial mapping, after optimizations we observe resulting circuits with reduced gate count and depth over the approach proposed in [26], with an increasing improvement for larger circuits.

The remainder of this paper is organized as follows. In Section II we provide a review of quantum circuits and IBM QX architectures. In Section III we discuss the proposed approach for mapping a quantum circuit to the IBM QX architecture. Experimental results and discussions are presented in Section IV. Finally, in Section V, we provide some concluding remarks.

## II. PRELIMINARIES

### A. Quantum Operations

Quantum computers operate on *quantum bits* or *qubits*. A qubit is the basic computational unit in the quantum domain. Like classical bits that can only assume the basis states 0 or 1, a qubit can represent a two-level system with basis states  $|0\rangle$  and  $|1\rangle$  (written in Dirac notation). However, a qubit, unlike a classical bit, can also be in a superposition of these basis states, i.e.  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$  where  $\alpha$  and  $\beta$  are complex numbers and  $|\alpha|^2 + |\beta|^2 = 1$ . Any general qubit state like  $|\psi\rangle$  also represents a vector  $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$  in a 2-dimensional Hilbert space. The state of  $n$  such qubits is the tensor product of individual states, i.e.  $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}^{\otimes n}$ . This allows parallel quantum operations over all  $2^n$  possible basis states.

Unitary operations are performed on one or more qubits to evolve the qubits states. An  $n$ -qubit operator is represented by a  $2^n \times 2^n$  unitary matrix. The elementary operators can be represented by the following matrices:

$$\begin{aligned} X &= NOT = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} & S &= \sqrt{Z} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \\ Y &= \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} & T &= \sqrt[4]{Z} = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1+i}{\sqrt{2}} \end{pmatrix} \\ Z &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} & H &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \end{aligned}$$

Here in the first column  $X$ ,  $Y$  and  $Z$  are the Pauli rotation operators and the second column represents phase  $S$ , square-root of phase  $T$  and Hadamard  $H$  operators.

Together, they form the Clifford+T library [11]. For completeness, two-qubit CNOT gate is also included in this library. The  $4 \times 4$  matrix for  $CNOT(q_i, q_j): |q_i, q_j\rangle \mapsto |q_i, q_i \oplus q_j\rangle$  operation is represented as a sum of tensor products of  $2 \times 2$  matrices,  $|0\rangle\langle 0|_{q_i} \otimes I_{q_j} + |1\rangle\langle 1|_{q_i} \otimes X_{q_j}$  where  $I$  is the identity matrix and  $\langle k|$  is the dual of  $|k\rangle$ .

### B. Quantum Circuits

In this work quantum circuits are represented by their corresponding circuit diagrams. There are quantum languages like Scaffold [31], Quipper [32], and OpenQASM [33] which can also be used to describe quantum circuits. In a circuit diagram, qubits are shown in space (y-axis) and unitary operations that are performed on the qubits are shown as applications of these operations in the axis of time (x-axis).

**Example 1.** Fig. 1 shows a quantum circuit composed of 7 unitary operators and 3 qubits. The elementary gates  $H$ ,  $S$  and  $T^\dagger$  are represented by rectangular boxes and the boxes are labeled with corresponding operator names. The  $CNOT(q_i, q_j)$  gates are represented by a  $\bullet$  on the control qubit  $q_i$  and  $\oplus$  on the target qubit  $q_j$ .

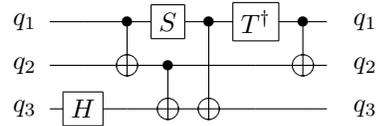


Fig. 1: An example quantum circuit

### C. IBM QX Architecture

The project IBM Q [6] has provided several architectures that are shown in Fig. 2. A framework named Qiskit [27] is also presented for describing, simulating and executing quantum circuits on a real quantum device that has been made available in the cloud. These real devices are comprised of 5 or 16 physical qubits connected through coplanar waveguide bus resonators [34]. Microwave pulses are applied to perform quantum operations on these qubits.

All of these architectures support the elementary operation  $U(\theta, \phi, \lambda)$  that can be expressed by unitary rotation operations, i.e.  $R_z(\phi)R_y(\theta)R_z(\lambda)$ , where  $\theta$  rotations about  $Z$ -axis and  $Y$ -axis are represented by  $R_z(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$  and  $R_y(\theta) = \begin{pmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix}$  respectively. Arbitrary elementary operations are performed on physical qubits by adjusting the parameters  $\theta$ ,  $\phi$  and  $\lambda$ .

Further, all of these architectures restrict the application of the two-qubit CNOT operation. A  $CNOT(Q_i, Q_j)$  operation on a physical qubit pair  $Q_i$  and  $Q_j$  can only be applied if a *coupling-map*  $Q_i \rightarrow Q_j$  exists between them where  $Q_i$  and  $Q_j$  act as control and target qubits, respectively. Fig. 2 shows the *coupling-map* for the four IBM QX architectures.

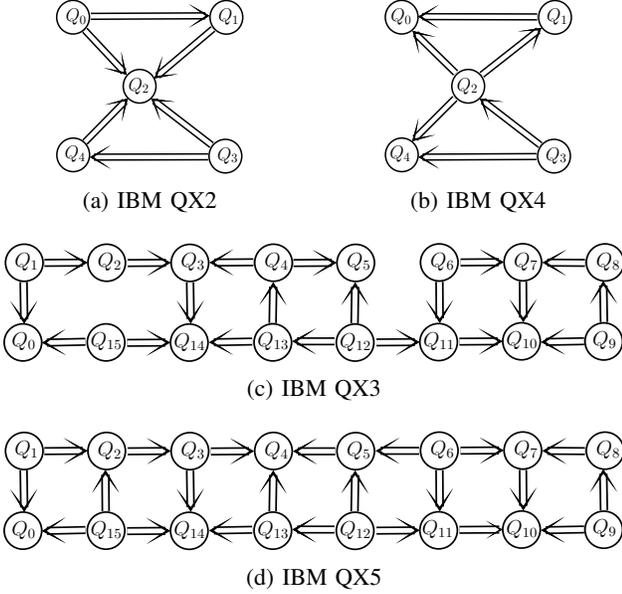


Fig. 2: Physical association of qubits in IBM QX architectures

### III. PHYSICAL QUBIT MAPPING

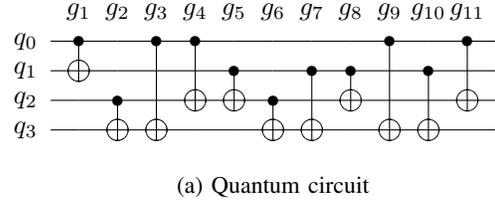
#### A. Physical Qubit Selection

To map a given circuit with  $n$  logical qubits  $(q_0, \dots, q_{n-1})$  to a given IBM QX architecture with  $m$  ( $n \leq m$ ) physical qubits  $(Q_0, \dots, Q_{m-1})$ , the physical qubits can be selected in  $\binom{m}{n}$  ways. Assuming the qubit association as a directed graph (see Fig. 2), the selected physical qubits for mapping with logical qubits must be a connected component. Initially, both logical qubits and physical qubits are ordered based on their degree of association. For the coupling-map  $Q_i \rightarrow Q_j$  and  $Q_i \rightarrow Q_k$ , the degree of association of  $Q_i$  is 2, i.e.  $\text{deg}(Q_i) = 2$ . Similarly for the logical qubit associations  $q_x \rightarrow q_y$  and  $q_x \rightarrow q_z$ ,  $\text{deg}(q_x)$  is 2. For a given quantum circuit, the degree of association of each logical qubit is the out-degree of the corresponding vertex in the incidence graph,  $IG = (V, E)$  where each vertex  $q_i \in V$  represents a logical qubit and each directed edge  $(q_i, q_j) \in E$  denotes a  $\text{CNOT}(q_i, q_j)$  operation. The following example illustrates the qubit ordering.

