

Improved Synthesis of Toffoli-Hadamard Circuits

Matthew Amy¹, Andrew N. Glaudell², Sarah Meng Li³, and Neil J. Ross⁴

¹ School of Computing Science, Simon Fraser University

matt_amy@sfu.ca

² Photonic Inc.

andrewglaudell@gmail.com

³ Institute for Quantum Computing, University of Waterloo

sarah.li@uwaterloo.ca

⁴ Department of Mathematics and Statistics, Dalhousie University

neil.jr.ross@dal.ca

Abstract. The matrices that can be exactly represented by a circuit over the Toffoli-Hadamard gate set are the orthogonal matrices of the form $M/\sqrt{2}^k$, where M is an integer matrix and k is a nonnegative integer. The exact synthesis problem for this gate set is the problem of constructing a circuit for a given such matrix. Existing methods produce circuits consisting of $O(2^n \log(n)k)$ gates, where n is the dimension of the matrix. In this paper, we provide two improved synthesis methods. First, we show that a technique introduced by Kliuchnikov in 2013 for Clifford+ T circuits can be straightforwardly adapted to Toffoli-Hadamard circuits, reducing the complexity of the synthesized circuit from $O(2^n \log(n)k)$ to $O(n^2 \log(n)k)$. Then, we present an alternative synthesis method of similarly improved cost, but whose application is restricted to circuits on no more than three qubits. Our results also apply to orthogonal matrices over the dyadic fractions, which correspond to circuits using the 2-qubit gate $H \otimes H$, rather than the usual single-qubit Hadamard gate H .

Keywords: Quantum circuits · Exact synthesis · Toffoli-Hadamard

1 Introduction

Recent experimental progress has made it possible to carry out large computational tasks on quantum computers faster than on state-of-the-art classical supercomputers [3,26]. However, qubits are incredibly sensitive to decoherence, which leads to the degradation of quantum information. Moreover, physical gates implemented on real quantum devices have poor gate fidelity, so that every additional gate in a circuit introduces a small error to the computation. To harness the full power of quantum computing, it is therefore crucial to design resource-efficient compilation techniques.

Over the past decade, researchers have taken advantage of a correspondence between quantum circuits and matrices of elements from algebraic number rings [2,7,10,14,15]. This number-theoretic perspective can reveal important properties of gate sets and has resulted in several improved synthesis protocols. An important instance of this correspondence occurs in the study of the Toffoli-Hadamard gate set $\{X, CX, CCX, H\}$ [2,9]. Circuits over this gate set correspond exactly to orthogonal matrices of the form $M/\sqrt{2}^k$, where M is an integer matrix and k is a nonnegative integer. A closely related instance of this correspondence arises with the gate set $\{X, CX, CCX, H \otimes H\}$, where the 2-qubit gate $H \otimes H$ replaces the usual single-qubit Hadamard gate H . Circuits over this second gate set correspond exactly to orthogonal matrices over the ring of dyadic fractions $\mathbb{Z}[1/2]$ [2]. The Toffoli-Hadamard gate set is arguably the simplest universal gate set for quantum computation [1,22]; the corresponding circuits have been studied in the context of diagrammatic calculi [23], path-sums [24], and quantum logic [5], and play a critical role in quantum error correction [4,6,20], fault-tolerant quantum computing [11,19,25], and the quantum Fourier transform [17].

In this paper, we leverage the number-theoretic structure of the aforementioned circuits to design improved synthesis algorithms. Our approach is to focus on the matrix groups associated with the gate sets $\{X, CX, CCX, H\}$ and $\{X, CX, CCX, H \otimes H\}$. For each group, we use a convenient set of generators and study the factorization of group elements into products of these generators. Because

each generator can be expressed as a short circuit, a good solution to this factorization problem yields a good synthesis algorithm.

Exact synthesis algorithms for Toffoli-Hadamard circuits were introduced in [9] and independently in [2]. We refer to the algorithm of [2], which we take as our baseline, as the *local synthesis algorithm* because it factors the input matrix column by column. This algorithm produces circuits of $O(2^n \log(n)k)$ gates, where n is the dimension of the input matrix.

We propose two improved synthesis methods. The first, which we call the *Householder synthesis algorithm*, is an adaptation to the Toffoli-Hadamard gate set of the technique introduced by Kliuchnikov in [13] for the Clifford+ T gate set. This algorithm proceeds by embedding the input matrix in a larger one, and then expressing this larger matrix as a product of Householder reflections. The Householder synthesis algorithm produces circuits of $O(n^2 \log(n)k)$ gates. We then introduce a *global synthesis algorithm*, inspired by the work of Russell in [21] and of Niemann, Wille, and Drechsler in [18]. In contrast to the local synthesis algorithm, which proceeds one column at a time, the global algorithm considers the input matrix in its entirety. In its current form, this last algorithm is restricted to matrices of dimensions 2, 4, and 8. As a result of this restriction, the dimension of the input matrix can be dropped in the asymptotic analysis, and the circuits produced by the global algorithm consist of $O(k)$ gates.

The rest of this paper is organized as follows. In [Section 2](#), we introduce the exact synthesis problem, as well as the matrices, groups, and rings that will be used throughout the paper. In [Section 3](#), we review the local synthesis algorithm of [2]. The Householder synthesis algorithm and the global synthesis algorithm are discussed in [Sections 4](#) and [5](#), respectively. We conclude in [Section 6](#).

2 The Exact Synthesis Problem

In this section, we introduce the *exact synthesis problem*. We start by defining the matrices, groups, and rings that will be used in the rest of the paper.

Definition 1. *The ring of dyadic fractions is defined as $\mathbb{Z}[1/2] = \{u/2^k; u \in \mathbb{Z}, k \in \mathbb{N}\}$.*

Definition 2. *$\mathcal{O}_n(\mathbb{Z}[1/2])$ is the group of n -dimensional orthogonal dyadic matrices. It consists of the $n \times n$ orthogonal matrices of the form $M/2^k$, where M is an integer matrix and k is a nonnegative integer. For brevity, we denote this group by \mathcal{O}_n .*

Definition 3. *\mathcal{L}_n is the group of n -dimensional orthogonal scaled dyadic matrices. It consists of the $n \times n$ orthogonal matrices of the form $M/\sqrt{2}^k$, where M is an integer matrix and k is a nonnegative integer.*

\mathcal{O}_n is infinite if and only if $n \geq 5$. Moreover, \mathcal{O}_n is a subgroup of \mathcal{L}_n . When n is odd, we in fact have $\mathcal{O}_n = \mathcal{L}_n$ [2, Lemma 5.9]. When n is even, \mathcal{O}_n is a subgroup of \mathcal{L}_n of index 2. As a result, it is also the case that \mathcal{L}_n is infinite if and only if $n \geq 5$.

Definition 4. *Let $t \in \mathbb{Z}[1/2]$. A natural number k is a denominator exponent of t if $2^k t \in \mathbb{Z}$. The least such k is called the least denominator exponent of t , and is denoted by $\text{lde}(t)$.*

Definition 5. *Let $t = u/\sqrt{2}^k$, where $u \in \mathbb{Z}$ and $k \in \mathbb{N}$. A natural number k is a scaled denominator exponent of t if $\sqrt{2}^k t \in \mathbb{Z}$. The least such k is called the least scaled denominator exponent of t , and is denoted by $\text{lde}_{\sqrt{2}}(t)$.*

We extend [Definitions 4](#) and [5](#) to matrices with appropriate entries as follows. A natural number k is a (scaled) denominator exponent of a matrix M if it is a (scaled) denominator exponent of all of the entries of M . Similarly, the least such k is called the least (scaled) denominator exponent of M . We denote the least denominator exponent of M and the least scaled denominator of M by $\text{lde}(M)$ and $\text{lde}_{\sqrt{2}}(M)$, respectively.

We now leverage some well-known quantum gates to define generators for \mathcal{O}_n and \mathcal{L}_n .

Definition 6. The matrices (-1) , X , CX , CCX , H , and K are defined as follows:

$$(-1) = [-1], \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

$CX = \text{diag}(I_2, X)$, $CCX = \text{diag}(I_6, X)$, and $K = H \otimes H$.

The matrix (-1) is a scalar. The matrices X , CX , and CCX are known as the *NOT*, *CNOT*, and *Toffoli* gates, respectively, while the matrix H is the *Hadamard* gate. In Definition 6, CX and CCX are defined as block matrices, while K is defined as the twofold tensor product of H with itself. Below we explicitly write out matrices for CX , CCX , and K :

$$CX = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad CCX = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad K = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

Definition 7. Let M be an $m \times m$ matrix, $m \leq n$, and $0 \leq a_0 < \dots < a_{m-1} < n$. The m -level matrix of type M is the $n \times n$ matrix $M_{[a_0, \dots, a_{m-1}]}$ defined by

$$M_{[a_0, \dots, a_{m-1}]_{i,j}} = \begin{cases} M_{i',j'} & \text{if } i = a_{i'} \text{ and } j = a_{j'} \\ I_{i,j} & \text{otherwise.} \end{cases}$$

The dimension n of the matrix $M_{[a_0, \dots, a_{m-1}]}$ is left implicit most of the time, as it can often be inferred from the context. As an example, the 2-level matrix $H_{[0,2]}$ of dimension 4 is

$$H_{[0,2]} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & \sqrt{2} & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & \sqrt{2} \end{bmatrix}.$$

Definition 8. The set \mathcal{G}_n of n -dimensional generators of \mathcal{O}_n is the subset of \mathcal{O}_n defined as

$$\mathcal{G}_n = \{(-1)_{[a]}, X_{[a,b]}, K_{[a,b,c,d]} ; 0 \leq a < b < c < d < n\}.$$

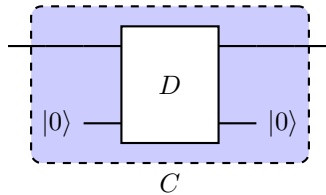
Definition 9. The set \mathcal{F}_n of n -dimensional generators of \mathcal{L}_n is the subset of \mathcal{L}_n defined as $\mathcal{F}_n = \mathcal{G}_n$ when n is odd and as

$$\mathcal{F}_n = \{(-1)_{[a]}, X_{[a,b]}, K_{[a,b,c,d]}, I_{n/2} \otimes H ; 0 \leq a < b < c < d < n\}$$

when n is even.

In Definition 9, the condition on the parity of n ensures that $I_{n/2} \otimes H$ is only included when it is meaningful to do so. In what follows, for brevity, we ignore the subscript in $I_{n/2} \otimes H$ and simply write $I \otimes H$. It is known that \mathcal{G}_n and \mathcal{F}_n are indeed generating sets for \mathcal{O}_n and \mathcal{L}_n , respectively [2].

Circuits over a set \mathbb{G} of quantum gates are constructed from the elements of \mathbb{G} through composition and tensor product. Circuits can use ancillary qubits, but these must be initialized and terminated in the computational basis state $|0\rangle$. For example, in the diagram below the circuit C uses a single ancilla.



Ancillas provide additional computational space and, as we will see below, can be useful in reducing the gate count of circuits.

If C is a circuit over some gate set, we write $\llbracket C \rrbracket$ for the matrix represented by C and we say that C represents $\llbracket C \rrbracket$. If \mathbb{G} is a gate set, we write $\mathcal{U}(\mathbb{G})$ for the collection of matrices representable by a circuit over \mathbb{G} . That is, $\mathcal{U}(\mathbb{G}) = \{\llbracket C \rrbracket ; C \text{ is a circuit over } \mathbb{G}\}$.

Definition 10. *The exact synthesis problem for a gate set \mathbb{G} is the following: given $U \in \mathcal{U}(\mathbb{G})$, find a circuit C over \mathbb{G} such that $\llbracket C \rrbracket = U$. A constructive solution to the exact synthesis problem for \mathbb{G} is known as an exact synthesis algorithm for \mathbb{G} .*

The *Toffoli-Hadamard* gate set consists of the gates CCX and H . Because the Toffoli gate is universal for classical reversible computation with ancillary bits in the 0 or 1 state [8], one can express both X and CX over this gate set. As a result, and by a slight abuse of terminology, we refer to the gate set $\{X, CX, CCX, H\}$ as the Toffoli-Hadamard gate set.

