Synthesizing Controlled or Distributed Clifford Circuits

Tuomas Laakkonen
Plasma Science and Fusion Center, Massachusetts Institute of Technology
tsrl@mit.edu

While circuit optimization is a hard problem in general, Clifford circuits are an important building block in many quantum circuits that admit good synthesis algorithms in various settings due to their restricted structure. In this work we consider two tasks:

- Synthesizing controlled versions of Clifford circuits minimizing non-Clifford operations,
- Synthesizing Clifford circuits for distributed quantum architectures minimizing non-local gates.

In both cases, we show that although an arbitrary Clifford circuit may require $O(n^2)$ gates in general, it can always be implemented using only $O(n)$ expensive operations, and that this is asymptotically optimal. Additionally for CNOT circuits, we show in the distributed case that our algorithm is approximately optimal within a factor of three.

1 Introduction

Clifford circuits, which are the quantum circuits comprised solely of the Hadamard gate H, phase gate $S = \sqrt{Z}$, and CNOT gate, are widely used in quantum protocols and algorithms. For example, in quantum teleportation [2], and in quantum error correction and magic state distillation [5]. Remarkably, despite demonstrating the fundamentally quantum properties of both entanglement and superposition, Clifford circuits can be simulated classically in polynomial time via the celebrated Gottesman-Knill theorem. This is because of their restricted structure, which implies that they can be completely described by a set of tensor products of Pauli matrices, known as their stabilizers. While we will not use the stabilizer formalism much in this work, it has a number of useful consequences. In particular we will use the following two facts, that for any Clifford circuit on n qubits,

1. It can be expressed by an equivalent circuit of at most $O(n^2)$ gates, and
2. It can be written in a variety of normal forms [1] [11], comprised of only layers of specific types of gates.

Moreover, due to their importance and relative simplicity (compared to more general quantum circuits [27]), many algorithms have been devised to synthesize Clifford circuits, or subsets such as CNOT circuits, 'optimally' either exactly or heuristically. The definition of optimal depends on the specific situation - previously considered scenarios include minimizing

1. the number of gates [6] or two-qubit gates in Clifford circuits, or their depth [20],
2. the gate count or depth of CNOT circuits in all-to-all connected topologies [22], linear nearest-neighbor topologies [17], or arbitrary topologies [8] [13].
3. the number of Hadamard gates in Clifford circuits [7],
4. the size of CNOT circuits containing arbitrary symbolic unitaries [21], or conversely the number of parameters in a Clifford circuit containing parameterized phase gates [28],
5. the total error of a CNOT circuit when implemented on a NISQ device [10].

Submitted to:
QPL 2025

In this work, we will consider two tasks, the first of which is:

Task 1. *Given a Clifford circuit with unitary U on n qubits, synthesize a controlled unitary $c(U)$ on $n+1$ qubits such that*

$$c(U)(|0\rangle \otimes I_n) = |0\rangle \otimes I_n \qquad c(U)(|1\rangle \otimes I_n) = |1\rangle \otimes U$$

using as few non-Clifford operations as possible.

The motivation for this task is that in many quantum error correction schemes (which are believed to be necessary to achieve large-scale quantum computation), non-Clifford operations are the most expensive operation [18], and so minimizing them is desirable. Controlled circuits are widely used in quantum algorithms such as phase estimation, and the general synthesis problem has received some attention [29]. Controlled Clifford circuit synthesis could form a part of this procedure, but these circuits are also used directly, for instance in variants of the SWAP test [25], or as a subroutine for minimizing non-Clifford gates more generally [15]. The second task is as follows:

Task 2. *Suppose we have a Clifford circuit U on n qubits, where n is even, that are partitioned into two subsets A and B of equal size $\frac{n}{2}$. Synthesize a circuit equivalent to U which minimizes the number of gates that act simultaneously on A and B (which we call* non-local*).*

The motivation for this task is that in realizing large-scale quantum computation, it may be easier to combine many quantum processors with fewer qubits together via shared entanglement, than to build one large quantum processor with the same total number of qubits. This technique is known as distributed quantum computing [9], and the first steps towards this have recently been demonstrated experimentally [19]. In this scenario, the amount of shared entanglement between the processors is a scarce resource, so it ought to be minimized. We model this for the case of two processors of equal size, with the number of non-local gates as a proxy for shared entanglement (since these gates can be implemented using teleportation protocols that consume Bell pairs [23]).

Our results are given in Theorems 2 and 3 respectively. For both tasks, the fact that any Clifford circuit can be expressed with at most $O(n^2)$ gates yields a naive bound of $O(n^2)$. However, we show that in either case only $O(n)$ expensive operations (non-Clifford or non-local gates, respectively) are required, and we give the constant factor in the $O(n)$ explicitly. Furthermore, this is asymptotically optimal, since we construct circuits that require at least $O(n)$ expensive operations. In the case of CNOT circuits, we show in Lemma 5 that Theorem 3 is also approximately optimal, in that for any input circuit, the synthesized circuit will have at most three times the optimal number of non-local gates for this circuit.

2 Preliminaries

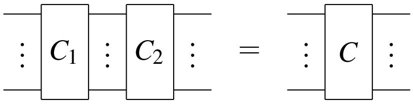
We will call circuits that consist solely of CNOT gates CNOT circuits, and likewise CZ circuits for the $\mathrm{CZ} = (I \otimes H)\mathrm{CNOT}(I \otimes H)$ gate. The most important tool we will use is the parity matrix of a CNOT circuit, see [16, Chapter 4] for a more comprehensive overview. This is defined for any CNOT circuit with unitary U on n qubits as the $n \times n$ matrix A over $\mathbb{F}_2$ such that

$$U|\vec{x}\rangle = |A\vec{x}\rangle$$

for all bitstrings $\vec{x} \in \mathbb{F}_2^n$. If we apply a CNOT gate on qubits i and j, to circuit C on the left, so that $C' = \text{CNOT}_{ij} \cdot C$, the parity matrix changes by adding row i to j:

