Logic Synthesis for Quantum State Generation

Philipp Niemann
Department of Computer Science,
University of Bremen,
D-28359 Bremen, Germany
Email: pniemann@cs.uni-bremen.de

Rhitam Datta
Indian Institute of Engineering
Science & Technology,
Shibpur, India
Email: rhitamdatta4@gmail.com

Robert Wille
Institute for Integrated Circuits,
Johannes Kepler University,
A-4040 Linz, Austria
Email: robert.wille@jku.at

***Abstract*—Quantum computation established itself as a promising emerging technology and, hence, attracted considerable attention in the domain of computer-aided design (CAD). However, quantum mechanical phenomena such as superposition, phase shifts, or entanglement lead to a logic model which poses serious challenges to the development of a proper design flow for quantum circuits. Consequently, researchers addressed synthesis of quantum circuits not as a single design step, but considered sub-tasks such as synthesis of Boolean components or synthesis of restricted subsets of quantum functionality. Generating a particularly desired quantum state is another of these sub-tasks. However, logic synthesis of quantum circuits accomplishing that has hardly been considered yet. In this work, we propose a generic method which automatically synthesizes a quantum circuit generating any desired quantum state from an initially given basis state. The proposed method allows for both, a theoretical determination of upper bounds as well as an experimental evaluation of the number of quantum gates needed for this important design step.**

I. Introduction

Quantum computation [1] established itself as a promising emerging technology for many practically relevant problems such as factorization (and its application in cryptography), database search, graph/algebraic problems, and many more. Driven by these prospects as well as recent accomplishments in the physical realization of corresponding devices (see e.g. [2]), how to design a quantum circuit description realizing the desired functionality attracted considerable attention in the recent years. Originally, researchers considered the respective issues theoretically. This led, amongst others, to the term *quantum algorithm* as a description of several computational steps to be conducted (see e.g. the initial accomplishments of Grover's Algorithm [3], Shor's Algorithm [4], as well as the general investigations on how to describe and realize arbitrary quantum functionality [5]–[11]).

With increasing interest in this domain, these efforts were enriched by investigations towards the development of methods for *computer-aided design* (CAD). Here, researchers consider the underlying problems from a logic design and logic synthesis perspective. The overall goal is to provide methods which, similar to the design and synthesis of conventional circuits, *automatically* generate a circuit description of the desired (quantum) functionality. While this general scheme is perfectly in-line with established design flows for conventional circuits (where electro-technical devices are eventually abstracted and designed on a Boolean logic level), how to synthesize quantum circuits significantly differs from established CAD methods and flows.

In fact, quantum mechanical phenomena such as superposition, phase shifts, or entanglement lead to a logic model which is significantly different to the established logic models available for conventional CAD. In fact, no discrete representations, an infinite space of possible states, or operations working in high-dimensional Hilbert spaces and described by unitary matrices which may include complex numbers pose serious challenges to the development of proper and efficient CAD methods for quantum circuits. Although approaches for the synthesis of arbitrary quantum functionality have been considered in the past (see e.g. [5]–[9]), they lead to a significant amount of gates and additionally rely on arbitrary sets of gates (rather than a dedicated gate library). Hence, in order to efficiently tackle these obstacles, researchers considered synthesis of quantum circuits not as a single design step, but as a separation of concerns. For example, researchers separately considered:

- *Synthesis of Boolean Components:*
  Boolean components usually constitute a large building block in many quantum circuits (e.g. the oracle function in Grover's Algorithm [3] or the modular exponentiation in Shor's Algorithm [4]). Since quantum computations are inherently reversible, these components are usually realized in a two-stage fashion: First, the desired Boolean functionality is realized as a *reversible circuit* (for which various synthesis approaches have been proposed in the past; see [12]–[17]). Then, the resulting reversible circuit is mapped into a cascade of equivalent quantum gates (using mapping schemes as initially introduced in [5] and refined in [18]–[20]).

- *Synthesis of Restricted Quantum Functionality:*
  Many design objectives can be realized without employing the full power of arbitrary quantum operations (e.g. stabilizer circuits for error-correcting codes [21] as well as the realization of applications such as the Greenberger-Horne-Zeilinger experiment [22], quantum teleportation [23], or dense quantum coding [24]). In this case, the consideration of a subset of quantum functionality, e.g. Clifford group operations (see e.g. [25]), is sufficient. Corresponding synthesis methods addressing this particular need have been proposed e.g. in [26].

In this regard, the synthesis of arbitrary quantum states, i.e. the generation of a circuit which realizes a desired quantum state, is another important concern in the design of quantum circuits. In fact, many quantum algorithms inherently assume a particular, pre-defined initial state in order to perform the desired computations. However, the respective physical device may only allow to initialize a limited range of states, but not the desired one. Hardly any CAD method addressing this need has been proposed and evaluated yet.

In this work, we are addressing this gap. We propose a method which automatically synthesizes a quantum circuit generating any desired quantum state. To this end, a three-stage-scheme is employed which iteratively modifies an initially given basis state until the desired target state is obtained. The resulting methodology allows for determining theoretical results on the number of gates needed in order to realize an arbitrary quantum state. Furthermore, the method has been implemented by means of a CAD tool which allows for an experimental evaluation of the quantum circuits for dedicated quantum states. In fact, experiments show that many quantum states can be realized with a significantly smaller number of gates than the upper bound would suggest.