It was shown in [9] and later, independently, in [2], that the operators exactly representable by an m -qubit Toffoli-Hadamard circuit are precisely the elements of \mathcal{L}_{2^m} . The proof of this fact takes the form of an exact synthesis algorithm. The algorithm of [2], following prior work of [10], proceeds in two steps. First, one shows that, when n is a power of 2, every operator in \mathcal{F}_n can be exactly represented by a circuit over $\{X, CX, CCX, H\}$. Then, one shows that every element of \mathcal{L}_n can be factored as a product of matrices from \mathcal{F}_n . Together, these two steps solve the exact synthesis problem for the Toffoli-Hadamard gate set. By considering the gate set $\{X, CX, CCX, K\}$, rather than the gate set $\{X, CX, CCX, H\}$, one obtains circuits that correspond precisely to the elements of \mathcal{O}_n [2]. The exact synthesis problem for this gate set is solved similarly, with the exact synthesis algorithm using \mathcal{G}_n rather than \mathcal{F}_n .

Each element of \mathcal{F}_n (resp. \mathcal{G}_n) can be represented by a circuit containing $O(\log(n))$ gates (so a constant number of gates when n is fixed). It is therefore the complexity of the factorization of elements of \mathcal{L}_n (resp. \mathcal{O}_n) into elements of \mathcal{F}_n (resp. \mathcal{G}_n) that determines the complexity of the overall synthesis algorithm. For this reason, in the rest of the paper, we focus on finding improved solutions to this factorization problem.

3 The Local Synthesis Algorithm

In this section, we revisit the solution to the exact synthesis problem for $\{X, CX, CCX, H\}$ (and $\{X, CX, CCX, K\}$) proposed in [2]. The algorithm, which we call the *local synthesis algorithm*, is an analogue of the *Giles-Selinger algorithm* introduced in [10] for the synthesis of Clifford+ T circuits. In a nutshell, the local synthesis algorithm proceeds one column at a time, reducing each column of the input matrix to a basis vector. This process is repeated until the input matrix is itself reduced to the identity. The algorithm is local in the sense that the matrix factorization is carried out column by column and that, at each step, the algorithm only uses information about the column currently being reduced. We now briefly recall the main points of [2, Section 5.1] in order to better understand the functionality of the local synthesis algorithm. We encourage the reader to consult [2] for further details.

Lemma 1. *Let v_0, v_1, v_2, v_3 be odd integers. Then there exists $\tau_0, \tau_1, \tau_2, \tau_3 \in \mathbb{Z}_2$ such that*

$$K_{[0,1,2,3]}(-1)_{[0]}^{\tau_0}(-1)_{[1]}^{\tau_1}(-1)_{[2]}^{\tau_2}(-1)_{[3]}^{\tau_3} \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} v'_0 \\ v'_1 \\ v'_2 \\ v'_3 \end{bmatrix},$$

where v'_0, v'_1, v'_2, v'_3 are even integers.

Lemma 2. *Let $|u\rangle \in \mathbb{Z}[1/2]^n$ be a unit vector with $\text{lde}(|u\rangle) = k$. Let $|v\rangle = 2^k |u\rangle$. If $k > 0$, the number of odd entries in $|v\rangle$ is a multiple of 4.*

Proof. Since $\langle u|u \rangle = 1$, $\langle v|v \rangle = 4^k$. Thus $\sum v_j^2 = 4^k$. Since the only squares modulo 4 are 0 and 1, and $v_j^2 \equiv 1 \pmod{4}$ if and only if v_j is odd, the number of v_j in $|v\rangle$ such that $v_j^2 \equiv 1 \pmod{4}$ is a multiple of 4. \square

Lemmas 1 and 2 imply the *Column Lemma*, the crux of the local synthesis algorithm.

Lemma 3 (Column Lemma). *Let $|u\rangle \in \mathbb{Z}[1/2]^n$ be a unit vector and $|j\rangle$ be a standard basis vector. There exists a sequence of generators $G_0, \dots, G_q \in \mathcal{G}_n$ such that $(G_q \cdots G_1)|u\rangle = |j\rangle$.*

Proof. Let $k = \text{lde}(|u\rangle)$ and proceed by induction on k . When $k = 0$, $|u\rangle = \pm |j'\rangle$ for some $0 \leq j' < n$. Indeed, since $|u\rangle$ is a unit vector, we have $\sum u_i^2 = 1$. Since $u_i \in \mathbb{Z}$, there must be exactly one i such that $u_i = \pm 1$ while all the other entries of $|u\rangle$ are 0. If $|j'\rangle = |j\rangle$ there is nothing to do. Otherwise, map $|u\rangle$ to $|j\rangle$ by applying an optional one-level (-1) generator followed by an optional two-level X generator. When $k > 0$, by **Lemma 2**, the number of odd entries in $|v\rangle = 2^k |u\rangle$ is a multiple of 4. We can then group these odd entries into quadruples and apply **Lemma 1** to each quadruple to reduce the least denominator exponent of the vector. By induction, we can continuously reduce k until it becomes 0, which is the base case. \square

Proposition 1. *Let U be an $n \times n$ matrix. Then $U \in \mathcal{O}_n$ if, and only if, U can be written as a product of elements of \mathcal{G}_n .*

Proof. The right-to-left direction follows from the fact that $\mathcal{G}_n \subseteq \mathcal{O}_n$. For the converse, use **Lemma 3** to reduce the leftmost unfixed column U_j to $|j\rangle$, $0 \leq j < n$. After that, repeat the column reduction on the next leftmost unfixed column until U is reduced to the identity. \square

The local synthesis algorithm establishes the left-to-right implication of **Proposition 1**. It expresses an element of \mathcal{O}_n as a product of generators from \mathcal{G}_n and thereby solves the exact synthesis problem for $\{X, CX, CCX, K\}$. A small extension of the algorithm shows that \mathcal{F}_n generates \mathcal{L}_n , solving the exact synthesis problem for $\{X, CX, CCX, H\}$.

Corollary 1. *Let U be an $n \times n$ matrix. Then $U \in \mathcal{L}_n$ if, and only if, U can be written as a product of elements of \mathcal{F}_n .*

Proof. As before, the right-to-left direction follows from the fact that $\mathcal{F}_n \subseteq \mathcal{L}_n$. Conversely, let $U \in \mathcal{L}_n$ and write U as $U = M/\sqrt{2}^q$, where M is an integer matrix and $q = \text{lde}_{\sqrt{2}}(U)$. If q is even, then $U \in \mathcal{O}_n$. By **Proposition 1**, U can be written as a product of elements of $\mathcal{G}_n \subset \mathcal{F}_n$. If q is odd, then by [2, Lemma 5.9] n must be even. It follows that $(I \otimes H)U \in \mathcal{O}_n$. We can conclude by applying **Proposition 1** to $(I \otimes H)U$. \square

In the rest of this section, we analyze the gate complexity of the local synthesis algorithm. In the worst case, it takes exponentially many generators in \mathcal{G}_n to decompose a unitary in \mathcal{O}_n . Since \mathcal{L}_n is simply a scaled version of \mathcal{O}_n , the same gate complexity holds for the local synthesis of \mathcal{L}_n over \mathcal{F}_n .

Lemma 4. *Let $|u\rangle \in \mathbb{Z}[1/2]^n$ with $\text{lde}(|u\rangle) = k$. Let $|j\rangle$ be a standard basis vector. The number of generators in \mathcal{G}_n required by **Lemma 3** to reduce $|u\rangle$ to $|j\rangle$ is $O(nk)$.*

Proof. Let $|v\rangle = 2^k |u\rangle$, then $|v\rangle \in \mathbb{Z}^n$. We proceed by case distinction. When $k = 0$, there is precisely one non-zero entry in $|v\rangle$, which is either 1 or -1 . We need at most a two-level X gate and a one-level (-1) gate to send $|v\rangle$ to $|j\rangle$. Hence the gate complexity over \mathcal{G}_n is $O(1)$. When $k > 0$, there are odd entries in $|v\rangle$ and the number of such entries must be doubly-even (i.e., a multiple of 4). To reduce k by 1 as in **Lemma 3**, we need to make all of the odd entries even. By **Lemma 1**, for each quadruple of odd entries, we need at most four one-level (-1) gates and precisely one four-level K gate. In the worst case, there are $\lfloor n/4 \rfloor$ quadruples of odd entries in $|v\rangle$. To reduce k to 0, we thus need at most $(4+1)\lfloor n/4 \rfloor k \in O(nk)$ elements of \mathcal{G}_n . Therefore, the total number of generators in \mathcal{G}_n required by **Lemma 3** to reduce $|u\rangle$ to $|j\rangle$ is $\max(O(nk), O(1)) = O(nk)$. \square

Proposition 2. *Let $U \in \mathcal{O}_n$ with $\text{lde}(U) = k$. Then, using the local synthesis algorithm, U can be represented by a product of $O(2^{nk})$ elements of \mathcal{G}_n .*

Proof. The local synthesis algorithm starts from the leftmost column of U that is not yet reduced. In the worst case, this column is U_0 and $\text{lde}(U_0) = k$. By [Lemma 4](#), we need $O(nk)$ generators in \mathcal{G}_n to reduce U_0 to $|0\rangle$. While reducing U_0 , the local synthesis algorithm may increase the least denominator exponent of the other columns of U . Each row operation potentially increases the least denominator exponent by 1. Therefore, the least denominator exponent of any other column in U may increase to $2k$ during the reduction of U_0 . Now let f_{U_i} be the cost of reducing U_i to $|i\rangle$. As the algorithm proceeds from the left to the right of U , f_{U_i} increases as shown below.

$$f_{U_0} \in O(nk), \quad f_{U_1} \in O((n-1)2k), \quad f_{U_2} \in O((n-2)2^2k), \quad \dots, \quad f_{U_{n-1}} \in O(2^{n-1}k).$$

In total, the number of generators from \mathcal{G}_n that are required to synthesize U is

$$S_n = \sum_{i=0}^{n-1} f_{U_i} = \sum_{i=0}^{n-1} (n-i)2^i k. \quad (1)$$

Multiplying both sides of [Equation \(1\)](#) by 2 yields

$$2S_n = (2n + (n-1)2^2 + (n-2)2^3 + (n-3)2^4 + \dots + 2^n) k. \quad (2)$$

Subtracting [Equation \(1\)](#) from [Equation \(2\)](#) yields

$$S_n = (-n + 2 + 2^2 + \dots + 2^{n-1} + 2^n) k = (-n + 2^{n+1} - 2) k \in O(2^n k).$$

Hence, the complexity of the local synthesis algorithm of \mathcal{O}_n over \mathcal{G}_n is $O(2^n k)$. \square

Corollary 2. *Let $U \in \mathcal{L}_n$ with $\text{lde}_{\sqrt{2}}(U) = k$. Then, using the local synthesis algorithm, U can be represented by a product of $O(2^n k)$ elements of \mathcal{F}_n .*

Proof. When k is even, $U \in \mathcal{O}_n$ and, by [Proposition 2](#), U can be represented by $O(2^n k)$ generators in $\mathcal{G}_n \subset \mathcal{F}_n$. When k is odd then, by [\[2, Lemma 5.9\]](#), n must be even so that $(I \otimes H)U \in \mathcal{O}_n$. Applying [Proposition 2](#) to $(I \otimes H)U$ yields a sequence of $O(2^n k)$ generators over \mathcal{G}_n for $(I \otimes H)U$. Hence, the complexity of synthesizing U over \mathcal{F}_n is $O(2^n k)$. \square

In the context of quantum computation, the dimension of the matrix to be synthesized is exponential in the number of qubits. That is, $n = 2^m$, where m is the number of qubits. Moreover, the cost of synthesizing an m -qubit circuit for any element of \mathcal{F}_{2^m} is linear in m . Therefore, the gate complexity of an m -qubit Toffoli-Hadamard circuit synthesized using the local synthesis algorithms is $O(2^{2^m} mk)$.