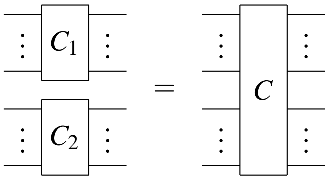
$$A_{C'} = \begin{pmatrix} \ddots & & & & \\ & 1 & \cdots & 0 & \\ & \vdots & & \vdots & \\ & 1 & \cdots & 1 & \\ & & & & \ddots \end{pmatrix} A_C = \begin{pmatrix} & \vdots & \\ (A_C)_{i1} & \cdots & (A_C)_{in} \\ & \vdots & \\ (A_C)_{j1} \oplus (A_C)_{i1} & \cdots & (A_C)_{jn} \oplus (A_C)_{in} \\ & \vdots & \end{pmatrix}$$

Likewise, applying a CNOT gate from the right corresponds to adding column i to column j. Moreover, appending two circuits, corresponds to multiplying both their unitaries and their parity matrices:



$$\iff \quad A_C = A_{C_2} A_{C_1}, \quad U_C = U_{C_2} U_{C_1}$$

Likewise, composing two c circuits in parallel corresponds to the direct sum of their parity matrices, and the tensor product of their unitaries:



$$\iff \quad A_C = A_{C_1} \oplus A_{C_2}, \quad U_C = U_{C_1} \otimes U_{C_2}$$

Thus, all parity matrices are invertible. Given any CNOT circuit, deriving the corresponding parity matrix can be done easily by starting with the identity matrix (corresponding to the empty circuit) and applying these row operations incrementally. In the other direction, a CNOT circuit can be synthesized from any invertible $n \times n$ matrix over $\mathbb{F}_2$ using Gaussian elimination.

For any CNOT circuit with parity matrix A on n qubits, where n is even, which are divided into two equal subsets A and B as in Task 2, we will call this a *partitioned CNOT circuit*, and without loss of generality assume that the first $\frac{n}{2}$ qubits are set A and the remainder are set B. In this case, the matrix A can be partitioned into four quadrants

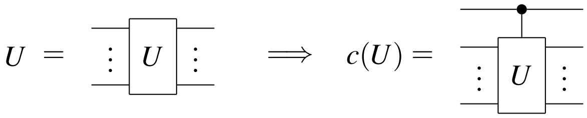
$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

where each submatrix A_{ij} is $\frac{n}{2} \times \frac{n}{2}$. Gates that act on qubits in both A and B simultaneously we will call *non-local*, and gates that act only on A or B individually are *local*. Note that if $A_{12} = A_{21} = 0$ then $A = A_{11} \oplus A_{22}$ so the original circuit does not contain any non-local gates.

Lastly, for any unitary U on n qubits, we will define the controlled unitary $c(U)$ on $n+1$ qubits by

$$c(U)(|0\rangle \otimes I_n) = |0\rangle \otimes I_n \qquad c(U)(|1\rangle \otimes I_n) = |1\rangle \otimes U$$

and it is denoted in circuit notation as



and we have the following identities:

$$c(UV) = c(U)c(V) \qquad c(UVU^{-1}) = (I \otimes U)c(V)(I \otimes U^{-1})$$

3 Controlled Clifford Circuits

To construct controlled Clifford circuits, we will begin by constructing controlled CNOT circuits and CZ circuits, and apply the normal form of [11] to reduce the general case to these two special cases. We will start with CNOT circuits.

Our construction for CNOT circuits is based in a particular matrix decomposition called the *rational canonical form*. This is a decomposition that can be computed for square matrices over any field, using only arithmetic within that field. It is a canonical form in the sense that two matrices A and B have the same rational canonical form if and only if they are similar – that is, $A = SBS^{-1}$ for some invertible S – so that A and B represent the same action in different bases. In particular, we have the following.

Theorem 1 (Rational Canonical Form)**.** *Given an $n \times n$ matrix A over an arbitrary field $\mathbb{F}$, there exists k polynomials $f_1, f_2, \dots, f_k \in \mathbb{F}[x]$, and an invertible matrix $S \in GL_n(\mathbb{F})$ such that*

$$SAS^{-1} = \begin{pmatrix} C_{f_1} & & & \\ & C_{f_2} & & \\ & & \ddots & \\ & & & C_{f_k} \end{pmatrix} \qquad C_{f_i} = \begin{pmatrix} & & & & -a_{i0} \\ 1 & & & & -a_{i1} \\ & 1 & & & -a_{i2} \\ & & \ddots & & \vdots \\ & & & 1 & -a_{id_i} \end{pmatrix}$$

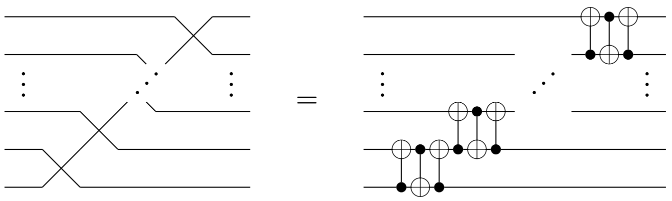
where C_{f_i} is the companion matrix corresponding to $f_i = a_{i0} + a_{i1}x + a_{i2}x^2 + \cdots + a_{i,d_i}x^{d_i} + x^{d_i+1}$. Moreover, we have $f_1 \mid f_2 \mid \cdots \mid f_k$, and each f_i is called an invariant factor *of A.*

Proof. There are many proofs in the literature, for example [14]. See [12] for a construction that can be made practical relatively easily. □

Since each C_{f_i} is sparse, this immediately shows the remarkable fact that *every square matrix is similar to a sparse matrix*. This applies not just for diagonalizable matrices over $\mathbb{C}$, where this is given by the eigendecomposition, but for arbitrary matrices over arbitrary fields. We will exploit this by applying this decomposition to the parity matrix of a CNOT circuit as a matrix over $\mathbb{F}_2$. We will show that the resulting sparse matrix can be synthesized with few CNOTs, and hence synthesize controlled CNOT circuits with few Toffoli gates. Firstly, we consider each C_{f_i} individually:

