Synthesis of Quantum Circuits for Dedicated Physical Machine Descriptions

Philipp Niemann[1], Saikat Basu[2],
Amlan Chakrabarti[2], Niraj K. Jha[3], and Robert Wille[1,4]

[1] Institute of Computer Science, University of Bremen, D-28359 Bremen, Germany
[2] A.K. Choudhury School of I.T, University of Calcutta, Calcutta, India
[3] Department of Electrical Engineering, Princeton University, NJ 08544, USA
[4] Cyber-Physical Systems, DFKI GmbH, D-28359 Bremen, Germany
{pniemann,rwille}@informatik.uni-bremen.de
{acakcs}@caluniv.ac.in
jha@princeton.edu

Abstract. Quantum computing has been attracting increasing attention in recent years because of the rapid advancements that have been made in quantum algorithms and quantum system design. Quantum algorithms are implemented with the help of quantum circuits. These circuits are inherently reversible in nature and often contain a sizeable Boolean part that needs to be synthesized. Consequently, a large body of research has focused on the synthesis of corresponding reversible circuits and their mapping to the quantum operations supported by the quantum system. However, reversible circuit synthesis has usually not been performed with any particular target technology in mind, but with respect to an abstract cost metric. When targeting actual physical implementations of the circuits, the adequateness of such an approach is unclear. In this paper, we explicitly target synthesis of quantum circuits at selected quantum technologies described through their *Physical Machine Descriptions* (PMDs). We extend the state-of-the-art synthesis flow in order to realize quantum circuits based on just the primitive quantum operations supported by the respective PMDs. Using this extended flow, we evaluate whether the established reversible circuit synthesis methods and metrics are still applicable and adequate for PMD-specific implementations.

1 Introduction

Supporting the design of quantum circuits is one of the main applications of reversible logic. Quantum circuits [1] promise to significantly speed up solutions for computing problems of practical interest. This is enabled by quantum mechanical properties, such as superposition and entanglement. Quantum circuits execute a sequence of quantum operations. These quantum operations are inherently reversible. Often, significant parts of a quantum computation (e.g., database search [2] and modular exponentiation [3]) are Boolean in nature. Thus, leveraging existing reversible logic synthesis methods for implementing those parts is an obvious first step.

Over the past few years, a popular synthesis flow has been to

- realize the desired functionality as a reversible circuit and
- map the resulting reversible circuit to an equivalent cascade of quantum gates/operations.

A large body of research has been targeted at both these steps (e.g., reversible circuit synthesis [4–9] and mapping to quantum circuits [10–13]). Most of these methods target NOT, controlled-NOT, and controlled-**V** as the set of primitive quantum operations. This set is popularly referred to as *NCV* and its use was originally motivated by one of the first works on reversible-to-quantum circuit mapping by Barenco et al. [10]. However, several new quantum systems have emerged in recent years. The ARDA quantum computing roadmap [14] lists some of them. These systems are described using *Physical Machine Descriptions* (PMDs) [15]. They describe different technologies for the realization of quantum circuits based on the respective quantum mechanical properties. Moreover, each PMD supports a specific set of primitive quantum operations. Thus, mapping a quantum circuit to a PMD is not yet compatible with the established synthesis flow that targets NCV-based circuit implementations only.

In this work, we investigate and evaluate the applicability of the state-of-the-art NCV-based synthesis flow, which has emerged over the last 10-20 years, for mapping quantum circuits to a particular PMD [15]. We first review today's established NCV-based synthesis steps. Then, we propose extensions to this flow, e.g., mapping schemes from reversible circuits or NCV-based quantum circuits to PMD-specific quantum circuits, and analyze the circuit cost. Finally, we perform an experimental evaluation to a) compare synthesis flows with different extensions and b) investigate whether the established synthesis methods and metrics are still applicable and adequate for the PMD-specific circuit realization. This can throw light on the drawbacks and provide potential for improvements in quantum circuit synthesis.

The remainder of this paper is organized as follows. The next section briefly reviews the basics of quantum and reversible circuits. Then, a review of the PMDs targeted in this work is provided in Section 3. Section 4 describes the synthesis flows. First, the state-of-the-art synthesis flow is discussed, followed by its extension to a PMD-specific synthesis flow. Then, the cost metrics used in the evaluation are described in Section 5. The evaluation and discussion follow in Section 6 and the paper is concluded in Section 7.

2 Background

This section reviews the basics of quantum and reversible circuits.

2.1 Quantum Circuits

First, we discuss the preliminaries of quantum logic. Quantum operations manipulate qubits rather than classical bits. A qubit can represent 0 or 1 as well as superpositions of the two. More formally:

Definition 1. *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 $|\alpha|^2 + |\beta|^2 = 1$.*

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 transformation operations satisfying the following:

Definition 2. *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 $\mathbf{H} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, which is the well-known Hadamard operation [1]. 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 $\mathbf{T} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$, ●—⊕ $(CNOT)$, etc. are applied successively from left to right.

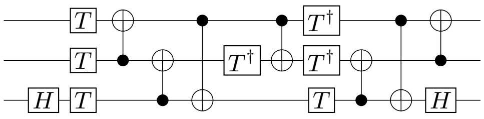


Fig. 1. A quantum circuit

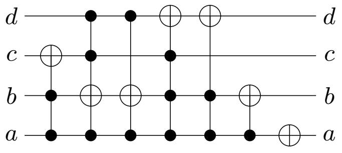


Fig. 2. A reversible circuit

Several libraries of quantum operations have been presented in the literature. From a theoretical point of view, the set of arbitrary one-qubit gates (unitary 2×2 matrices) and a single 2-qubit gate, namely the controlled-NOT ($CNOT$) gate, is sufficient to approximate any quantum operation to an arbitrary precision [1]. However, the technologies that are actually used for the physical realization of quantum circuits support a small subset of quantum operations only. This is discussed in more detail in Section 3. Moreover, these technologies are much more fault-prone than classical technologies since the phenomenon of quantum decoherence forces the qubit states to decay – resulting in a loss of quantum information. To address this issue, specific *fault-tolerant* (FT) quantum gate libraries have been presented for the synthesis of quantum circuits.

