Quantum LDPC codes from intersecting subsets

Dimiter Ostrev

German Aerospace Center (DLR), Institute of Communications and Navigation, 82234 Weßling, Germany

Abstract

This paper introduces a construction of quantum CSS codes from a tuple of component CSS codes and two collections of subsets. The resulting codes have parallelizable encoding and syndrome measurement circuits and built-in redundancy in the syndrome measurements. In a certain subfamily of the general construction, the resulting codes are related to a natural generalization of classical Reed-Muller codes, and this leads to formulas for the distance of the quantum code as well as for the distance of the associated classical code that protects against errors in the syndrome. The paper gives a number of examples of codes with block size $2^m, m = 3, \ldots, 9$, and with syndrome measurements involving 2, 4 or 8 qubits. These include codes for which the distance exceeds the syndrome measurement weight, as well as codes which provide asymmetric protection against bit flip and phase flip errors.

1 Introduction

Quantum error correction [33, 35] protects quantum states from the effects of noise during communication or computation. Many of the codes studied so far can be described using the stabilizer formalism [13, 6].

In recent years, quantum low-density parity check codes have received considerable attention [23, 37, 17, 4, 29, 22, 16, 21, 9]. In these codes, each syndrome measurement involves only a few qubits, and this is considered favourable for fault-tolerance. Moreover, there is hope that low-complexity message passing decoders [12, 31] known from classical LDPC codes can be adapted to the quantum case.

Research on quantum LDPC codes can be classified according to various criteria. This introduction considers three: the ideas motivating the code design, the axis communication-fault tolerance, and the axis finite block length-asymtotic regime. The following paragraphs give a high-level description and some references for each, then explain where the present work stands with respect to each criterion.

Ideas motivating the code design The construction methods used for classical LDPC codes do not easily translate to their quantum counterparts, because of the constraint that stabilizer elements must commute. In recent years, much progress in quantum LDPC codes has been achieved using ideas from homological algebra, geometry and topology. Exploring in detail the many and varied results is beyond the scope of this introduction; fortunately, the recent survey [5] gives an excellent overview. Subsequent to the publication of that survey, the long term goal of constructing asymptotically good quantum LDPC codes was achieved by lifted product codes [29] and quantum Tanner codes [22, 16, 21, 9].

By contrast, the present paper uses a code design that is arguably much simpler and much closer to classical coding.

Consider the first construction of classical LDPC codes due to Gallager [12] in relationship to the earlier construction of classical product codes [10]. Specifically, think first of the parity check matrix of a classical product of single parity check codes. The resulting parity check matrix has layers of rows, with each layer having rows of disjoint support, whose union covers all columns. For example, the parity check matrix of the two-fold product of the [3,2,2] single parity check code with itself is

$$\begin{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \\ \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{pmatrix} = \left(\begin{array}{ccccccccc} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ \hline 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right)$$

Now, think of Gallager's construction of classical LDPC codes [12, Section 2.2 and Figure 2.1]. Gallager uses as a basic building block a layer of rows with disjoint support whose union covers all columns. He then builds the full parity check matrix from several such layers, each of which is a random column permutation of the first layer. Thus, Gallager's LDPC codes can be viewed as a generalization of products of single parity check codes, obtained by allowing more general column permutations for the layers.

How can Gallager's design using layers of rows be extended to quantum Calderbank-Shor-Steane codes? Since the parity check matrices for bit flips and phase flips must be orthogonal to each other, completely arbitrary column permutations of the layers are not possible. However, progress may be obtained by imposing some additional structure. Recently, Hivadi [18] showed how to construct a $[16s^2, 16s^2 - 16s + 2, 4]]$ quantum CSS code such that one of the parity check matrices is the classical 2-fold product of the $[4s, 4s - 1]$ single parity check code, and the other parity check matrix is a column permutation of the first one. Later, this was generalized in [27] to classical D-fold products for any D.

The present paper presents new quantum Calderbank-Shor-Steane codes that are a natural generalization of the product code construction of [27]. In the context of the preceding discussion, this generalization can be viewed as introducing greater flexibility in choosing column permutations for the layers,

while at the same time retaining sufficient structure to ensure that the two parity check matrices of the quantum CSS code are orthogonal. For reasons that will become clear later, this new family is called intersecting subset codes.

Using a design with layers of rows as described above automatically leads to parity check matrices that are not full rank: for example, the sum of rows in each layer is the all-ones vector. When the column permutations of different layers are random, it seems difficult to say much more about the linear dependence of the rows. However, it turns out that the additional structure present in intersecting subset codes is enough to compute explicitly a basis for the stablizer. Moreover, it turns out that this basis consists of rows of a tensor product of invertible matrices, which points to a surprising connection to polar [1] and Reed-Muller [30, 25] codes. Thus, there are at least two structured choices of a spanning set for the stabilizer of intersecting subset codes; the correspondence between them generalizes the connection between classical product and polar codes observed in [7]. From these two representations, a number of desirable properties of intersecting subset codes are derived.

The first choice of a spanning set for the stabilizer corresponds to the original motivation of obtaining parity check matrices similar to Gallager's design. These can be divided into layers, with the measurements in each layer done in parallel, and the layers measured in sequence. Moreover, for suitable examples of intersecting subset codes, each measurement involves only a few qubits.

The second choice of a spanning set for the stabilizer consists of subsets of the rows of a tensor product of invertible matrices and its inverse transpose. From this, an encoding circuit follows, which can also be divided into layers, with operations in each layer done in parallel, and the different layers applied in sequence. The second representation is used to derive independent generators for the normalizer and a canonical choice for the logical operators. Moreover, in a certain subfamily of intersecting subset codes, the encoding circuit simplifies to the same recursive pattern of CNOT gates that is used to encode classical Reed-Muller and polar codes. In this subfamily, the X and Z parity check matrices are related to a natural generalization of classical Reed-Muller codes, and this allows one to derive a formula for the quantum code distance.

It is worth noting that while classical Reed-Muller codes have been used previously to construct quantum stabilizer codes [19, 36, 32], the methods proposed there lead to codes with stabilizer weight exceeding the weight of logical operators. It is only through generalizing classical Reed-Muller codes that the present work is able to obtain examples with distance exceeding the syndrome measurement weight.

Communication or fault-tolerance In the standard model for communications, encoding and decoding are considered as perfect operations, and errors come only from the noisy channel in between. This model is easiest to work with, but does not accurately capture the imperfections of quantum hardware. The vast majority of previous works on quantum LDPC codes focus on this case.

On the other end of the specturm is the standard model for fault-tolerance. Here, a specific compilation of encoding and decoding in terms of elementary gates must be considered, and each component of the circuits introduces errors. This model is very difficult to work with and consequently there are very few works on quantum LDPC codes that deal directly with fault tolerance. Remarkable is the work [14], which shows that if a quantum LDPC family that satisfies certain condition exists, then fault tolerant quantum computation is possible with asymptotically constant overhead. Later, [11] showed that a hypergraph product [37] of classical expander codes [34] satisfies the assumtions of [14]. More recently, [3] has constructed a fault-tolerant quantum memory based on tensor product generalized bicycle codes [20]. Another interesting recent work is [8], which reduces the problem of correcting faults in a circuit of Clifford gates and Pauli measurements to the problem of correcting errors in a stabilizer code and identifies conditions under which the resulting code is LDPC.

A central issue in the study of fault-tolerance with quantum LDPC codes is the ability to correct errors even when syndrome extraction is noisy [14, 11, 3]. Recently, an intermediate error model has received attention [2, 26], in which syndrome outcomes are noisy but the full error propagation in the syndrome measurement circuit is not considered. The advantage of this intermediate model is that it is easier to work with than the full-circuit error model, while still giving some insight in the fault-tolerance properties of the studied error correcting codes.

Along the axis communication versus fault-tolerance, the present work falls at the intermediate stage of considering syndrome errors. The parity check matrices of intersecting subset codes have linearly dedendent rows. Correctly extracted syndromes always fall in the images of the parity check matrices, which can be viewed as classical linear error correcting codes. The distances of these spaces of valid syndromes are measures of the ability to correct errors in the syndrome. These distances can be computed explicitly for a large subfamily of the general construction, owing to the connection to generalized Reed-Muller codes mentioned earlier.

Moreover, the present work takes an initial step towards fault-tolerance, by providing an explicit compilation of encoding and syndrome measurements into elementary gates. In general, compilation into a circuit where many gates are performed in parallel is considered favorable for fault-tolerance, because the resulting total circuit depth is lower and there is less time for errors to accumulate. The special structure of intersecting subset codes implies that certain parallelization in the circuits for encoding and syndrome measurements is possible.

Finite block length versus asymptotic regime The focus of much recent work has been on properties of asymptotic families of LDPC codes. However, it has been noted that asymptotic constructions do not necessarily produce the best quantum LDPC codes for small or medium block lengths [28]. As a concrete example of this observation relevant to the present work, [27] shows

that a $[[512, 174, 8]]$ quantum CSS code obtained from the 3D classical product of the $[8, 7, 2]$ single parity check code has much better empirical performance than a quantum Tanner code with similar block size and rate.

The present work focuses on block sizes of a few tens or a few hundreds of qubits. There are two reasons for this choice. First, these numbers reflect the current limitations of quantum hardware. Second, modern classical communications systems continue to use these block sizes in some cases, even though there now exists classical hardware capable of handling much longer error correcting codes. Therefore, quantum LDPC codes with small and medium block sizes are interesting at present and are likely to remain so even in a hypothetical future with much better quantum hardware.

The rest of the paper is structured as follows. Section 2 covers notational conventions, background material, and some preparatory technical results. Then, section 3 gives the general case of intersecting subset codes and derives various properties. Section 4 considers a subfamily, defines a generalization of classical Reed Muller codes, and uses these to derive a formula for the distance of the quantum codes as well as the distances of the classical linear codes protecting against syndrome errors. Section 5 contains an extensive collection of examples with block sizes 2^m, $m = 3, \ldots, m$ and with syndrome measurements on 2, 4, or 8 qubits. Section 6 concludes the paper and gives some possible directions for future work.

2 Preliminaries

2.1 Conventions for row and column indexing

For an $l \times n$ matrix A, the rows will be indexed by $[l] = \{0, 1, \ldots, l-1\}$ and the columns will be indexed by $[n] = \{0, 1, \ldots, n-1\}$ Note that sometimes $[n]$ is used to denote the set $\{1, \ldots, n\}$, but in the present paper it will be more convenient to start at 0.

For a tuple of matrices A_i with respective sizes $l_i \times n_i$, rows and columns of $A_0 \otimes A_1 \otimes \cdots \otimes A_v$ will be indexed resepctively by $[\vec{l}\,] = [l_0] \times \cdots \times [l_v]$ and $[\vec{n}] = [n_0] \times \cdots \times [n_v]$.

If a linear indexing of rows and columns of $A_1 \otimes \cdots \otimes A_v$ is desired, this will be done lexicographically. Thus, the block decomposition

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \otimes E = \begin{pmatrix} A \otimes E & B \otimes E \\ C \otimes E & D \otimes E \end{pmatrix} \tag{1}$$

holds.

2.2 Conventions for subsets and indicator vectors

To $S \subset [n]$ is associated a $1 \times n$ indicator vector that has 1 in position i if $i \in S$ and zero otherwise. The indicator vector will also be denoted by S.

To a tuple

$$\mathbf{S} = \begin{pmatrix} S_0 \\ \vdots \\ S_{u-1} \end{pmatrix} \tag{2}$$

of subsets of $[n]$ is associated the $u \times n$ matrix of zeros and ones with rows $S_0, \ldots, S_{u-1}$. This indicator matrix will also be denoted by $\mathbf{S}$.

2.3 Incomplete permutation matrices

Take $l, n \in \mathbb{N}, S \subset [n]$, injective $\alpha : S \to [l]$ To l, n, S, α associate the matrix $\Pi(S) \in \mathbb{F}_2^{l \times n}$ that has one in positions $(\alpha(i), i), i \in S$ and zero everywhere else. To keep the notation simple, l, n, α are left implicit. One may call $\Pi(S)$ an incomplete permutation matrix with column support S.

An immediate consequence of the definition is $\Pi(S_1) \otimes \Pi(S_2) = \Pi(S_1 \times S_2)$. Indeed, both $\Pi(S_1) \otimes \Pi(S_2)$ and $\Pi(S_1 \times S_2)$ are zero everywhere except in positions with row index $(\alpha_1(i), \alpha_2(j))$ and column index (i, j) for $i \in S_1, j \in S_2$.

Besides this, the following will also be useful:

Theorem 1. *Take $n, l_0, \ldots, l_k \in \mathbb{N}$ and let $l = \sum_{i=0}^{k} l_i$. Take $S_0, \ldots S_k \subset [n]$ and let $T = \cup_{i=0}^{k} S_i$. Take injective $\alpha_i : S_i \to [l_i]$ and $\alpha : T \to [l]$. Then, there exists invertible $\Lambda(\mathbf{S}) \in \mathbb{F}_2^{l \times l}$ such that*

$$\begin{pmatrix} \Pi(S_0) \\ \vdots \\ \Pi(S_k) \end{pmatrix} = \Lambda(\mathbf{S}) \Pi(T) \tag{3}$$

where, to keep the notation simple, the dependence of Λ on $n, l_0, \ldots, l_k, \alpha_0, \ldots, \alpha_k, \alpha$ is left implicit.

Proof. Let $a_0 < a_1 < \cdots < a_r$ be the elements of T. For $j = 0, \ldots, r$, let $b_j = |\{i : a_j \in S_i\}|$ be the number of subsets in which a_j appears.

The matrix $\Lambda(\mathbf{S})$ is obtained as a product of two permutation matrices and a lower triangular matrix:

1. The first permutation matrix shuffles the rows of $\Pi(T)$ so that the non-zero elements are in positions $(j, a_j), j = 0, \ldots, r$.
2. The lower triangular matrix copies $(b_j - 1)$ times row j for $j = 0, \ldots, r$.
3. The second permutation matrix shuffles the rows to their correct final positions, specified by the injections $\alpha_0, \ldots, \alpha_k$ and the order of the sets $S_0, \ldots, S_k$.

□

Example: Let $n = 4$, $l_0 = l_1 = 2$. Then, $l = 4$. Let the tuple $\mathbf{S}$ consist of $S_0 = \{0, 1\}$ and $S_1 = \{0, 2\}$. Then, $T = \{0, 1, 2\}$. Let the injection α_0 :

$S_0 \to [2]$ be $\alpha_0(0) = 0, \alpha_0(1) = 1$. Then, the incomplete permutation matrix corresponding to S_0, α_0 is

$$\Pi(S_0) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{4}$$

Let the injection $\alpha_1 : S_1 \to [2]$ be $\alpha_1(0) = 0, \alpha_1(2) = 1$. Then, the incomplete permutation matrix corresponding to S_1, α_1 is

$$\Pi(S_1) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{5}$$

Let the injection $\alpha : T \to [4]$ be $\alpha(0) = 0, \alpha(1) = 1, \alpha(2) = 2$. Then, the incomplete permutation matrix corresponding to T, α is

$$\Pi(T) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{6}$$

Applying the above argument to $\Pi(S_0), \Pi(S_1), \Pi(T)$ gives $a_0 = 0, a_1 = 1, a_2 = 2$ as the elements of T in increasing order, $b_0 = 2, b_1 = 1, b_2 = 1$ as the number of subsets in which they appear, and

$$\begin{pmatrix} \Pi(S_0) \\ \Pi(S_1) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \Pi(T) \tag{7}$$

as the two permutation matrices and the lower triangular matrix that relate $\Pi(T)$ to $\Pi(S_0), \Pi(S_1)$.

2.4 Joint decomposition of a pair of orthogonal matrices

An important tool in linear algebra is the decomposition of a matrix into a product of two permutation matrices, a lower triangular, an upper triangular, and a diagonal matrix. This section shows that a joint decomposition exists for a pair of orthogonal matrices.

