The Design and Optimization of Quantum Circuits using the Palindrome Transform

Alfred V. Aho[1] * Krysta M. Svore[1] †

[1] *Department of Computer Science, Columbia University.*
1214 Amsterdam Avenue, New York, NY, 10027, USA.

Abstract. The design and optimization of quantum circuits is central to quantum computation. This paper presents a framework for quantum circuit compilation in which the first phase decomposes a unitary matrix into a sequence of $(n-1)$-controlled single-qubit and $(n-1)$-controlled-NOT gates. We present a general algebraic optimization technique, which we call the palindrome transform, that can be used to minimize the number of self-inverting gates in quantum circuits. For a fixed column ordering of two-level decomposition, we give an algorithm for minimal $(n-1)$-controlled-NOT circuit construction, which we call the palindromic optimization algorithm. Our work dramatically reduces the number of gates generated by the conventional method for construction of quantum circuits of $(n-1)$-controlled single-qubit and $(n-1)$-controlled-NOT gates.

Keywords: quantum circuit design, quantum circuit optimization, quantum computer compiler

1 Introduction

The quantum circuit is the primary model for quantum computation [3, 6, 7]. It allows us to represent an algorithm to be implemented by a quantum computer as a sequence of quantum gates. The design and optimization of two-level quantum circuits is part of the first phase of our quantum computer compiler (QCC) project. The ultimate goal of this project is to produce efficient quantum circuits for different quantum technologies from high-level specifications of quantum computations. No matter what quantum computer technology is used, it is important to reduce the size of the quantum circuit in the early phases of compilation since the later phases of quantum circuit compilation can increase the circuit size dramatically for each additional gate in the initial circuit representation [4, 5]. The focus of this paper is the reduction of $(n-1)$-controlled-NOT gates in a quantum circuit of $(n-1)$-controlled single-qubit and $(n-1)$-controlled-NOT gates derived from an arbitrary unitary matrix.

2 Background

In the quantum circuit model, a *quantum gate* on n qubits is a $2^n \times 2^n$ unitary matrix U. A quantum circuit C is a sequence of quantum gates $G_1 \ldots G_k$ where the product of $G_1 \ldots G_k$ represents the unitary operator computed by C.

We use the following gates in our circuits. The *Pauli-X* operator is similar to the classical XOR operation and takes the state $|x\rangle \rightarrow -|x\rangle$. *Control gates* perform the target operation G only if the control qubits are set appropriately. The $(n-1)$-controlled gate $\Lambda^{n-1}(G)$ denotes $n-1$ qubits controlling the application of the operator G to the target qubit. Throughout this paper, G is a single-qubit gate. The $\Lambda^1(X)$ gate is known as the controlled-NOT gate (CNOT) and performs the operation $|x, y\rangle \rightarrow |x, x \oplus y\rangle$. A *two-level* operator indicates non-trivial action on two or fewer vector components.

Two quantum circuits are *equivalent* if the compositions of their respective gates represent the same unitary matrix. That is, if circuit C_1 represents the matrix U_1 and C_2 represents U_2, and if $U_1 = U_2$, then C_1 is equivalent to C_2.

A set of quantum gates is *exactly universal* if it can represent any unitary operation exactly by a sequence of its gates.

The current front end of our quantum circuit compiler generates an exact quantum circuit for an arbitrary unitary matrix U using the universal set of $(n-1)$-controlled single-qubit and $(n-1)$-controlled-NOT gates [2]. The output of the front end is a quantum circuit $Y_1 ... Y_p$ consisting of single-qubit and CNOT gates [4]. The front end uses three phases of decomposition as shown below.

$$U \xrightarrow{\text{Phase I}} V_1 \ldots V_k \xrightarrow{\text{Phase II}} W_1 \ldots W_m \xrightarrow{\text{Phase III}} Y_1 \ldots Y_p$$

Phase I, the focus of this paper, takes a $2^n \times 2^n$ unitary matrix U and outputs a sequence of two-level matrices $V_1 ... V_k = U$, where $k \leq 2^{n-1}(2^n - 1)$, corresponding to an optimized circuit of $\Lambda^{n-1}(G)$ and $\Lambda^{n-1}(X)$ controlled gates. Our framework here builds on the conventional decomposition method described in [1, 6].

Phase II requires two steps to decompose the circuit of controlled operations into a circuit composed of single-qubit and CNOT gates in the model of [1]. The first step decomposes each gate of the optimized circuit from Phase I into sequences of $\Lambda^1(G)$ and $\Lambda^1(X)$ gates. The second step breaks each $\Lambda^1(G)$ gate into single-qubit and CNOT gates.

Phase III optimizes the sequence of single-qubit and CNOT gates for a given objective function. The objective function may be over the depth, size, or elapsed time of the circuit.

3 The Palindrome Transform

We call a gate G *self inverting* if $GG = I$. Note that an $(n-1)$-controlled-NOT gate is self in-

*aho@cs.columbia.edu
†kmsvore@cs.columbia.edu

verting. A sequence is called *self annihilating* if it is a sequence of self-inverting gates of the form $G_1G_2...G_{m-1}G_mG_mG_{m-1}...G_2G_1$. A subsequence of gates of the form $G_1G_2...G_m\beta G_m...G_2G_1$ for $m \geq 0$, where β is null, a single gate, or a sequence of gates in which the first and last gates of β are different is a *partial palindrome*.