The remainder of the paper is structured as follows: In Section II, the background of quantum computation and quantum circuits is reviewed. Section III introduces the main concepts of the proposed synthesis scheme, while the resulting synthesis algorithm is described in detail in Section IV. A theoretical discussion and experimental evaluation is presented in Section V. Finally, Section VI concludes the paper.

II. Quantum Computation and Circuits

To make this paper self-contained, this section reviews the basics of quantum computation and circuits. The respective descriptions are kept brief; readers wishing an in-depth introduction are referred to the respective literature such as e.g. [1].

A. Quantum Systems and Measurement

Quantum systems are composed of *qubits*. Analogously to conventional bits, a qubit can be in one of the computational *basis states* $|0\rangle$ and $|1\rangle$, but, more generally, also in a so called *superposition* $\alpha_0|0\rangle + \alpha_1|1\rangle$ for complex numbers α_0, α_1 with $|\alpha_0|^2 + |\alpha_1|^2 = 1$. More formally, a qubit can be described by a two-dimensional Hilbert space where its (quantum) state is given by a unit vector $\binom{\alpha_0}{\alpha_1}$ – the so called *state vector*. Accordingly, for larger quantum systems composed of n qubits, there are 2^n basis states $(|0\dots00\rangle, |0\dots01\rangle, \dots, |1\dots11\rangle)$, the system can be in an arbitrary superposition of these states $|\psi\rangle = \sum_{k=0}^{2^n-1} \alpha_k|k\rangle$ with $\sum_{k=0}^{2^n-1} |\alpha_k|^2 = 1$, and the corresponding state vector $(\alpha_k)_{0\leq k\leq 2^n-1}$ has dimension 2^n.

However, due to physical limitations there is no possibility to precisely read-out the state of a quantum system, i.e. to obtain the so called *amplitudes* α_k. In fact, it is only possible to perform a *measurement* which causes the system to collapse to some basis state where the probability for measuring basis state $|k\rangle$ is given by $|\alpha_k|^2$.

Example 1. *Consider the following three different states of a qubit:* $|x_1\rangle = \frac{1}{\sqrt{2}}\binom{1}{1}$, $|x_2\rangle = \frac{1}{\sqrt{2}}\binom{1}{-1}$, $|x_3\rangle = \frac{1}{\sqrt{2}}\binom{i}{i}$. *For all three states, the probability of measuring* $|0\rangle$ *is the same as measuring* $|1\rangle$*:* $|\alpha_0|^2 = |\alpha_1|^2 = \frac{1}{2}$.

In this regard, the amplitudes are often represented in *polar coordinates* by $\alpha_k = p_k \cdot e^{i\theta_k}$, i.e. as a decomposition into the *modulus* $p_k = |\alpha_k| \in [0, 1]$ (determining the probability of measuring the corresponding basis state $|k\rangle$) and the so called *phase* $\theta_k \in [0, 2\pi)$.

Example 2. *Consider again the three qubit states from Example 1. While the modulus is always the same* $(\frac{1}{\sqrt{2}})$*, we observe three different phases:* 0 $(e^{i\cdot 0} = 1)$*,* $\pi/2$ $(e^{i\cdot\pi/2} = i)$*, and* π $(e^{i\cdot\pi} = -1)$*. Moreover,* $|x_1\rangle$ *and* $|x_3\rangle$ *are equal up to a* global phase $(|x_3\rangle = i \cdot |x_1\rangle)$ *which means that they are physically indistinguishable and there is no way to find out which of them is actually present. In contrast,* $|x_2\rangle$ *– though having the same moduli – can in principle be distinguished from these states as shown later on in Example 3.*

Besides superposition and phase shifts, *entanglement* is another powerful feature of quantum systems. In an entangled state, measuring one qubit has an effect on the measurement of other qubits and might even completely determine the measurement result. However, as entanglement is not relevant for the present paper, we will not go into more detail here.

B. Quantum Operations and Circuits

By the postulates of quantum mechanics, the evolution of a quantum system due to a quantum operation can be described by a *unitary transformation matrix*, i.e. an invertible complex-valued matrix whose inverse is given by the adjoint matrix.

Commonly used quantum operations include the *rotation* operations R_x, R_y, R_z (parametrized by a rotation angle θ), the *Hadamard* operation H (setting a qubit into a balanced superposition), as well as the *NOT* operation X which flips the basis states $|0\rangle$ and $|1\rangle$. The corresponding unitary matrices are defined as

$$R_x(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix},\ R_y(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ +\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix},$$

$$R_z(\theta) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix},\ H = \tfrac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},\ X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Example 3. *Applying* H *to the basis state* $|0\rangle = \binom{1}{0}$*, i.e. computing* $H \times |0\rangle$ *yields the state* $|x_1\rangle = \frac{1}{\sqrt{2}}\binom{1}{1}$ *from Ex. 1. Similarly, we obtain that* $H \times |x_1\rangle = |0\rangle$ *and* $H \times |x_2\rangle = |1\rangle$*. This means that we can apply the Hadamard operation in order to distinguish between* $|x_1\rangle$ *and* $|x_2\rangle$ *which is not possible with a direct measurement (c.f. Example 2).*

Besides these operations that work on a single *target qubit*, there are also controlled operations on multiple qubits. The state of the additional *control qubits* determines which operation is performed on the target qubit. More precisely, the operation on the target qubit is executed if and only if the control qubits with a *positive* control are in the $|1\rangle$-state and the ones with a *negative* control are in the $|0\rangle$-state.