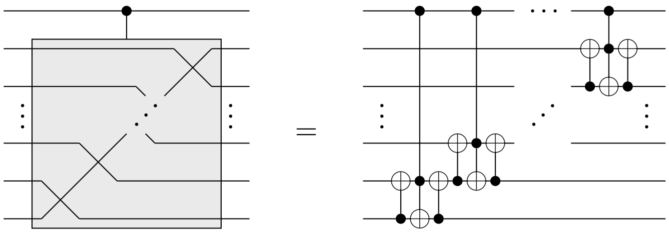
Lemma 1. *For any polynomial $f \in \mathbb{F}_2[x]$ of degree $d \geq 1$ with $f(0) = 1$, the controlled CNOT circuit with parity matrix C_f can be implemented with at most d Toffoli gates (for $f(x) = x^d + 1$ only $d - 1$ are required).*

Proof. First note that if $f(0) = 1$, we must have $a_{i0} = 1$. If $a_{ij} = 0$ for $j > 0$, the circuit with this parity matrix is given by a cyclic shift by one qubit, which can be represented either with SWAP gates, or with triples of CNOT gates:



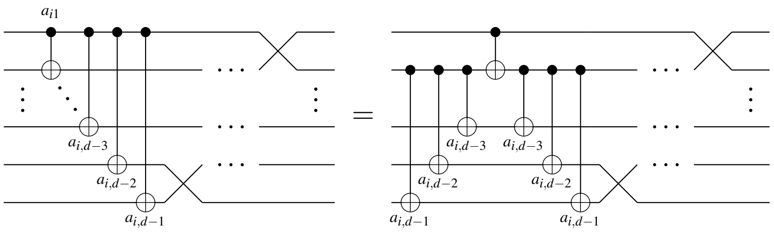
$$\iff \begin{pmatrix} & & & & 1 \\ 1 & & & & 0 \\ & 1 & & & 0 \\ & & \ddots & & \vdots \\ & & & 1 & 0 \end{pmatrix}$$

This can be made controlled by replacing the middle CNOT of each SWAP gate with a Toffoli gate:

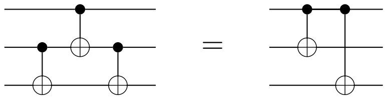


If $f(x) = x^d + 1$, then this is all we need, and only $d-1$ Toffoli gates are required. For the case where $a_{ij} = 1$ for some $j > 0$, we can perform row operations using CNOT gates to copy the one in the top-right corner of the above parity matrix to wherever it is required. Here we write a_{ij} next to each CNOT to indicate that it should be generated only when $a_{ij} = 1$.

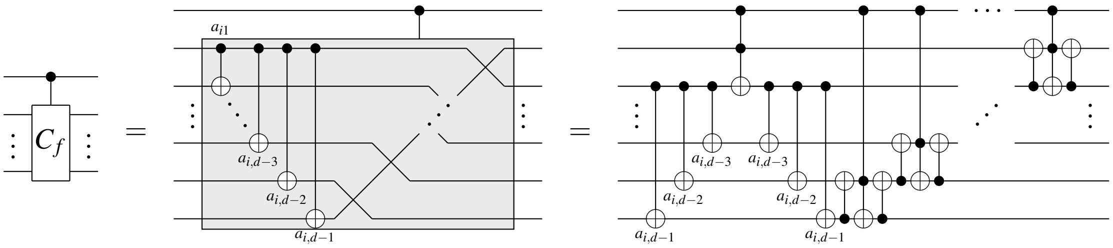
$$\begin{pmatrix} & & & & 1 \\ 1 & & & & a_{i1} \\ & 1 & & & a_{i2} \\ & & \ddots & & \vdots \\ & & & 1 & a_{i,d-1} \end{pmatrix} \iff$$



This can also be written as the circuit on the right, where the middle CNOT targets qubit k such that k is the smallest index where $a_{ik} = 1$ (in this diagram, it is drawn as if $k = 1$, but this is without loss of generality). Clearly such a k must exist since $f(x) \neq x^d + 1$. This follows from repeated application of the following identity:



Thus all of these CNOTs can be controlled simply by controlling the middle CNOT.



Putting this together, we get a circuit for $c(C_f)$ using only d Toffoli gates. □

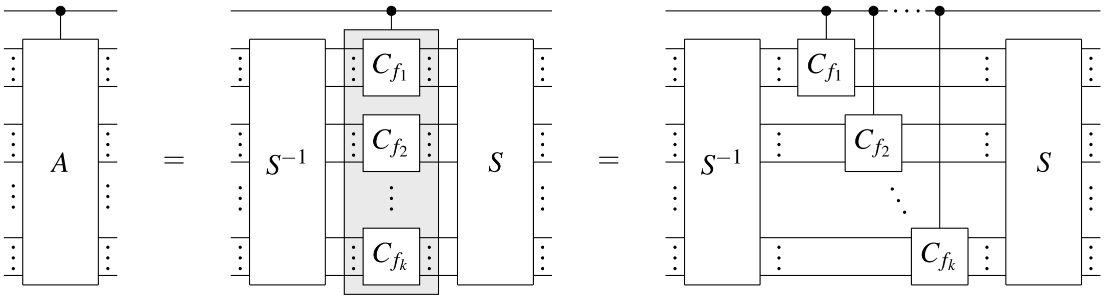
From this, we can construct controlled CNOT circuits in general:

Lemma 2. *The controlled version of any CNOT circuit on n qubits can be implemented using $n-c$ Toffoli gates, where $c \geq 0$ is the number of invariant factors of its parity matrix of the form $x^k + 1$.*

Proof. Let the $n \times n$ parity matrix of the circuit be A. Construct the rational canonical form of A as:

$$A = S(C_{f_1} \oplus C_{f_2} \oplus \cdots \oplus C_{f_k})S^{-1}$$

Using Gaussian elimination, synthesize CNOT circuits for S and S^{-1}. Then the controlled circuit can be constructed as follows, where each $c(C_{f_i})$ is constructed according to Lemma 1. This is possible because each f_i must have $f_i(0) = 1$, otherwise C_{f_i} would have an all-zero row, and the matrix would not be invertible, which is a contradiction since A is a parity matrix and must be invertible.