If $\alpha = G_1G_2...G_n$, then we use α^R to denote $G_n...G_2G_1$, the reversal of α. Define the *overlap* between two partial palindromes $\alpha_1\beta_1\alpha_1^R$ and $\alpha_2\beta_2\alpha_2^R$ to be the longest suffix γ^R of α_1^R, or equivalently the longest prefix γ of α_2, such that $\gamma^R\gamma$ is a self-annihilating sequence.

Our palindrome transform is a general technique for minimizing the number of self-inverting gates in quantum circuits containing partial palindromes. It takes as input a set of partial palindromes and produces as output a sequence of the partial palindromes. The output sequence maximizes the sum of the lengths of the overlaps between successive pairs of partial palindromes. It uses a data structure called a trie to identify the maximum length common prefixes of all partial palindromes. One efficient way to produce a maximal overlap sequence is to print the labels of the leaves of the trie in a depth-first-search order, which will group those subsequences of partial palindromes which must be juxtaposed to achieve the maximal possible overlaps. The overlaps are then canceled.

Theorem 1 *The palindrome transform produces a sequencing of a set of partial palindromes that maximizes the sum of the lengths of the overlaps between successive partial palindromes.*

4 Palindromic Optimization Algorithm

We now sketch our palindromic circuit generation algorithm that minimizes over fixed-column two-level matrix decompositions the number of $\Lambda^{n-1}(X)$ gates in the generated circuit by maximizing the overlap of $\Lambda^{n-1}(X)$ gates. Our algorithm takes as input a $2^n \times 2^n$ unitary matrix U, where n is the number of qubits, and produces as output a circuit of controlled single-qubit gates that has the fewest number of $\Lambda^{n-1}(X)$ gates. We perform a two-level decomposition of U, which we assume occurs in the column order $0, 1, \ldots, 2^n - 2$, where the columns of the matrix are labeled 0 to $2^n - 1$ [6], and in the Gray code bit flip order $2^0, 2^1, \ldots, 2^{n-1}$. We develop an optimal ordering of state pairs to minimize the number of $\Lambda^{n-1}(X)$ gates. This order of two-level decomposition directs the generation of palindromic subcircuits that compute on the two nontrivial vector components and it produces a sequence $V_1 \ldots V_k$ of two-level matrices such that $V_1 \ldots V_k = U$.

Theorem 2 *For a fixed-column two-level decomposition, the palindromic optimization algorithm produces a circuit that achieves the maximum length of overlaps between successive partial palindromes and thus minimizes the number of $\Lambda^{n-1}(X)$ gates in the quantum circuit of $(n-1)$-controlled single-qubit and $(n-1)$-controlled-NOT gates generated for an arbitrary $2^n \times 2^n$ unitary matrix.*

5 Results and Conclusions

The following equation gives the number of gates in the optimal circuit produced by the palindromic optimization algorithm for an arbitrary $2^n \times 2^n$ unitary matrix:

$$T_n = 2^{n-1}(2^n - 1) + 2(\frac{1}{3}(4^n) - 3(2^{n-1}) + \frac{5}{3}) \qquad (1)$$

Table 1 lists circuit sizes for $n = 2, \ldots, 7$ using our algorithm, the conventional algorithm, and no canceling of palindromes. Our palindromic optimization algorithm results in a dramatic reduction in circuit size.

Table 1: Number of gates in an n-qubit circuit using our algorithm, the conventional method, and the conventional method without canceling palindromes.

n	Palindromic	Conventional	No canceling
2	8	8	10
3	50	62	68
4	246	378	392
5	1086	2034	2064
6	4558	10210	10272
7	18670	49090	49216

We have presented the palindrome transform, a new algorithm for quantum circuit optimization. We have also provided the palindromic optimization algorithm for minimal $(n-1)$-controlled-NOT circuit construction based on the palindrome transform. This algorithm significantly improves the first phase of our framework for quantum circuit compilation. Our ongoing work is focused on developing effective optimization techniques for our quantum computer compiler.

References

[1] A. Barenco, C. Bennett, R. Cleve, D. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter. Elementary gates for quantum computation. *Phys. Rev. A*, 52:3457-3467,1995.

[2] D. Deutsch, A. Barenco, and A. Ekert. Universality in quantum computation. *Proc. R. Soc. London A*, 449(1937):669-677, 1995.

[3] D. Deutsch. Quantum computational networks. *Proc. R. Soc. London A*, 425:73, 1989.

[4] D. DiVincenzo. Two-bit gates are universal for quantum computation. *Phys. Rev. A*, 51(2):1015-1022, 1995.

[5] E. Knill. Approximating quantum circuits. quant-ph/9905086, 1995.

[6] M. A. Nielsen and I. L. Chuang. Quantum Computation and Quantum Information. Cambridge University Press, 2000.

[7] A. Yao. Quantum circuit complexity. In *Proc. of the 34th IEEE Symposium on Foundations of Computer Science*, 1993.