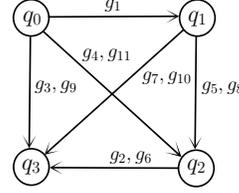
**Example 2.** Fig. 3a and 3b show a quantum circuit and the corresponding incidence graph, respectively. The qubit ordering derived from the incidence graph is shown in Fig. 3c.

For a given architecture physical qubits are also ordered in a similar way based on their degree of association obtained from the coupling-map, e.g. the out-degree of  $Q_2$  in IBM QX4 is 3 as can be verified from Fig. 2b.

Next, the shortest distances between physical qubits are computed to determine the number of SWAP operations required to satisfy the coupling-map constraints. For a coupling-map containing  $Q_i \rightarrow Q_j$ , the distance  $d_{ij}$  is set to 0, since the application of  $\text{CNOT}(Q_i, Q_j)$  requires no SWAP operation. If  $Q_i$  and  $Q_j$  are not adjacent but have a common adjacent qubit, the application of  $\text{CNOT}(Q_j, Q_i)$  requires 4 additional



(a) Quantum circuit



(b) Incidence graph

$q_x$	$\text{deg}(q_x)$
$q_0$	3
$q_1$	2
$q_2$	1
$q_3$	0

(c) Qubit ordering

Fig. 3: An example logical qubit ordering

Hadamard operations, and since a SWAP operation is realized using a network of 7 elementary operations, distance  $d_{ji}$  is set to  $\frac{4}{7} \approx 0.6$ . The following example illustrates the realization of a SWAP operation.

**Example 3.** Fig. 4a shows the SWAP operation, i.e.  $\text{SWAP}(q_x, q_y)$ . The operation can be realized by a network of three CNOT gates:

$$\text{CNOT}(q_x, q_y) \text{CNOT}(q_y, q_x) \text{CNOT}(q_x, q_y)$$

as shown in Fig. 4b. For the coupling-map shown in Fig. 4c and physical mapping of qubits  $(q_x, q_y) \rightarrow (Q_i, Q_j)$ , the second  $\text{CNOT}(q_y, q_x)$  operation can be realized using four Hadamard operations as shown in Fig. 4d.

$$\begin{array}{l} q_x : |\phi_1\rangle \\ q_y : |\phi_2\rangle \end{array} \begin{array}{c} \times \\ \times \end{array} \begin{array}{l} |\phi_2\rangle \\ |\phi_1\rangle \end{array}$$

(a) SWAP gate

$$\begin{array}{l} q_x : |\phi_1\rangle \\ q_y : |\phi_2\rangle \end{array} \begin{array}{c} \bullet \oplus \bullet \\ \oplus \bullet \oplus \end{array} \begin{array}{l} |\phi_2\rangle \\ |\phi_1\rangle \end{array}$$

(b) SWAP realization



(c) Coupling-map

$$\begin{array}{l} Q_i \leftarrow q_x \\ Q_j \leftarrow q_y \end{array} \begin{array}{c} \bullet \oplus \bullet \\ \oplus \bullet \oplus \end{array} \begin{array}{c} \boxed{H} \quad \boxed{H} \\ \boxed{H} \quad \boxed{H} \end{array} \begin{array}{c} \bullet \\ \oplus \end{array}$$

(d) SWAP realization

Fig. 4: Realizing SWAP operation on IBM QX architecture

For a given physical architecture, the distance  $d_{ij}$  between adjacent physical qubits  $Q_i$  and  $Q_j$  is assigned to 0 or 0.6, depending on their directions in the coupling-map. The distance  $d_{ij}$  for all non-adjacent physical qubit pairs  $Q_i \rightsquigarrow Q_j$  is initially set to  $\infty$  and further computed exploiting the *Floyd-Warshall* algorithm [35] for solving the *all-pairs-shortest-path* problem in the following way.

For a non-adjacent qubit pair  $Q_i \rightsquigarrow Q_j$  and an intermediate qubit  $Q_k$  if there exists a finite distance path between qubit pairs  $Q_i \rightsquigarrow Q_k$  and  $Q_k \rightsquigarrow Q_j$ , i.e.  $d_{ik} \neq \infty$  and  $d_{kj} \neq$

$\infty$  then the distance between  $Q_i$  and  $Q_j$  through  $Q_k$ ,  $d_{ij}^k$  is computed as follows:

$$d_{ij}^k = \begin{cases} d_{ki} + d_{jk} + 1.6; & \text{if } d_{ik} > d_{ki} \wedge d_{kj} > d_{jk} \\ \min\{d_{ik}, d_{ki}\} + \min\{d_{kj}, d_{jk}\} + 1; & \text{otherwise} \end{cases}$$

where both the distance expressions represent the normalized distance in terms of the number of SWAP operations and the expression  $d_{ki} + d_{jk} + 1.6$  is used in computing the distance when there exist only reverse coupling-map edges in the path  $Q_i \rightsquigarrow Q_k \rightsquigarrow Q_j$ ; otherwise the latter expression is used. Finally, the distance,  $d_{ij}$  between qubit pair  $Q_i \rightsquigarrow Q_j$  is assigned  $d_{ij}^k$ , if  $d_{ij} > d_{ij}^k$ . For an  $n$ -qubit physical layout this process continues considering physical qubit  $Q_k$  two times, initially for  $k = 0, 1, \dots, n-1$  and then for  $k = n-1, n-2, \dots, 0$  to find the shortest distance between each such non-adjacent pairs  $Q_i \rightsquigarrow Q_j$ , where  $i, j = 0, 1, \dots, n-1$  and  $i \neq j$ .

To obtain the shortest distances the algorithm is run twice for increasing and decreasing values of  $k$ . Although the runtime for computing the shortest distance between  $n$  physical qubits is  $\mathcal{O}(n^3)$ , the process is carried out only once for each physical architecture. The computation of the shortest distance is illustrated below.

**Example 4.** The initial distance between physical qubits and the final distance obtained by running the Floyd-Warshall algorithm [35] for IBM QX4 is shown in Table I. From the table, it can be verified that the application of  $CNOT(Q_1, Q_3)$  requires 1 SWAP operation and 4 Hadamard operations, whereas the application of  $CNOT(Q_3, Q_1)$  requires only 1 SWAP operation.