4 The Householder Synthesis Algorithm

In this section, we explore how using additional dimensions can be helpful in quantum circuit synthesis. These results are a direct adaptation to the Toffoli-Hadamard gate set of the methods introduced in [\[13\]](#) for the Clifford+ T gate set. Compared to the local synthesis algorithm, the algorithm presented in this section, which we call the *Householder synthesis algorithm*, reduces the gate complexity of the produced circuits from $O(2^n \log(n)k)$ to $O(n^2 \log nk)$, where n is the dimension of the input matrix.

Definition 11. *Let $|\psi\rangle$ be an n -dimensional unit vector. The reflection operator $R_{|\psi\rangle}$ around $|\psi\rangle$ is defined as*

$$R_{|\psi\rangle} = I - 2|\psi\rangle\langle\psi|.$$

Note that if R is a reflection operator about some unit vector, then R is unitary. Indeed, $R = R^\dagger$ and $R^2 = I$. As a result, if $|\psi\rangle$ is a unit vector of the form $|v\rangle/\sqrt{2^k}$ for some integer vector $|v\rangle$, then $R_{|\psi\rangle} \in \mathcal{L}_n$.

We start by showing that if $U \in \mathcal{L}_n$, then there is an operator U' constructed from U that can be conveniently factored as a product of reflections. In what follows, we will use two single-qubit states:

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad \text{and} \quad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

Proposition 3. Let $U \in \mathcal{L}_n$ and define

$$U' = |+\rangle\langle -| \otimes U + |- \rangle\langle +| \otimes U^\dagger.$$

Then $U' \in \mathcal{L}_{2n}$ and U' can be factored into n reflections in \mathcal{L}_{2n} . That is, $U' = R_{|\phi_0\rangle} \cdots R_{|\phi_{n-1}\rangle}$, where $R_{|\phi_0\rangle}, \dots, R_{|\phi_{n-1}\rangle} \in \mathcal{L}_{2n}$.

Proof. Let U and U' be as stated. It can be verified by direct computation that U' is unitary. Moreover, since $U \in \mathcal{L}_n$ and $|+\rangle\langle -|$ and $|- \rangle\langle +|$ are integral matrices scaled by $1/2$, it follows that $U' \in \mathcal{L}_{2n}$. It remains to show that U' is a product of reflection operators. Define

$$|\omega_j^\pm\rangle = \frac{|-\rangle|j\rangle \pm |+\rangle|u_j\rangle}{\sqrt{2}},$$

where $|u_j\rangle$ is the j -th column vector in U and $|j\rangle$ denotes the j -th computational basis vector. Since $\langle \omega_j^+ | \omega_j^+ \rangle = \langle \omega_j^- | \omega_j^- \rangle = 1$, both $|\omega_j^+\rangle$ and $|\omega_j^-\rangle$ are unit vectors. Moreover, it is easy to show that any two distinct $|\omega_j^\pm\rangle$ are orthogonal, so that $\{|\omega_j^\pm\rangle \mid j = 0, \dots, n-1\}$ forms an orthonormal basis. Now let $P_j^+ = |\omega_j^+\rangle\langle \omega_j^+|$ and $P_j^- = |\omega_j^-\rangle\langle \omega_j^-|$. It follows from the completeness equation that

$$I = \sum_{j=0}^{n-1} (|\omega_j^+\rangle\langle \omega_j^+| + |\omega_j^-\rangle\langle \omega_j^-|) = \sum_{j=0}^{n-1} (P_j^+ + P_j^-). \quad (3)$$

Furthermore, $|\omega_j^+\rangle$ and $|\omega_j^-\rangle$ are the $+1$ and -1 eigenstates of U' , respectively. Now note that U' is Hermitian and thus normal. Hence, by the spectral theorem, we have

$$U' = \sum_{j=0}^{n-1} (|\omega_j^+\rangle\langle \omega_j^+| - |\omega_j^-\rangle\langle \omega_j^-|) = \sum_{j=0}^{n-1} (P_j^+ - P_j^-). \quad (4)$$

From [Equations \(3\) and \(4\)](#), $I - U' = 2 \sum_{j=0}^{n-1} P_j^-$, which implies that

$$U' = I - 2 \sum_{j=0}^{n-1} P_j^- = I - 2 \sum_{j=0}^{n-1} |\omega_j^-\rangle\langle \omega_j^-| = \prod_{j=0}^{n-1} (I - 2 |\omega_j^-\rangle\langle \omega_j^-|) = \prod_{j=0}^{n-1} R_{|\omega_j^-\rangle}. \quad (5)$$

Since $|\omega_j^-\rangle$ is a unit vector of the form $|v_j\rangle/\sqrt{2^k}$ where $|v_j\rangle$ is an integer vector, $R_{|\omega_j^-\rangle} \in \mathcal{L}_{2n}$. This completes the proof. \square

By noting that $|+\rangle\langle -|$ and $|- \rangle\langle +|$ are matrices with dyadic entries, one can reason as in the proof of [Proposition 3](#) to show that an analogous result holds for $U \in \mathcal{O}_n$, rather than $U \in \mathcal{L}_n$.

Proposition 4. Let $U \in \mathcal{O}_n$ and define

$$U' = |+\rangle\langle -| \otimes U + |- \rangle\langle +| \otimes U^\dagger.$$

Then $U' \in \mathcal{O}_{2n}$ and U' can be factored into n reflections in \mathcal{O}_{2n} . That is, $U' = R_{|\phi_0\rangle} \cdots R_{|\phi_{n-1}\rangle}$, where $R_{|\phi_0\rangle}, \dots, R_{|\phi_{n-1}\rangle} \in \mathcal{O}_{2n}$.

Proposition 5. Let $|\psi\rangle = |v\rangle/\sqrt{2^k}$ be an n -dimensional unit vector, where $|v\rangle$ is an integer vector. Assume that $\text{Ide}_{\sqrt{2}}(|\psi\rangle) = k$. Then the reflection operator $R_{|\psi\rangle}$ can be exactly represented by $O(nk)$ generators over \mathcal{F}_n .

Proof. Let $|\psi\rangle$ be as stated. When k is even, then, by [Lemma 3](#), there exists a word G over \mathcal{G}_n such that

$$G|\psi\rangle = |0\rangle. \quad (6)$$

Since the elements of \mathcal{G}_n are self-inverse, the word G^\dagger obtained by reversing G is a word over \mathcal{G}_n such that $G^\dagger |0\rangle = |\psi\rangle$. Moreover, we have $G^\dagger R_{|0\rangle} G = R_{|\psi\rangle}$, since

$$G^\dagger R_{|0\rangle} G = G^\dagger (I - 2|0\rangle\langle 0|) G = I - 2(G^\dagger |0\rangle)(G^\dagger |0\rangle)^\dagger = R_{G^\dagger |0\rangle} = R_{|\psi\rangle}. \quad (7)$$

Hence the number of elements of \mathcal{G}_n that are needed to represent $R_{|\psi\rangle}$ is equal to the number of generators needed to represent G , G^\dagger , and $R_{|0\rangle}$. Note that

$$R_{|0\rangle} = I - 2|0\rangle\langle 0| = (-1)_{|0\rangle} \in \mathcal{G}_n.$$

Moreover, the number of generators needed to represent G^\dagger is equal to the number of generators needed to represent G . By [Lemma 4](#), $O(nk)$ generators are needed for this. Hence, $R_{|\psi\rangle}$ can be exactly represented by $O(nk)$ generators over $\mathcal{G}_n \subset \mathcal{F}_n$. When k is odd, we can reason as in [Corollary 1](#) to show that $R_{|\psi\rangle}$ can be represented as a product of $O(nk)$ generators from \mathcal{F}_n . \square

Proposition 6. *Let $U \in \mathcal{L}_n$ and $U' \in \mathcal{L}_{2n}$ be as in [Proposition 3](#) and assume that $\text{lde}_{\sqrt{2}}(U) = k$. Then U' can be represented by $O(n^2k)$ generators from \mathcal{F}_n .*

Proof. By [Proposition 3](#), U' can be expressed as a product n reflections. By [Proposition 5](#), each one of these reflections can be exactly represented by $O(nk)$ generators from \mathcal{F}_n . Therefore, to express U' , we need $n \cdot O(nk) = O(n^2k)$ generators from \mathcal{F}_n . \square

Corollary 3. *Let $U \in \mathcal{O}_n$ and $U' \in \mathcal{O}_{2n}$ be as in [Proposition 4](#) and assume that $\text{lde}(U) = k$. Then U' can be represented by $O(n^2k)$ generators from \mathcal{G}_n .*

To conclude this section, we use [Proposition 3](#) to define the Householder synthesis algorithm, which produces circuits of size $O(4^m mk)$. Suppose that $n = 2^m$, where m is the number of qubits on which a given operator $U \in \mathcal{L}_{2^m}$ acts. Suppose moreover that $\text{lde}_{\sqrt{2}}(U) = k$. The operator U' of [Proposition 6](#) can be represented as a product of $O(n^2k) = O(4^m k)$ elements of $\mathcal{F}_{2^{m+1}}$. Since any element of $\mathcal{F}_{2^{m+1}}$ can be represented by a Toffoli-Hadamard circuit of gate count $O(m)$, we get a circuit D of size $O(4^m mk)$ for U' . Now consider the circuit $C = (H \otimes I)D(HX \otimes I)$. For any state $|\phi\rangle$, we have

$$C|0\rangle|\phi\rangle = (H \otimes I)D(HX \otimes I)|0\rangle|\phi\rangle = (H \otimes I)D|-\rangle|\phi\rangle = (H \otimes I)|+\rangle U|\phi\rangle = |0\rangle U|\phi\rangle.$$

Hence, C is a Toffoli-Hadamard circuit for U (which uses an additional ancillary qubit).

The Householder exact synthesis algorithm can be straightforwardly defined in the case of circuits over the gate set $\{X, CX, CCX, K\}$, with the small caveat that two additional ancillary qubits are required, since one cannot prepare a single qubit in the state $|-\rangle$ over $\{X, CX, CCX, K\}$.

5 The Global Synthesis Algorithm

The local synthesis algorithm factorizes a matrix by reducing one column at a time. As we saw in [Section 3](#), this approach can lead to large circuits, since reducing the least (scaled) denominator exponent of one column may increase that of the subsequent columns. We now take a global view of the matrix, focusing on matrices of dimension 2, 4, and 8 (i.e., matrices on 1, 2, and 3 qubits). Through a careful study of the structure of these matrices, we define a synthesis algorithm that reduces the least (scaled) denominator exponent of the entire matrix at every iteration. We refer to this alternative synthesis algorithm as the *global synthesis algorithm*.

5.1 Binary Patterns

We associate a binary matrix (i.e., a matrix over \mathbb{Z}_2) to every element of \mathcal{L}_n . These binary matrices, which we call *binary patterns*, will be useful in designing a global synthesis algorithm.

Definition 12. *Let $U \in \mathcal{L}_n$ and write U as $U = M/\sqrt{2}^k$ with $\text{lde}_{\sqrt{2}}(U) = k$. The binary pattern of U is the binary matrix \bar{U} defined by $\bar{U}_{i,j} = M_{i,j} \pmod{2}$.*

$$\begin{aligned}
A &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}, \\
E &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad F = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}, \\
I &= \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad J = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad K = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad L = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}, \\
M &= \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad N = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.
\end{aligned}$$

Figure 1. Binary patterns for the elements of \mathcal{L}_8 .