2.2 Reversible Circuits

A special case of unitary matrices are permutation matrices. These matrices only contain entries 0 and 1 (there is a single 1 in every row/column) and represent classical reversible functions, i.e., Boolean functions $f : \mathbb{B}^n \to \mathbb{B}^n$ that map each input pattern to a unique output pattern. In other words, reversible functions are bijections that perform a permutation of the set of input patterns. A large body of research has focused on synthesizing initial representations of reversible functions, e.g., in terms of truth tables, two-level representations, binary decision diagrams, and permutation matrices, to *reversible circuits* [4–9]. These circuits commonly consist of a set of lines (corresponding to qubits) and reversible gates. The most established type of reversible gates is the *multiple-controlled Toffoli* (MCT) gate. MCT gates consist of a possibly empty set of control lines and a single target line that is inverted if and only if all control lines carry the value 1. Note that the MCT gate library also includes the special cases of NOT (empty set of controls) and controlled-NOT ($CNOT$) gates (singleton set of controls). For historical reasons and for brevity, we will simply use Toffoli gate to refer to the 2-controlled Toffoli gate.

Example 3. Consider the reversible circuit shown in Fig. 2 that realizes a modulo 10 counter. More precisely, if the input – taken as a binary number $dcba_2$ – is less than or equal to (the decimal number) 10, then the output is incremented and taken modulo 10, i.e., the output is $((dcba + 1)\%10)_2$. For binary numbers larger than 10, the circuit does not behave according to this formula. However, it is clear that – due to reversibility – the output also has to be larger than 10.

It is a common phenomenon that, as in the previous example, reversible circuits have a meaningful output only for a subset of the input patterns. This is because many reversible functions are obtained by *embedding* an irreversible function into a reversible one [16], by adding extra input/output lines in order to ensure a bijective mapping.

3 Physical Machine Descriptions

The physical realization of quantum circuits is a difficult task – especially for circuits with a large number of qubits [1]. It needs well-formed qubit states and their transformation through the time-dependent Hamiltonian of the physical system [1]. In general, a quantum circuit implements the unitary operator corresponding to the Hamiltonian evolution of the qubit states. A quantum technology describes a physical system for qubit realization and a set of primitive quantum operations for realizing the Hamiltonian. A broad survey of quantum systems has been conducted in the ARDA quantum computing roadmap [14]. This motivates the consideration of *Physical Machine Descriptions* (PMDs) [15]. Each PMD is different in terms of its quantum mechanical properties. This leads to different Hamiltonians and, hence, a different set of supported (primitive) operations.

In this work, we target PMDs of six quantum systems, namely *Quantum Dots* (QD), *Superconducting Qubits* (SC), *Ion Traps* (IT), *Neutral Atoms* (NA), *Linear Photonics* (LP), and *Non-linear Photonics* (NP). In this section, we provide a brief review of these quantum systems[1]. Then, we summarize the primitive quantum operations supported by the respective PMDs. This provides the basis for a detailed consideration of synthesis issues in the remainder of this paper. The targeted PMDs are described next.

- *Quantum Dots (QD)*
  In this system, a qubit is defined by the spin state of a single-electron quantum dot, which is confined by electrostatic potential. The desired quantum operations are implemented by gating of the tunneling barrier between neighboring dots [17].
- *Superconducting Qubits (SC)*
  In a superconducting system, a qubit is simply represented by the two rotation directions of the persistent super-current of Cooper pairs in a superconducting ring containing Josephson tunnel junctions [18]. The state of a qubit is defined by a distribution of voltages or currents, each characterized by an amplitude and phase, which are functions of time.
- *Ion Traps (IT)*
  Ion-trap quantum computation can be implemented by confining a string of ions in a single trap, exploiting their electronic states as qubit logic levels, and using mutual Coulomb interaction for transferring quantum information between ions [19].
- *Neutral Atoms (NA)*
  A system of trapped neutral atoms is a good candidate for implementing scalable quantum computing [20] [21]. That the atoms are neutral means that they are feebly coupled to the environment. Hence, decoherence is minimized. Trapped atoms can be cooled to the motional ground state of the quantized potential wells, and the initialization of the internal atomic states can be performed using standard techniques of laser spectroscopy. The different qubit levels can be described by various motional and internal states of the neutral atoms.

[1] We keep the respective descriptions brief, but provide references for further reading.

Table 1. Primitive quantum operations supported by different PMDs

PMD	One-qubit operations	Two-qubit operations
QD	R_x, R_z, σ_x, σ_z, S, T	CZ
SC	R_x, R_y, R_z	$iSWAP$, CZ
IT	R_{xy}, R_z	G
NA	R_{xy}	CZ
LP	R_x, R_y, R_z, σ_x, σ_y, σ_z, S, T, H	$CNOT$, CZ, $SWAP$, $ZENO$
NP	A_{squ}, R_x, R_y, R_z, H	$CNOT$

- *Linear Photonics (LP)*
  In linear photonics, the qubits are represented by the quantum state of single photons. Quantum logic gates can be constructed using only linear optical elements, such as mirrors and beamsplitters, additional resource photons, and triggering signals from a single-photon detector [22].
- *Non-Linear Photonics (NP)*
  In nonlinear photonics, quantum logic gates are implemented using interactions of photons with nonlinear photonic crystals. The photonic crystals include layers of a Kerr medium [23] and, thus, perform a nonlinear shift of the photonic wave function.