Theorem 2. *Take $A \in \mathbb{F}_2^{m_a \times n}, B \in \mathbb{F}_2^{m_b \times n}$ such that $AB^T = 0$. Denote the ranks of A, B by r_a, r_b. Then, there exist*

1. *permutation matrices $P_a \in \mathbb{F}_2^{m_a \times m_a}, P_b \in \mathbb{F}_2^{m_b \times m_b}, Q \in \mathbb{F}_2^{n \times n}$,*
2. *lower triangular $L_a \in \mathbb{F}_2^{m_a \times m_a}$ with ones along the diagonal.*
3. *upper triangular $L_b \in \mathbb{F}_2^{m_b \times m_b}, R \in \mathbb{F}_2^{n \times n}$ with ones along the diagonal.*
4. *$D_a \in \mathbb{F}_2^{m_a \times n}$ that has a one in the (i, i) entry for $i = 0, \dots r_a - 1$ and zero everywhere else.*

5. $D_b \in \mathbb{F}_2^{m_b \times n}$ *that has a one in the* $(m_b - r_b + i, n - r_b + i)$ *entry for* $i = 0, \ldots r_b - 1$ *and zero everywhere else.*

such that

$$P_a A Q = L_a D_a R \tag{8}$$
$$P_b B Q = L_b D_b R^{-T} \tag{9}$$

where $R^{-T} = (R^{-1})^T$ *is the transpose of the inverse of* R*.*

Proof. If both A, B are zero, the theorem holds trivially.

If one of A, B is zero, the claim follows from the usual decomposition of the non-zero matrix.

If both A, B are non-zero, then take i, j such that entry i, j of A is one, and take k such that row k of B is non-zero. The assumption $AB^T = 0$ implies row i of A and row k of B are orthogonal. Then, there exists $l \neq j$ such that entry k, l of B is one. Then, there exist permutation matrices P_a, P_b, Q such that $P_a A Q$ and $P_b B Q$ have the block form

$$P_a A Q = \begin{pmatrix} 1 & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \end{pmatrix} \tag{10}$$
$$P_b B Q = \begin{pmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & 1 \end{pmatrix} \tag{11}$$

where the number of rows in the blocks is $(1, m_a - 1)$ for A and $(m_b - 1, 1)$ for B, and the number of columns in the blocks is $(1, n - 2, 1)$ for both.

Now take

$$R = \begin{pmatrix} 1 & A_{12} & B_{21} \\ 0 & I & B_{22}^T \\ 0 & 0 & 1 \end{pmatrix} \tag{12}$$

where the number of rows and of columns in the blocks is $(1, n - 2, 1)$.

Note that $AB^T = 0$ implies

$$0 = P_a A Q Q^T B^T P_b^T = \begin{pmatrix} 1 & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \end{pmatrix} \begin{pmatrix} B_{11}^T & B_{21}^T \\ B_{12}^T & B_{22}^T \\ B_{13}^T & 1 \end{pmatrix} \tag{13}$$

The resulting relations for the blocks A_{ij}, B_{kl} can be used to verify that

$$R^{-T} = \begin{pmatrix} 1 & 0 & 0 \\ A_{12}^T & I & 0 \\ A_{13} & B_{22} & 1 \end{pmatrix} \tag{14}$$
$$P_a A Q R = \begin{pmatrix} 1 & 0 & 0 \\ A_{21} & A_{21} A_{12} + A_{22} & 0 \end{pmatrix} \tag{15}$$
$$P_b B Q R^{-T} = \begin{pmatrix} 0 & B_{12} + B_{13} B_{22} & B_{13} \\ 0 & 0 & 1 \end{pmatrix} \tag{16}$$

Now, take

$$L_a = \begin{pmatrix} 1 & 0 \\ A_{21} & I \end{pmatrix} \tag{17}$$

$$L_b = \begin{pmatrix} I & B_{13} \\ 0 & 1 \end{pmatrix} \tag{18}$$

where the number of rows and columns in the blocks is $(1, m_a - 1)$ for L_a and $(m_b - 1, 1)$ for L_b.

Note that

$$L_a P_a A Q R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & A_{21}A_{12} + A_{22} & 0 \end{pmatrix} \tag{19}$$

$$L_b P_b B Q R^{-T} = \begin{pmatrix} 0 & B_{12} + B_{13}B_{22} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{20}$$

From here, the proof can be completed by induction on $max(m_a, m_b, n)$. First, note that

$$L_a P_a A Q R (L_b P_b B Q R^{-T})^T = L_a P_a A B^T P_b^T L_b^T = 0$$

and therefore $(A_{21}A_{12} + A_{22})(B_{12} + B_{13}B_{22})^T = 0$. By the induction hypothesis,

$$L_a P_a A Q R = \begin{pmatrix} 1 & 0 \\ 0 & P_a'^{-1} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & L_a' \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & D_a' & 0 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 \\ 0 & R' & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & Q'^{-1} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{21}$$

$$L_b P_b B Q R^{-T} = \begin{pmatrix} P_b'^{-1} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} L_b' & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & D_b' & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 \\ 0 & R'^{-T} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & Q'^{-1} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{22}$$

for suitable $L_a', L_b', R', P_a', P_b', Q', D_a', D_b'$.

Now, move P_a', P_b', Q' to the other side, past L_a, L_b, R, R^{-T} and merge them with P_a, P_b, Q. Moving P_a', P_b', Q' past L_a, L_b, R, R^{-T} is acheived using the

relations:

$$\begin{pmatrix} 1 & 0 \\ 0 & P'_a \end{pmatrix} \begin{pmatrix} 1 & 0 \\ A_{21} & I \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ P'_a A_{21} & I \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & P'_a \end{pmatrix} \tag{23}$$

$$\begin{pmatrix} P'_b & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} I & B_{13} \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} I & P'_b B_{13} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} P'_b & 0 \\ 0 & 1 \end{pmatrix} \tag{24}$$

$$\begin{pmatrix} 1 & A_{12} & B_{21} \\ 0 & I & B_{22}^T \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & Q' & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & Q' & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & A_{12}Q' & B_{21} \\ 0 & I & (Q')^{-1} B_{22}^T \\ 0 & 0 & 1 \end{pmatrix} \tag{25}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ A_{12}^T & I & 0 \\ A_{13} & B_{22} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & Q' & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & Q' & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ (Q')^{-1} A_{12}^T & I & 0 \\ A_{13} & B_{22}Q' & 1 \end{pmatrix} \tag{26}$$

Finally, move the so transformed L_a, L_b, R, R^{-T} to the other side and merge them with $L'_a, L'_b, R', {R'}^{-T}$. This completes the inductive step and the proof. □

2.5 Matrices with layers of tensor products and their decompositions

Take a tuple of matrices $\mathbf{H} = (H_0, \ldots, H_{m-1}), \quad H_i \in \mathbb{F}_2^{l_i \times n_i}, i = 0, \ldots, m-1$. Take a subset $X \subset [m]$. To the pair $\mathbf{H}, X$ associate the matrix

$$M(\mathbf{H}, X) = \otimes_{i=0}^{m-1} \begin{cases} H_i & \text{if } i \in X \\ I_{n_i} & \text{otherwise} \end{cases} \tag{27}$$

Example: $m = 2, H_0 = H_1 = \begin{pmatrix} 1 & 1 \end{pmatrix}, X = \{0\}$,

$$M(\mathbf{H}, X) = \begin{pmatrix} 1 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \tag{28}$$

Now, suppose that tuples of permutation matrices $\mathbf{P}, \mathbf{Q}$, tuples of invertible matrices $\mathbf{L}, \mathbf{R}$, and a tuple of diagonal matrices $\mathbf{D}$ are known such that the decompositions $P_i H_i Q_i = L_i D_i R_i, \quad i = 0, \ldots, m-1$ hold. From these, obtain the decomposition

$$P(\mathbf{H}, X) M(\mathbf{H}, X) Q(\mathbf{H}) = L(\mathbf{H}, X) D(\mathbf{H}, X) R(\mathbf{H}) \tag{29}$$

where

$$P(\mathbf{H}, X) = \otimes_{i=0}^{m-1} \begin{cases} P_i & \text{if } i \in X \\ Q_i^{-1} & \text{otherwise} \end{cases} \tag{30}$$

$$Q(\mathbf{H}) = \otimes_{i=0}^{m-1} Q_i \tag{31}$$

$$L(\mathbf{H}, X) = \otimes_{i=0}^{m-1} \begin{cases} L_i & \text{if } i \in X \\ R_i^{-1} & \text{otherwise} \end{cases} \tag{32}$$

$$D(\mathbf{H}, X) = \otimes_{i=0}^{m-1} \begin{cases} D_i & \text{if } i \in X \\ I_{n_i} & \text{otherwise} \end{cases} \tag{33}$$

$$R(\mathbf{H}) = \otimes_{i=0}^{m-1} R_i \tag{34}$$

To keep the notation simple, assume that the particular decompositions used for the matrices in $\mathbf{H}$ are known from context and do not need to be specified explicitly.

Example: Take the decomposition

$$\begin{pmatrix} 1 & 1 \end{pmatrix} = 1 \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \tag{35}$$

and take $m = 2, H_0 = H_1 = \begin{pmatrix} 1 & 1 \end{pmatrix}, X = \{0\}$ as before. Then,

$$\begin{aligned} M(\mathbf{H}, X) &= \begin{pmatrix} 1 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ &= \left(1 \otimes \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \right) \left(\begin{pmatrix} 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right) \left(\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \right) \end{aligned} \tag{36}$$

Now, take a tuple of matrices $\mathbf{H}$ as before, but this time take a tuple of subsets of $[m]$

$$\mathbf{X} = \begin{pmatrix} X_0 \\ \vdots \\ X_{k-1} \end{pmatrix}, \quad X_i \subset [m], i = 0, \ldots, k-1 \tag{37}$$

To the pair $\mathbf{H}, \mathbf{X}$, associate the matrix

$$M(\mathbf{H}, \mathbf{X}) = \begin{pmatrix} M(\mathbf{H}, X_0) \\ \vdots \\ M(\mathbf{H}, X_{k-1}) \end{pmatrix} \tag{38}$$

obtained by taking $M(\mathbf{H}, X_0), \ldots, M(\mathbf{H}, X_{k-1})$ as layers of rows.

The matrix $M(\mathbf{H}, \mathbf{X})$ when $\mathbf{X}$ is a tuple of subsets can also be decomposed as

$$P(\mathbf{H}, \mathbf{X}) M(\mathbf{H}, \mathbf{X}) Q(\mathbf{H}) = L(\mathbf{H}, \mathbf{X}) D(\mathbf{H}, \mathbf{X}) R(\mathbf{H}) \tag{39}$$

where

$$P(\mathbf{H},\mathbf{X}) = \begin{pmatrix} P(\mathbf{H},X_0) & & \\ & \ddots & \\ & & P(\mathbf{H},X_{k-1}) \end{pmatrix} \tag{40}$$

$$Q(\mathbf{H}) = \otimes_{i=0}^{m-1} Q_i \tag{41}$$

$$L(\mathbf{H},\mathbf{X}) = \begin{pmatrix} L(\mathbf{H},X_0) & & \\ & \ddots & \\ & & L(\mathbf{H},X_{k-1}) \end{pmatrix} \Lambda(\mathbf{S}) \tag{42}$$

$$D(\mathbf{H},\mathbf{X}) = \Pi(T) \tag{43}$$

$$R(\mathbf{H}) = \otimes_{i=0}^{m-1} R_i \tag{44}$$

and where S_i is the column support of $D(\mathbf{H},X_i)$ viewed as an incomplete permutation matrix, $T = \cup_i S_i$, and $\Lambda(\mathbf{S})$ is defined in Theorem 1 in section 2.3.

Example: Take the decomposition

$$\begin{pmatrix} 1 & 1 \end{pmatrix} = 1 \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \tag{45}$$

and take $m = 2, H_0 = H_1 = \begin{pmatrix} 1 & 1 \end{pmatrix}$ as before. Take $X_0 = \{0\}, X_1 = \{1\}$. The matrix $M(\mathbf{H},\mathbf{X})$ has two layers; decomposing each layer separately gives

$$M(\mathbf{H},\mathbf{X}) = \begin{pmatrix} M(\mathbf{H},X_0) \\ M(\mathbf{H},X_1) \end{pmatrix} = \begin{pmatrix} L(\mathbf{H},X_0) & 0 \\ 0 & L(\mathbf{H},X_1) \end{pmatrix} \begin{pmatrix} D(\mathbf{H},X_0) \\ D(\mathbf{H},X_1) \end{pmatrix} R(\mathbf{H}) \tag{46}$$

The matrix

$$D(\mathbf{H},X_0) = \begin{pmatrix} 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{47}$$

is an incomplete permutation matrix with column support $S_0 = \{0,1\}$. The matrix

$$D(\mathbf{H},X_1) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{48}$$

is an incomplete permutation matrix with column support $S_1 = \{0,2\}$. The example after Theorem 1 in section 2.3 shows that

$$\begin{pmatrix} D(\mathbf{H},X_0) \\ D(\mathbf{H},X_1) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \Lambda(\mathbf{S})\Pi(T) \tag{49}$$

Thus, the decomposition of $M(\mathbf{H},\mathbf{X})$ from equation (39) has $P(\mathbf{H},\mathbf{X}) = Q(\mathbf{H}) =$

I_4, and

$$L(\mathbf{H},\mathbf{X}) = \begin{pmatrix} L(\mathbf{H},X_0) & 0 \\ 0 & L(\mathbf{H},X_1) \end{pmatrix} \Lambda(\mathbf{S}) = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{50}$$

$$D(\mathbf{H},\mathbf{X}) = \Pi(T) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{51}$$

$$R(\mathbf{H}) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{52}$$

2.6 The partial order on integer tuples

For $x, y \in \mathbb{Z}^n$, $x < y \Leftrightarrow \forall i, x_i \leq y_i$ AND $\exists i, x_i < y_i$ and this defines a partial order on $\mathbb{Z}^n$; for example, $(1,2,3) < (2,3,4) < (3,4,5)$, but $(1,2,3)$ and $(4,3,2)$ are incomparable.

Now, take $n_0, \ldots, n_{k-1} \in \mathbb{N}$ and consider the set $[\vec{n}] = [n_0] \times [n_2] \times \cdots \times [n_{k-1}]$ This set inherits from $\mathbb{Z}^k$ the partial order.

A subset $T \subset [\vec{n}]$ is called increasing if $x \in T$ AND $x < y \Rightarrow y \in T$ and decreasing if $x \in T$ AND $y < x \Rightarrow y \in T$.The minimal elements of T are $\mathcal{MIN}(T) = \{x \in T : \forall y \in T, \mathrm{NOT}(y < x)\}$ and the maximal elements of T are $\mathcal{MAX}(T) = \{x \in T : \forall y \in T, \mathrm{NOT}(x < y)\}$. The increasing subset generated by T is $\langle T \rangle_{\uparrow} = \{x \in S : \exists y \in T, y \leq x\}$ and the decreasing subset generated by T is $\langle T \rangle_{\downarrow} = \{x \in S : \exists y \in T, x \leq y\}$.

2.7 Quantum stabilizer codes

2.7.1 The Pauli group

The Pauli matrices are

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{53}$$

The Pauli group on n qubits is

$$\mathcal{P}_n = \left\{ w\sigma_x^{u_1}\sigma_z^{v_1} \otimes \cdots \otimes \sigma_x^{u_n}\sigma_z^{v_n} : w \in \{\pm 1, \pm i\}, u, v \in \mathbb{F}_2^{1\times n} \right\}$$

The shorthand notation $\sigma_x^u \sigma_z^v = \sigma_x^{u_1}\sigma_z^{v_1} \otimes \cdots \otimes \sigma_x^{u_n}\sigma_z^{v_n}$ will be used below.

The map $\Phi : \mathcal{P}_n \to \mathbb{F}_2^{1\times 2n}, \Phi(w\sigma_x^u\sigma_z^v) = \begin{pmatrix} u & v \end{pmatrix}$ is a surjective group homomorphism with kernel $\{\pm I, \pm iI\}$. Two elements $w\sigma_x^u\sigma_z^v, w'\sigma_x^{u'}\sigma_z^{v'}$ of the Pauli group anti-commute if $\Phi(w\sigma_x^u\sigma_z^v)\Omega\Phi(w'\sigma_x^{u'}\sigma_z^{v'})^T = 1$ and commute if $\Phi(w\sigma_x^u\sigma_z^v)\Omega\Phi(w'\sigma_x^{u'}\sigma_z^{v'})^T = 0$ where

$$\Omega = \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix} \tag{54}$$

2.7.2 Stabilizer codes

An abelian subgroup $\mathcal{S}$ of $\mathcal{P}_n$ that does not contain $-I$ is called a stabilizer subgroup. The joint $+1$-eigenspace of the elements of $\mathcal{S}$ is a quantum error correcting code. If elements $s_1, \ldots, s_m \in \mathcal{S}$ generate $\mathcal{S}$, then the matrix H with rows $\Phi(s_1), \ldots, \Phi(s_m)$ is a parity check matrix for the code stabilized by $\mathcal{S}$. If a quantum state is encoded in the subspace stablized by $\mathcal{S}$, a Pauli error $E \in \mathcal{P}_n$ occurs, and $s_1, \ldots, s_m$ are measured, then the resulting syndrome is $H\Omega\Phi(E)^T$.