TABLE I:  $d_{ij}$  Measures for IBM QX4

	Initial					Final				
	$Q_0$	$Q_1$	$Q_2$	$Q_3$	$Q_4$	$Q_0$	$Q_1$	$Q_2$	$Q_3$	$Q_4$
$Q_0$	—	0.6	0.6	$\infty$	$\infty$	—	0.6	0.6	1.6	1
$Q_1$	0	—	0.6	$\infty$	$\infty$	0	—	0.6	1.6	1
$Q_2$	0	0	—	0.6	0	0	0	—	0.6	0
$Q_3$	$\infty$	$\infty$	0	—	0	1	1	0	—	0
$Q_4$	$\infty$	$\infty$	0.6	0.6	—	1	1	0.6	0.6	—

Initially, the logical qubit with the maximum degree of association is mapped to a physical qubit having the maximum degree of association. For subsequent mapping of the logical qubit with the next highest degree of association, an unmapped physical qubit which has the shortest distance from all the mapped physical qubits is selected. The process continues until all logical qubits are mapped. To obtain a mapping of physical qubits with evenly distributed degree of association, the distance between physical qubits  $Q_i$  and  $Q_j$   $d_{ij}$  is rounded to the nearest integer value. It can be verified that the average runtime of this algorithm is  $\mathcal{O}(n^2)$  in mapping  $n$  physical qubits. The mapping generated by running the proposed algorithm on various IBM QX architecture is illustrated by the following example.

**Example 5.** Fig. 5a and 5b show the selection of physical qubits for mapping the quantum circuit shown in Fig. 3a, to the IBM QX3 and QX5 architectures, respectively. The logical to physical qubit mappings are shown in Fig. 5c.

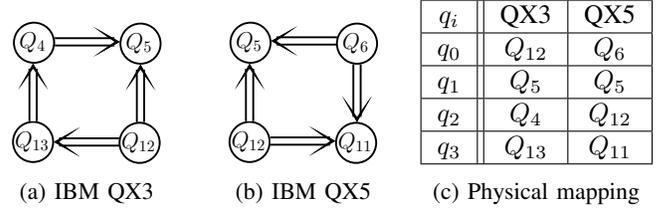


Fig. 5: Physical mapping of logical qubits

## B. Qubit Ordering

For a given quantum circuit, the physical mapping of logical qubits  $(q_0, q_1, \dots, q_{n-1}) \rightarrow (Q_i, Q_j, \dots, Q_k)$  obtained in the previous section may not satisfy the coupling-map constraints for all the CNOT gates present in the circuit. To satisfy the constraints SWAP operations are performed on a pair of qubits to exchange their states. For a coupling-map  $Q_i \rightarrow Q_j$ , the  $CNOT(Q_j, Q_i)$  operation is realized using 4 additional Hadamard operations.

For a physical mapping  $(q_0, \dots, q_{n-1}) \rightarrow (Q_i, \dots, Q_k)$ , if a CNOT gate at depth  $i$  does not satisfy the coupling-map constraints, the state of the qubits are swapped to obtain a new permutation  $\pi_i$ . Since the physical layout of IBM QX architectures is 2-dimensional, there exists more than one permutation that may satisfy the coupling-map constraint. In order to obtain a permutation that minimizes the SWAP requirement for subsequent depths  $(i+1, i+2, \dots)$ , a heuristic is used. Initially, all CNOT gate operations are prioritized based on their depth such that for a given circuit of depth  $n$ , the priority,  $p_i$  of the CNOT gate at depth  $i$  satisfies the following criteria

$$p_i > p_{i+1} + p_{i+2} + \dots + p_n \quad (1)$$

The priority values are selected in such a way that Eq. 1 is satisfied for all possible values of  $i$ .

**Lemma 1.** For any real number  $M \geq 2$ ,  $n > 1$  and  $0 \leq i \leq n-1$ , the priority assignment  $p_i = M^{n-i}$  satisfies Eq. 1.

*Proof.* For  $0 \leq i \leq n$ , the RHS of Eq. 1 is

$$\begin{aligned} p_{i+1} + p_{i+2} + \dots + p_n &= M^{n-(i+1)} + M^{n-(i+2)} + \dots + 1 \\ &= \frac{1}{M-1} (M^{n-i} - 1) \\ &< M^{n-i} \end{aligned}$$

for  $M \geq 2$ .  $\square$

The cost metric *coupling-cost* for a physical mapping of  $n$  logical qubits  $(q_0, q_1, \dots, q_{n-1}) \rightarrow (Q_i, Q_j, \dots, Q_k)$  is defined as

$$\text{coupling-cost} = \sum_{x,y \in \{1,2,\dots,n\}} d_{xy} w_{xy} \quad (2)$$

where for a physical mapping  $q_x \rightarrow Q_s$  and  $q_y \rightarrow Q_t$ ,  $d_{xy}$  is the distance between the physical qubit pair  $\{Q_s, Q_t\}$ , and  $w_{xy}$  is the weight of the corresponding edge  $(q_x, q_y)$  in

the incidence graph. Given a quantum circuit with  $n$  logical qubits  $\{q_0, q_1, \dots, q_{n-1}\}$  and depth  $m$ , the weight  $w_{xy}$  of the directed edge  $(q_x, q_y)$  in the incidence graph is defined as

$$w_{xy} = \sum_{k \in \{1, 2, \dots, m\}} p_k^{xy} \quad (3)$$

where  $p_k^{xy}$  is the priority of the  $\text{CNOT}(q_x, q_y)$  gate at depth  $k \leq m$ . The following example illustrates the computation of physical mapping cost.

**Example 6.** Fig. 6a shows the same quantum circuit of depth 8 (see Fig. 3a) where the priority  $p_i = M^{8-i}$  at each depth  $i$  is computed with  $M = 2$ . The corresponding weighted incidence graph for the circuit is shown in Fig. 6b. It can be verified that the weight of the edge  $(q_2, q_3)$  is the sum of the priorities of the  $\text{CNOT}$  gates  $g_2$  and  $g_6$  at depth 1 and 5, i.e.  $p_1 + p_5 = 128 + 8 = 136$ . The coupling-cost for the physical mapping  $(q_0, q_1, q_2, q_3) \rightarrow (Q_2, Q_0, Q_3, Q_4)$  to IBM QX3 is calculated as

$$0 \times 128 + 0 \times 136 + 0 \times 66 + 0.6 \times 33 + 1.6 \times 18 + 1 \times 5 = 53.6.$$

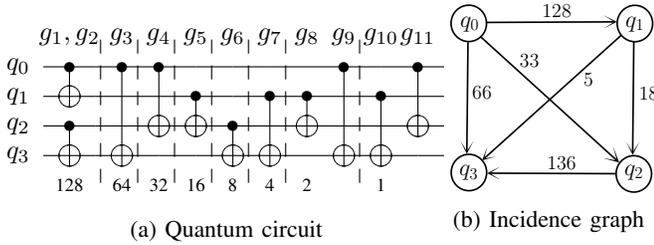


Fig. 6: Coupling-cost estimation for mapping logical qubits on IBM QX3

For a given physical mapping of  $n$  logical qubits  $(q_0, \dots, q_{n-1}) \rightarrow (Q_i, \dots, Q_k)$ , there are  $n!$  permutations, including the current permutation  $\pi_i$ . To minimize the number of SWAP gates, the selection of a specific permutation  $\pi_j$  and the insertion of SWAP gates to transform the current permutation  $\pi_i \rightarrow \pi_j$  is carried out in such a way that the coupling-cost of the physical mapping for subsequent  $\text{CNOT}$  gates is minimized. The following example illustrates this process.