Each of the PMDs described above relies on a different quantum mechanical property and, hence, a different set of supported (primitive) quantum operations. Table 1 provides a list of supported one-qubit and two-qubit operations [15]. More precisely:

- R_x, R_y, and R_z realize rotations around the x, y, and z axis of the Hamiltonian, respectively. They are parametrized by a rotation angle θ. The corresponding matrices are

  $$R_x(\theta) = \left(\begin{smallmatrix} \cos(\frac{\theta}{2}) & -i\sin(\frac{\theta}{2}) \\ -i\sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{smallmatrix}\right), R_y(\theta) = \left(\begin{smallmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{smallmatrix}\right), \text{ and } R_z(\theta) = \left(\begin{smallmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{smallmatrix}\right).$$

  For FT implementations, the angle θ must be a multiple of $\frac{\pi}{4}$ [24].
- The Pauli operations σ_x (=NOT), σ_y, and σ_z (sometimes also denoted by X, Y, and Z) are special cases of these rotations for $\theta = \pi$ (up to global phase, i.e., a physically indistinguishable multiplicative factor).
- $S = \left(\begin{smallmatrix} 1 & 0 \\ 0 & i \end{smallmatrix}\right)$ and $T = \left(\begin{smallmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{smallmatrix}\right)$ are special cases of the R_z gate with rotation angle $\theta_S = \frac{\pi}{2}$ and $\theta_T = \frac{\pi}{4}$, respectively (also up to global phase).
- R_{xy} and A_{squ} are multi-rotation gates with two parameters. For our purpose, it is sufficient to know that R_x and R_y are special cases of R_{xy} and that R_z rotations can be implemented by two R_{xy} rotations.
- In the case of two-qubit operations, it is sufficient to know about operations CZ and G, which perform phase shifts. CZ denotes the controlled-Z operation (defined analogously to the controlled-NOT operation). It is represented by the 4×4 diagonal matrix $CZ = \mathrm{diag}(1, 1, 1, -1)$, whereas the parametrized G operation is represented by $G(\theta) = \mathrm{diag}(1, e^{i\theta}, e^{i\theta}, 1)$.

These different PMD-specific sets of supported operations pose a significant challenge for synthesis: mapping of the circuit has to be performed to each PMD separately. Most existing synthesis methods do not target the gate libraries given in Table 1. Hence, in the remainder of this paper, we address the question of how we can utilize existing synthesis flows for the synthesis of PMD-specific quantum circuits.

4 Synthesis Flow

Since synthesis of quantum circuits is a complex task, many (automatic) methods employ a synthesis flow that *does not directly* realize the given quantum functionality, but employs a *multiple-step* approach. For this purpose, two main characteristics are exploited, namely

– many important quantum algorithms, like Grover's database search algorithm [2] and Shor's factorization algorithm [3], contain a considerable reversible (Boolean) component that needs to be synthesized, and
– all quantum operations are inherently reversible.

Consequently, the quantum functionality of Boolean components are first realized as a reversible circuit, rather than a quantum circuit. This significantly reduces synthesis complexity. Besides, a huge variety of synthesis approaches is already available (e.g., [4–9]). Then, the resulting reversible circuit is mapped to an equivalent quantum circuit representation.

In this section, we first review this established synthesis flow and its current assumptions. Then, we discuss how this flow can be extended to obtain PMD-specific realizations that can be executed in the respective technologies. A comparison of these different extensions at a theoretical level (with respect to the resulting cost metrics) as well as through an experimental evaluation will follow in Sections 5 and 6, respectively.

4.1 State-of-the-art Synthesis

The established synthesis flow for quantum circuits is sketched by solid lines and boxes in Fig. 3. Starting with the desired functionality (e.g., provided in the form of truth tables, two-level representations, binary decision diagrams, or permutation matrices), the first step is to generate a reversible circuit realizing the corresponding function (Step (a) in Fig. 3). A large body of research has focused on this step [4–9].

In the following step, the resulting reversible circuit is mapped to an equivalent quantum circuit representation (Step (b) in Fig. 3). The key to this task can be found in the seminal work by Barenco et al. [10] for realizing the Toffoli gate at the quantum level. This is done using the *NCV* library that is composed of

– NOT gates,
– controlled-NOT (*CNOT*) gates, as defined in Section 2,
– controlled-$\mathbf{V}$ gates that are defined analogously, but, when activated, perform the operation $\mathbf{V} = \frac{1+i}{2}\left(\begin{smallmatrix} 1 & -i \\ -i & 1 \end{smallmatrix}\right)$, and
– controlled-$\mathbf{V}^\dagger$ gates[2] that realize the inverse operation $\mathbf{V}^\dagger = \frac{1-i}{2}\left(\begin{smallmatrix} 1 & i \\ i & 1 \end{smallmatrix}\right)$.

[2] Since two $\mathbf{V}$ gates or two $\mathbf{V}^\dagger$ gates in a sequence realize a *CNOT* operation, their corresponding operation is usually called "the square root of NOT".