Conversely, any matrix H such that $H\Omega H^T = 0$ is the parity check matrix of some stabilizer code.

2.7.3 The Clifford group and the symplectic group

The Clifford group on n qubits is the normalizer of the Pauli group in the $2^n \times 2^n$ unitary matrices: $\mathcal{CL}_n = \mathcal{N}_{\mathcal{U}_{2^n}}(\mathcal{P}_n) = \{U \in \mathcal{U}_{2^n} : U\mathcal{P}_n = \mathcal{P}_n U\}$. Each Clifford group element can be expressed as a product of CNOT, Hadamard and Phase gates and a global phase.

The symplectic group consists of matrices which preserve the symplectic product: $\mathcal{SP}_{2n} = \left\{A \in \mathbb{F}_2^{2n \times 2n} : A\Omega A^T = \Omega\right\}$.

There is a surjective group homomorphism $\Psi : \mathcal{CL}_n \to \mathcal{SP}_{2n}$ with the property:

$$\forall P \in \mathcal{P}_n,\ \forall U \in \mathcal{CL}_n, \quad \Phi(U^{-1}PU) = \Phi(P)\Psi(U) \tag{55}$$

2.7.4 Encoding circuits for stabilizer codes

For $k \leq n \in \mathbb{N}$, and for $U \in \mathcal{CL}_n$, the Pauli group elements $U^\dagger \sigma_{zi} U, i = 1, \ldots, n-k$ generate a stabilizer subgroup of $\mathcal{P}_n$, where σ_{zi} denotes σ_z acting on the i-th qubit.

Conversely, for each stabilizer subgroup $\mathcal{S}$ of $\mathcal{P}_n$, there exist many $U \in \mathcal{CL}_n$ such that $U^\dagger \sigma_{zi} U, i = 1, \ldots, n-k$ generate $\mathcal{S}$. Given one such U, an encoding circuit for the code stabilized by $\mathcal{S}$ is the following:

1. Prepare qubits $1, \ldots n-k$ in the state $|0\rangle$ and prepare the remaining k qubits in the state to be encoded.
2. Apply a circuit of CNOT, Hadamard and Phase gates that realizes $U^\dagger$ (up to global phase).

The quantum code stabilized by $\mathcal{S}$ encodes k qubits in n qubits and is therefore called an $[[n, k]]$ code.

2.7.5 Normalizer, distance, logical operators

The normalizer of a stablizer subgroup $\mathcal{S}$ in $\mathcal{P}_n$ is

$$\mathcal{N}_{\mathcal{P}_n}(\mathcal{S}) = \{P \in \mathcal{P}_n : P\mathcal{S} = \mathcal{S}P\}$$

The distance of the quantum code stabilized by $\mathcal{S}$ is the lowest weight of an element of $\mathcal{N}_{\mathcal{P}_n}(\mathcal{S}) \setminus \langle iI, \mathcal{S}\rangle$, where weight means the number of non-identity elements in the tensor product of Pauli matrices.

If $U \in \mathcal{CL}_n$ is such that $U^\dagger \sigma_{z1} U, \ldots, U^\dagger \sigma_{z\,n-k} U$ generate $\mathcal{S}$, then $U^\dagger \sigma_{zi} U, i = 1, \ldots, n$, $U^\dagger \sigma_{xi} U, i = n-k+1, \ldots, n$ and iI generate $\mathcal{N}_{\mathcal{P}_n}(\mathcal{S})$. The Pauli group elements $U^\dagger \sigma_{xi} U, U^\dagger \sigma_{zi} U, i = n-k+1, \ldots, n$ are one possible choice of logical operators for the quantum code stablized by $\mathcal{S}$.

2.7.6 Quantum CSS codes

If a stabilizer subgroup has a set of generators such that each generator contains either only σ_x and I terms or only σ_z and I terms, then the associated quantum code is called a Calderbank-Shor-Steane (CSS) code. A CSS code has a parity check matrix with the block diagonal form

$$H = \begin{pmatrix} H^x & 0 \\ 0 & H^z \end{pmatrix} \tag{56}$$

where $H^x (H^z)^T = 0$. Conversely, for every pair of matrices H^x, H^z such that $H^x (H^z)^T = 0$, there is a CSS code that has the parity check matrix $\begin{pmatrix} H^x & 0 \\ 0 & H^z \end{pmatrix}$.

A CSS code has an encoding circuit in the following form:

1. Prepare qubits $1, \ldots, m_x$ in the state $|+\rangle$, prepare qubits $m_x+1, \ldots, m_x+m_z$ in the state $|0\rangle$, prepare the remaining qubits in the state to be encoded.

2. Apply a circuit consisting of only CNOT gates.

To a quantum CSS code are associated two distances

$$d_x = \min\left\{|v| : v \in Ker(H^z), v \notin Im((H^x)^T)\right\} \tag{57}$$

$$d_z = \min\left\{|v| : v \in Ker(H^x), v \notin Im((H^z)^T)\right\} \tag{58}$$

which capture the ability of the code to correct, respectively, σ_x and σ_z errors. The distance of the CSS code when all Pauli errors are considered together is $d = \min(d_x, d_z)$.

3 Large CSS code from a tuple of smaller CSS codes and two collections of subsets

This section begins by associating a quantum CSS code to two tuples of matrices and two tuples of subsets (subsection 3.1). Then, a number of properties are established. Matrix decompositions of the parity check matrices of the code are derived in subsection 3.2. From these matrix decompositions can be computed the block size and rate (subsection 3.3), a basis for the stabilizer (subsection 3.4), an encoding circuit (subsection 3.5), the normalizer and the logical operators (subsection 3.6) and the linear error-correcting code protecting against syndrome errors (subsection 3.9). Moreover, the syndrome measurements can be parallelized (subsection 3.8), and the number of qubits in each measurement

can be kept low by suitable choices of the components and subsets (subsection 3.7). Throughout the section, the general results are illustrated by a particular small example.

3.1 Definition of the CSS code

Definition 1. *Take two tuples of matrices*

$$\mathbf{H^x} = (H_0^x, \dots H_{m-1}^x), H_i^x \in \mathbb{F}_2^{l_i^x \times n_i}, i = 0, \dots, m-1 \tag{59}$$

$$\mathbf{H^z} = (H_0^z, \dots H_{m-1}^z), H_i^z \in \mathbb{F}_2^{l_i^z \times n_i}, i = 0, \dots, m-1 \tag{60}$$

such that

$$\forall i, H_i^x (H_i^z)^T = 0 \tag{61}$$

Take two tuples of subsets of $[m]$*:*

$$\mathbf{X} = \begin{pmatrix} X_0 \\ \vdots \\ X_{u-1} \end{pmatrix} \quad \mathbf{Z} = \begin{pmatrix} Z_0 \\ \vdots \\ Z_{v-1} \end{pmatrix} \tag{62}$$

such that

$$\forall i \forall j, X_i \cap Z_j \neq \emptyset \tag{63}$$

To $\mathbf{H^x}, \mathbf{H^z}, \mathbf{X}, \mathbf{Z}$*, associate the CSS code with stabilizer parity check matrix*

$$CSS(\mathbf{H^x}, \mathbf{H^z}, \mathbf{X}, \mathbf{Z}) = \begin{pmatrix} M(\mathbf{H^x}, \mathbf{X}) & 0 \\ 0 & M(\mathbf{H^z}, \mathbf{Z}) \end{pmatrix} \tag{64}$$

where the notation of section 2.5 is used.

Theorem 3. $CSS(\mathbf{H^x}, \mathbf{H^z}, \mathbf{X}, \mathbf{Z})$ *is a valid stabilizer parity check matrix.*

Proof. (63) and (61) imply

$$\forall i \forall j, M(\mathbf{H^x}, X_i) M(\mathbf{H^z}, Z_j)^T = 0 \tag{65}$$

□

Running example: Take $m = 4$. Take $\mathbf{H^x}, \mathbf{H^z}$ so that for each $i \in [4]$, $H_i^x = H_i^z = \begin{pmatrix} 1 & 1 \end{pmatrix}$. Take

$$\mathbf{X} = \begin{pmatrix} X_0 \\ X_1 \end{pmatrix} = \begin{pmatrix} \{0,1\} \\ \{2,3\} \end{pmatrix} \quad \mathbf{Z} = \begin{pmatrix} Z_0 \\ Z_1 \end{pmatrix} = \begin{pmatrix} \{0,2\} \\ \{1,3\} \end{pmatrix} \tag{66}$$

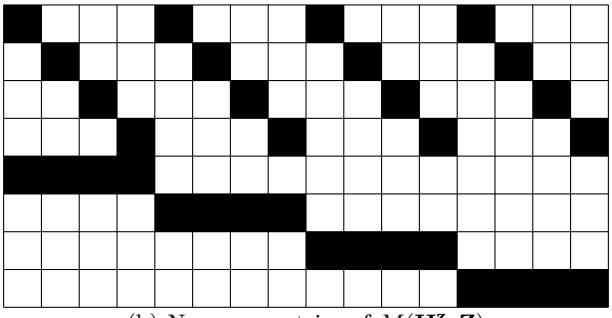
The two parity check matrices are

$$M(\mathbf{H^x}, \mathbf{X}) = \begin{pmatrix} \begin{pmatrix} 1 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 \end{pmatrix} \end{pmatrix} \tag{67}$$

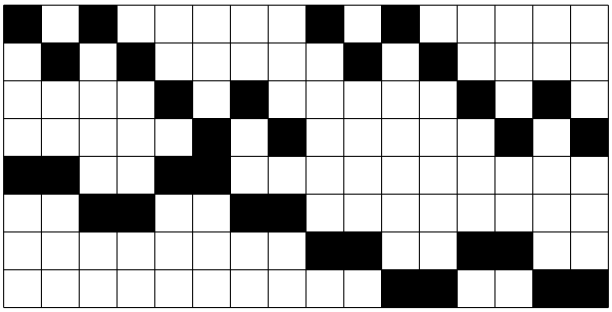
$$M(\mathbf{H^z}, \mathbf{Z}) = \begin{pmatrix} \begin{pmatrix} 1 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 \end{pmatrix} \end{pmatrix} \tag{68}$$

Figure 1: The code $CSS(\mathbf{H^x}, \mathbf{H^z}, \mathbf{X}, \mathbf{Z})$ in the running example

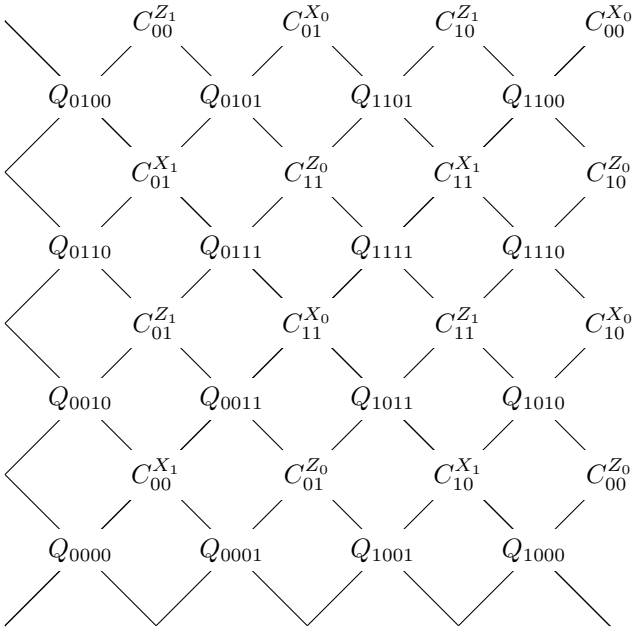
(a) Non-zero entries of $M(\mathbf{H^x}, \mathbf{X})$



(b) Non-zero entries of $M(\mathbf{H^z}, \mathbf{Z})$



(c) Tanner graph embedded in the torus. Qubits are denoted by Q_{abcd} for $a, b, c, d \in \{0, 1\}$. Checks are denoted by $C_{cd}^{A_b}$ for $A \in \{X, Z\}, b, c, d \in \{0, 1\}$, so $C_{00}^{X_0}$ is the 00 row of $M(\mathbf{H^x}, X_0)$, etc.



$M(\mathbf{H^x}, \mathbf{X})$ is the parity check matrix of the classical 2D product of the $[4, 3]$ single parity check code; $M(\mathbf{H^z}, \mathbf{Z})$ is isomorphic to $M(\mathbf{H^x}, \mathbf{X})$ by row and column permutation. A visualization of $M(\mathbf{H^x}, \mathbf{X})$ and $M(\mathbf{H^z}, \mathbf{Z})$ is given in Figures 1a and 1b.

The resulting [[16,2,4]] quantum CSS code has been previously considered in [18, 27] from the point of view of using classical 2D products to construct quantum CSS codes. However, it turns out that this is also a toric code.[1] Figure 1c shows the Tanner graph of $CSS(\mathbf{H^x}, \mathbf{H^z}, \mathbf{X}, \mathbf{Z})$ embedded in the torus.

3.2 Matrix decomposition

A matrix decomposition of $CSS(\mathbf{H^x}, \mathbf{H^z}, \mathbf{X}, \mathbf{Z})$ follows from the observations in sections 2.3, 2.4, 2.5, 2.6. From this decomposition follow many properties of the code.

For $\mathbf{H^x}, \mathbf{H^z}$ as above, for $i = 0, \ldots, m-1$, let

$$P_i^x H_i^x Q_i = L_i^x D_i^x R_i \tag{69}$$

$$P_i^z H_i^z Q_i = L_i^z D_i^z R_i^{-T} \tag{70}$$

be the joint decomposition of the pair H_i^x, H_i^z from Theorem 2.

Using these component decompositions, let

$$P(\mathbf{H^x}, \mathbf{X}) M(\mathbf{H^x}, \mathbf{X}) Q(\mathbf{H^x}) = L(\mathbf{H^x}, \mathbf{X}) D(\mathbf{H^x}, \mathbf{X}) R(\mathbf{H^x}) \tag{71}$$

$$P(\mathbf{H^z}, \mathbf{Z}) M(\mathbf{H^z}, \mathbf{Z}) Q(\mathbf{H^z}) = L(\mathbf{H^z}, \mathbf{Z}) D(\mathbf{H^z}, \mathbf{Z}) R(\mathbf{H^z}) \tag{72}$$

be the resulting decompositions of $M(\mathbf{H^x}, \mathbf{X}), M(\mathbf{H^z}, \mathbf{Z})$ from equation (39) of section 2.5.

Recall from section 2.5 that $D(\mathbf{H^x}, \mathbf{X})$ and $D(\mathbf{H^z}, \mathbf{Z})$ are incomplete permutation matrices. The immediate goal is to compute their column support. To do this, recall that equation (29) in section 2.5 gives a matrix decomposition for individual layers $M(\mathbf{H^x}, X_i), M(\mathbf{H^z}, Z_j)$. The column support of $D(\mathbf{H^x}, \mathbf{X})$ is the union of the column supports of $D(\mathbf{H^x}, X_i)$, and the column support $D(\mathbf{H^z}, \mathbf{Z})$ is the union of the column supports of $D(\mathbf{H^z}, Z_j)$.

The following theorem computes the relevant column supports:

Theorem 4. *Let*

$$\mathbf{S^x} = \vec{1}_u \vec{1}_m^T Diag(\vec{n} - \vec{1}_m) - \mathbf{X} Diag(\vec{n} - \vec{r^x}) \tag{73}$$

$$\mathbf{S^z} = \mathbf{Z} Diag(\vec{n} - \vec{r^z}) \tag{74}$$

where $\vec{1}_m$ is the $m \times 1$ vector of ones, $\vec{n} = (n_0, \ldots, n_{m-1})^T$, where $\vec{r^x} = (r_0^x, \ldots, r_{m-1}^x)^T$, $\vec{r^z} = (r_0^z, \ldots, r_{m-1}^z)^T$ are the vectors of ranks of the matrices in the tuples $\mathbf{H^x}, \mathbf{H^z}$, and where $Diag(\vec{w})$ is the diagonal matrix obtained

[1] The author would like to thank the anonymous reviewer who pointed out that a particular tessellation of the torus also gives a [[16,2,4]] CSS code. It turns out that this is a coincindence not just of the parameters but of the code itself.

from vector $\vec{w}$. Here, $\mathbf{X}, \mathbf{Z}$ are interpreted as $u \times m, v \times m$ indicator matrices as in Section 2.2 and $\mathbf{S^x}, \mathbf{S^z}$ are interpreted as $u \times m, v \times m$ matrices with each row $S_0^x, \ldots, S_{u-1}^x$, $S_0^z, \ldots, S_{v-1}^z$ specifying an element of $[\vec{n}]$.