**Example 7.** Fig. 7 shows the physical mapping  $(q_0, q_1, q_2, q_3) \rightarrow (Q_6, Q_5, Q_{12}, Q_{11})$  of a quantum circuit in IBM QX5 (see Fig. 2d). The first three  $\text{CNOT}$  gates  $g_1, g_2$  and  $g_3$  satisfy the coupling-map constraint. For the gate  $g_4$ ,  $\text{CNOT}(q_0, q_2)$ , there is no such coupling-map  $Q_6 \rightarrow Q_{12}$  in IBM QX5 architecture. All possible permutations considering the coupling-maps  $Q_6 \rightarrow Q_5 \leftarrow Q_{12}$  and  $Q_6 \rightarrow Q_{11} \leftarrow Q_{12}$  are explored and a permutation that minimizes the coupling-cost of the subsequent  $\text{CNOT}$  gates  $(g_5, \dots, g_{11})$  is selected for the physical mapping.

### C. Initial Permutation

Given a quantum circuit to be mapped to a specific architecture, the selection of physical qubits  $\{Q_i, \dots, Q_k\}$  and the

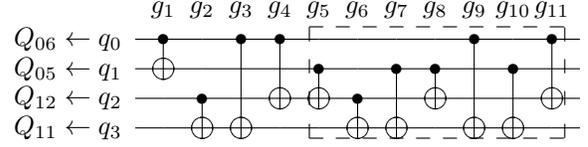


Fig. 7: An example logical qubit ordering

initial permutation  $\pi_0$  of logical qubits  $\{q_0, \dots, q_{n-1}\}$  have a significant impact in minimizing the number of gates as well as depth. It is necessary to estimate the  $\text{SWAP-cost}$  for each of the initial permutations,  $\pi_0^0, \pi_0^1, \dots, \pi_0^{k-1}$  to determine the permutation,  $\pi_0^i$ , ( $0 \leq i \leq k-1$ ) with the smallest  $\text{SWAP-cost}$ .

To estimate the  $\text{SWAP-cost}$  of a specific initial permutation  $\pi_0^i$  for a given circuit of depth  $m$ , the incidence graph for the circuit is traversed  $m$  times,  $d = m-1, m-2, \dots, 0$ , and setting the  $\text{SWAP-cost}$  to 0 at the beginning. In each iteration the edges  $(q_x, q_y)$  with weight  $w_{xy} > M^d$  are updated with  $w_{xy} = w_{xy} - M^d$ . For the current physical mapping  $(q_x, q_y) \rightarrow (Q_i, Q_j)$  if the distance  $d_{xy}$  is greater than 0, the  $\text{SWAP-cost}$  is updated by adding  $d_{xy}$  and the current permutation is transformed to  $\pi_d^i$  such that for this new permutation,  $\pi_d^i$ ,  $d_{xy} = 0$  and the coupling-cost is minimized. The following example illustrates this.

**Example 8.** The weights of the edges  $(q_0, q_1)$  and  $(q_2, q_3)$  of the incidence graph shown in Fig. 6b can be updated for  $d = 7$  to  $w_{01} = 128 - 2^7 = 0$  and  $w_{23} = 136 - 2^7 = 8$ , respectively, considering circuit depth 8 and  $M = 2$ . Since for the physical mapping  $(q_0, q_1, q_2, q_3) \rightarrow (Q_6, Q_5, Q_{12}, Q_{11})$  in IBM QX5 (see Fig. 7), the corresponding distances  $d_{01}$  and  $d_{23}$  are 0, the initial permutation is left unchanged. Similarly, for  $d = 6$ , the edge weight  $w_{03}$  is only updated to  $w_{03} = 2$  without affecting the initial permutation as shown in Fig. 8a. Next, for  $d = 5$ , the edge weight  $w_{02}$  is updated to  $w_{02} = 1$ . Since the distance  $d_{02}$  is 1 due to this initial permutation, the  $\text{SWAP-cost}$  is updated to 1 and a transformation of the initial permutation  $(q_0, q_1, q_2, q_3) \rightarrow (Q_6, Q_{12}, Q_5, Q_{11})$  results in distance  $d_{0,2} = 0$  and coupling-cost

$$1 \times 0 + 1 \times 8 + 0 \times 2 + 0 \times 33 + 0 \times 18 + 0 \times 5 = 8.$$

The coupling-cost can be verified from the sub-circuit shown in Fig. 8b.

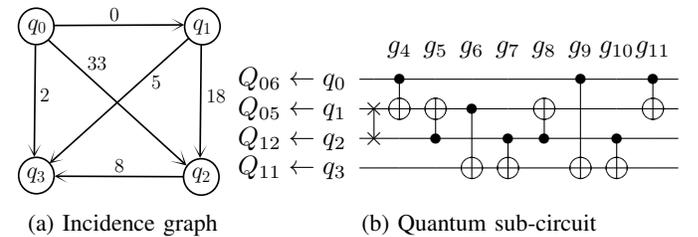


Fig. 8: SWAP-cost estimation for quantum sub-circuit mapped to the IBM QX5

An exhaustive search for an optimal initial permutation of  $n$  logical qubits has a complexity of  $\mathcal{O}(n!)$  w. r. t. the number

of qubits. In order to reduce the run-time, we explore the permutations using a genetic algorithm based search. The initial population is created by random reordering of the physical mapping  $(q_0, \dots, q_{n-1}) \rightarrow (Q_i, \dots, Q_k)$  and the search is carried out on the basis of *SWAP-cost* as fitness estimation of each permutation. The physical mapping obtained by running the genetic algorithm may not be optimal as it may contain redundant gates. The mapped circuit can be further optimized by canceling or merging set of gates as discussed in the next section.

#### D. Optimization

In the IBM QX architecture, all the single qubit gates are realized using the  $U(\theta, \phi, \lambda)$  operation with appropriate values for  $\theta$ ,  $\phi$ , and  $\lambda$ . The single qubit gates are classified as  $U_1(\lambda) = U(0, 0, \lambda)$ ,  $U_2(\phi, \lambda) = U(\frac{\pi}{2}, \phi, \lambda)$ , and  $U_3(\theta, \phi, \lambda) = U(\theta, \phi, \lambda)$  based on the values of these three parameters. The realizations of all the single qubit gates from the Clifford+T library are shown in Table II.