The matrix \overline{U} is the binary matrix obtained by taking the residue modulo 2 of every entry of the integral part of U (when U is written using its least scaled denominator exponent). The next two lemmas establish important properties of binary patterns.

Lemma 5. *Let $U \in \mathcal{L}_n$ with $\text{lde}_{\sqrt{2}}(U) = k$. If $k > 1$, then the number of 1's in any column of \overline{U} is doubly-even.*

Proof. Consider an arbitrary column $|u\rangle = |v\rangle / \sqrt{2}^k$ of U . Let $|\overline{u}\rangle$ be the corresponding column in \overline{U} . Since $\langle u|u\rangle = 1$, we have $\sum v_i^2 = 2^k$. Thus, when $k > 1$, we have $\sum v_i^2 \equiv 0 \pmod{4}$. Since $v_i^2 \equiv 1 \pmod{4}$ if and only if $v_i \equiv 1 \pmod{2}$, and since the only squares modulo 4 are 0 and 1, the number of odd v_i must be a multiple of 4. Hence, the number of 1's in any column of \overline{U} is doubly-even. \square

Lemma 6. *Let $U \in \mathcal{L}_n$ with $\text{lde}_{\sqrt{2}}(U) = k$. If $k > 0$, then any two distinct columns of \overline{U} have evenly many 1's in common.*

Proof. Consider two distinct columns $|u\rangle$ and $|w\rangle$ of U . Let $|\overline{u}\rangle$ and $|\overline{w}\rangle$ be the corresponding columns in \overline{U} . Since U is orthogonal, we have

$$\langle u|w\rangle = \sum_{i=0}^{n-1} u_i w_i = 0. \quad (8)$$

Taking [equation \(8\)](#) modulo 2 implies that $|\{i : \overline{u}_i = \overline{w}_i = 1\}| \equiv 0 \pmod{2}$, as desired. \square

[Lemmas 5](#) and [6](#) also hold for the rows of \overline{U} . The proofs are similar, so they are omitted here. These lemmas show that the binary matrices that are the binary pattern of an element of \mathcal{L}_n form a strict subset of $\mathbb{Z}_2^{n \times n}$. The proposition below gives a characterization of this subset for $n = 8$. The proof of the proposition is a long case distinction which can be found in [Appendix A](#).

Proposition 7. Let $U \in \mathcal{L}_8$ with $\text{lde}_{\sqrt{2}}(U) \geq 2$. Then up to row permutation, column permutation, and taking the transpose, \overline{U} is one of the 14 binary patterns in [Figure 1](#).

Definition 13. Let n be even and let $B \in \mathbb{Z}_2^{n \times n}$. We say that B is row-paired if the rows of B can be partitioned into identical pairs. Similarly, we say that B is column-paired if the columns of B can be partitioned into identical pairs.

Note that, for $U \in \mathcal{L}_n$, if \overline{U} is row-paired, then \overline{U}^\top is column-paired. Indeed, if \overline{U} is row-paired, then \overline{U}^\top is column-paired so that $\overline{U}^\top = \overline{U}^\top$ is column-paired.

Row-paired binary patterns will play an important role in the global synthesis algorithm. Intuitively, if \overline{U} is row-paired, then one can permute the rows of U to place identical rows next to one another, at which point a single Hadamard gate can be used to globally reduce the least scaled denominator exponent of U . This intuition is detailed in [Lemma 7](#), where S_n denotes the symmetric group on n letters.

Lemma 7. Let n be even and let $U \in \mathcal{L}_n$. If \overline{U} is row-paired, then there exists $P \in S_n$ such that

$$\text{lde}_{\sqrt{2}}((I \otimes H)PU) < \text{lde}_{\sqrt{2}}(U).$$

Proof. Let $U = M/\sqrt{2}^k$ and let r_0, \dots, r_{n-1} be the rows of M . Because \overline{U} is row-paired, there exists some $P \in S_n$ such that

$$PU = \frac{1}{\sqrt{2}^k} \begin{bmatrix} r_0 \\ \vdots \\ r_{n-1} \end{bmatrix},$$

with $r_0 \equiv r_1, r_2 \equiv r_3, \dots$, and $r_{n-2} \equiv r_{n-1}$ modulo 2. Since $I \otimes H$ is the block diagonal matrix $I \otimes H = \text{diag}(H, H, \dots, H)$, left-multiplying PU by $I \otimes H$ yields

$$(I \otimes H)PU = \begin{bmatrix} r_0 \\ \vdots \\ r_{n-1} \end{bmatrix} = \frac{1}{\sqrt{2}^{k+1}} \begin{bmatrix} r_0 + r_1 \\ r_0 - r_1 \\ \vdots \\ r_{n-2} + r_{n-1} \\ r_{n-2} - r_{n-1} \end{bmatrix} = \frac{2}{\sqrt{2}^{k+1}} \begin{bmatrix} r'_0 \\ \vdots \\ r'_{n-1} \end{bmatrix} = \frac{1}{\sqrt{2}^{k-1}} \begin{bmatrix} r'_0 \\ \vdots \\ r'_{n-1} \end{bmatrix},$$

for some integer row vectors r'_0, \dots, r'_{n-1} . Thus, $\text{lde}_{\sqrt{2}}((I \otimes H)PU) < \text{lde}_{\sqrt{2}}(U)$ as desired. \square

Lemma 8. Let n be even and let $U \in \mathcal{L}_n$. If \overline{U} is column-paired, then there exists $P \in S_n$ such that

$$\text{lde}_{\sqrt{2}}(UP(I \otimes H)) < \text{lde}_{\sqrt{2}}(U).$$

Proof. Since \overline{U} is column-paired, \overline{U}^\top is row-paired. By [Lemma 7](#), there exists $Q \in S_n$ such that $\text{lde}_{\sqrt{2}}((I \otimes H)QU^\top) < \text{lde}_{\sqrt{2}}(U^\top)$. Hence, letting $P = Q^\top$, and using the fact that the least scaled denominator exponent of an element of \mathcal{L}_n is the same as that of its transpose, we get

$$\text{lde}_{\sqrt{2}}(UP(I \otimes H)) = \text{lde}_{\sqrt{2}}((UP(I \otimes H))^\top) = \text{lde}_{\sqrt{2}}((I \otimes H)QU^\top) < \text{lde}_{\sqrt{2}}(U^\top) = \text{lde}_{\sqrt{2}}(U). \quad \square$$

Lemma 9. Let $U \in \mathcal{L}_8$ with $\text{lde}_{\sqrt{2}}(U) = k$. If \overline{U} is neither row-paired nor column-paired, then, up to row permutation, column permutation, and taking the transpose, $(I \otimes H)\overline{U}(I \otimes H)$ is row-paired and $\text{lde}_{\sqrt{2}}((I \otimes H)U(I \otimes H)) \leq \text{lde}_{\sqrt{2}}(U)$.

Proof. Let U be as stated. By [Proposition 7](#), up to row permutation, column permutation, and taking the transpose, \overline{U} is one of the binary patterns in [Figure 1](#). Since \overline{U} is neither row-paired nor column-paired, \overline{U} is L , M , or N . Write \overline{U} as the 4×4 block matrix

$$\overline{U} = \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} \\ P_{1,0} & P_{1,1} & P_{1,2} & P_{1,3} \\ P_{2,0} & P_{2,1} & P_{2,2} & P_{2,3} \\ P_{3,0} & P_{3,1} & P_{3,2} & P_{3,3} \end{bmatrix},$$

where $P_{i,j}$ is a 2×2 binary matrix. By inspection of [Figure 1](#), since \bar{U} is one of L , M , or N , we see that each $P_{i,j}$ is one of the binary matrices below:

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

In particular, each $P_{i,j}$ has evenly many nonzero entries. Now write U as the 4×4 block matrix

$$U = \frac{1}{\sqrt{2^k}} \begin{bmatrix} Q_{0,0} & Q_{0,1} & Q_{0,2} & Q_{0,3} \\ Q_{1,0} & Q_{1,1} & Q_{1,2} & Q_{1,3} \\ Q_{2,0} & Q_{2,1} & Q_{2,2} & Q_{2,3} \\ Q_{3,0} & Q_{3,1} & Q_{3,2} & Q_{3,3} \end{bmatrix},$$

where $Q_{i,j}$ is a 2×2 integer matrix such that $Q_{i,j} = P_{i,j}$ modulo 2. As $I \otimes H = \text{diag}(H, H, H, H)$, we have

$$(I \otimes H)U(I \otimes H) = \frac{1}{\sqrt{2^k}} \begin{bmatrix} Q'_{0,0} & Q'_{0,1} & Q'_{0,2} & Q'_{0,3} \\ Q'_{1,0} & Q'_{1,1} & Q'_{1,2} & Q'_{1,3} \\ Q'_{2,0} & Q'_{2,1} & Q'_{2,2} & Q'_{2,3} \\ Q'_{3,0} & Q'_{3,1} & Q'_{3,2} & Q'_{3,3} \end{bmatrix},$$

where $Q'_{i,j} = HQ_{i,j}H$. Since $Q_{i,j}$ is an integer matrix with evenly many odd entries and, since for any integers w, x, y , and z , we have

$$H \begin{bmatrix} w & x \\ y & z \end{bmatrix} H = \frac{1}{2} \begin{bmatrix} w+x+y+z & w-x+y-z \\ w+x-y-z & w-x-y+z \end{bmatrix},$$

it follows that $Q'_{i,j} = HQ_{i,j}H$ is itself an integer matrix. Thus, $\text{lde}_{\sqrt{2}}((I \otimes H)U(I \otimes H)) \leq \text{lde}_{\sqrt{2}}(U)$. A long but straightforward calculation shows that $(I \otimes H)U(I \otimes H)$ is in fact row-paired. \square

5.2 The 1- and 2-Qubit Cases

We now discuss the exact synthesis problem for \mathcal{L}_2 and \mathcal{L}_4 . The problem is simple in these cases because the groups are finite. Despite their simplicity, these instances of the problem shed some light on our method for defining a global synthesis algorithm for \mathcal{L}_8 .

Proposition 8. *If $U \in \mathcal{L}_2$, then $\text{lde}_{\sqrt{2}}(U) \leq 1$.*

Proof. Let $k = \text{lde}_{\sqrt{2}}(U)$ and suppose that $k \geq 2$. Let $|u\rangle$ be the first column of U with $\text{lde}(|u\rangle) = k$ and let $|v\rangle = 2^k |u\rangle$. As $\langle u|u\rangle = 1$, we have $v_0^2 + v_1^2 = 2^k \equiv 0 \pmod{4}$, since $k \geq 2$. Therefore, $v_0 \equiv v_1 \equiv 0 \pmod{2}$. This is a contradiction since at least one of v_0 and v_1 must be odd for k to be minimal. \square

Lemma 10. *Let $a \in \mathbb{Z}$. Then $a^2 \equiv 1 \pmod{8}$ if and only if $a \equiv 1 \pmod{2}$.*

Proof. If $a \equiv 0 \pmod{2}$, then a^2 is even, so $a^2 \not\equiv 1 \pmod{8}$. If $a \equiv 1 \pmod{2}$, then $a = 2q + 1$ for some $q \in \mathbb{Z}$, so that $a^2 = 4q^2 + 4q + 1$. If $q = 2p$ for some $p \in \mathbb{Z}$, then $a^2 = 1 + 8(2p^2 + p) \equiv 1 \pmod{8}$. Otherwise, $q = 2p + 1$ for some $p \in \mathbb{Z}$ and $a^2 = 1 + 8(2p^2 + 3p + 1) \equiv 1 \pmod{8}$. \square

Proposition 9. *If $U \in \mathcal{L}_4$ then $\text{lde}_{\sqrt{2}}(U) \leq 2$.*

Proof. Let $k = \text{lde}_{\sqrt{2}}(U)$ and suppose that $k \geq 3$. Let $|u\rangle$ be the first column of U with $\text{lde}_{\sqrt{2}}(|u\rangle) = k$ and let $|v\rangle = \sqrt{2}^k |u\rangle$. By reasoning as in [Lemma 2](#), we see that the number of odd entries in $|v\rangle$ must be doubly-even. Hence, $v_0 \equiv v_1 \equiv v_2 \equiv v_3 \equiv 1 \pmod{2}$. By [Lemma 10](#), $v_0^2 \equiv v_1^2 \equiv v_2^2 \equiv v_3^2 \equiv 1 \pmod{8}$. As $\langle u|u\rangle = 1$, we have $v_0^2 + v_1^2 + v_2^2 + v_3^2 = 4^k \equiv 0 \pmod{8}$. This is a contradiction since we in fact have $v_0^2 + v_1^2 + v_2^2 + v_3^2 \equiv 4 \pmod{8}$. \square

It follows from [Proposition 9](#) that \mathcal{L}_4 is finite. Indeed, by [Proposition 9](#), the least scaled denominator exponent of an element of \mathcal{L}_4 can be no more than 2. As a consequence, the number of possible columns for a matrix in \mathcal{L}_4 is upper bounded by the number of integer solutions to the equation $v_0^2 + v_1^2 + v_2^2 + v_3^2 = 2^k$, which is finite since $k \leq 2$. [Proposition 8](#) similarly implies that \mathcal{L}_2 is finite.