With $\mathbf{S^x}, \mathbf{S^z}$ so defined, the following hold:

1. *For $i = 0, \ldots, u-1$, the column support of $D(\mathbf{H^x}, X_i)$ is $\langle \{S_i^x\} \rangle_\downarrow = \{t \in [\vec{n}] : t \leq S_i^x\}$.*
2. *The column support of $D(\mathbf{H^x}, \mathbf{X})$ is $\langle \mathbf{S^x} \rangle_\downarrow = \{t \in [\vec{n}] : \exists i, t \leq S_i^x\}$.*
3. *For $j = 0, \ldots, v-1$, the column support of $D(\mathbf{H^z}, Z_j)$ is $\left\langle \{S_j^z\} \right\rangle_\uparrow = \{t \in [\vec{n}] : t \geq S_j^z\}$.*
4. *The column support of $D(\mathbf{H^z}, \mathbf{Z})$ is $\langle \mathbf{S^z} \rangle_\uparrow = \{t \in [\vec{n}] : \exists j, t \geq S_j^z\}$.*
5. *$\langle \mathbf{S^x} \rangle_\downarrow$ and $\langle \mathbf{S^z} \rangle_\uparrow$ are disjoint.*

Proof. **Part 1:** Take any $i \in [u]$. Recall that

$$D(\mathbf{H^x}, X_i) = \otimes_{k=0}^{m-1} \begin{cases} D_k^x & \text{if } k \in X_i \\ I_{n_k} & \text{otherwise} \end{cases} \tag{75}$$

Then, column $t \in [\vec{n}]$ of $D(\mathbf{H^x}, X_i)$ is non-zero if and only if t_k is in the column support of D_k^x for $k \in X_i$. Recall further that the column support of D_k^x is $0, \ldots, r_k^x - 1$. Thus, column $t \in [\vec{n}]$ of $D(\mathbf{H^x}, X_i)$ is non-zero if and only if

$$t_k \leq \begin{cases} r_k^x - 1 & \text{if } k \in X_i \\ n_k - 1 & \text{otherwise} \end{cases} \tag{76}$$

This is the same as $t \leq S_i^x$.

Part 2 follows from Part 1 because the column support of $D(\mathbf{H^x}, \mathbf{X})$ is the union of the column supports of $D(\mathbf{H^x}, X_i)$ for $i \in [u]$.

Part 3 is similar to Part 1, except that the column support of D_k^z is $n_k - r_k^z, \ldots, n_k - 1$, so column t of $D(\mathbf{H^z}, Z_j)$ is non-zero if and only if

$$t_k \geq \begin{cases} n_k - r_k^z & \text{if } k \in Z_j \\ 0 & \text{otherwise} \end{cases} \tag{77}$$

This is the same as $t \geq S_j^z$.

Part 4 follows from Part 3 because the column support of $D(\mathbf{H^z}, \mathbf{Z})$ is the union of the column supports of $D(\mathbf{H^z}, Z_j$ for $j \in [v]$.

Part 5: suppose for a contradiction that $t \in \langle \mathbf{S^x} \rangle_\downarrow \cap \langle \mathbf{S^z} \rangle_\uparrow$. Take i, j so that $S_j^z \leq t \leq S_i^x$. Take $k \in X_i \cap Z_j$. Then, $n_k - r_k^z \leq t_k \leq r_k^z - 1$. Then, $r_k^x + r_k^z \geq n_k + 1$, which contradicts $H_k^x (H_k^z)^T = 0$. □

Running example: For the running example of this section, the component decompositions (69) and (70) are

$$H_i^x = \begin{pmatrix} 1 & 1 \end{pmatrix} = 1 \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \tag{78}$$

$$H_i^z = \begin{pmatrix} 1 & 1 \end{pmatrix} = 1 \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \tag{79}$$

The matrices $\mathbf{S^x}$ and $\mathbf{S^z}$ from equations (73) and (74) are

$$\mathbf{S^x} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix} \tag{80}$$

$$\mathbf{S^z} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \tag{81}$$

The incomplete permutation matrix

$$D(\mathbf{H^x}, X_0) = \begin{pmatrix} 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{82}$$

is supported on columns indexed by

$$\langle \{S_0^x\} \rangle_\downarrow = \{0000, 0001, 0010, 0011\} \tag{83}$$

where shorthand notation is used for elements of $\{0,1\}^4$. The incomplete permutation matrix

$$D(\mathbf{H^x}, X_1) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \end{pmatrix} \tag{84}$$

is supported on columns indexed by

$$\langle \{S_1^x\} \rangle_\downarrow = \{0000, 0100, 1000, 1100\} \tag{85}$$

The incomplete permutation matrix $D(\mathbf{H^x}, \mathbf{X})$ is supported on columns indexed by

$$\langle \mathbf{S^x} \rangle_\downarrow = \langle \{S_0^x\} \rangle_\downarrow \cup \langle \{S_1^x\} \rangle_\downarrow = \{0000, 0001, 0010, 0011, 0100, 1000, 1100\} \tag{86}$$

Similarly, the incomplete permutation matrix

$$D(\mathbf{H^z}, Z_0) = \begin{pmatrix} 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{87}$$

is supported on columns indexed by

$$\langle \{S_0^z\} \rangle_\uparrow = \{1010, 1011, 1110, 1111\} \tag{88}$$

The incomplete permutation matrix

$$D(\mathbf{H^z}, Z_1) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \end{pmatrix} \tag{89}$$

is supported on columns indexed by

$$\langle\{S_1^z\}\rangle_\uparrow = \{0101, 0111, 1101, 1111\} \tag{90}$$

The incomplete permutation matrix $D(\mathbf{H^z}, \mathbf{Z})$ is supported on columns indexed by

$$\langle\mathbf{S^z}\rangle_\uparrow = \langle\{S_0^z\}\rangle_\uparrow \cup \langle\{S_1^z\}\rangle_\uparrow = \{0101, 0111, 1010, 1011, 1101, 1110, 1111\} \tag{91}$$

3.3 Block size and rate

Theorem 5. *Let $\mathbf{S^x}, \mathbf{S^z}$ be as in Theorem 4, equations (73), (74), and let $\mathbf{K} = [\vec{n}]\backslash(\langle\mathbf{S^x}\rangle_\downarrow \cup \langle\mathbf{S^z}\rangle_\uparrow)$. Then, $CSS(\mathbf{H^x}, \mathbf{H^z}, \mathbf{X}, \mathbf{Z})$ is a $[[\prod_{i=0}^{m-1} n_i, |\mathbf{K}|]]$ quantum CSS code.*

Proof. The matrix decompositions (71) and (72) and the computation of the column supports of $D(\mathbf{H^x}, \mathbf{X})$ and $D(\mathbf{H^z}, \mathbf{Z})$ in Theorem 4 imply that

$$rank(M(\mathbf{H^x}, \mathbf{X})) = |\langle\mathbf{S^x}\rangle_\downarrow| \text{ and } rank(M(\mathbf{H^z}, \mathbf{Z})) = |\langle\mathbf{S^z}\rangle_\uparrow|$$

Moreover, $\langle\mathbf{S^x}\rangle_\downarrow$ and $\langle\mathbf{S^z}\rangle_\uparrow$ are disjoint, again by Theorem 4. Then, $CSS(\mathbf{H^x}, \mathbf{H^z}, \mathbf{X}, \mathbf{Z})$ has block size $\prod_{i=0}^{m-1} n_i$ qubits and has

$$\prod_{i=0}^{m-1} n_i - rank(M(\mathbf{H^x}, \mathbf{X})) - rank(M(\mathbf{H^z}, \mathbf{Z})) = |\mathbf{K}| \tag{92}$$

encoded qubits. □

Running example: for the running example of this section,

$$\mathbf{K} = \{0,1\}^4\backslash(\langle\mathbf{S^x}\rangle_\downarrow \cup \langle\mathbf{S^z}\rangle_\uparrow) = \{0110, 1001\} \tag{93}$$

3.4 Basis of the stabilizer

Theorem 6. *The non-zero rows of $D(\mathbf{H^x}, \mathbf{X})R(\mathbf{H^x})Q(\mathbf{H^x})^{-1}$ are a basis for $RowSpan(M(\mathbf{H^x}, \mathbf{X}))$ and the non-zero rows of $D(\mathbf{H^z}, \mathbf{Z})R(\mathbf{H^z})Q(\mathbf{H^z})^{-1}$ are a basis for $RowSpan(M(\mathbf{H^z}, \mathbf{Z}))$.*

Proof. Recall the matrix decomposition (71). The non-zero rows of $D(\mathbf{H^x}, \mathbf{X})R(\mathbf{H^x})Q(\mathbf{H^x})^{-1}$ are linearly independent because $R(\mathbf{H^x}), Q(\mathbf{H^x})$ are invertible and $D(\mathbf{H^x}, \mathbf{X})$ is an incomplete permutation matrix. Moreover, the row spans of $D(\mathbf{H^x}, \mathbf{X})R(\mathbf{H^x})Q(\mathbf{H^x})^{-1}$ and $M(\mathbf{H^x}, \mathbf{X})$ are the same, because $P(\mathbf{H^x}, \mathbf{X}), L(\mathbf{H^x}, \mathbf{X})$ are invertible. Similarly, (72), implies the statement for the Z stabilizers. □

Running example: for the running example of this section, a basis of $RowSpan(M(\mathbf{H^x}, \mathbf{X}))$ is obtained from the rows of $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{\otimes 4}$ indexed by $\langle\mathbf{S^x}\rangle_\downarrow$,

and a basis of $RowSpan(M(\mathbf{H^z}, \mathbf{Z}))$ is obtained from the rows of $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{\otimes 4}$ indexed by $\langle \mathbf{S^z} \rangle_\uparrow$. These are shown in figures 2a and 2b. Note that the basis of the stabilizer contains two elements of weight 16, eight elements of weight 8, and four elements of weight 4. On the other hand, the dependent generators of the stabilizer given by $M(\mathbf{H^x}, \mathbf{X}), M(\mathbf{H^z}, \mathbf{Z})$ contain 16 elements of weight 4. This illustrates one advantage in using the matrices $M(\mathbf{H^x}, \mathbf{X}), M(\mathbf{H^z}, \mathbf{Z})$ to specify syndrome measurements.

3.5 Encoding circuit

In describing the encoding circuit, it is convenient to index qubits by tuples in $[\vec{n}]$.

Theorem 7. *Let $\mathbf{S^x}, \mathbf{S^z}$ be as in Theorem 4, equations (73), (74), and let $\mathbf{K} = [\vec{n}] \backslash (\langle \mathbf{S^x} \rangle_\downarrow \cup \langle \mathbf{S^z} \rangle_\uparrow)$. Then, $CSS(\mathbf{H^x}, \mathbf{H^z}, \mathbf{X}, \mathbf{Z})$ has the encoding circuit:*

1. *Prepare qubits indexed by $\langle \mathbf{S^x} \rangle_\downarrow$ in state $|+\rangle$.*
2. *Prepare qubits indexed by $\langle \mathbf{S^z} \rangle_\uparrow$ in state $|0\rangle$.*
3. *Prepare qubits indexed by $\mathbf{K}$ in the state to be encoded.*
4. *Apply CNOT circuit $U \in \mathcal{CL}_{[\vec{n}]}$, where the image of U under the surjective group homomorphism Ψ satisfies*

$$\Psi(U^{-1}) = \begin{pmatrix} R(\mathbf{H^x})Q(\mathbf{H^x})^{-1} & 0 \\ 0 & R(\mathbf{H^z})Q(\mathbf{H^z})^{-1} \end{pmatrix} \tag{94}$$

Proof. Before the application of U, the state to be encoded is placed in the joint $+1$ eigenspace of the single-qubit Pauli operators $\sigma_{xi}, i \in \langle \mathbf{S^x} \rangle_\downarrow$ and $\sigma_{zj}, j \in \langle \mathbf{S^z} \rangle_\uparrow$. After the application of U, the encoded state is in the joint $+1$ eigenspace of the Pauli operators $U\sigma_{xi}U^{-1}, i \in \langle \mathbf{S^x} \rangle_\downarrow$ and $U\sigma_{zj}U^{-1}, j \in \langle \mathbf{S^z} \rangle_\uparrow$. Using (55) and (94), deduce $\Phi(U\sigma_{xi}U^{-1})$ is row i of $\begin{pmatrix} R(\mathbf{H^x})Q(\mathbf{H^x})^{-1} & 0 \end{pmatrix}$ and $\Phi(U\sigma_{zj}U^{-1})$ is row j of $\begin{pmatrix} 0 & R(\mathbf{H^z})Q(\mathbf{H^z})^{-1} \end{pmatrix}$. Then, Theorem 6 implies that the final stabilizer subspace can equivalently be described using the parity check matrix $CSS(\mathbf{H^x}, \mathbf{H^z}, \mathbf{X}, \mathbf{Z})$. □

Moreover, the encoding circuit can be parallelized in the following sense:

Theorem 8. *Suppose that there exist $d_0, \ldots, d_{m-1}$ such that for each $i \in [m]$, the symplectic matrix*

$$\begin{pmatrix} R_i Q_i^{-1} & 0 \\ 0 & R_i^{-T} Q_i^{-1} \end{pmatrix}$$

has a preimage in the Clifford group given by a depth d_i CNOT circuit. Then, the symplectic matrix

$$\begin{pmatrix} R(\mathbf{H^x})Q(\mathbf{H^x})^{-1} & 0 \\ 0 & R(\mathbf{H^z})Q(\mathbf{H^z})^{-1} \end{pmatrix}$$

has a preimage in the Clifford group given by a depth $\sum_{i=0}^{m-1} d_i$ CNOT circuit.

Proof. Write

$$\begin{pmatrix} R(\mathbf{H^x})Q(\mathbf{H^x})^{-1} & 0 \\ 0 & R(\mathbf{H^z})Q(\mathbf{H^z})^{-1} \end{pmatrix} = \prod_{i=0}^{m-1} \begin{pmatrix} I \otimes \cdots \otimes R_i Q_i^{-1} \otimes \cdots \otimes I & 0 \\ 0 & I \otimes \cdots \otimes R_i^{-T} Q_i^{-1} \otimes \cdots \otimes I \end{pmatrix} \quad (95)$$

The i-th term has preimage in the Clifford group that is a depth d_i CNOT circuit: the preimage of

$$\begin{pmatrix} R_i Q_i^{-1} & 0 \\ 0 & R_i^{-T} Q_i^{-1} \end{pmatrix}$$

performed in parallel on all groups of qubits $PROJ^{-1}_{\{i\}^c}(y), \quad y \in (\times_{j \neq i}[n_j])$, where $PROJ_{\{i\}^c} : [\vec{n}] \to \left(\times_{i \in X_1^c}[n_i]\right)$ is the coordinate projection on positions other than i. □

Running example: For the running example of this section,

$$\begin{pmatrix} R(\mathbf{H^x})Q(\mathbf{H^x})^{-1} & \\ & R(\mathbf{H^z})Q(\mathbf{H^z})^{-1} \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{\otimes 4} & \\ & \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{\otimes 4} \end{pmatrix}$$

The symplectic matrix

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

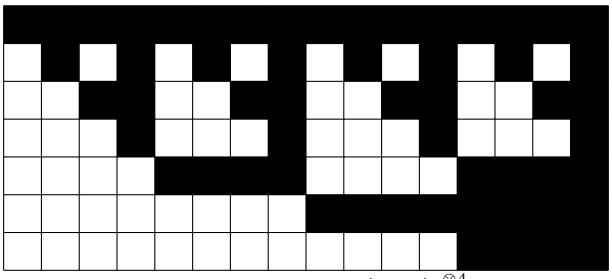
has preimage in the Clifford group a CNOT gate with control qubit 0 and target qubit 1. The symplectic matrix

$$\begin{pmatrix} I_8 \otimes \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} & \\ & I_8 \otimes \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \end{pmatrix}$$

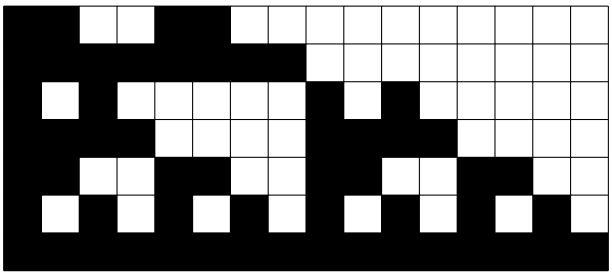
has a depth 1 preimage in the Clifford group: CNOT gates with control qubit $(y, 0)$ and target qubit $(y, 1)$ for $y \in \{0, 1\}^3$. Similarly, the other three terms in the encoding circuit factorization (95) have preimages of depth 1. The full depth 4 encoding circuit can be seen in Figure 2c. The CNOT part of the circuit follows the same recursive pattern as is used to encode Polar and Reed-Muller codes.