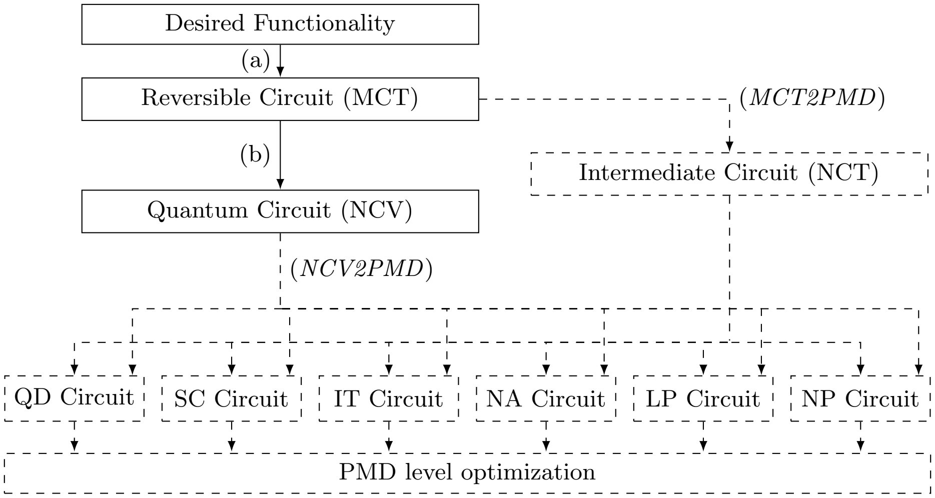


Fig. 3. Synthesis flow for reversible circuits

Fig. 4 shows the mapping of the Toffoli gate (with two control lines) to the NCV library. This mapping can be extended to MCT gates. A naive way to do this would be to decompose the MCT gate into a cascade of Toffoli gates that can, in turn, be mapped to the NCV library (Fig. 4). However, researchers have come up with highly optimized, direct mapping of MCT gates to the NCV library (e.g., [10–13]). These mapping methods have had a significant impact on how reversible circuits are optimized. In fact, the number of NCV gates required to realize an MCT gate has become a major optimization criterion for the synthesis of reversible circuits in Step (a). This has led to *NCV library based quantum cost* to become a widely accepted cost metric for evaluating reversible circuits.

4.2 PMD-specific Synthesis

Quantum circuits can be mapped to PMDs in a manner similar to how they are mapped to the NCV library, i.e., actual synthesis is conducted at the reversible circuit level whereas the desired PMD-specific circuit is obtained through a mapping scheme. Since PMD-specific mappings tend to create the potential for further circuit minimization at the PMD level, another optimization step is usually carried out at this level [15, 24, 25].

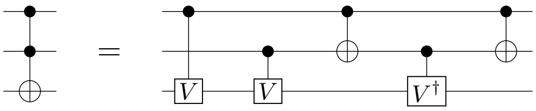


Fig. 4. Quantum level decomposition of the Toffoli gate [10]

The above considerations lead to the synthesis flow for PMD-specific quantum circuits sketched by dashed lines and boxes in Fig. 3. More precisely, the mapping to PMD-specific circuits can be accomplished in two ways, namely by

- a mapping from a reversible circuit based on MCT gates (*MCT2PMD*) or
- a mapping from the NCV library based quantum circuit (*NCV2PMD*).

As neither the MCT nor the NCV library is directly supported by any of the PMDs (see Table 1), both approaches eventually require a mapping scheme of the respective gates from these libraries to the PMD level. However, since it is already a challenging task to find corresponding mappings for gates operating on a small number of qubits, we do not aim to obtain direct mappings for large MCT gates with more than two control lines. Instead, we propose to employ the decomposition of large MCT gates into cascades of Toffoli gates [10]. In this way, an arbitrary MCT gate can be implemented using the reduced *NCT* library composed of only *N*OT, *C*ontrolled-NOT, and *T*offoli gates. With this intermediate representation, a mapping to the PMD level is finally required for only three gates (NCT). Moreover, these mappings can be reused to also generate PMD-specific circuits from NCV representations – only one additional mapping, namely for controlled-**V** gates, is required for this purpose[3].

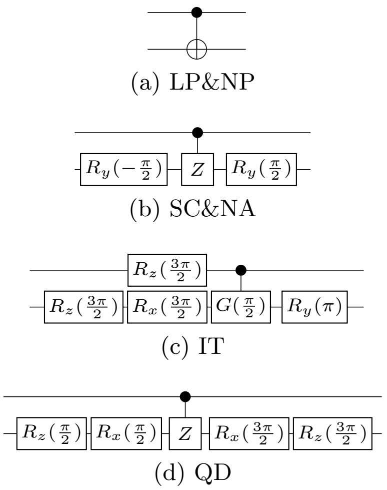
Mappings for the controlled-NOT, controlled-**V**, and Toffoli gates are shown in Fig. 5, Fig. 6, and Fig. 7, respectively[4]. Note that the presented mappings employ an FT quantum gate library [24]. Cheaper mappings are available when dropping the FT implementation requirement [15]. However, as faults are a major concern in quantum circuits, the use of FT quantum gates is important. To this end, the FT gate library which we employ here is ideal to use with quantum error-correcting (QEC) codes as it is closely related to the Clifford+T library (as also discussed in [24]). In fact, some post-processing is necessary to use a specific QEC code, but this is beyond the scope of this paper.

The above mappings enable the application of the two proposed schemes, *MCT2PMD* and *NCV2PMD*, for the synthesis of quantum circuits for direct execution on the corresponding PMD. We have summarized the respective costs of single NCT/NCV gates for each PMD in Table 2[5]. As can be seen, the costs are significantly different across PMDs. Even more importantly, the actual costs of implementing a *CNOT*, controlled-**V**, and Toffoli gate are up to a factor of 8 higher than in the case of the NCV library based quantum cost. The consequences of these cost differences will be further analyzed in the following section. Then, the efficiency of both schemes will be compared through an experimental evaluation in Section 6.

[3] Note that mappings for the controlled-$\mathbf{V}^\dagger$ gate are not needed explicitly as they can be derived by applying the corresponding mapping of the controlled-**V** gate in reverse order and with inverted gates.

[4] Mappings for the *CNOT* gate were presented earlier in [15], however, are shown here for the sake of completeness.

[5] Recall that an R_z rotation is implemented by two R_{xy} rotations in NA. Consequently, an R_z operation has a gate count of 2.