TABLE II: Realization of Single Qubit Gate Operations

Gate	$\theta$	$\phi$	$\lambda$	Gate	$\theta$	$\phi$	$\lambda$
$X$	$\pi$	0	$\pi$	$S$	0	0	$\frac{\pi}{2}$
$Y$	$\pi$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$S^\dagger$	0	0	$-\frac{\pi}{2}$
$H$	$\frac{\pi}{2}$	0	$\pi$	$T$	0	0	$\frac{\pi}{4}$
$Z$	0	0	$\pi$	$T^\dagger$	0	0	$-\frac{\pi}{4}$

A sequence of gates that operates on the same qubit can be merged into a single gate and potentially canceled if the sequence implements the identity operation on combining. Two  $CNOT(q_i, q_j)$  gates can be canceled if there is no gate present in between them whose target is  $q_i$  or control is  $q_j$ . Based on the classification  $U_1$ ,  $U_2$ , and  $U_3$ , single qubit gates can be merged in 9 different ways:

$$\begin{array}{lll} U_1 \times U_1 & U_1 \times U_2 & U_1 \times U_3 \\ U_2 \times U_1 & U_2 \times U_2 & U_2 \times U_3 \\ U_3 \times U_1 & U_3 \times U_2 & U_3 \times U_3 \end{array}$$

For merging  $U_1 \times U_1$ ,  $U_1 \times U_2$ ,  $U_1 \times U_3$ ,  $U_2 \times U_1$ , and  $U_3 \times U_1$ , the following  $Z - Y$  decomposition and merging rule are used:

$$U(\theta, \phi, \lambda) = R_z(\phi) \times R_y(\theta) \times R_z(\lambda)$$

$$R_z(\lambda + \phi) = R_z(\lambda) \times R_z(\phi)$$

where  $R_x(\theta) = U(\theta, -\frac{\pi}{2}, \frac{\pi}{2})$ ,  $R_y(\theta) = U(\theta, 0, 0)$ , and

$$R_z(\theta) = U(0, 0, \frac{\theta}{2}).$$

For the four merging operations  $U_2 \times U_2$ ,  $U_2 \times U_3$ ,  $U_3 \times U_2$ ,  $U_3 \times U_3$ , we are restricted to only Clifford+T gates. In general, merging two gates  $U(\theta_1, \phi_1, \lambda_1)$  and  $U(\theta_2, \phi_2, \lambda_2)$  is carried out in the following way:

- i. If  $\lambda_1 + \phi_2 = 2\pi$  or 0 then

$$U(\theta_1, \phi_1, \lambda_1) \times U(\theta_2, \phi_2, \lambda_2) = U(\theta_1 + \theta_2, \phi_1, \lambda_2)$$

- ii. Otherwise

$$U(\theta_1, \phi_1, \lambda_1) \times U(\theta_2, \phi_2, \lambda_2) = U(\theta_2 - \theta_1, \pi + \phi_1, \lambda_2)$$

The rule is only applicable to the merging of Clifford+T gates. The following example illustrates the optimization process.

**Example 9.** Fig. 9a shows a quantum circuit. The third and final CNOT gates operate on the same qubits, and can be canceled as there is no gate in between whose target is their control qubit or control is their target qubit. The fifth X gate and the next Z gate similarly can be merged. The optimized circuit is shown in Fig. 9b.

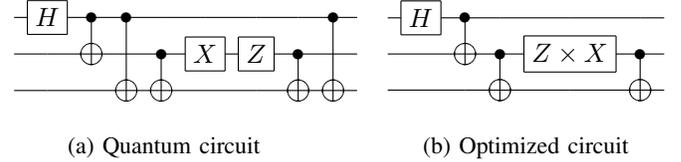


Fig. 9: An example optimization

#### IV. EXPERIMENTAL EVALUATION

To evaluate the performance of the proposed approach, two sets of experiments are conducted.

In the first set of experiments the benchmarks from RevLib [36] and pre-compiled quantum algorithms (by the Scaffold compiler [37]) written in languages like Quipper [32] or Scaffold [31] are considered for mapping to the IBM QX5 architecture. The results are reported in Table III and also compared with the recent work by Zulehner et al. [26].

In conducting the second set of experiments, IBM QX2 and QX4 architectures are used for mapping benchmarks from Roetteler et al. [38] and Zulehner et al. [26], as well as a comparison with the results given by Dueck et al. [25] is presented in Table IV.

The evolutionary algorithm starts with a random population where the best 13.34% of the current population are directly considered for the next generation and the remaining population is generated with crossover and mutation rates of 0.7 and 0.1, respectively. We have restricted the population size and generation exploration to 30 and 500, respectively, for larger benchmarks with 10 or more qubits.

All the experiments are carried out on an Intel Xeon ES 26S0v4, 2.2 GHz server with 16 GB memory. In both these experiments, the observed run-time  $t$  is reported in seconds as the total time taken including selection of physical qubits, initial mapping to logical qubits, insertion of SWAP gates to satisfy coupling-map constraints, and optimizing the final mapped netlist.

##### A. Mapping to the 16-qubit Architecture

Table III shows the mapping results of benchmarks for the IBM QX5 architecture using the proposed approach, which is compared with the earlier reported work by Zulehner et al. [26]. In the first four columns the benchmark name, the number of logical qubits  $n$ , the number of gates  $g$  and the circuit depth  $d$  are presented. The next three columns represent the number of gates  $g$ , the circuit depth  $d$  and the time  $t$  taken

to realize the final quantum circuits satisfying the coupling-map constraint using the A\*-based heuristic search approach proposed by Zulehner et al. [26]. The results observed by the proposed approach are presented in the form of number of gates  $g$ , circuit depth  $d$  and time  $t$  taken to obtain the mapped circuit in the next three columns. Finally, the percentage improvement (%) observed in terms of the number of gates  $\Delta_g$  and the depth  $\Delta_d$  over the results from Zulehner et al. [26] are presented in the last two columns.

It can be seen from the table that, although the run-time of the proposed approach is higher, it provides better results in terms of the number of gates. On average, compared to the A\*-based method proposed by Zulehner et al. [26], 8% reduction in the number of gates is observed while the circuit depth increases by 0.6% using the proposed method. This is mainly due to the fact that for circuits like *ising\_model\_13* and *ising\_model\_16* our proposed approach—although reducing the number of gates extensively compared to the result from Zulehner et al. [26] (17% and 26.5%, respectively)—fails to improve the depth of the final circuit over the corresponding circuit depth observed by the authors in [26]. The circuit depth of these benchmarks are increased to a great extent, i.e. 31.9% and 25% increase in depth are observed in mapping the benchmarks *ising\_model\_13* and *ising\_model\_16* to the IBM QX5 architecture over results presented by Zulehner et al. [26].