Figure 2: Independent generators of the stabilizer and encoding circuit for the running example.

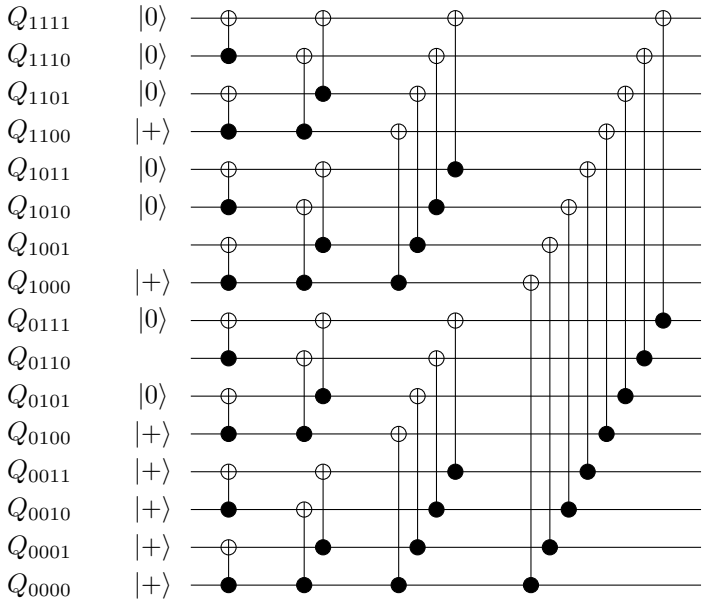
(a) Non-zero elements of the rows of $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{\otimes 4}$ indexed by $\langle \mathbf{S^x} \rangle_{\downarrow}$



(b) Non-zero elements of the rows of $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{\otimes 4}$ indexed by $\langle \mathbf{S^z} \rangle_{\uparrow}$



(c) Encoding circuit



3.6 Normalizer, logical operators

Theorem 9. *The normalizer for $CSS(\mathbf{H^x}, \mathbf{H^z}, \mathbf{X}, \mathbf{Z})$ is generated by iI and the Pauli group elements corresponding to the non-zero rows of*

$$\begin{pmatrix} \Pi\left(\langle \mathbf{S^z} \rangle_{\uparrow}^{c}\right) R(\mathbf{H^x}) Q(\mathbf{H^x})^{-1} & \\ & \Pi\left(\langle \mathbf{S^x} \rangle_{\downarrow}^{c}\right) R(\mathbf{H^z}) Q(\mathbf{H^z})^{-1} \end{pmatrix} \tag{96}$$

A canonical choice of logical operators for $CSS(H^x, H^z, X, Z)$ is obtained from the non-zero rows of

$$\begin{pmatrix} \Pi\left(\mathbf{K}\right) R(\mathbf{H^x}) Q(\mathbf{H^x})^{-1} & \\ & \Pi\left(\mathbf{K}\right) R(\mathbf{H^z}) Q(\mathbf{H^z})^{-1} \end{pmatrix} \tag{97}$$

Proof. Think of the encoding circuit in Theorem 7. Before the application of U, the subspace stabilized by the single qubit Pauli operators $\sigma_{xi}, i \in \langle \mathbf{S^x} \rangle_{\downarrow}$ and $\sigma_{zj}, j \in \langle \mathbf{S^z} \rangle_{\uparrow}$ has a canonical choice for generators of the normalizer: iI and the single qubit Pauli operators $\sigma_{xi}, i \in \langle \mathbf{S^z} \rangle_{\uparrow}^{c}$ and $\sigma_{zj}, j \in \langle \mathbf{S^x} \rangle_{\downarrow}^{c}$. It also has a canonical choice of logical operators: the single qubit Pauli operators $\sigma_{xi}, \sigma_{zi}, i \in \mathbf{K}$. After the application of U, the single qubit Pauli operators in the original set of normalizer generators and logical operators are conjugated by U to become the Pauli operators corresponding to the rows of $\Psi(U^{-1})$ given in (96) and (97). □

Running example: For the running example, the X logical operators are determined by the rows of $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{\otimes 4}$ indexed by $\mathbf{K} = \{0110, 1001\}$. The first one of these corresponds to Pauli-σ_x on qubits $Q_{0000}, Q_{0001}, Q_{1001}, Q_{1000}$, and the second to Pauli-σ_x on qubits $Q_{0000}, Q_{0010}, Q_{0110}, Q_{0100}$. Similarly, the Z logical operators are given by the rows of $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{\otimes 4}$ indexed by $\mathbf{K} = \{0110, 1001\}$. The first of these corresponds to Pauli-σ_z on qubits $Q_{0000}, Q_{0010}, Q_{0110}, Q_{0100}$ and the second to Pauli-σ_z on qubits $Q_{0000}, Q_{0001}, Q_{1001}, Q_{1000}$. The logical operators can be visualized using Figure 1c. It can be seen there that they correspond to the non-contractible loops of the torus.

3.7 Row and column weight

While the present paper is focused on quantum CSS codes with block size a few tens or a few hundreds of qubits and with sparse parity check matrices, it is worth mentioning the asymptotic behaviour of the row and column weights of the parity check matrices $CSS(\mathbf{H^x}, \mathbf{H^z}, \mathbf{X}, \mathbf{Z})$.

Theorem 10. *Let w_r, w_c be upper bounds on the weight of rows, respectively columns, of matrices in the tuples $\mathbf{H^x}, \mathbf{H^z}$. Let p be an upper bound on the size of the sets in the tuples $\mathbf{X}, \mathbf{Z}$. Then, w_r^p is an upper bound on the weight of rows of $CSS(\mathbf{H^x}, \mathbf{H^z}, \mathbf{X}, \mathbf{Z})$ and on the degree of checks in the Tanner graph,*

$\max(u,v)w_c^p$ *is an upper bound on the weight of columns of* $CSS(\mathbf{H^x},\mathbf{H^z},\mathbf{X},\mathbf{Z})$ *and* $(u+v)w_c^p$ *is an upper bound on the degree of qubits in the Tanner graph.*

Proof. Each layer

$$M(\mathbf{H^x},X_i)=\otimes_{j=0}^{m-1}\begin{cases}H_j & \text{if } j\in X_i\\ I_{n_j} & \text{otherwise}\end{cases}$$

has rows of weight at most $w_r^{|X_i|}$ and columns of weight at most $w_c^{|X_i|}$. The same applies to layers of $M(\mathbf{H^z},\mathbf{Z})$. Maximizing over the layers gives the upper bound on row weight/check degree, and summing over the layers gives the upper bound on column weight/qubit degree. □

Therefore, keeping u,v,p,w_r,w_c constant and letting (some of) the n_j go to infinity gives quantum ldpc codes. The properties of asymptotic families obtained in this way are left for future research, while the rest of this paper focuses on the finite block length regime.

Running example: for the running example of this section, $w_r=p=u=v=2$, $w_c=1$. The parity check matrix $CSS(\mathbf{H^x},\mathbf{H^z},\mathbf{X},\mathbf{Z})$ has row weight 4 and column weight 2. In the Tanner graph, all vertices have degree 4.

3.8 Syndrome measurement circuit

Theorem 11. *Let* $p=\max\left(\left\{\prod_{j\in X_i^c}n_j : i\in[u]\right\}\cup\left\{\prod_{j\in Z_i^c}n_j : i\in[v]\right\}\right)$. *Let* d,a *be such that the following two statements hold:*

1. *For each* $i\in[u]$, *the* σ_x *measurements specified by* $\otimes_{j\in X_i}H_j^x$ *can be measured by a circuit with at most* a *ancilla qubits and depth at most* d.
2. *For each* $i\in[v]$, *the* σ_z *measurements specified by* $\otimes_{j\in Z_i}H_j^z$ *can be measured by a circuit with at most* a *ancilla qubits and depth at most* d.

Then, $CSS(\mathbf{H^x},\mathbf{H^z},\mathbf{X},\mathbf{Z})$ *has a syndrome measurement circuit with at most* pa *ancilla qubits and depth at most* $(u+v)d$.

Proof. Consider the layer $M(\mathbf{H^x},X_0)$. There exist row and column permutations that map $M(\mathbf{H^x},X_0)$ to

$$\left(\otimes_{j\in X_0^c}I_{n_j}\right)\otimes\left(\otimes_{j\in X_0}H_j^x\right)=\begin{pmatrix}\otimes_{j\in X_0}H_j^x & & \\ & \ddots & \\ & & \otimes_{j\in X_0}H_j^x\end{pmatrix}\tag{98}$$

Therefore, the syndrome measurements specified by $M(\mathbf{H^x},X_0)$ can be performed with at most pa ancilla qubits and depth at most d: perform the circuit for $\otimes_{i\in X_0}H_i^x$ in parallel on all groups of qubits $PROJ_{X_0^c}^{-1}(y),\ \ y\in\left(\times_{i\in X_0^c}[n_i]\right)$ where $PROJ_{X_0^c}:[\vec{n}]\to\left(\times_{i\in X_0^c}[n_i]\right)$ is the coordinate projection on the positions in X_0^c. A similar argument applies for the other layers. Then, all syndrome measurements of $CSS(\mathbf{H^x},\mathbf{H^z},\mathbf{X},\mathbf{Z})$ can be performed by a circuit with at most pa ancilla qubits and depth at most $(u+v)d$. □

Running example: For the running example of this section, $p = 4$. The row vector $\begin{pmatrix}1 & 1\end{pmatrix} \otimes \begin{pmatrix}1 & 1\end{pmatrix} = \begin{pmatrix}1 & 1 & 1 & 1\end{pmatrix}$ specifies a syndrome measurement that can be performed with one ancilla qubit and depth 6: preparation of the ancilla in state $|+\rangle$ or state $|0\rangle$, 4 CNOT gates and measurement of the ancilla in the σ_x or σ_z eigenbasis. Therefore, the running example $CSS(\mathbf{H^x}, \mathbf{H^z}, \mathbf{X}, \mathbf{Z})$ has a syndrome measurement circuit with 4 ancilla qubits and depth 24.

3.9 Protection against syndrome errors

The vector spaces $Im(M(\mathbf{H^x}, \mathbf{X}))$ and $Im(M(\mathbf{H^z}, \mathbf{Z}))$ can be called the spaces of valid syndromes: if no error occurs during the measurements, then the syndromes for phase flips and bit flips belong respectively to $Im(M(\mathbf{H^x}, \mathbf{X}))$ and $Im(M(\mathbf{H^z}, \mathbf{Z}))$. When there are errors in the measurement outcomes, then $Im(M(\mathbf{H^x}, \mathbf{X}))$, $Im(M(\mathbf{H^z}, \mathbf{Z}))$ can be viewed as classical linear codes. The matrix decompositions (71) and (72) give the following information about these vector spaces:

Theorem 12. *1. $Im(M(\mathbf{H^x}, \mathbf{X}))$ is a $\left[\sum_{j=0}^{u-1} \left(\prod_{i\in X_j} l_i^x\right) \left(\prod_{i\in X_j^c} n_i\right), \left|\langle \mathbf{S^x}\rangle_\downarrow\right|\right]$ classical linear code. A basis for it is given by the non-zero columns of $P(\mathbf{H^x}, \mathbf{X})^{-1} L(\mathbf{H^x}, \mathbf{X}) D(\mathbf{H^x}, \mathbf{X})$. A basis for its orthogonal complement is given by rows of $L(\mathbf{H^x}, \mathbf{X})^{-1} P(\mathbf{H^x}, \mathbf{X})$ indexed by the complement of the row support of $D(\mathbf{H^x}, \mathbf{X})$.*

2. $Im(M(\mathbf{H^z}, \mathbf{Z}))$ is a $\left[\sum_{j=0}^{v-1} \left(\prod_{i\in Z_j} l_i^z\right) \left(\prod_{i\in Z_j^c} n_i\right), \left|\langle \mathbf{S^z}\rangle_\uparrow\right|\right]$ classical linear code. A basis for it is given by the non-zero columns of $P(\mathbf{H^z}, \mathbf{Z})^{-1} L(\mathbf{H^z}, \mathbf{Z}) D(\mathbf{H^z}, \mathbf{Z})$. A basis for its orthogonal complement is given by rows of $L(\mathbf{H^z}, \mathbf{Z})^{-1} P(\mathbf{H^z}, \mathbf{Z})$ indexed by the complement of the row support of $D(\mathbf{H^z}, \mathbf{Z})$.

Proof. A single layer $M(\mathbf{H^x}, X_j)$ has $\left(\prod_{i\in X_j} l_i^x\right) \left(\prod_{i\in X_j^c} n_i\right)$ rows. Therefore, the total number of rows of $M(\mathbf{H^x}, \mathbf{X})$ is $\sum_{j=0}^{u-1} \left(\prod_{i\in X_j} l_i^x\right) \left(\prod_{i\in X_j^c} n_i\right)$. The dimension of $Im(M(\mathbf{H^x}, \mathbf{X}))$ is $rank(M(\mathbf{H^x}, \mathbf{X})) = \left|\langle \mathbf{S^x}\rangle_\downarrow\right|$. Now, use (71) to find a basis for $Im(M(\mathbf{H^x}, \mathbf{X}))$ and its orthogonal complement. Since $Q(\mathbf{H^x})$ and $R(\mathbf{H^x})$ are invertible, $Im(M(\mathbf{H^x}, \mathbf{X})) = Im(P(\mathbf{H^x}, \mathbf{X})^{-1} L(\mathbf{H^x}, \mathbf{X}) D(\mathbf{H^x}, \mathbf{X}))$. Since $P(\mathbf{H^x}, \mathbf{X})$ and $L(\mathbf{H^x}, \mathbf{X})$ are invertible and $D(\mathbf{H^x}, \mathbf{X})$ is an incomplete permutation matrix, a basis for this vector space is given by the non-zero columns of $P(\mathbf{H^x}, \mathbf{X})^{-1} L(\mathbf{H^x}, \mathbf{X}) D(\mathbf{H^x}, \mathbf{X})$, which are also the columns of $P(\mathbf{H^x}, \mathbf{X})^{-1} L(\mathbf{H^x}, \mathbf{X})$ indexed by the row support of $D(\mathbf{H^x}, \mathbf{X})$. Therefore, a basis for the orthogonal complement is given by rows of $L(\mathbf{H^x}, \mathbf{X})^{-1} P(\mathbf{H^x}, \mathbf{X})$ indexed by the complement of the row support of $D(\mathbf{H^x}, \mathbf{X})$. This proves part one. Similarly, (72) implies part two. □