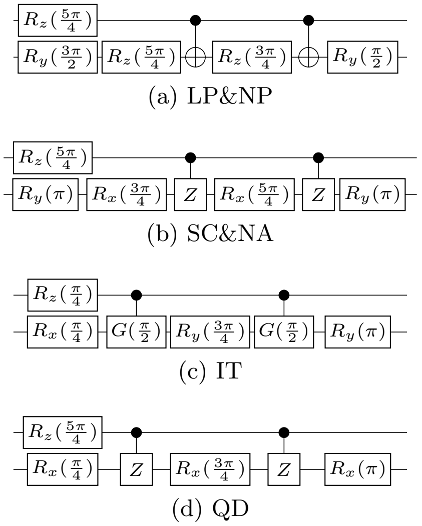
(a) LP&NP

(b) SC&NA

(c) IT

(d) QD

Fig. 5. Mapping a *CNOT* to PMDs



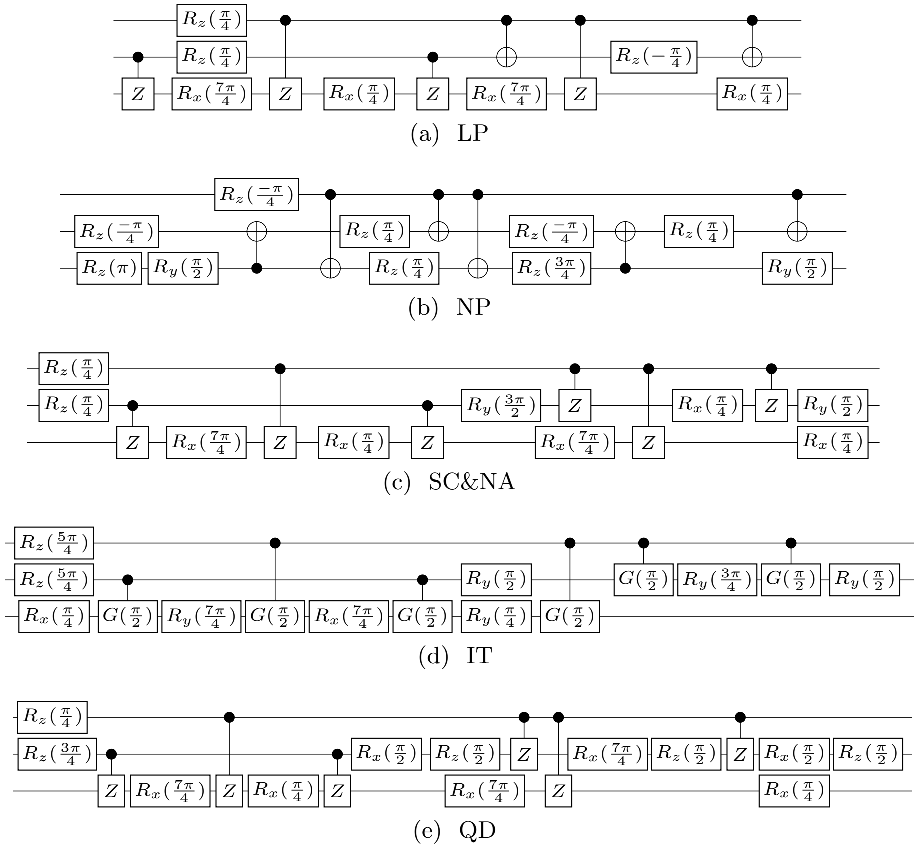
(a) LP&NP

(b) SC&NA

(c) IT

(d) QD

Fig. 6. Mapping of controlled-**V** gate to PMDs



(a) LP

(b) NP

(c) SC&NA

(d) IT

(e) QD

Fig. 7. Mapping of Toffoli gate to PMDs

Table 2. Gate counts of the mappings from Figs. 5-7

PMD	NOT	CONTROLLED-NOT	CONTROLLED-**V**	TOFFOLI
QD	1	5	6	18
SC	1	3	7	15
IT	1	5	6	15
NA	1	3	8	17
LP	1	1	7	13
NP	1	1	7	16
NCV	1	1	1	5

5 Resulting Cost Metrics for MCT Circuit Synthesis

In the previous section, we proposed two different extensions to the state-of-the-art synthesis flow to obtain quantum circuits that can be executed on particular PMDs. More precisely, in the *MCT2PMD* scheme, MCT circuits are, first, transformed to an intermediate representation in terms of NCT gates which are, then, mapped step-by-step to their respective PMD implementations. In contrast, in the *NCV2PMD* scheme, highly optimized NCV circuits are mapped directly to the particular PMD.

We observed that the PMD-specific costs of NCV and NCT gates are substantially higher compared with the usually applied NCV library based quantum costs. As a consequence, MCT gates – which are the actual input of both synthesis schemes – are significantly more expensive when realized on PMDs and the PMD-specific costs differ significantly for the various technologies. Hence, the NCV library based cost metric is no longer valid for PMD-specific synthesis. This poses a problem since NCV library based costs are commonly used in almost all synthesis approaches aimed at generating MCT circuits. Consequently, it could have a significant impact on the synthesis process if PMD-specific cost metrics were used when synthesizing MCT circuits.

In fact, the two mapping schemes give rise to their *own dedicated cost metrics*, as shown in Table 3. Here, the first column denotes the number of control lines of the MCT gate. In the following columns, the costs for the *MCT2PMD* scheme are given. In the first three columns, the required numbers of NOT, $CNOT$, and Toffoli gates for realizing the corresponding MCT gate in the NCT library are provided (based on the decomposition in [10]). Based on these numbers, PMD-specific costs are computed using Table 2 and presented in the following six columns. In the remainder of the table, this procedure is likewise performed for the *NCV2PMD* scheme: first, the numbers of NOT, $CNOT$, and controlled-$\mathbf{V}$ gates are obtained (based on the state-of-the-art mapping [12]) and, then, the PMD-specific costs are computed and shown in the remaining columns.