### B. Mapping to the 5-qubit Architecture

In the second stage of evaluation, the mapping results of the proposed approach for IBM QX2 and QX4 architectures are compared with the results presented by Dueck et al. [25]. The details are presented in Table IV. The names of the benchmarks and number of logical qubits  $n$  are presented in the first two columns of the table. In the next two columns, the best mapping results to IBM QX2 architecture using *swap transformation* and *template transformation* approaches given by Dueck et al. [25] are presented in terms of the number of gates  $g$  and circuit depth  $d$ . Following this the mapping results obtained by the proposed approach are presented as the number of gates  $g$ , circuit depth  $d$  and run-time  $t$ . The improvements gained by the proposed approach over the best mapped result observed by Dueck et al. [25] in reducing number of gates  $\Delta_g$  and circuit depth  $\Delta_d$  to map to IBM QX2 are listed next. It can be verified that on average 18.2% reduction in the number of gates and 11.8% reduction in circuit depth are observed when mapping to the IBM QX2 architecture using our proposed approach. Similarly, the best results obtained from mapping to IBM QX4 using *swap transformation* and *template transformation* approaches proposed by Dueck et al. [25] are presented in the next two columns, i.e. the number of gates  $g$  and circuit depth  $d$ . The outcome of mapping the circuits by running the proposed approach are listed next as the number of gates  $g$ , circuit depth  $d$  and time taken to produce the mapped circuit  $t$ . Finally the decrease in the number of gates  $\Delta_g$  and circuit depth  $\Delta_d$  over the best results from Dueck et al. [25] are presented. Our approach shows an average improvement of 10.1% in the number of gates and 3.8% in the depth of the final mapped circuit.

## V. CONCLUSION

In this work, we presented a generic framework for compiling quantum circuits to the IBM QX architectures, satisfying the coupling-map constraints. For mapping quantum circuits to a specific IBM QX architecture, the proposed approach addresses this mapping problem in three stages: (i) a greedy approach for the initial selection of  $n$  out of  $m$  ( $m \geq n$ ) physical qubits available in the architecture for mapping a quantum circuit composed of  $n$  logical qubits, (ii) define a fitness measure for the initial permutation of  $n$  logical qubits for mapping physical qubits and perform an evolutionary search using a genetic algorithm to find the best initial permutation  $\pi_0$ , and (iii) devise a strategy for obtaining the local qubit permutation  $\pi_i$  at the circuit depth  $i$  that further reduces the overhead of additional gate operations while satisfying the coupling-map constraints for the remaining gates starting from depth  $i$ . For completeness, we further discussed the post-mapping optimization methods in this work. The proposed method provides better results for the number of gates while not providing improvement over circuit depth for some circuits. A reduction in the number of gates further increases the reliability of quantum circuits by reducing the noise introduced by every gate involved in realizing the desired operation. The proposed generic approach addresses the issue of efficient mapping of quantum circuits to any of the current or projected IBM QX architectures.

## ACKNOWLEDGMENTS

This work was supported by the Dept. of Science and Technology, Govt. of India, by the Austrian Agency for International Cooperation in Education and Research (OeAD), and the LIT Secure and Correct System Lab funded by the State of Upper Austria.

## REFERENCES

- [1] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Symp. on Foundations of Computer Science*, Nov 1994, pp. 124–134.
- [2] L. Grover, "A fast quantum mechanical algorithm for database search," in *ACM Symp. on Theory of computing*, Jul 1996, pp. 212–219.
- [3] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. Sawaya, S. Sim, L. Veis, and A. Aspuru-Guzik, "Quantum chemistry in the age of quantum computing," *arXiv preprint arXiv:1812.09976*, 2018.
- [4] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*. Cambridge Univ. Press, Oct 2000.
- [5] N. M. Linke, D. Maslov, M. Roetteler, S. Debnath, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, "Experimental comparison of two quantum computing architectures," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3305–3310, 2017.
- [6] "IBM Q," <https://www.research.ibm.com/ibm-q/>, [Accessed: 2019-03-20].
- [7] A. Kole and K. Datta, "Improved NCV gate realization of arbitrary size Toffoli gates," in *Int'l Conf. on VLSI Design*, Jan 2017, pp. 289–294.
- [8] 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 Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 6, pp. 818–830, June 2013.
- [9] R. Wille, M. Soeken, C. Otterstedt, and R. Drechsler, "Improving the mapping of reversible circuits to quantum circuits using multiple target lines," in *Asia and South Pacific Design Automation Conf.*, Jan 2013, pp. 145–150.

TABLE III: Mapping to the IBM QX5 Architecture Compared to Zulehner et al. [26]