Running example: The matrix decomposition of $M(\mathbf{H^x}, \mathbf{X})$ for the run-

ning example of this section is

$$M(\mathbf{H}^{\mathbf{x}},\mathbf{X}) = L(\mathbf{H}^{\mathbf{x}},\mathbf{X})D(\mathbf{H}^{\mathbf{x}},\mathbf{X})R(\mathbf{H}^{\mathbf{x}})$$
$$= \left(\left(\begin{pmatrix}1&1\\0&1\end{pmatrix}^{\otimes 2} \quad \begin{pmatrix}1&1\\0&1\end{pmatrix}^{\otimes 2}\right)\left(I_8 + e_{8,4}e_{8,0}^T\right)\right)$$
$$* \left(\left(I_8 + e_{8,4}e_{8,0}^T\right)\begin{pmatrix}D(\mathbf{H}^{\mathbf{x}},X_0)\\D(\mathbf{H}^{\mathbf{x}},X_1)\end{pmatrix}\right)\begin{pmatrix}1&1\\0&1\end{pmatrix}^{\otimes 4} \quad (99)$$

where $e_{n,0},\ldots,e_{n,n-1}$ denotes the standard basis of $\mathbb{F}_2^n$. The incomplete permutation matrix $D(\mathbf{H}^{\mathbf{x}},\mathbf{X}) = \left(I_8 + e_{8,4}e_{8,0}^T\right)\begin{pmatrix}D(\mathbf{H}^{\mathbf{x}},X_0)\\D(\mathbf{H}^{\mathbf{x}},X_1)\end{pmatrix}$ is supported on rows $0,1,2,3,5,6,7$. Therefore, a basis for $Im(M(\mathbf{H}^{\mathbf{x}},\mathbf{X}))$ is given by columns $0,1,2,3,5,6,7$ of

$$L(\mathbf{H}^{\mathbf{x}},\mathbf{X}) = \left(\begin{pmatrix}1&1\\0&1\end{pmatrix}^{\otimes 2} \quad \begin{pmatrix}1&1\\0&1\end{pmatrix}^{\otimes 2}\right)\left(I_8 + e_{8,4}e_{8,0}^T\right)$$
$$= \begin{pmatrix}1&1&1&1&0&0&0&0\\0&1&0&1&0&0&0&0\\0&0&1&1&0&0&0&0\\0&0&0&1&0&0&0&0\\1&0&0&0&1&1&1&1\\0&0&0&0&0&1&0&1\\0&0&0&0&0&0&1&1\\0&0&0&0&0&0&0&1\end{pmatrix} \quad (100)$$

and a basis for the orthogonal complement of $Im(M(\mathbf{H}^{\mathbf{x}},\mathbf{X}))$ is given by row 4 of

$$L(\mathbf{H}^{\mathbf{x}},\mathbf{X})^{-1} = \left(I_8 + e_{8,4}e_{8,0}^T\right)\left(\begin{pmatrix}1&1\\0&1\end{pmatrix}^{\otimes 2} \quad \begin{pmatrix}1&1\\0&1\end{pmatrix}^{\otimes 2}\right)$$
$$= \begin{pmatrix}1&1&1&1&0&0&0&0\\0&1&0&1&0&0&0&0\\0&0&1&1&0&0&0&0\\0&0&0&1&0&0&0&0\\1&1&1&1&1&1&1&1\\0&0&0&0&0&1&0&1\\0&0&0&0&0&0&1&1\\0&0&0&0&0&0&0&1\end{pmatrix} \quad (101)$$

Therefore, $Im(M(\mathbf{H^x}, \mathbf{X}))$ is the $[8, 7, 2]$ single parity check code. Similarly, $Im(M(\mathbf{H^z}, \mathbf{Z}))$ is also the $[8, 7, 2]$ single parity check code.

4 A subfamily related to a generalization of Reed-Muller codes

The simplest class of examples from the construction is obtained when all matrices in the tuples $\mathbf{H^x}, \mathbf{H^z}$ are $\begin{pmatrix} 1 & 1 \end{pmatrix}$. The matrices $M(\mathbf{H}, \mathbf{X})$ from equation (38) in section 2.5 in this case depend only on $\mathbf{X}$, so the shorthand notation $M(\mathbf{X}), CSS(\mathbf{X}, \mathbf{Z})$ will be used in this section. The equations (73), (74) determining the generators $\mathbf{S^x}, \mathbf{S^z}$ of the decreasing, respectively increasing, subset of column indices simplify to

$$\mathbf{S^x} = \vec{1}_u \vec{1}_m^T - \mathbf{X} \tag{102}$$

$$\mathbf{S^z} = \mathbf{Z} \tag{103}$$

Properties of examples in this class follow from properties of a family of vector spaces that generalize classical Reed-Muller codes.

4.1 Generalization of Reed-Muller codes

Let

$$R_m = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{\otimes m} \tag{104}$$

To decreasing $\mathbf{S} \subset \{0, 1\}^m$ associate the vector space

$$\mathcal{GRM}(\mathbf{S}) = RowSpan\left(\Pi(\mathbf{S}) R_m\right) \tag{105}$$

and to increasing $\mathbf{T} \subset \{0, 1\}^m$ associate the vector space

$$\mathcal{GRM}'(\mathbf{T}) = RowSpan\left(\Pi(\mathbf{T}) R_m^{-T}\right) \tag{106}$$

It will be seen later that these are two different parametrizations of the same family of vector spaces.

The vector spaces $\mathcal{GRM}(\mathbf{S})$ for $\mathbf{S}$ decreasing generalize Reed-Muller codes. If $\mathbf{S}(r, m) = \{s \in \{0, 1\}^m : |s| \leq r\}$ then the corresponding subspace is $\mathcal{GRM}(\mathbf{S}(r, m)) = \mathcal{RM}(r, m)$, the usual Reed-Muller code with parameters r, m. This is because the row of R_m with index $s \subset [m]$ is the truth table of the monomial $\mu_s(v_0, \ldots, v_{m-1}) = \prod_{i \in s} v_i$, so $RowSpan\left(\Pi(\mathbf{S}(r, m)) R_m\right)$ is the span of the truth tables of all monomials of degree at most r.

Moreover, it turns out that some of the properties of Reed-Muller codes can be extended to the vector spaces $\mathcal{GRM}(\mathbf{S})$.

First, the following theorem shows that $\mathcal{GRM}(\mathbf{S}), \mathcal{GRM}'(\mathbf{T})$ are two parametrizations of the same family of vector spaces. It also generalizes [24, Chapter 13, Theorems 4 and 12]: it shows that $\mathcal{GRM}(\mathbf{S})$ is spanned by certain low-weight elements and that the orthogonal complement of such a vector space is another vector space of the same family.

Theorem 13. *Take a tuple of subsets of* $[m]$

$$\mathbf{S} = \begin{pmatrix} S_0 \\ \vdots \\ S_{u-1} \end{pmatrix} \tag{107}$$

and take $\mathbf{T} = \vec{1}_u \vec{1}_m^T - \mathbf{S}$. *Then,*

$$\mathcal{GRM}\left(\langle \mathbf{S} \rangle_{\downarrow}\right) = \mathcal{GRM}'\left(\langle \mathbf{T} \rangle_{\uparrow}\right) = RowSpan(M(\mathbf{T})) \tag{108}$$

and

$$\mathcal{GRM}\left(\langle \mathbf{S} \rangle_{\downarrow}\right)^{\perp} = \mathcal{GRM}\left(\langle \mathbf{T} \rangle_{\uparrow}^{c}\right) \tag{109}$$

Proof. For the pair of orthogonal matrices $\begin{pmatrix} 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 \end{pmatrix}$, Theorem 2 gives the decompositions

$$\begin{pmatrix} 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \tag{110}$$

$$\begin{pmatrix} 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \tag{111}$$

From the two decompositions (110) and (111), and from equation (39) in section 2.5 follow two alternative decompositions of $M(\mathbf{T})$:

$$M(\mathbf{T}) = L(\mathbf{T})\Pi\left(\langle \mathbf{S} \rangle_{\downarrow}\right) R_m = L'(\mathbf{T})\Pi\left(\langle \mathbf{T} \rangle_{\uparrow}\right) R_m^{-T} \tag{112}$$

Equation (112) implies the Theorem. Indeed, since $L(\mathbf{T}), L'(\mathbf{T})$ are invertible, (112) imples (108). Moreover, the orthogonal complement of $RowSpan\left(\Pi\left(\langle \mathbf{T} \rangle_{\uparrow}\right) R_m^{-T}\right)$ is $RowSpan\left(\Pi\left(\langle \mathbf{T} \rangle_{\uparrow}^{c}\right) R_m\right)$, and this gives (109). □

Moreover, the inductive argument using the $(u, u+v)$ construction for the distance of Reed-Muller codes [24, Chapter 13, Theorems 2 and 3] also generalizes to the vector spaces $\mathcal{GRM}(\mathbf{S})$, and to the distances of quantum CSS codes built from these vector spaces. For $\mathbf{S}, \mathbf{T}$ decreasing, $\mathbf{S} \subset \mathbf{T}$, let

$$r(\mathbf{T}, \mathbf{S}) = \max\{|t| : t \in \mathbf{T}, t \notin \mathbf{S}\} \tag{113}$$

$$d(\mathbf{T}, \mathbf{S}) = \min\{|v| : v \in \mathcal{GRM}(\mathbf{T}), v \notin \mathcal{GRM}(\mathbf{S})\} \tag{114}$$

Then:

Theorem 14. *Let* $\mathbf{S}, \mathbf{T} \subset \{0,1\}^m$ *be decreasing with* $\mathbf{S} \subset \mathbf{T}$. *Then,*

$$d(\mathbf{T}, \mathbf{S}) = 2^{m - r(\mathbf{T}, \mathbf{S})}$$

Proof. Take

$$\mathbf{S}_0 = \{s \in \{0,1\}^{m-1} : (0,s) \in \mathbf{S}\} \tag{115}$$

$$\mathbf{S}_1 = \{s \in \{0,1\}^{m-1} : (1,s) \in \mathbf{S}\} \tag{116}$$

$$\mathbf{T}_0 = \{t \in \{0,1\}^{m-1} : (0,t) \in \mathbf{T}\} \tag{117}$$

$$\mathbf{T}_1 = \{t \in \{0,1\}^{m-1} : (1,t) \in \mathbf{T}\} \tag{118}$$

so that

$$\mathbf{S} = (0, \mathbf{S}_0) \cup (1, \mathbf{S}_1) \tag{119}$$

$$\mathbf{T} = (0, \mathbf{T}_0) \cup (1, \mathbf{T}_1) \tag{120}$$

Note that $r(\mathbf{T}, \mathbf{S}) = \max(r(\mathbf{T}_0, \mathbf{S}_0), r(\mathbf{T}_1, \mathbf{S}_1) + 1)$.

Note further that

$$\Pi(\mathbf{S}) R_m = \begin{pmatrix} \Pi(\mathbf{S}_0) & \\ & \Pi(\mathbf{S}_1) \end{pmatrix} \begin{pmatrix} R_{m-1} & R_{m-1} \\ & R_{m-1} \end{pmatrix} \tag{121}$$

$$\Pi(\mathbf{T}) R_m = \begin{pmatrix} \Pi(\mathbf{T}_0) & \\ & \Pi(\mathbf{T}_1) \end{pmatrix} \begin{pmatrix} R_{m-1} & R_{m-1} \\ & R_{m-1} \end{pmatrix} \tag{122}$$

so $\mathcal{GRM}(\mathbf{S})$ is obtained from $\mathcal{GRM}(\mathbf{S}_0)$ and $\mathcal{GRM}(\mathbf{S}_1)$ using the $(u, u+v)$ construction and, similarly, $\mathcal{GRM}(\mathbf{T})$ is obtained from $\mathcal{GRM}(\mathbf{T}_0)$ and $\mathcal{GRM}(\mathbf{T}_1)$ using the $(u, u+v)$ construction. In the special case $\mathbf{S} = \emptyset, \mathcal{GRM}(\mathbf{S}) = \{0\}$, the formula for the distance of the $(u, u+v)$ construction [24, Chapter 2, Theorem 33] is sufficient to finish the argument. In the general case, Theorem 15 in the present paper gives $d(\mathbf{T}, \mathbf{S}) = \min(2d(\mathbf{T}_0, \mathbf{S}_0), d(\mathbf{T}_1, \mathbf{S}_1))$. The proof is completed by induction on m. □

The remaining step of the proof of Theorem 14 is

Theorem 15. *For $\mathbf{S}, \mathbf{T}, \mathbf{S}_0, \mathbf{S}_1, \mathbf{T}_0, \mathbf{T}_1$ as in Theorem 14,*

$$d(\mathbf{T}, \mathbf{S}) = \min(2d(\mathbf{T}_0, \mathbf{S}_0), d(\mathbf{T}_1, \mathbf{S}_1))$$

Proof. First, $\mathcal{GRM}(\mathbf{T}) \setminus \mathcal{GRM}(\mathbf{S})$ contains vectors of the form (u, u) for $u \in \mathcal{GRM}(\mathbf{T}_0) \setminus \mathcal{GRM}(\mathbf{S}_0)$ and $(0, v)$ for $v \in \mathcal{GRM}(\mathbf{T}_1) \setminus \mathcal{GRM}(\mathbf{S}_1)$. Therefore, $d(\mathbf{T}, \mathbf{S}) \leq \min(2d(\mathbf{T}_0, \mathbf{S}_0), d(\mathbf{T}_1, \mathbf{S}_1))$.

It remains to prove the other direction. Note that, since $\mathbf{S}, \mathbf{T}$ are decreasing, $\mathbf{S}_0, \mathbf{S}_1, \mathbf{T}_0, \mathbf{T}_1$ are also decreasing and $\mathbf{S}_1 \subset \mathbf{S}_0$, $\mathbf{T}_1 \subset \mathbf{T}_0$. Moreover, $\mathbf{S} \subset \mathbf{T}$ implies $\mathbf{S}_0 \subset \mathbf{T}_0$, $\mathbf{S}_1 \subset \mathbf{T}_1$.

Now, take any $w \in \mathcal{GRM}(\mathbf{T}) \setminus \mathcal{GRM}(\mathbf{S})$ and write $w = (u, u+v)$ for $u \in \mathcal{GRM}(\mathbf{T}_0), v \in \mathcal{GRM}(\mathbf{T}_1)$. Take $t \in \mathbf{T} \setminus \mathbf{S}$ such that, when w is expressed as a linear combination of the rows of $\Pi(\mathbf{T}) R_m$, it has a non-zero coefficient for row t. Write $t = (b, t'), b \in \{0,1\}, t' \in \{0,1\}^{m-1}$ and consider cases:

Case 1: $b = 1, t' \in \mathbf{T}_1 \setminus \mathbf{S}_1$**.** Then, $v \in \mathcal{GRM}(\mathbf{T}_1) \setminus \mathcal{GRM}(\mathbf{S}_1)$, so

$$|(u, u+v)| = |u| + |u+v| \geq |v| \geq d(\mathbf{T}_1, \mathbf{S}_1) \tag{123}$$

Case 2: $b = 0, t' \in \mathbf{T}_0 \backslash (\mathbf{S}_0 \cup \mathbf{T}_1)$**.** Then,

$$(u, u+v) \in \mathcal{GRM}(\mathbf{T}_0) \backslash \mathcal{GRM}(\mathbf{S}_0 \cup \mathbf{T}_1)$$

so

$$|(u, u+v)| = |u| + |u+v| \geq 2d(\mathbf{T}_0, \mathbf{S}_0 \cup \mathbf{T}_1) \geq 2d(\mathbf{T}_0, \mathbf{S}_0) \tag{124}$$

where the last step follows from $\mathbf{S}_0 \subset \mathbf{S}_0 \cup \mathbf{T}_1$.

Case 3: $b = 0, t' \in \mathbf{T}_1 \backslash \mathbf{S}_0$**.** Then, $u \in \mathcal{GRM}(\mathbf{T}_1) \backslash \mathcal{GRM}(\mathbf{T}_1 \cap \mathbf{S}_0)$, so

$$|(u, u+v)| \geq |u| \geq d(\mathbf{T}_1, \mathbf{T}_1 \cap \mathbf{S}_0) \geq d(\mathbf{T}_1, \mathbf{S}_1) \tag{125}$$

where the last step follows from $\mathbf{S}_1 \subset \mathbf{T}_1 \cap \mathbf{S}_0$. □

It will be convenient later to have an expression for the minimum distance also in terms of the other parametrization of the family of vector spaces. For $\mathbf{S}, \mathbf{T} \subset \{0,1\}^m$ increasing, $\mathbf{S} \subset \mathbf{T}$, let

$$r'(\mathbf{T}, \mathbf{S}) = \max\{m - |t| : t \in \mathbf{T} \backslash \mathbf{S}\} \tag{126}$$

$$d'(\mathbf{T}, \mathbf{S}) = \min\{|v| : v \in \mathcal{GRM}'(\mathbf{T}) \backslash \mathcal{GRM}'(\mathbf{S})\} \tag{127}$$

Then:

Corollary 1. *Let* $\mathbf{S}, \mathbf{T} \subset \{0,1\}^m$ *be decreasing,* $\mathbf{S} \subset \mathbf{T}$*. Then,* $d'(\mathbf{T}, \mathbf{S}) = 2^{m - r'(\mathbf{T}, \mathbf{S})}$.