Overall, the numbers indicate a significant difference between the two mapping schemes with a clear advantage for *NCV2PMD*. However, the actual difference between the two approaches and comparison with NCV library based quantum costs need to be evaluated in practice, especially with respect to optimization performed at the PMD level after technology mapping. This experimental evaluation is conducted in the following section.

Table 3. PMD-specific cost metrics for MCT gates

	MCT2PMD									*NCV2PMD*								
#Contr.	N	C	T	QD	SC	IT	NA	LP	NP	N	C	V	QD	SC	IT	NA	LP	NP
0	1			1	1	1	1	1	1	1			1	1	1	1	1	1
1		1		5	3	5	3	1	1		1		5	3	5	3	1	1
2			1	18	15	15	17	13	16		2	3	28	27	28	30	23	23
3			4	72	60	60	68	52	64		4	10	80	82	80	92	74	74
4			8	144	120	120	136	104	128		4	16	116	124	116	140	116	116
5			12	216	180	180	204	156	192		8	24	184	192	184	216	176	176
6			26	468	390	390	442	338	416		8	36	256	276	256	312	260	260
7			32	576	480	480	544	416	512		12	44	324	344	324	388	320	320
8			40	720	600	600	680	520	640		14	62	442	476	442	538	448	448
9			48	864	720	720	816	624	768		18	70	510	544	510	614	508	508
10			56	1008	840	840	952	728	896		20	88	628	676	628	764	636	636

6 Experimental Evaluation

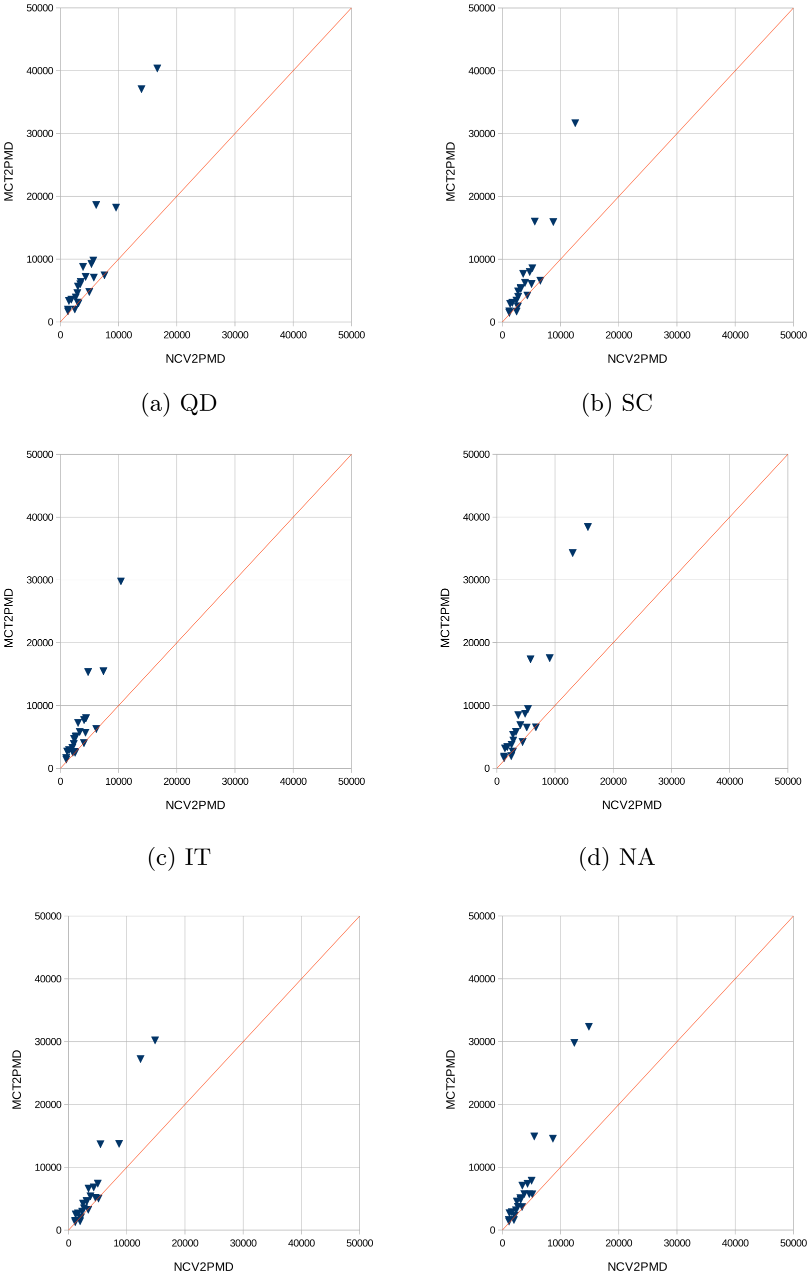
In this section, we summarize the experimental evaluations conducted using the newly proposed synthesis flows: *MCT2PMD* and *NCV2PMD*. More precisely, we investigate which of the two flows actually performs better. In addition, we also evaluate the difference between the commonly used NCV library based cost metric and the PMD-specific cost metrics presented in the previous section.

We synthesized various MCT circuits from the RevLib benchmark suite [26] as PMD-specific quantum circuits using both flows. More precisely, we synthesized medium-sized circuits with an NCV library based quantum cost in the 100 to 15,000 range, such that, on the one hand, the circuits are large enough to enable a meaningful evaluation and, on the other hand, are still amenable to the application of highly elaborate NCV optimization.