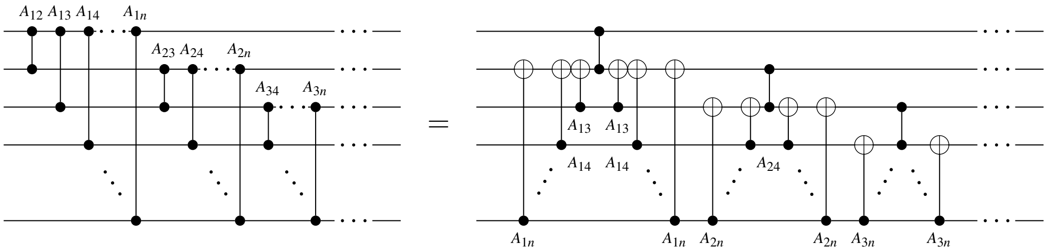
Since $C_{f_1} \oplus \cdots C_{f_k}$ has the same dimensions as A, the sum of the degrees d_i of each f_i is n. Each $c(C(_{fi}))$ requires d_i Toffoli gates, except when $f_i(x) = x^{d_i} + 1$, when only $d_i - 1$ are required. Therefore, the total number of Toffoli gates will be $n - c$ where c is the number of f_i with this form. □

We will show in Theorem 2 that this construction is asymptotically optimal, but it may be that it is also approximately optimal within a constant factor. For each degree-zero invariant factor of the parity matrix A, there is a corresponding linearly independent eigenvector $\vec{x}$ (over $\mathbb{F}_2$, with eigenvalue one), and this can be used to construct a stabilizer of $c(A)|+\rangle$ as $I \otimes \bigotimes_{i=1}^{n} X^{x_i}$. If there are λ such factors, then the number of Toffoli gates required is at most $n - \lambda$. Conversely, this is very similar to the lower bounds provided by the stabilizer nullity technique [3], which shows that the number of Toffoli gates required to synthesize $c(A)|+\rangle$ is at least $\frac{1}{3}(n + 1 - \lambda')$ where λ' is the number of independent stabilizers. If the stabilizers constructed above were the only stabilizers, then this construction would be approximately optimal, but it is not clear whether or not this is true in general. Now we will move on to CZ circuits:

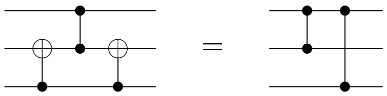
Lemma 3. *The controlled version of any CZ circuit on n qubits can be implemented using at most n Toffoli gates.*

Proof. First, note that all gates in a CZ circuit commute, and that CZ is self-inverse $CZ^2 = I$. Therefore, we only need to consider circuits where there is at most one CZ gate connecting each pair of qubits, and they can be considered in any order. We will organize this information into the *adjacency matrix* of the circuit. Let A be an $n \times n$ symmetric matrix such that $A_{ij} = 1$ when there is a CZ gate acting on qubits i and j, and $A_{ij} = 0$ otherwise.

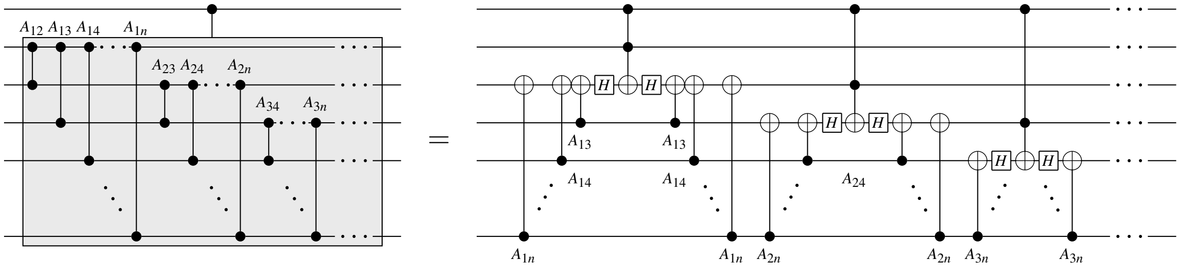
We can reorder the gates so that all the gates acting on the first qubit are first, then all those acting on the second qubit, and so on. This yields a circuit like the following on the left, where we label a gate with A_{ij} if it should be present only when $A_{ij} = 1$.



The circuit on the right follows from assuming that $A_{i,i+1} = 1$, and combining all CZs that act on a common qubit into one, by repeatedly applying the following identity:



If this is not the case, if there is a $j > i$ such that $A_{i,j} = 1$, then this can be swapped with qubit $i+1$, and swapped back afterwards. If there is no such j, then only one CZ is present in that portion of the circuit anyway, so it does not need to be combined.



The the whole circuit can be controlled by controlling each remaining CZ, as shown. There are at most n remaining CZs, since they must each act on a unique qubit, and hence at most n Toffoli gates. □

We can now put these two together to consider general Clifford circuits.

Theorem 2. *The controlled version of any Clifford circuit on n qubits can be implemented using at most* $3n$ *Toffoli gates,* $6n$ *controlled-S gates, and one T-gate. Moreover, there is a controlled CNOT circuit that requires at least* $O(n)$ *non-Clifford operations, so this is asymptotically optimal.*

Proof. We will use the normal form for Clifford circuits proven in [11], where it is shown that any Clifford circuit can be written as a sequence of smaller subcircuits

$$\left[\bigotimes_{i=1}^{n} L_{1,i}\right] \cdot C_1 \cdot C_2 \cdot \left[\bigotimes_{i=1}^{n} H^{h_i}\right] \cdot C_3 \cdot \left[\bigotimes_{i=1}^{n} L_{1,i}\right] = \left[\bigotimes_{i=1}^{n} L_{2,i}\right] \cdot C_1 \cdot C_2 \cdot \left[\bigotimes_{i=1}^{n} H^{h_i}\right] \cdot C_3 \cdot \left[\bigotimes_{i=1}^{n} H^{h_i}\right] \cdot \left[\bigotimes_{i=1}^{n} L_{3,i}\right]$$

where C_1 and C_3 are CZ circuits, C_2 is a CNOT circuit, $L_{k,i}$ are single-qubit Clifford circuits, and we define $L_{3,i} = H^{h_i} L_{2,i}$. Then this circuit can be controlled by controlling $C_{\{1,2,3\}}$, $L_{1,i}$ and $L_{3,i}$. For $C_{\{1,2,3\}}$, this contributes at most $3n$ Toffoli gates total from Lemmas 2 and 3. To handle each $L_{k,i}$ note that by Euler decomposition these can be written as

$$L_{k,i} = e^{i\frac{\pi}{4}w_{k,i}} S^{a_{k,i}} H S^{b_{k,i}} H S^{c_{k,i}}$$

and so their controlled versions can be implemented by at most three controlled-S gates, for a total of $6n$, and a controlled global phase $\frac{\pi}{4}w_{k,i}$. These global phases can be implemented as phase gates on the control qubit, and can be combined into one phase gate $R_Z(\frac{\pi}{4}\sum_{i,k} w_{k,i})$, which can be implemented with at most one T-gate.