Benchmark	n	[26]			Proposed solution			Improvement (%)			
		g	d	t	g	d	t	$\Delta_g$	$\Delta_d$		
hwb9	10	207 775	116 199	655 220	375 105	1422.33	605 385	367 472	24 924.80	7.61	2.03
ising_model_10	10	480	70	251	47	4.14	149	47	89.19	40.64	0.00
max46	10	27 126	14 257	84 914	46 270	185.82	79 236	46 652	640.56	6.69	-0.83
mini_alu	10	173	69	474	225	1.25	489	219	223.77	-3.16	2.67
qft_10	10	200	63	447	170	1.25	371	138	150.98	17.00	18.82
rd73	10	230	92	656	301	1.52	673	315	302.30	-2.59	-4.65
sqn	10	10 223	5458	32 095	17 801	68.97	30 172	17 907	316.78	5.99	-0.60
sym9	10	21 504	12 087	66 637	38 489	145.37	62 620	38 057	415.30	6.03	1.12
sys6-v0	10	215	75	613	250	1.36	641	258	296.69	-4.57	-3.20
urf3	10	125 362	70 702	440 509	239 702	873.84	413 778	242 256	9856.78	6.07	-1.07
9symml	11	34 881	19 235	116 508	64 279	254.25	104 360	62 623	880.59	10.43	2.58
dc1	11	1914	1038	5946	3378	12.38	5661	3342	164.22	4.79	1.07
life	11	22 445	12 511	74 632	41 767	166.95	67 764	40 799	575.46	9.20	2.32
sym9	11	34 881	19 235	116 508	64 297	251.42	104 361	62 588	907.26	10.43	2.66
urf4	11	512 064	264 330	1 650 845	878 249	3534.79	1 501 254	840 075	149 023.00	9.06	4.35
wim	11	986	514	2985	1711	6.30	2816	1638	229.29	5.66	4.27
z4	11	3073	1644	9717	5335	20.92	9162	5412	223.80	5.71	-1.44
cm152a	12	1221	684	3738	2155	8.02	3401	2068	142.83	9.02	4.04
cycle10_2	12	6050	3386	19 857	11 141	42.26	18 131	11 045	193.95	8.69	0.86
rd84	12	13 658	7261	45 497	24 473	99.89	42 413	24 800	421.34	6.78	-1.34
sqrt8	12	3009	1659	9744	5501	19.66	9149	5386	229.93	6.11	2.09
sym10	12	64 283	35 572	215 569	118 753	501.02	192 500	115 651	2112.40	10.70	2.61
sym9	12	328	127	955	425	2.08	897	415	472.86	6.07	2.35
adr4	13	3439	1839	11 301	6205	23.17	10 358	5962	266.83	8.34	3.92
dist	13	38 046	19 694	125 867	66 318	291.90	116 175	66 991	1040.44	7.70	-1.01
ising_model_13	13	633	71	329	47	5.11	273	62	167.64	17.02	-31.91
plus63mod4096	13	128 744	72 246	439 918	243 861	1086.48	398 142	239 826	10 436.10	9.50	1.65
radd	13	3213	1781	10 441	5872	22.00	9637	5702	463.61	7.70	2.90
rd53	13	275	124	942	469	1.93	903	496	491.87	4.14	-5.76
root	13	17 159	8835	57 874	30 068	120.82	52 580	30 022	359.35	9.15	0.15
squar5	13	1993	1049	6267	3448	12.96	5887	3424	398.76	6.06	0.70
clip	14	33 827	17 879	114 336	60 882	327.55	106 540	62 388	975.25	6.82	-2.47
cm42a	14	1776	940	5431	3013	11.95	5267	3120	145.31	3.02	-3.55
cm85a	14	11 414	6374	37 746	21 189	242.80	35 538	21 845	491.87	5.85	-3.10
plus63mod8192	14	187 112	105 142	640 204	354 076	1443.33	571 600	344 156	21 845.80	10.72	2.80
pm1	14	1776	940	5431	3013	11.10	4956	2927	174.04	8.75	2.85
sao2	14	38 577	19 563	131 002	66 975	283.90	117 695	67 365	1219.05	10.16	-0.58
sym6	14	270	135	852	456	1.84	935	443	591.03	-9.74	2.85
co14	15	17 936	8570	63 826	30 366	133.71	55 388	31 009	694.89	13.22	-2.12
dc2	15	9462	5242	30 680	17 269	72.53	28 711	17 012	418.14	6.42	1.49
ham15	15	8763	4819	28 310	15 891	68.75	26 590	15 834	356.11	6.08	0.36
misex1	15	4813	2676	15 185	8729	33.11	14 549	8837	169.46	4.19	-1.24
rd84	15	343	110	971	353	2.33	1016	441	740.70	-4.63	-24.93
square_root	15	7630	3847	25 212	13 205	55.35	24 010	14 091	298.19	4.77	-6.71
urf6	15	171 840	93 645	580 295	313 011	1436.16	530 843	293 405	19 046.20	8.52	6.26
cnt3-5	16	485	209	1376	669	3.00	1380	712	794.36	-0.29	-6.43
inc	16	10 619	5863	34 375	19 176	72.85	32 471	19 543	359.61	5.54	-1.91
ising_model_16	16	786	71	426	48	6.47	313	60	264.41	26.53	-25.00
qft_16	16	512	105	1341	404	16.43	994	320	1397.78	25.88	20.79

TABLE IV: Mapping to the IBM QX2 and QX4 Architecture Compared to Dueck et al. [25]

Benchmark	n	QX2						QX4							
		[25]		Proposed solution		Improvement (%)		[25]		Proposed solution		Improvement (%)			
		g	d	g	d	t	$\Delta_g$	$\Delta_d$	g	d	g	d	t	$\Delta_g$	$\Delta_d$
17	4	101	68	97	65	8.98	3.96	4.41	87	56	79	51	9.22	9.20	8.93
a2x_c	4	55	40	39	30	4.18	29.09	25.00	40	28	37	29	4.25	7.50	-3.57
a3x_c	5	86	56	43	32	7.93	50.00	42.86	56	43	46	34	6.14	17.86	20.93
07	5	112	73	105	67	14.19	6.25	8.22	111	64	88	57	13.05	20.72	10.94
Full_Adder_c	4	27	22	27	20	2.89	0.00	9.09	26	23	31	24	2.44	-19.23	-4.35
Toffoli_c	3	17	14	16	15	0.92	5.88	-7.14	17	14	16	15	0.92	5.88	-7.14
4mod5-v0_18	5	152	98	111	75	26.7	26.97	23.47	132	80	138	91	16.36	-4.55	-13.75
3_17_e	3	41	23	33	23	2.72	19.51	0.00	41	23	26	18	2.68	36.59	21.74
01	5	77	38	60	38	8.93	22.08	0.00	77	40	64	40	8.13	16.88	0.00