Example 4. *An example is the* controlled NOT *(CNOT) operation on two qubits (with a positive control) which applies the NOT operation to the target if the control qubit is in the* $|1\rangle$*-state. On the matrix level, it is defined by*

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Realizations of such elementary quantum operations are represented by *quantum gates* g_i which eventually form a *quantum circuit* $G = g_1 \dots g_d$ with $1 \leq i \leq d$.

Quantum operations and circuits are inherently reversible, i.e. there is an operation/circuit that realizes the inverse transformation. In most cases, inverting a given quantum circuit is a relatively easy task. In fact, most quantum gates are self-inverse (like the Hadamard, NOT, and CNOT gate) or determining the inverse is straight-forward (e.g. for rotations by taking the negated rotation angle). Consequently, the inverse quantum circuit is obtained by (1) reverting the gate order and (2) replacing each gate by its individual inverse.

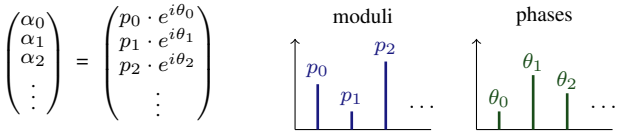
III. General Methodology

In this section, we introduce the general idea of the proposed synthesis approach. Furthermore, we discuss which set of gates (eventually forming a CAD *gate library*) is adequate for this purpose. Based on that, the technical details of the synthesis approach are afterwards provided in Section IV.

A. General Idea

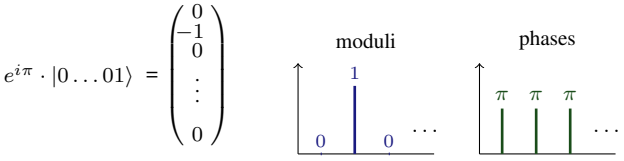
The logic synthesis task considered in this work is to determine a quantum circuit that efficiently generates a desired quantum state $|\tau\rangle$ (denoted as *target state* in the following) which is given as a state vector (α_k) with amplitudes $\alpha_k = p_k \cdot e^{i\theta_k}$ $(0 \leq k \leq 2^n - 1)$. While basis states are easy to generate physically, the target states considered here may be of arbitrary nature and, hence, require explicit quantum operations to be applied. More precisely, target states may inherit various characteristics to be addressed, namely:

- *Superposition*: The target state may have multiple non-zero amplitudes, i.e. its amplitudes may satisfy $|\{k \mid \alpha_k \neq 0\}| > 1$. In contrast, basis states have a



(a) Given target state $|\tau\rangle$. (b) Sketch of its characteristics.

Fig. 1. Visualization of a target state's characteristics.



(a) Given basis state. (b) Sketch of its characteristics.

Fig. 2. Visualization of a basis state's characteristics.

single non-zero amplitude α_j only (with $p_j = 1$), while all other amplitudes α_k with $k \neq j$ are set to 0.

- *Different Probabilities*: The multiple non-zero amplitudes of the target state may have different moduli p_k. Consequently, when measuring a quantum system in this state, there will be (multiple) possible outcomes which occur with different probabilities. In contrast, measuring a basis state always leads to the same outcome.
- *Different Phases*: In addition to the moduli, the amplitudes of a target state may also inherit arbitrarily different phases θ_k which, potentially, can also be measured as outlined in Section II. In contrast, the amplitudes of a basis state all have the same (global) phase which has no physical meaning and cannot directly be measured.

Besides that, the target state may inherit further characteristics such as entanglement which, however, were not found to be relevant for the proposed approach and are, thus, not discussed in more detail in this motivational description.

Example 5. *Fig. 1 visually illustrates the discussed characteristics of a target state to be synthesized. More precisely, Fig. 1(a) gives the state vector description of an arbitrary quantum state $|\tau\rangle$ while Fig. 1(b) sketches the discussed characteristics with respect to the modulus (left-hand side) and the phases (right-hand side) of $|\tau\rangle$'s amplitudes. In contrast, Fig. 2 shows the corresponding characteristics of an exemplary basis state.*

Hence, the goal of logic synthesis is to determine a circuit which takes an easy to generate basis state as input and applies quantum operations (i.e. gates – preferably from a restricted, predefined gate library) until the desired target state with its respective characteristics results. In order to address this problem, we consider the task in an inverse fashion. That means, we take the given target state $|\tau\rangle$ as input and, based on its particular characteristics, determine which quantum operations are to be applied so that a transformation from $|\tau\rangle$ to a basis state is accomplished. By inverting the resulting circuit (which can easily be conducted as reviewed in Section II), the desired quantum circuit transforming a basis state to $|\tau\rangle$ is synthesized.

Following this scheme, a given target state $|\tau\rangle$ is transformed to a basis state as sketched in Fig. 3. More precisely, the following three steps are performed:

1) *Unify phases*, i.e. transform the potentially different phases θ_k of $|\tau\rangle$'s amplitudes to a single (global) phase.
2) *Unify probabilities*, i.e. transform the potentially different moduli p_k of $|\tau\rangle$'s amplitudes to an equal probability distribution.