We used the state-of-the-art NCV library based mapping scheme presented in [12] to generate the NCV circuits for the *NCV2PMD* scheme. This scheme uses optimized mappings for MCT gates and then performs several heuristic optimizations on the resulting NCV circuit. In both approaches, we used the FTQLS tool [24] – enriched by the FT mappings presented in Section 4.2 – to generate the FT PMD-specific implementations from the NCV and NCT circuits. After this mapping, additional optimization steps, as described in [24], are performed.

6.1 Comparison of the Synthesis Flows

In the first evaluation, we compared the efficacy of the two proposed synthesis flows with respect to circuit cost. The results are summarized in Fig. 8. In each graph, the y axis represents the quantum cost when using the *MCT2PMD* mapping, whereas the x axis represents the quantum cost when using the *NCV2PMD* mapping. The diagonal line represents the cost equilibrium, i.e., circuits that have the same cost for both mapping schemes appear on this line. Circuits that can be realized cheaper with the *MCT2PMD* scheme than with the *NCV2PMD* scheme appear below this line, whereas circuits that can be realized cheaper with the *NCV2PMD* scheme than with the *MCT2PMD* scheme (modulo better future mappings of MCT gates in the latter scheme) appear above this line.



(a) QD

(b) SC

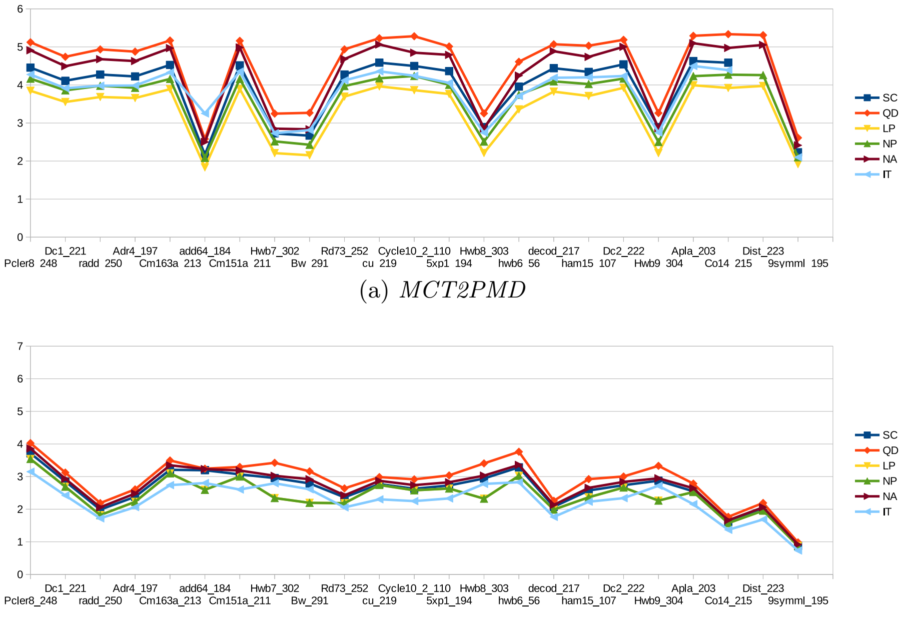
(c) IT

(d) NA

(e) LP

(f) NP

Fig. 8. *MCT2PMD* vs. *NCV2PMD*



(a) *MCT2PMD*

(b) *NCV2PMD*

Fig. 9. NCV library based quantum cost vs. PMD-specific cost

In summary, a small advantage of the *NCV2PMD* scheme can be observed for all PMDs as indicated by the cost metric presented in Section 5.

6.2 Comparison of PMD and NCV Costs

In the second evaluation, we were interested in the difference between the NCV library based quantum costs and the two PMD-specific quantum costs (as proposed in Section 5). This is of particular interest because, thus far, reversible circuits are still being optimized with respect to an NCV library based cost metric. Hence, it is important to understand whether and, if so, what differences exist between these cost metrics.

For this purpose, we compared the obtained NCV library based quantum costs of the initial MCT benchmark circuits to their PMD-specific costs. The relationship between these costs is summarized in Fig. 9 for both mapping schemes. The x axis depicts benchmarks circuits, whereas the y axis provides the ratio of the PMD costs to the NCV costs. Average values and standard deviations for all PMDs are shown in Table 4.

First of all, we observe that for many circuits the cost ratio is close to the average value, i.e., the real cost of the circuit differs from the estimated NCV library based cost only by a constant, PMD-specific, multiplicative factor. This holds for both the *MCT2PMD* and *NCV2PMD* mapping schemes. In these cases, NCV library based quantum cost can be used as a useful proxy for actual cost estimation. However, there are several circuits that significantly deviate from the average, both towards the top and the bottom. In these cases, using NCV library based quantum cost is not an adequate proxy.

Table 4. Statistics for the ratio between PMD-specific and NCV library based cost

MCT2PMD	QD	SC	IT	NA	LP	NP
Average value	4.54	3.90	3.78	4.27	3.35	3.63
Standard deviation	0.95	0.85	0.70	0.97	0.78	0.79
NCV2PMD	QD	SC	IT	NA	LP	NP
Average value	2.89	2.61	2.28	2.70	2.40	2.40
Standard deviation	0.68	0.61	0.55	0.63	0.56	0.56

Overall, the use of PMD-specific cost metrics as optimization criterion is likely to lead to MCT circuits that are better suited for a later mapping to PMDs. Nevertheless, the currently popular NCV library based cost metric may still serve as a useful approximation.