Let C be the circuit consisting of the tensor products of CNOT (or equivalently CZ) gates, then $c(C)$ requires at least $O(n)$ non-Clifford operations to implement, so this construction is asymptotically optimal. See Appendix A for the proof. □

4 Distributed Clifford Circuits

We will now switch gears and focus on the second task, synthesizing Clifford circuits for distributed architectures. We will consider the following: there are two distinct sets of qubits $[0, \frac{n}{2})$, and $[\frac{n}{2}, n)$, that model two interconnected quantum processors. We wish to minimize the number of gates that act on both of these sets simultaneously. In particular, we shall consider a model where the only allowed two-qubit gates are the CNOT and the SWAP gate. We consider these to have the same cost, since they can both be implemented using classical communication between the processors and a single shared Bell pair [23]. We start with CNOT circuits.

Lemma 4. *Any partitioned CNOT circuit on n qubits with parity matrix A can be implemented with only* $\min\{\mathrm{rank}(A_{12}), \mathrm{rank}(A_{21})\} + \min\{\mathrm{rank}(A_{12}) + \mathrm{rank}(A_{21}), \frac{n}{2}\} \leq n$ *non-local gates.*

Proof. Let the parity matrix A be partitioned into $\{A_{11}, A_{12}, A_{21}, A_{22}\}$ as in Section 2, where A_{11} and A_{22} are the submatrices local to each processor and A_{21} and A_{12} are the non-local submatrices. We will show how to reduce the parity matrix to the identity using row and column operations, since this can be translated into an equivalent CNOT and SWAP circuit. We will aim to reduce the rank of the A_{12} and A_{21} submatrices to zero, since when this is the case the remaining circuit will be fully local.

First, start by performing Gaussian elimination on both the rows and columns of A_{11}, so that this submatrix is diagonal. Then, perform Gaussian elimination on the rows of A_{21} to bring it to reduced row echelon form. At this point, the form of the matrix will be as follows:

$$\left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right) = \left(\begin{array}{ccccc|c} 1 &&&&& \\ & \ddots &&&& \\ && 1 &&& \cdots \\ &&& 0 && \\ &&&& \ddots & \\ \hline 1 & * & * & * & * & \\ & \ddots & * & * & * & \cdots \\ && 1 & * & * & \end{array}\right)$$

We can now start applying non-local gates to reduce the rank of A_{21}. Note that since the all parity matrices are invertible, the rank of A must be n, and the rank of the (A_{11}, A_{21}) submatrix must be $\frac{n}{2}$, and this is preserved by the local operations we have applied so far. Since A_{11} is diagonal, any all-zero rows must also correspond to all-zero columns. Therefore, because the rank of (A_{11}, A_{21}) is $\frac{n}{2}$, there must be rows in A_{21} that contain pivots for the corresponding columns in A_{21}. For each of these, perform a non-local swap of that row with an all-zero row in A_{11}. Since this replaces a pivot row of A_{21} with an all-zero row, it must decrease the rank of A_{21} by one, and increase the rank of A_{11} by one.

After all this, A_{11} will have full rank, so we can perform local row operations such that for every non-zero row in A_{21}, there is an identical row in A_{11}. Finally, we can perform a non-local CNOT to add these rows from A_{11} to A_{21}, canceling out each row in A_{21} and reducing it to the zero matrix. Local operations can then be used to bring A_{11} to the identity matrix. At this point the form of the matrix is:

$$\left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right) = \left(\begin{array}{ccc|c} 1 &&& \\ & \ddots && \cdots \\ && 1 & \\ \hline &&& \\ & 0 && \cdots \\ &&& \end{array}\right)$$

Reducing A_{12} to the zero matrix is much easier. First, perform local column operations to bring it to column reduced echelon form. Then use local column operations to create copies in A_{11} of every non-zero column in A_{12}, and use non-local column additions to cancel out all columns in A_{12}, thus reducing it to the zero matrix. Each of these non-local operations will reduce the rank of A_{12} by one, and since A_{21} is all zero, it will be unaffected. The remaining matrix can be reduced to the identity with local operations.

The non-local swaps used in reducing A_{21} to zero may increase the rank of A_{12} by at most one each, so the total number of non-local gates used is $\mathrm{rank}(A_{21})$ to reduce A_{21} to zero and at most $\min\{\mathrm{rank}(A_{21})+\mathrm{rank}(A_{12}),\frac{n}{2}\}$ to reduce A_{12} to zero (since the rank of A_{12} can never be more than $\frac{n}{2}$). To achieve the bound given in the lemma statement, note that by exchanging row and column operations, we effectively swap A_{12} and A_{21}, so whichever yields the fewest non-local gates can be used. □

Since this procedure is quite long, an example is helpful to see how it works. We will notate row additions as R_{ij}, row swaps as S_{ij}, column additions as C_{ij}, and column swaps as S'_{ij}. To start, we do row and column reduction on A_{11} and row reduction on A_{21}:

$$\left(\begin{array}{cc|cc} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{array}\right) \xrightarrow{S'_{12}} \left(\begin{array}{cc|cc} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{array}\right) \xrightarrow{S_{34}} \left(\begin{array}{cc|cc} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ \hline 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right)$$

Then, since A_{11} doesn't have full rank, we can see that the second column is all-zeros, and that the first row of A_{21} has a pivot for this column, so we swap it (non-locally) with the all-zero row in A_{11}. This reduces A_{21} to zero:

$$\left(\begin{array}{cc|cc} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ \hline 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right) \xrightarrow{S_{23}} \left(\begin{array}{cc|cc} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{array}\right)$$

Now A_{11} has full rank. We can apply column reduction to A_{12}, and create the same non-zero column in A_{11} with local operations:

$$\left(\begin{array}{cc|cc} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{array}\right) \xrightarrow{S'_{34}} \left(\begin{array}{cc|cc} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{array}\right) \xrightarrow{C_{21}} \left(\begin{array}{cc|cc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{array}\right)$$