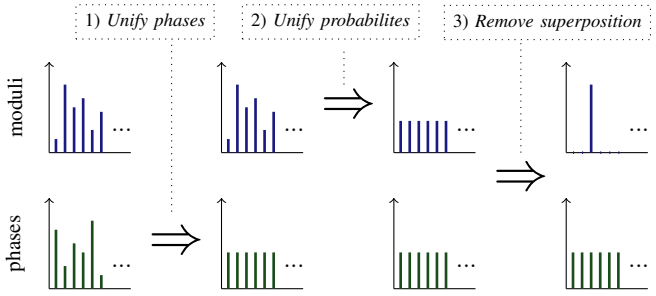


Fig. 3. General methodology of the proposed synthesis approach.

3) *Remove superposition*, i.e. transform $|\tau\rangle$'s unified amplitudes to a state with a single non-zero amplitude and, by this, generate a basis state (potentially with a negligible global phase).

Before we describe the respective steps in detail, we first discuss which set of gates (eventually forming a CAD gate library) might be adequate for these sub-tasks.

B. Applied Gate Library

In order to evaluate which set of gates might be required to conduct the sub-tasks sketched above, we first consider each step with respect to a quantum system composed of a single qubit which is assumed to be in the state $\binom{\alpha}{\beta}$. Then, the proposed transformations can be conducted as follows:

1) Unify Phases: In order to unify the phases of two amplitudes, we can use R_z rotation gates as illustrated by means of Fig. 4(a). In fact, the R_z rotation applies phase shifts, i.e.

$$R_z(\theta) \times \begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \begin{pmatrix}e^{-i\theta/2}\alpha\\ e^{i\theta/2}\ \ \beta\end{pmatrix}.$$

Hence, assuming that $\alpha = a \cdot e^{i\phi}$ and $\beta = b \cdot e^{i\psi}$, a rotation by $\phi - \psi$ gives

$$R_z(\phi-\psi) \times \begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \begin{pmatrix}\tilde{\alpha}\\ \tilde{\beta}\end{pmatrix} = e^{i\kappa}\begin{pmatrix}a\\ b\end{pmatrix}, \text{ where } \kappa = \tfrac{\phi+\psi}{2}.$$

As a result, both amplitudes have the same phase.

2) Unify Probabilities: Given that the amplitudes have the same phase, R_y rotations can be employed to unify their moduli and, hence, their probabilities as sketched in Fig. 4(b). In fact, a R_y rotation combines amplitudes, i.e.

$$R_y(\theta) \times \begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \begin{pmatrix}\cos\frac{\theta}{2}\alpha - \sin\frac{\theta}{2}\beta\\ \sin\frac{\theta}{2}\alpha + \cos\frac{\theta}{2}\beta\end{pmatrix}.$$

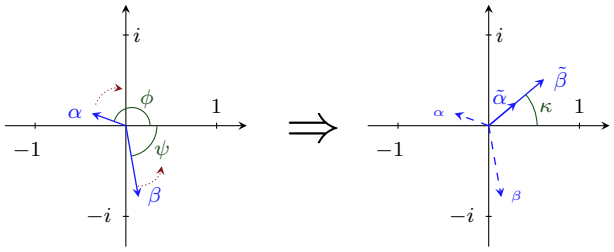
Hence, assuming that $\alpha = a \cdot e^{i\kappa}$ and $\beta = b \cdot e^{i\kappa}$, a rotation angle θ exists such that

$$R_y(\theta) \times \begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \begin{pmatrix}\gamma\\ \gamma\end{pmatrix} = e^{i\kappa}\begin{pmatrix}c\\ c\end{pmatrix} \text{ with } c = \sqrt{\tfrac{a^2+b^2}{2}}.$$

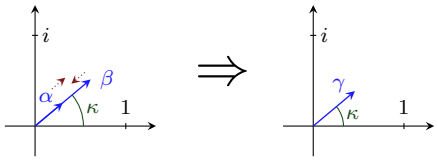
A precise calculation yields that an appropriate angle for this purpose is given by

$$\theta = 2\sin^{-1}\left(\frac{a-b}{\sqrt{2(a^2+b^2)}}\right).$$

These rotations can be applied in order to modify an arbitrary set of amplitudes until all amplitudes are unified, i.e. each amplitude of the state has the same probability.



(a) Unify phases.



(b) Unify probabilites.

Fig. 4. Applying rotation gates.

3) Remove superposition: After the transformations from the previous steps, removing superposition and, by this, determining a basis state can easily be realized by applying Hadamard gates – more precisely by

$$H \times \begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \frac{e^{i\kappa}}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} \times \begin{pmatrix}c\\ c\end{pmatrix} = e^{i\kappa}\begin{pmatrix}1\\ 0\end{pmatrix}.$$

Overall, a gate library composed of R_y and R_z rotation gates as well as Hadamard gates[1] provides a set of operations (and, hence, a gate library) which is sufficient to conduct the sub-tasks sketched above. Besides that, we additionally need CNOT gates in order to handle quantum systems composed of more than a single qubit (described in detail in the next section). Fortunately, all these gates are well-studied and can be either executed directly or implemented with little overhead in many technologies that are actually used for quantum computation like, e.g. quantum dots, ion traps, or superconducting qubits (see e.g. [27]). With this as basis, a detailed description of the resulting overall synthesis approach can be provided.