- [10] D. Miller, R. Wille, and Z. Sasanian, "Elementary quantum gate realizations for multiple-control Toffoli gates," in *Int'l Symp. on Multiple-Valued Logic*, May 2011, pp. 288–293.
- [11] K. Matsumoto and K. Amano, "Representation of quantum circuits with Clifford and  $\pi/8$  gates," *arXiv preprint arXiv:0806.3834*, 2008.
- [12] A. Barenco, C. H. Bennett, R. Cleve, D. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter, "Elementary gates for quantum computation," *Phys. Rev. A*, vol. 52, no. 5, pp. 3457–3467, Nov 1995.
- [13] A. Paler, A. Zulehner, and R. Wille, "NISQ circuit compilers: search space structure and heuristics," *CoRR*, vol. abs/1806.07241, 2018.
- [14] A. Kole, K. Datta, and I. Sengupta, "A new heuristic for  $N$ -dimensional nearest neighbor realization of a quantum circuit," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 37, no. 1, pp. 182–192, Jan 2018.
- [15] R. Wille, A. Lye, and R. Drechsler, "Exact reordering of circuit lines for nearest neighbor quantum architectures," *IEEE Trans. on CAD*, vol. 33, no. 12, pp. 1818–1831, Dec 2014.
- [16] A. Lye, R. Wille, and R. Drechsler, "Determining the minimal number of swap gates for multi-dimensional nearest neighbor quantum circuits," in *Asia and South Pacific Design Automation Conf.*, Jan 2015, pp. 178–183.
- [17] R. Wille, O. Keszoce, M. Walter, P. Rohrs, A. Chattopadhyay, and R. Drechsler, "Look-ahead schemes for nearest neighbor optimization of 1D and 2D quantum circuits," in *Asia and South Pacific Design Automation Conf.*, Jan 2016, pp. 292–297.
- [18] A. Shafaei, M. Saeedi, and M. Pedram, "Optimization of quantum circuits for interaction distance in linear nearest neighbor architectures," in *Design Automation Conf.*, May 2013, pp. 41:1–41:6.
- [19] M. Alfaiakawi, L. Alterkawi, I. Ahmad, and S. Hamdan, "Line ordering of reversible circuits for linear nearest neighbor realization," *Quant. Info. Proc.*, vol. 12, no. 10, pp. 3319–3339, Oct 2013.
- [20] A. Matsuo and S. Yamashita, "Changing the gate order for optimal LNN conversion," in *Int'l Conf. of Reversible Computation*. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 89–101.
- [21] M. Saeedi, R. Wille, and R. Drechsler, "Synthesis of quantum circuits for linear nearest neighbor architectures," *Quant. Info. Proc.*, vol. 10, no. 3, pp. 355–377, Jun 2011.
- [22] Y. Hirata, M. Nakanishi, S. Yamashita, and Y. Nakashima, "An efficient conversion of quantum circuits to a linear nearest neighbor architecture," *Quantum Info. Comput.*, vol. 11, no. 1, pp. 142–166, Jan 2011.
- [23] M. Perkowski, M. Lukac, D. Shah, and M. Kameyama, "Synthesis of quantum circuits in linear nearest neighbor model using Positive Davio Lattices," *Elec. Energ.*, vol. 24, no. 1, pp. 71–87, 2012.
- [24] A. Zulehner and R. Wille, "Compiling SU (4) quantum circuits to IBM QX architectures," in *Asia and South Pacific Design Automation Conf.*, 2019, pp. 185–190.
- [25] G. W. Dueck, A. Pathak, M. M. Rahman, A. Shukla, and A. Banerjee, "Optimization of circuits for IBM's five-qubit quantum computers," in *Euromicro Conf. on Digital System Design*, 2018, pp. 680–684.
- [26] A. Zulehner, A. Paler, and R. Wille, "An efficient methodology for mapping quantum circuits to the IBM QX architectures," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 38, no. 7, pp. 1226–1236, 2019. [Online]. Available: [http://ic.jku.at/eda/research/ibmq\\_mapping/](http://ic.jku.at/eda/research/ibmq_mapping/)
- [27] "Qiskit Python SDK," <https://github.com/Qiskit/qiskit-terra>, [Accessed: 2019-03-20].
- [28] M. Y. Siraichi, V. F. dos Santos, S. Collange, and F. M. Q. Pereira, "Qubit allocation," in *Int'l Symp. on Code Generation and Optimization*, 2018, pp. 113–125.
- [29] A. Botea, A. Kishimoto, and R. Marinescu, "On the complexity of quantum circuit compilation," in *Int'l Symp. on Combinatorial Search*, 2018, pp. 138–142.
- [30] R. Wille, L. Burgholzer, and A. Zulehner, "Mapping quantum circuits to IBM QX architectures using the minimal number of SWAP and H operations," in *Design Automation Conf.*, 2019, p. 142.
- [31] A. J. Abhari, A. Faruque, M. J. Dousti, L. Svec, O. Catu, A. Chakrabati, C.-F. Chiang, S. Vanderwilt, J. Black, and F. Chong, "Scaffold: Quantum programming language," Princeton univ nj dept of computer science, Tech. Rep., 2012.
- [32] A. S. Green, P. L. Lumsdaine, N. J. Ross, P. Selinger, and B. Valiron, "Quipper: a scalable quantum programming language," in *ACM SIGPLAN Notices*, vol. 48, no. 6, 2013, pp. 333–342.
- [33] A. W. Cross, L. S. Bishop, J. A. Smolin, and J. M. Gambetta, "Open quantum assembly language," *arXiv preprint arXiv:1707.03429*, 2017.
- [34] "IBM QX backend information," <https://github.com/Qiskit/ibmq-device-information>, [Accessed: 2019-03-20].
- [35] P. Z. Ingerman, "Algorithm 141: path matrix," *Communications of the ACM*, vol. 5, no. 11, p. 556, 1962.
- [36] R. Wille, D. Große, L. Teuber, G. Dueck, and R. Drechsler, "RevLib: An online resource for reversible functions and reversible circuits," in *Int'l Symp. on Multi-Valued Logic*, 2008, pp. 220–225, RevLib is available at <http://www.revlib.org>.
- [37] A. JavadiAbhari, S. Patil, D. Kudrow, J. Heckey, A. Lvov, F. T. Chong, and M. Martonosi, "Scaffcc: A framework for compilation and analysis of quantum computing programs," in *Computing Frontiers Conference*, 2014, pp. 1:1–1:10.
- [38] M. Roetteler, M. Soeken, and N. Wiebe, "Reversible logic synthesis and quantum computing benchmarks," <https://github.com/knsmith/optimal-single-target-gates>, [Accessed: 2019-05-23].



**Abhoy Kole** (M'16) received the M.Tech. degree in information and communication technology from the Indian Institute of Technology Kharagpur, Kharagpur, India, in 2015.

He is currently pursuing his Ph.D. degree in Rajendra Mishra School of Engineering Entrepreneurship at the Indian Institute of Technology, Kharagpur, India. He has published eleven papers in peer reviewed journals and conferences. His current research interests include reversible and quantum computing, synthesis and optimization of quantum circuits, nearest neighbor realizations, and quantum fault tolerance.



**Stefan Hillmich** (S'19) received the B.Sc. and M.Sc. in Computer Science from the University of Bremen, Germany, in 2015 and 2018, respectively.

Since 2018, he is a researcher at the Johannes Kepler University Linz, Austria, and is working towards his Ph.D. His research interests include simulation, compilation, and verification as parts of design automation for quantum computing.



**Kamalika Datta** (S'07-M'13) completed her Master of Science (MS) from Indian Institute of Technology Kharagpur, India in 2010, and Ph.D. from Indian Institute of Engineering Science and Technology (IEST), Shibpur, India in 2014.

She joined the National Institute of Technology Meghalaya, India as an Assistant Professor in the Department of Computer Science and Engineering in the year 2014. She is presently working as a Research Fellow at the Nanyang Technological Institute Singapore. She has published more than 60 papers

in peer reviewed journals and conferences. Her research interests include logic design using emerging technologies, and synthesis and optimization of reversible and quantum circuits.



**Robert Wille** (M06SM15) is Full Professor at the Johannes Kepler University Linz, Austria. He received the Diploma and Dr.-Ing. degrees in Computer Science from the University of Bremen, Germany, in 2006 and 2009, respectively. Since then, he worked at the University of Bremen, the German Research Center for Artificial Intelligence (DFKI), the University of Applied Science of Bremen, the University of Potsdam, and the Technical University Dresden. Since 2015, he is working in Linz. His research interests are in the design of circuits and

systems for both conventional and emerging technologies. In these areas, he published more than 250 papers in journals and conferences and served in editorial boards and program committees of numerous journals/conferences such as TCAD, ASP-DAC, DAC, DATE, and ICCAD. For his research, he was awarded, e.g., with a Best Paper Award at ICCAD, a DAC Under-40 Innovator Award, a Google Research Award, and more.



**Indranil Sengupta** (SM'13) obtained his B.Tech., M.Tech. and Ph.D. degrees in Computer Science from the University of Calcutta in the years 1983, 1985 and 1990 respectively.

He joined Indian Institute of Technology Kharagpur, India, as a faculty member in the year 1988, in the Department of Computer Science and Engineering, where he is presently a Full Professor. He had been the former Heads of the Department of Computer Science and Engineering, and School of Information Technology. He has over 30 years of

teaching and research experience, guided 21 PhD students and published over 200 papers in peer reviewed journals and conferences. He has served as the Program Chair / General Chair in several International Conferences in the areas of reversible computing, VLSI design/test and information security. His research interests include reversible and quantum computing, VLSI design and test, and network security.