In principle, one can therefore define an exact synthesis algorithm for \mathcal{L}_4 by explicitly constructing a circuit for every element of the group using, e.g., the local algorithm of [Section 3](#). We now briefly outline a different approach to solving this problem.

Lemma 11. *Let $U \in \mathcal{L}_4$. If $\text{lde}_{\sqrt{2}}(U) \geq 1$, then, up to row permutation and column permutation \bar{U} is one of the binary patterns below.*

$$B_0 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

Proof. By [Proposition 9](#), we only need to consider the cases $\text{lde}_{\sqrt{2}}(U) = 1$ and $\text{lde}_{\sqrt{2}}(U) = 2$. When $\text{lde}_{\sqrt{2}}(U) = 2$, by [Lemma 5](#), $\bar{U} = B_2$. When $\text{lde}_{\sqrt{2}}(U) = 1$, then the rows and columns of $\sqrt{2}U$ are integer vectors of norm no more than 2 and must therefore contain 0 or 2 odd entries. It then follows from [Lemma 6](#) that the only two possible binary patterns for U are B_0 and B_1 , up to row permutation and column permutation. \square

Proposition 10. *Let $U \in \mathcal{L}_4$. Then U can be represented by $O(1)$ generators in \mathcal{F}_4 .*

Proof. Let $\text{lde}_{\sqrt{2}}(U) = k$. By [Proposition 9](#), $k \leq 2$. When $k = 0$, U is a signed permutation matrix and can therefore be written as a product of no more than 3 two-level X gates and 4 one-level (-1) gates. When $k > 0$, then, by [Lemma 11](#), \bar{U} is one of B_0 , B_1 , or B_2 . Since all of these binary patterns are row-paired, we can apply [Lemma 7](#) to reduce the least scaled denominator exponent of U . \square

The exact synthesis algorithm given in the proof of [Proposition 10](#) is the global synthesis algorithm for \mathcal{L}_4 . The algorithm relies on [Lemma 11](#), which characterizes the possible binary patterns for elements of \mathcal{L}_4 .

5.3 The 3-Qubit Case

We now turn to the case of \mathcal{L}_8 (and \mathcal{O}_8). This case is more complex than the one discussed in the previous section, because \mathcal{L}_8 is an infinite group. Luckily, the characterization given in [Proposition 7](#) allows us to proceed as in [Proposition 10](#).

Proposition 11. *Let $U \in \mathcal{L}_8$ with $\text{lde}_{\sqrt{2}}(U) = k$. Then U can be represented by $O(k)$ generators in \mathcal{F}_8 using the global synthesis algorithm.*

Proof. By induction on k . There are only finitely many elements in \mathcal{L}_8 with $k \leq 1$, so each one of them can be represented by a product of $O(1)$ elements of \mathcal{F}_8 . When $k \geq 2$, by [Proposition 7](#), \bar{U} must be one of the 14 binary patterns in [Figure 1](#). When \bar{U} is row-paired, by [Lemma 7](#), there exists some $P \in S_8$ such that

$$\text{lde}_{\sqrt{2}}((I \otimes H)PU) \leq k - 1.$$

If \bar{U} is not row-paired, then, by inspection of [Figure 1](#), \bar{U} is neither row-paired nor column-paired and so, by [Lemma 9](#), $U' = (I \otimes H)U(I \otimes H)$ is row-paired and $\text{lde}_{\sqrt{2}}(U') \leq \text{lde}_{\sqrt{2}}(U)$. Thus, by [Lemma 7](#), there exists $P \in S_8$ such that

$$\text{lde}_{\sqrt{2}}((I \otimes H)PU') \leq k - 1.$$

Continuing in this way, and writing each element of S_8 as a constant number of elements of \mathcal{F}_8 , we obtain a sequence of $O(k)$ elements of \mathcal{F}_8 whose product represents U . \square

We end this section by showing that the global synthesis algorithm for \mathcal{L}_8 given in [Proposition 11](#) can be used to define a global synthesis algorithm \mathcal{O}_8 of similar asymptotic cost. The idea is to consider an element U of \mathcal{O}_8 as an element of \mathcal{L}_8 (which is possible since $\mathcal{O}_8 \subseteq \mathcal{L}_8$) and to apply the algorithm of [Proposition 11](#) to U . This yields a decomposition of U that contains evenly many $I \otimes H$ gates, but these can be removed through rewriting as in [\[16\]](#).

Lemma 12. *For any word W over $\{(-1)_{[a]}, X_{[a,b]} ; 0 \leq a < b < n\}$, there exists a word W' over \mathcal{G}_n such that $(I \otimes H)W = W'(I \otimes H)$. Moreover, if W has length ℓ , then W' has length $c\ell$ for some positive integer c that depends on n .*

Proof. Consider the relations below, where a is assumed to be even in [Equations \(10\) and \(12\)](#) and a is assumed to be odd in [Equations \(11\) and \(13\)](#).

$$(I \otimes H)(I \otimes H) = \epsilon \quad (9)$$

$$(I \otimes H)(-1)_{[a]} = (-1)_{[a]}X_{[a,a+1]}(-1)_{[a]}(I \otimes H) \quad (10)$$

$$(I \otimes H)(-1)_{[a]} = X_{[a-1,a]}(I \otimes H) \quad (11)$$

$$(I \otimes H)X_{[a,a+1]} = (-1)_{[a+1]}(I \otimes H) \quad (12)$$

$$(I \otimes H)X_{[a,a+1]} = K_{[a-1,a,a+1,a+2]}X_{[a,a+1]}(I \otimes H) \quad (13)$$

The relations show that commuting $I \otimes H$ with $(-1)_{[a]}$ or $X_{[a,a+1]}$ adds only a constant number of gates. To commute $I \otimes H$ with $X_{[a,b]}$, one can first express $X_{[a,b]}$ in terms of $X_{[a,a+1]}$, and then apply the relations above. The result then follows by induction on the length of W . \square

Proposition 12. *Let $U \in \mathcal{O}_8$ with $\text{lde}(U) = k$. Then U can be represented by $O(k)$ generators in \mathcal{G}_8 using the global synthesis algorithm.*

Proof. By [Proposition 11](#), one can find a word W of length $O(k)$ over \mathcal{F}_8 that represents U and contains evenly many occurrences of $I \otimes H$. By construction, each pair of $I \otimes H$ gates is separated by a word over $\{(-1)_{[a]}, X_{[a,b]} ; 0 \leq a < b < n\}$ and can thus be eliminated by an application of [Lemma 12](#). This yields a new word W' over \mathcal{G}_8 of length $O(k)$. \square

6 Conclusion

In this paper, we studied the synthesis of Toffoli-Hadamard circuits. We focused on circuits over the gate sets $\{X, CX, CCX, H\}$ and $\{X, CX, CCX, K\}$. Because circuits over these gate sets correspond to matrices in the groups \mathcal{L}_n and \mathcal{O}_n , respectively, each circuit synthesis problem reduces to a factorization problem in the corresponding matrix group. The existing local synthesis algorithm was introduced in [\[2\]](#). We proposed two alternative algorithms.

Our first algorithm, the Householder synthesis algorithm, is an adaptation of prior work by Kliuchnikov [\[13\]](#) and applies to matrices of arbitrary size. The Householder algorithm first factors the given matrix as a product of reflection operators, and then synthesizes each reflection in this factorization. The Householder algorithm uses an additional qubit, but reduces the overall complexity of the synthesized circuit from $O(2^n \log(n)k)$ to $O(n^2 \log(n)k)$.

Our second algorithm, the global synthesis algorithm, is inspired by prior work of Russell, Niemann and others [\[21, 18\]](#). The global algorithm relies on a small dictionary of binary patterns which ensures that every step of the algorithm strictly decreases the least denominator exponent of the matrix to be synthesized. Because this second algorithm only applies to matrices of dimension 2, 4, and 8, it is difficult to compare its complexity with that of the other methods. However, the global nature of the algorithm makes it plausible that it would outperform the method of [\[2\]](#) in practice, and we leave this as an avenue for future research.

Looking forward, many questions remain. Firstly, it would be interesting to compare the algorithms in practice. Further afield, we would like to find a standalone global synthesis for \mathcal{O}_8 , rather than relying on the corresponding result for \mathcal{L}_8 and the commutation of generators. This may require a careful study of residue patterns modulo 4, rather than modulo 2, as we did here. Finally, we hope to extend the global synthesis method to larger, or even arbitrary, dimensions.

7 Acknowledgement

Part of this research was carried out during SML’s undergraduate honours work at Dalhousie University. The authors would like to thank Jiaxin Huang and John van de Wetering for enlightening discussions. The circuit diagrams in this paper were typeset using Quantikz [12].