Then we can use this column in A_{11} to (non-locally) eliminate the column in A_{12}, reducing it to zero. The final local operations on A_{22} can then be performed:

$$\left(\begin{array}{cc|cc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{array}\right) \xrightarrow{C_{13}} \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{array}\right) \xrightarrow{R_{43}} \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right)$$

Lemma 5. *Any partitioned CNOT circuit with parity matrix A cannot be implemented using only CNOT or SWAP gates with fewer than* $\max\{\mathrm{rank}(A_{12}),\mathrm{rank}(A_{21})\}$ *non-local gates.*

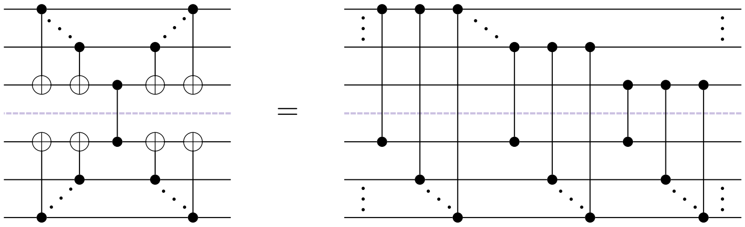
Proof. Any realization of a partitioned CNOT circuit using only CNOT or SWAP gates can be translated to a series of elementary row and column operations that take the parity matrix to the identity matrix. Since any non-local row or column operation can only decrease the rank of the A_{12} and A_{21} submatrices by at most one, at least $\max\{\mathrm{rank}(A_{12}),\mathrm{rank}(A_{21})\}$ operations are required to reduce them both to the all-zero matrix. □

This shows that the construction in Lemma 4 is optimal within a factor of three, since in the worst case we may have $\text{rank}(A_{12}) = \text{rank}(A_{21}) \leq \frac{n}{4}$ and then we will use at most $2\text{rank}(A_{21}) + \text{rank}(A_{12}) = 3 \cdot \max\{\text{rank}(A_{12}), \text{rank}(A_{21})\}$ non-local gates. The last piece before we can tackle general Clifford circuits is CZ circuits, which we shall do by way of the *rank decomposition* of a matrix.

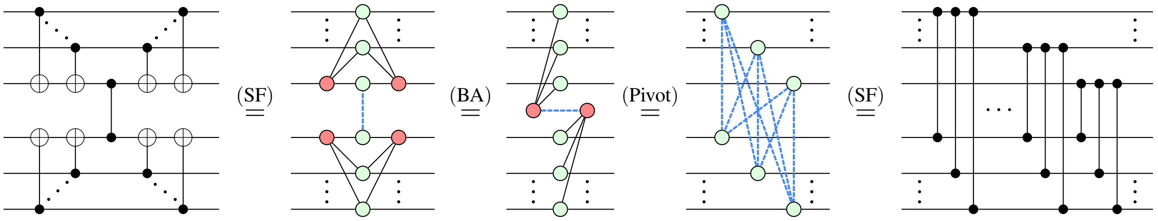
Lemma 6. *Any partitioned CZ circuit on n qubits can be implemented with only* $\text{rank}(A_{12}) \leq \frac{1}{2}n$ *non-local CZs where A is the adjacency matrix of the circuit.*

Proof. As in Lemma 3 we consider CZ circuits where at most one acts between any pair of qubits, without loss of generality, and define the adjacency matrix to be an $n \times n$ matrix A over $\mathbb{F}_2$ such that $A_{ij} = 1$ if and only if there is a CZ acting on qubits i and j. This is partitioned into four quadrants in the same way as for parity matrices, and we will show that the off-diagonal A_{12} and A_{21} submatrices can be implemented using $\text{rank}(A_{12})$ non-local CZs. The rest of A can be synthesized with only local CZs.

Suppose that A_{12} had the form of an outer product $A_{12} = xy^T$ for $x, y \in \mathbb{F}_2^{n/2}$. Then it is possible to implement this using only one non-local CZ, by applying the following identity



where the circuit on the left represents CZs connected from every qubit in the first register to every qubit in the second register. This can be proven via induction on n, or very easily using the ZX-calculus [26]:



We apply this with the targets of the CNOTs on qubits i and j with non-zero values x_i and y_j respectively, and the controls of the CNOTs on every qubit $k \neq i, j$ such that x_k or y_k are non-zero. This implements $A_{12} = xy^T$ as required. The natural generalization of the outer product is to write A_{12} as a sum (over $\mathbb{F}_2$) of r outer products:

$$A_{12} = \sum_{i=1}^{r} x_i y_i^T$$

This is always possible with $r = \text{rank}(A_{12})$, and is known as the *rank factorization* or *rank decomposition* of a matrix [24]. It can be constructed via Gaussian elimination. Since CZs are self inverse, the sum of two adjacency matrices over $\mathbb{F}_2$ corresponds exactly to concatenating the equivalent circuits, and hence any CZ circuit may be implemented with at most $r = \text{rank}(A_{12})$ non-local CZs. □

Now we can combine these results together with a Clifford normal form, as for the controlled case, to synthesize general Clifford circuits minimizing the non-local gates.

Theorem 3. *Any partitioned Clifford circuit on n qubits can be implemented with at most $2n$ non-local gates. Moreover, there is such a circuit that requires at least $\frac{1}{2}n$ non-local CNOT or SWAP gates, so this is asymptotically optimal.*

Proof. The normal form given in [11] shows that any Clifford circuit can be written using as the product of two CZ circuits and a CNOT circuit, interleaved with single-qubit Clifford gates. Since single-qubit operations are inherently local, the only non-local gates come from the CNOT and CZ circuits, which can be optimized using Lemmas 4 and 6 respectively for a total of $2n$ non-local gates. The circuit that swaps the first $\frac{n}{2}$ qubits with the last $\frac{n}{2}$ qubits cannot be implemented with fewer SWAP or CNOT gates, so this construction must also be asymptotically optimal. □

5 Conclusion

In this work we have developed heuristic methods for solving two tasks: synthesizing controlled Clifford circuits with minimal non-Clifford gates, and synthesizing Clifford circuits for a specific distributed quantum computing architecture in a way that minimizes non-local gates. In both cases, we show that only $O(n)$ expensive operations are required for circuits on n qubits, which is asymptotically better than the naive bound of $O(n^2)$ and is asymptotically optimal.