IV. The Resulting Synthesis Approach

In the previous section, we sketched the general methodology of the proposed synthesis approach and discussed which gate library may be adequate to perform the identified three steps. Based on that, this section first describes on an abstract level which building blocks with a dedicated functionality are composed together in order to transform a given target state into a basis state. Afterwards, the actual realization of those building blocks in terms of elementary quantum gates from the assumed gate library is provided.

A. Generating the Circuit for Arbitrary Quantum States

In Section III-B, we demonstrated for a *pair* of amplitudes how their phases and moduli can be unified using R_z and R_y rotations. However, Step 1 and Step 2 of the overall methodology aim for unifying *all* 2^n amplitudes of the target state. For this purpose, we apply a scheme which successively unifies pairs of amplitudes until all amplitudes have the same phases and moduli.

Figure 5 exemplarily illustrates this scheme for a target state $(a_k)_{0\leq k\leq 7}$ composed of $n = 3$ qubits. More precisely:

- In the first rotation step, we unify adjacent pairs of amplitudes by applying correspondingly chosen R_z and R_y rotations. Appropriate rotation angles can be computed as shown before in Section III-B. After this step, the vector consists of pairs of unified amplitudes $\beta_0, \ldots, \beta_3$.
- Then, in a first swapping step, these equal pairs are split up by permuting the amplitudes. On an abstract level, we interchange $|q_0q_1q_2\rangle$ with $|q_0q_2q_1\rangle$, i.e. we swap qubits q_1 and q_2. For the state vector, this results in interchanging the amplitude corresponding to $|001\rangle$ with the one of $|010\rangle$ and $|101\rangle$ with $|110\rangle$. This gives an interleaved structure of the amplitudes with two blocks β_0, β_1 followed by two blocks β_2, β_3.
- In the next rotation step, we repeat the first step and obtain a vector that consists of quadruples of unified amplitudes γ_0 and γ_1. Note that the upper/lower two pairs are unified using the same set of rotations.
- In another swapping step, we again split up equal pairs by permuting the amplitudes. Now, we swap the qubits q_0 and q_2 which corresponds to interchanging the amplitudes of $|001\rangle$ with the one of $|100\rangle$ and $|011\rangle$ with $|110\rangle$. This again gives an interleaved structure of the amplitudes with several γ_0, γ_1 blocks.
- In the last rotation step, we again repeat the first step and unify adjacent pairs. This finally gives us a state vector with all amplitudes being the same. More precisely, the qubits are in a balanced superposition with amplitude $\delta = \frac{1}{2\sqrt{2}} \cdot e^{i\Theta}$ and some global phase Θ.
- Finally, this balanced superposition is removed by applying Hadamard gates on each qubit (not depicted in Fig. 5), resulting in the basis state $e^{i\Theta}|0\ldots0\rangle$.

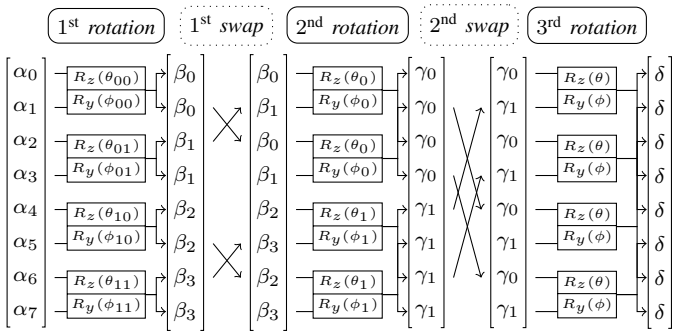


Fig. 5. Construction scheme for $n = 3$ qubits.

For a larger number of qubits $n > 3$, another rotation and swapping step is introduced for each additional qubit such that the rotation steps unify adjacent pairs and the swapping steps restore an interleaved structure by swapping qubits.

Note that the swaps required in the swapping steps can be realized by simply applying three consecutive CNOT gates on the respective qubits (with control and target swapped for the central gate). In some technologies, there are even dedicated SWAP gates for this purpose. Thus, it only remains open how to realize the rotation steps of the approach using gates from the gate library considered in Section III-B. This matter will be discussed next.

B. Realizing a Rotation Block

In order to realize a single rotation step of the scheme sketched above, we have to apply rotations by individual rotation angles to (pairwise) different pairs of amplitudes. However, applying an elementary rotation gate does not only effect a single pair of amplitudes. In fact, the entire state vector will be affected.

Example 6. *Consider a quantum system consisting of two qubits q_0, q_1 being in the state $|\psi\rangle = (\alpha_{00}, \alpha_{01}, \alpha_{10}, \alpha_{11})^T$.*

[1] The Hadamard gate can also be interpreted as a combined rotation (up to global phase), i.e. $H = e^{i\pi/2}R_y(\pi/2) \cdot R_z(\pi)$.