References

1. Aharonov, D.: A simple proof that Toffoli and Hadamard are quantum universal. arXiv preprint quant-ph/0301040 (2003)
2. Amy, M., Glaudell, A.N., Ross, N.J.: Number-theoretic characterizations of some restricted Clifford+ T circuits. *Quantum* **4**, 252 (2020)
3. Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J.C., Barends, R., Biswas, R., Boixo, S., Brandao, F.G., Buell, D.A., et al.: Quantum supremacy using a programmable superconducting processor. *Nature* **574**(7779), 505–510 (2019)
4. Cory, D.G., Price, M., Maas, W., Knill, E., Laflamme, R., Zurek, W.H., Havel, T.F., Somaroo, S.S.: Experimental quantum error correction. *Physical Review Letters* **81**(10), 2152 (1998)
5. Dalla Chiara, M.L., Ledda, A., Sergioli, G., Giuntini, R.: The Toffoli-Hadamard gate system: an algebraic approach. *Journal of Philosophical Logic* **42**, 467–481 (2013)
6. Fedorov, A., Steffen, L., Baur, M., da Silva, M.P., Wallraff, A.: Implementation of a Toffoli gate with superconducting circuits. *Nature* **481**(7380), 170–172 (2012)
7. Forest, S., Gosset, D., Kliuchnikov, V., McKinnon, D.: Exact synthesis of single-qubit unitaries over Clifford-cyclotomic gate sets. *Journal of Mathematical Physics* **56**(8), 082201 (2015)
8. Fredkin, E., Toffoli, T.: Conservative logic. *International Journal of Theoretical Physics* **21**(3-4), 219–253 (1982)
9. Gajewski, D.C.: Analysis of groups generated by quantum gates. Ph.D. thesis, University of Toledo (2009)
10. Giles, B., Selinger, P.: Exact synthesis of multiqubit Clifford+ T circuits. *Physical Review A* **87**(3), 032332 (2013)
11. Gottesman, D.: Stabilizer codes and quantum error correction. California Institute of Technology (1997)
12. Kay, A.: Tutorial on the quantikz package. arXiv preprint arXiv:1809.03842 (2018)
13. Kliuchnikov, V.: Synthesis of unitaries with Clifford+ T circuits. arXiv preprint arXiv:1306.3200 (2013)
14. Kliuchnikov, V., Maslov, D., Mosca, M.: Fast and efficient exact synthesis of single-qubit unitaries generated by Clifford and T gates. *Quantum Information & Computation* **13**(7-8), 607–630 (2013)
15. Kliuchnikov, V., Yard, J.: A framework for exact synthesis. arXiv preprint arXiv:1504.04350 (2015)
16. Li, S.M., Ross, N.J., Selinger, P.: Generators and relations for the group $O_n(\mathbb{Z}[1/2])$. In: Heunen, C., Backens, M. (eds.) *Proceedings 18th International Conference on Quantum Physics and Logic, QPL 2021, Gdansk, Poland, and online, 7-11 June 2021. EPTCS*, vol. 343, pp. 210–264 (2021). <https://doi.org/10.4204/EPTCS.343.11>, <https://doi.org/10.4204/EPTCS.343.11>
17. Nielsen, M.A., Chuang, I.L.: Quantum computation and quantum information. *Phys. Today* **54**(2), 60 (2001)
18. Niemann, P., Wille, R., Drechsler, R.: Advanced exact synthesis of Clifford+ T circuits. *Quantum Information Processing* **19**(9), 1–23 (2020)
19. Paetznick, A., Reichardt, B.W.: Universal fault-tolerant quantum computation with only transversal gates and error correction. *Physical Review Letters* **111**(9), 090505 (2013)
20. Reed, M.D., DiCarlo, L., Nigg, S.E., Sun, L., Frunzio, L., Girvin, S.M., Schoelkopf, R.J.: Realization of three-qubit quantum error correction with superconducting circuits. *Nature* **482**(7385), 382–385 (2012)
21. Russell, T.: The exact synthesis of 1- and 2-qubit Clifford+ T circuits. arXiv preprint arXiv:1408.6202 (2014)
22. Shi, Y.: Both Toffoli and controlled-NOT need little help to do universal quantum computing. *Quantum Information & Computation* **3**(1), 84–92 (2003)
23. Vilmart, R.: A ZX-calculus with triangles for Toffoli-Hadamard, Clifford+ T , and beyond. In: Chiribella, G., Selinger, P. (eds.) *15th International Conference on Quantum Physics and Logic (QPL 2018)*. p. 313–344. *Electronic Proceedings in Theoretical Computer Science (EPTCS 287)*, Halifax, Canada (2018). <https://doi.org/10.4204/EPTCS.287.18>, <https://doi.org/10.4204/EPTCS.287.18>
24. Vilmart, R.: Completeness of sum-over-paths for Toffoli-Hadamard and the dyadic fragments of quantum computation. In: Klin, B., Pimentel, E. (eds.) *31st EACSL Annual Conference on Computer Science Logic (CSL 2023)*. *Leibniz International Proceedings in Informatics (LIPIcs)*, vol. 252, pp. 36:1–36:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2023). <https://doi.org/10.4230/LIPIcs.CSL.2023.36>, <https://drops.dagstuhl.de/opus/volltexte/2023/17497>

25. Yoder, T.J.: Universal fault-tolerant quantum computation with Bacon-Shor codes. arXiv preprint arXiv:1705.01686 (2017)
26. Zhu, Q., Cao, S., Chen, F., Chen, M.C., Chen, X., Chung, T.H., Deng, H., Du, Y., Fan, D., Gong, M., et al.: Quantum computational advantage via 60-qubit 24-cycle random circuit sampling. *Science Bulletin* **67**(3), 240–245 (2022)

A Proof of Proposition 7

Proposition 7. *Let $U \in \mathcal{L}_8$ with $\text{lde}_{\sqrt{2}}(U) \geq 2$. Then up to row permutation, column permutation, and taking the transpose, \overline{U} is one of the 14 binary patterns in Figure 1.*

Proof. Let u_i denote the i -th column of U , and u_i^\dagger denote the i -th row of U , $0 \leq i < 8$. Let $\|v\|$ denote the hamming weight of v , where v is a string of binary bits. Proceed by case distinction.

Case 1. There are identical rows or columns in U . Up to transposition, suppose U has two rows that are identical. By Proposition 13, $U \in \mathcal{B}_0$ up to permutation.

Case 2. There are no identical rows or columns in U . By Proposition 14, $U \in \mathcal{B}_1$ up to permutation. \square

A.1 Binary Patterns that are either Row-paired or Column-paired

Definition 14. We define the set \mathcal{B}_0 of binary matrices as $\mathcal{B}_0 = \{A, B, C, D, E, F, G, H, I, J, K\}$, where

$$\begin{aligned}
 A &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}, \\
 D &= \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}, \quad E = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad F = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}, \\
 G &= \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad I = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}, \\
 J &= \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad K = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.
 \end{aligned}$$

Proposition 13. *Let $U \in \mathbb{Z}_2^{8 \times 8}$. Suppose U satisfies Lemmas 5 and 6. If U has two rows that are identical, then $U \in \mathcal{B}_0$ up to permutation and transposition.*

Proof. Let u_i denote the i -th column of U , and u_i^\dagger denote the i -th row of U , $0 \leq i < 8$. Let $\|v\|$ denote the hamming weight of v , where v is a string of binary bits. Up to permutation, suppose $\|u_0^\dagger\| = \|u_1^\dagger\|$. By Lemma 5, $\|u_0^\dagger\| = 8$ or $\|u_0^\dagger\| = 4$. Proceed by case distinction, we summarized the derivation of binary patterns in Figure 2 and Figure 3.

Case 1. $\|u_0^\dagger\| = \|u_1^\dagger\| = 8$.

Subcase 1.1. $\|u_0\| = 8$.

Subcase 1.1.1. $\|u_1\| = 8$.

Subcase 1.1.1.1. $\|u_2\| = 8$.

Subcase 1.1.1.1.1. $\|u_3\| = 8$.

Subcase 1.1.1.1.1.1. $\|u_4\| = 8$, then

[illegible]

Subcase 1.1.1.1.1.2. $\|u_4\| = 4$, then

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & & & \\ 1 & 1 & 1 & 1 & 1 & & & \\ 1 & 1 & 1 & 1 & 0 & & & \\ 1 & 1 & 1 & 1 & 0 & & & \\ 1 & 1 & 1 & 1 & 0 & & & \\ 1 & 1 & 1 & 1 & 0 & & & \end{bmatrix} \xrightarrow{\text{Lemma 5}} U = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & \textcolor{teal}{1} & \textcolor{teal}{1} & \textcolor{teal}{1} \\ 1 & 1 & 1 & 1 & 1 & \textcolor{teal}{1} & \textcolor{teal}{1} & \textcolor{teal}{1} \\ 1 & 1 & 1 & 1 & 0 & \textcolor{teal}{0} & \textcolor{teal}{0} & \textcolor{teal}{0} \\ 1 & 1 & 1 & 1 & 0 & \textcolor{teal}{0} & \textcolor{teal}{0} & \textcolor{teal}{0} \\ 1 & 1 & 1 & 1 & 0 & \textcolor{teal}{0} & \textcolor{teal}{0} & \textcolor{teal}{0} \\ 1 & 1 & 1 & 1 & 0 & \textcolor{teal}{0} & \textcolor{teal}{0} & \textcolor{teal}{0} \end{bmatrix} = B.$$

Subcase 1.1.1.1.2. $\|u_3\| = 4$. Let (x, y) be a pair of the entries in the i -th column of rows 2 and 3 as shown below, for $4 \leq i < 8$. By [Lemma 6](#) with $u_3, x = y$. Hence $u_2^\dagger = u_3^\dagger$.

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & x & & & \\ 1 & 1 & 1 & 1 & y & & & \\ 1 & 1 & 1 & 1 & 0 & & & \\ 1 & 1 & 1 & 1 & 0 & & & \\ 1 & 1 & 1 & 1 & 0 & & & \\ 1 & 1 & 1 & 1 & 0 & & & \end{bmatrix}.$$

Subcase 1.1.1.1.2.1. $\|u_2^\dagger\| = 8$, then

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & & & & \\ 1 & 1 & 1 & 0 & & & & \\ 1 & 1 & 1 & 0 & & & & \\ 1 & 1 & 1 & 0 & & & & \end{bmatrix} \xrightarrow{\text{Lemma 5}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} = B.$$

Subcase 1.1.1.1.2.2. $\|u_2^\dagger\| = 4$, then

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ \textcolor{red}{1} & \textcolor{red}{1} & \textcolor{red}{1} & \textcolor{red}{1} & 0 & 0 & 0 & 0 \\ \textcolor{red}{1} & \textcolor{red}{1} & \textcolor{red}{1} & \textcolor{red}{0} & & & & \\ 1 & 1 & 1 & 0 & & & & \\ 1 & 1 & 1 & 0 & & & & \\ 1 & 1 & 1 & 0 & & & & \end{bmatrix}.$$

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & x & & & & & \\ 1 & 1 & y & & & & & \\ 1 & 0 & & & & & & \\ 1 & 0 & & & & & & \\ 1 & 0 & & & & & & \\ 1 & 0 & & & & & & \end{bmatrix}.$$

Subcase 1.1.2.1. $\|u_2^\dagger\| = 8$, then

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & & & & & & \\ 1 & 0 & & & & & & \\ 1 & 0 & & & & & & \\ 1 & 0 & & & & & & \end{bmatrix} \xrightarrow{\text{Lemma 5}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & & & & & & \\ 1 & 0 & & & & & & \\ 1 & 0 & & & & & & \end{bmatrix} \xrightarrow{\text{Lemma 5}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} = B.$$

Subcase 1.1.2.2. $\|u_2^\dagger\| = 4$. By [Lemma 6](#) with row 3, for u_2 and u_3 , precisely one of them has hamming weight 8, and the other has hamming weight 4. Up to column permutation, let $\|u_2\| = 8$ and $\|u_3\| = 4$.

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & & & & \\ 1 & 0 & 1 & 0 & & & & \\ 1 & 0 & 1 & 0 & & & & \\ 1 & 0 & 1 & 0 & & & & \end{bmatrix} \xrightarrow{\text{Lemma 5}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & & & \\ 1 & 0 & 1 & 0 & 0 & & & \\ 1 & 0 & 1 & 0 & 0 & & & \end{bmatrix} \xrightarrow{\text{Lemma 6}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & & \\ 1 & 0 & 1 & 0 & 0 & & & \\ 1 & 0 & 1 & 0 & 0 & & & \end{bmatrix} \xrightarrow{\text{Lemma 5}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} = C.$$

Subcase 1.2. $\|u_0\| = 4$. Let (x, y) be a pair of the entries in the i -th column of rows 2 and 3 as shown below, for $1 \leq i < 8$. By [Lemma 6](#) with u_0 , $x = y$. Hence $u_2^\dagger = u_3^\dagger$.

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & x & & & & & & \\ 1 & y & & & & & & \\ 0 & & & & & & & \\ 0 & & & & & & & \\ 0 & & & & & & & \\ 0 & & & & & & & \end{bmatrix}.$$

Subcase 1.2.1. $\|u_2^\dagger\| = 8$. By [Lemma 5](#), $\|u_4^\dagger\| = 4$ or $\|u_4^\dagger\| = 0$. Then we have

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & & & & & & & \\ 0 & & & & & & & \\ 0 & & & & & & & \\ 0 & & & & & & & \end{bmatrix}.$$

Subcase 1.2.1.1. $\|u_4^\dagger\| = 4$, then

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & & & & & & & \\ 0 & & & & & & & \\ 0 & & & & & & & \end{bmatrix} \xrightarrow{\text{Lemma 5}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} = B.$$

Subcase 1.2.1.2. $\|u_4^\dagger\| = 0$, then

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & & & & & & & \\ 0 & & & & & & & \\ 0 & & & & & & & \end{bmatrix} \xrightarrow{\text{Lemma 5}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = K.$$

Subcase 1.2.2. $\|u_2^\dagger\| = 4$, then the binary matrix is shown below. By [Lemma 6](#), there can be two cases: precisely two of $\{u_1, u_2, u_3\}$ have hamming weight 8, or all of $\{u_1, u_2, u_3\}$ have hamming weight 4.