For future work, it may be useful to integrate these procedures into more general methods for synthesizing controlled or distributed quantum circuits. This may be useful not just for implementations on quantum computers, but also for classical simulations: for instance, tree tensor network simulation methods are much slower at processing non-local gates, and the runtime of stabilizer decomposition methods scales exponentially with the number of non-Clifford operations. Finally, for the distributed case we showed that our method is also approximately optimal (within a factor of three), but as discussed in Section 3, further investigation is required to determine if this may also be true for the controlled case. This may require generalizing the construction to use the generalized Jordan decomposition [4, Theorem 5.4] rather than the rational canonical form.

Acknowledgements

Thanks to John van de Wetering for advice on lower bounds for non-Clifford operations, and to Pablo Andrés-Martínez for discussions about tree tensor network simulation methods. TL acknowledges support from U.S. Department of Energy Grant No. DE-SC0020264.

References

[1] Scott Aaronson & Daniel Gottesman (2004): *Improved simulation of stabilizer circuits.* Physical Review A 70(5), doi:10.1103/physreva.70.052328.

[2] Charles H. Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres & William K. Wootters (1993): *Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels.* Phys. Rev. Lett. 70, pp. 1895–1899, doi:10.1103/PhysRevLett.70.1895.

[3] Michael Beverland, Earl Campbell, Mark Howard & Vadym Kliuchnikov (2020): *Lower bounds on the non-Clifford resources for quantum computations.* Quantum Science and Technology 5(3), p. 035009, doi:10.1088/2058-9565/ab8963.

[4] P. B. Bhattacharya, S. K. Jain & S. R. Nagpaul (1994): *Basic Abstract Algebra*, 2nd ed. edition. Cambridge University Press, New York.

[5] Sergey Bravyi & Alexei Kitaev (2005): *Universal quantum computation with ideal Clifford gates and noisy ancillas.* Phys. Rev. A 71, p. 022316, doi:10.1103/PhysRevA.71.022316.

[6] Sergey Bravyi, Joseph A. Latone & Dmitri Maslov (2022): *6-qubit optimal Clifford circuits.* npj Quantum Information 8(1), doi:10.1038/s41534-022-00583-7.

[7] Sergey Bravyi & Dmitri Maslov (2021): *Hadamard-Free Circuits Expose the Structure of the Clifford Group*. IEEE Transactions on Information Theory 67(7), p. 4546–4563, doi:10.1109/tit.2021.3081415.

[8] Timothée Goubault de Brugière, Marc Baboulin, Benoît Valiron, Simon Martiel & Cyril Allouche (2022): *Decoding techniques applied to the compilation of CNOT circuits for NISQ architectures*. Science of Computer Programming 214, p. 102726, doi:10.1016/j.scico.2021.102726.

[9] Marcello Caleffi, Michele Amoretti, Davide Ferrari, Jessica Illiano, Antonio Manzalini & Angela Sara Cacciapuoti (2024): *Distributed quantum computing: A survey*. Computer Networks 254, p. 110672, doi:10.1016/j.comnet.2024.110672.

[10] Marc G. Davis, Ethan Smith, Ana Tudor, Koushik Sen, Irfan Siddiqi & Costin Iancu (2020): *Towards Optimal Topology Aware Quantum Circuit Synthesis* . In: 2020 IEEE International Conference on Quantum Computing and Engineering (QCE), IEEE Computer Society, Los Alamitos, CA, USA, pp. 223–234, doi:10.1109/QCE49297.2020.00036.

[11] Ross Duncan, Aleks Kissinger, Simon Perdrix & John van de Wetering (2020): *Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus*. Quantum 4, p. 279, doi:10.22331/q-2020-06-04-279.

[12] Meinolf Geck (2020): *On Jacob's construction of the rational canonical form of a matrix*. The Electronic Journal of Linear Algebra 36(36), p. 177–182, doi:10.13001/ela.2020.5055.

[13] Arianne Meijer-van de Griend & Sarah Meng Li (2023): *Dynamic Qubit Routing with CNOT Circuit Synthesis for Quantum Compilation*. Electronic Proceedings in Theoretical Computer Science 394, p. 363–399, doi:10.4204/eptcs.394.18.

[14] Robert E. Hartwig (1996): *Roth's Removal Rule and the Rational Canonical Form*. The American Mathematical Monthly 103(4), pp. 332–335.

[15] Luke Heyfron & Earl T. Campbell (2018): *An Efficient Quantum Compiler that reduces T count*. arXiv:1712.01557.

[16] Aleks Kissinger & John van de Wetering (2024): *Picturing Quantum Software: An Introduction to the ZX-Calculus and Quantum Compilation*. Preprint.

[17] Samuel A. Kutin, David Petrie Moulton & Lawren M. Smithline (2007): *Computation at a distance*. arXiv:quant-ph/0701194.

[18] Daniel Litinski (2019): *A Game of Surface Codes: Large-Scale Quantum Computing with Lattice Surgery*. Quantum 3, p. 128, doi:10.22331/q-2019-03-05-128.

[19] D. Main, P. Drmota, D. P. Nadlinger, E. M. Ainley, A. Agrawal, B. C. Nichol, R. Srinivas, G. Araneda & D. M. Lucas (2025): *Distributed quantum computing across an optical network link*. Nature 638(8050), p. 383–388, doi:10.1038/s41586-024-08404-x.

[20] Dmitri Maslov & Ben Zindorf (2022): *Depth Optimization of CZ, CNOT, and Clifford Circuits*. IEEE Transactions on Quantum Engineering 3, p. 1–8, doi:10.1109/tqe.2022.3180900.

[21] Ewan Murphy & Aleks Kissinger (2023): *Global Synthesis of CNOT Circuits with Holes*. Electronic Proceedings in Theoretical Computer Science 384, p. 75–88, doi:10.4204/eptcs.384.5.

[22] K. N. Patel, I. L. Markov & J. P. Hayes (2003): *Efficient Synthesis of Linear Reversible Circuits*. arXiv:quant-ph/0302002.