Proof. Let ϕ be the function that switches ones with zeros and zeros with ones.

Theorem 13 implies $\mathcal{GRM}(\phi(\mathbf{S})) = \mathcal{GRM}'(\mathbf{S})$ and $\mathcal{GRM}(\phi(\mathbf{T})) = \mathcal{GRM}'(\mathbf{T})$. Then, $d'(\mathbf{T}, \mathbf{S}) = d(\phi(\mathbf{T}), \phi(\mathbf{S}))$.

Moreover, the maximum number of zeros in an element of $\mathbf{T} \backslash \mathbf{S}$ is equal to the maximum number of ones in an element of $\phi(\mathbf{T}) \backslash \phi(\mathbf{S})$, so $r'(\mathbf{T}, \mathbf{S}) = r(\phi(\mathbf{T}), \phi(\mathbf{S}))$. Theorem 14 completes the proof. □

4.2 Properties of the $CSS(\mathbf{X}, \mathbf{Z})$ codes

The $CSS(\mathbf{X}, \mathbf{Z})$ codes inherit from the general construction all properties in section 3. Note in particular that these codes have especially simple encoding and syndrome measurement circuits, as illustrated already by the running example in Section 3. In addition, the connection to generalized Reed-Muller codes and the results of section 4.1 give formulas for the distances.

Theorem 16. *For* $\mathbf{X}, \mathbf{Z} \subset \{0,1\}^m$ *satisfying* (63)*, let* $\mathbf{S^x} = \vec{1}_u \vec{1}_m^T - \mathbf{X}$*,* $\mathbf{S^z} = \mathbf{Z}$*,* $\mathbf{K} = \{0,1\}^m \backslash \left(\langle \mathbf{S^x} \rangle_\downarrow \cup \langle \mathbf{S^z} \rangle_\uparrow \right)$*. Then, the distances of* $CSS(\mathbf{X}, \mathbf{Z})$ *are*

$$d_x = 2^{\min\{m - |v| : v \in \mathbf{K}\}} \tag{128}$$

$$d_z = 2^{\min\{|v| : v \in \mathbf{K}\}} \tag{129}$$

Proof. From the results of section 3 deduce that $d_x = d(\langle \mathbf{S^z} \rangle_{\uparrow}^c, \langle \mathbf{S^x} \rangle_{\downarrow})$ and $d_z = d'(\langle \mathbf{S^x} \rangle_{\downarrow}^c, \langle \mathbf{S^z} \rangle_{\uparrow})$. Then, from Theorem 14 and Corollary 1 deduce that $d_x = 2^{m-r(\langle \mathbf{S^z} \rangle_{\uparrow}^c, \langle \mathbf{S^x} \rangle_{\downarrow})}$ and $d_z = 2^{m-r'(\langle \mathbf{S^x} \rangle_{\downarrow}^c, \langle \mathbf{S^z} \rangle_{\uparrow})}$. Finally, note that $r(\langle \mathbf{S^z} \rangle_{\uparrow}^c, \langle \mathbf{S^x} \rangle_{\downarrow}) = \max\{|v| : v \in \mathbf{K}\}$ and $r'(\langle \mathbf{S^x} \rangle_{\downarrow}^c, \langle \mathbf{S^z} \rangle_{\uparrow}) = \max\{m - |v| : v \in \mathbf{K}\}$ to complete the proof. □

Moreover, the connection to generalized Reed-Muller codes allows the computation of the minimum distance of $Im(M(\mathbf{S}))$ for any tuple of subsets $\mathbf{S}$. Recall that in the context of quantum error correction, the vector space $Im(M(\mathbf{S}))$ has the following interpretation: if the outcome of the syndrome measurements specified by $M(\mathbf{S})$ is error-free, then it is a vector in $Im(M(\mathbf{S}))$. Therefore, the minimum distance of $Im(M(\mathbf{S}))$ is a measure of the ability to correct errors in the syndrome.

Theorem 17. *Take any tuple* $\mathbf{S}$ *consisting of* u *subsets* $S_0, \ldots, S_{u-1}$ *of* $[m]$. *The minimum weight of a non-zero element of* $Im(M(\mathbf{S}))$ *is*

$$\min \left\{ 2^{|T|} |\{i : S_i \cap T = \emptyset\}| : T \subset [m], \exists i : T \cap S_i = \emptyset \right\} \tag{130}$$

Proof. First, $Im(M(\mathbf{S})) = Im(M(\mathbf{S})R_m)$, because R_m is invertible.

Next, for each $i \in [u]$,

$$M(S_i)R_m = \otimes_{j=0}^{m-1} \begin{cases} \begin{pmatrix} 1 & 0 \end{pmatrix} & \text{if } j \in S_i \\ \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} & \text{otherwise} \end{cases} \tag{131}$$

Therefore, $M(S_i)R_m$ has the $2^{m-|S_i|}$ columns of $R_{m-|S_i|}$ placed in positions indexed by $T \subset [m]$ such that $T \cap S_i = \emptyset$ and all the remaining columns are zero.

From the above, deduce that the weight of the column of $M(\mathbf{S})R_m$ indexed by $T \subset [m]$ is $2^{|T|}|\{i : S_i \cap T = \emptyset\}|$. Then, the expression (130) gives the minimum weight of a non-zero column of $M(\mathbf{S})R_m$, and, therefore, it is an upper bound on the minimum distance of $Im(M(\mathbf{S}))$. It remains to prove that it is also a lower bound.

Take any collection of v subsets $T_0, \ldots, T_{v-1}$ and consider the sum $\sum_j M(\mathbf{S})R_m e_{T_j}$ of the corresponding columns of $M(\mathbf{S})R_m$ (here e_{T_j} denotes the standard basis vector corresponding to T_j). Suppose without loss of generality that each column $M(\mathbf{S})R_m e_{T_j}$ in the sum is non-zero.

Partition the sets $S_0, \ldots, S_{v-1}$ in two groups depnding on whether $\sum_j M(S_i)R_m e_{T_j}$ is zero or non-zero. Without loss of generality, $\sum_j M(S_i)R_m e_{T_j}$ is non-zero for $i = 0, \ldots, k-1$ and it is zero for $i = k, \ldots u-1$.

Let T be a subset of $[m]$ of minimum size subject to the constraint that it intersects each of $S_k, \ldots, S_{u-1}$ and that there is some $i \in [k]$ such that T does not intersect S_i.

Claim 1: each T_j satisfies the given constraint, so $|T| \leq |T_j|$ for each $j \in [v]$.

Proof of Claim 1: Note that for each i, the non-zero columns of $M(S_i)R_m$ are linearly independent. Then, $\sum_j M(S_i)R_m e_{T_j} = 0$ implies $\forall j : S_i \cap T_j \neq \emptyset$. Deduce that for all $i = k, \ldots, u-1$, for all $j \in [v]$, $S_i \cap T_j \neq \emptyset$. Finally, since each column $M(\mathbf{S})R_m e_{T_j}$ is non-zero, deduce that $\forall j \in [v], \exists i \in [k] : T_j \cap S_i = \emptyset$.

Claim 2: $\left|\sum_j M(\mathbf{S})R_m e_{T_j}\right| \geq 2^{|T|}|\{i : S_i \cap T = \emptyset\}|$.

Proof of Claim 2: Take any $i \in [k]$. $\sum_j M(S_i)R_m e_{T_j}$ is a non-zero linear combination of certain columns of $R_{m-|S_i|}$. Corollary 1 implies that its weight is at least $2^{\min\{|T_j| : T_j \cap S_i = \emptyset\}}$. Claim 1 implies that this is at least $2^{|T|}$. Summing over i proves Claim 2.

Claim 2 implies that the expression (130) is a lower bound on the minimum distance of $Im(M(\mathbf{S}))$, and completes the proof of the theorem. □

5 Examples

This section is divided in three parts.

First, 5.1 considers quantum CSS codes based on standard Reed-Muller codes. The quantum distances of these are half the number of qubits in a syndrome measurement. Except for the few examples with block size ≤ 16 these CSS codes have syndrome measurements on more than 8 qubits.

The other two subsections consider quantum CSS codes based on generalized Reed-Muller codes. These provide much greater flexibility in designing quantum LDPC codes, including codes for which the distance is greater than the syndrome measurement weight.

5.2 gives a number of examples for which $d_x = d_z$. These examples are suitable for noise models that are symmetric with respect to bit flip and phase flip errors, for example the depolarizing channel or the quantum erasure channel.

Finally, 5.3 gives a number of examples with $d_x \neq d_z$. These examples are better suited for biased noise, where either bit flip errors occur with higher probability than phase flip errors or vice versa.

The examples in this section are in the subfamily from section 4; all components of the matrix tuples $\mathbf{H^x}, \mathbf{H^z}$ are $H_i^x = H_i^z = \begin{pmatrix} 1 & 1 \end{pmatrix}$, and the shorthand notation $M(\mathbf{X}), M(\mathbf{Z}), CSS(\mathbf{X}, \mathbf{Z})$ is used.

5.1 Quantum CSS codes based on standard Reed-Muller codes

Note that Theorem 16 implies that for CSS codes obtained from generalized Reed-Muller codes,

$$d_x d_z \leq 2^m, \tag{132}$$

with equality if and only if all elements of $\mathbf{K}$ have the same weight.

Now take[2]

$$\mathbf{X} = \{v \in \{0,1\}^m : |v| = m - r + 1\} \tag{133}$$

$$\mathbf{Z} = \{v \in \{0,1\}^m : |v| = r + 1\} \tag{134}$$

$$\langle \mathbf{S^x} \rangle_{\downarrow} = \{v \in \{0,1\}^m : |v| \leq r - 1\} \tag{135}$$

$$\mathbf{K} = \{v \in \{0,1\}^m : |v| = r\} \tag{136}$$

$$\langle \mathbf{S^z} \rangle_{\uparrow} = \{v \in \{0,1\}^m : |v| \geq r + 1\} \tag{137}$$

Then, $CSS(\mathbf{X}, \mathbf{Z})$ is a $[[2^m, \binom{m}{r}]]$ code with

1. $\binom{m}{r-1}2^{r-1}$ product-of-Pauli-σ_x syndrome measurements on 2^{m-r+1} qubits each.
2. $\binom{m}{r+1}2^{m-r-1}$ product-of-Pauli-σ_z syndrome measurements on 2^{r+1} qubits each.
3. Distances $d_x = 2^{m-r}$ and $d_z = 2^r$.

These codes have been considered previously in in [36] and in [32, Lemma 4.1].

$CSS(\mathbf{X}, \mathbf{Z})$ achieves equality $d_x d_z = 2^m$ in (132), and, moreover, it does so optimally because it uses the entire layer $\{v \in \{0,1\}^m : |v| = r\}$ for logical operators. However, $CSS(\mathbf{X}, \mathbf{Z})$ also has an undesirable property: the weight of syndrome measurements is twice the respective distance.

If it is desired that syndrome measurements involve at most 8 qubits, then only examples with block size ≤ 16 remain. The most interesting of these is the $[[16, 6, 4]]$ quantum code where both $Ker(M(\mathbf{X}))$ and $Ker(M(\mathbf{Z}))$ are the Reed-Muller(2,4) code. This example matches the best possible distance for a $[[16, 6]]$ stabilizer code[3]. The rows of the matrices $M(\mathbf{X}), M(\mathbf{Z})$ specify 16 measurements of 8 qubits each. The spaces of valid syndromes $Im(M(\mathbf{X}))$, $Im(M(\mathbf{Z}))$ are [8,5,2] classical linear codes.

Below, it will be seen how sacrificing part of the layer $\{v \in \{0,1\}^m : |v| = r\}$ for syndrome measurements leads to codes with better trade-off between distance and syndrome measurement weight.

5.2 Examples based on generalized Reed-Muller codes with $d_x = d_z$

5.2.1 Quantum codes based on classical product codes

CSS code from the 2D product of SPC codes This is a CSS code where each of $M(\mathbf{X}), M(\mathbf{Z})$ is isomorphic (by row and column permutation) to the parity check matrix of the 2D product of the $[4, 3]$ single parity check code. For more details, see the running example in Section 3.

[2] In the rest of the paper, $\mathbf{X}, \mathbf{Z}$ are "tuples of subsets" or "tuples of indicator vectors". However, sometimes, it is more convenient to give $\mathbf{X}, \mathbf{Z}$ as "sets of subsets" or "sets of indicator vectors". To convert the latter to the former, the lexicographic order (or any other order) can be used.

[3] Obtained from [15], specifically `http://codetables.de/QECC.php?q=4&n=16&k=6`

CSS code from the 3D product of SPC codes This is a CSS code where each of $M(\mathbf{X}), M(\mathbf{Z})$ is isomorphic (by row and column permutation) to the parity check matrix of the 3D product of the $[8, 7]$ single parity check code. The code appears previously in [27]. The code has parameters $[[512, 174, 8]]$, has 384 syndrome measurements of weight 8 and is obtained by choosing $m = 9$ and

$$\mathbf{X} = \begin{pmatrix} \{0,1,2\} \\ \{3,4,5\} \\ \{6,7,8\} \end{pmatrix} \qquad \mathbf{Z} = \begin{pmatrix} \{0,3,6\} \\ \{1,4,7\} \\ \{2,5,8\} \end{pmatrix} \tag{138}$$

i.e. $\mathbf{X}, \mathbf{Z}$ correspond to the rows and columns of

$$\begin{pmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{pmatrix} \tag{139}$$

The spaces of valid syndromes $Im(M(\mathbf{X}))$, $Im(M(\mathbf{Z}))$ are [192,169,3] classical linear codes.

5.2.2 Examples based on a cyclic pattern

In these examples, X, Z are obtained from several cyclic shifts of the first subset.

A $[[32, 14, 4]]$ code with 24 syndrome measurements of weight 8 is obtained by choosing $m = 5$ and

$$\mathbf{X} = \mathbf{Z} = \begin{pmatrix} \{0,1,3\} \\ \{1,2,4\} \\ \{2,3,0\} \end{pmatrix} \tag{140}$$

The spaces of valid syndromes $Im(M(\mathbf{X}))$, $Im(M(\mathbf{Z}))$ are [12,9,2] classical linear codes.

A $[[64, 8, 8]]$ code with 96 syndrome measurements of weight 8 is obtained by choosing $m = 6$ and

$$\mathbf{X} = \mathbf{Z} = \{013, 124, 235, 340, 451, 502\} \tag{141}$$

$$\mathbf{K} = \{012, 123, 234, 345, 450, 501, 024, 135\} \tag{142}$$

where shorthand notation omitting curly brackets and commas is used for subsets of $\{0, 1, 2, 3, 4, 5\}$ (see also footnote 2). The spaces of valid syndromes $Im(M(\mathbf{X}))$, $Im(M(\mathbf{Z}))$ are [48,28,4] classical linear codes.

A $[[128, 10, 8]]$ with 192 syndrome measurements of weight 8 is obtained by choosing $m = 7$ and

$$\mathbf{X} = \mathbf{Z} = \{013, 124, 235, 346, 450, 561\} \tag{143}$$

$$\mathbf{K} = \{345, 145, 135, 134, 1345, 026, 0256, 0246, 0236, 0126\} \tag{144}$$

The spaces of valid syndromes $Im(M(\mathbf{X}))$, $Im(M(\mathbf{Z}))$ are [96,59,4] classical linear codes.

5.2.3 A further example with block size 128

A $[[128, 24, 8]]$ code with 160 syndrome measurements of weight 8 is obtained by choosing $m = 7$ and

$$\mathbf{X} = \{012, 013, 234, 356, 456\} \tag{145}$$

$$\mathbf{Z} = \{143, 146, 360, 325, 025\} \tag{146}$$

The spaces of valid syndromes $Im(M(\mathbf{X}))$, $Im(M(\mathbf{Z}))$ are [80,52,4] classical linear codes.

5.2.4 Examples with distance greater than the syndrome measurement weight

A $[[256, 6, 16]]$ code with 512 syndrome measurements of weight 8 is obtained by choosing $m = 8$ and

$$\mathbf{X} = \{012, 123, 234, 345, 456, 567, 670, 701\} \tag{147}$$

$$\mathbf{Z} = \{136, 247, 350, 461, 572, 603, 714, 025\} \tag{148}$$

$$\mathbf{K} = \{2367, 1357, 1256, 0347, 0246, 0145\} \tag{149}$$

The spaces of valid syndromes $Im(M(\mathbf{X}))$, $Im(M(\mathbf{Z}))$ are [256,125,8] classical linear codes.