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & & & & & & & \\ 0 & & & & & & & \\ 0 & & & & & & & \\ 0 & & & & & & & \end{bmatrix}.$$

Subcase 1.2.2.1. Up to column permutation, let $\|u_1\| = \|u_2\| = 8$ and $\|u_3\| = 4$.

[illegible]

Subcase 1.2.2.2. $\|u_1\| = \|u_2\| = \|u_3\| = 4$.

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & & & & \\ 0 & 0 & 0 & 0 & & & & \\ 0 & 0 & 0 & 0 & & & & \\ 0 & 0 & 0 & 0 & & & & \end{bmatrix} \xrightarrow{\text{Lemma 5}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & & & \\ 0 & 0 & 0 & 0 & 1 & & & \\ 0 & 0 & 0 & 0 & 0 & & & \\ 0 & 0 & 0 & 0 & 0 & & & \end{bmatrix} \xrightarrow{\text{Lemma 5}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \end{bmatrix} = E.$$

Case 2. $\|u_0^\dagger\| = \|u_1^\dagger\| = 4$.

Subcase 2.1. $\|u_0\| = 8$.

Subcase 2.1.1. $\|u_1\| = 8$.

Subcase 2.1.1.1. $\|u_2\| = 8$, then

[illegible]

Subcase 2.1.1.1.1. $\|u_2^\dagger\| = 8$, then

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & & & & \\ 1 & 1 & 1 & 1 & & & & \\ 1 & 1 & 1 & 1 & & & & \\ 1 & 1 & 1 & 1 & & & & \\ 1 & 1 & 1 & 1 & & & & \end{bmatrix} \xrightarrow{\text{Lemma 5}} \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & & & \\ 1 & 1 & 1 & 1 & 1 & & & \\ 1 & 1 & 1 & 1 & 1 & & & \\ 1 & 1 & 1 & 1 & 0 & & & \\ 1 & 1 & 1 & 1 & 0 & & & \end{bmatrix} \xrightarrow{\text{Lemma 5}} \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} = B.$$

Subcase 2.1.1.1.2. $\|u_2^\dagger\| = 4$. By [Lemma 5](#), there can be two cases: precisely four of $\{u_3^\dagger, u_4^\dagger, u_5^\dagger, u_6^\dagger, u_7^\dagger\}$ have hamming weight 8, or all of $\{u_3^\dagger, u_4^\dagger, u_5^\dagger, u_6^\dagger, u_7^\dagger\}$ have hamming weight 4.

Subcase 2.1.1.1.2.1. Up to row permutation, $\|u_3^\dagger\| = \|u_4^\dagger\| = \|u_5^\dagger\| = \|u_6^\dagger\| = 8$.

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} = B.$$

Subcase 2.1.1.1.2.2. $\|u_3^\dagger\| = \|u_4^\dagger\| = \|u_5^\dagger\| = \|u_6^\dagger\| = \|u_7^\dagger\| = 4$.

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} = K^\top.$$

Subcase 2.1.1.2. $\|u_2\| = 4$. Let (x, y) be a pair of the entries in the i -th column of rows 2 and 3 as shown below, for $3 \leq i < 8$. By [Lemma 6](#) with $u_3, x = y$. Hence $u_2^\dagger = u_3^\dagger$.

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & & & & & \\ 1 & 1 & 1 & & & & & \\ 1 & 1 & 0 & & & & & \\ 1 & 1 & 0 & & & & & \\ 1 & 1 & 0 & & & & & \\ 1 & 1 & 0 & & & & & \end{bmatrix} \xrightarrow{\text{Lemma 6}} \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & x & & & \\ 1 & 1 & 1 & 1 & y & & & \\ 1 & 1 & 0 & & & & & \\ 1 & 1 & 0 & & & & & \\ 1 & 1 & 0 & & & & & \\ 1 & 1 & 0 & & & & & \end{bmatrix}.$$

Subcase 2.1.1.2.1. $\|u_2^\dagger\| = 8$, then

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & & & & \\ 1 & 1 & 0 & 0 & & & & \\ 1 & 1 & 0 & 0 & & & & \\ 1 & 1 & 0 & 0 & & & & \end{bmatrix} \xrightarrow{\text{Lemma 5}} \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & & & \\ 1 & 1 & 0 & 0 & 0 & & & \\ 1 & 1 & 0 & 0 & 0 & & & \end{bmatrix} \xrightarrow{\text{Lemma 6}} \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & & & \\ 1 & 1 & 0 & 0 & 0 & & & \\ 1 & 1 & 0 & 0 & 0 & & & \end{bmatrix}$$

$$\xrightarrow{\text{Lemma 5}} \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & & & \\ 1 & 1 & 0 & 0 & 0 & & & \end{bmatrix} \xrightarrow{\text{Lemma 5}} \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & \end{bmatrix} = C.$$

Subcase 2.1.1.2.2. $\|u_2^\dagger\| = 4$, then

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & & & & \\ 1 & 1 & 0 & 0 & & & & \\ 1 & 1 & 0 & 0 & & & & \\ 1 & 1 & 0 & 0 & & & & \end{bmatrix} \xrightarrow{\text{Lemma 5}} \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} = D.$$

Subcase 2.1.2. $\|u_1\| = 4$. Let x, y, z, w be the entries in U as shown below. By [Lemma 6](#) with row 1, $x = y$ and $z = w$. By [Lemma 6](#) with column 1, $x = z$ and $y = w$. Hence $x = y = z = w$. Moreover, since (x, z) can be any pair of the entries coming from any column i of rows 2 and 3, for $2 \leq i < 8$, we have $u_2^\dagger = u_3^\dagger$.

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & x & y & & & & \\ 1 & 1 & z & w & & & & \\ 1 & 0 & & & & & & \\ 1 & 0 & & & & & & \\ 1 & 0 & & & & & & \\ 1 & 0 & & & & & & \end{bmatrix}$$

Subcase 2.1.2.1. $x = y = z = w = 1$, then

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & & & & \\ 1 & 1 & 1 & 1 & & & & \\ 1 & 0 & & & & & & \\ 1 & 0 & & & & & & \\ 1 & 0 & & & & & & \\ 1 & 0 & & & & & & \end{bmatrix} \xrightarrow{\text{Lemma 6}} \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & & & & \\ 1 & 1 & 1 & 1 & & & & \\ 1 & 0 & 1 & 0 & & & & \\ 1 & 0 & & & & & & \\ 1 & 0 & & & & & & \\ 1 & 0 & & & & & & \end{bmatrix} \xrightarrow{\text{Lemma 5}} \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & & & & \\ 1 & 1 & 1 & 1 & & & & \\ 1 & 0 & 1 & 0 & & & & \\ 1 & 0 & 1 & 0 & & & & \\ 1 & 0 & 1 & 0 & & & & \\ 1 & 0 & 1 & 0 & & & & \end{bmatrix}$$

Subcase 2.1.2.1.1. $\|u_2^\dagger\| = 8$, then

[illegible]

Subcase 2.1.2.1.2. $\|u_2^\dagger\| = 4$, then

$$\begin{aligned}
 U = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & & & & \\ 1 & 0 & 1 & 0 & & & & \\ 1 & 0 & 1 & 0 & & & & \\ 1 & 0 & 1 & 0 & & & & \end{bmatrix} & \xrightarrow{\text{Lemma 5}} & \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & & & \\ 1 & 0 & 1 & 0 & 1 & & & \\ 1 & 0 & 1 & 0 & 1 & & & \end{bmatrix} & \xrightarrow{\text{Lemma 6}} \\
 \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & & \\ 1 & 0 & 1 & 0 & 1 & 1 & & \\ 1 & 0 & 1 & 0 & 1 & 1 & & \end{bmatrix} & \xrightarrow{\text{Lemma 5}} & \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} = D.
 \end{aligned}$$

Subcase 2.1.2.2. $x = y = z = w = 0$, then we have what follows.

$$\begin{aligned}
 U = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & & & & \\ 1 & 1 & 0 & 0 & & & & \\ 1 & 0 & & & & & & \\ 1 & 0 & & & & & & \\ 1 & 0 & & & & & & \\ 1 & 0 & & & & & & \end{bmatrix} & \xrightarrow{\text{Lemma 5}} & \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & & & & & \\ 1 & 0 & 1 & & & & & \\ 1 & 0 & 0 & & & & & \\ 1 & 0 & 0 & & & & & \end{bmatrix} & \xrightarrow{\text{Lemma 6}}
 \end{aligned}$$

Let (x, y) be a pair of the entries in the i -th column of rows 4 and 5 as shown below, for $4 \leq i < 8$. By Lemma 6 with u_2 , $x = y$. Hence $u_4^\dagger = u_5^\dagger$.

$$\begin{aligned}
 \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & x & & & \\ 1 & 0 & 1 & 0 & y & & & \\ 1 & 0 & 0 & 1 & & & & \\ 1 & 0 & 0 & 1 & & & & \end{bmatrix} & \xrightarrow{\text{Lemma 6}} & \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & & & & \\ 1 & 0 & 0 & 1 & & & & \end{bmatrix} & \xrightarrow{\text{Lemma 5}} & \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} = F.
 \end{aligned}$$

Subcase 2.2. $\|u_0\| = 4$. Let (x, y) be a pair of the entries in the i -th column of rows 2 and 3 as shown below, for $1 \leq i < 8$. By Lemma 6 with u_0 , $x = y$. Hence $u_2^\dagger = u_3^\dagger$. By Lemma 6 with u_0^\dagger , there must be odd many 1's in $\{x, z, w\}$. Up to column permutation, consider the following two cases: $x = z = w = 1$ or $x = 1, z = w = 0$.

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & x & z & w & & & & \\ 1 & y & & & & & & \\ 0 & & & & & & & \\ 0 & & & & & & & \\ 0 & & & & & & & \\ 0 & & & & & & & \end{bmatrix}.$$

Subcase 2.2.1. $x = z = w = 1$, we have

Subcase 2.2.1.1. $\|u_2^\dagger\| = 8$, then

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & & & & & & & \\ 0 & & & & & & & \\ 0 & & & & & & & \\ 0 & & & & & & & \end{bmatrix}.$$

By **Lemma 6** with u_0^\dagger , there must be evenly many columns among $\{u_0, u_1, u_2, u_3\}$ that have hamming weight 8. Since $\|u_0\| = 4$, up to column permutation, there can be two cases.

Subcase 2.2.1.1.1. $\|u_1\| = \|u_2\| = 8$ and $\|u_3\| = 4$.

$$\begin{aligned} U = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & & & & \\ 0 & 1 & 1 & 0 & & & & \\ 0 & 1 & 1 & 0 & & & & \\ 0 & 1 & 1 & 0 & & & & \end{bmatrix} &\xrightarrow{\text{Lemma 5}}& \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & \color{green}{1} & \color{green}{1} & \color{green}{0} & \color{green}{0} \\ 0 & 1 & 1 & 0 & \color{green}{1} & & & \\ 0 & 1 & 1 & 0 & \color{green}{0} & & & \\ 0 & 1 & 1 & 0 & \color{green}{0} & & & \end{bmatrix} &\xrightarrow{\text{Lemma 6}}& \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & \color{green}{1} & & \\ 0 & 1 & 1 & 0 & 0 & & & \\ 0 & 1 & 1 & 0 & 0 & & & \end{bmatrix} \\ &\xrightarrow{\text{Lemma 5}}& \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & \color{green}{1} & \color{green}{0} & \color{green}{0} \\ 0 & 1 & 1 & 0 & 0 & \color{green}{0} & & \\ 0 & 1 & 1 & 0 & 0 & \color{green}{0} & & \end{bmatrix} &\xrightarrow{\text{Lemma 5}}& \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & \color{green}{1} & \color{green}{1} & \\ 0 & 1 & 1 & 0 & 0 & \color{green}{1} & \color{green}{1} & \end{bmatrix} = C. \end{aligned}$$

Subcase 2.2.1.1.2. $\|u_1\| = \|u_2\| = \|u_3\| = 4$.