[23] Christophe Piveteau & David Sutter (2024): *Circuit Knitting With Classical Communication*. IEEE Transactions on Information Theory 70(4), p. 2734–2745, doi:10.1109/tit.2023.3310797.

[24] R. Piziak & P. L. Odell (1999): *Full Rank Factorization of Matrices*. Mathematics Magazine 72(3), pp. 193–201.

[25] Yihui Quek, Eneet Kaur & Mark M. Wilde (2024): *Multivariate trace estimation in constant quantum depth*. Quantum 8, p. 1220, doi:10.22331/q-2024-01-10-1220.

[26] John van de Wetering (2020): *ZX-calculus for the working quantum computer scientist*. arXiv:2012.13966.

[27] John van de Wetering & Matt Amy (2024): *Optimising quantum circuits is generally hard*. arXiv:2310.05958.

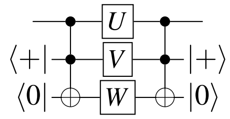
[28] John van de Wetering, Richie Yeung, Tuomas Laakkonen & Aleks Kissinger (2024): *Optimal compilation of parametrised quantum circuits*. arXiv:2401.12877.

[29] Pei Yuan & Shengyu Zhang (2023): *Optimal (controlled) quantum state preparation and improved unitary synthesis by quantum circuits with any number of ancillary qubits*. Quantum 7, p. 956, doi:10.22331/q-2023-03-20-956.

A Proof that Theorem 2 is Asymptotically Optimal

Let $U \in \{I, X, Y, Z, |0\rangle\langle 0|, |1\rangle\langle 1|\}$ be a 2×2 matrix, and $V, W \in \{I, X, Y, Z\}$ be Pauli matrices. Define the linear map M from U, V, W to 2×2 matrices by

$$M(U,V,W) \;=$$



and note the following facts, which can be verified by brute-force calculation of the matrices:

Fact 1 We have $M(U,V,W) = cU'$ where $U' \in \{I, X, Y, Z, |0\rangle\langle 0|, |1\rangle\langle 1|\}$ and $|c| \leq 1$.

Fact 2 If $U \in \{|0\rangle\langle 0|, |1\rangle\langle 1|\}$ then $U' \in \{|0\rangle\langle 0|, |1\rangle\langle 1|\}$.

Fact 3 If $U' \in \{I, X, Y, Z\}$, then $|c| = 1$ if and only if $V = W = I$ and $U = Z$ or $U = I$. In these cases, we have $c = 1$ and $U' = U$.

Now consider the following state defined on $2n+1$ qubits:

$$|\psi_n\rangle = c\left(\prod_{i=1}^{n} \mathrm{CNOT}_{2i,2i+1}\right)\left[|+\rangle \otimes (|+\rangle \otimes |0\rangle)^{\otimes n}\right]$$

Let $P = \otimes_{i=1}^{2n+1} P_i$ with $P_1 \in \{I, X, Y, Z, |0\rangle\langle 0|, |1\rangle\langle 1|\}$, and $P_i \in \{I, X, Y, Z\}$ for $i > 1$. Then we have that

$$\langle \psi_n|P|\psi_n\rangle = \langle \psi_{n-1}|\left[M(P_1,P_2,P_3) \otimes \bigotimes_{i=4}^{2n+1} P_i\right]|\psi_{n-1}\rangle$$

and so we can define the sequence U_i by

$$U_1 = P_1, \qquad U_{i+1} = M(U_i, P_{2i}, P_{2i+1}) = c_i U_i', \qquad \text{where } U_i' \in \{I, X, Y, Z, |0\rangle\langle 0|, |1\rangle\langle 1|\}$$

and we have that

$$\langle \psi_n|P|\psi_n\rangle = \langle \psi_0|U_n|\psi_0\rangle = \langle +|U_n|+\rangle = \langle +|U_n'|+\rangle\left(\prod_{i=1}^{n} c_i\right)$$

by linearity of M. Now suppose that $n > 0$, $P_1 \in \{I, X, Y, Z\}$, and P is a stabilizer of $|\psi_n\rangle$. Then we must have $\langle \psi_n|P|\psi_n\rangle = \langle +|U_n'|+\rangle(\prod_{i=1}^{n} c_i) = 1$. In order for this to be true, we must have that $|c_i| = 1$ for all i, and that $U_n' \in \{I, X\}$ (otherwise we would have $|\langle +|U_n'|+\rangle| < 1$). From Fact 2, we must have $U_i' \notin \{|0\rangle\langle 0|, |1\rangle\langle 1|\}$ for all i. From Fact 3, we must have $U_i' = I$ for all i, since if $U_i' \notin \{I, Z\}$ for some i then $c_i \leq 1$, and if $U_i' = Z$ for some i then either $U_{i+1}' = Z$ or $|c_{i+1}| \leq 1$.

Therefore, the only stabilizer of $|\psi_n\rangle$ is $P = I^{\otimes 2n+1}$. In [3], is is shown that the number of Toffoli gates required to synthesize a state $|\phi\rangle$ on n qubits is lower-bounded by

$$\frac{\nu(|\phi\rangle)}{3} = \frac{n - \log_2 |\mathrm{Stab}|\phi\rangle|}{3}$$

where $\mathrm{Stab}|\phi\rangle$ is the set of all stabilizers of $|\phi\rangle$, and the quantity $\nu(|\phi\rangle)$ is called the *stabilizer nullity*. Applying this to $|\psi_n\rangle$, since $I^{\otimes 2n+1}$ is the only stabilizer, we have $\log_2|\mathrm{Stab}|\psi_n\rangle| = 0$, and so the number of Toffoli gates required to synthesize it is at least $\frac{2n+1}{3}$ if $n > 0$. Conversely, the construction in Lemma 2 requires at most n Toffoli gates, so it is asymptotically optimal. Moreover, since

$$c\left(\prod_{i=1}^{n}\mathrm{CNOT}_{2i,2i+1}\right) = \left[\prod_{i=1}^{n}H_{2i+1}\right] c\left(\prod_{i=1}^{n}CZ_{2i,2i+1}\right)\left[\prod_{i=1}^{n}H_{2i+1}\right]$$

this also implies that Lemma 3 is asymptotically optimal.