7 Conclusions

In this paper, we considered the design of PMD-specific quantum circuits. PMDs correspond to quantum systems whose quantum mechanical properties are used to implement quantum circuits. As part of its specification, each PMD supports only a restricted set of primitive quantum operations. Consequently, when synthesizing quantum circuits for these PMDs, the specific gate library has to be taken into account. The commonly used synthesis flow for quantum circuits employs a multiple-step scheme in which a reversible circuit (based on MCT gates) is realized first and then mapped to an equivalent cascade of quantum gates. However, this mapping leads to NCV library based quantum circuits that are not directly supported by any of the PMDs. To overcome this problem, we proposed extensions to the existing synthesis flow aimed at synthesis of PMD-specific quantum circuits. To this end, we proposed FT mappings to the PMD level for various quantum gates (from the NCV and the NCT library). An analysis showed that these mappings lead to much higher costs for the realization of MCT gates compared to the commonly used NCV library based quantum costs. An experimental evaluation indeed indicated that there is no simple relation between PMD-specific and NCV library based cost. This motivates the need for a more detailed consideration of PMD-specific synthesis at the reversible circuit level based on the metrics proposed in this work.

References

1. Nielsen, M., Chuang, I.: Quantum Computation and Quantum Information. Cambridge Univ. Press (2000)
2. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Theory of Computing. (1996) 212–219
3. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: Foundations of Computer Science. (1994) 124–134
4. Shende, V., Prasad, A., Markov, I., Hayes, J.: Synthesis of reversible logic circuits. IEEE Trans. on CAD **22**(6) (June 2003) 710–722
5. Gupta, P., Agrawal, A., Jha, N.K.: An algorithm for synthesis of reversible logic circuits. IEEE Trans. on CAD **25**(11) (Nov. 2006) 2317–2330

6. Fazel, K., Thornton, M., Rice, J.: ESOP-based Toffoli gate cascade generation. In: IEEE Pacific Rim Conference on Communications, Computers and Signal Processing. (Aug. 2007) 206–209
7. Wille, R., Drechsler, R.: BDD-based synthesis of reversible logic for large functions. In: ACM Design Automation Conference. (July 2009) 270–275
8. Soeken, M., Wille, R., Hilken, C., Przigoda, N., Drechsler, R.: Synthesis of reversible circuits with minimal lines for large functions. In: ASP Design Automation Conference. (Jan. 2012)
9. Lin, C.-C., Jha, N.K.: RMDDS: Reed-Muller decision diagram synthesis of reversible logic circuits. ACM J. Emerg. Technol. Comput. Syst. **10**(2) (2014)
10. Barenco, A., Bennett, C.H., Cleve, R., DiVincenzo, D., Margolus, N., Shor, P., Sleator, T., Smolin, J., Weinfurter, H.: Elementary gates for quantum computation. The American Physical Society **52** (1995) 3457–3467
11. Maslov, D., Young, C., Dueck, G.W., Miller, D.M.: Quantum circuit simplification using templates. In: Design, Automation and Test in Europe Conference. (2005) 1208–1213
12. Miller, D.M., Wille, R., Sasanian, Z.: Elementary quantum gate realizations for multiple-control Toffoli gates. In: Int. Symposium on Multiple-Valued Logic. (2011) 288–293
13. Wille, R., Soeken, M., Otterstedt, C., Drechsler, R.: Improving the mapping of reversible circuits to quantum circuits using multiple target lines. In: ASP Design Automation Conference. (Jan. 2013) 145–150
14. ARDA: Quantum computation roadmap http://qist.lanl.gov/qcomp_map.shtml.
15. Lin, C.-C., Chakrabarti, A., Jha, N.K.: Optimized quantum gate library for various physical machine descriptions. IEEE Trans. on Very Large Scale Integration (VLSI) Systems **21**(11) (Nov. 2013) 2055–2068
16. Wille, R., Keszöcze, O., Drechsler, R.: Determining the minimal number of lines for large reversible circuits. In: Design, Automation and Test in Europe Conference. (2011) 1204–1207
17. Taylor, J.M., Petta, J.R., Johnson, A.C., Yacoby, A., Marcus, C.M., Lukin, M.D.: Relaxation, dephasing, and quantum control of electron spins in double quantum dots. Phys. Rev. B **76** (2007)
18. Strauchand, F.W., Johnson, P.R., Dragt, A.J., Lobb, C.J., Anderson, J.R., Wellstood, F.C.: Quantum logic gates for coupled superconducting phase qubits. Phys. Rev. Lett. **91** (2003)
19. Cirac, J.I., Zoller, P.: Quantum computations with cold trapped ions. Phys. Rev. Lett. **74**(20) (1995) 4091–4094
20. Deutsch, I., Brennen, G., Jessen, P.: Quantum computing with neutral atoms in an optical lattice. Fortschritte der Physik [Progress of Physics] **48** (2000) 925–943
21. Briegel, H.J., Calarco, T., Jaksch, D., Cirac, J.I., Zoller, P.: Quantum computing with neutral atoms. Journal of Modern Optics **47** (2000) 415–451
22. Knil, E., LaFlamme, R., Milburn, G.J.: A scheme for efficient quantum computation with linear optics. Nature **409** (2001) 46–52
23. Inoue, S.I., Aoyagi, Y.: Design and fabrication of two-dimensional photonic crystals with predetermined nonlinear optical properties. Phys. Rev. Lett. **94** (2005)
24. Lin, C.-C., Chakrabarti, A., Jha, N.K.: FTQLS: Fault-tolerant quantum logic synthesis. IEEE Transactions on Very Large Scale Integration (VLSI) Systems **22**(6) (June 2014) 1350–1363
25. Lin, C.-C., Chakrabarti, A., Jha, N.K.: QLib: Quantum module library. ACM J. Emerg. Technol. Comput. Syst. **11**(1) (2014)
26. Wille, R., Große, D., Teuber, L., Dueck, G.W., Drechsler, R.: RevLib: An online resource for reversible functions and reversible circuits. In: Int. Symposium on Multiple-Valued Logic. (2008) 220–225 RevLib is available at http://www.revlib.org.