Applying an (uncontrolled) $R_z(\theta)$ gate on qubit q_1 yields

$$\begin{pmatrix} e^{-i\theta} & 0 & 0 & 0 \\ 0 & e^{i\theta} & 0 & 0 \\ 0 & 0 & e^{-i\theta} & 0 \\ 0 & 0 & 0 & e^{i\theta} \end{pmatrix} \times |\psi\rangle = \begin{pmatrix} e^{-i\theta}\alpha_{00} \\ e^{i\theta}\ \alpha_{01} \\ e^{-i\theta}\alpha_{10} \\ e^{i\theta}\ \alpha_{11} \end{pmatrix}$$

meaning that all amplitudes of $|\psi\rangle$ are affected.

Consequently, to address distinct pairs of amplitudes, we need to apply controlled rotations.

Example 7. *Consider the same setting of a quantum system consisting of two qubits q_0, q_1 as in the previous example. Now, applying a controlled-$R_z(\theta)$ gate on qubit q_1 with a negative control on qubit q_0 yields*

$$\begin{pmatrix} e^{-i\theta} & 0 & 0 & 0 \\ 0 & e^{i\theta} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times |\psi\rangle = \begin{pmatrix} e^{-i\theta}\alpha_{00} \\ e^{i\theta}\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix}$$

such that only one particular pair of amplitudes is affected.

Clearly, the higher the number of qubits, the more controls have to be applied for each rotation in order to address one particular pair of amplitudes. In a naive fashion, we can read from Fig. 5 that we need 2^{n-1} controlled rotation gates with $n-1$ controls for R_z and R_y in each rotation step. However, this number only holds for the first step. In fact, in the remaining steps, several pairs of rotations have the same rotation angles and can, thus, be combined to a single rotation with less controls.

Example 8. *Consider again Fig. 5. In the second rotation step the rotation angles of the upper two and lower two pairs of amplitudes are identical. Consequently, the corresponding rotation gates do not need a control on qubit q_1, but only on qubit q_0. In the third/last rotation step, the rotation angle is the same for all pairs of amplitudes. Consequently, no control at all is required in this step.*

In summary, for the k^{th} rotation step we require at most 2^{n-k} controlled rotations with $n-k$ controls (for R_z and R_y rotations). On the first view, this seems to be very expensive, as according to the well-known decomposition presented in [5], each multiple-controlled gate has a cost of at least $2^{n-k}-2$ CNOTs and $2^{n-1}-1$ controlled rotation gates (Lemma 7.1) which, in turn, require one CNOT and two elementary rotations each (Lemma 5.5). However, in our special case we can exploit the fact that similar multiple-controlled gates occur together where each combination of positive/negative controls is present (to address each block of the state vector). This setting is also called a *uniformly* controlled (rotation) gate [7] and a relatively cheap realization of the whole block of multiple-controlled R_z or R_y gates using only 2^k CNOTs and 2^k rotation gates in total (where k is the number of controls) has been presented in [10].

Example 9. *Consider Fig. 6 for a sketch of the construction for $n = 3$ qubits. On the left-hand side we see the four 2-controlled rotations used in the first rotation step of Fig. 5. Horizontal lines denote the three qubits, positive (negative) controls are indicated by a black (white) dot. On the right-hand side we see the implementation in terms of gates from our gate library using 4 CNOTs (●–⊕) and 4 rotation gates.*

V. Discussion and Experimental Evaluation

The synthesis methodology together with the detailed descriptions given in the previous sections provides the basis for (1) a theoretical discussion e.g. on the number of quantum gates needed in order to realize an arbitrary quantum state as well as (2) an implementation of a CAD tool which can be used to experimentally evaluate the actual number of gates needed to generate an explicitly given (practically relevant) quantum state. In this section, we discuss the results obtained from this basis.

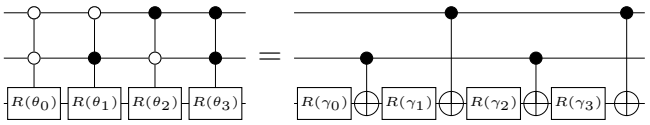


Fig. 6. Uniformly controlled rotation for $n = 3$ qubits.

A. Theoretical Discussion

The proposed approach is a structural synthesis scheme from which an upper bound on the number of gates can be derived. More precisely, we showed in the previous section that the k^{th} rotation step (for an n-qubit quantum state) requires 2^{n-k} CNOTs and 2^{n-k} elementary rotations (for R_z and R_y each). Moreover, we argued that the swaps required in the swapping steps can be realized at a cost of three CNOTs and that the final step requires n Hadamard gates.

Overall, in order to generate an arbitrary quantum state in an n-qubit quantum system, the approach requires at most

- n Hadamard gates to finally remove superposition,
- $3(n-1)$ CNOT gates to realize the swapping steps and $2 \cdot (2^{n-1} + 2^{n-2} + \ldots + 2^1 + 0) = 2^{n+1} - 4$ CNOTs to realize the rotation steps (no CNOT is required in the last step as that particular rotation is uncontrolled, c.f. Example 8), as well as
- $2 \cdot (2^{n-1} + 2^{n-2} + \ldots + 2^1 + 1) = 2^{n+1} - 2$ elementary R_z and R_y rotation gates.