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & & & & \\ 0 & 0 & 0 & 0 & & & & \\ 0 & 0 & 0 & 0 & & & & \\ 0 & 0 & 0 & 0 & & & & \end{bmatrix} \xrightarrow{\text{Lemma 5}} \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & \color{green}{1} & \color{green}{1} & \color{green}{1} & \color{green}{1} \\ 0 & 0 & 0 & 0 & \color{green}{1} & \color{green}{1} & \color{green}{1} & \color{green}{1} \\ 0 & 0 & 0 & 0 & \color{green}{0} & \color{green}{0} & \color{green}{0} & \color{green}{0} \\ 0 & 0 & 0 & 0 & \color{green}{0} & \color{green}{0} & \color{green}{0} & \color{green}{0} \end{bmatrix} = E.$$

Subcase 2.2.1.2. $\|u_2^\dagger\| = 4$, then

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & & & & & & & \\ 0 & & & & & & & \\ 0 & & & & & & & \\ 0 & & & & & & & \end{bmatrix}.$$

By [Lemma 6](#) with u_2 , $x = y$. Hence $u_4^\dagger = u_5^\dagger$.

Lemma 6 \rightarrow

Subcase 2.2.2.2. $\|u_1\| = 4$, then

$$U =$$

By [Lemma 6](#) with $u_2, x = y$. Hence $u_4^\dagger = u_5^\dagger$. Moreover, by [Lemma 6](#) with u_0^\dagger , we have

$$U =$$

By [Lemma 6](#) with u_2^\dagger , $x = z$ and $y = w$. Since $x = y$ and $z = w$, $x = y = z = w$.

Subcase 2.2.2.2.1. $x = y = z = w = 1$

$$U =$$

Subcase 2.2.2.2.2. $x = y = z = w = 0$.

$$U =$$

☐

Subcase 1.2.1. There is a column with hamming weight 8, consider

[illegible]

Subcase 1.2.2. There is no column whose hamming weight is 8. Up to row and column permutation, consider

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & & & & & & & \\ 1 & & & & & & & \\ 0 & & & & & & & \\ 0 & & & & & & & \\ 0 & & & & & & & \\ 0 & & & & & & & \end{bmatrix} \xrightarrow[\text{Lemma 6}]{\text{Lemma 5}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & & & & & & & \\ 0 & & & & & & & \\ 0 & & & & & & & \\ 0 & & & & & & & \\ 0 & & & & & & & \end{bmatrix} \xrightarrow{\text{Lemma 6}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & & & & & & \\ 0 & 0 & & & & & & \\ 0 & 0 & & & & & & \\ 0 & 0 & & & & & & \\ 0 & 0 & & & & & & \end{bmatrix}.$$

Note that $u_0 = u_1$, but it contradicts our assumption that there are no identical columns in U . Thus this case is not possible.

Case 2. There is no row with hamming weight 8. Up to row and column permutation, $\|u_0^\dagger\| = 4$.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \xrightarrow{\text{Lemma 5}} \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} = M.$$

Subcase 2.2.2. $x = 0$

$$\begin{aligned}
U = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} & \xrightarrow{\text{Lemma 6}} & \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} & \xrightarrow[\text{Lemma 6}]{\text{Lemma 5}} \\
\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} & \xrightarrow{\text{Lemma 5}} & \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} = N.
\end{aligned}$$

Hence, when there are no identical rows nor columns in U , $U \in \mathcal{B}_1$ up to permutation.

□

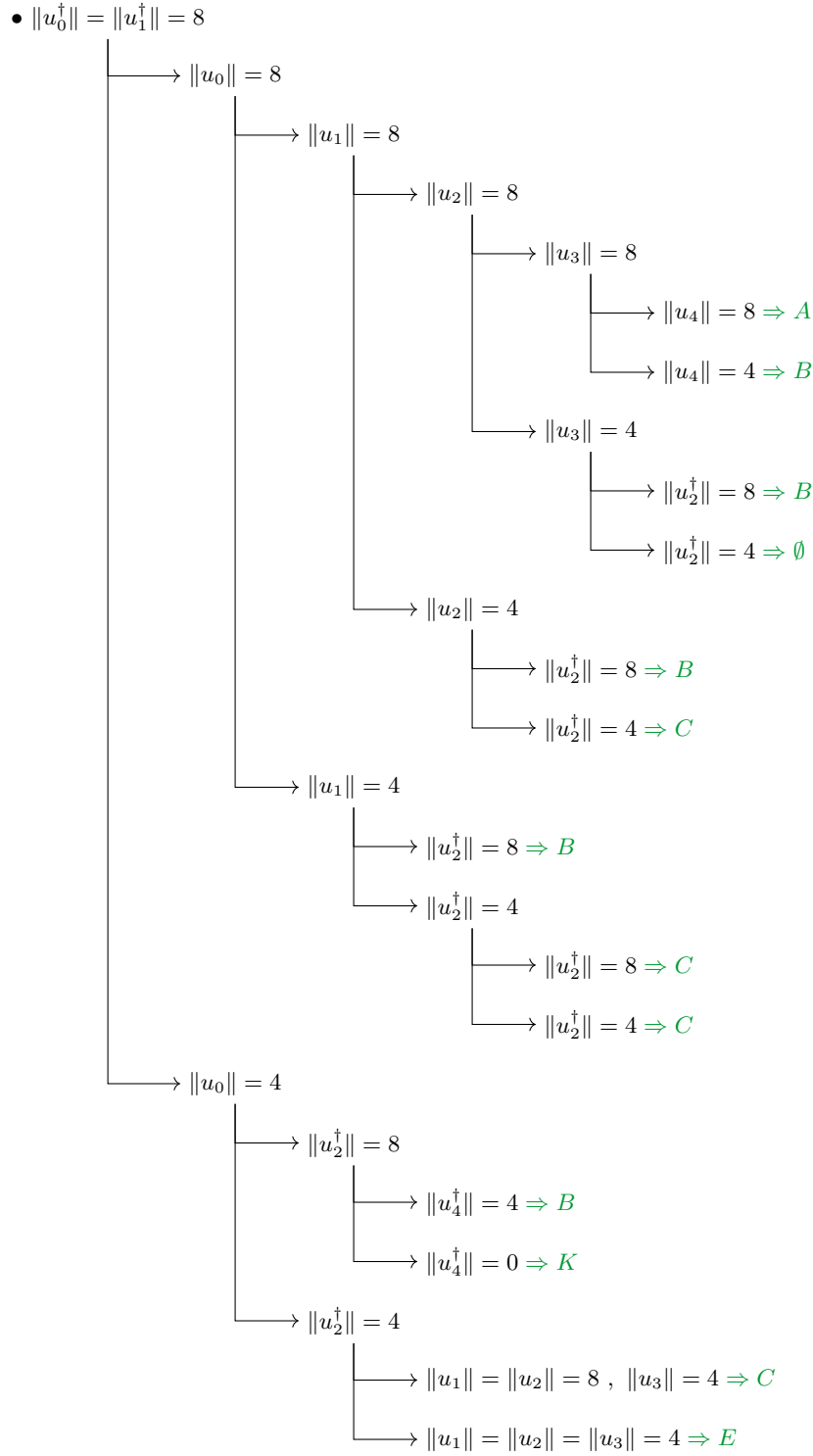


Figure 2. Case distinction for $\|u_0^\dagger\| = \|u_1^\dagger\| = 8$.

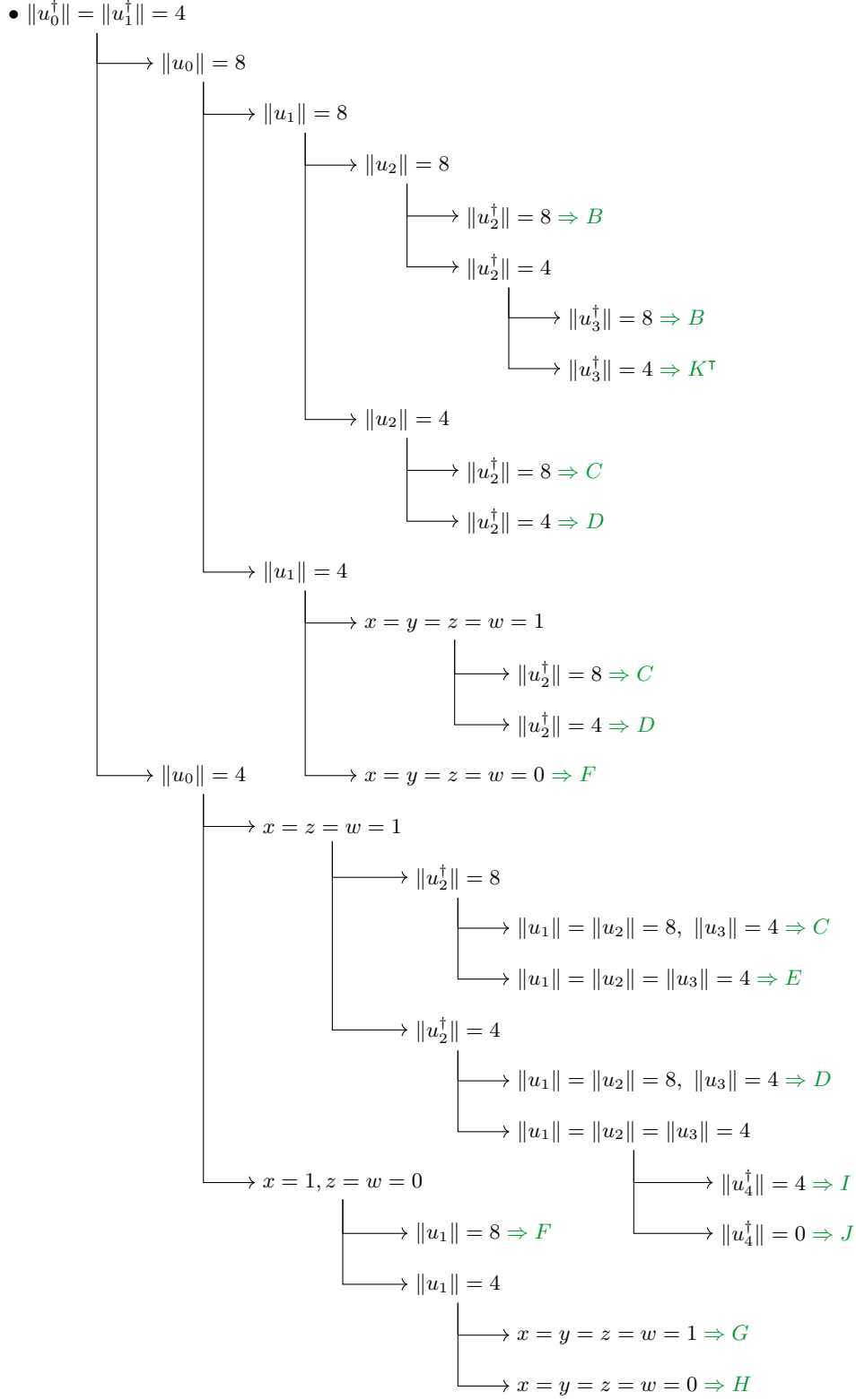


Figure 3. Case distinction for $\|u_0^\dagger\| = \|u_1^\dagger\| = 4$.