A $[[512, 18, 16]]$ code with 768 syndrome measurements of weigth 8 is obtained by taking $m = 9$,

$$\mathbf{X} = \{012, 345, 678, 048, 156, 237\} \tag{150}$$

$$\mathbf{Z} = \{036, 147, 258, 246, 138, 057\} \tag{151}$$

i.e. $\mathbf{X}$ corresponds to the rows and one set of diagonals, and $\mathbf{Z}$ to the columns and the other set of diagonals of (139). The spaces of valid syndromes $Im(M(\mathbf{X}))$, $Im(M(\mathbf{Z}))$ are [384,247,6] classical linear codes.

5.3 Examples based on generalized Reed-Muller codes with $d_x \neq d_z$

Note that in any of the examples of these section, the role of $\mathbf{X}, \mathbf{Z}$ can be switched, depending on whether bit flip or phase flip errors are more likely in the noise model of interest.

5.3.1 Highly asymmetric CSS codes with distances $2, 2^{m-1}$, $m = 3, 4, \ldots$

Take

$$\mathbf{X} = \{\{0\}\} \tag{152}$$
$$\mathbf{Z} = \{\{0, i\} : i = 1, \ldots, m-1\}\} \tag{153}$$
$$\langle \mathbf{S^x} \rangle_{\downarrow} = \{v \in \{0,1\}^m : v \leq 01 \ldots 1\} \tag{154}$$
$$\mathbf{K} = \{10 \ldots 0\} \tag{155}$$
$$\langle \mathbf{S^z} \rangle_{\uparrow} = \{v \in \{0,1\}^m : \exists i, v \geq \{0, i\}\} \tag{156}$$

Then, $CSS(\mathbf{X}, \mathbf{Z})$ is a $[[2^m, 1]]$ code with

1. 2^{m-1} product-of-Pauli-σ_x syndrome measurements on 2 qubits each.
2. $(m-1)2^{m-2}$ product-of-Pauli-σ_z syndrome measurements on 4 qubits each.
3. Distances $d_x = 2^{m-1}$ and $d_z = 2$.

The spaces of valid syndromes $Im(M(\mathbf{X}))$, $Im(M(\mathbf{Z}))$ are $[2^{m-1}, 2^{m-1}, 1]$ and $[(m-1)2^{m-2}, 2^{m-1}-1, m-1]$ classical linear codes.

5.3.2 Block size 32

A $[[32, 2]]$ code with distances $d_x = 8, d_z = 4$, 48 syndrome measurements of weight 4 and 4 syndrome measurements of weight 8 is obtained by choosing:

$$\mathbf{X} = \{\{0, 1\}, \{2, 3, 4\}\} \tag{157}$$
$$\mathbf{Z} = \{\{0, 2\}, \{1, 3\}, \{0, 4\}, \{1, 4\}, \{1, 3\}\} \tag{158}$$
$$\mathbf{K} = \{\{0, 3\}, \{1, 2\}\} \tag{159}$$

The spaces of valid syndromes $Im(M(\mathbf{X}))$, $Im(M(\mathbf{Z}))$ are [12,11,2] and [40,19,4] classical linear codes.

Another $[[32, 2]]$ code with distances $d_x = 8, d_z = 4$ is obtained by choosing:

$$\mathbf{X} = \{\{0, 1, 4\}, \{2, 3, 4\}\} \tag{160}$$
$$\mathbf{Z} = \{\{0, 2\}, \{1, 3\}, \{4\}\} \tag{161}$$
$$\mathbf{K} = \{\{0, 3\}, \{1, 2\}\} \tag{162}$$

This code has 8 syndrome measurements of weight 8, 16 syndrome measurements of weight 4, and 16 syndrome measurements of weight 2. The spaces of valid syndromes $Im(M(\mathbf{X}))$, $Im(M(\mathbf{Z}))$ are [8,7,2] and [32,23,3] classical linear codes.

It turns out that this code has the same stabilizer as a toric code.[4] This can be seen by a sequence of row operations that transform the layers $M(Z_0), M(Z_1)$

[4] The author would like to thank the anonymous reviewer who pointed out that a refinement of the tesselation of the torus used in Figure 1c leads to a $[[32, 2, d_x = 8, d_z = 4]]$ toric code.

of the matrix $M(\mathbf{Z})$. The row operations are conveniently described using the shorthand notation $h = \begin{pmatrix} 1 & 1 \end{pmatrix}, e_0^T = \begin{pmatrix} 1 & 0 \end{pmatrix}, e_1^T = \begin{pmatrix} 0 & 1 \end{pmatrix}, I = e_0 e_0^T + e_1 e_1^T$. Now, note that

$$M(Z_0) + (h \otimes I \otimes h \otimes I \otimes e_1)\, M(Z_2) = h \otimes I \otimes h \otimes I \otimes \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \tag{163}$$

which is just two copies of $h \otimes I \otimes h \otimes I \otimes e_0^T$. Similarly,

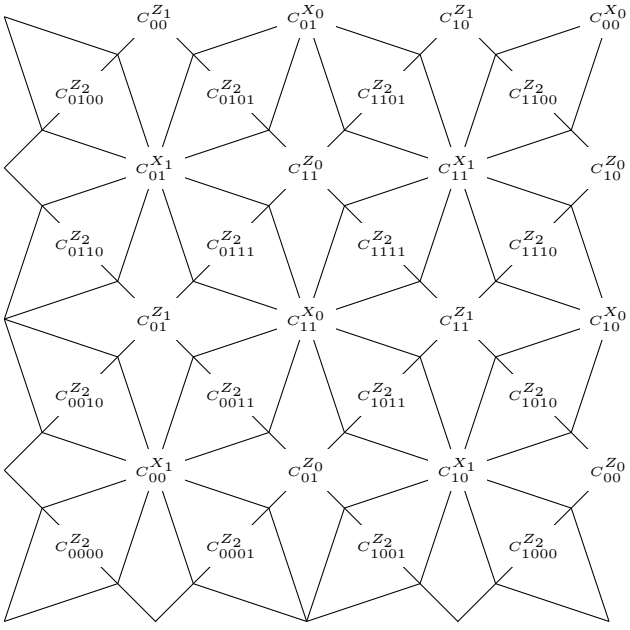
$$M(Z_1) + (I \otimes h \otimes I \otimes h \otimes e_0)\, M(Z_2) = I \otimes h \otimes I \otimes h \otimes \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} \tag{164}$$

which is just two copies of $I \otimes h \otimes I \otimes h \otimes e_1^T$. Thus, $M(\mathbf{Z})$ is equivalent by row operations to

$$M'(\mathbf{Z}) = \begin{pmatrix} M'(Z_0) \\ M'(Z_1) \\ M(Z_2) \end{pmatrix} = \begin{pmatrix} h \otimes I \otimes h \otimes I \otimes e_0^T \\ I \otimes h \otimes I \otimes h \otimes e_1^T \\ I \otimes I \otimes I \otimes I \otimes h \end{pmatrix} \tag{165}$$

The two parity check matrices $M(\mathbf{X})$ and $M'(\mathbf{Z})$ specify a Tanner graph that can be embedded in the torus; this is shown in figure 3.

Figure 3: Tanner graph of the toric code equivalent to $CSS(\{014, 234\}, \{02, 13, 4\})$. For $a, b, c \in \{0, 1\}$, $C_{bc}^{X_a}$ denotes the check associated to row bc of $M(X_a)$, and $C_{bc}^{Z_a}$ denotes the check associated to row bc of $M'(Z_a)$. For $a, b, c, d \in \{0, 1\}$, $C_{abcd}^{Z_2}$ denotes the check associated to row $abcd$ of $M(Z_2)$. For $a, b, c, d, e \in \{0, 1\}$, qubit Q_{abcde} (not labelled in the picture) is the node between $C_{abcd}^{Z_2}$ and the nearest node of type C^{Z_e}.



Note however that $M'(\mathbf{Z})$ has columns of weight 2, so the ability to correct one error in the bit flip syndrome is lost when $M'(\mathbf{Z})$ is used instead of $M(\mathbf{Z})$.

5.3.3 Block size 128

A $[[128, 3]]$ code with distances $d_x = 8, d_z = 16$ and with 224 syndrome measurements on 8 qubits each is obtained by choosing

$$\mathbf{X} = \{013, 124, 235, 346, 450, 561, 602, 134\} \tag{166}$$

$$\mathbf{Z} = \{013, 124, 235, 346, 450, 561\} \tag{167}$$

$$\mathbf{K} = \{0246, 0236, 0126\} \tag{168}$$

The spaces of valid syndromes $Im(M(\mathbf{X}))$, $Im(M(\mathbf{Z}))$ are [128,66,8] and [96,59,4] classical linear codes.

6 Conclusion and future work

The present paper introduced intersecting subset codes, established a number of useful properties, and gave many examples with small and moderate block sizes.

One direction for future work concerns algorithms that prescribe a corretion based on the syndrome. Since intersecting subset codes have connections to classical LDPC, polar, and Reed-Muller codes, there are many possible low-complexity decoding algorithms that can potentially be applied. It would be interesting to evaluate the various options empirically and to determine which gives the best performance.

A second direction for future work concerns the performance of intersecting subset codes when used on noisy near term quantum hardware. As already explained, for a large subfamily of intersecting subset codes it is possible to calculate the distances for both data and syndrome errors, and to construct a large number of interesting examples. A next step could be an investigation of the behavior of these examples under the standard error model for fault tolerance, in which each elementary gate in the syndrome measurement circuit is followed by independent Pauli errors.

A third direction for future work concerns further exploration of the structure of intersecting subset codes. As explained in the introduction, the code design was inspired by the relation between Gallager's LDPC codes and classical products of single parity check codes. There was no a priori reason to expect the connection to Reed-Muller codes in section 4 or the two examples of intersecting subset codes that turn out to also be toric codes. These coincidences indicate that intersecting subset codes have rich and interesting structure that may hold further surprises.

Acknowledgements

This work was supported by the QuantERA II Programme, through the project Error Correction for Quantum Information Processing (EQUIP). This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement no. 731473 and 101017733.

The author would like to thank Pradeep Sarvepalli and two anonynous reviewers for the detailed and constructive comments and useful suggestions for improvement.

References

[1] Erdal Arikan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, 55(7):3051–3073, July 2009.

[2] Alexei Ashikhmin, Ching-Yi Lai, and Todd A. Brun. Quantum data-syndrome codes. *IEEE Journal on Selected Areas in Communications*, 38(3):449–462, 2020.

[3] Sergey Bravyi, Andrew W Cross, Jay M Gambetta, Dmitri Maslov, Patrick Rall, and Theodore J Yoder. High-threshold and low-overhead fault-tolerant quantum memory. *Nature*, 627(8005):778–782, 2024.

[4] Nikolas P. Breuckmann and Jens N. Eberhardt. Balanced product quantum codes. *IEEE Transactions on Information Theory*, 67(10):6653–6674, 2021.

[5] Nikolas P. Breuckmann and Jens Niklas Eberhardt. Quantum low-density parity-check codes. *PRX Quantum*, 2(4), oct 2021.

[6] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane. Quantum error correction and orthogonal geometry. *Phys. Rev. Lett.*, 78:405–408, Jan 1997.

[7] Mustafa Cemil Coşkun, Gianluigi Liva, Alexandre Graell i Amat, Michael Lentmaier, and Henry D. Pfister. Successive cancellation decoding of single parity-check product codes: Analysis and improved decoding. *IEEE Transactions on Information Theory*, 69(2):823–841, Feb 2023.

[8] Nicolas Delfosse and Adam Paetznick. Spacetime codes of clifford circuits, 2023.

[9] Irit Dinur, Min-Hsiu Hsieh, Ting-Chun Lin, and Thomas Vidick. Good quantum ldpc codes with linear time decoders. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, page 905–918, New York, NY, USA, 2023. Association for Computing Machinery.

[10] P. Elias. Error-free coding. *Transactions of the IRE Professional Group on Information Theory*, 4(4):29–37, Sep. 1954.

[11] Omar Fawzi, Antoine Grospellier, and Anthony Leverrier. Constant overhead quantum fault-tolerance with quantum expander codes. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 743–754, 2018.

[12] Robert G. Gallager. *Low-Density Parity-Check Codes*. The MIT Press, 09 1963.

[13] Daniel Gottesman. Class of quantum error-correcting codes saturating the quantum hamming bound. *Phys. Rev. A*, 54:1862–1868, Sep 1996.

[14] Daniel Gottesman. Fault-tolerant quantum computation with constant overhead. *Quantum Information & Computation*, 14(15-16):1338–1372, 2014.

[15] Markus Grassl. Bounds on the minimum distance of linear codes and quantum codes. Online available at `http://www.codetables.de`, 2007. Accessed on 2024-03-07.

[16] Shouzhen Gu, Christopher A. Pattison, and Eugene Tang. An efficient decoder for a linear distance quantum ldpc code. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, page 919–932, New York, NY, USA, 2023. Association for Computing Machinery.

[17] Matthew B. Hastings, Jeongwan Haah, and Ryan O'Donnell. Fiber bundle codes: breaking the n1/2 polylog(n) barrier for quantum ldpc codes. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, page 1276–1288, New York, NY, USA, 2021. Association for Computing Machinery.

[18] Morteza Hivadi. On quantum spc product codes. *Quantum Information Processing*, 17(12):1–14, 2018.

[19] E. Knill, R. Laflamme, and W. Zurek. Threshold accuracy for quantum computation, 1996.

[20] Alexey A. Kovalev and Leonid P. Pryadko. Quantum kronecker sum-product low-density parity-check codes with finite rate. *Phys. Rev. A*, 88:012311, Jul 2013.

[21] Anthony Leverrier and Gilles Zémor. *Efficient decoding up to a constant fraction of the code length for asymptotically good quantum codes*, pages 1216–1244.

[22] Anthony Leverrier and Gilles Zémor. Quantum tanner codes. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 872–883, Oct 2022.

[23] D.J.C. MacKay, G. Mitchison, and P.L. McFadden. Sparse-graph codes for quantum error correction. *IEEE Transactions on Information Theory*, 50(10):2315–2330, Oct 2004.

[24] Florence Jessie MacWilliams and Neil James Alexander Sloane. *The theory of error-correcting codes*, volume 16. Elsevier, 1977.

[25] D. E. Muller. Application of boolean algebra to switching circuit design and to error detection. *Transactions of the I.R.E. Professional Group on Electronic Computers*, EC-3(3):6–12, Sep. 1954.

[26] Andrew Nemec. Quantum data-syndrome codes: Subsystem and impure code constructions. *Quantum Information Processing*, 22(11):408, 2023.

[27] Dimiter Ostrev, Davide Orsucci, Francisco Lázaro, and Balazs Matuz. Classical product code constructions for quantum Calderbank-Shor-Steane codes. *Quantum*, 8:1420, July 2024.

[28] Pavel Panteleev and Gleb Kalachev. Degenerate Quantum LDPC Codes With Good Finite Length Performance. *Quantum*, 5:585, November 2021.

[29] Pavel Panteleev and Gleb Kalachev. Asymptotically good quantum and locally testable classical ldpc codes. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, page 375–388, New York, NY, USA, 2022. Association for Computing Machinery.

[30] I. Reed. A class of multiple-error-correcting codes and the decoding scheme. *Transactions of the IRE Professional Group on Information Theory*, 4(4):38–49, Sep. 1954.

[31] T.J. Richardson and R.L. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Transactions on Information Theory*, 47(2):599–618, Feb 2001.

[32] Pradeep Kiran Sarvepalli, Andreas Klappenecker, and Martin Rötteler. Asymmetric quantum codes: constructions, bounds and performance. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 465(2105):1645–1672, 2009.

[33] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, Oct 1995.

[34] M. Sipser and D.A. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996.

[35] A. M. Steane. Error correcting codes in quantum theory. *Phys. Rev. Lett.*, 77:793–797, Jul 1996.

[36] A.M. Steane. Quantum reed-muller codes. *IEEE Transactions on Information Theory*, 45(5):1701–1703, 1999.

[37] Jean-Pierre Tillich and Gilles Zémor. Quantum ldpc codes with positive rate and minimum distance proportional to the square root of the block-length. *IEEE Transactions on Information Theory*, 60(2):1193–1202, 2014.