B. Experimental Evaluation

In order to experimentally evaluate the proposed synthesis methodology by means of explicitly given quantum states, the methods and schemes described in Section IV have been implemented in MatLab. Afterwards, we applied different target states to the resulting algorithm and recorded the obtained circuits. To this end, a set of target states has been considered which provides a representative variety – including quantum states needed in practically relevant applications. More precisely, the following target states have been considered:

- A set of states in which both, the phases and the moduli, have been generated randomly (denoted by *random*).
- A set of superposed states (denoted by *superposed*) which are required e.g. as input for Grover's algorithm and, usually, can easily be realized by just applying H gates. This set of states is considered in order to evaluate the automation overhead caused by applying the proposed synthesis methodology.
- A set of superposed states which additionally contain (randomly) chosen phases (denoted by *superposed+phase*). This set of states is similar to the *superposed*-set but inherits characteristics which cannot be handled as easily anymore.
- A set of states that are obtained by applying an arbitrary Clifford group operation to a randomly chosen basis state (denoted by *clifford*). These states have practical applications e.g. for stabilizer circuits or error-correcting codes.
- A set of states which have been derived similarly using Quantum Fourier Transformations (denoted by *qft*). This set of states has applications e.g. in Shor's algorithm.

Table I summarizes the obtained results[2]. The first two columns provide the size (in terms of number of qubits) as

[2]All results have been obtained in negligable run-time (less than a CPU second) on an current machine.

TABLE I
EXPERIMENTAL EVALUATION

#Qubits	Target state	R_z, R_y	CNOT + SWAP	H	Total
2	upper_bound	6	$4 + 3 \cdot 1$	2	15
	random	5	$4 + 3 \cdot 1$	2	14
	superposed	1	$0 + 3 \cdot 0$	2	5
	superposed+phase	3	$4 + 3 \cdot 1$	2	12
	clifford	2	$0 + 3 \cdot 1$	2	7
	qft	2	$0 + 3 \cdot 1$	2	7
3	upper_bound	14	$12 + 3 \cdot 2$	3	35
	random	14	$12 + 3 \cdot 2$	3	35
	superposed	4	$4 + 3 \cdot 0$	3	11
	superposed+phase	5	$12 + 3 \cdot 2$	3	26
	clifford	4	$4 + 3 \cdot 0$	3	11
	qft	3	$0 + 3 \cdot 2$	3	12
5	upper_bound	62	$60 + 3 \cdot 4$	5	139
	random	57	$60 + 3 \cdot 4$	5	134
	superposed	1	$0 + 3 \cdot 0$	5	6
	superposed+phase	27	$30 + 3 \cdot 4$	5	74
	clifford	16	$60 + 3 \cdot 4$	5	93
	qft	5	$0 + 3 \cdot 4$	5	22
10	upper_bound	2046	$2044 + 3 \cdot 9$	10	4127
	random	2046	$2044 + 3 \cdot 9$	10	4127
	superposed	1	$0 + 3 \cdot 0$	10	11
	superposed+phase	1008	$2044 + 3 \cdot 9$	10	3089
	clifford	514	$2044 + 3 \cdot 8$	10	2592
	qft	10	$0 + 3 \cdot 9$	10	47

well as the name (according to the denotations introduced above) of the respectively considered target state. Afterwards, the required number of gates (distinguished between rotation gates, CNOT/SWAP[3] gates, and Hadamard gates) are reported. In order to evaluate the improvement of the resulting CAD tool with respect to the theoretical results available thus far, additionally the upper bound of the required gates is provided for each number of qubits (denoted by *upper_bound*).

The results provide interesting insights: For example, randomly generated target states almost always hit the upper bound. This goes in line with many observations in logic synthesis: While for practical relevant functionality significantly smaller results are often possible, randomly generated functions frequently require worst case efforts. Another interesting observation can be made when considering the set of *superposed*-benchmarks. Here, a simple Hadamard gate applied to each qubit would be sufficient. The proposed approach requires slightly more than that. This nicely illustrates the required automation overhead caused by applying a scheme which is generically applicable to arbitrary quantum states.

But despite these special cases, the results clearly show the advancement of the implemented CAD tool: In many cases, *dedicated* quantum states (including practically relevant ones) can be realized by circuits which are significantly smaller than the upper bound would suggest. To the best of our knowledge, this provides the first evaluation on logic synthesis for quantum state generation.

VI. CONCLUSIONS

In this work, we consider the synthesis of circuits realizing arbitrary quantum states. Together with other tasks, such as the synthesis of Boolean components or the synthesis of restricted subsets of quantum functionality, this constitutes an important step in the computer-aided design of devices for quantum computation. The proposed methodology explicitly takes the characteristics of the quantum states to be realized into consideration and, accordingly, applies operations for their generation. To this end, gates from a gate library are employed for which implementations in many technologies such as quantum dots, ion traps, or superconducting qubits already exist. From the resulting synthesis approach, theoretical bounds on the number of required gates have been derived. An experimental evaluation confirmed that, for many dedicated quantum states, significantly less gates are required.

[3]Note that, as discussed in Section IV, swap gates are either also part of the gate library for many technologies or can easily be realized by simply applying three consecutive CNOT gates on the respective qubits.

ACKNOWLEDGMENTS

This work has partially been supported by the EU COST Action IC1405.

REFERENCES

[1] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*. Cambridge Univ. Press, 2000.
[2] R. V. Meter and M. Oskin, "Architectural implications of quantum computing technologies," *J. Emerg. Technol. Comput. Syst.*, vol. 2, no. 1, pp. 31–63, 2006.
[3] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Theory of computing*, 1996, pp. 212–219.
[4] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," *Foundations of Computer Science*, pp. 124–134, 1994.
[5] A. Barenco, C. H. Bennett, R. Cleve, D. DiVinchenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter, "Elementary gates for quantum computation," *The American Physical Society*, vol. 52, pp. 3457–3467, 1995.
[6] J. J. Vartiainen, M. Mottonen, and M. M. Salomaa, "Efficient decomposition of quantum gates," *Phys. Rev. Lett.*, vol. 92, no. 177902, 2004.
[7] V. Bergholm, J. J. Vartiainen, M. Mottonen, and M. M. Salomaa, "Quantum circuits with uniformly controlled one-qubit gates," *Phys. Rev. A*, vol. 71, no. 052330, 2005.
[8] V. V. Shende, S. S. Bullock, and I. L. Markov, "Synthesis of quantum-logic circuits," *IEEE Trans. on CAD*, vol. 25, no. 6, pp. 1000–1010, 2006.
[9] M. Saeedi, M. Arabzadeh, M. S. Zamani, and M. Sedighi, "Block-based quantum-logic synthesis," *Quantum Information & Computation*, vol. 11, no. 3&4, pp. 262–277, 2011.
[10] M. Möttönen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa, "Transformation of quantum states using uniformly controlled rotations," *Quantum Info. Comput.*, vol. 5, no. 6, pp. 467–473, 2005.
[11] M.-X. Luo, S.-Y. Ma, Y. Deng, and X. Wang, "Deterministic generations of quantum state with no more than six qubits," *Quantum Information Processing*, vol. 14, no. 3, pp. 901–920, 2015.
[12] V. V. Shende, A. K. Prasad, I. L. Markov, and J. P. Hayes, "Synthesis of reversible logic circuits," *IEEE Trans. on CAD*, vol. 22, no. 6, pp. 710–722, 2003.
[13] G. Yang, X. Song, W. N. N. Hung, and M. A. Perkowski, "Fast synthesis of exact minimal reversible circuits using group theory," in *ASP Design Automation Conf.*, 2005, pp. 1002–1005.
[14] M. Saeedi, M. S. Zamani, M. Sedighi, and Z. Sasanian, "Synthesis of reversible circuit using cycle-based approach," *J. Emerg. Technol. Comput. Syst.*, vol. 6, no. 4, 2010.
[15] R. Wille and R. Drechsler, "BDD-based synthesis of reversible logic for large functions," in *Design Automation Conf.*, 2009, pp. 270–275.
[16] R. Wille, E. Schonborn, M. Soeken, and R. Drechsler, "SyReC: A hardware description language for the specification and synthesis of reversible circuits," *Integration, the VLSI Journal*, 2015.
[17] M. Soeken, R. Wille, C. Hilken, N. Przigoda, and R. Drechsler, "Synthesis of reversible circuits with minimal lines for large functions," in *ASP Design Automation Conf.*, 2012, pp. 85–92.
[18] D. M. Miller, R. Wille, and Z. Sasanian, "Elementary quantum gate realizations for multiple-control Toffolli gates," in *Int'l Symp. on Multiple-Valued Logic*, 2011, pp. 288–293.
[19] R. Wille, M. Soeken, C. Otterstedt, and R. Drechsler, "Improving the mapping of reversible circuits to quantum circuits using multiple target lines," in *ASP Design Automation Conf.*, 2013, pp. 85–92.
[20] P. Niemann, S. Basu, A. Chakrabarti, N. K. Jha, and R. Wille, "Synthesis of quantum circuits for dedicated physical machine descriptions," in *Reversible Computation*, 2015, pp. 248–264.
[21] N. D. Mermin, *Quantum Computer Science: An Introduction*. Cambridge University Press, 2007.
[22] D. M. Greenberger, M. A. Horne, and A. Zeilinger, *Bells Theorem, Quantum Theory, and Conceptions of the Universe*. Kluwer Academic Press, 1989.
[23] C. Bennett, G. Brassard, C. Crepeau, R. Jozsa, A. Peres, and W. Wootters, "Teleporting an unknown quantum state by dual classical and EPR channels," *Phys. Rev. Lett.*, vol. 70, no. 1895-1898, 1993.
[24] C. Bennett and S. J. Wiesner, "Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states," *Phys. Rev. Lett.*, vol. 69, no. 2881, 1992.
[25] V. Kliuchnikov, D. Maslov, and M. Mosca, "Asymptotically optimal approximation of single qubit unitaries by Clifford and T circuits using a constant number of ancillary qubits," *CoRR*, vol. abs/1212.0822, 2012.
[26] P. Niemann, R. Wille, and R. Drechsler, "Efficient synthesis of quantum circuits implementing Clifford group operations," in *ASP Design Automation Conf.*, 2014, pp. 483–488.
[27] C.-C. Lin, A. Chakrabarti, and N. K. Jha, "Optimized quantum gate library for various physical machine descriptions," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 11, pp. 2055–2068, 2013.